Exploring a Modelling Method with Semantic Link Network and Resource Space Model

MUHAMMAD ADNAN RAFI

Doctor of Philosophy

Aston University

November 2022

Abstract

To model the complex reality, it is necessary to develop a powerful semantic model. A rational approach is to integrate a relational view and a multi-dimensional view of reality. The Semantic Link Network (SLN) is a semantic model based on a relational view and the Resource Space Model (RSM) is a multi-dimensional view for managing, sharing and specifying versatile resources with a universal resource observation.

The motivation of this research consists of four aspects: (1) verify the roles of Semantic Link Network and the Resource Space Model in effectively managing various types of resources, (2) demonstrate the advantages of the Resource Space Model and Semantic Link Network, (3) uncover the rules through applications, and (4) generalize a methodology for modelling complex reality and managing various resources.

The main contribution of this work consists of the following aspects:

1. A new text summarization method is proposed by segmenting a document into clauses based on semantic discourse relations and ranking and extracting the informative clauses according to their relations and roles. The Resource Space Model benefits from using semantic link network, ranking techniques and language characteristics. Compared with other summarization approaches, the proposed approach based on semantic relations achieves a higher recall score. Three implications are obtained from this research.

2. An SLN-based model for recommending research collaboration is proposed by extracting a semantic link network of different types of semantic nodes and different types of semantic links from scientific publications. Experiments on three data sets of scientific publications show that the model achieves a good performance in predicting future collaborators. This research further unveils that different semantic links play different roles in representing texts.

3. A multi-dimensional method for managing software engineering processes is developed. Software engineering processes are mapped into multiple dimensions for supporting analysis, development and maintenance of software systems. It can be used to uniformly classify and manage software methods and models through multiple dimensions so that software systems can be developed with appropriate methods. Interfaces for visualizing Resource Space Model are developed to support the proposed method by keeping the consistency among interface, the structure of model and faceted navigation.

Keywords – Resources Space Model; Semantic Link Network; Resource Management; Classification Dimensions; Software Engineering Models; Probabilistic Graphical Model; Discourse Structure; Text Summarization

Acknowledgements

I would like to thank the following people, without whom I would not have been able to complete this research work.

First of all, I would like to express my best regards and thanks to my supervisor Professor Albert Hai Zhuge for giving me the opportunity to undertake this research study and to persevere and complete it satisfactorily. Without his valuable guidance, this achievement would not have been possible. My deep thanks to my Associate Supervisor Dr Xiaorui Jiang for his advice and guidance throughout the duration of this study. I have been very lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly.

I would particularly like to thanks the research and administrative staff at the School of Engineering and Applied Sciences for their support during my study. Many thanks to Kanchan Patel, Sandra Mosley, and Helen Yard.

I would like to thanks to my friend Mr Muhammad Babar Ali Khan for his frequent help, support and encouragement. For that, I am eternally grateful. I am truly lucky to count you amongst my friends Dr Mazhar Hussain Malik, thank you for all the support and the encouragement. I am forever thankful for their friendship and support, and for creating a cordial working environment.

Finally, I would like to say thanks to my family, parents and grandparents for their infinite love, care and support. I wish to thank all the people whose assistance was a milestone in the completion of this research project.

The work in the thesis would not have been productive without the support of my parents, teachers and friends.

Table of Contents

Abstract 2

Acknowledgements 4

Table of Contents 5

List of tables, figures and equations 8

List of Abbreviations 10

1 CHAPTER ONE: INTRODUCTION 13

1.1 Background 13

1.2 Resource Space Model 14

1.2.1 Basic definition of Resource Space Model 15

1.3 Semantic Link Network 18

1.4 Research Contributions 19

1.5 Motivation for this study 20

1.6 Organisation of this thesis 20

2 CHAPTER TWO: A LITERATURE REVIEW 23

2.1 An Overview of the Text Summarisation 23

2.2 EFFORTS TO INCORPORATE SEMANTICS INTO PROBABILISTIC GRAPHICAL MODEL25

2.3.1 Associating Probabilistic Graphical Model with Semantics 34

2.3.2 EFFORTS TO INCORPORATE UNCERTAINTY INTO GRAPH-BASED KNOWLEDGE REPRESENTATION 37

2.3.3 Discussion 43

2.3.3 To Sum Up 49

2.4 Software Engineering Approaches Background 50

2.4.1 The Plan Driven Approach 54

2.4.1 The Agile Approach 55

2.5 Summary 58

3 CHAPTER THREE: EFFECTIVE SUMMARIZATION BASED ON SEMANTIC LINKS 59

3.1 Introduction 59

3.2 Related Work 60
3.3 Proposed Model 61
3.4 Extract and parse into RST tree 61
3.5 Choose informative text clauses 62
3.6 Refine summary by ranking on Semantic Link Network 63
3.7 Evaluate the performance of summaries....................64
3.8 Dataset 64
3.9 Experimental Results 65

4 CHAPTER FOUR: RECOMMENDING RESEARCH COLLABORATORS BASED ON SEMANTIC LINK 67
4.1 Introduction....................67
4.2 Related Work 68
4.3 Building the SLN 69
4.4 Semantic Paths 70
4.5 The Framework of Recommending Future Collaborators 72
4.6 Experiment – Dataset 73
4.7 Experiment Setting....................73
4.8 Results and Analysis....................74

5 CHAPTER FIVE: A MULTI-DIMENSIONAL METHOD SPACE FOR MANAGING SOFTWARE PROCESSES....................77
5.1 Introduction....................77
5.2 RSM use to Manage Multiple Types of Software Processes from Multiple Dimensions 77
5.3 Property Inventory Tracker 80
5.4 Eclipse Project 83

6 CHAPTER SIX: VISUALIZATION OF MULTI-DIMENSIONAL RESOURCE SPACE MODEL .87
6.1 Introduction....................87
6.2 Architecture....................89
6.3 Technology....................90
6.4 Interface Implementation 91
6.5 Test....................93

7 CHAPTER SEVEN: CONCLUSIONS AND FUTURE WORK....................97

Appendices....................99

6 References..100

List of tables, figures and equations

Table 2.1 Six common kinds of semantic networks 40

Table 2.2 Brief Comparison of Models. 48

Table 2.3 Various types of the software applications. 51

Table 3.1 The Words Amount of Papers in Dataset 64

Table 3.2 The Performance of RST Summary. 65

Table 3.3 The Performance of Different Models 65

Table 3.4 The Examples of Proposed model. 65

Table 4.1 Five semantic paths. 71

Table 4.2 Details of three datasets. 74

Table 4.3 Results of experiment. 75

Table 4.4 Details of SLNS. 76

Figure 1-1 Search in Multi-dimensional classifications map [3]. 15

Figure 1-2 Email Resource Space example [135] 16

Figure 1-3 Thesis modules and research workflow. 21

Figure 2-1 A basic structure of an ATS. [143] 24

Figure 2-2 A Bayesian network for Example 1. 28

Figure 2-3 An example of Markov random field. 31

Figure 2-4 Propositions, first-order formulas, constants, and ground MRF in Example 2. 36

Figure 2-5 A Semantic Link Network of a paper [53]. 43

Figure 2-6 A timeline of the efforts 45

Figure 2-7 Software Engineering Models 53

Figure 2-8 Waterfall Model 54

Figure 2-9 Incremental Model 55

Figure 2-10 Extreme Programming Model 57

Figure 2-11 Scrum Model 58

Figure 3-1 SLN-based extractive summarization pipeline 61

Figure 3-2 An example of RST 62

Figure 4-1 The schema of the initial SLN 69

Figure 4-2 Training part and test part. 71

Figure 4-3 Framework of the Model. 72

Figure 5-1 Software Engineering Processes of the Property Inventory Tracker in RSM. 81

Figure 5-2 Eclipse Project in Multi-Dimensional Resource Space. 84

Figure 6-1 An example of 3-Dimensional Resource Space. 88

Figure 6-2 Architecture for visualization 90

Figure 6-3 User Input Form. 92

Figure 6-4 Circular view of inputs. 92

Figure 6-5 Dimensions with attributes for letting agencies. 93

Figure 6-6 Comparison with other Interfaces. 96

List of Abbreviations

RS	Resource Space
RSM	Resource Space Model
SN	Semantic Network
SLN	Semantic Link Network
PSLN	Probabilistic Semantic Link Network
ANNs	Artificial Neural Networks
AI	Artificial Intelligence
NLP	Natural Language Processing
TS	Text Summarization
PGM	Probabilistic Graphical Model
FOL	First Order Logic
MRFs	Markov Random Fields
BNs	Bayesian Networks
MLN	Markov Logic Network
PSL	Probabilistic Soft Logic
OWL	Web Ontology Language
RDF	Resource Description Framework
ADF	Active e-Document Framework
KG	Knowledge Graph
DAG	Directed Acyclic Graph
CRF	Conditional Random Field
GBKR	Graph-Based Knowledge Representation

CPSoSLN	Cyber-Physical-Social Semantic Link Network
GRASP	GRowth-based Approach with Staged Pruning
HHMM	Hierarchical Hidden Markov Model
HBN	Hierarchical Bayesian Network
SMEs	Small and Medium Enterprises
GBKR	Graph-based Knowledge Representation
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
PHP	Hypertext Preprocessor
JS	JavaScript
D3	Data-Driven Documents
SVG	Scalable Vector Graphics
MIT	Massachusetts Institute of Technology
ACL	Association for Computational Linguistics
RUP	Rational Unified Process
OOP	Object Oriented Programming
SE	Software Engineering
XP	Extreme Programming
IXP	Industrial Extreme Programming
PIT	Property Inventory Tracker
IDE	Integrated Development Environment
SDK	Software Development Kit
JDT	Java Development Tools

SRS	Software Requirements Specification
PMC	Project Management Committee
API	Application Programming Interface
PDE	Plug-in Development Environments
SDD	Software Design Document
TPTP	Test and Performance Tools Platform
RST	Rhetorical Structure Theory
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
CPS	Cyber-Physical Society
URL	Uniform Resource Locator
RDBM	Relational Database Mode
ROM	Read Only Memory
PC	Personal Computer
SDLC	Software Development Life Cycle
ATS	Automatic Text Summarisation
TF-IDF	Term Frequency – Inverse Dense Frequency

1 CHAPTER ONE: INTRODUCTION

1.1 Background

Classification is an elementary methods for humans to identify resources. Single dimensional classification is generally used in real life. For example, shops classify and arrange the stock according to customers purchasing habit and for their facility. The computer file system is another example of the single-dimensional classification. There are some scenarios where we need more than one dimension to classify resources. For example, Two-dimensional space or three-dimensional space is used to locate the location on the map. The web browsing is another example of the multidimensional space. Humans recognized classification spaces in minds throughout the learning process. They have established consensus on classification on physical sections and social functions.

The Resource Space Model (RSM) is a resource organization model which has a complete method, theory and model based on multi-dimensional classifications. RSM is used to establish an appropriate classification space for the resources. The resources managed by the RSM could be any form. The classification nature of the RSM enables it to automatically generate the resource space by using classification or clustering algorithms on a given set of resources.

A systematic theory on RSM has been published. These studies and formal characteristics make it valiant to adopt it. One purpose of this study is to assess the extent to which these factors and solutions can solve the real-world problems. RSM is a multi-dimensional, content-based and classification-based space model for efficiently and effectively managing various resources. RSM is a non-relational data model, which has a relatively complete theoretical foundation and has significant applications in cyber-physical society and faceted search. There are some applications in picture resources and email resources [135] that are presented by the implementation of the RSM. The email resources application is described in section 1.2 for further consideration which shows the relationship between RSM and SN help to build the application. Emails can be automatically added into the email resource space through semantic relationships [135].

As a multi-dimensional semantic model for uniformly managing various resources, RSM can be used to uniformly classify and manage software methods and models through multiple dimensions so that software systems can be developed with appropriate methods. This research will apply the RSM to classify existing software engineering models and features from multiple dimensions so that software methods can be utilized in the most effective way to build real-time systems. Real-time system respond to external events in a timely fashion and the response time is guaranteed. Real-time systems are defined as those systems in which the overall correctness of the system depends on both the functional correctness and the timing correctness. Examples are, Multimedia streaming DVD Player must decode both the audio and the video streams from the disk simultaneously [146].

Two cases of applying the Resource Space Model in managing the software engineering projects are studied and show that RSM can map the software engineering activities into one space for easy management. New dimensions and coordinates can be easily adopted for managing new features found during the software process. It offers a new way to manage the whole lifecycle of the project.

A Resource Space (RS) is used for accurately finding and efficiently organizing resources that can also be the integration of the Resource Space Model and the Semantic Link Network. An SLN can be constructed from text, where the nodes represent text units and the semantic links signify semantic relations between language units. SLN was used to summarize texts by extracting diverse semantic links from texts and then ranking the network. Therefore, a multi-dimensional resource space model can represent both link semantics and classification semantics [81].

Modelling is a basic but vital task in computer science. Graph model, uncertainty, and semantics are three key aspects for building a comprehensible, credible, and efficient model of the real world [5][6]. They enable computers to represent and analyse information logically, discover new relationships and generate new knowledge, in a complex environment with unknown, imperfect and even contradictory information. The combination of these three aspects can provide a systematic and effective method for modelling the reality. This research presents the efforts towards combining these aspects furthermore Visualization of multi-dimensional Resource Space Model can provide a user friendly interface when applying it to manage various resources [2].

1.2 Resource Space Model

An important part of human intelligence is to think and understand the real world through dimensions. Automatically discovering of the dimensions from a set of resources provides the multi-dimensional observations for operating resources effectively and efficiently. A multi-dimensional approach can be a space of methodologies for accomplishing, mapping, coordinating and combining procedures as well as leading the application procedure to build and predict the development process. Figure 1.1 is an example of using the resource space model. Users can search accommodations from four dimensions: time, region, function and location, which can reduce the search space prominently.

"Classification is one of the most basic methods for humans to recognize and organize various resources. The Resource Space Model is a resource organization model based on multi-dimensional classifications." (P-211) [3].

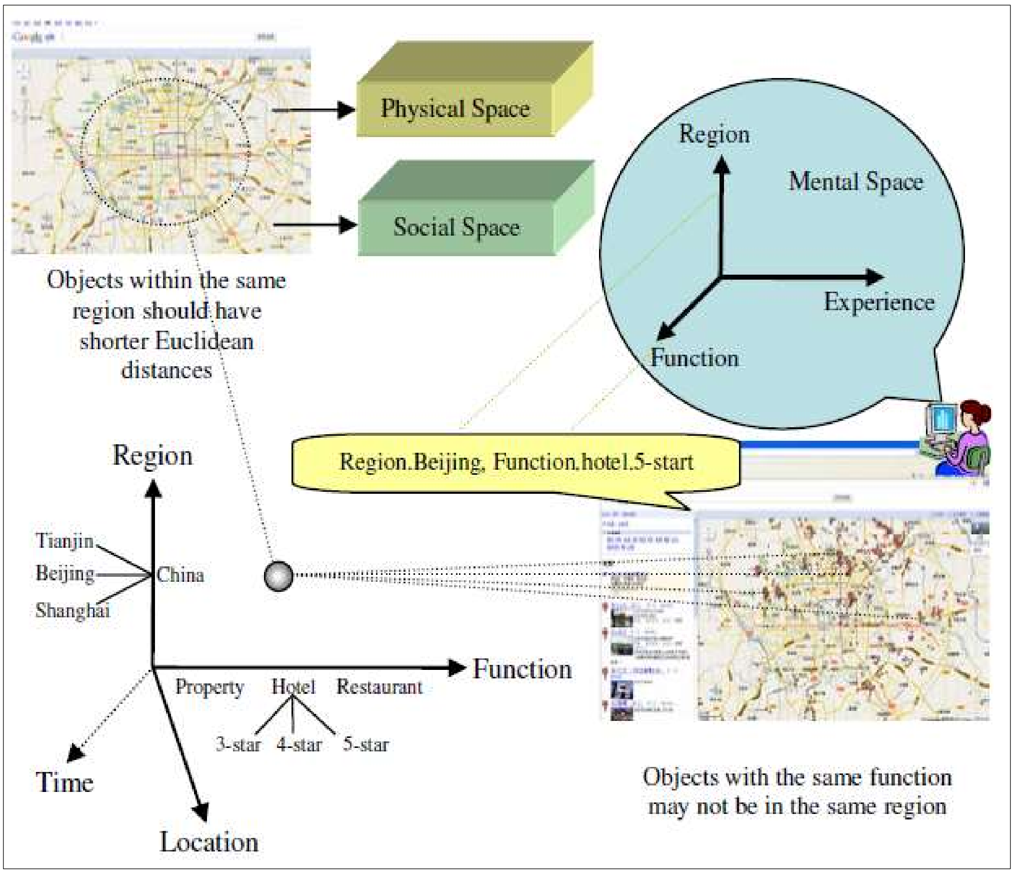

Physical Space
Social Space
Region
Mental Space
Experience
Function
Objects within the same region should have shorter Euclidean distances
Region.Beijing, Function.hotel.5-start
Region
Tianjin
Beijing
Shanghai
China
Property
Hotel
Restaurant
Function
3-star
4-star
5-star
Time
Location
Objects with the same function may not be in the same region

Figure 1-1 Search in Multi-dimensional classifications map [3].

1.2.1 Basic definition of Resource Space Model

The Resource Space Model (RSM) is used to manage versatile resources with a multi-dimensional classification space. It supports generalization and specialization on multidimensional classifications [1-16]. Resource Space RS of n-dimensional characterizes n kinds of classification method X_1, X_2, … , and X_n on a set of resources represented as RS(X_1, X_2, … , X_n). The resource set denoted by dimension X_i ($1 \leq i \leq n$) is the union of the resource sets represented by all of its coordinates denoted as $R(X_i)=R(C_{i1}) \cup R(C_{i2}) \cup \ldots \cup R(C_{im})$, in simple $X_i = \{C_{i1}, C_{i2}, \ldots, C_{im}\}$ [3][4]. One coordinate at every axis specifies a point p in the space represented as $p(C_1,j_1, C_2,j_2, \ldots, C_n,j_n)$ or $(C_1,j_1, C_2,j_2, \ldots, C_n,j_n)$. $R(p)=R(C_1,j_1) \cap R(C_1,j_2) \cap \ldots \cap R(C_1,j_n)$ [2]. RSM has been integrated with the Semantic Link Network model to form a semantic space with the features of self-organization and multi-dimensional classification [34-36]. Therefore, RSM is based on classification and links, which are elementary mechanisms in forming and evolving spaces [11][12].

RSM has further been utilized to manage various resources in complex space by coordinating the physical space, cyber space, mental space and social space [1] [4]. RSM has been developed into a methodology for managing various resources from multiple dimensions [4]-[17].

RSM is a multi-dimensional, content-based and classification-based space model for efficiently and effectively managing various resources. RSM is a non-relational data model, which has a relatively complete theoretical foundation and has significant applications in cyber-physical society and faceted search. There are some applications in picture resources and email resources [135] that are presented by the implementation of the RSM.

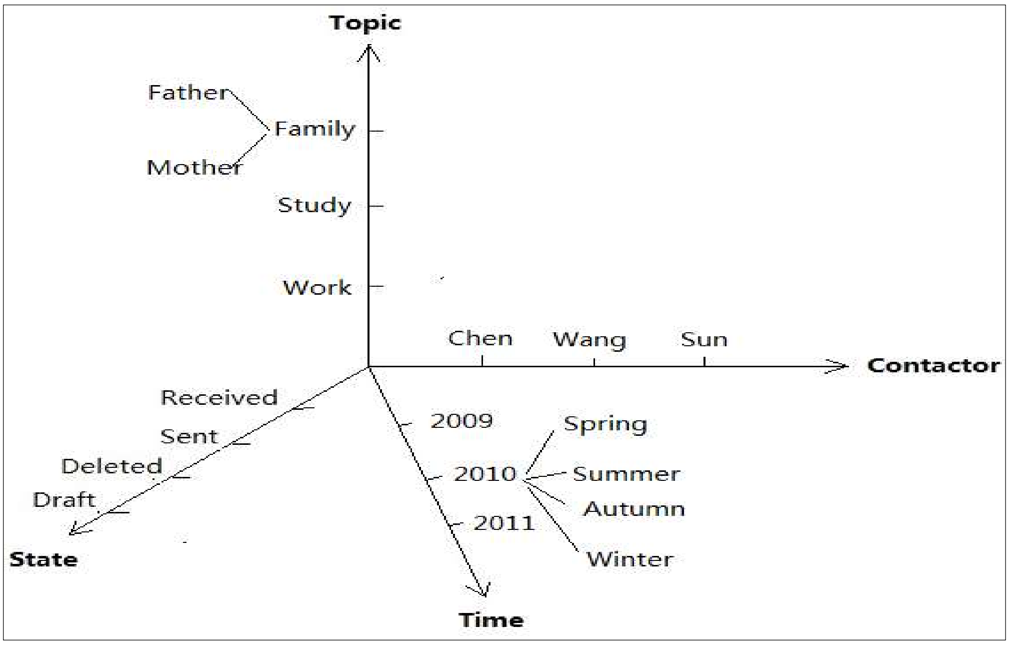


Figure 1-2 Email Resource Space example [135]

Electronic mail (e-mail or email) is a way of communication between individuals using electronic devices. Now a days email is became a universal method for communication. Email address is an elementary and essential part of numerous processes in commerce, business, administration, training, entertainment, and other daily life routines. Every day we communication through emails and this number is increasing with the passage of the time. How to competently attain the amount of emails becomes a question to be addressed. Figure 1.2 shows an email resource space model, the dimensions of which are Contactor, Topic, Time, and State. Time and Topic are hierarchical dimensions. All of these four dimensions are orthogonal with each other, i.e. R(Topic)=R(Contactor)=R(State)=R(Time) [135]. Each resource could be acquired from any of the dimensions. When a new email is added to the resource space, its corresponding coordinates on each axis should be specified. Emails can be automatically added into the email resource space through semantic relationships. For example, it is practical that emails are categorized and added into the same topic. Responding emails have strong relation with initiating emails, so they should be in the same category [135]. Towards the future interconnection environment ideal, a complex semantic space is proposed by integrating the RSM with the Semantic Link Network to utilize advantages of the both models [135].

A. Dimension

Humans created numerous spaces while living in the real world which helps them to observe and think through different aspects. Physical spaces include homes and buildings, parks and roads, and cities. Managing and recognizing physical objects humans create dimensions of space such as location and time [135][136].

A set of objects of the dimensions can have both subjective view and the objective view [4]. The subjective view is the way to represent and recognize real-world objects. For example, clients and customers write reviews related to buying products such as quality, price and service. The natural features of a set of objects are presented by the objective dimension such as colour, size and shape of the product. Subjective dimensions could be changed as dependent on the human observations while the objective dimensions are comparatively stable.

One concept that is closely associated to the dimension is the category that philosophy, linguistics, mathematics and computer science define it in different ways. Aristotle defined ten categories [4] to classify all the major things. The prime categories are substance, quality, quantity and relation. The subordinate categories are time, place, action, situation, and passion. They further reduce the ten categories into five categories which are relation, substance, motion, quantity, and quality. Finally, they are reduced to two categories: relation and substance. In the twentieth century some philosophers started to move from the metaphysics of categorization in the direction of the linguistic problem of describing the words being used. A category and a sub-category can’t be the two dimensions for the same space.

B. Dimension and Resource Space

A dimension is a partitioning of a set of objects from the algebra point of view. In text summarization, various dimensions are used to establish efficient operations. For Example, in the Vector Space Model each dimension represents a word with a weight [4]. In the resource space model (RSM) different dimensions denote the different categories of different procedures of categorization on the identical set of representations. Multiple dimensions that are dependent on each other with the same interest space. There are two main scenarios in which we need to expand the existing dimensions. The first one is that the existing dimension does not meet the current requirements of the application. The second one is that category hierarchy is general for the construction of the resource space. The resource space needs to be adapted when the resources change quickly.

A Resource Space (RS) is different from conventional distance space because it is a multi-dimensional classification space where the dimensions (axes) and coordinates are discrete [136]. A coordinate on a dimension presents an abstract class of a set of resource with a definite classification technique. In order to develop and manage the resources more rigorously and accurately, the integrity constraints and normal forms theory of the RSM was proposed [136]. To make the future interconnection environment ideal, a complex semantic space is proposed by integrating the RSM with the semantic link network to take advantage of both models [135].

C. Multi-dimensional Evaluation

A systematic determination of the merit, value and significance of a method is based on the evaluation by using the comparison with the existing methods or the objective criteria [4] and common practice where formal methods are not appropriate. Previous evaluation approaches are based on human experience and focus on the result. In the text summarization application, criteria-based method is used to evaluate the internal structure (e.g. coverage and coherence) and its impact on the summary.

1.3 Semantic Link Network

The Semantic Link Network (SLN) is a graph-based semantic model, which was initially proposed for managing Web resources. An instance model of SLN mainly consists of four components: semantic nodes, semantic links, rules on semantic links, and operations performing on nodes and semantic links. Different from the Semantic Net, the SLN model supports self-organization operations of a complex system, emerging semantics, and automatic discovery of semantic links [95].

A. Uncertainty on semantic link

There are various types of the semantic links that helps understanding and representation. In general, the semantic links are uncertain between the natural language representations due to the following reasons [4]: (1) people often misunderstand the meaning and representations while chatting online and when emailing; (2) in natural language representation there are several way to present the same meaning; (3) different writers have different knowledge of content and different ways to present it; (4) readers have different level of knowledge of content on different topics and different level of understating; and, (5) different cultures and different fields of a subject have different effects on understating both writers and readers [4].

1.4 Research Contributions

Classification and link are the most basic methods for humans to know, organize and manage resources in multiple spaces i.e., socio space, physical space, mental space and cyber space. It is a fundamental approach to creating a semantic space based on classification and links for organizing resources in various spaces. The complex semantic space integrating the Resource Space Model and Semantic Link Network based on multidimensional classifications is a promising model for managing various resources. Semantic Link Network, which is more capable of modelling semantics than traditional graph structure [53] a study has been presented. We adopt the SLN and present the effective way of summarization and chapter 3 and chapter 4.

Previous studies have specified [18], there is no single software engineering method that fulfils all requirements and present a methodology that is most suitable to build a specific type of application [31]. RSM develops new approaches and methods for achieving the ultimate goals which is presented in the chapter 5. The research work contribution is based on analysis of the challenges facing the present software engineering and text summarization research with an indication of the problems addressed in the current work and those contributed to their solution by adopting the SLN and RSM.

The main contribution of this work consists of the following aspects:

1. An SLN-based text summarization method is proposed by segmenting a document into clauses based on semantic discourse relations and ranking and extracting the informative clauses according to their relations and roles. The model benefits from using semantic link network, ranking techniques and language characteristics when building the SLN on the scientific papers for recommending research collaboration by constructing three types of semantic nodes. Compared with other summarization approaches on different granularity, the proposed approach achieves a higher recall score.
2. An SLN-based model for recommending research collaboration is proposed by constructing different types of semantic nodes and semantic links extracted from

scientific publications. Experiments on three data sets of scientific publications show that the model achieves good performance in predicting future collaborators. Research also unveil that different semantic links play different roles in representing texts.

3. A method for managing software engineering processes based on multi-dimensional Resource Spaces are proposed to manage software processes. Software engineering processes are mapped into multiple dimensions for supporting analysis, development and maintenance of software system. It can be used to uniformly classify and manage software methods and models through multiple dimensions so that software systems can be developed with appropriate methods. Interface for visualizing Resource Space Model was developed to support the method.

1.5 Motivation for this study

The Resource Space Model is to discourse the concern of how to competently build an applicable classification space for the resources of an application domain to enable users to easily operate the space to accomplish the contents of a number of resources and to normalize the space. Previous resource management approaches only concern the efficiency of managing resources [135], in this research we explore a complex space involved in many different resources with indeterminate behaviours. In this research, we will present background research and a review of the literature on RSM, managing and classifying the resources (taking software engineering models as experimental data), described the semantic link and discovering dimensions from texts (taking scientific papers as experimental data).

The motivation of this research consists of the following four aspects:

- Verify the roles of Semantic Link Network and the Resource Space Model in effectively managing various types of resources. Although the two models have solid theory basis, it is necessary to verify and complete them in various applications.
- Demonstrate the advantages of the models so as to provide application experiences.
- Explore the rules of the Resource Space Model that enhance the knowledge of representations and understanding through applications i.e. software engineering process and text summarization.
- Generalize a methodology for modelling complex reality and managing various resources. The methodology can help transform tradition methods in real applications.

1.6 Organisation of this thesis

The thesis consists of seven chapters. The first and seven chapter is written with the introduction and the conclusion correspondingly. Chapter two contains the review of the existing literature and the other four chapters include the comprehensive description, experiments and evaluation of the thesis innovative contributions, and present the research work which has been done further down this study. The particular structure of the thesis is described and presented in Figure 1.3.

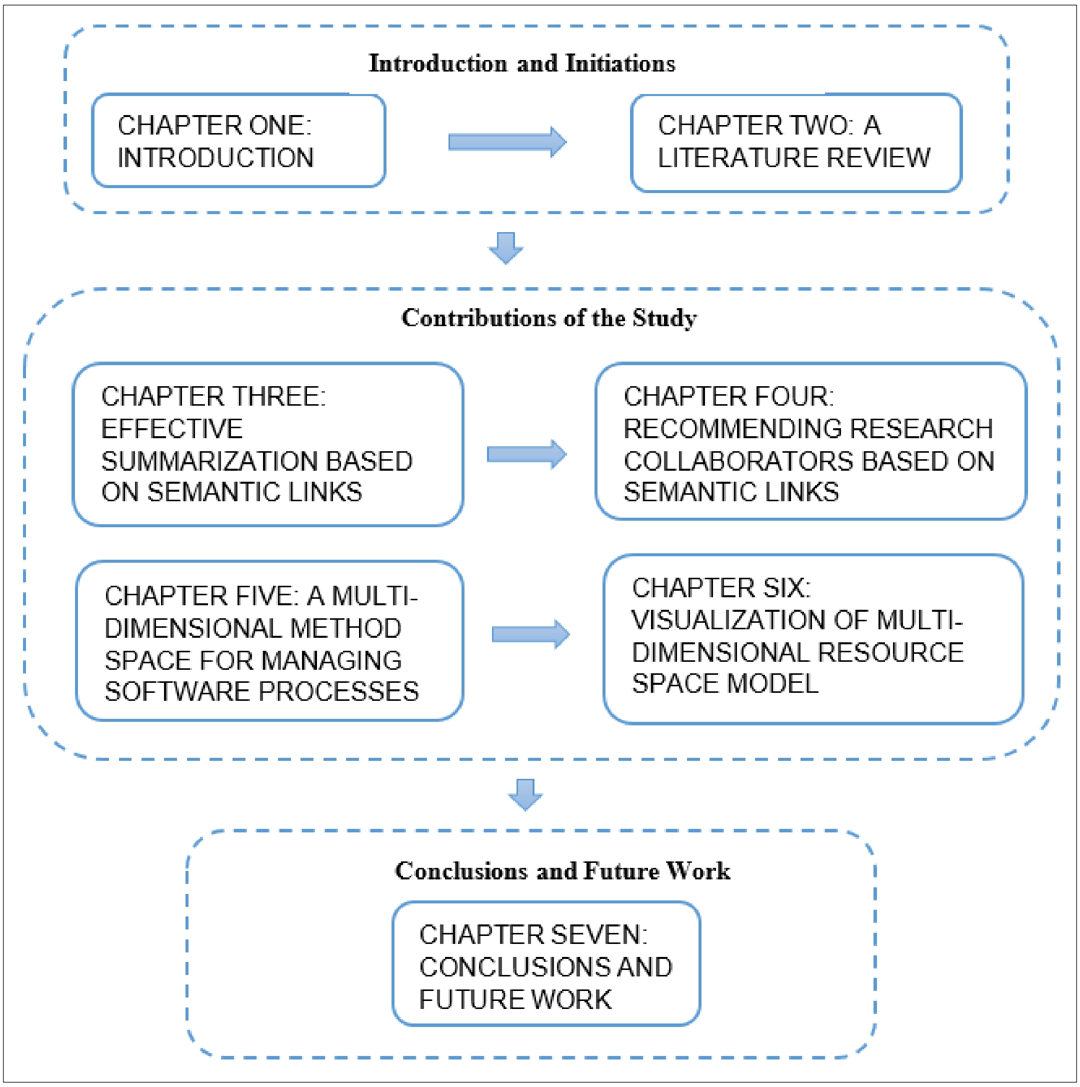

Introduction and Initiations
CHAPTER ONE: INTRODUCTION
CHAPTER TWO: A LITERATURE REVIEW
Contributions of the Study
CHAPTER THREE: EFFECTIVE SUMMARIZATION BASED ON SEMANTIC LINKS
CHAPTER FOUR: RECOMMENDING RESEARCH COLLABORATORS BASED ON SEMANTIC LINKS
CHAPTER FIVE: A MULTI-DIMENSIONAL METHOD SPACE FOR MANAGING SOFTWARE PROCESSES
CHAPTER SIX: VISUALIZATION OF MULTI-DIMENSIONAL RESOURCE SPACE MODEL
Conclusions and Future Work
CHAPTER SEVEN: CONCLUSIONS AND FUTURE WORK

Figure 1-3 Thesis modules and research workflow.

Chapter 01 introduces the Resource Space Model (RSM) and the Semantic Link Network (SLN), and defines the objectives of the research, motivation and structure of the thesis.

Chapter 02 presents the elementary concepts of the text summarization, and the graphical, uncertain and semantic approaches to building models. The combination of different approaches is a way to develop a stronger modelling method. We present background research and a review of existing literature for managing and classifying the resources (taking software engineering models as experimental data), followed by the instruction and background description of the existing software engineering techniques. That background knowledge plays a very important role to understand the research (Chapter 05) and purposes of a novel approach for developing the software engineering projects while implementing the Resource Space Model RSM.

Chapter 03 presents a clause-based extractive summarization algorithm by ranking and extracting semantic clauses from the original document. We segment the document into clauses and evaluate the importance of clauses based on semantic relations, and then, rank and extract them coarsely, and utilize graph rank to refine the extracted clauses. Based on this research we sum up the following two conclusions. Firstly, compared with the other summarization algorithms on different granularity, the clause-based summarization achieves higher recall score; and, secondly different discourse relations have different importance.

Chapter 04 proposes to build the SLN on scientific papers for recommending research collaboration. In this research, we propose a scientific recommendation approach to utilize semantic link networks to gain the required results. The SLN is constructed by three types of semantic nodes including author, paper and keyword and three types of semantic links including *write* link, *cite* link and *contain* link.

Chapter 05 introduces the way to manage the existing software methods through multiple dimensions so that software systems can be developed with appropriate methods. The main aim of this chapter is to apply the RSM to classify existing software engineering models and features from multiple dimensions so that software methods can be utilized in the most effective way to build real systems. Two case studies of applying the Resource Space Model to manage software engineering projects are studied, which show that RSM can map the software engineering activities into one space for management effectively. It offers a new approach to managing the whole lifecycle of a project to help software engineers to develop and design the solutions according to customers' requirements.

Chapter 06 presents a visualization of the multi-dimensional interface by adopting the Resource Space Model and demonstrates its advantages while purposing a user-friendly interface. The purpose visualization of multi-dimensional resource space is based on the theory of the Resource Space Model.

Chapter 07 is the summary of the thesis, including the main contribution of this research work and future research work.

2 CHAPTER TWO: A LITERATURE REVIEW

In this chapter, we present background research and a review of existing literature on the software engineering processes, text summarisation, semantic network models and a review of existing literature for managing and classifying the resources. This includes a definition of text summarization, elementary concepts of text summarization, and the graphical, and semantic approaches to building the models. The combination of different approaches and models is a way to develop a stronger modelling method. That background knowledge plays a very important role to understand the research that purposes a new approach for developing and building the application.

2.1 An Overview of the Text Summarisation

This section describes a review of existing literature and background research on text summarisation. It covers a definition of automatic text summarisation, categorisation of machine generated summaries based on context and language factors, approaches used to summarize text documents.

Text Summarization is the reduction of source text to produce a summary of a text by selecting and simplifying the most important sentences of the text. It presents a concise summary description of the text while maintaining the original concepts. Humans are the best summarisers for they possess the knowledge to understand and interpret the meaning of text documents. Automatic Text Summarisation (ATS) is the automation of this process by equipping computers with the knowledge required to carry out the summarisation.

Research on text summarisation started nearly 6 decades ago when Luhn [138] investigated the summarisation of scientific documents using statistical features such as the frequency of words. He used this frequency information to identify the salient sentences through the importance of their constituent words. Luhn's work has been extended by other researchers who used alternative shallow features such as the position of a sentence within a document, pragmatic words (e.g., significant, impossible and hardly), and heading/title words. These earlier pioneering works showed that summarising texts using machines is feasible. Since then, the field has been continuous evolved from simple statistical approaches to the application of robust NLP and artificial intelligence (AI) methods including machine learning, graph representation linguistic knowledge-based approaches and heuristic methods.

Human summarization is about knowledge, understanding and language use while automatic summarization concerns the modelling of text. There are two models of streams on texts: one stream (including the vector space model and topic model) assumes that words are independent of each other, and the other stream (including semantic link network) assumes that words are inter-related to render themes (this work distinguishes theme from topic according to the semantic link point of view). Human reading involves different strategies in different cases. Integrating the two streams of models is a way to establish a powerful model for summarization.

The Basic architecture of the ATS is shown in Figure 2.1. The input layer contains two types of documents which are single documents and multi-documents. Single-document summarization (SDS) converts a source text into a condensed, shorter text in which the relevant information is presented in a concise way. Multi-document summarization (MDS) where multiple documents are processed as input to generate the summary.

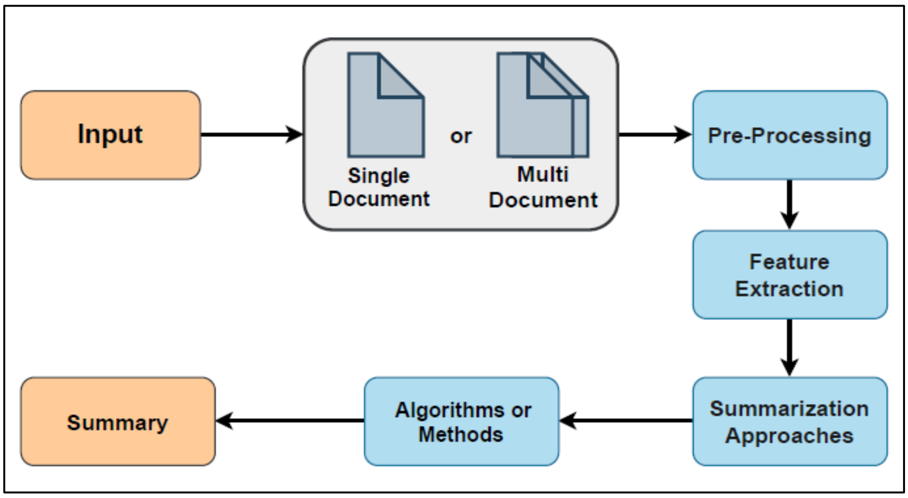


Figure 2-1 A basic structure of an ATS. [143]

A text summarization system usually consists of the following components:

A. PRE-PROCESSING in the pre-processing phase, the linguistic techniques are used to pre-process input texts using crucial techniques such as sentence segmentation, punctuation marks removal, filtering stop-words, stemming (reducing common root words), etc.
B. FEATURE EXTRACTION is vital for the entire summarization process by selecting different features within the source text. Selected features are applied to each sentence, and the highly scored sentences are chosen for the summary at the feature extraction phase.
C. SUMMARIZATION APPROACH is to determine an efficient methodologies. Some methods involve selecting the essential words and lines from the texts, while others involve paraphrasing one sentence and condensing the original content.

D. ALGORITHM or method is a more definite way of defining text summarization. Different algorithms and methods under various approaches are applied to obtain a better version of the summarized text.

2.2 EFFORTS TO INCORPORATE SEMANTICS INTO PROBABILISTIC GRAPHICAL MODEL

A. Probabilistic Graphical Model

A probabilistic graphical model (PGM, sometimes also called a graphical model or structured probabilistic model) is a probabilistic model that encodes probability distributions in a diagrammatic representation, where each node represents a random variable (or a group of random variables), and edges signify direct probabilistic interactions between these variables [54].

PGM provides a graph-based framework for manipulating large probability distributions efficiently. The role of the graph structure of PGM can be interpreted from two perspectives [54]: One perspective is that the graph is a compact representation of a set of conditional independences that hold on the distribution. Conditional independence describes which observations are irrelevant or redundant when evaluating the certainty of a hypothesis. It is usually formulated in terms of conditional probability: If A is a hypothesis, and B and C are observations, conditional independence can be stated as $P(A \mid B, C) = P(A \mid C)$, where $P(A \mid B, C)$ is the probability of A given both B and C, and $P(A \mid C)$ is the probability of A given only C. This equation expresses that B contributes nothing to the certainty of A given C, because the probability of A given C is the same as the probability of A given both B and C. In this case, A and B are said to be conditionally independent given C, denoted by $(A \perp\!\!\!\perp B \mid C)$, or sometimes $(A \perp B \mid C)$. Since conditional independence indicates which random variables are uninformative, it is useful for reducing redundant model parameters and unnecessary calculations. Conditional independence in PGM can be induced by simple graphical manipulations on its graph.

The other perspective is that the graph defines a skeleton for compactly representing a high-dimensional distribution, Rather than encoding the probability of every possible assignment to all variables in the domain, this “breaks up” the distribution into smaller factors, each over a much smaller space of possibilities, and then it defines the overall joint distribution as a product of these factors. This factorization lets us only process the factors that are much smaller than the overall joint distribution, so the number of parameters of the model is reduced, as well as the computational complexity. Such factorization in PGM can be achieved by inspection of its graph.

It has been turned out that these two perspectives are equivalent in a deep sense. The independence properties of the distribution are precisely what allow it to be represented compactly in a factorized form; a particular factorization of the distribution guarantees that certain independences hold [54].

Compared to the pure algebraic manipulation in traditional probability theory, PGMs offer several useful properties [55]: (1) They provide a simple way to visualize the structure of a probabilistic model and can be used to design and motivate new models; (2) Insights into the properties of the model, including conditional independence properties, can be obtained by inspection of the graph; (3) Complex computations, required to perform inference and learning in sophisticated models, can be expressed in terms of graphical manipulations, in which underlying mathematical expressions are carried along implicitly.

PGMs have two major families: Bayesian networks and Markov random fields. We introduce these two families by showing the graph representation, factorization, and conditional independence of each family. Then, we compare these two families. Finally, we introduce some extensions of them.

B. Bayesian Network

A Bayesian network (BN, also known as Bayes network, belief network, decision network, Bayesian model, and probabilistic directed acyclic graphical model) is a PGM whose network structure is a directed acyclic graph (DAG) [56]. The edge of BN describes a direct causal influence (or cause-effect relationship) between two random variables, pointing from the "cause" side to the "effect" side. The strength of this influence is quantified by conditional probability.

For random variables $x_1, \ldots, x_n$, BN encodes a joint distribution as a product of local conditional distributions, denoted by $P(x_1, \ldots, x_n) = \prod_i P(x_i | S_i)$, where S_i is the set of parents for variable x_i (i.e., the variables that have direct causal influences on x_i).

The conditional independence properties that hold in a BN can be induced according to a criterion called d-separation (where "d" stands for "directed") [57]. It states as follows: Consider a DAG in which A, B, and C are arbitrary nonintersecting sets of nodes (whose union may be smaller than the complete set of nodes in the graph). A path between any node in A and any node in B is said to be blocked by C if it includes a node such that either (1) the arrows on the path meet either head-to-tail (in the form of $a \rightarrow c \rightarrow b$) or tail-to-tail ($a \leftarrow c \rightarrow b$) at the node, and the node is in the set C, or (2) the arrows meet head-to-head ($a \rightarrow c \leftarrow b$) at the node, and neither the node, nor any of its descendants (a node x is said to be a descendant of a node y if there is a directed path from y to x, i.e., $y \to \cdots \to x$), is in the set C. If all paths between A and B are blocked by C, then A is said to be d-separated from B by C, and the joint distribution over all of the variables in the graph will satisfy that A and B are conditionally independent given C.

Here is an example of BN to show its graph representation, factorization, and conditional independence specifically. This example is about the medical diagnosis, which is a typical application of BN.

Example 1. Consider a simple medical diagnosis setting. There are two diseases: bronchitis and lung cancer. They are not mutually exclusive, i.e., a patient can have both, either, or none. They respectively correspond to two 2-valued random variables, Br and Ca (possible values: present, absent). There is a symptom and a medical test result: dyspnea and chest X-ray result, respectively corresponding to two 2-valued random variables, Dy (possible values: present, absent) and Xr (possible values: normal, abnormal). Besides, there is a 2-valued random variable Sm standing for smoking (possible values: smoker, nonsmoker) and a 4-valued random variable Se standing for the season (possible values: spring, summer, autumn, winter). Our task is to construct a joint distribution over this probability space to support principled probabilistic reasoning. Overall, corresponding to the possible assignments to these six variables, the probability space has $2 \times 2 \times 2 \times 2 \times 2 \times 4 = 128$ values. Because the sum of the probabilities of all these values must be 1, the probability of the last value is fully determined by the others. So, there are 127 non-redundant parameters to model this distribution.

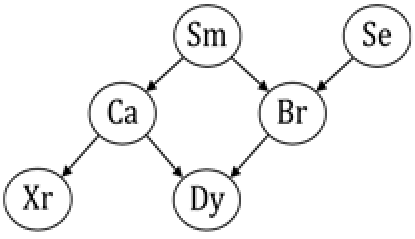
Specifying such a joint distribution with 127 parameters seems fairly daunting. Fortunately, we can use BN to encode this distribution more simply. With domain knowledge, we know that both bronchitis and lung cancer can cause dyspnea, but only lung cancer can cause abnormal chest X-ray results. Smoking can cause both bronchitis and lung cancer, but the season is only correlated with bronchitis. With these causal relationships, we can construct a BN as shown in Figure 2.2.

The BN enables us to model the joint distribution with much fewer parameters, which is helpful to represent the distribution compactly and manipulate the distribution (such as querying and inference) efficiently. As shown in Figure 1(b), the BN decomposes the overall joint distribution into six small factors ($P(Sm)$, $P(Se)$, $P(Ca \mid Sm)$, $P(Br \mid Sm, Se)$, $P(Xr \mid Ca)$, and $P(Dy \mid Ca, Br)$). Corresponding to the possible assignments to the variables in these factors, there are respectively 2 (i.e., $(Sm = smoker)$ and $(Sm = nonsmoker)$), 4 (i.e., $(Se = spring)$, $(Se = summer)$, $(Se = autumn)$, and $(Se = winter)$), $2 \times 2 = 4$ (i.e., $(Ca = present \mid Sm =$

$smoker)$, $(Ca = absent \mid Sm = smoker)$, $(Ca = present \mid Sm = nonsmoker)$, and $(Ca = absent \mid Sm = nonsmoker))$, $2 \times 2 \times 4 = 16$, $2 \times 2 = 4$, and $2 \times 2 \times 2 = 8$ possible values for each factor. Because the sum of the probabilities of all outcomes must equal 1, for each factor and each given condition, the probability of the last possible value is fully determined by the others. For example, for the factor $P(Ca \mid Sm)$, the probability $P(Ca = absent \mid Sm = smoker) = 1 - P(Ca = present \mid Sm = smoker)$, and $P(Ca = absent \mid Sm = nonsmoker) = 1 - P(Ca = present \mid Sm = nonsmoker)$. So, to model this factor, there are 2 nonredundant parameters. Therefore, to model these six factors, there are respectively 1, 3, $1 \times 2 = 2$, $1 \times 2 \times 4 = 8$, $1 \times 2 = 2$, and $1 \times 2 \times 2 = 4$ nonredundant parameters.

As the overall joint distribution is simply a product of these six factors, no more parameter is needed and the total number of parameters to model this overall joint distribution is the sum of them, i.e., $1 + 3 + 2 + 8 + 2 + 4 = 20$. This parameterization is significantly more compact, as opposed to 127 nonredundant parameters for the original joint distribution.

(a) Graph Representation



(b) Factorization

$$P(Sm, Se, Ca, Br, Xr, Dy) \\ = P(Sm)P(Se)P(Ca|Sm)P(Br|Sm, Se)P(Xr|Ca)P(Dy|Ca, Br)$$

(c) Independence

$$(Se \perp\!\!\!\perp Sm, Ca, Xr) \\ (Ca \perp\!\!\!\perp Br \mid Sm) \\ (Xr \perp\!\!\!\perp Sm, Se, Br, Dy \mid Ca) \\ (Dy \perp\!\!\!\perp Sm, Se, Xr \mid Ca, Br)$$

Figure 2-2 A Bayesian network for Example 1.

The BN also enables us to obtain conditional independences that hold in the distribution by inspection of its graph. The conditional independence properties in Example 1 are shown in Figure 2.2 (c). It is induced from the graph according to the d-separation. For example, $(Ca \perp\!\!\!\perp Br \mid Sm)$ holds because all the paths between Ca and Br are blocked by Sm (the path $Ca \leftarrow Sm \rightarrow Br$ is blocked according to the condition (1), and the path $Ca \rightarrow Dy \leftarrow Br$ is blocked according to the condition (2)), whereas $(Ca \perp\!\!\!\perp Br \mid Sm, Dy)$ does not hold because the path $Ca \rightarrow Dy \leftarrow Br$ is not blocked by $\{Sm, Dy\}$ according to the condition (2). These conditional independence properties indicate which variables are irrelevant, so that we can simplify our manipulations on the distribution. For example, as the conditional independence

$(Ca \perp\!\!\!\perp Br \mid Sm)$ holds, we have the equation $P(Ca \mid Br, Sm) = P(Ca \mid Sm)$. So, when we calculate the probability of a patient having lung cancer given that the patient smokes, we can ignore whether the patient has bronchitis, because it is irrelevant to the lung cancer in this case. Besides, considering the joint distribution of Ca and Br conditioned on Sm, we have $P(Ca, Br \mid Sm) = P(Ca \mid Br, Sm)P(Br \mid Sm) = P(Ca \mid Sm)P(Br \mid Sm)$.

This equation demonstrates that, conditioned on Sm, the joint distribution of Ca and Br can be factorized into the product of the marginal distribution of Ca and the marginal distribution of Br. So, when we calculate the probability of a patient having both lung cancer and bronchitis given that the patient smokes, we can separately calculate the probability of a patient having lung cancer given that the patient smokes and the probability of the patient having bronchitis given that the patient smokes, and finally simply multiply these two probabilities to get the answer. It allows us to avoid the complexity caused by the join of probabilities.

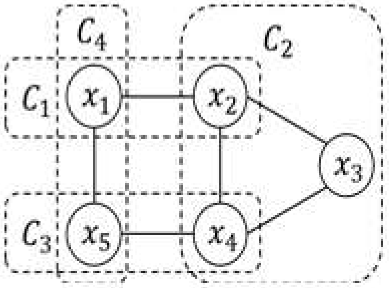
C. Markov Random Field

A Markov random field (MRF, also known as Markov network and undirected graphical model) is a PGM, where random variables have a Markov property described by an undirected graph. The edge of MRF describes a symmetrical relationship between two random variables. The prototype of MRF is the Ising model (a mathematical model of ferromagnetism in statistical mechanics) [58]. The theory of the MRF may trace to [59][60]. We introduce the conditional independence in MRF first, and then the factorization. To illustrate them intuitively, we give a simple example of MRF as shown in Figure 2.3.

Conditional independence in MRF can be directly induced from the graph simply according to the graph separation. In detail, for three nonintersecting node sets A, B, and C, the conditional independence property $(A \perp\!\!\!\perp B \mid C)$ is satisfied by a probability distribution defined by an MRF if all paths connecting nodes in set A and nodes in set B pass through at least one node in set C. An alternative view of this conditional independence test is to imagine removing all nodes in set C from the graph together with all edges connecting to these nodes. Then the conditional independence property $(A \perp\!\!\!\perp B \mid C)$ must hold if there is no path connecting any node in set A and any node in set B. In our MRF example, the conditional independence properties obtained from the graph are shown in Figure 2.3 (c). For example, $(x_1 \perp\!\!\!\perp x_3, x_4 \mid x_2, x_5)$ holds because all paths connecting x_1 and the nodes in $\{x_3, x_4\}$ pass through $\{x_2, x_5\}$, whereas $(x_1 \perp\!\!\!\perp x_3, x_4 \mid x_2)$ does not hold because the path (x_1, x_5, x_4) connects x_1 and x_4 but does not pass through x_2.

According to the above conditional independence test, we know that two nonadjacent nodes must be conditionally independent given all other nodes in an MRF. Therefore, the factorization of the joint distribution must be such that nonadjacent nodes do not appear in the same factor to guarantee that the conditional independence property can hold for all possible distributions belonging to the graph. This leads us to consider decomposing the joint distribution according to the cliques (a clique in a graph is a subset of the nodes such that every two distinct nodes are adjacent) in an MRF. Furthermore, without loss of generality, the joint distribution of an MRF can be defined as a product over functions of the variables in the maximal cliques (a maximal clique is a clique that cannot be extended by including one more adjacent node), because other cliques must be subsets of maximal cliques and including another factor defined over them would be redundant. (Note that some authors define cliques in a way that requires them to be maximal and use other terminology for complete subgraphs that are not maximal.) The functions over the cliques are usually called potential functions, denoted by $\phi(\dots)$ or $\psi(\dots)$. In our MRF example, there are 4 maximal cliques, as shown in Figure 2.3(a), out by dotted lines. The factorization is shown in Figure 2.3(b). In general, for discrete random variables $x_1, \dots, x_n$, with cliques $C_1, \dots, C_m$ over them, MRF encodes a joint distribution as a product of potential functions over these cliques and normalizing, denoted by $P(x_1, \dots, x_n) = \frac{1}{Z}\prod_{j=1}^{m} \phi_j(X_j)$, where $X_j \subseteq \{x_1, \dots, x_n\}$ is the set of random variables in clique C_j, $\phi_j(X_j)$ is the potential function over clique C_j, $Z = \sum_{x_1,\dots,x_n} \prod_{j=1}^{m} \phi_j(X_j)$ is a normalization constant called partition function. By considering only potential functions that satisfy $\phi_j(X_j) \geq 0$, we ensure that $P(x_1, \dots, x_n) \geq 0$. Although this factorization is defined for discrete variables, the framework is equally applicable to continuous variables, or a combination of the two, in which the summation is replaced by the appropriate combination of summation and integration.

(a) Graph Representation



(b) Factorization

$$P(x_1, x_2, x_3, x_4, x_5) = \frac{1}{Z}\phi_1(x_1, x_2)\phi_2(x_2, x_3, x_4)\phi_3(x_4, x_5)\phi_4(x_5, x_1),$$
$$where\ Z = \sum_{x_1, x_2, x_3, x_4, x_5} \phi_1(x_1, x_2)\phi_2(x_2, x_3, x_4)\phi_3(x_4, x_5)\phi_4(x_5, x_1)$$

(c) Independence

$$(x_1 \perp\!\!\!\perp x_3, x_4 \mid x_2, x_5)$$
$$(x_3 \perp\!\!\!\perp x_1, x_5 \mid x_2, x_4)$$
$$(x_5 \perp\!\!\!\perp x_2, x_3 \mid x_1, x_4)$$

Figure 2-3 An example of Markov random field.

In principle, not all MRFs can be factorized according to the cliques. A simple example can be constructed on a cycle of 4 nodes with some infinite energies, i.e., configurations of zero probabilities [61]. An MRF can be factorized if at least one of the following two conditions is fulfilled: (1) the joint distribution (or density) is strictly positive (by the Hammersley-Clifford theorem); (2) the graph is chordal (a chordal graph is one in which all cycles of 4 or more nodes have a chord, which is an edge that is not part of the cycle but connects two nodes of the cycle) (by equivalence to a BN).

D. Comparison

The difference in the underlying graphs lets BN and MRF have their own strength. The DAG-based structure makes BN good at supporting inter-causal reasoning. It is used for a wide range of tasks related to causality, including prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction and decision making under uncertainty. In contrast, the undirected structure makes MRF good at capturing symmetrical relationships. It is often used in areas such as spatial statistics, statistical natural language processing, and communication networks, in which there is little causal structure to guide the construction of a directed graph.

From the perspective of factorization, BN and MRF decompose the joint distribution into different formalisms. The factor (or parameter) in BN is the conditional distribution of the corresponding variable, conditioned on the state of its parents. It has a specific probability interpretation and is easy to estimate. Sometimes it can be set manually, which makes it easy to build models not only from data but also from expert opinions. In MRF, the factor is the potential function. It may have no specific probabilistic interpretation as marginal or conditional distributions. It is trickier to estimate than in BN, and usually learned from samples. But in special cases where the undirected graph is constructed by starting with a directed graph, the potential functions may have a probabilistic interpretation.

From the perspective of conditional independence, BN and MRF provide different ways to describe the conditional independences that hold in the distribution. The DAG-based structure brings asymmetry between parent and child nodes to BN. It asks to consider not only the connectivity but also the directionality in the graph when testing conditional independence in BN. Fortunately, the entire set of conditional independence properties in a BN can be obtained in polynomial time by d-separation [57], which is still efficient. In MRF, the edges have no directionality so conditional independence can be determined simply by the graph separation, which is easier than in BN.

E. Some Extensions of Bayesian Network and Markov Random Field

Some efforts extended the typical BN and MRF to enhance their expressiveness and handle uncertainty with more information.

One important kind of these extensions is to enable PGM to deal with hierarchical information and structured domains. Hierarchical Bayesian Network (HBN) is an extension of BN, which uses knowledge about the structure of the data to introduce a bias that can contribute to improving inference and learning methods [62]. Unlike the typical BN that can only deal with propositional domains, HBN is able to deal with structured domains. It defines the types of data in a tree-like structure to reflect the hierarchical information, where the elementary types are called atomic types, corresponding to the external nodes of the tree, and the composite types (or called aggregative types) are defined as Cartesian products of atomic types, corresponding to the internal nodes of the tree. The graph of HBN represents not only the probabilistic dependences between the random variables but also the "part-of" relationships that describe the hierarchical structure of variables. The nodes in HBN may represent variables of atomic types (like nodes in typical BN) or that of (possibly nested) aggregations of atomic types (corresponding to a group of nodes in typical BN, but with more structural information). Every aggregate node is itself an HBN that models the relationships inside a subset of the whole world under consideration. The edges in HBN signify probabilistic dependences the same way as in typical BN, but these edges may lie at any level of nesting into the data structure. Like HBN extends typical BN to support structured domains, hierarchical hidden Markov model (HHMM) generalizes the standard hidden Markov model

(HMM, which can be viewed as a special case of BN) to support structured multi-level stochastic processes [63]. The main idea of HHMM is simple: It recursively defines each hidden state as an HHMM as well, so that through a recursive activation process, the states of an HHMM can emit sequences rather than a single symbol. This recursively defined structure brings many advantages to HHMM, such as better modelling the different stochastic levels and length scales, and better inferring correlated observations over long periods in the observation sequence via the higher levels of the hierarchy. Recently, to make PGM able to model hierarchical data and meanwhile scalable enough to be suitable for big data problems, an efficient and scalable probabilistic-based model for massive hierarchical data (PGMHD) was proposed [64]. The model structure of PGMHD is simply a multi-level directed graph, where nodes are partitioned into different levels, and each edge is restricted to only connect from one level to the next. Although the model can be seen as a special case of BN, the experiments show that PGMHD improves the scalability of typical BN. The leveled directed graph that explicitly defines one node per outcome of the random variables leads to an easily scalable implementation, improves the readability and expressiveness of the implemented network, and simplifies and facilitates the training of the model.

Another kind of the extensions is to consider the temporal information. Time Delayed Probabilistic Graphical Model (TD-PGM) is a model designed for detecting global behaviour anomalies in multiple disjoint cameras (e.g., detecting traffic accidents via several cameras monitoring different road junctions) [65]. Each node in TD-PGM represents activities in a semantically decomposed region from one of the camera views, and the directed links between the nodes encode the causal relationships between the activities. By associating each observation with a time index and using time delayed mutual information [66] analysis, temporal information is leveraged in this model. And with its novel learning method, TD-PGM finally models the causal relationships between the observations at different moments, which is called time-delayed dependences. Time-dynamic Markov random field (TD-MRF) is an extension of typical MRF, designed for uncertain reasoning in the detection and tracking of a hazardous plume (e.g., poisonous smoke from a chemical plant) with a sensor network [67]. Each node in TD-MRF represents a random variable. For each sensor location and each time step, there is a pair of random variables: One represents the true state (a Boolean value); the other represents the output of that sensor (a continuous value between 0 and 1). Edges in TD-MRF indicate dependences between the random variables. There are three kinds of dependences in TD-MRF: The first is the dependence between a true state and its corresponding sensor output. The second represents the spatial correlation, which connects the nodes for the true states that correspond to the sensors at different but neighbouring locations in the same time step. The last represents the temporal correlation, which connects the nodes for the true states that correspond to the sensors with time-delayed influence. Intuitively, when wind blows at a certain speed and toward some direction, it carries the plume from one location to another location after some time steps, so the sensors at these two locations may be related in a time-delayed manner. However, the temporal correlation depends on some additional conditions, such as the wind condition, and there may be no sensor in the network to detect these conditions. Therefore, unlike the typical MRF where the structure is known before inference, part of the structure of TD-MRF is unknown and needs to be determined dynamically during the inference. TD-MRF has a new inference algorithm to achieve this without the knowledge about the additional conditions.

2.3.1 Associating Probabilistic Graphical Model with Semantics

In practice, many applications require both representing domain knowledge and handling uncertainty, so there is a need to associate PGMs with semantics. Since around 2000, the interest in this problem has grown due to its relevance to statistical relational learning [68]. Nowadays, this issue becomes more and more general and important in many areas, including machine learning, natural language processing, computer vision, social networks, biological networks, and the Web.

To handle uncertainty in first-order knowledge bases, some work tries to combine PGM and first-order logic (FOL, which can compactly represent a wide variety of knowledge) into a new representation. One representative of these efforts is the Markov logic network (MLN) [68]. The basic idea in MLNs is to soften the constraints of the formulas in a first-order knowledge base. In detail, a first-order knowledge base can be seen as a set of hard constraints on the set of possible worlds: If a world violates even one formula, it has zero probability. MLNs soften these constraints: When a world violates one formula in the knowledge base it is less probable, but not impossible. The fewer formulas a world violates, the more probable it is. Each formula has an associated weight that reflects how strong a constraint it is: the higher the weight, the greater the difference in log probability between a world that satisfies the formula and one does not, other things being equal.

An MLN is a single representation combining MRFs and FOL without restrictions other than the finiteness of the domain. It is a first-order knowledge base with a weight attached to each formula (or clause). Together with a set of constants representing objects in the domain, it specifies a ground MRF containing one feature for each possible grounding of a first-order formula in the knowledge base, with the corresponding weight. (In MRFs that are represented as log-linear models, each clique potential is replaced by an exponentiated weighted sum of features of the state of the variables in that clique, leading to $P(X = x) = \frac{1}{Z} exp\left(\sum_j w_j f_j(x)\right)$, where $f_j(x)$ is the feature function for the clique indexed by j, and w_j is the weight associated with this feature; a feature may be any real-valued function of the state, but MLN only focuses on binary features, $f_j(x) \in \{0,1\}$) The ground MRF contains one binary node for each possible grounding of each predicate (or called ground atom) appearing in the first-order formulas. The value of the node is 1 if the ground atom is true, and 0 otherwise. The MRF contains an edge between two nodes if and only if the corresponding ground atoms appear together in at least one grounding of one of the formulas. Thus, the atoms in each ground formula form a (not necessarily maximal) clique, and each possible grounding of a formula can correspond to one feature of the MRF. The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is the weight associated with its corresponding formula. The probability distribution over possible world x specified by a ground MRF is given by $P(X = x) = \frac{1}{Z} exp\left(\sum_j w_j n_j(x)\right) = \frac{1}{Z}\prod_j \phi_j\left(x_{\{j\}}\right)^{n_j(x)}$, where w_j is the weight associated with the formula F_j, $n_j(x)$ is the number of true groundings of F_j in x, $x_{\{j\}}$ is the state (truth values) of the atoms appearing in F_j, and $\phi_j\left(x_{\{j\}}\right) = e^{w_j}$.

Here is an example to illustrate how MLN constructs an MRF for a simple first-order knowledge base. The propositions, first-order formulas, constants, and the ground MRF in this example are shown in Figure 2.4.

Example 2. Consider a simple knowledge base consisting of 3 propositions: (1) One who studies hard will get high scores. (2) Friends of one who studies hard will also study hard. (3) Friends of friends are friends. These propositions can be represented in FOL, as shown in Figure 2.4(b), where $Sh(x)$ stands for “x studies hard”, $Hs(x)$ stands for “x gets high score” and $Fr(x,y)$ stands for “x and y are friends”. They can convert to clausal form, as shown in Figure 2.4(c). Each formula has an associated weight. By applying the formulas to the constants Alice(A), Bob(B) and Charles(C), a ground MRF can be constructed as shown in Figure 2.4(e). In this MRF, there is a node for each ground atom (e.g., $Fr(A,B)$) and there is an edge between two nodes if and only if the corresponding ground atoms appear together in at least one grounding of one formula (e.g., there is an edge between $Sh(A)$ and $Hs(A)$ because these two atoms appear together in the grounding of formula as shown in Figure 2.4(c.1) by applying x to constant A). This MRF can be used to infer the probability such as that Alice and Bob are friends given their scores, and that Charles studies hard given his friendship with Alice and Bob, and whether Alice and Bob study hard, etc.

(a) Propositions

(a.1) One who studies hard will get high scores.
(a.2) Friends of one who studies hard will also study hard.
(a.3) Friends of friends are friends.

(b) FOL formulas

(b.1) $\forall x\, Sh(x) \Rightarrow Hs(x)$
(b.2) $\forall x \forall y\, Sh(x) \wedge Fr(x,y) \Rightarrow Sh(y)$
(b.3) $\forall x \forall y \forall z\, Fr(x,y) \wedge Fr(y,z) \Rightarrow Fr(x,z)$

(c) FOL formulas in clausal form with associated weights

(c.1) $\neg Sh(x) \vee Hs(x)$ (weight: 1.5)
(c.2) $\neg Sh(x) \vee \neg Fr(x,y) \vee Sh(y)$ (weight: 1.1)
(c.3) $\neg Fr(x,y) \vee \neg Fr(y,z) \vee Fr(x,z)$ (weight: 0.7)

(d) Constants

(d.1) Alice (A)
(d.2) Bob (B)
(d.3) Charles (C)

(e) Ground MRF

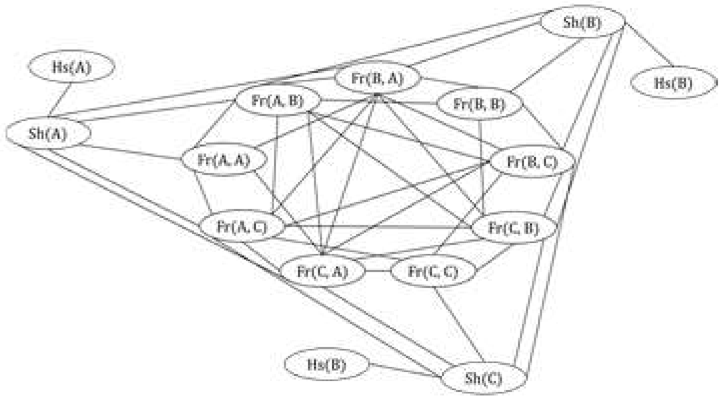


Figure 2-4 Propositions, first-order formulas, constants, and ground MRF in Example 2.

MLN represents abundant semantics by FOL and handles uncertainty by MRFs, thus setting up a common bridge between MRFs and FOL. It can soundly handle uncertainty, tolerate imperfect and even contradictory knowledge, and reduce brittleness. It can be regarded as a first-order knowledge base with a weight for each formula, or a template for constructing MRFs according to the first-order knowledge base. It has been used for a variety of statistical relational learning tasks at first and has wide usage in AI.

Some work combines MLN with other models or techniques to improve its expressiveness. For example, LPMLN is a combination of logic programs under the stable model semantics and MLNs in a single framework [69]. It enhances the ability of MLNs to represent defaults, causal relations, inductive definitions, and aggregates.

Another notable work is probabilistic soft logic (PSL) [70][71]. Like MLN, PSL relaxes the constraints of the formulas in the knowledge base. Furthermore, it relaxes the truth values of random variables. Its motivation is that many problems addressed by statistical relational learning, such as ontology alignment and collective classification, have a characteristic called imprecision, which can’t be reflected well in the general frameworks for probabilistic reasoning at that time, including MLNs and Bayesian logic programs. PSL softens the truth values of random variables to continuous values in the interval [0,1] via fuzzy logic. To relax the truth values of random variables in MRFs, a new kind of PGM based on Łukasiewicz logic, called hinge-loss Markov random field (HL-MRF), was proposed [70][72]. Then PSL was developed for making HL-MRFs easily applied to a broad range of structured machine learning problems by defining templates for potentials and constraints. Now PSL keeps being developed to satisfy the growing need for modelling rich, structured, and highly scalable data and is used in many problems in machine learning, including social networks, biological networks, the Web, natural language, computer vision, sensor networks, and so on [70].

Similar to that MLN can be regarded as a template for constructing MRFs that are defined over Boolean variables, PSL can be regarded as a templating language for HL-MRFs, which are defined over continuous variables. Compared to Boolean MRFs and MLNs, HL-MRFs and PSL scale much better while retaining the rich expressivity and accuracy of their discrete counterparts [70]. In addition, HL-MRFs and PSL can directly reason about continuous data.

MLN and PSL provide an efficient way to deal with uncertainty in first-order knowledge bases via PGMs. They associate PGMs with FOL, which enriches the semantics and enhances the expressiveness of the model. Compared to classical PGMs, MLN and PSL not only represent the structure of dependences via their MRFs but also represent the semantics of knowledge via their logic formulas. Compared to FOL, MLN and PSL enable us to treat issues via logic as well as statistics, which is helpful for inductive reasoning.

However, MLN and PSL only represent domain knowledge as logic formulas. Their graphs are just MRFs, where the complicated semantic information carried by the predicates of formulas is uniformly rendered as the nodes in MRFs. It is good for focusing on handling uncertainty. But on the other hand, it is inconvenient to distinguish different types of information shown on the nodes. And it is difficult to discover new relationships and represent further deep knowledge structures.

2.3.2 EFFORTS TO INCORPORATE UNCERTAINTY INTO GRAPH-BASED KNOWLEDGE REPRESENTATION

People have a long history of using graphs to describe knowledge. Nowadays, graph-based knowledge representation (GBKR) becomes an important field in artificial intelligence, computer science, and cognitive science. Generally, the nodes of a GBKR model can represent anything, including entities, objects, events, concepts, classes, and so on, and the edges signify various relationships between the nodes. Both nodes and edges can have rich semantics, which is typically indicated by the associated labels or tags. It can be regarded as a combination of graphics and semantics.

A. Semantic Network, Semantic Web, and Knowledge Graph

The semantic network (SN, also called a semantic net) is one of the influential GBKR models. An SN is a graphic notation for representing knowledge in patterns of interconnected nodes and edges. It consists of nodes representing concepts, and edges representing semantic relations between concepts, mapping or connecting semantic fields [73].

The earliest documented use of SNs is the Greek philosopher Porphyry’s commentary on Aristotle’s categories in the third century AD. In computer science, SN was first implemented by Richard Hook Richens as an “interlingua” for machine translation of natural languages in 1956 [74][75].

SNs were summarized as the following six types: definitional networks, assertional networks, implicational networks, executable networks, learning networks, and hybrid networks (Sowa, 2006). A brief description of these six kinds is shown in Table 2.1. The common point of all SNs is a declarative graphical representation that can be used to represent knowledge and support automated systems for knowledge reasoning.

	Brief description	*Main feature*
Definitional network	It emphasizes the subtype or “is-a” relation between a concept type and a newly defined subtype. The resulting network, also called a generalization or subsumption hierarchy, supports the rule of inheritance for copying properties defined for a supertype to all its subtypes. The information in these networks is often assumed to be necessarily true since definitions are true by definition.	Its purpose is to define concepts. It describes the intension and extension of a concept, the relationship between concepts, and the characteristics and attributes of an object.
Assertional network	It is designed to assert propositions. The information in an assertional network is assumed to be contingently true unless it is explicitly marked with a modal operator. Some assertional networks have been proposed as models of the conceptual structures underlying natural language semantics. Note that in computer science, another one of the influential GBKR models, the conceptual graph, first introduced by Sowa in 1976 [76], can be considered as a kind of assertional networks.	Its purpose is to assert propositions. It represents logical relationships and natural language semantics with graphs.
Implicational network	It uses implication as the primary relationship for connecting nodes. It may be used to represent patterns of beliefs, causality, or inferences. It can be regarded as a special case of propositional SNs (a kind of assertional networks) as well.	It draws the implicational structures but ignores other relationships or nests them inside the propositional nodes to highlight the relationships that are useful for inference and reasoning.

Executable network	It includes some mechanisms, such as marker passing or attached procedures, which can perform inferences, pass messages, or search for patterns and associations.	It essentially describes events or procedures and simulates their execution. It contains not only the semantic graphs but also the mechanism that cause some change to the network itself.
Learning network	It builds or extends its representation by acquiring knowledge from examples. The new knowledge may change the old network by adding or deleting nodes and edges or by modifying numerical values (called weights) associated with the nodes and edges.	It can respond to new information by modifying its internal representations. It enables the system to respond more effectively to its environment.
Hybrid network	It combines two or more previous techniques, either in a single network or in a set of separate but closely interacting networks.	Its features depend on the combined technologies.

Table 2.1 Six common kinds of semantic networks

The Semantic Web proposed in 2001 can be regarded as an application of SN on the Web [77]. The ultimate goal of Semantic Web is to make computers understand Internet data. To encode the rich semantics with data, some well-known technologies were proposed, including Resource Description Framework (RDF) and Web Ontology Language (OWL). These technologies formally represent the semantics of information. In 2006, Linked Data was proposed to enable computers to share and explore the information on the Web, read information automatically, and deal with semantic queries.

SNs provide a graph-based formalism for describing information with plentiful semantics. They represent the structures of knowledge visibly and describe various semantics logically, which is useful to explore the information automatically and achieve reasoning. However, most research on SNs only focuses on how to represent semantics and build links between data. It lacks a systematic theory to manage diverse semantics.

Uncertainty is less considered in the study of SN. However, it is important to deal with uncertainty when applying SNs to knowledge bases in practice. Many common issues such as relationship completion (completing missing relationships according to observed facts) and entity disambiguation (assigning a unique identity in the knowledge base to the mention of an ambiguous entity in text) may need to cope with uncertain information. Such issues have been studied under the new cap of Knowledge Graph (KG).

The term Knowledge Graph was introduced as a restricted SN in the 1980s (Van de Riet & Meersman, 1992). It was reintroduced by Google as an enhancement of Google Search engine with semantics in 2012. The existing definitions of KG are vague and even contradictory. KG seems to describe semantics in a graph-based formalism, which inherits SN in nature. It is often used to store interlinked descriptions of entities, including real-world objects and events and abstract concepts, where the descriptions have formal semantics that allow both people and computers to process them in an efficient and unambiguous manner. Moreover, most KGs are assigned some functions of traditional knowledge bases such as knowledge acquisition, integration, and reasoning. Some examples that are generally considered KG are Google Knowledge Graph, DBPedia, Geonames, WordNet, and FactForge.

To deal with uncertainty in tasks such as completion, disambiguation, and inference, there are two main categories of methods. One is representing information stored in KG by FOL, and then using MLN, PSL, or other probabilistic logic programming languages to handle the uncertainty. These approaches are able to leverage domain knowledge with FOL and meanwhile handle the uncertainty with PGM, but they are usually difficult to inference because of the complicated and large-scale graph structures. The other is applying machine learning methods to handle the uncertainty by using KG embedding methods such as TransE [78] and ComplEx [79] to convert the KG to the distributed representation so that it can be used as the input of machine learning methods, or using the graph neural network [80] to directly process the KG in machine learning models (e.g., [80][82]. These approaches can carry out reasoning more efficiently than PGM, but they are hard to utilize domain knowledge because logic rules cannot be applied to these approaches, and they may ask to retrain the machine learning models to leverage additional domain knowledge.

Although many research works have been done under the cap of KG, the model has no significant difference from the SN and Semantic Web in nature. Moreover, they lack in-depth understanding of knowledge. More exploration on modelling knowledge was introduced under the framework of the Knowledge Grid [3].

B. Semantic Link Network

Research on SLN can trace to the definition of inheritance rules for flexible model retrieval [42] and the development of Active e-Document Framework (ADF) [139]. The systematic theory and model were created for the first time in 2004 [13]. Since then, research has developed towards a self-organized social semantic networking method [140][36].

The self-organized SLN is integrated with a multi-dimensional concept space to form a semantic space to support advanced applications with multi-dimensional abstractions and self-organized semantic links [2][5]. SLN has been successfully used in many applications, e.g., a general summarization method [4]. It has been verified that SLN plays an important role in understanding and representation through text summarization applications [3][53].

Compared to the SN, SLN focuses more on diverse semantic links (including 12 types of general semantic links) and opens to incorporate more semantic links and linking rules, especially the social linking rules, reflecting uncertain and inconsistent characteristics of social intelligence [141]. It has been verified that these semantic links play an important role in applications, e.g., the cause-effect link is important for understanding text and summarization on scientific documents [3]. Interaction, reasoning, community, and automatic discovery of implicit links play an important role in SLN [14][34].

To reflect and handle uncertainty, a Probabilistic Semantic Link Network (P-SLN) was proposed [3]. It associates each semantic link with a lower probability bound and an upper probability bound and associates each rule with a certainty degree to introduce uncertainty into SLN. In addition to the typical relational reasoning rules, P-SLN adds some new kinds of reasoning rules, including statistical inference rules, assertion rules, classification rules, and attribute-to-resource rules, to deal with more kinds of reasoning caused by uncertainty.

SLN represents a kind of emerging semantics with the evolution of semantic link network with operations on nodes and links as well as reasoning on the network through its semantic space. P-SLN makes it possible to reflect and deal with uncertain semantic links. SLN emphasizes on social semantics and interactive semantics [34]. Previous models such as SN and PGM can be regarded as a special case of SLN.

The notion of Cyber-Physical-Social Intelligence is the natural extension of the original insight of the Future Interconnection Environment [37]. Cyber-Physical-Social Semantic Link Network (CPSoSLN) reflects the basic structure in cyber-physical-social space, where nodes can be objects in different spaces [141][142].

SLN maintains a semantic space to systematically manage the complex and diverse semantics of its nodes and links. It enables SLN to work across different spaces and different applications and provides an infrastructure for studying the role of different semantics. The semantic space also holds several kinds of reasoning rules, which supports reasoning on SLN to complete missing links and discover new links. The applications of SLN in text summarization show that it can help find the core representation of text through transforming texts into SLN [3][53]. P-SLN has the ability to describe unknown, imperfect, fake, and contradictory information through introducing uncertainty into SLN.

A scientific paper can be represented as a Semantic Link Network of semantic nodes (e.g., sections, sub-sections, paragraphs, sentences and words) and semantic links (e.g., is-part-of, similar-to and co-occurrence) between nodes [53]. An example is shown in Figure 2.5. All those components can be easily parsed out from the paper in HTML.

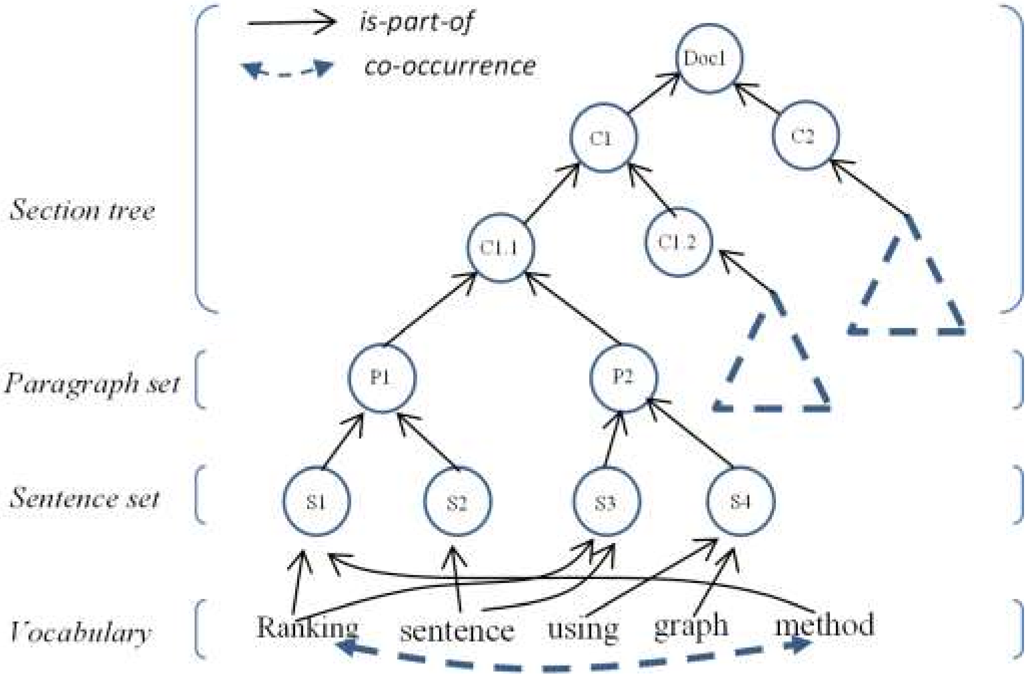


Figure 2-5 A Semantic Link Network of a paper [53].

A semantic node can be anything including a concept, a set, and a physical object. Various semantic links, category hierarchies, and the reasoning rules on semantic links constitute the basic model of the Semantic Link Network, which is more capable of modelling semantics than traditional graph structure [53].

2.3.3 Discussion

A timeline of the development of the techniques is shown in Figure 2.6. A brief comparison of the models is shown in Table 2.2.

In the last century, some tasks that involve probability, such as inference, required efficient analysis of the relationship between random variables. PGMs introduced graphics into probability theory to clearly describe the relationships through the graph-based structure. For example, BN uses DAGs to represent the causal relationship and MRF uses undirected graphs to show symmetrical relationships such as the co-occurrence relationship and the spatial adjacent relationship. These PGMs provide a way to decompose large probability distributions into small factors and reveal the independences between random variables, which is helpful to reduce redundant parameters and unnecessary calculations. Their graph-based structure enables us to replace complex algebraic operations with simple graphical operations. Also in the last century, SN introduced graphics into natural language processing to describe the information in propositions and highlight the relationships between concepts. After that, the graph-based formalism was generalized to express various knowledge structures. For example, the Semantic Web uses graphs to represent the relationships on Internet data. SLN models the social relationships with semantic links and reasoning on semantic links. The recent SLN model includes richer semantics and social space [141]. The graph-based representation is intuitive and is beneficial to discover new relationships, support reasoning, and motivate new models.

Since the 1990s, the rapid development of the World Wide Web has led the era of big data and brought vast requirements for efficient information analysis and data mining. The Web makes it easier for researchers to communicate and exchange ideas as well. Therefore, as shown in Figure 2.6, since the ubiquity of the World Wide Web, the evolution of the models has accelerated significantly.

The fast-growing data brought new research issues and new requirements. For example, statistical relational learning requires both representing domain knowledge and handling uncertainty, and some issues of constructing GBKR models in practice, such as relationship completion and entity disambiguation, ask to deal with uncertainty in graph-based knowledge bases. These issues lead us to develop previous techniques and combine their advantages, i.e., to develop a combination of graphics, uncertainty, and semantics. MLN combines PGM and FOL to efficiently handle uncertainty in first-order knowledge bases that are defined over Boolean variables. Similarly, PSL combines PGM and Łukasiewicz logic to handle uncertainty in knowledge bases that are defined over continuous variables. To deal with uncertainty in GBKR models, KG uses MLN or machine learning methods, and SLN is extended to P-SLN.

The models based on PGM provide an efficient way to handle uncertainty but lack rich semantics to describe domain knowledge. The models based on GBKR provide an excellent way to represent various knowledge and highlight the relationships but lack a systematic way to handle uncertainty. A possible approach is to combine the strengths of these two kinds of models, more precisely, a simple approach uses GBKR models with probability to model abundant knowledge and constructing PGMs on the GBKR models to compute the probability. In this way, the GBKR part uses graphics with plentiful semantics to represent domain knowledge and describe relationships in one formalism; the PGM part uses graphics to represent dependences between undetermined information and handle uncertainty efficiently. But some work is still needed to integrate GBKR models and PGMs systematically. Another approach is evolving a systematic method to handle uncertainty in GBKR models to develop a model that can use rich-semantic graphics to describe the world and can handle both determined and uncertain information. P-SLN is a promising idea, which can develop with incorporating the techniques for the automatic discovery of links and rules.

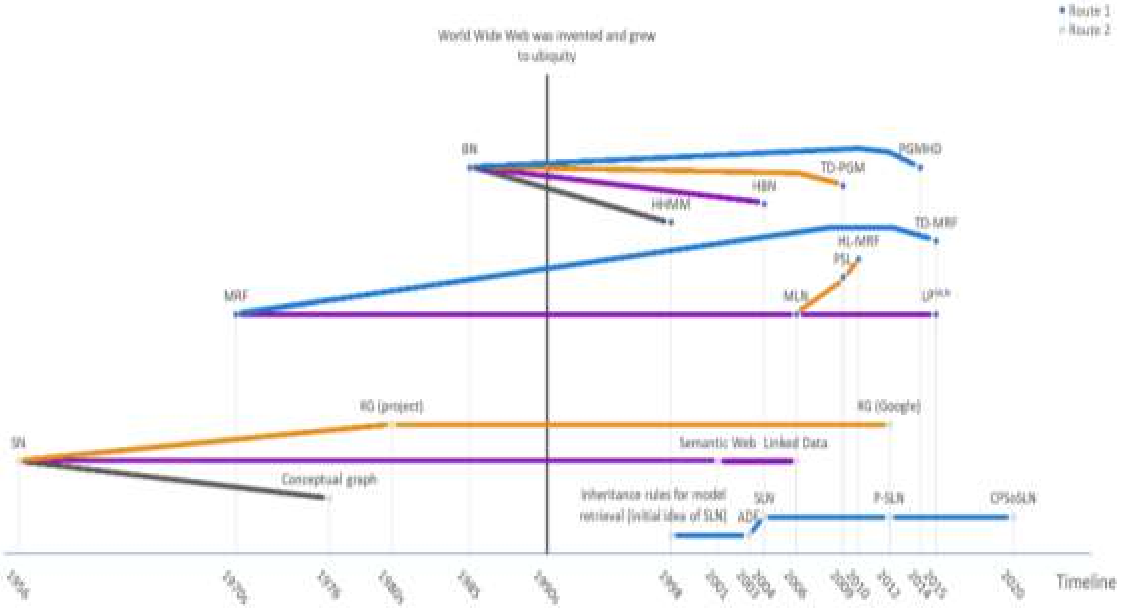


Figure 2-6 A timeline of the efforts

	Goal	***Graph semantics***	***Can handle uncertainty?***	***Strength***
Classic PGMs ***(BN and MRF)***	To obtain, represent, and query large probability distributions efficiently.	Node: random variable. Edge: direct probabilistic interaction between these random variables.	Yes	They provide an effective formalism for obtaining, representing, and querying large probability distributions. The visible structure is helpful to design and motivate new models. The inspection of the graph can provide insights into the properties of the model, including conditional independence properties.
Extensions of Classic PGMs ***(HHMM, HBN, PGMHD, TD-PGM, TD-MRF, etc.)***	To handle probability with more information.	Node: random variable, or group of random variables. Edge: direct probabilistic interaction between these random variables.	Yes	They enable PGMs to represent the dependences with structured domains and handle probability with more information such as time and hierarchy.

MLN and PSL	To handle uncertainty in knowledge bases.	Node: random variable corresponding to possible grounding of predicate appearing in the first-order knowledge base. Edge: direct dependence between these random variables, yielded according to the knowledge base.	Yes	They provide an efficient way to deal with uncertainty in first-order knowledge bases via PGMs. They not only represent the structure of dependences via their PGMs but also represent the semantics of knowledge via their logic formulas. They handle issues via logic as well as statistics, which is helpful for inductive reasoning.
SN	To represent knowledge in graphs, build links between data, and then discover new links.	Node: concept or resource. Edge: semantic relations between these concepts and resources.	No	It provides a graph-based formalism to describe the information with rich semantics. It represents the structures of knowledge intuitively and describes various semantics logically, which is useful to explore the information automatically, deal with semantic queries, and achieve reasoning.

Basic SLN	To describe and understand self-organized social semantics, with relational reasoning, and pursue semantic richness.	Node: anything, including text, image, individual (human or agent), event, concept, class, or even an SLN. Edge: semantic relation between the semantic nodes, indicated by a relation indicator.	No	It represents rich semantics of knowledge in a graph-based formalism, especially the diverse semantics of relationships. It provides a way to manage various semantics, do reasoning on the network, and discover new relationships effectively.
P-SLN	To reflect and handle uncertainty in SLN.	Node: resource or resource cluster, which can represent anything, just like nodes in the basic SLN. Edge: semantic relation with a probability range to reflect the probability of the semantic relation derived from multiple semantic paths.	Yes	It enables SLN to reflect and deal with uncertainty.

Table 2.2 Brief Comparison of Models.

The development of these models benefits from insights into human-cognition processes. For example, BN is inspired by the characteristics of human inference [56]: Human judgments are issued swiftly and reliably for a small number of propositions, but difficultly and hesitantly for a conjunction of many propositions. This suggests that the elementary building blocks that make up human knowledge are not entries of a giant joint-distribution table, but rather low-order probabilistic relations between small clusters of semantically related propositions. Besides, a person who is reluctant to give a numerical estimate for a conditional probability $P(A \mid B)$, can normally state whether propositions A and B are dependent or independent given

C without hesitation. Evidently, the notion of conditional dependence is more basic than the numerical values attached to probability judgments. This suggests that the fundamental structure of human judgmental knowledge can be represented by dependency graphs and that the mental tracing of links in these graphs is responsible for the basic steps in querying and updating that knowledge. These insights into human judgments led Pearl to introduce graphics to describe the influences between the propositions and use graphical operations to query and update the probability distributions. New models could be developed with in-depth understanding of some human-cognition processes. Relevant work concerns the philosophy and future AI [141].

2.3.3 To Sum Up

A combination of graphics, uncertainty and semantics can form a stronger model to help efficiently understand complex reality and support applications.

Probabilistic graphical model is a combination of graphics and uncertainty. It can efficiently obtain, represent, and query large probability distributions. Its visible structure is beneficial to design and motivate new models. The inspection of its graph provides insights into the properties of the model, such as conditional independence properties. Its two major families, Bayesian networks and Markov random fields, render probability distributions in different forms. The Bayesian network renders the probability distribution as a product of local conditional distributions and is good at describing the causal relationship. Whereas the Markov random field renders the probability distribution as a product of potential functions defined over the cliques in its graph, and it is good at representing symmetrical relationships. The extensions of Bayesian network and Markov random field enhance their expressiveness. Hierarchical hidden Markov model and Hierarchical Bayesian Network enable Bayesian networks to deal with hierarchical information and structured domains. Time Delayed Probabilistic Graphical Model and Time-dynamic Markov random field enable probabilistic graphical models to consider the temporal information and reflect the time-delayed dependences. Probability graphical models lack the capability to carry plentiful semantics. Therefore, they cannot represent domain knowledge, and some methods are needed to apply probability graphical models to deal with the uncertainty in knowledge bases.

Markov logic network combines Markov random fields and first-order logic in a single representation. Probability soft logic combines probability graphical models and fuzzy logic. They provide a way to apply probability graphical models to first-order knowledge bases and can be regarded as a combination of graphics, uncertainty, and semantics. They can describe domain knowledge by their logic formula part and meanwhile handle uncertainty by their probability graphical model part. They treat issues via logic as well as statistics, which is helpful for inductive reasoning. As the domain knowledge is only listed as logic formulas, it is tricky to further explore the relationships hidden in the formulas, especially the semantic information.

Graph-based knowledge representation models (including the semantic network, conceptual graph, Semantic Web, Knowledge Graph, and Semantic Link Network) combine graphics and semantics. They try to represent domain knowledge in a graph-based formalism. They provide an infrastructure for reasoning, discovering new relationships, and processing semantic queries. To handle uncertainty, some Knowledge Graph implementations provide methods that use Markov logic network to deal with probability, and some use machine learning methods. The methods based on the Markov logic network are usually difficult to inference on large-scale graphs because of the high computational complexity. The methods based on machine learning cannot apply logic rules and are hard to interpret. To reflect uncertainty, the Semantic Link Network is extended to Probabilistic Semantic Link Network by associating its semantic links and reasoning rules with probability. The recent development of the Semantic Link Network is the Cyber-Physical-Social Semantic Link Network model, which unveils the structure of the emerging cyber-physical society.

2.4 Software Engineering Approaches Background

Software engineering is a systematic approach to developing the software systems. The methods, techniques and tools used depend on the organization which is developing the software according to customer’s needs. Previous studies have reported [18], there is no single software engineering method that fulfils all requirements so the most significant aspect is to determine which software engineering methodology is most suitable to build a specific type of application. Table 2.1 gives some types of software applications [18].

Stand-alone Application	These are the applications that are run on a personal computer or on a mobile.
Interactive transaction-based application	Interactive transaction-based application systems are the web based application that can be accessed from a mobile device or personal computer through the internet.
Embedded control systems	These are the software used to control hardware devices.

Batch processing systems	Batch processing systems are used to process data in large batches and are normally used to support business systems which are used to process the payments, salaries and bills.
Entertainment systems	These are systems which are related to computer games and films. These systems normally need high resolutions hardware to support the best quality.
Systems for modelling and simulation	These systems normally need parallel execution for high performance and are used to map the real-world scenarios.
Data Collection and analysis systems	Data collection and analysis systems collect data from the environment using sensors and send it to the central system for further processing.
System of systems	These systems are used by the enterprises, in which some are generic and some are specially developed for the companies.

Table 2.3 Various types of the software applications.

Each software type given in the Table 2.1 requires a specific type of software engineering approach.

A. Internet based software engineering

The development of the World Wide Web (WWW) has had special effects on our lives. Initially, the WWW was used to run a software system which is only accessible within an organization. In 2000 [18], the web-based software systems start to evolve and more enhanced versions started to develop. Web server-based approach makes the development of software very easy. Programmers and engineers need to upgrade only the main server system and customers and users can access the updated information from the main server while accessing on the Web. This approach is much easier and cheaper as there was no need to install the software on each and every customer Personal Computer (PC). During that time a lot of business shifted on to the web-based software systems. This software as a service approach was first proposed in the early 21st century [18] recent evidence suggests that is nowadays used as a standard approach. The customer can access the software and use the application software while using the web-based software systems remotely. The *cloud computing approach* is a huge number of linked computers and much cheaper for the user as they do not need to buy whole software system. Customers can pay how much they access and use the software. The server-based approach is very suitable where the customer gets information or uses the software online, but it's not feasible when the customer needs to process data on client-side, i.e. when the customer has to process the large scale of data on client-side.

B. Cyber Physical System based software engineering systems

Software systems now operate globally and in a rapidly changing environment. The issue has grown in importance in light of recent large system development. They have to respond to and fulfil the requirements of these changing markets, changing economic conditions, and competing services. These systems are operating in a changing environment, so it is practically impossible to rely on traditional software development approaches. Plan-driven software engineering is a software engineering approach that is completely based on the customers' requirements and it design could not compete for this fast-moving development process in the current environment. There is a need for a semantics-based software engineering approach which can handle these changes and predict and purpose the suitable solutions for developing the software projects.

Large software systems such as air traffic control systems generally have an extended lifetime, For example, the military system has an average lifetime of 30 years and successful software products and applications are usually developed many years ago. The first version of the Microsoft Word was introduced in 1983 its more than 30 years [18]. Business changes and user expectations are accomplished by releasing with new versions every few years. Software system adapts evolution during their initial development to the final phase. It is suggested by the data that between 60% to 90% cost of the software are evolution costs so companies have to use the software for the long run to recover the investment cost. Questions have been raised about the safety of prolonged use of these systems that's why the development of these software systems requires development methodologies.

C. Managing Software Processes

A software system is a production of a set of *associated activities*, which are important pillars of software engineering models. Four general high level software activities are discussed in this section which are the main parts of all software production process. These are the four essential software process activities: specification; development; testing; and evolution [18].

Four major activities have further sub-activities such as user requirements, design, testing and maintenance. Software processes are multifaceted in nature and it's often difficult to choose the right software engineering model to build the solution. It's usually based on the software engineer’s judgment to select the most effective and suitable model according to the nature of a software product. The classification characteristic of RSM makes this job easier for the development team.

There are many different types of software engineering models existing to develop the software systems, However there is no universal software model invented which could apply to build all types of software systems [18]. Several attempts have been made to design a universal model for software development. Many companies even build and design their own software models to solve the specific problem but not suitable for the general purpose to find a solution for all kind of software projects. There is a scope to improve the existing models and also to purpose the new software models and techniques.

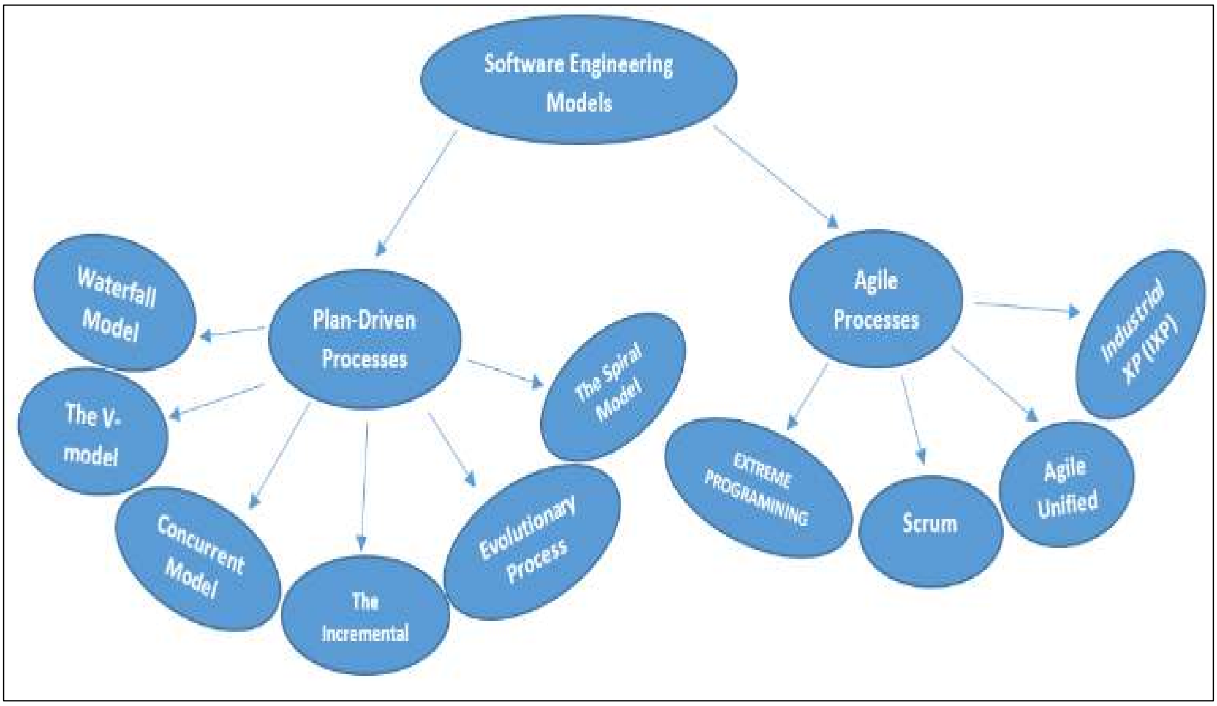


Figure 2-7 Software Engineering Models

There are two main software engineering approaches, *Plan-Driven Approach* and *Agile-Approach* established on software engineering models shown in Figure 2-7. Each approach has software engineering models some of which are shown in Figure 2-7 have the inherent nature of their own methodology.

Many efforts have been made to develop a universal software engineering model to cover all the major returns of the existing models. Rational Unified Process (RUP) is one of the best model developed by the United States software engineering company called Rational. RUP was created on the most known and widely used software engineering methods but not gain more acceptance. But it left a gap and scope to develop new approaches and methods for achieving the ultimate goals [31].

2.4.1 The Plan Driven Approach

The Plan-driven approach is a “traditional” way to develop the software. This approach has numerous advantages to avoid severe economic damage and provide high assurance, predictability and stability [32]. Traditional methods are preferred due to advance planning, documenting, offshore development, reliability, safety, and high-quality control [21], [27]. This approach uses specific formalized requirements, explicit, documented knowledge to develop the safety-critical systems for better coordination between the large developing teams [32].

A. The Waterfall Model

The Waterfall model is the first published model in software engineering process designed to develop the large military software systems in 1970 [18]. This model is applied when the requirements are well understood, cleared and stabled. It is implemented in a linear fashion and based on cascade nature, which means when one phase finishes then another phase will start. This linear approach is based on the systematic progressive approach that starts with customer’s requirements and performs a final phase of maintenance as shown in Figure 2-8 [18].

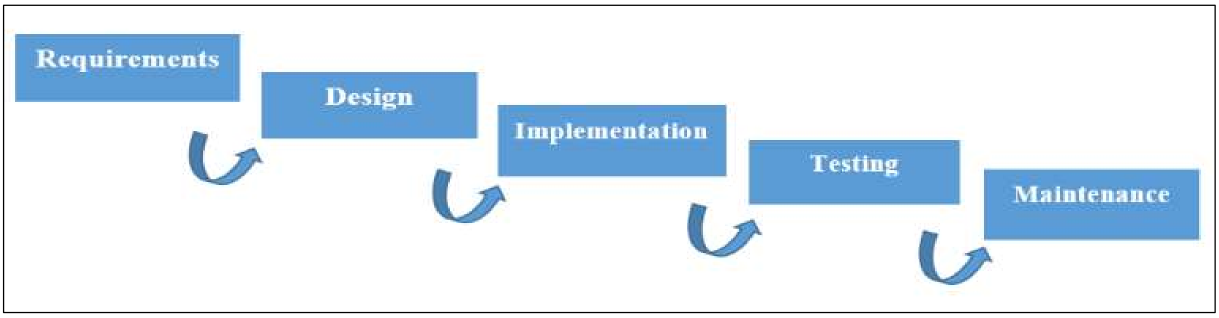


Figure 2-8 Waterfall Model

This model is also known as the baseline for many other software models [27]. Although many modified waterfall models have been introduced still the pure waterfall exit in its original shape and used in many software projects.

B. Incremental Process Models

The Incremental model delivers a series of software products which offers more functionality to the customers. In some situations, customers need a software package with some crucial functionalities that can be enhanced in later versions, in those scenarios an incremental model is the best selection [18]. This model is the combination of linear and parallel process flows, shown in Figure 2-9. The first software product in this process is called the core product which can be enhanced with known or unknown functionalities on later increments. The customers gave their feedback while using the core product and inform if need any further functionally or need to improve the existing software system. Based on this feedback its quality and functionally can be improved in the coming versions.

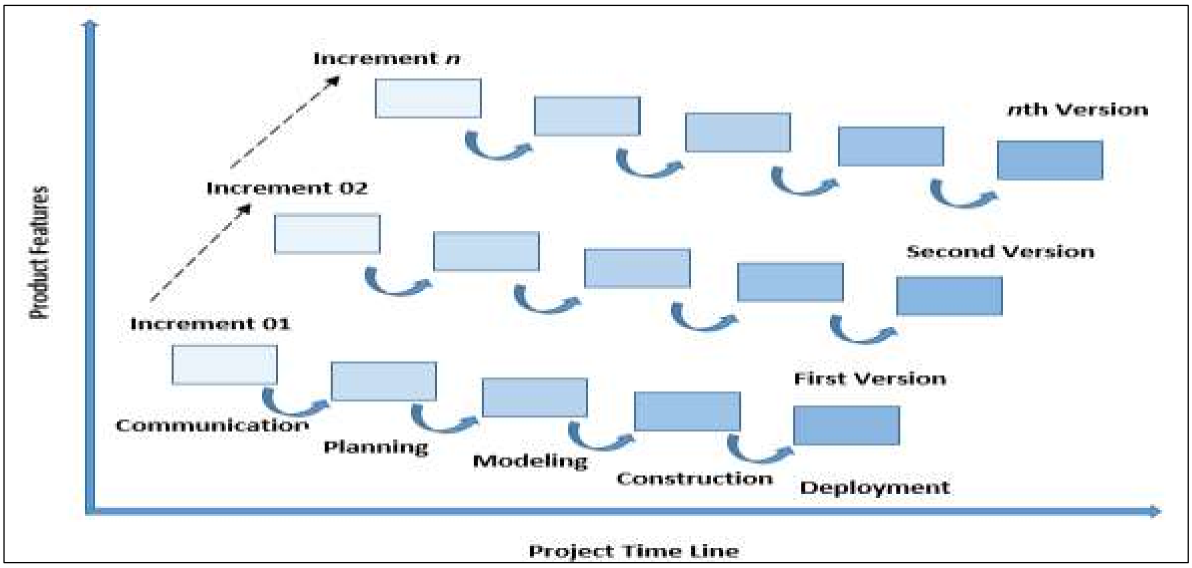


Figure 2-9 Incremental Model

2.4.1 The Agile Approach

In the current era, the market is growing at rapid speed and businesses are transferring from traditional ways to online. The concept of the plan-driven approach has recently been challenged by studies demonstrating in [18] that requires a new software approach to deal with the current software engineering challenges. This modern business tactic can be achieved using a software engineering approach called Agile. The Agile process's main goal is customer satisfaction through continuous increments. Agile plays a vital role where things getting quickly change even in those situations where requirements change in the last development phase [24]. To accommodate these rapid change software engineers must be fast and Agile [18]. Agile uses simple design which is easy to rework if need to enhance. Change is unavoidable in real-world projects, planning for future projects to avoid chance is a waste of time and effort [32].

Agile software engineering is a philosophy and also a set of development activities. The philosophy is end-users and stakeholders satisfaction with the product which delivered to them on commitment date. Agile software engineers, customers and other stockholders are working together most of the time during the software development process. Informal and verbal communication establishes between all the members of the team, based in a single room or sometimes in a small company instead of distributed offices or organizations. Agile uses informal, user-prioritized stories as project requirements [17] [32] to avoid unnecessary documentation. Agile software engineering models are more effective to develop small and medium enterprises (SMEs) software systems and mostly used in the organization where customers directly involved in the development processes.

Agile methodologies are people-oriented that believe people are a main success factor in the project and considered a very important role in the project development life cycle [21]. Agile developers need more talent, communication skills, and amicability including technical skills [32]. Extreme Programming (XP) and Scrum are the Agile-based models described in the following sections. The Agile approach develops from rapid prototyping and believes in the philosophy that programming is a craft than mechanical process [32].

A. Extreme Programming

Extreme Programming (XP) is a widely used Agile model which based on refining knowledge and experience of developing the information systems [18][32]. XP is focused on verbal (informal) collaboration between developers and customers, to avoid unnecessary documentation and continue to improve functionality according to their feedback. Simplicity is a core principle of Agile manifesto which precise software engineers to design the system only with immediate needs. Developers use Object Oriented Programming (OOP) and pair programming approach for implementation where two programmers work together on the same project [17], [18]. XP model uses a set of practice and processes which continually collaborating on planning, design, coding, and test phase that is shown in Figure 2-10 [18].

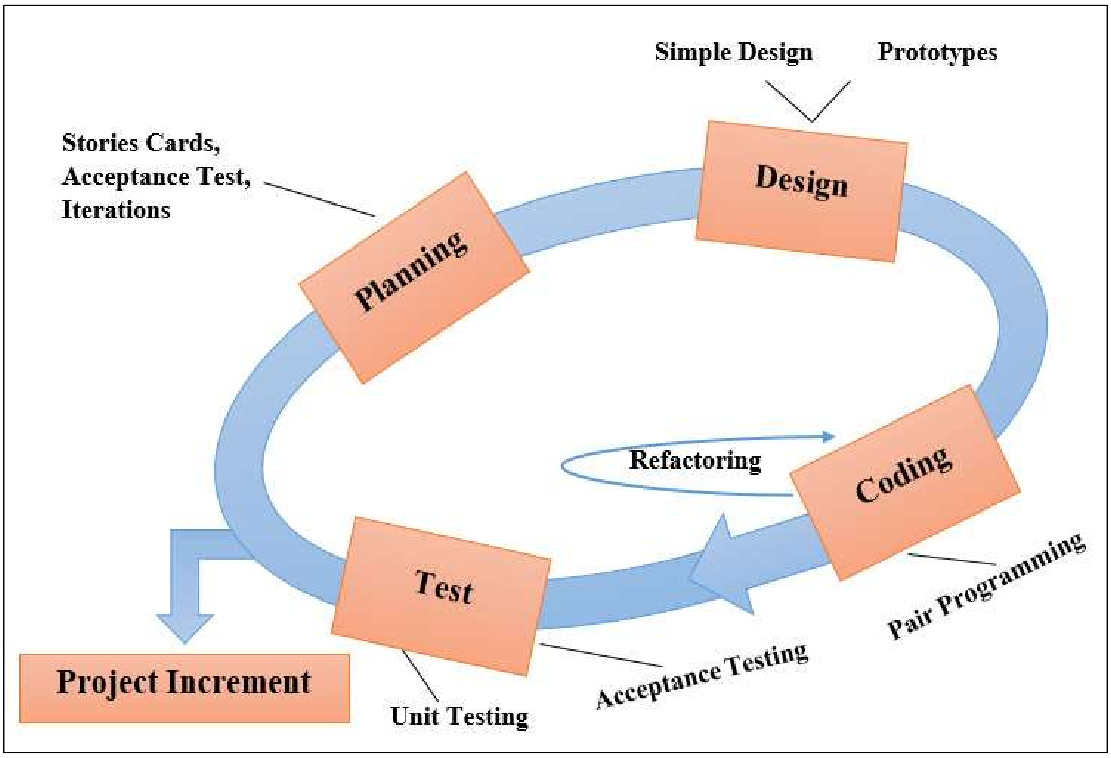


Figure 2-10 Extreme Programming Model

B. Scrum

Scrum was originally designed to speed up the software development process and used to deliver the most needed functionality. Scrum relies on self-commitment, self-organization, and emergence rather than authoritarian measures [22]. It gives more freedom to the development team to engineer the solutions and includes a daily meeting to gauge the developing progress. Requirements those have high priority send in the backlog at any time. The Backlog is based on the prioritized requirements list which has the information of features those are required to implement. Sprints are the tasks which normally be complete in a time frame of 2 – 4 weeks iterations shown in Figure 2-11. Scrum meetings are short but held on a daily basis to maintain coordination [18]. Scrum is mainly applied to software projects although it has been used for non-software projects. The Scrum principles are same to manage any project [32].

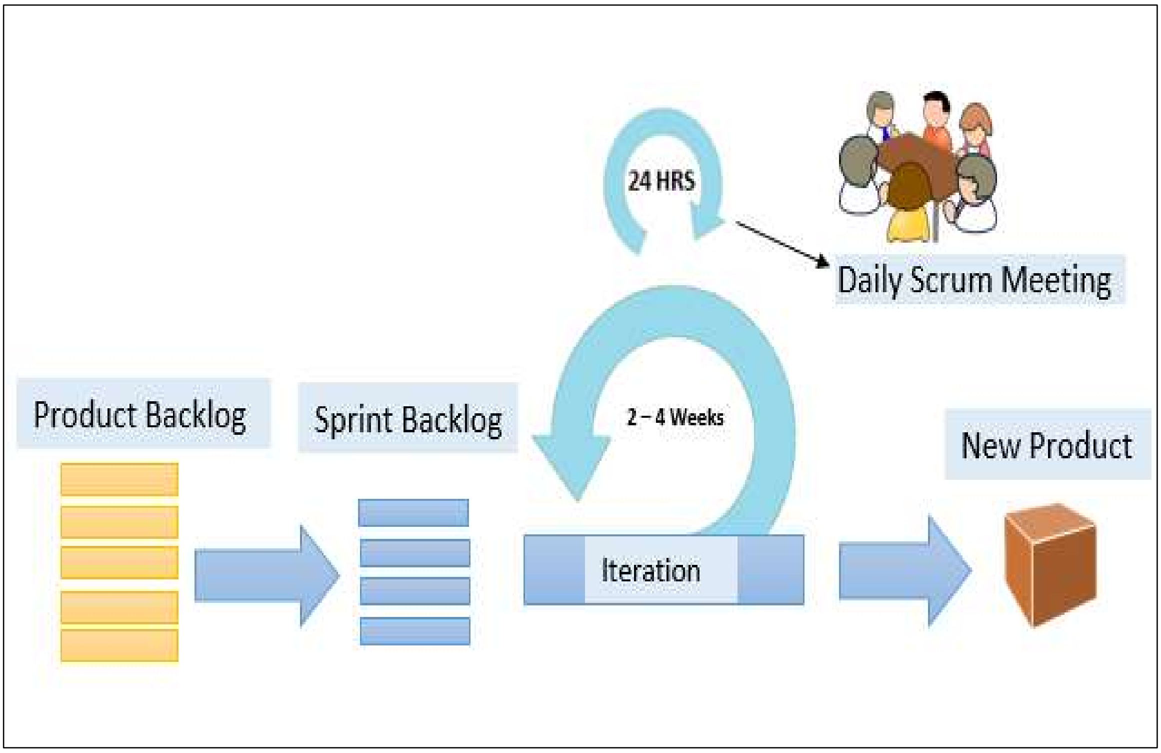


Figure 2-11 Scrum Model

2.5 Summary

This chapter reviews existing literature on the topic of text summarisation, Semantic Networks were summarized following by its six types, and software engineering processes. A combination of graphics, uncertainty and semantics can form a stronger model to help efficiently understand complex reality and support applications. Semantic Link Network, which is more capable of modelling semantics than traditional graph structure [53]. Software engineering is an efficient methodology to developing the software systems. The methods, techniques and tools used depend on the organization which is developing the software according to customer's needs. Previous studies have stated [18], there is no single software engineering method that fulfils all requirements so the most significant aspect is to determine which software engineering methodology is most suitable to build a specific type of application [31]. But it left a gap and scope to develop new approaches and methods for achieving the ultimate goals. The chapter wraps up with an analysis of the challenges facing the present software engineering and text summarization research with an indication of the problems addressed in the current work and those contributed to their solution by adopting the RSM and SLN.

3 CHAPTER THREE: EFFECTIVE SUMMARIZATION BASED ON SEMANTIC LINKS

This research proposes a clause-based extractive summarization algorithm by ranking and extracting semantic clauses from an original document. Discourse structure relation is useful for identifying semantically important parts of the source document. We segment the document into clauses and evaluate the importance of clauses based on semantic relations, rank and extract them coarsely, and utilize graph ranking to refine the extracted clauses. This way can create a more concise summary with more information and less redundancy. Research reach the following results: 1) compared with the other summarization algorithms on different granularity, the clause-based summarization achieves higher recall score; and, 2) different discourse relations have different importance.

3.1 Introduction

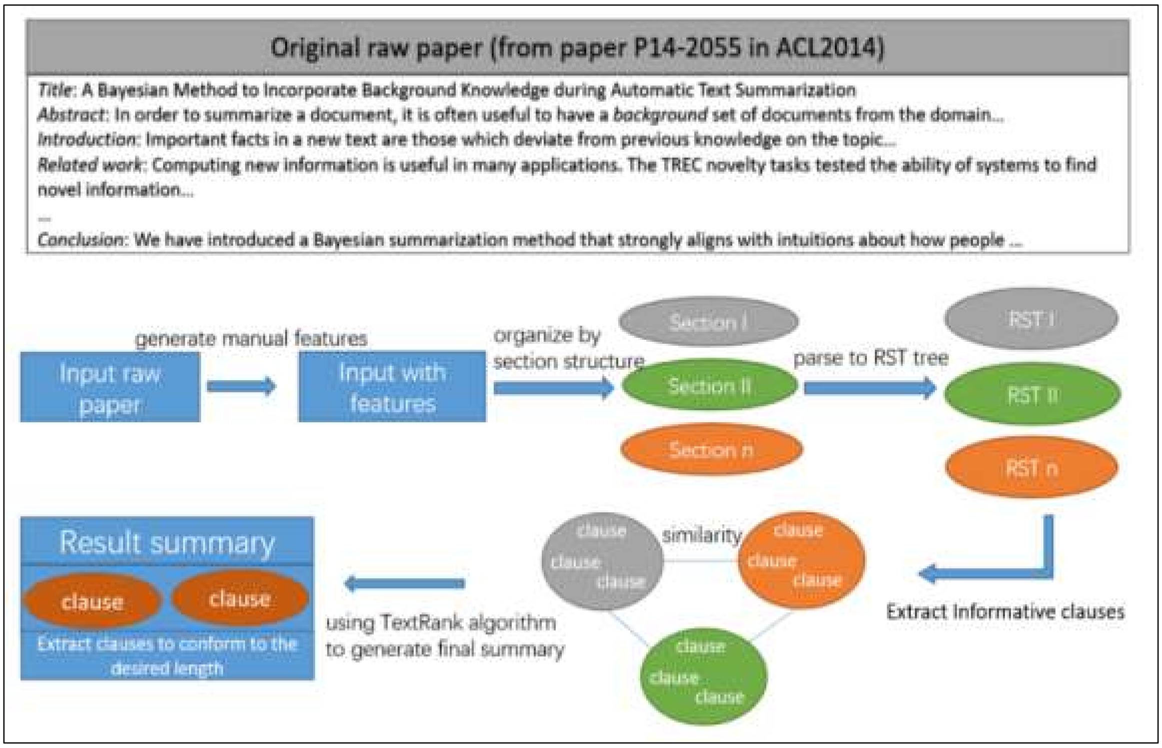
The purpose of text summarization is to represent important information according to original document. Text summarization tasks can be divided into extractive summarization or abstractive summarization, single-document summarization or multi-document summarization, general summarization or query-based summarization [89]. The extractive summarization is to extract important sentences from source texts to compose a summary. Some summarization approaches adopted different units [90], such as word [93], phrase [94], sentence, sentence group or paragraph [95], which is more coherent but with more redundancy. Limited by the desired length of the generated summary, the recall and precision of existing models can't get both better, namely, we can't improve the diversity of important words (recall) and make the summary as short as possible (precision) at the same time. Therefore, we need a more flexible unit to refine the summary with less redundancy.

Discourse structure analysis is a way to understand the semantics of documents. It can reflect semantic information and indicate the importance of each text fragment [91], lots of research works have been done on the definition of discourse relations in text such as tree structure and graph structure [92]. Discourse relations are defined to hold between two non-overlapping text spans, and the spans are segmented by structure. Therefore, it retains the semantic integrity and less redundancy which is not presented in other granularity.

This research proposes an extractive summarization method that can be regarded as clauses and extracts important parts based on discourse relations. We evaluate the proposed algorithm on scientific literature and compare it with some other extractive summarization systems. The result shows that the summary which consists of clauses has better performance than other systems with different granularity.

3.2 Related Work

Discourse relation structure is a linguistic relation model based on a set of predefined semantic relations among natural language texts. It can handle different relations between segmented spans, such as cause-effect, elaboration, and same-unit, and the relations and clauses as nodes are used to construct SLN. Rhetorical Structure Theory (RST) was proposed by Mann with a tree structure to reflect semantics and organize text [96], it is a kind of discourse structure relation model. It defines various concrete relations and the format implementation. RST addresses text organization by means of relations to hold between parts of text, and is widely used in natural language processing task, such as text generation and text summarization. RST and other text features were used to find informative content in summarization [97]. The discourse relations were classified into positive and negative by their semantic meaning with different weights for sentimental analysis [98]. The influence of granularity is also studied, and adopted phrase and sentence respectively as basic unit to summarize, and [15] compared the performance of sentence, sentence group and paragraph level. In our work, we adopt the clause which is segmented semantically by discourse relation. We make use of it to rank and extract informative clauses coarsely and refine and generate summary based SLN. Different from other algorithms, we only use the RST and SLN model instead of machine learning techniques to reflect the importance of semantics in summarization. A text summarization approach that extracts semantic link network from scientific paper has been proposed in [53].



Original raw paper (from paper P14-2055 in ACL2014)

Title: A Bayesian Method to Incorporate Background Knowledge during Automatic Text Summarization
Abstract: In order to summarize a document, it is often useful to have a *background* set of documents from the domain...
Introduction: Important facts in a new text are those which deviate from previous knowledge on the topic...
Related work: Computing new information is useful in many applications. The TREC novelty tasks tested the ability of systems to find novel information...
...
Conclusion: We have introduced a Bayesian summarization method that strongly aligns with intuitions about how people ...

Figure 3-1 SLN-based extractive summarization pipeline

3.3 Proposed Model

Figure 3.1 illustrates how to generate a more concise summary of scientific literature. First, we need to convert the raw text into the RST format. The whole paper is too long to generate a good RST tree, therefore, we segment it into section structure and parse each section into an RST tree. Then, a score is computed for each clause based on its relationship in discourse relation analysis, and preserve informative clauses. Finally, we use the SLN rank algorithm to choose the top-K clauses as the generated summary. Overall, the model is a coarse-to-fine structure, and discourse relation in the model is to filter the irrelevant content to the summary coarsely.

3.4 Extract and parse into RST tree

RST is a tree structure for representing semantic relations between clauses of a document. Relations are defined to hold between two non-overlapping text spans, and its role in the relationship can be *nucleus* (important units) or *satellite* (relatively unimportant units) to indicate its importance in the relationship, an example is shown in Figure 3-2. Lots of RST parsing tools, such as DPLP [99] and HILDA [102] are available. DPLP parser shows an F-score improvement of around 2.5% in nucleus and 6% improvement on relation type prediction over previous tools based on an SVM [100] classifier trained on RST Treebank corpus. In this work, we use DPLP to segment text because of its better performance. Most of the parsers are implemented as a shift-reduce structure and require manual features as input. The parser requires POS tags, the distance from the beginning and other manual features to parse the text to the RST tree. The features can be extracted by the Stanford CoreNLP toolkits [101].

For preprocessing scientific papers, we separate them into section structure. In the ablation study, we found each section has different focusing points and may generate a bad RST tree between crossing sections. And DPLP is trained on shorter corpus than papers, it fits better on the shorter text (i.e., a section) instead of the full paper.

1. The next music day is scheduled for July 21(Saturday), noon-midnight.
2. I'll post more details later,
3. But this is a good time to reserve the place on your calendar.

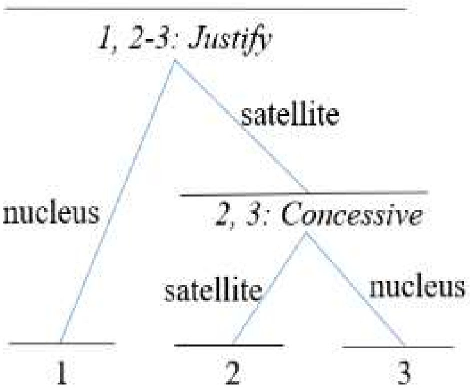


Figure 3-2 An example of RST

3.5 Choose informative text clauses

Based on the nucleus or satellite pattern and the category of relationship, we can evaluate the importance of segmented clauses. The *clause* is shorter than its original *sentence*, namely, its length and position of the clause are not fixed. We regard the clause as the basic unit to summarize. Towards composing a summary, we need to rank and extract more informative clauses.

Algorithm 1 shows how to evaluate a clause by its relationship and role, where r_i represents an RST tree of a section, and s_i represents the clause segmented by the RST algorithm. We mainly use ***P*** as the coarse summary, which drops the redundancy and keeps the informative content. The importance of the relationship is selected empirically and adjusted according to the performance. This algorithm works as a coarse filter to drop detailed parts of the sentence, because the count of informative clauses can't be controlled easily, namely, the heuristic rules can't handle the situation that multiple clauses have equal score.

Algorithm 1 The Clause Ranking and Selection Algorithm

```
Input: a set of RST object $R$ = {$r_1$, $r_2$, …, $r_n$}
Output: a set of preserved clauses $P$, dropped clauses $D$
$P \leftarrow \emptyset$
$D \leftarrow \emptyset$
for each $r$ in {$r_1$, $r_2$, …, $r_n$} do
    for each $s$ in $r$.clauses do
        if ($s$.role = nucleus and !is_info($s$.relation)) \
        or ($s$.role = satellite and is_info($s$.relation)) then
            append $s$ to $P$
        else
            append $s$ to $D$
        end if
    end for
end for
return $P$, $D$
```

3.6 Refine summary by ranking on Semantic Link Network

After obtaining a set of informative clauses, a trimming algorithm is required to refine and trim the content to conform to the desired length of the summary. An SLN is a graphical semantic model that is suitable for summarization. The extracted clauses can play the role of semantic nodes and the similar-to relations as the semantic link. The value of the similarity link can be calculated easily in various ways between clauses. In our case, we choose the simplest Jaccard function. The extracted clauses are picked carefully, therefore, we expect to get better performance compared with the identical algorithm applied to the full text.

TextRank [93] is a sophisticated summarization algorithm inspired by PageRank to rank the nodes by their values on links, it iteratively computes the weight by the similarity graph. TextRank function can rank SLN that consists of clauses, and we can change the threshold to generate a flexible summary. Let $v(u_i)$ be the importance value of a node u_i (i.e. clause), and

$Link(u_i)$ is the set of nodes connected to the node u_i. The TextRank algorithm will update the importance of nodes iteratively as shown in Eq. (1). Finally, we can get the importance of each clause within the SLN, and rank them by the value.

$$v(u_i) = (1 - \alpha) + \alpha * \sum_{u_j \in Link(u_i)} \frac{sim(u_i, u_j) * v(u_j)}{\sum_{u_j \in Link(u_i)} sim(u_i, u_j)} \qquad (1)$$

The hyperparameter α is used to adjust the proportion of value from language units in $Link(u_i)$ as a random walker. The common value is set to 0.85. The length of the generated summary is flexible when we adjust the threshold to cut off the clauses for conforming to the desired length.

3.7 Evaluate the performance of summaries

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [103] is a common metric to evaluate automatically the performance of a generated summary. ROUGE-N ($N \in \{1,2\}$) represents the *n*-gram units used to evaluate how well the summary matches the standard reference texts which are written by humans. ROUGE-L addresses the matching of words sequence by Longest Common Subsequence. Generally, we consider *recall* score more than the *precision* score in summarization when the length of summary is fixed to a predefined length of words (for example, 200 words). This work adopts ROUGE-N and ROUGE-L to evaluate the generated summary, the desired length is designed to match the standard abstract.

3.8 Dataset

The dataset we used contains 173 ACL2014 conference papers collected from ACL Anthology. Table 3-1 shows the brief description of the dataset. Overall, each paper has 3989 words on average, and we use *Abstract* of paper as the standard summary.

	Min	**max**	**average**
Text	1865	6670	3989
Abstract	53	220	116

Table 3.1 The Words Amount of Papers in Dataset

	Min	**max**	**average**
#RST-words	1125	4511	2604
Full-text Rouge-1	0.640	0.962	0.815
RST Rouge-1	0.526	0.930	0.767

Full-text Rouge-2	0.191	0.827	0.436
RST Rouge-2	0.122	0.659	0.379
Full-text Rouge-L	0.593	0.949	0.788
RST Rouge-L	0.516	0.899	0.744

Table 3.2 The Performance of RST Summary.

	Rouge-1	**Rouge-2**	**Rouge-L**	**#words**
Complete (Ours)	**0.355**	**0.101**	**0.319**	186
Group (Cao)	0.336	0.086	0.296	187
Reinforce (Sun)	0.284	0.076	0.251	**118**

Table 3.3 The Performance of Different Models

	Original and selected clause
Case1	For example, in the simplest setting of multi-document summarization of news, systems are asked **to summarize an input set of topically-related news documents** to reflect its central content.
Case2	In this work, we compare our system to topic word-based ones since the latter is also a general method **to find surprising new words in a set of input documents** but is not a bayesian approach.
Case3	Rather the method creates a summary **by optimizing for high similarity of the summary with the input word distribution**.

Table 3.4 The Examples of Proposed model.

3.9 Experimental Results

Table 3.2 shows the performance between the RST summary (before applying TextRank) and the original text. It indicates how semantic features influence the quality of the extracted informative clauses. After constructing RST and extracting clauses, the count of words decreases to an average of 35% compared with the original full text, we can find that the key information still remains after RST trimming. The effect of the semantic discourse structure is to help remove details and useless content and keep the informative parts.

Table 3.3 shows the performance of summaries generated by our model and other models respectively. Cao et al [95] proposed a group language unit method to extract summary, and Sun et al [53] used reinforcement ranking on the summarization of scientific paper. By comparing different models of SLN ranking on the same dataset, we find that the semantic features are important to extract informative content. Although we get better performance, the proposed model is more complex in practice which relies on the POS tagger and RST parser.

Table 3.4 shows several examples generated by our model, where the extracted clauses are highlighted in bold which is selected as part of the generated summary.

The following implications can be drawn from analysing the result.

Implication 1. *Semantic discourse relations can indicate informative clauses and helps make a summary.*

The semantic discourse relation can be used to remove redundancy and keep the informative parts. It describes the relationship between two or more parts of a sentence, compared with the original text and extracted RST clauses in Table 3-2, we conclude that semantics is a key feature to indicate important information.

Implication 2. *Different kinds of semantic relations have different importance in discourse structure.*

In experiments, we set higher weights to *cause-effect*, *purpose*, *contrast*, and *topic* which means more informative, while *elaboration*, *list*, *same-unit*, *textual organization*, *attribution*, *restatement*, and *means* are discarded in summarization. We got the best performance under this configuration, it can be adjusted by preference and feedback.

Implication 3. *The clauses are flexible for extractive summaries.*

Comparing with the summarization with other granularity, clause is more flexible to generate summary with less redundancy and more information. However, clauses don’t follow the complete syntax rule, post-processing is required to make summary fluent and coherent. We can insert some functional words or remove prepositions to complete it.

4 CHAPTER FOUR: RECOMMENDING RESEARCH COLLABORATORS BASED ON SEMANTIC LINK

Recommending appropriate collaborators to researchers can promote their research. In many cases, however, it is difficult for researchers to find proper collaborators from large number candidates. This research proposes a scientific collaborator recommendation approach based on the semantic link networks, where nodes are authors, papers and interests indicated by keywords, and semantic links are write links, cite links, and contain links between these semantic nodes. Five semantic paths on the semantic link networks are proposed for deriving future collaboration between authors. Experiments on three datasets of scientific journal papers show that our method achieves good performance in predicting future collaborators. Comparing the combinations of five semantic paths reaches the following results: (1) co-author relationship, keyword information, and citation relationship play an important role in finding appropriate collaborators; and, (2) combining all the five semantic paths can get the best results on collaborator recommendation task.

4.1 Introduction

Successful collaboration is a way to promote the productivity of scientific research. However, it is time-consuming, laborious and difficult for researchers to select appropriate collaborators from large-scale candidates, especially for young researchers who have less information about other researchers. Therefore, it is significant to study the approach for recommending research collaborators. This paper proposes a new approach to recommending collaborators for researchers by analysing the information contained in their scientific papers.

The Semantic Link Network (SLN) is a self-organized semantic model for representing and operating the semantic structure of complex systems. It consists of semantic nodes representing categories of things and semantic links representing the semantic relations between nodes [1-17]. SLN was early proposed for managing models and realizing Active Document Framework. The SLN has powerful semantic expression capabilities, and it can be used to manage meaningful semantic relations between anything with semantic information, including the semantic relations between scientific papers and their authors. It was integrated with a model based on multi-dimensional classifications for supporting advanced applications [34-52]. Application of SLN on recommending collaborators are studies in [81]. This paper implements the idea with a computing model and verifies it on three scientific literature data sets.

4.2 Related Work

Research on recommendation of research collaborators can be classified into the following categories: content-based, homogeneous network-based, and heterogeneous network-based approaches [120].

The content-based approaches mainly focus on the similarity between the experts' own features, such as profiles, expertise, and interests. A representative content-based expert recommendation system can screen out suitable cooperation candidates according to users' query based on keyword matching [121]. Two different strategies based on the probabilistic model seek experts by using document to train a language model [122], which is used to estimate the probability of a candidate according to the query. Some similar works such as topic model and candidate model were studied by [123]. A hybrid method linearly merged the weighted language and the topic-based model was proposed by [123]. The weighted language model was based on [122], which add the weight factor into the language model. The topic-based model introduces a topic layer between the candidate and the query.

However, the content-based approaches involve only the features of experts without considering the social features of the experts. Homogeneous network-based approaches utilize social network technology to recommend collaborators. Some studies were based on social network [124]. Work group graph and workplace sociability graph are established to recommend collaborators by social matching [124]. Some other studies were based on co-author network. A two regularization-based hybrid model based on the Adamic-Adar method (neighbour-based) [125] and the Katz index (path-based) [126] proposed by [127]. These two metrics are utilized to estimate the relevance of two authors in the co-author network to recommend collaborators [127]. Some other studies took other metrics such as Jaccard's coefficient and cosine similarity into account to do the collaborators recommendation task [128].

Based on homogeneous network-based approaches, heterogeneous network-based approaches take some other information into account such as expertise [129], [130], institutional network [120], and citation relationship between the scientists and papers [131]. A hybrid recommendation algorithm based on the experts' research expertise and social network is proposed to predict co-author relationships among the biomedical scientists [132]. A background knowledge Medical Subject Headings vocabulary has been added to calculate the cosine similarity score based on TF-IDF (Term Frequency – Inverse Dense Frequency) vectors between two scientists. Combining the Jaccard similarity to measure the proximity of social network, the potential collaborator can be recommended. A meta-path-based method was proposed by building a heterogeneous network based on author, paper, and venue [131] which transformed the relevance of two scientists into different metrics meta paths between them.

However, previous studies rarely considered the semantic relation between scientists, especially the citation relationship between scientists and scientists, and scientists and papers [131][132]. Our approach is based on semantic link network of scientific paper. It takes co-author, citation, and keyword information into account to build the recommendation model and measures the relevance of two experts in semantic link networks by the path-count metric. The method is evaluated on three scientific literature data set.

4.3 Building the SLN

We view author, paper, and keyword as three basic elements in the context of scientific papers, and build semantic nodes for them. A semantic link represents a relation between objects such as “author” and “paper”. Figure 4.1 shows the schema of the semantic link network on researchers and papers.

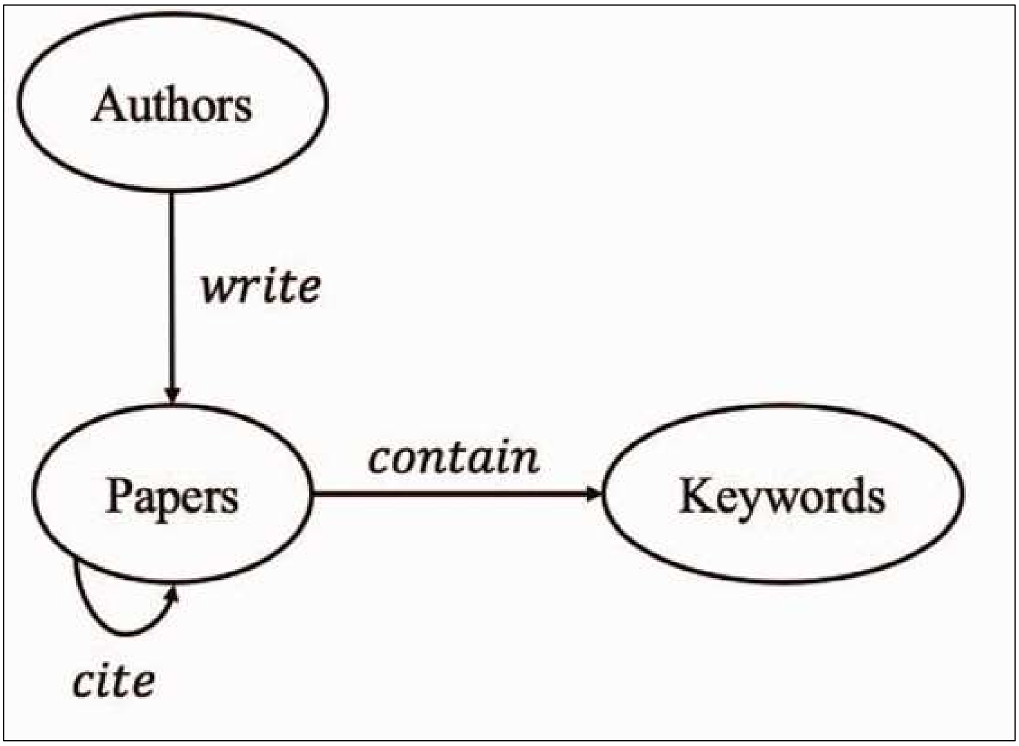


Figure 4-1 The schema of the initial SLN

4.4 Semantic Paths

Let A={A_1,A_2,…,A_N} denote the set of “author” nodes, P={P_1,P_2, …,P_M} denote the set of “paper” nodes, and K={K_1,K_2,K_L] denote the set of “keyword” nodes. A semantic path is a path in the SLN which can represent a specific semantic between two semantic nodes.

We proposed five semantic paths aimed to express the relevance between two authors. As is shown in Table 4.1.

1. *Path1* represents the semantic that author A_i and A_j research the same topic indicated by K_p. The more *Path1* between author Ai and A_j, the more common topics they had researched. Hence, *Path1* can be used to measure the similarity of two researchers' research topics or directions. The more similar of two researchers' research topics or directions, the higher the probability that they can collaborate.

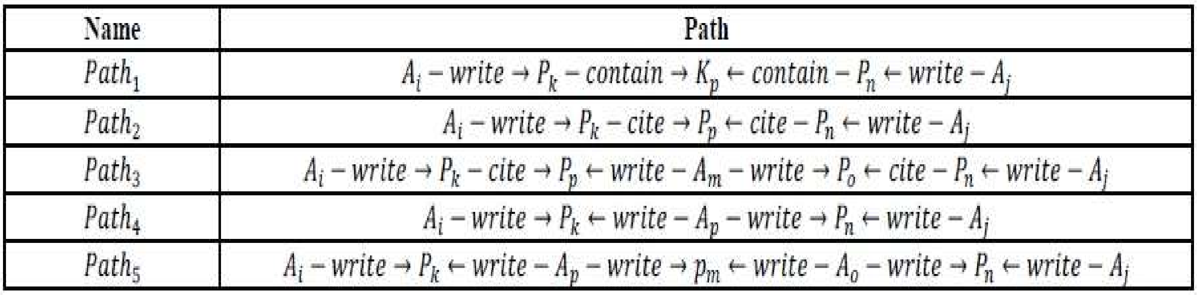
2. *Path2* represents the semantic that A_i and A_j cite the same paper P_p. The set of papers an author cited can represent his research direction potentially. The more *Path2* between author Ai and A_j, the more similar their research directions are. Similar to the *Path1*, *Path2* can be used to measure the similarity of two researchers' research directions. And the more similar of two researchers' research directions, the higher the probability that they can collaborate.

3. *Path3* represents the semantic that Ai and A_j cite the same author A_m's paper. If there is a *Path3* between author Ai and A_j, their research direction may have some relevance. The more Path3 between author Ai and A_j, the more relevance their research directions may be. And the more relevance of two researchers' research directions, the higher the probability that they can collaborate.

4. *Path4* represents the semantic that Ai and A_j have the same collaborator Am. If Ai and Am had the co-author relationship, Am and A_j had the co-author relationship, the probability of A_i and A_j can collaborate is higher than the condition that Ai and Am had no co-author relationship, Am and A_j had no co-author relationship. And the more common authors that two researchers collaborate with, the higher the probability that they can collaborate.

5. *Path5* represents the semantic that Ai and A_p have the co-author relationship, A_p and A_o have the co-author relationship, A_o and A_j have the co-author relationship. If there is a *Path5* between author A_i and A_j, that is to say, A_i ' s co-author and A_i ' s co-author had the co-author relationship. *Path5* is extended from *Path4*. It

does not require that A*i* and A*j* have a common co-author, but that there is the co-author relationship between their respective co-authors.



Name	Path
$Path_1$	$A_i - write \rightarrow P_k - contain \rightarrow K_p \leftarrow contain - P_n \leftarrow write - A_j$
$Path_2$	$A_i - write \rightarrow P_k - cite \rightarrow P_p \leftarrow cite - P_n \leftarrow write - A_j$
$Path_3$	$A_i - write \rightarrow P_k - cite \rightarrow P_p \leftarrow write - A_m - write \rightarrow P_o \leftarrow cite - P_n \leftarrow write - A_j$
$Path_4$	$A_i - write \rightarrow P_k \leftarrow write - A_p - write \rightarrow P_n \leftarrow write - A_j$
$Path_5$	$A_i - write \rightarrow P_k \leftarrow write - A_p - write \rightarrow p_m \leftarrow write - A_o - write \rightarrow P_n \leftarrow write - A_j$

Table 4.1 Five semantic paths.

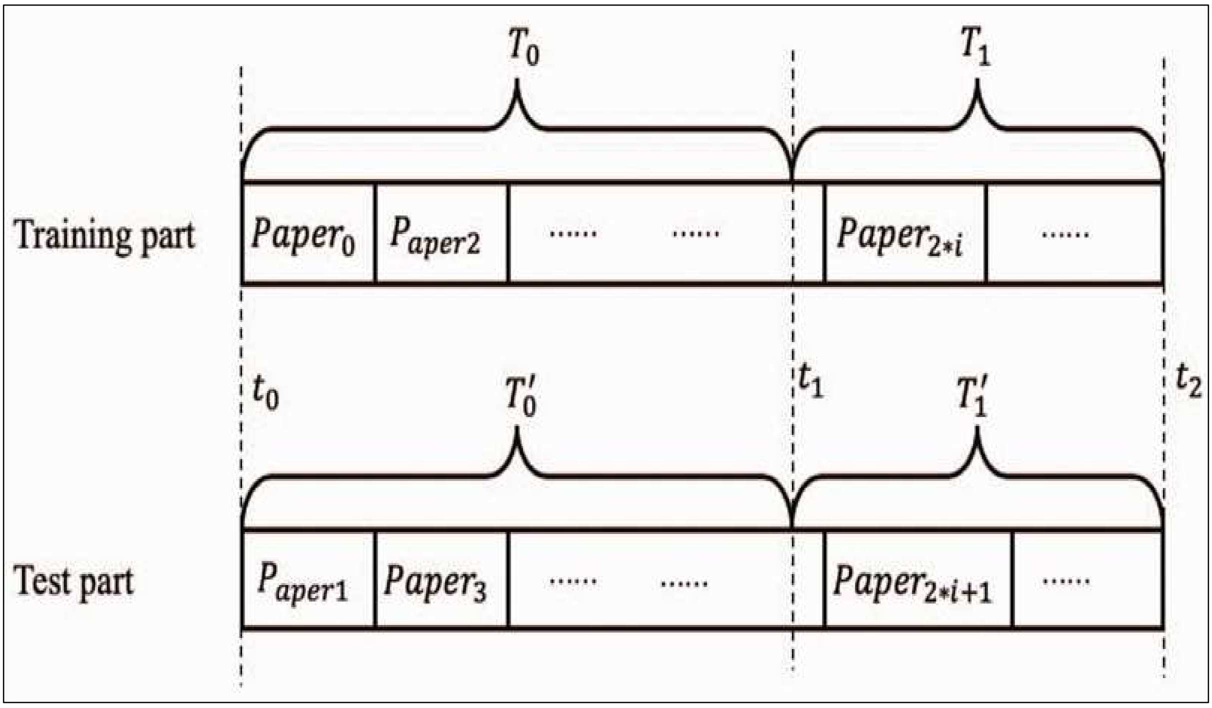


Figure 4-2 Training part and test part.

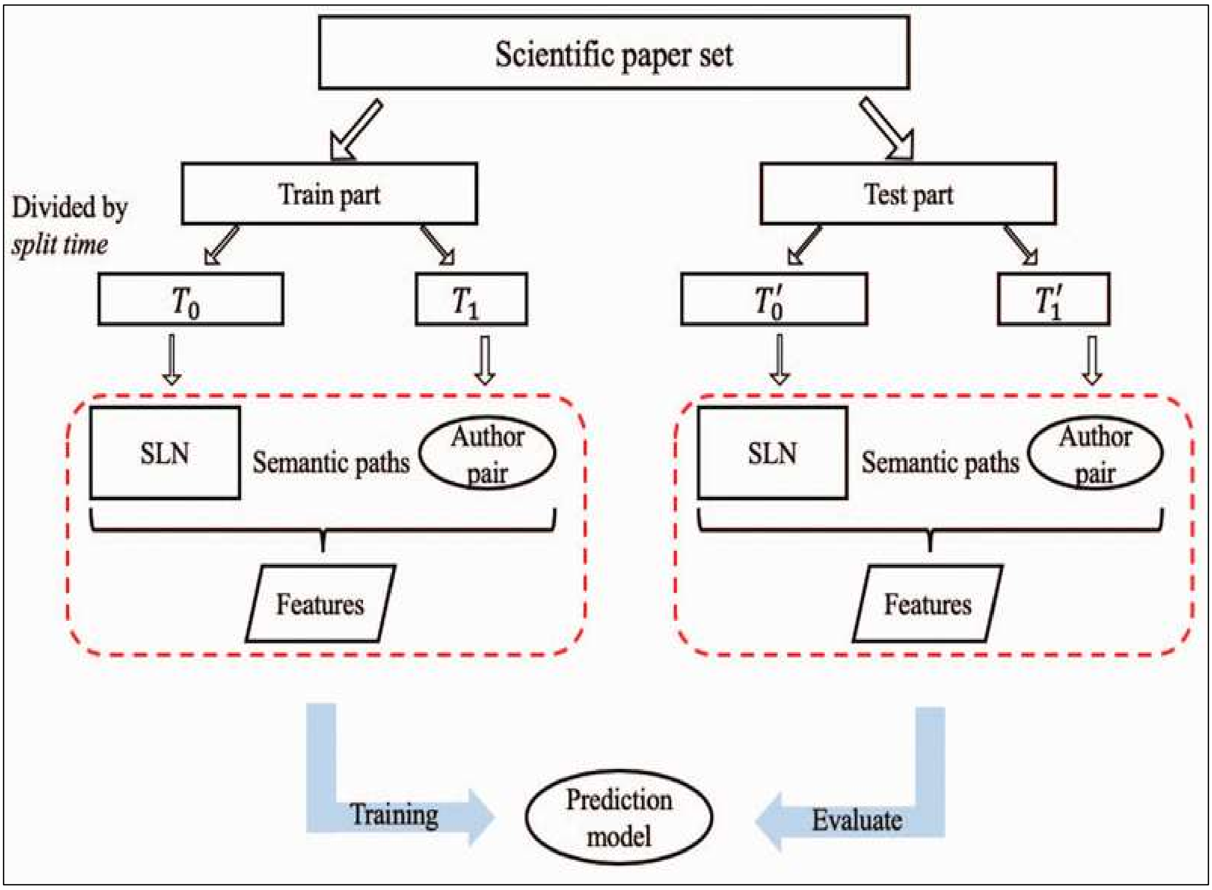


Figure 4-3 Framework of the Model.

4.5 The Framework of Recommending Future Collaborators

Let Paper = {$Paper_0$,$Paper_1$, $Paper_2$,…, $Paper_H$} be as the list of scientific papers, which is sorted by their publication dates. It is divided into two parts: the training part and the test part. We set the even -indexed papers as the training part, denoted as $Paper_{train}$={$Paper_{2*i}$|$Paper_{2*i}$∈Paper, *i* =0,1,2…} and the odd-indexed papers as the test part, denoted as $Paper_{test}$={$Paper_{2*i+1}$|$Paper_{2*i+1}$∈Paper, *i* =0,1,2…}, as shown in Figure 4-2.

We set up the SLN through the paper set T_0. Then, we select the set of author pairs <A_i,A_j> from T_1 as the training data, and the selected author pairs meet the following conditions: 1) Ai has written at least one paper in T_0;2)>A_j has written at least one paper in T_0. For each author pair < A_i,A_j >, we construct a feature vector x={x_1,x_2, …,x_n} by extracting features from established the SLN with semantic paths, where xi is a value that measures the correlation of At and A_j under a semantic path. If Ai and Aj have co-author relationship in T_1, the label for <Ai,Aj> is 1, otherwise 0. In this paper, we choose the classic method, namely, the Gaussian naive Bayes model, as the prediction model. Naive Bayes classifier, simply called naive Bayes, is an efficient classifier that is one of the top ten algorithms in data mining [144]. Naive Bayes is a useful classifier that is used widely in many applications such as data stream classification, document judgment, and text categorization [145]. Finally, we use the test part to evaluate the prediction model. The framework of the model is shown in Figure 4.3.

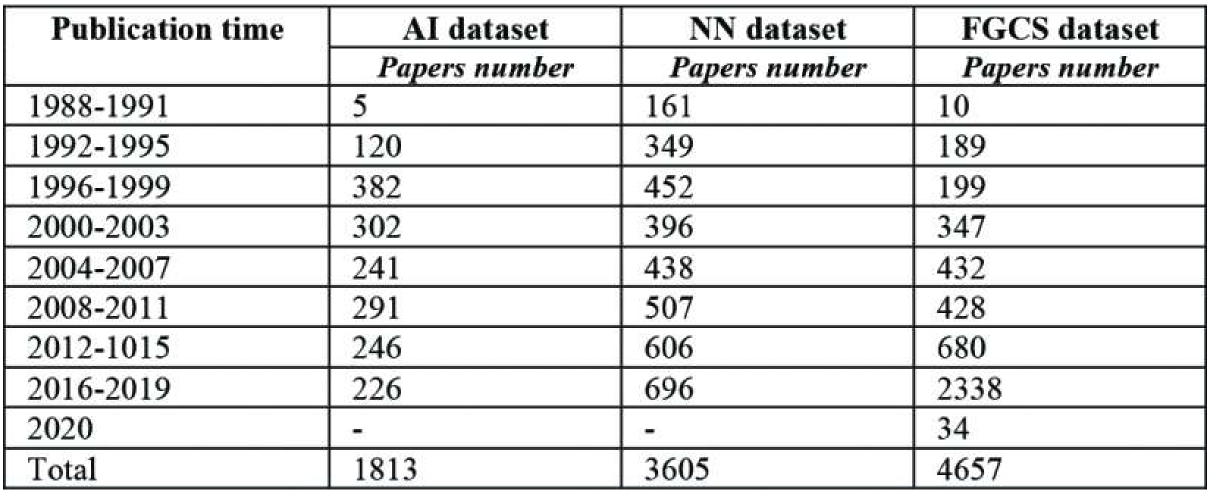
4.6 Experiment – Dataset

The AI dataset is a collection of *Artificial Intelligence* journal, with a total of 1,813 papers. The NN dataset is a collection of *Neural Networks* journal, with a total of 3,605 papers. The FGCS dataset is a collection of *Future Generation Computer Systems* journal, with a total of 4,657 papers. Each paper contains the author, publication time, and title. Most of these papers contain keywords and reference. A small number of papers contain the abstract and the text. The details of publication time are shown in Table 4.2.

4.7 Experiment Setting

We set 2012 as the *split time* t_1 to split the training part and test part. 74.0% of papers in the AI dataset, 63.9% of papers in the NN dataset, and 34.5% of papers in the FGCS dataset are used to establish the SLN for extracting semantic paths features. T_0 is the set of papers whose publication time is in the time interval [1988,2012) in training part, and T_1 is the set of papers whose publication time is in the future time interval [2012, 2020] in the training part. So do T'_0 and T'_1 in the test part.

For each training author pair < A_i, A_j >, Both A_i and A_j have published papers in the past time interval [1988,2012) and the future time interval [2012, 2020]. And the feature vector for them is formulated as $x=\{x_1, x_2, x_3, x_4, x_5\}$, where x_i represents the path count of the semantic path *Path*$_i$ that measures the relevance of A_i and A_j.



Publication time	AI dataset	NN dataset	FGCS dataset
	Papers number	*Papers number*	*Papers number*
1988-1991	5	161	10
1992-1995	120	349	189
1996-1999	382	452	199
2000-2003	302	396	347
2004-2007	241	438	432
2008-2011	291	507	428
2012-1015	246	606	680
2016-2019	226	696	2338
2020	-	-	34
Total	1813	3605	4657

Table 4.2 Details of three datasets.

Besides, we also set up several comparison models as the baselines: (1) model1: the feature vector is denoted as x={X*1*} based on *path1* with only keyword information. (2) *model2*: the feature vector is denoted as x={x2,x*3*} based on *path2* and *path3* with only citation relationship information. (3) *model3*: the feature vector is denoted as x={x*4*,x*5*} based on path4 and path5 with only co-author relationship information. (4) model4: the feature vector is denoted as x={x*1*,x2,x*3*} based on path1,path2 and path3 without co-author relationship information. (5) *model5*: the feature vector is denoted as x={x*1*,x2,x*3*} based on *path1, path4* and *path5* without using citation relationship information. (6) *model6*: the feature vector is denoted as x={x2,x3,x4,x5} based on *path2*, *path3*, *path4* and *path5* without using keyword information. Our model is *model7*, based on the combination of all semantic paths.

The precision, recall, and F1 are used to evaluate models.

$$precision = \frac{|TP|}{|TP| + |FP|}$$

$$recall = \frac{|TP|}{|TP| + |FN|}$$

$$F1 - -score = \frac{2 * precision * recall}{precision + recall}$$

|*TP*|*,* |*FP*| and |*FN*| are corresponding to the number of positive vectors that are predicted precisely (True Positives), the number of positive vectors that are predicted imprecisely (False Positives) and the number of negative vectors that are predicted imprecisely (False Negatives).

4.8 Results and Analysis

The results of experiment is shown in Table 4.3. Table 4.4 shows the details of the established the SLNs in the training part and test part.

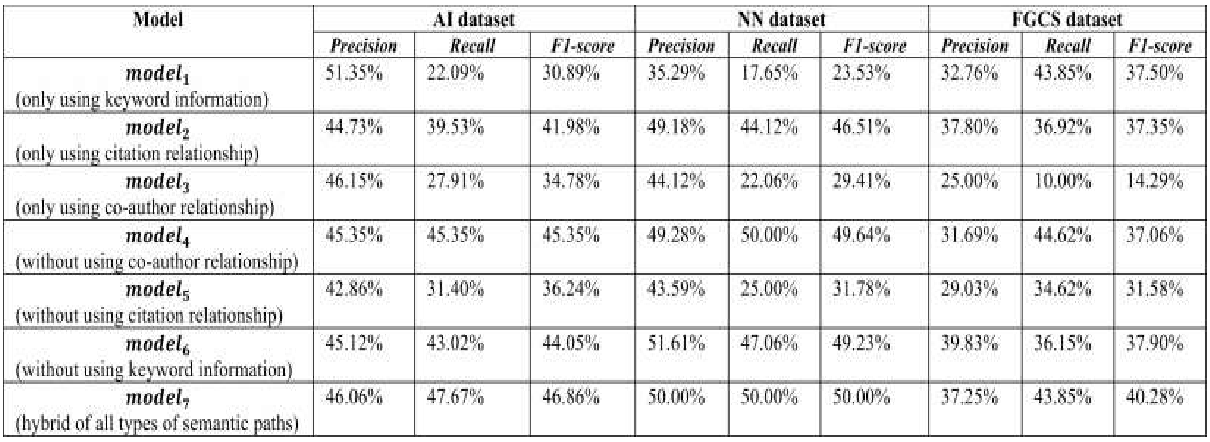
It can be seen that the five semantic paths proposed in this paper achieved a promising score. Thus, these five semantic paths can be used to recommend collaborators.

In Table 4.3, the results of *model7, model5*, and *model2* show that using citation relationship information are effective in finding collaborators. Citing the same authors or papers can help narrow the scope of finding a collaborator. Hence, the cited authors and the cited papers can potentially and comprehensively reflect the researchers' directions. The more similar two researchers' directions, the more likely they are to be collaborators.

From the results of *model*$_7$, *model*$_6$, and model$_1$, we can find that keyword information has a positive effect on recommending collaborators.

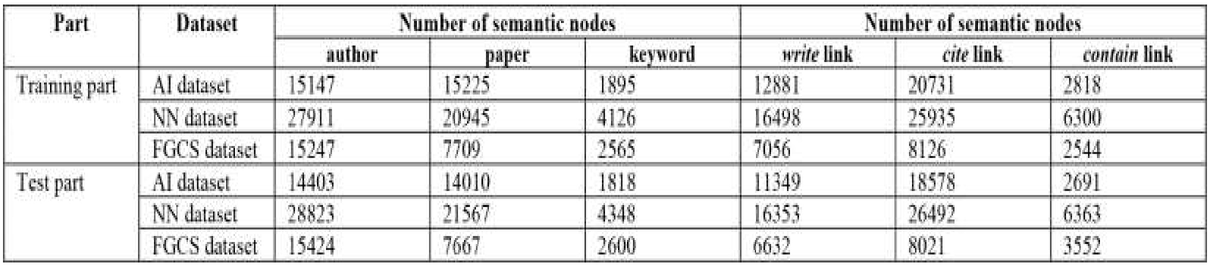
From the results of *model*$_7$, *model*$_4$, and *model*$_3$, we can find that the co-author information has a positive effect on recommending collaborators. But when finding future collaborators, combining both co-author information and research direction information (keyword information and citation relationship information) will obtain a better performance because future collaboration can be driven more by research direction and, expertise than by neighbourhood.

From the results of the AI dataset and NN dataset, we can find that the model with keyword information or citation relationship information can always get a higher precision than the model without keyword information and citation relationship information. The model with co-author information can always get a higher recall than the model without co-author information. From the results of the FGCS dataset, the model with keyword information or citation relationship information can also get high precision, but the model with only co-author information get the lowest recall. This may be because that, on one hand, although the historical cooperation between the authors of the FGCS dataset is closer than that of the AI and NN dataset, the FGCS journal is a multidisciplinary journal, the papers in the FGCS dataset cover a wide range and many types, on the other hand, only 34.5% of the papers published before 2012, there are not enough papers to reflect the features which can impact on future collaborators.



Model	AI dataset			NN dataset			FGCS dataset		
	Precision	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*
$\boldsymbol{model_1}$ (only using keyword information)	51.35%	22.09%	30.89%	35.29%	17.65%	23.53%	32.76%	43.85%	37.50%
$\boldsymbol{model_2}$ (only using citation relationship)	44.73%	39.53%	41.98%	49.18%	44.12%	46.51%	37.80%	36.92%	37.35%
$\boldsymbol{model_3}$ (only using co-author relationship)	46.15%	27.91%	34.78%	44.12%	22.06%	29.41%	25.00%	10.00%	14.29%
$\boldsymbol{model_4}$ (without using co-author relationship)	45.35%	45.35%	45.35%	49.28%	50.00%	49.64%	31.69%	44.62%	37.06%
$\boldsymbol{model_5}$ (without using citation relationship)	42.86%	31.40%	36.24%	43.59%	25.00%	31.78%	29.03%	34.62%	31.58%
$\boldsymbol{model_6}$ (without using keyword information)	45.12%	43.02%	44.05%	51.61%	47.06%	49.23%	39.83%	36.15%	37.90%
$\boldsymbol{model_7}$ (hybrid of all types of semantic paths)	46.06%	47.67%	46.86%	50.00%	50.00%	50.00%	37.25%	43.85%	40.28%

Table 4.3 Results of experiment.



Part	Dataset	Number of semantic nodes			Number of semantic nodes		
		author	**paper**	**keyword**	***write* link**	***cite* link**	***contain* link**
Training part	AI dataset	15147	15225	1895	12881	20731	2818
	NN dataset	27911	20945	4126	16498	25935	6300
	FGCS dataset	15247	7709	2565	7056	8126	2544
Test part	AI dataset	14403	14010	1818	11349	18578	2691
	NN dataset	28823	21567	4348	16353	26492	6363
	FGCS dataset	15424	7667	2600	6632	8021	3552

Table 4.4 Details of SLNS.

5 CHAPTER FIVE: A MULTI-DIMENSIONAL METHOD SPACE FOR MANAGING SOFTWARE PROCESSES

The main aim of Software Engineering is to develop a software system, which fulfils the user requirements within time and budget constraints. This research uses the multi-dimensional Resource Space Model to manage multiple types of software engineering processes and maps their features into multiple dimensions for supporting analysis, development and maintenance of software system. Two case studies show that the Resource Space Model is feasible for managing the software processes and data. RSM is utilized to build the solution for the email space model in [135].

5.1 Introduction

The term of software engineering first introduced in 1968 at a conference which was held to discuss the issue related to software crisis (Naur and Randell 1968) after that in the 1970s and 1980s new software methodology were introduced [18].

Software engineering is the engineering discipline which is generally concerned with all features of a software development process from initial stages to the final maintenance stage. The systematic approach that is used in software engineering is also called *software process*, which is the sequence of the activities.

This chapter studies two cases of applying the Resource Space Model to manage the software engineering projects and show that RSM can map the software engineering activities into one space for managing software processes easily. New dimensions and coordinates can be easily adopted for managing new features found during software process. It offers the new way to manage the whole lifecycle of projects for development and design the application solution of the problems.

5.2 RSM use to Manage Multiple Types of Software Processes from Multiple Dimensions

Software process models have different features which can be differentiated and categorized into different dimensions. The main features are shown in the Table 5.1 [24][25][27]-[30]. This features categorization well fits into RSM as a useful way for managing software processes.

Process Models Features	Waterfall Model	Incremental Model	Scrum Model	Extreme Programming (XP)
Form of Requirement	Complete Documentation	Incremental Documentation	Verbal conversation	Frequent Verbal conversation
User feedback	In the end	In the process	In the process	In the process
Adaptive to change	Low	Medium	High	Highest
Predictability of final results	Low	Medium	High	Highest
Stage of risk identification	At beginning	At different stages	In the short process	In the shortest process
Successful Rate	Low	Medium	High	Highest
Customer Satisfaction	Low	Medium	High	Highest
Variation	Yes (V model)	No	No	Yes (Industrial IXP)
Understandability	Easy	Medium	Hardest	Hard
Knowledge	High	Medium	Low	Lowest
Usability	Basic	Medium	High	Very High
Reliability	High	High	Medium	Low
Cost	Low	Medium	High	Very High
People	Software engineers	Software Engineers	Customers, stakeholders, developers, team members, and end-users	Customers, stakeholders, developers, team members, and end-users
Elasticity	Very Low	Low	High	Very High

Table 5.1 Dimensions of Software Engineering Processes.

- *Form of requirements.* In the Plan-Driven approach, requirements are formal and clear however in Agile approach requirements are not in documentation form. This is an important dimension which needs to accurately consider while choosing the most effective model, for development the project.
- *User Feedback.* In some software processes user feedback is an important factor but in Plan-driven it is impossible until the final project delivery done. Agile normally has a meeting with clients, sometimes on a daily basis so its very easy to get the user feedback.

- *Adaptive to change.* Move from Plan-driven approach to Agile approach this feature is increased from the high to the highest level. Agile can handle changes more efficiently then Plan-Driven approach.
- *Predictability of final results*. In Plan-driven approach we cannot predict the final results until the final product is ready to use instead in an Agile approach it could be known in initial functionality.
- *Stage of risk identification*. In Plan-driven approach it’s identified in early stages as detail documentation needed in requirement phase, however, in Agile it can be identified during the process.
- *Successful Rate*. Waterfall model has a low rate of success if deploy in a project, where normally change occurs. Agile approaches more popular in recent decades and has more success rate to handle change.
- *Customer Satisfaction.* The Agile has high customer’s satisfaction as the customer directly involved in the project development process. Plan-Driven has lowered customer satisfaction as compared to the agile software engineering approach.
- *Variation.* In both approaches have some variation; Extreme Programming model has a variation of Industrial Extreme Programming (IXP) and Waterfall model has a variation of V model.
- *Understandability.* Customer point of view, it’s crucial to know, what is happening with software products i.e. when it will finish and handover to them. In the Agile approach, customers are directly involved and well aware of the non-functional requirements.
- *Knowledge.* Some models are required pre-knowledge, pre-qualification and training which are based on formal methods.
- *Usability.* Plan-driven approach has low usability as compared to Agile Approach which used Object Oriented approach with high reusability.
- *Reliability.* Plan-Driven models are more reliable as mostly used for critical safety systems which directly involve humans and follow the formal methods. On the other hand, Agile is getting mature to meet these standards.
- *Cost.* The Plan-driven approach is between low to medium scale as pre standards followed. It could be developed using to outsource techniques which help to reduce the implementation cost and most of the time finish within the estimated budget that carefully planned in early stages. Agile approach some time increase cost as new functionality added.

- *People.* People have very low-level involvement in Plan-driven approach as they follow the predefined steps. Agile development approach team has more freedom to make a decision to find better solutions.
- *Elasticity.* Plan-Driven approach just follows the planned steps. Waterfall model followed the linear model approach, but Agile accept the change even in the last stages.

There are various dimensions of the software engineering process which are shown in Table 5.1. According to software engineering features, it can be easy to choose the software method depending on the nature of the project. RSM supports generalization and specialization on multidimensional classifications [6]. It manages these features into multiple dimensions which are easy for software engineering teams to choose the effective model. The selected model fulfils the stakeholder's requirements for developing the required system. It classifies the different properties of the software methods, and those provide a roadmap to achieve the goal and select the right process to build quality software systems.

RSM helps to choose the different models features which belong to different approaches, to build the same project, it's helpful to find the balance for selecting more than one approaches. For instance, we could select the requirement dimension from the waterfall model if documentation is required while using the agile approach features as shown in Table 5.1.

5.3 Property Inventory Tracker

This section studies how to use the RSM to manage Property Inventory Tracker. Dimensions and coordinates are written in *italic* which is used in both case studies, shown in Figure 5-1 and Figure 5-2. In the given case study of the Property Inventory Tracker (PIT), the requirements have been gathered from the PIT team with several group meetings held to understand the business problem. The main business of this company is to, for each client, assess the client's property and create a detail report of its current condition. The report highlights the major damages in the property so that the landlord can repair it and, if needed, charge the tenant(s) for the damage. Most of the company's work is being done manually i.e. access the property, manually take pictures of the damages and record videos of the property's current condition. PIT has some booking issues with their system including video recording and uploading. The company wants to sort out these issues while updating their current system.

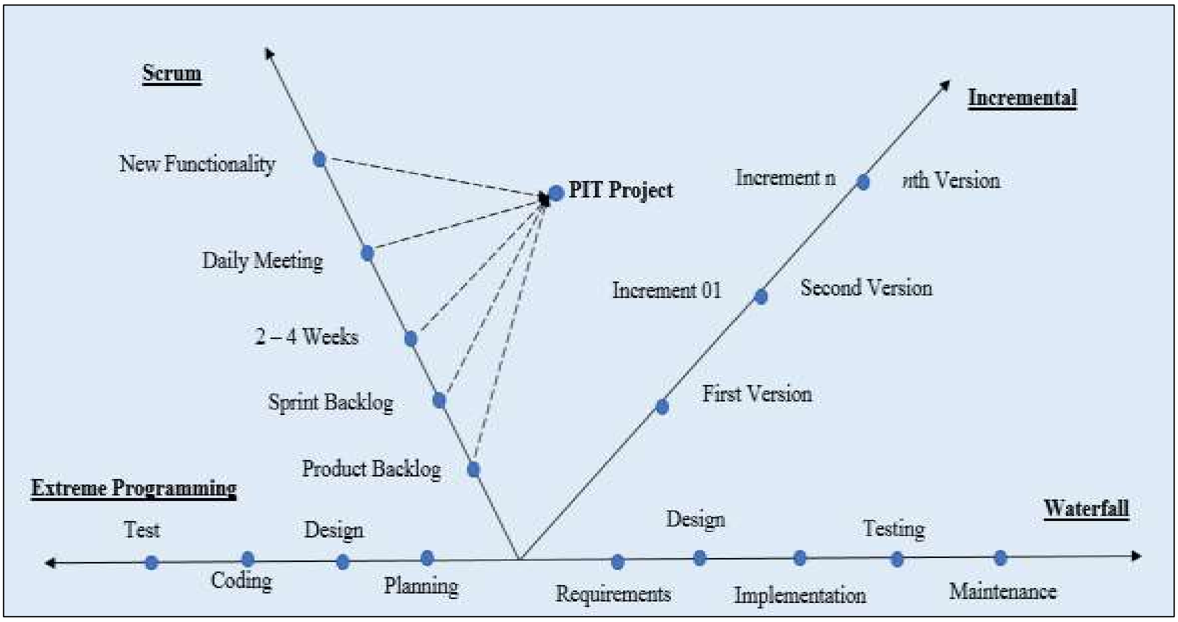
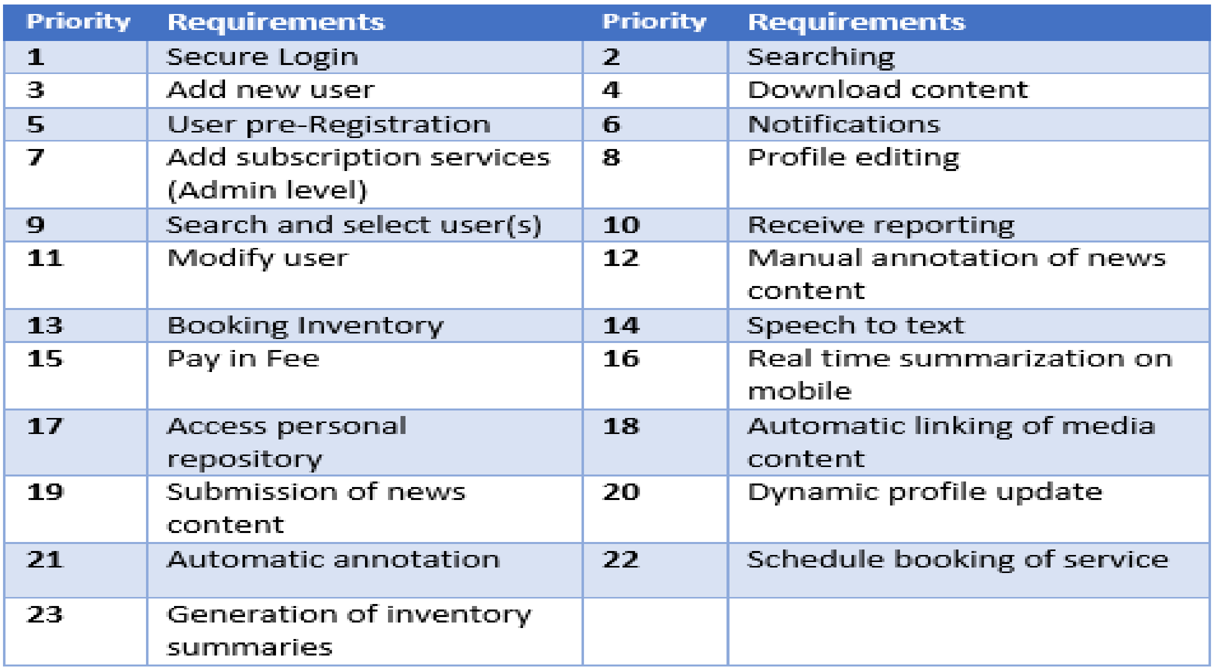


Figure 5-1 Software Engineering Processes of the Property Inventory Tracker in RSM.

The main aim is to develop the commercial software system which overcomes the limitations of the existing system. The RSM helps to develop the PIT system by mapping the real data collected in the form of requirements and discussions into a one-dimensional space. To accomplish this, the requirements have been first prioritized as shown in Table 5.2. According to the stakeholder's requirements the most needed functionalities, a booking system.

As an example, four software engineering processes, *Extreme Programming, Scrum, Incremental* and *Waterfall*, are shown in a multi-dimensional space in Figure 5-1. Each process constitutes one dimension of the space, and the coordinates are the activities of the existing model. For instance, the *Extreme Programming* dimension shows the activities including *Planning, Design, Coding*, and *Test*. Other software processes are shown in Figure 5.1 in the same way. The above example strongly justifies that RSM supports to manage various software engineering models including those shown in Figure 2.1 of chapter 2.

The RSM is used to map the PIT project as shown in Figure 5.1. The *Scrum* dimension shows its activities as the coordinates. The coordinate *Product Backlog* is the activity in which we gather customer requirements and understand the overall system scope. As a result of this stage, Table 5.2 shows all desired functionalities which are important from the user's point of view. At this stage, all the gathered requirements are also prioritized as shown in Table 5.2. This phase has user's stories which are used to understand the narratives and perspective of the system requirements. The ultimate product concept is made clear at this stage as Table 5.2 lists all the functionalities which need to be done in the project.



Priority	Requirements	Priority	Requirements
1	Secure Login	2	Searching
3	Add new user	4	Download content
5	User pre-Registration	6	Notifications
7	Add subscription services (Admin level)	8	Profile editing
9	Search and select user(s)	10	Receive reporting
11	Modify user	12	Manual annotation of news content
13	Booking Inventory	14	Speech to text
15	Pay in Fee	16	Real time summarization on mobile
17	Access personal repository	18	Automatic linking of media content
19	Submission of news content	20	Dynamic profile update
21	Automatic annotation	22	Schedule booking of service
23	Generation of inventory summaries		

Table 5.2 PIT Requirements.

In the PIT project, stakeholders want the working software with the most important functionality. PIT project properties define that an Agile approach such as Scrum is more suitable for implementing this project based on the most needed functionality, i.e. the booking system required in the first phase and followed by the rest of requirements. Another justification for Scrum is that this project does not require formal documentation that is required by the plan-driven processes. Instead, for the purpose of fast implementation, we only shortlist the most important requirements as shown in Table 5.2, so we prefer an Agile software engineering approach such as Scrum over the rigid Plan-driven approach.

The next coordinate is *Sprint Backlog*, where we will take 2-4 weeks for implementing each specific functionality, such as secure login, searching, add a new user, and so on, as prioritized in Table 5.2. The development and PIT teams hold regular meetings on a daily basis, which could even be as short as 15 minutes, so that when user requirements change they immediately update the list in the first and second coordinates of the *Scrum* dimension. In this way, all the given requirements shown in Table 5.2 are implemented in an iterative fashion (See Figure 2.5 too in Chapter 2) and users are able to give their feedback at any time, even after using the first delivered functionality of the software system.

In the last phase (*New Functionality*), the implemented functionality is delivered to the client. The client is then able to use the delivered system and give feedback to improve the existing system.

The case study demonstrates that RSM helps to choose the most effective software process which has the ability to deliver the right product to clients, and how RSM makes software engineer's job easy by following the steps in one resource space.

5.4 Eclipse Project

Eclipse is an open-source development environment (IDE) which provides a set of commercial-quality tools and an industrial environment to develop highly integrated software, debug software program, write code, and share with the team. Eclipse is freely available to be adopted by third parties to design plug-ins. The Eclipse software development kit (SDK) is used as a tool for building web services, support Java programming and embedded system programming by Java Development Tools (JDT).

The RSM is used for managing the Eclipse project [33] in one space using the Waterfall model [18]. Waterfall model is effective to use for managing this project based on these properties shown in Table 5.1, such as a large distributed project, documentation required and reliability etc. Waterfall model is mapped into five dimensions from *requirements* to *maintenance* as shown in Figure 5.2. Each dimension has coordinates which represent the methods of that specific dimension. As shown in Figure 5.2, the *Implementation* dimension has two coordinates *Programming* and *Coding* Standards. The RSM shows all the steps to complete the project from user *requirements* to the *maintenance* phase; all steps are shown in Figure 5.2 and explained in the following sections.

There is no single existing requirement documentation of the Eclipse project. Many software engineers and developers have contributed to this project so there is a need that its requirements selectively quoted in this case study are collected and recorded in a standardized way. In fact, the Eclipse project uses Software Requirements Specification (SRS), which is developed by the committee of the IEEE, a group of very experienced software engineers [33]. More specifically, the SRS IEEE 830-1998 standard is used for the documentation process of this project.

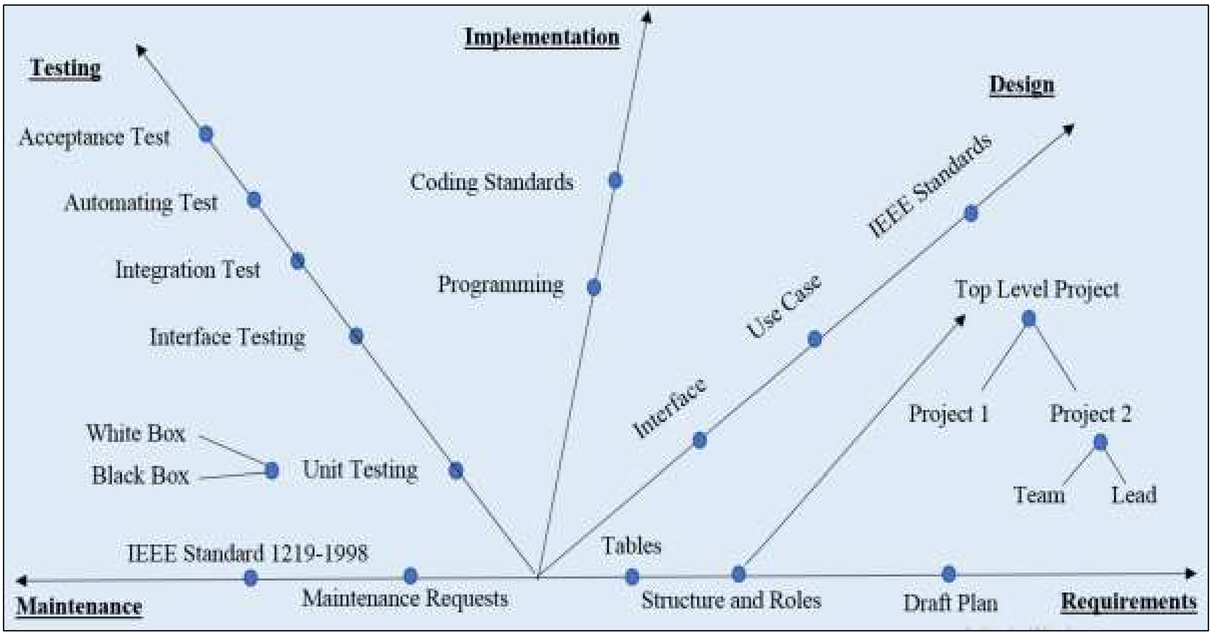


Figure 5-2 Eclipse Project in Multi-Dimensional Resource Space.

The requirements are the first and core step to build the Eclipse project. The RSM manages all the requirements in the *requirement* dimension which contain its own coordinates. The coordinates of the *requirement* dimension describe the project structure and roles of the team members at the initial stage. The Eclipse project is a *Top-Level Project* which is further divided into subprojects and a Project Management Committee (PMC) is responsible for each of these subprojects. Each subproject has its own project *team* and project *lead*. The project team consists of a number of developers and committers as well. PMC's role is to make sure that project is operating effectively and all project plans, documents and reports are available to all team members and publicly available too as transparency and open participation are needed by its open-source nature. PMC is the overall controlling body of the Eclipse project.

The next coordinate is *Draft Plan* defined by Eclipse-PMC to specify the important release milestones, deliverables, java developments tools (JDT), plug-in development environments (PDE) and application programming interface (API) [33]. Software verification and validation are also a part of *Draft Plan* and use the IEEE 1012-2004 framework to meet customers' needs and wants.

In the *Design* dimension, the software *interface* is designed, including the windows, toolbars, toolbox, menu bar, header bar, form designer area, view code, project explorer, console, editor, shortcut bar and navigation panel, etc. *Use cases* and classes are considered and designed in this phase. There is a package of core classes called "Platform.Core.Runtime" used in this project. The IEEE Software Design Document (SDD) standard 1016-1998 has the guidelines for design documentation and this project uses these standards with only slight modifications.

The *Implementation* dimension has two coordinates which are based on the findings of the Eclipse project. It contains a large body of code and artifacts. This project follows the *Coding Standards* and *Programming* conventions for implementation. Naming conventions for this project are used in Java packages, workspace projects, plug-ins, methods and variables. In this project the following naming conventions are used: methods should be used as verbs and written in lowercase; the first letter of each internal word will be capitalized i.e. "getValue()", "setBackground()", and "get()".

Testing is an important dimension which normally starts after the implementation phase. There are two *Unit Testing* techniques used in this project; *White Box* testing and *Black Box* testing. White box testing is the technique where the internal structure of the system is tested including the coding, functions, methods and classes. Black box testing method is used to test at the application level where test cases are built to test whether the system functions correctly. Unit testing is applied to determine whether Eclipse project works in line with to anticipations. It is preferred that unit testing is done by the same developers of the Eclipse project because they are more familiar with the internal structure of the system. *Interface Testing* is done from the user point of view where each module is tested according to user requirements. *Integration Test* is the testing technique which is used during the assembling of the modules in this project.

In the Eclipse project, testing is done from class level to the interface level before the project is finalized. Eclipse is a large project so not only exiting testing techniques are applied but *automating test* tools are also used, such as TPTP (Eclipse Test and Performance Tools Platform), which is adapted for testing, monitoring, and tracing. In addition, TPTP is also used to test Web applications which are normally built using Eclipse. Finally, the Eclipse Project needs to pass the *Acceptance Test*. This is the last test and at the same time an important test which checks whether the user's requirements are fulfilled or not.

Finally, the last dimension is mapping the *Maintenance* process of the project. Software system needs continuous maintenance to keep up-to-date and in working order. If we need to enhance or repair the software system, called *Maintenance Request* in IEEE terminology, we need the IEEE standard 1219-1998 which describes the process for managing and executing software maintenance activities that exist in this project.

To conclude, the above demonstrates that RSM helps for the non-functional aspects of the project progress, i.e. the current status of the project, how long it would take to finish the entire project and up to what point it is completed. It is noteworthy that the Eclipse project requires formal documentation and standards. This is why a plan-driven model, say the Waterfall model (See Figure 2.2 too in chapter 2), is preferred and utilized to manage this project in multidimensional resource space.

6 CHAPTER SIX: VISUALIZATION OF MULTI-DIMENSIONAL RESOURCE SPACE MODEL

6.1 Introduction

In this digital age, businesses are constantly generating data and use big data to support smart businesses [85][86]. It is very important for information systems to provide a solution for enterprises to manage big data in a uniform way from the high-level user interface to the underlying model for managing data.

Traditional data models like the relational data model [85] and graph data model data space can be regarded as graph data models [88]. These two models are limited to supporting the high-level interface and the underlying data structure.

The Resource Space Model (RSM) provides a systematic theory, model and method for managing various resources with multi-dimensional category space at both the interface level and the data management level [1-17].

The faceted navigation provides a new way to browse website but it lacks the supporting theory and model [88]. RSM can be adopted as the underlying model for realizing faceted navigation. This section introduces the implementation of a multi-dimensional interface by adopting the Resource Space Model and demonstrates the advantages.

Figure 6-1 is a 3-dimensional resource space Spec-Apart-Gend (Specialty, Apartment, and Gender) specifying student information of a college. Three axes are Specialty = {math, computer, physics}, Apartment = {1#, 2#, 3#} and Gender = {male, female}. Each point indicates a class of students, for example, the point (math, 1#, male) represents all the male students belonging to the department of mathematics and living in apartment no.1 of this college. We could choose a point by clicking on the “male” coordinate from the gender dimension and the “math” coordinate from the specialty dimension and it would display the results form according to that prospects.

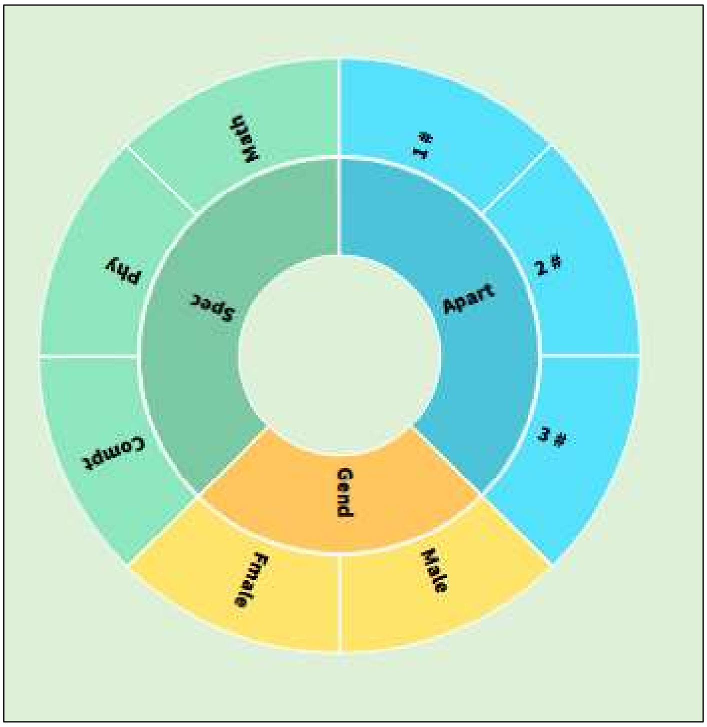


Figure 6-1 An example of 3-Dimensional Resource Space.

Coordinates directly residing at axis are called top-level coordinates. For instance in Figure 6-1 “male” and “female” are the coordinates of the gender demission. Each top-level coordinate can be refined top-down to a coordinate hierarchy representing classifications at different levels and different granularities. Each node in the hierarchy can be named by the path from the root. For example, the top-level coordinate computer at axis Specialty shown in Figure 6-1 has a coordinate hierarchy Speciality (Math, Computer, Physics). The primitive of Resource Space Model is {resource space, resource, axis, coordinate}, where each element is based on two basic mathematical concepts: set and partition [4].

Visualization of multi-dimensional resource space is based on the formal theory of the Resource Space Model. The RSM property of multi-dimensions is inherited in this visualization and implemented while using the programming languages in Figure 6.2. The good colours scheme could enhance the presentation of a software interface [86], which not only makes it more attractive and user-friendly but also classifies user interests. Classification of user interests enable users to quickly locate the interested points in the space. This also makes a consistency between interface and model.

Faceted navigating or faceted search or faceted browsing is an approach to accessing the information that is classified or organized according to different categories and sub-categories by filtering results. Facets are used for categorizing the information into different forms. Different items or objects have their modules according to their similarity and groups.

The faceted navigation although it's provide the users information in the form of categories and presented in organized form consider the basics search methods and operations. Its shows information in tags method which is in the most websites even given example interface of Figure 6.6 (a) and Figure 6.6 (b) are in the form of tags and hyperlinks in some cases dropdown lists shown which users have to go bit further deep to select their required information. In the other hand the proposed interface of visualization of multi-dimensional resource space is based on the graphical interface. Graphic design of interface direct to users in a way how to do and what to do to find the information [88]. Its interface is based on graphics so its size and position and combinations of colour should make clear and easy searchable choices for the users.

6.2 Architecture

The architecture of the visualization interface is shown in Figure 6-2 where end users can interact with software applications to perform the operations while entering their inputs using the interface according to his/her requirements. The application interface panel illustrates a number of applications to indicate that the implemented interface could be used for more than one application. For instance, Figure 6-2 shows the implementation for property letting application other applications could be public gallery or online e-commerce web sites. In the programming panel a number of programming languages such as HTML, JavaScript, CSS, JQuery and PHP are shown to implement visualization of multi-dimension interface. Both front end and server side programming languages are used for development. These programming languages interact with the data source to get data and then display according to the user's requirements. The end-user can access the required information while connecting through a user-friendly interface. In Figure 6-2 programming languages are platform-independent. They are used to build the interfaces of software that are supported by most Internet explorers without installing other software or plug-ins. These applications are connected to the database to get the data which are processed and displayed using the multi-dimension interface.

The proposed architecture will overcome the major issues of complexity to access the required information without any tussle. It is flexible and allows facilitating more and more business for providing services online. It helps in enhancing overall performance in the business.

At this stage, as shown in architecture Figure 6-2 makes a clear difference to other implementation of software interfaces where this proposed interface also achieves the deficiencies of the formal methods and theories.

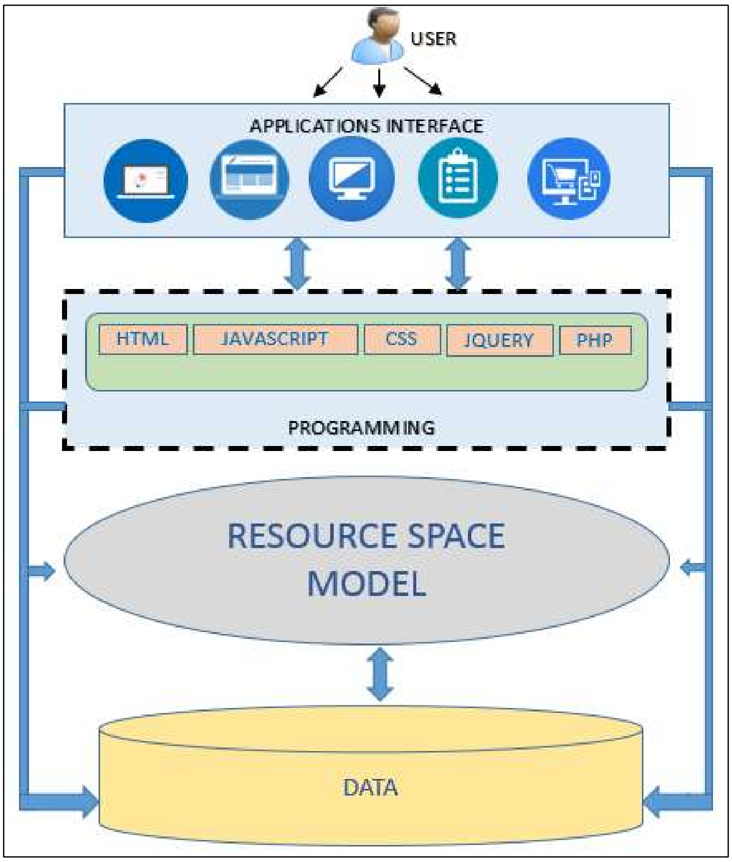


Figure 6-2 Architecture for visualization

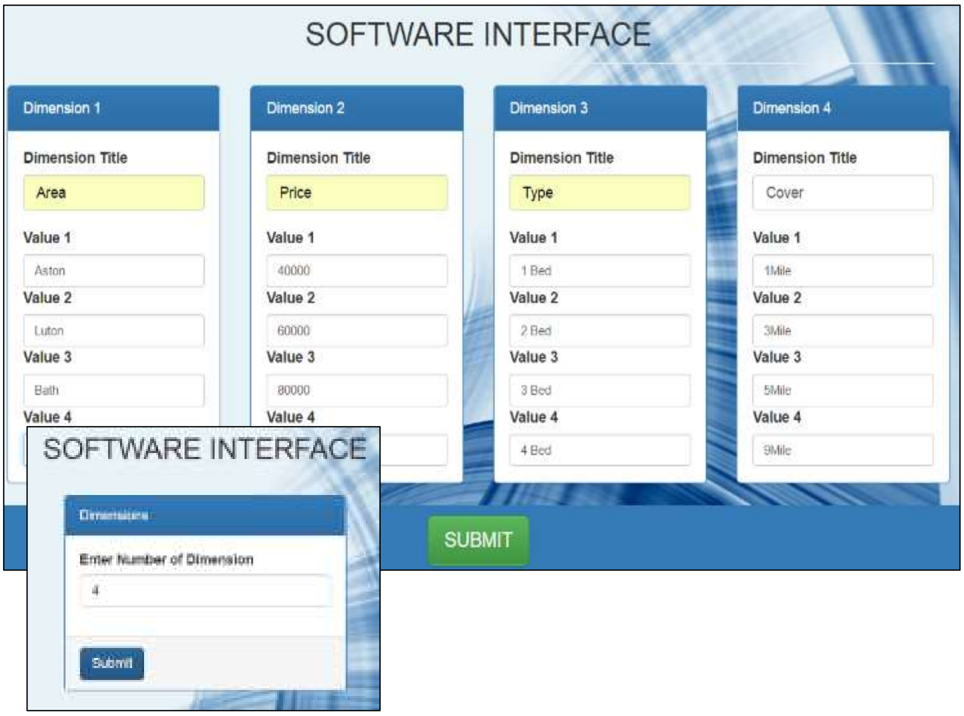
6.3 Technology

A rich choice of options is available nowadays to design interface and for the software development process. In general, most of the programming languages can execute on many platforms including our presented visualization interface. These are the programming languages that are used to build this interface, Hypertext Markup Language (HTML), Cascading Style Sheet (CSS), JavaScript, JQuery and Data-Driven Documents (D3). HTML and CSS are two of the major and core technologies for building interfaces, and web context pages. HTML provides the structure of the web pages for a range of devices including the smart mobile, gadgets, iPads and laptops. CSS is used for Web pages to display information in the appropriate presentation format. CSS files can be used to define size, colour, font, spacing, location and border of the HTML. It could also be used to create a master page template to maintain a similar look through all the web pages.

JQuery is based on JavaScript Library. JavaScript is abbreviated as JS. It is a high-level language which can interpret and handle the challenges of dynamic application behaviour at the run time. JQuery is a fast, open-source programming language under the MIT (Massachusetts Institute of Technology) License. It is a powerful language used to handle events, create animations and also Ajax based application. D3 (Data-Driven Documents) is a JavaScript library used mainly for manipulating the context of Web-based data. D3 allows dynamic applications to become live using HTML, SVG (Scalable Vector Graphics), and CSS. D3 is important for Web standards which allow us to utilize the capabilities of modern browsers without working on another framework. It is a powerful visualization mechanism and data-driven approach. All these languages are used for frontend interface design. In other words they are used to implement user-friendly interface visualization. From the server point of view, the language interacting with the data is in PHP (Hypertext Pre-processor language). PHP is a server scripting language and is a powerful tool for making dynamic and interactive web pages. All the data that is generated for end-users by using this interface is managed by PHP for producing the required results.

6.4 Interface Implementation

In Figure 6.1 different colours are used to differentiate the dimensions of the space. The software can be used to input the number of dimensions as shown in Figure 6.3. This software has the capability to work up to four dimensions as shown in the given example. The number of dimensions can be increased easily to fulfil additional specifications. Further increased number of dimensions can be differentiated using different colours scheme. Figure 6-3 shows the interface can generate the output like shown in Figure 6-5 suiting to the user inputs. Figure 6-3 shows the title of each dimension along with their values. These results based on the inputs are shown in Figure 6-4 and Figure 6-5. The web sites are providing the option to search for a home, flat or an apartment in the United Kingdom. Using this interface we could use more and more features easily with just a few clicks. For example, if someone is searching for a house or a property in Leeds for a price range of 90000 pounds then the user can select according to his/her demands. In Figure 6-5, he/she is only using all dimensions of the interface which are *Area, Type, Cover* and Price. Google Maps is used to find destinations and only a few clicks are needed to set the destination with comparison, using a drop-down list needs more time to select the required option.



SOFTWARE INTERFACE

Dimension 1

Dimension Title: Area

Value 1: Aston

Value 2: Luton

Value 3: Bath

Value 4

Dimension 2

Dimension Title: Price

Value 1: 40000

Value 2: 60000

Value 3: 80000

Value 4

Dimension 3

Dimension Title: Type

Value 1: 1 Bed

Value 2: 2 Bed

Value 3: 3 Bed

Value 4: 4 Bed

Dimension 4

Dimension Title: Cover

Value 1: 1Mile

Value 2: 3Mile

Value 3: 5Mile

Value 4: 9Mile

SOFTWARE INTERFACE

Dimensions

Enter Number of Dimension: 4

Figure 6-3 User Input Form.

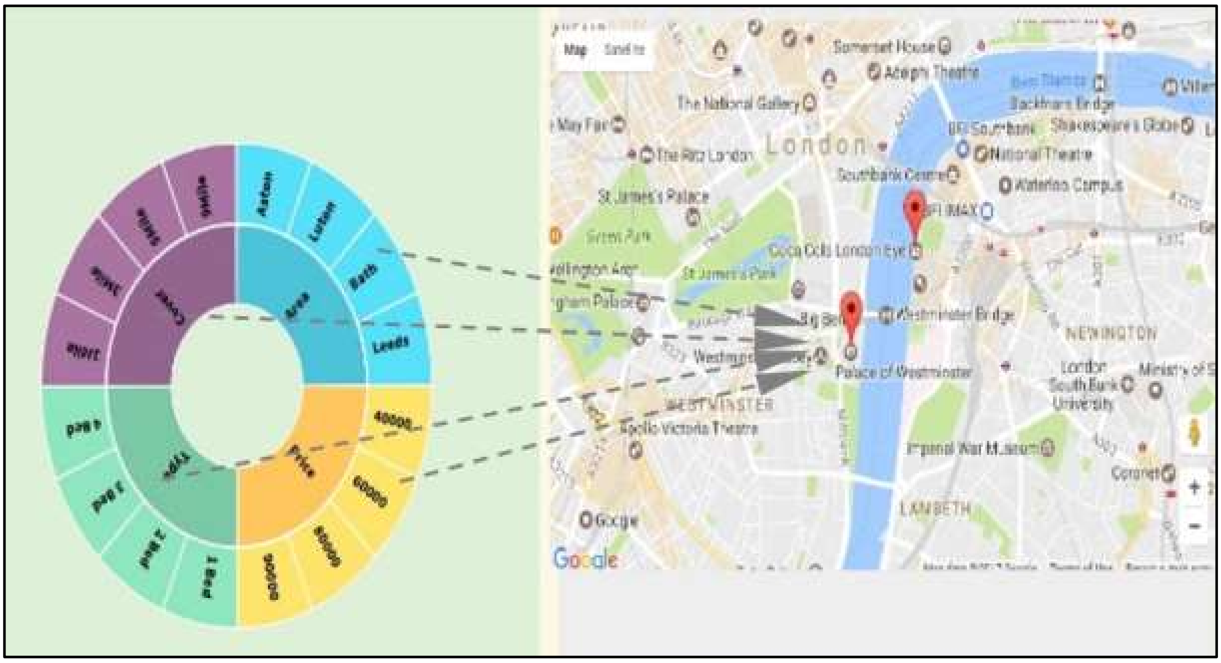


Figure 6-4 Circular view of inputs.

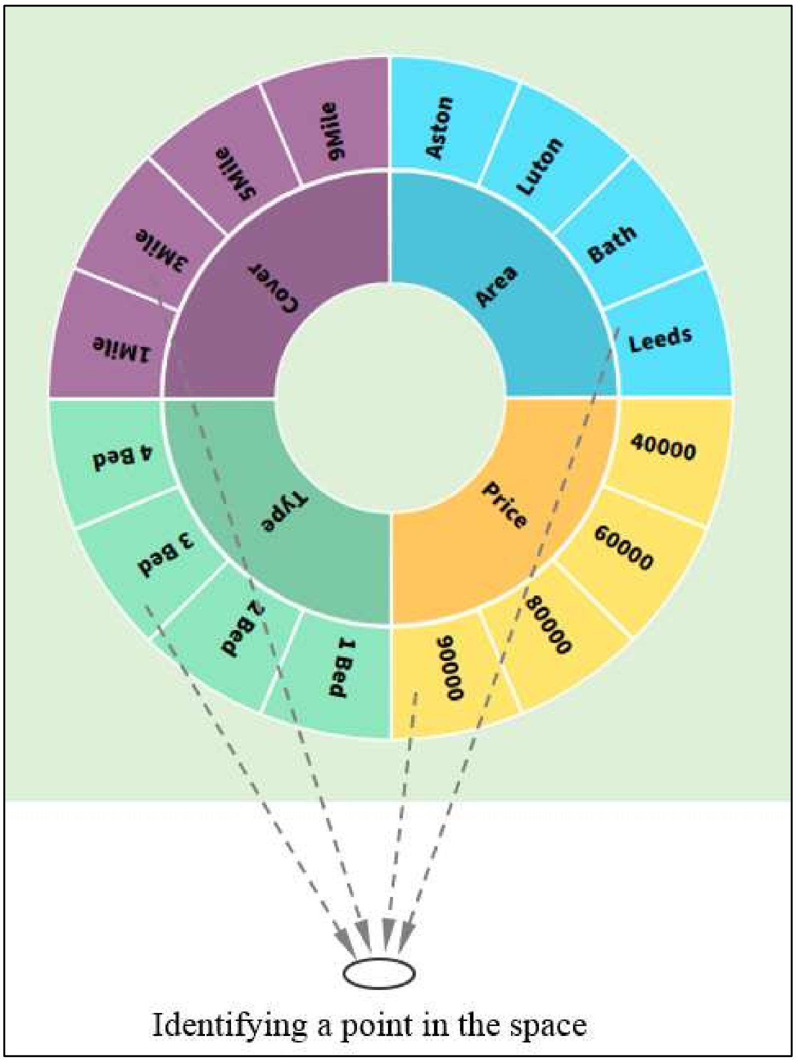


Figure 6-5 Dimensions with attributes for letting agencies.

The implementation shows that if interface is developed by using the multi-dimensions concept it provides the attractive visualization and it is also useful and easier to navigate. The graphical user interface is more helpful that provides easy options to the end-user for selection. They do not need to struggle and scroll down to find their options.

Keeping in view of given examples, it is difficult for the users to make a selection of each value from the drop-down list to make his/her required option. In conclusion, RSM plays a significant important role in web design to display the dimensions and coordinates to get the required results.

6.5 Test

This software interface is designed for general-purpose use and its dynamic nature enhances its usability, efficiency and user friendly. In the subsequent, real-world examples of property letting, Right Move and Aston University demonstrate as use cases study to compare the results. The system has been tested by the comparison of the following cases:

Case 1: Property letting software interface is used to select multiple options in Figure 6-5. For example, users can select any area from given choices of *Area* dimensions of Aston, Luton, Bath and Leeds. The results will be processed according to the selection like identifying a point in the section that is selected by clicking on Leeds. In Figure 6-5 one point is selected which will produce the results of a property based in Leeds area and price is 90000, the type is 3-bedrooms and exists within 3 miles radius from the city centre. One point is selected in the space according to these four selections shown in Figure 6-5. The user search results are shown in Figure 6-4 in Google map while displaying in red tags.

Case 2. In the Figure 6-6 interface shown 6.6 (b) from the web site of rightmove.co.uk a property letting site, the main application of this website is to find properties to rent and buy for the customers. In this interface, for example, a user is looking for a property in Birmingham area within three miles radius from the city centre. The possibilities for selection to the users are given in dropdown lists. In this case, users have to select according to his/her requirement and then finally click on find properties button to search for the results.

Case 3. The interface shown in Figure 6.6 (c-d) from Aston University website display the courses for selection, which is another kind of interface for users. In this example, it displays the short professional courses to select where a user can select according to his/her requirements. In this given example the user wants to find all courses witch provide part-time degrees. Although, in this example, user can select multiple options from Figure 6.6 (c) while using proposed multi-dimensional interface these options can be enhanced and grouped into different dimensions to categorize. Using the normal forms for specification, although Figure 6.6 (d) is displays a simple interface for search but there is nothing to select and the interface only allows for typing keywords and sentences i.e. book name, authors and paper name.

The given interfaces in the use case the Right Move Letting Agency provides the dropdown list for multiple sections and the Aston University case provides the interface for selection. In comparison, the proposed model in this research provides the options for selection in a multi-dimensional way. These options are displayed in a user-friendly graphic design. The proposed interface is flexible for adding subcategories. For example, the Aston University web site interface, to add full time course in two or three-year courses time limits and part-time for four or five-year courses duration, we just add two new coordinates “full time” and “part-time” as a new dimension.

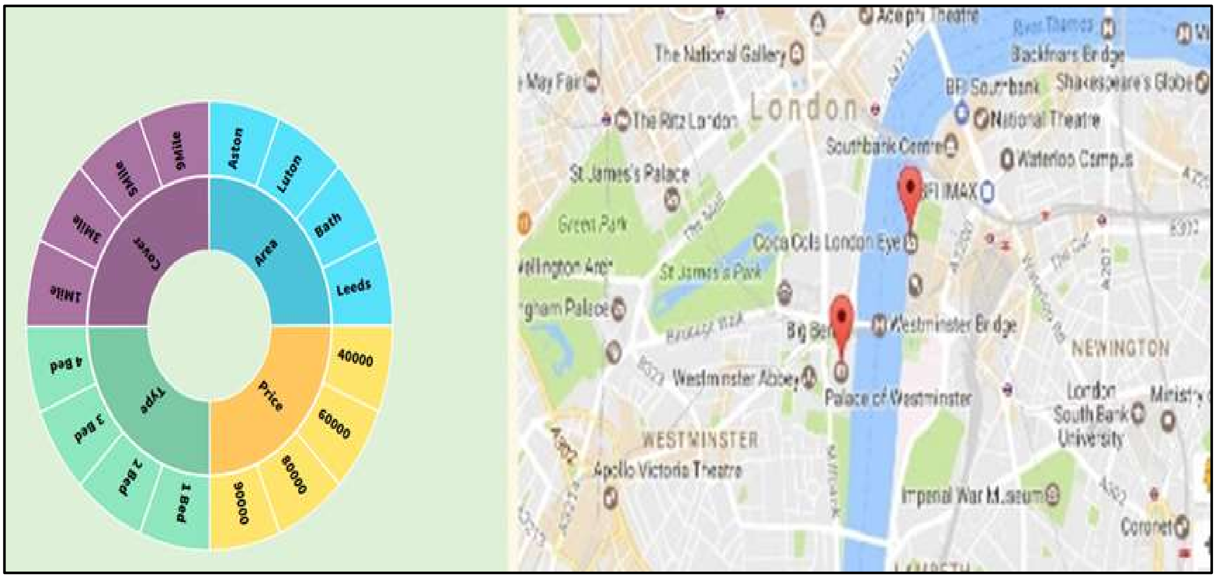
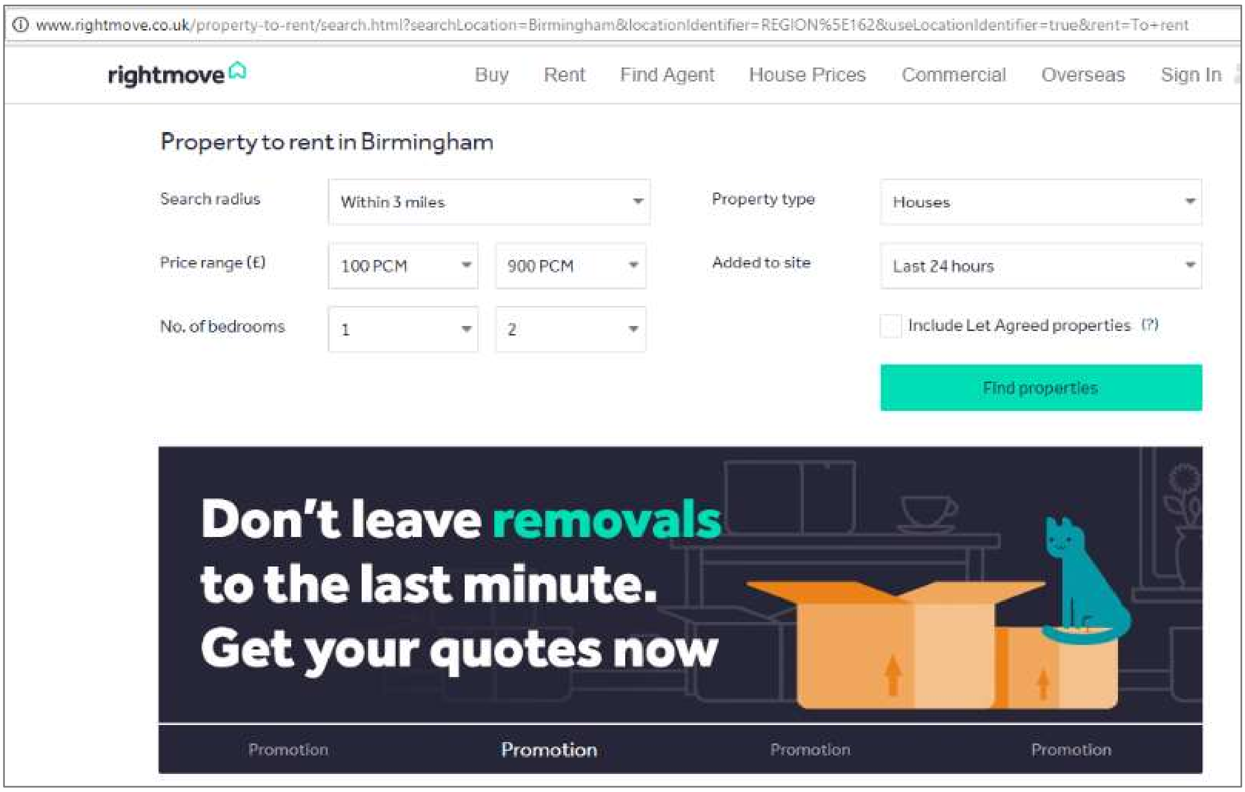


Figure 6.6 (a)



ⓘ www.rightmove.co.uk/property-to-rent/search.html?searchLocation=Birmingham&locationIdentifier=REGION%5E162&useLocationIdentifier=true&rent=To+rent

rightmove

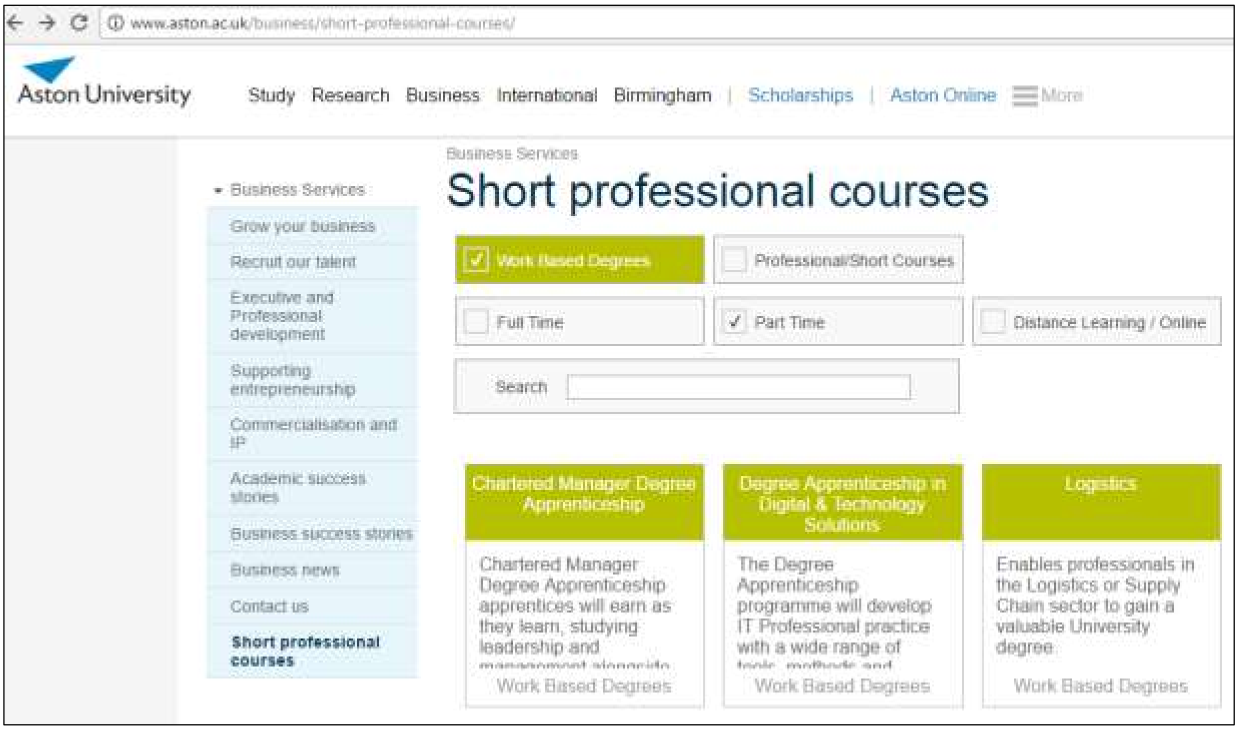
Buy Rent Find Agent House Prices Commercial Overseas Sign In

Property to rent in Birmingham

Search radius	Within 3 miles	Property type	Houses
Price range (£)	100 PCM / 900 PCM	Added to site	Last 24 hours
No. of bedrooms	1 / 2		□ Include Let Agreed properties (?)

Find properties

Figure 6.6 (b)



www.aston.ac.uk/business/short-professional-courses/

Aston University

Study Research Business International Birmingham | Scholarships | Aston Online ☰ More

- Business Services
  - Grow your business
  - Recruit our talent
  - Executive and Professional development
  - Supporting entrepreneurship
  - Commercialisation and IP
  - Academic success stories
  - Business success stories
  - Business news
  - Contact us
  - **Short professional courses**

Business Services

Short professional courses

☑ Work Based Degrees □ Professional/Short Courses

□ Full Time ☑ Part Time □ Distance Learning / Online

Search

Chartered Manager Degree Apprenticeship

Chartered Manager Degree Apprenticeship apprentices will earn as they learn, studying leadership and management alongside

Work Based Degrees

Degree Apprenticeship in Digital & Technology Solutions

The Degree Apprenticeship programme will develop IT Professional practice with a wide range of tools, methods and

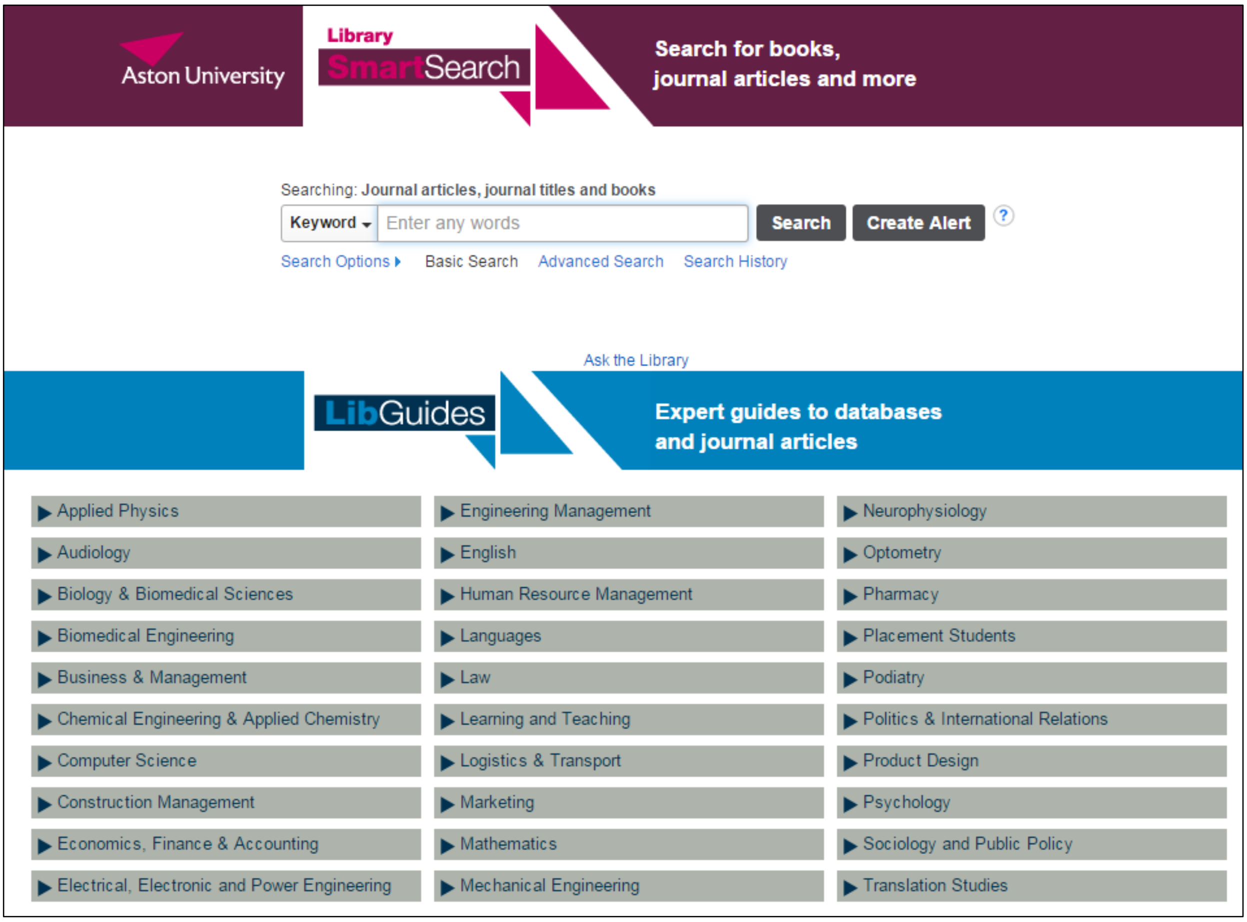
Work Based Degrees

Logistics

Enables professionals in the Logistics or Supply Chain sector to gain a valuable University degree

Work Based Degrees

Figure 6.6 (c)



Aston University

Library SmartSearch

Search for books, journal articles and more

Searching: **Journal articles, journal titles and books**

Keyword ▾ Enter any words **Search** **Create Alert** ?

Search Options ▸ Basic Search Advanced Search Search History

Ask the Library

LibGuides

Expert guides to databases and journal articles

- ▶ Applied Physics
- ▶ Audiology
- ▶ Biology & Biomedical Sciences
- ▶ Biomedical Engineering
- ▶ Business & Management
- ▶ Chemical Engineering & Applied Chemistry
- ▶ Computer Science
- ▶ Construction Management
- ▶ Economics, Finance & Accounting
- ▶ Electrical, Electronic and Power Engineering
- ▶ Engineering Management
- ▶ English
- ▶ Human Resource Management
- ▶ Languages
- ▶ Law
- ▶ Learning and Teaching
- ▶ Logistics & Transport
- ▶ Marketing
- ▶ Mathematics
- ▶ Mechanical Engineering
- ▶ Neurophysiology
- ▶ Optometry
- ▶ Pharmacy
- ▶ Placement Students
- ▶ Podiatry
- ▶ Politics & International Relations
- ▶ Product Design
- ▶ Psychology
- ▶ Sociology and Public Policy
- ▶ Translation Studies

Figure 6.6 (d)

Figure 6-6 Comparison with other Interfaces.

7 CHAPTER SEVEN: CONCLUSIONS AND FUTURE WORK

A challenge is to create a unified model for managing various resources in different spaces. The model should reflect the basic form and motion of various spaces.This research verifies the roles of semantic link network and the Resource Space Model in effectively representing and managing various types of resources, demonstrates the advantages of the models, uncovers some rules through applications, and generalize a methodology for analyzing and managing various resources with semantic links and dimensions.

Human consciously and subconsciously wave various semantic link networks in lifetime, and act intelligently based on the semantic link networks in various spaces and through spaces. The Semantic Link Network model includes form and semantics. Previous studies show that Semantic Link Network in representing and understanding documents for multi-document summarization. We propose a novel summarization framework by first transforming documents into a Semantic Link Network and based on semantic discourse relations and ranking and extracting the informative clauses according to their relations and roles, and also purpose a model for recommending research collaboration.

In the experiments, we compare our proposed approach of RSM, managing and classifying the resources by taking software engineering models as experimental data and proposing a new method. Two case studies are included to draw the results that implementing the RAM is a way to find the application solutions. In the previous studies defied there is no universal method to develop the solution by using the software methods.

The main contribution of this work consists of the following aspects:

1. A new text summarization method is proposed by segmenting a document into clauses based on semantic discourse relations and ranking and extracting the informative clauses according to their relations and roles. The model benefits from using semantic link network, ranking techniques and language characteristics when building the semantic link network on the scientific papers. Compared with other summarization approaches, the proposed approach achieves a higher recall score. Three implications are obtained from this research.
2. A model for recommending research collaboration is proposed by extracting a semantic link network of three types of semantic nodes and three types of semantic links from scientific publications. Experiments on three data sets of scientific publications show that the model achieves a good performance in predicting future

collaborators. Research further unveils different semantic links play different roles in representing texts.

3. A multi-dimensional method for managing software engineering processes is developed. Software engineering processes are mapped into multiple dimensions for supporting analysis, development and maintenance of software systems. It can be used to uniformly classify and manage software methods and models through multiple dimensions so that software systems can be developed with appropriate methods. Further, interface for visualizing Resource Space Model is developed to demonstrate the advantages of the proposed method by keeping consistency among interface, the model structure and operations on the resource space.

The work makes a significant contribution to semantic modelling and effective management of various resources through applications in multiple areas.

This thesis focuses on verifying the roles of Semantic Link Network and the Resource Space Model in effectively managing various types of resources and demonstrating the advantages the models. It takes an initial step toward the organization and management of large scientific resources based on resource space model. Future research will involve multiple areas such as natural language processing, software engineering processes and data science. The following are future works.

1. Automatically discover hierarchical classification dimension from a set of texts without any human involvement, and use the co-occurrence words of texts in the same way to construct the dimensions without the need to manually set any parameters of the model. With the model, a text can belong to one or more categories according to its contents, and retrieval operations can be applied to hierarchical dimensions to help users retrieve texts on specific topics or texts on multiple topics. We shall use scientific papers as experimental data.

2. Use the RSM to uniformly manage more software methods, software programs and documents so that software development can carry out with the support from methods, software programs and documents.

Appendices

My Publications.

[1] M. A. Rafi, "Visualization of Multi-dimensional Resource Space," *2017 13th International Conference on Semantics, Knowledge and Grids (SKG)*, Beijing, 2017, pp. 182-187, doi: 10.1109/SKG.2017.00038.

[2] M. A. Rafi, "Managing Software Processes with the Multi-Dimensional Resource Space Model," *2018 14th International Conference on Semantics, Knowledge and Grids (SKG)*, Guangzhou, China, 2018, pp. 76-83, doi: 10.1109/SKG.2018.00018.

[3] J. Li and M. A. Rafi, "Utilize Discourse Relations to Segment Document for Effective Summarization," 2019 15th International Conference on Semantics, Knowledge and Grids (SKG), Guangzhou, China, 2019, pp. 12-15, doi: 10.1109/SKG49510.2019.00010.

[4] J. Zhou and M. A. Rafi, "Recommendation of Research Collaborator Based on Semantic Link Network," 2019 15th International Conference on Semantics, Knowledge and Grids (SKG), Guangzhou, China, 2019, pp. 16-20, doi: 10.1109/SKG49510.2019.00011.

[5] Y. Gao and M. A. Rafi, "Efforts towards Combining Graphics, Uncertainty, and Semantics: A Survey," 2019 15th International Conference on Semantics, Knowledge and Grids (SKG), Guangzhou, China, 2019, pp. 81-88, doi: 10.1109/SKG49510.2019.00022.

[6] Gao Y, Rafi MA. Combination of graphics, uncertainty, and semantics: A survey. Concurrency Computat Pract Exper. 2021; e6711. doi: 10.1002/cpe.6711

6 References

[1] H. Zhuge, Resource space model, its design method and applications. The Journal of Systems and Software, Vol. 72, Issue 1, pp. 71–81, 2004.

[2] H. Zhuge, The Web Resource Space Model, Springer, 2008.

[3] H. Zhuge, The Knowledge Grid toward Cyber-Physical Society, World Scientific Publishing Co. Singapore, 2012.

[4] H. Zhuge, Multi-Dimensional Summarization in Cyber-Physical Society, Morgan Kaufmann, 2016.

[5] H. Zhuge, Y. Xing and P. Shi. Resource Space Model, OWL and Database: Mapping and Integration, ACM Transactions on Internet Technology, Vol. 8, Issue 4, 2008.

[6] H. Zhuge and Y. Xing. Probabilistic Resource Space Model for Managing Resources in Cyber-Physical Society, IEEE Transactions on Service Computing, Vol. 5, pp. 404-421, 2012.

[7] X. Yu, L. Peng, Z. Huang and H. Zhuge. A framework for automated construction of resource space based on background knsedge. Future Generation Computer Systems. Vol. 32, pp. 222-231, 2014.

[8] B. Xu and H. Zhuge. Faceted navigation through keyword interaction. World Wide Web, Vol. 17,

[9] H. Zhuge and L. He. Automatic maintenance of category hierarchy. Future Generation Computer System. Vol. 67, pp. 1-12, 2017.

[10] B. Xu and H. Zhuge. An angle-based interest model for text recommendation. Future Generation Computer Systems. Vol. 64, pp. 211-226, 2016.

[11] H. Zhuge, The Complex Semantic Space Model. IEEE 20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. pp. 9-15, 2011.

[12] H. Zhuge and Y. Xing. Integrity Theory for Resource Space Model and Its Application. Springer Verlag Berlin Heidelberg. pp. 8-24, 2005.

[13] H. Zhuge, The Knowledge Grid, World Scientific Publishing Co., Singapore, 2004.

[14] H. Zhuge, Communities and Emerging Semantics in Semantic Link Network: Discovery and Learning, IEEE Transactions on Knowledge and Data Engineering, vol.21, no.6, 2009, pp. 785-799.

[15] H. Zhuge, Semantic linking through spaces for cyber-physical-socio intelligence: A methodology, Artificial Intelligence, pp. 988-1019, 2011.

[16] H. Zhuge and Z. Zeng. A Bigtree Index for Resource Space Model. Sixth International Conference on Semantics, Knowledge and Grids, Beijing pp. 424-425, 2010.

[17] H. Zhuge, Human-Machine-Nature Symbiosis on Cyber-Physical-Social Intelligence, Springer, 2019.

[18] I. Sommerville Software Engineering 10 Edition, Global Edition. Pearson Education Limited England, 2016.

[19] R. S. Pressman. Software Engineering: A practitioner's approach. The McGraw-Hill Companies, Inc., United States, 2010.

[20] K. Wiegers and J. Beatty. Software Requirements Third Edition: Microsoft Press A Division of Microsoft Corporation One Microsoft Way Redmond, Washington, 2013.

[21] L. Mikael, R. B. Victor, B. Barry, C. Patrícia, C. D. Kathleen, S. Forrest, T. T. Roseanne, A. W. Laurie and Z. Marvin. Empirical Findings in Agile Methods. LNCS, Vol. 2418. pp. 197-207, 2002.

[22] N. Keshta and Y. L. Moregan. Comparison between Traditional Plan-based and Agile Software Processes According to Team Size & Project Domain. 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference. pp. 567-575, 2017.

[23] B. M. Amen and J. Lu. Sketch of Big Data Real-Time Analytics Model. The Fifth International Conference on Advances in Information Mining and Management. pp. 48-53, 2015.

[24] F. Ji and T. Sedano. Comparing Extreme Programming and Waterfall Project Results. 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T), Honolulu, HI. pp. 482-486, 2011.

[25] V. Rastogi, Software Development Life Cycle Models - Comparison, Consequences. International Journal of Computer Science and Information Technologies. Vol. 6 no.1, pp. 168-172, 2015.

[26] M. A. Rafi, Visualization of Multi-Dimensional Resource Space. 13th International Conference on Semantics, Knowledge and Grids (SKG), Beijing. pp. 182-187, 2017.

[27] S. A. Peixoto, Human-Computer Interface Expert System for Agile Methods. Proceedings of the ITI 2009 31ST International Conference on Information Technology Interfaces. pp. 311-316, 2009.

[28] G. Kumar and P. K. Bhatia. Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies. Fourth International Conference on Advanced Computing & Communication Technologies. pp. 189-196, 2014.

[29] R. R. Raval and H. Rathod. Comparative Study of Various Process Model in Software Development. International Journal of Computer Applications. Vol. 82, pp. 16-19, 2013.

[30] A. F. Chowdhury and M. N. Huda, Comparison between Adaptive Software Development and Feature Driven Development. International Conference on Computer Science and Network Technology, Vol. 1, pp. 363-367, 2011.

[31] B. Boehm and R. Turner. Balancing Agility and Discipline: A Guide for the Perplexed. Boston, MA; London: Addison-Wesley, 2004.

[32] J. A. Highsmith, Agile Software Development Ecosystems. Boston, MA : Addison-Wesley, 2002.

[33] J. Braude and M. E. Bernstein. Software Engineering Modern Approaches Second Edition, Waveland Press, Inc. 2016.

[34] H. Zhuge, Interactive Semantics, Artificial Intelligence, vol.174, pp.190-204, 2010.

[35] H. Zhuge and X. Li, Peer-to-Peer in Metric Space and Semantic Space, IEEE Transactions on Knowledge and Data Engineering, vol.19, no.6, pp.759-771, 2007.

[36] H. Zhuge, Socio-Natural Thought Semantic Link Network: A Method of Semantic Networking in the Cyber Physical Society, Keynote at IEEE AINA 2010, Perth, Australia, 20-23 April, pp.19-26, 2010.

[37] H. Zhuge, The Future Interconnection Environment, IEEE Computer, vol. 38, no. 4, pp. 27-33, 2005.

[38] H. Zhuge, Discovery of Knowledge Flow in Science, Communications of the ACM, vol. 49, no.5, pp. 101-107, 2006.

[39] H. Zhuge, Mapping Big Data into Knowledge Space with Cognitive Cyber-Infrastructure. 2015. CoRR abs/1507.06500.

[40] H. Zhuge, A knowledge flow model for peer-to-peer team knowledge sharing and management. Expert System with Applications, vol. 23, no.1, pp. 23-30, 2002.

[41] H. Zhuge, J. Ma, X. Shi, Abstraction and analogy in cognitive space: A software process model. Information & Software Technology, vol. 39, no.7, pp. 463-468, 1997.

[42] H. Zhuge, Inheritance rules for flexible model retrieval. Decision Support Systems, vol. 22, no. 4, pp. 379-390, 1998.

[43] H. Zhuge and X. Shi, Communication cost of cognitive co-operation for distributed team development. Journal of Systems and Software, vol. 57, no.3, pp. 227-233, 2001.

[44] H. Zhuge, T. Y. Cheung, H. K. Pung, A timed workflow process model. Journal of Systems and Software, vol. 55, no.3, pp. 231-243, 2001.

[45] H. Zhuge, Conflict decision training through multi-space co-operation. Decision Support Systems, vol. 29, no.2, pp. 111-123, 2000.

[46] H. Zhuge, A process matching approach for flexible workflow process reuse. Information & Software Technology, vol. 44, no. 8, 2002, pp.445-450, 2002.

[47] H. Zhuge, A knowledge flow model for peer-to-peer team knowledge sharing and management. Expert Systems with Applications, vol. 23, no.1, pp.23-30, 2002.

[48] H. Zhuge, Component-based workflow systems development. Decision Support Systems, vol. 35, no.4, pp.517-536, 2003.

[49] H. Zhuge, Workflow- and agent-based cognitive flow management for distributed team Cooperation. Information & Management, vol. 40, no.5, pp. 419-429, 2003.

[50] H. Zhuge, An inexact model matching approach and its applications. Journal of Systems and Software, vol. 67, no.3, pp. 201-212, 2003.

[51] H. Zhuge, Fuzzy resource space model and platform. Journal of Systems and Software, vol. 73, no.3, pp. 389-396, 2004.

[52] H. Zhuge, J. Chen, Y. Feng, X. Shi, A federation-agent-workflow simulation framework for virtual organisation development. Information & Management, vol. 39, no.4, pp. 325-336, 2002.

[53] X. Sun and H. Zhuge, Summarization of Scientific Paper through Reinforcement Ranking on Semantic Link Network, IEEE Access, vol. 6, pp. 40611 – 40625, 2018.

[54] Koller D, Friedman N. Probabilistic Graphical Models: Principles and Techniques. The MIT Press; 2009.

[55] Bishop CM. Pattern Recognition and Machine Learning. Springer; 2006.

[56] Pearl J. Bayesian networks: a model of self-activated memory for evidential reasoning. Proceedings of the 7th Conference of the Cognitive Science Society; 1985:329-334.

[57] Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann; 1988. https://doi.org/10.1016/C2009-0-27609-4

[58] Kindermann R, Snell JL. Markov random fields and their applications. Vol. 1, American Mathematical Society; 1980. doi:10.1090/conm/001

[59] Preston CJ. Gibbs States on Countable Sets. Cambridge University Press; 1974. doi:10.1017/CBO9780511897122

[60] Spitzer FL. Random Fields and Interacting Particle Systems. Mathematical Association of America; 1971.

[61] Moussouris J. Gibbs and Markov random systems with constraints. J Stat Phys. 1974;10(1):11-33. doi:10.1007/BF01011714

[62] GyftodimosE,FlachPA.HierarchicalBayesiannetworks:anapproachtoclassificationandlearning forstructureddata.In:VourosGA,PanayiotopoulosT, eds. Lecture Notes in Computer Science: Vol. 3025. Methods and Applications of Artificial Intelligence. Springer; 2004:291-300. https://doi.org/10.1007/9783-540-24674-9_31

[63] Fine S, Singer Y, Tishby N. The hierarchical hidden Markov model: analysis and applications. Mach Learn. 1998;32(1):41-62. doi:10.1023/A: 1007469218079

[64] AlJadda K, Korayem M, Ortiz C, Grainger T, Miller JA, York WS. PGMHD: a scalable probabilistic graphical model for massive hierarchical data problems. Proceedings of 2014 IEEE International Conference on Big Data (Big Data); 2014:55-60. 10.1109/BigData.2014.7004213

[65] Loy CC, Xiang T, Gong S. Modelling activity global temporal dependencies using time delayed probabilistic graphical model. Proceedings of 2009 IEEE 12th International Conference on Computer Vision; 2009:120-127. 10.1109/ICCV.2009.5459156

[66] Fraser AM, Swinney HL. Independent coordinates for strange attractors from mutual information. Phys Rev A. 1986;33(2):1134-1140. doi:10.1103/ PhysRevA.33.1134

[67] Shakya S, Zhang J. Uncertain reasoning using time-dynamic Markov random field for sensor-network applications. Proceedings of the 28th International Florida Artificial Intelligence Research Society Conference; 2015:588-593.

[68] Richardson M, Domingos P. Markov logic networks. Mach Learn. 2006;62(1):107-136. https://doi.org/10.1007/s10994-006-5833-1

[69] Lee J, Wang Y. A probabilistic extension of the stable model semantics. Logical Formalizations of Commonsense Reasoning: Papers from the 2015 AAAI Spring Symposium, Technical report SS-15-04; 2015:96-102.

[70] Bach SH, Broecheler M, Huang B, Getoor L. Hinge-loss Markov random fields and probabilistic soft logic. J Mach Learn Res. 2017;18(1):3846-3912.

[71] Broecheler M, Getoor L. Probabilistic similarity logic. Proceedings of International Workshop on Statistical Relational Learning (SRL); 2009. https://dtai. cs.kuleuven.be/events/ilp-mlg-srl/USBStick/papers/SRL09-22.pdf

[72] Broecheler M, Getoor L. Computing marginal distributions over continuous Markov networks for statistical relational learning. Proceedings of the 23 rd International Conference on Neural Information Processing Systems; vol. 1, 2010:316-324.

[73] Sowa JF. Semantic networks. In: Nadel L, ed. Encyclopedia of Cognitive Science. Wiley; 2006. doi:10.1002/0470018860.s00065

[74] Lehmann F. Semantic Networks in Artificial Intelligence. Pergamon Press; 1992.

[75] Richens RH. Preprogramming for mechanical translation. Mech Transl Comput Linguist. 1956;3(1):20-25.

[76] Sowa JF. Conceptual graphs for a data base interface. IBM J Res Develop. 1976;20(4):336-357. doi:10.1147/rd.204.0336

[77] Berners-Lee T, Hendler J, Lassila O. The semantic web. Sci Am. 2001;284(5):34-43.

[78] Bordes A, Usunier N, Garcia-Durán A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. Proceedings of the 26 th International Conference on Neural Information Processing Systems; vol. 2, 2013:2787-2795.

[79] Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G. Complex embeddings for simple link prediction. Proceedings of the 33rd International Conference on Machine Learning; Vol. 48, 2016:2071–2080. https://proceedings.mlr.press/v48/trouillon16.html

[80] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. IEEE Trans Neural Netw. 2009;20(1):61-80. doi:10.1109/ TNN.2008.2005605

[81] Hamaguchi T, Oiwa H, Shimbo M, Matsumoto Y. Knowledge transfer for out-of-knowledge-base entities: a graph neural network approach. Proceedings of the 26th International Joint Conference on Artificial Intelligence; 2017:1802-1808. 10.24963/ijcai.2017/250

[82] Schlichtkrull M, Kipf TN, Bloem P, van den Berg R, Titov I, Welling M. Modeling relational data with graph convolutional networks. In: Gangemi A, Navigli R, Vidal M, et al., eds. Lecture Notes in Computer Science: Vol. 10843. The Semantic Web. Springer; 2018:593-607. https://doi.org/10.1007/978-201032010319-201093417-20104_38

[83] Cao M, Sun X, Zhuge H. The contribution of cause-effect link to representing the core of scientific paper—the role of semantic link network. PLOS One. 2018;13(6):e0199303. doi:10.1371/journal.pone.0199303

[84] Gao Y & Rafi M Efforts towards combining graphics, uncertainty, and semantics: a survey. Proceedings of 2019 15th International Conference on Semantics, Knowledge and Grids (SKG); 2019:81-88. 10.1109/SKG49510.2019.00022

[85] Davy C., Arno D. B. M., and Mohamed A. Introducing Data Science. Manning Publications Co.: United States of America., 2016.

[86] D. David, H. Barry, and Y. Beibei EMC Education Services Data Science & Big Data Analytics: Discovering, Analysing, Visualizing and Presenting Data. John Wiley & Sons, Inc.: United States of America and Published simultaneously in Canada. 2015.

[87] A. Halevy, M. Franklin and D. Maier, Principles of dataspace systems, ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems (PDOS), 2006, pp.1-9.

[88] M. A. Hearst. Design Recommendations for Hierarchical Faceted Search Interfaces. ACM SIGIR workshop on faceted search, 2006.

[89] A. Nenkova, S. Maskey, and L. Yang, "Automatic summarization," in Meeting of the Association for Computational Linguistics: Tutorial Abstracts of Acl, 2011, pp. 103-233.

[90] A. Nenkova and K. McKeown, "A survey of text summarization techniques," in Mining text data: Springer, 2012, pp. 43-76.

[91] S. Kobayashi and K. Nomizu, "Interscience Tracts in Pure and Applied Mathematics," FOUNDATIONS, vol. 1, no. 15, 1969.

[92] F. Wolf and E. Gibson, "Representing discourse coherence: A corpus-based study," Computational linguistics, vol. 31, no. 2, pp. 249-287, 2005.

[93] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in Proceedings of the 2004 conference on empirical methods in natural language processing, 2004, pp. 404-411.

[94] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," Journal of artificial intelligence research, vol. 22, pp. 457-479, 2004.

[95] M. Cao and H. Zhuge, "What Size of Language Unit Is More Appropriate for Text Summarization?," in 2018 14th International Conference on Semantics, Knowledge and Grids (SKG), 2018, pp. 196-202: IEEE.

[96] W. C. Mann and S. A. Thompson, "Rhetorical structure theory: Toward a functional theory of text organization," Text-interdisciplinary Journal for the Study of Discourse, vol. 8, no. 3, pp. 243-281, 1988.

[97] A. Louis, A. Joshi, and A. Nenkova, "Discourse indicators for content selection in summarization," in Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2010, pp. 147-156: Association for Computational Linguistics.

[98] M. Kraus and S. Feuerriegel, "Sentiment analysis based on rhetorical structure theory: Learning deep neural networks from discourse trees," Expert Systems with Applications, vol. 118, pp. 65-79, 2019.

[99] Y. Ji and J. Eisenstein, "Representation learning for text-level discourse parsing," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, vol. 1, pp. 13-24.

[100] C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273-297, 1995.

[101] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, 2014, pp. 55-60.

[102] H. Hernault, H. Prendinger, and M. Ishizuka, "HILDA: A discourse parser using support vector machine classification," Dialogue & Discourse, vol. 1, no. 3, 2010.

[103] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in Text summarization branches out, 2004, pp. 74-81.

[104] J. Paredaens, P. D. Bra, M. Gyssens, and D. V. Gucht, "The Structure of the Relational Database Model," Springer Science & Business Media, vol. 17, 2012.

[105] M. Franklin, A. Halevy, and D. Maier, "From databases to dataspaces: a new abstraction for information management," ACM Sigmod Record, vol. 34, no. 4, pp. 27-33, 2005.

[106] K. Nigam, A. K. Mccallum, S. Thrun, and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents using EM," Machine Learning, vol. 39, no. 2-3, pp. 103-134, 2000.

[107] B. Liu, X. Li, W. S. Lee, and P. S. Yu, "Text Classification by Labeling Words," in Proceedings of the Nineteenth National Conference on Artificial Intelligence, vol. 4, pp. 425-430, 2004.

[108] C. Li, X. Jian, A. Sun, and Z. Ma, "Effective Document Labeling with Very Few Seed Words: A Topic Model Approach," Proceedings of the 25th ACM international on conference on information and knowledge management, pp. 85-94, 2016.

[109] S. J. Blair, Y. Bi, and M. D. Mulvenna, "Unsupervised Sentiment Classification: A Hybrid Sentiment-Topic Model Approach," in IEEE International Conference on Tools with Artificial Intelligence, pp. 453-460, 2017.

[110] D. Zha and C. Li, "Multi-label Dataless Text Classification with Topic Modeling," Knowledge and Information Systems, vol. 61, no. 1, pp. 137-160, 2019.

[111] R. C. Dubes and A. K. Jain, "Algorithms for clustering data," ed: Prentice hall Englewood Cliffs, 1988.

[112] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 7, pp. 881-892, 2002.

[113] I. S. Dhillon, J. Fan, and Y. Guan, "Efficient clustering of very large document collections," in Data mining for scientific and engineering applications: Springer, pp. 357-381, 2001.

[114] G. Hu, S. Zhou, J. Guan, and X. Hu, "Towards effective document clustering: A constrained K-means based approach," Information Processing & Management, vol. 44, no. 4, pp. 1397-1409, 2008.

[115] T. L. Griffiths and M. Steyvers, "A probabilistic approach to semantic representation," in Proceedings of the annual meeting of the cognitive science society, vol. 24, no. 24, pp. 381-386, 2002.

[116] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," Journal of machine Learning research, vol. 3, no. Jan, pp. 993-1022, 2003.

[117] P. Xie and E. P. Xing, "Integrating document clustering and topic modeling," arXiv preprint arXiv:1309.6874, pp. 694-703, 2013.

[118] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513-523, 1987.

[119] H. Van Ditmarsch, S. Ghosh, R. Verbrugge, and Y. Wang, "Hidden protocols: Modifying our expectations in an evolving world," Artificial Intelligence, vol. 208, pp. 18-40, 2014.

[120] C. Yang, Collaborator recommendation on research social network platforms, Anhui:PhD, University of Science and Technology of China, 2015.

[121] D. W. McDonald and M. S. Ackerman, "Expertise recommender: a flexible recommendation system and architecture", Proceedings of the 2000 ACM conference on Computer supported cooperative work, pp. 231-240, 2000.

[122] K. Balog, L. Azzopardi and M. De Rijke, "Formal models for expert finding in enterprise corpora", Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 43-50, 2006.

[123] H. Deng, I. King and M. R. Lyu, "Formal models for expert finding on dblp bibliography data", 2008 Eighth IEEE International Conference on Data Mining, pp. 163-172, 2008.

[124] D. W. McDonald, "Recommending collaboration with social networks: a comparative evaluation", Proceedings of the SIGCHI conference on Humanf actors in computing systems, pp. 593-600, 2003.

[125] L. A. Adamic and E. Adar, "Friends and neighbors on the web", Social networks, vol. 25, no. 3, pp. 211-230, 2003.

[126] L. Katz, "A new status index derived from sociometric analysis", Psychometrika, vol. 18, no. 1, pp. 39-43, 1953.

[127] S. Han, D. He, P. Brusilovsky and Z. Yue, "Coauthor prediction for junior researchers", International Conference on Social Computing Behavioral-Cultural Modeling and Prediction, pp. 274-283, 2013.

[128] T. Huynh, K. Hoang and D. Lam, "Trend based vertex similarity for academic collaboration recommendation", International Conference on Computational Collective Intelligence, pp. 11-20, 2013.

[129] G. R. Lopes, M. M. Moro, L. K. Wives and J. P. M. De Oliveira, "Collaboration recommendation on academic social networks", International conference on conceptual modeling, pp. 190-199, 2010.

[130] S. Cohen and L. Ebel, "Recommending collaborators using keywords", Proceedings of the 22nd International Conference on World Wide Web, pp. 959-962, 2013.

[131] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal and J. Han, "Co-author relationship prediction in heterogeneous bibliographic networks", 2011 International Conference on Advances in Social Networks Analysis and Mining, pp. 121-128, 2011.

[132] D. H. Lee, P. Brusilovsky and T. Schleyer, "Recommending collaborators using social features and mesh terms", Proceedings of the American Society for Information Science and Technology, vol. 48, no. 1, pp. 1-10, 2011.

[133] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613_620.

[134] G. Salton, A. Singhal, M. Mitra, C. Buckley, Automatic text structuring and summarization, Inf. Process. Manage. 33 (2) (1997) 193_207.

[135] J. Chen, "A Survey on the Resource Space Model," *2011 Seventh International Conference on Semantics, Knowledge and Grids*, Beijing, 2011, pp. 183-186, doi: 10.1109/SKG.2011.43.

[136] J. Zhou, "Resource Space Model: A Survey," 2018 14th International Conference on Semantics, Knowledge and Grids (SKG), Guangzhou, China, 2018, pp. 277-280, doi: 10.1109/SKG.2018.00051.

[137] Awad, M.A., 2005. A comparison between agile and traditional software development methodologies. University of Western Australia, 30.

[138] B. Ma and H. Zhuge, "Discovering Classification Dimensions for Managing Scientific Resources," 2019 15th International Conference on Semantics, Knowledge and Grids (SKG), Guangzhou, China, 2019, pp. 97-102, doi: 10.1109/SKG49510.2019.00024.

[139] Zhuge H. Active e-document framework ADF: model and tool. Inf Manag. 2003;41(1):87-97. https://doi.org/10.1016/S0378-20107206(03)0002920106

[140] Zhuge H. Autonomous semantic link networking model for the knowledge grid. Concurr Comput Pract Exper. 2007;19(7):1065-1085. doi:10.1002/cpe. 1097

[141] Zhuge H. Cyber-Physical-Social Intelligence: On Human-Machine-Nature Symbiosis. Springer; 2020. https://doi.org/10.1007/978-2010981-20101320107311-20104

[142] Zhuge H, Xu B. Basic operations, completeness and dynamicity of cyber physical socio semantic link network CPSocio-SLN. Concurr Comput Pract Exper. 2011;23(9):924-939. doi:10.1002/cpe.1623

[143] M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan and M. M. Kabir, "A Survey of Automatic Text Summarization: Progress, Process and Challenges," in IEEE Access, doi: 10.1109/ACCESS.2021.3129786.

[144] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, S.Y. Philip et al., "Top 10 algorithms in data mining", Knowledge and information systems, vol. 14, no. 1, pp. 1-37, 2008

[145] L. Jiang, Z. Cai, H. Zhang and D. Wang, "Naive bayes text classifiers: a locally weighted learning approach", Journal of Experimental & Theoretical Artificial Intelligence, vol. 25, no. 2, pp. 273-286, 2013.

[146] Qing Li and Caroline Yao, “Real-Time Concepts for Embedded Systems”, CMP Books, ISBN: , 2003