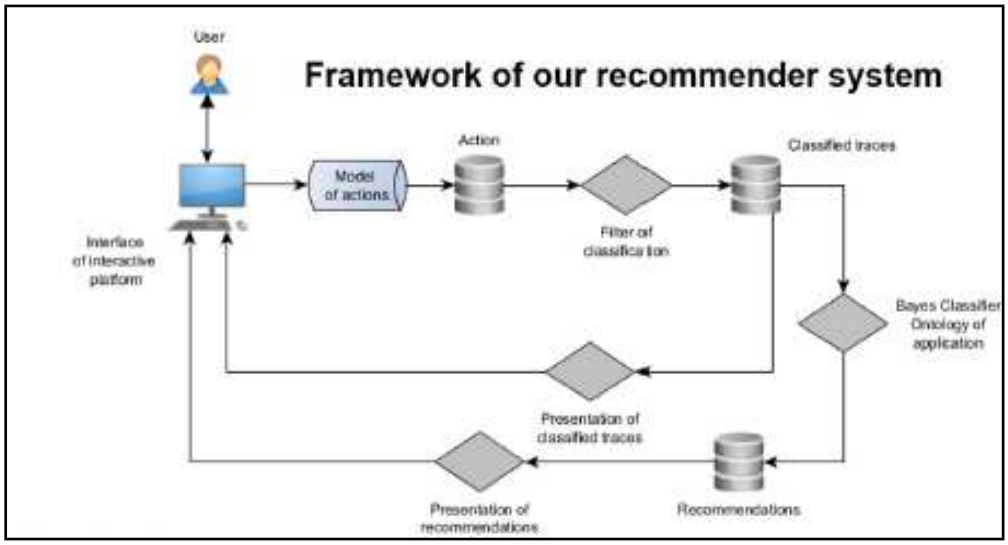
Towards a competency recommender system from collaborative traces

Ning Wang

Par **Ning WANG**

Towards a competency recommender system from collaborative traces

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 20 octobre 2016
Spécialité : Information Technology : Unité de recherche Heudyasic (UMR-7253)
D2300

UNIVERSITY OF TECHNOLOGY OF COMPIÈGNE

DOCTORAL THESIS

Towards a competency recommender system from collaborative traces

Author:
Ning WANG

Supervisors:
Marie-Hélène ABEL
Jean-Paul BARTHÈS
Elsa NEGRE

A thesis submitted in fulfillment of the requirements for the degree of Doctor in Information Technology

in the

Information, Knowledge, Interaction
Heudiasyc Laboratory

20 october 2016

"Live a more than desultory life; open arms to poems and distant lands."

Xiaosong Gao

UNIVERSITY OF TECHNOLOGY OF COMPIÈGNE

Abstract

Heudiasyc Laboratory

Doctor in Information Technology

Towards a Competency Recommender System from Collaborative Traces

by Ning WANG

With the development of information and Internet technology, human society has stepped into an era of information overload. Owing to the overwhelming quantity of information, both information providers and information consumers are facing challenges: information providers want the information to be transferred to the target audience while information consumers need to find the information most relevant to their need. To bridge the gap, recommender systems have been designed and applied in a variety of applications to help making decisions on movies, music, news and even services and persons. In a Collaborative Working Environment, recommender systems are also needed to guide collaboration and allocate task efficiently.

When people exchange information and resources, they leave traces in some way or other. For a typical Web-based Collaborative Working Environment, traces can be recorded which are mainly produced by collaborative activities or interactions. The modelled traces represent knowledge as well as experience concerning the interactive actions among users and resources. Such traces can be defined, modelled and exploited in return to offer a clue on a variety of deductions. Firstly they can indicate whether a user is active or not concerning interactions on a certain subject. Combining with users' evaluation of the information and resources during interaction, we can further evaluate a user's competency on each subject. This aids the decision for further collaboration because knowing the specialization of users helps to distribute tasks reasonably.

This thesis focuses on implementing a recommender system by exploiting various collaborative traces in the group shared/collaborative workspace. To achieve this goal, firstly we collect traces and get them filtered by system filters. For evaluating shared resources we propose a system of vote and combine the result with collaborative traces. Furthermore, we present two mathematical approaches (TF-IDF and Bayes Classifier) with semantic meanings of traced resources and a machine learning method (Logistic Regression) with user profile to exploit traces, and then discuss comprehensive examples. As a practical experience we tested our prototype in the context of the E-MEMORAe collaborative platform. By comparing the results of experiments we assess the strengths and weaknesses of each of the three methods and in which scenario they perform better. Cases show that our exploitation framework and various methods can facilitate both personal and collaborative work and help decision-making.

Vers un système de recommandations de compétences à partir de traces collaboratives

Par Ning Wang

Soutenue le 20 octobre 2016 devant le jury composé de :

M.	D. LENNE (Président)
Mme	M-H. ABEL (Directrice de thèse)
M.	J-P. BARTHÈS (Directeur de thèse)
Mme	E. NÈGRE (Directrice de thèse)
M.	M. CHEVALIER (Rapporteur)
Mme	S. DESPRÉS (Rapporteur)
M.	B. MAYAG
Mme	I. SAAD

Acknowledgements

This thesis and all the work related to it were carried out in the laboratory Heudiasyc at the University of Technology of Compiègne (UTC) from September 2013 to present. Huge thanks go to the encouragement and support from my directors, colleagues, friends, and family members that help me overcome all the challenges and difficulties.

First and foremost, I owe my greatest debt of gratitude to my beloved country (China Schloarship council and Harbin Institute of Technology) that backed me up for five years staying in France. At the meantime, I would also like to express my sincerest gratitude to my supervisors: Mrs. Marie-Hélène ABEL, professor at the University of Technology of Compiègne, Mr. Jean-Paul BARTHÈS, Emeritus professor at the University of Technology of Compiègne, and Ms. Elsa NEGRE, Associate professor at the Paris Dauphine University. They have guided me throughout my thesis with their great patience, dedication, and knowledge respectively in their domain. Their perfect collaborative guidance is a must for the success of this thesis.

Moreover, I want to thank all the members of the laboratory Heudiasyc and colleagues in the ICI team, especially Marcio FUCKNER, Ala ATRASH, Hanen BELLILI, Majd SAIED, Yue KANG, Kun JIANG, Chunlei YU, Gregoryy WANDERLEY, Idir BENOUARET, and Freddy KAMDEM SIMO for their help, accompanying, and friendship.

Finally, my deepest appreciation is dedicated to my family, especially my parents who supported me all along. Your love, comprehension and encouragement power my improvement.

Contents

List of Figures

List of Tables

List of Abbreviations

CF	**C**ollaborative **F**iltering
CMS	**C**competency **M**anagement **S**ystem
CQA	**C**ommunity-based **Q**uestion **A**nswering service
CRS	**C**ompetency **R**ecommender **S**ystem
CT	**C**ollaborative **T**race
CWE	**C**ollaborative **W**orking **E**nvironment
ES&H	**E**nvironment, **S**afety & **H**ealth
ICT	**I**nformation **C**ommunications **T**echnology
KBS	**K**nowledge-**B**ased **S**ystem
KDD	**K**nowledge **D**iscovery and **D**ata mining
NASA	**N**ational **A**eronautics **S**pace **A**dministration
RecSys	**Rec**ommender **Sys**tems
SIGIR	**S**pecial **I**nterest **G**roup on **I**nformation **R**etrieval
TF-IDF	**T**erm **F**requency-**I**nverse **D**ocument **F**requency
TISS	**T**rauma **I**njury **S**everity **S**core

Dedicated to my Beloved Parents…

Chapter 1

Introduction

1.1 Context

With the progress of science and technology, solving a technical or engineering problem needs more than ever numerous persons with competencies in different domains to work together. Collaboration is a kind of group work pattern that unites every member's characteristics and competencies. Collaborative attitude has become a required quality by more and more companies. In human history, collaboration is important in almost all domains such as business, education, entertainment and even in wartime.

Collaboration is obtained by applying collaboration tools. The purpose of a collaboration tool is to support a group of two or more individuals to accomplish a common goal or objective they have set themselves (Lomas, Burke, and Page, 2008). Before the Information Age, people used to collaborate through non-technological tools such as paper, flipchart, Post-it notes or whiteboard. Nowadays complex and web-based collaborative software like Wiki or SharePoint integrated in an agile work environment make us more efficient (Waggener et al., 2009). Types of communicating and collaborative tools, such as computer and smartphone, allow people to work together regardless of distance and time. These tools can easily record people behaviors and activities based on which we make judgment of somebody's previous participation and reasonably assess his/her competency. Participants of collaboration work with such tools. Thus, recording and collecting such activities becomes an easy task in a digital environment.

Collaboration is realized among members of a working group. To organize a competitive group, we need to evaluate group members' competencies to better allocate tasks and resources. Thus estimating a person's competency is not only necessary, but also crucial for collaboration. The term competency may be synonymous with skills. A broader definition would be that competency is the sum of skills, knowledge, and behaviors. For example, higher educational institutions are more focused on the informational dimension of competency. Hence for many professions, formal education and graduation are followed by a period of practice typically under the direction of qualified practitioners. Such post-education practical work is where someone picks up skills and behaviors needed to be a competent

practitioner. The need to acquire education, skills, and an ability to perform professional behavior are frequently the requirements of a competent practitioner. More sophisticated definitions of competency would add two more dimensions: (1) the 'level' at which a person may be required to work 'competently'; and (2) the context in which a competency is being exercised.

While competencies are not new to most organizations and companies, what is new is their increased application across varied human resource functions (i.e., recruitment/selection; learning and development; performance management; career development and succession planning; human resource planning). Organizations are looking for new ways to acquire, manage and retain the precious talent needed to be successful.

At the same time, measuring competency is never an easy task because of its intangibility. Moreover, the results of competency assessment is inclined to be influenced by lack of neutrality or subjectivity. Competency management has always been a key sector in Sociology and in Management Science. The management of competency is an old, widely used practice that consists of all of an organization's formal, organized approaches to ensuring that it has the human talents needed to meet its goals. The practice defines the skills in which an organization is interested, such as the ability to use a certain application or knowledge of an accounting practice. Once the skills are defined, each member (or subcontractor) is described based on these standardized definitions. These skills and personnel descriptions are then used to forecast needs, determine training goals, and measure progress toward those goals. Software applications can help organizations to store, search, and analyze competency-related data. Graphical dashboards provide quick views of information across an entire enterprise, and search functions help users identify who has a certain skill. And because the software is an enterprise system installed on a corporate server or provided over the cloud, it facilitates collaboration and knowledge-sharing across departments.

However, in our daily life we are still bothered with the problem that there is a gap between collaboration activities and competency assessment. Following is a scenario describing this problem:

scenario: *"Peter is a college student and joins a study group composed of his peers. Ordinarily, they express their opinions by sharing materials such as their notes, the documents they have read, ask and answer questions to others, etc. Sometimes they appreciate what others share and propose and sometimes they don't. Each of them has his/her own specialty, i.e., a group member Julie is good at C++ programming. When a member meets with difficulties on a certain course, he tries to ask other group members for help. One day Peter needs to find someone from this study group to help him finish a Java project. He does not know exactly who is most competent on Java but he vaguely knows who has worked with it and who has competency on C++, another object-oriented programming language. Peter hesitates whom should he seek for help."*

To help people like "Peter" to find a competent colleague for collaboration, this thesis focuses on modeling, collecting, and analyzing user competencies in a collaborative environment. In return, we try to provide suggestions and help for decision-making on judging long board and short board of an organization, or individuals. To achieve this goal, several pieces of work are needed:

Building or improving a collaborative platform containing a various of collaborative tools. In our laboratory, we are running the project E-MEMORAe 2.0 which is conceived and developed to facilitate organizational learning and knowledge capitalization by proposing to associate: 1) Knowledge engineering and educational engineering: support of capitalization; 2) Semantic Web: support of sharing and interoperability; 3) Web 2.0 technologies: support of the social process. E-MEMORAe 2.0 can manage the fields of expertise of the organization and favor collaboration. For the purpose of defining, structuring and capitalizing explicit knowledge, the learning organizational memory is structured by means of ontologies that define knowledge within the organization on this platform. Generally, on this platform, a user can:

- Manage users and user groups (transactions reserved to the administrator);
- Manage memories, private spaces and group workspaces: these spaces associated with the memories to which the user has access are simultaneously visible, and it is easy to transfer content from one space to another;
- Access knowledge map (ontology) and content (resources) based on the active shared space: i.e., individual, group, and organizational spaces;
- Add and share resources, e.g., PDFs or images;
- View and navigate through the concept map. Concept maps are graphical tools for organizing and representing knowledge. In a concept map, concepts are presented by nodes. Relationships between concepts are indicated by a connecting line linking two concepts;
- Annotate concepts and resources;
- Use concepts and individuals of the knowledge map to index resources;
- Collaborate by means of Web 2.0 tools to support informal communication and spontaneous production of knowledge, e.g., semantic Wiki, chat or forum;
- Manage each user's or group's entry points (a set of concepts that represent a particular interest for the user or the group): via the interface, the user can directly access ontological concepts of his/her choice.

Users gain and show their competencies when using collaborative tools in the platform. Apart from the above features, we try to complement the platform with other features that not only facilitate user collaboration, but also

help measure user competencies.

Building a model of competency. As said above, the assessment of competencies is limited to its intangibility, lack of neutrality, and subjectivity. Enlightened by the work of Li (2013) on modeling traces by means of semantics, we propose a semantic explanation on competency and competency measurement.

Adapting mathematical methods from other domains. A final step of any assessment work is to quantify what we observe by specific comparable numbers. With the modeled competency collected from the platform, one must adapt some mathematical methods to our scenario.

1.2 Our Approaches and Contributions

To respond to the scientific problems stated in Section 1.1, we propose a solution including three parts:

Complementing the MEMORAe approach. Apart from the above features, we propose to complement types of resources of the platform with a Community-based Question Answering Service (CQA) and a voting system.

Among all the collaborative tools, the Community-based Question Answering Service (CQA) is the one flourishing recently. CQA sites, such as Yahoo! Answers [1], Baidu Knows [2], as well as more social-oriented newcomers such as Zhihu [3] and Quora [4], have gained substantial popularity over the recent years, effectively filling a niche left by the mainstream Web search engines. CQA is an important type of resource both for collaborating and sharing information, and applying user competencies. Users vote according to the relevance and suitability of a resource on a subject in a group. Even voted by the same user concerning an identical subject, the results can also vary for different sharing space, as the viewers in different groups has a various level of cognition. Moreover, each sharing space may have different concept focus, which also differentiates the vote.

Building a semantic model of competency. We integrate the concepts of competency into the existing semantic model of the MEMORAe approach so that applying knowledge by the form of activities, activities based on resources, and competencies inferred by realizing activities have an organic integration.

[1] https://answers.yahoo.com/
[2] https://zhidao.baidu.com/
[3] http://www.zhihu.com/
[4] https://www.quora.com/

Adapting a variety of mathematical methods from several domains. We apply TF-IDF method from information retrieval, Bayes Classifier and Logistic Regression from machine learning to our case for the measurement of utility of competency. We carry out experiments on a large-scale dataset- "A4 - Yahoo Data Targeting User Modeling, Version 1.0" which contains a set of user profiles and their interests generated from several months of user activities at Yahoo webpages from Yahoo Webscope Program [5].

Proposing a competency recommender system. Typical recommender systems propose items to potential buyers. However, adapting this method can also help recommending competent people to the collaboration group with a certain need. We propose a competency recommender system based on MEMORAe approach which integrates a semantic model of competency and the mathematical solutions.

1.3 Dissertation Organization

The rest of this thesis is organized as follows:

Chapter 2 Literature Review: This chapter is dedicated to the literature review. We start by giving some background of the concept *"collaboration"* and how an informational environment helps collaboration. We also introduce an important type of collaborative environment service where we collect user traces: Community Question Answering Service. Then we focus on the definition and modeling of traces and collaborative traces in the information science area. The fourth part is mainly about competency classification and competency evaluation. Finally, we discuss the state of the art of recommender system.

Chapter 3 Our Competency Recommender System: In this chapter, to answer the needs of a competency recommender system (CRS), two main parts of our work are completed. Firstly, a model that is capable of representing traces and competency is needed. We present our work for modeling traces and competencies. This work is based on the MEMORAe approach and realized on E-MEMORAe-core 2 platform. Secondly, we apply mathematical methods (TF-IDF, Bayes Classifier, and Logistic Regression) to measure and capitalize what we represent from the model and return recommendations accordingly.

Chapter 4 Experiments and Evaluation: In order to compare our methods, we apply them to a dataset prepared from real life and available for non-commercial use by academics and scientists. With the results we discuss and conclude how the scenario of each method best fits the balance of efficiency vs. accuracy. In this chapter we introduce the dataset on which we test our proposition. Then we apply each method to the dataset accordingly. Based on the results, we discuss advantages and disadvantages of each method and report the scenarios that each method fits best.

[5] Yahoo Webscope Program http://webscope.sandbox.yahoo.com/

Chapter 5 Prototypes: In this chapter we apply our method including competency model on a web-based collaborative platform E-MEMORAe2.0. Firstly, we introduce our prototype: MEMORAe-CRS Web Application based on E-MEMORAe approach. Several collaborative tools will be explained with some explicit figures. Then a usage scenario will be presented in detail to show the results of our recommender system.

Chapter 6 Conclusion, Perspectives and Future Work: In this chapter, we conclude our work and give perspectives for future work.

Chapter 7 Publications: Publications related to this work are presented in this chapter.

Chapter 2

Literature Review

2.1 Introduction

Humans, or more precisely, almost all the organisms could not live alone without any interactions with other species (Thompson, Nuismer, and Gomulkiewicz, 2002). In human history, collaboration is important in domains like business, education, entertainment and even in wartime. It is a kind of group work pattern that unites every member's characteristics and competencies. To reach a better collaboration outcome, we should evaluate members' competencies to better allocate tasks and resources. Estimating a person's competency is not only necessary, but also crucial for collaboration. Types of communicating and collaborative tools, like computers or smartphones, allow people to work together regardless of distance and time. Such tools can easily record people behavior and activities based on which we may infer a person's previous participation and to reasonably assess his/her competency.

In this chapter, we also introduce a type of popular knowledge exchange service: Community Question Answering (CQA) Service. In CQA service, users not only propose and respond to questions but also vote for the answers. CQA is an important source of traces for evaluating a person's competency.

Finally we introduce the recommender system. Typical recommender systems propose items to potential buyers. However, adapting this method can also help recommending competent people to the collaboration group concerning a certain need.

The following part of this chapter is organized as follows: Section 2.2 briefly provides background of the concept *"collaboration"* and how an informational environment helps collaboration. We also introduce an important type of collaborative environment service where we collect user traces: Community Question Answering Service. Section 2.3 focuses on the definition and modeling of traces and collaborative traces in the information science area. Section 2.4 is mainly about competency classification and competency evaluation. Finally in Section 2.5 we discuss the state of the art of recommender systems and make a conclusion at the end of this chapter.

2.2 Collaboration and Collaborative Working Environment

2.2.1 Definition of Collaboration

Collaboration means the action of working with someone to produce something[1]. This word originated from the French word *collaboration* [2]. It was composed by a Late Latin noun "collaboratus" plus the French part "-ion". The origin issued from the Late Latin verb "collabōrāre" that is formed by two terms: "col-" (one form of "con-": with, together or joint) and "-labōrāre"(from "labor": work, toil). In short, "collaborate" initial signification is "work together" (Li, 2013).

> As its Latin roots suggest and reduced to its simplest definition, collaboration means "to work together." The search for a more comprehensive definition leads to a myriad of possibilities each having something to offer and none being entirely satisfactory on its own. These range from the academic ("a process of joint decision making among key stakeholders of a problem domain about the future of that domain") to the esoteric ("an interactive process having a shared transmutational purpose") (London, 1995).

One of the more durable and widely-cited definitions is from (Gray, 1989)

> [Collaboration is] a process through which parties who see different aspects of a problem can constructively explore their differences and search for solutions that go beyond their own limited vision of what is possible.

In fields as diverse as business, science, recreation, health care, social work, engineering, or governance, collaboration is seen as the way to address problems, add value, and achieve desired outcomes (Martinez-Moyano, 2006). Collaboration might be realized to resolve a neighborhood or environmental dispute. It could be a springboard for economic development in a community or region. Or it could be used to promote greater civic participation and involvement. London also points that the process works best when:

- Different groups or organizations with a vested interest depend on each other in some way;
- Those with a stake in a problem have yet to be identified or organized;
- Some stakeholders have more power or resources than others;
- Those with a vested interest have different levels of expertise and access to information about the issue.

[1] Oxford Dictionnaries Online, 2016, http://www.oxforddictionaries.com/definition/english/collaboration

[2] Collins English Dictionary - Complete & Unabridged 10th Edition, http://www.dictionary.com/browse/collaboration

Thus we could conclude that people participating in a collaboration should have different competencies to deal with different parts of a problem.

2.2.2 Collaborative Working Environment

Collaboration is realized through a collaboration tool. The purpose of a collaboration tool is to support a group of two or more individuals to accomplish a common goal or objective they have set themselves (Lomas, Burke, and Page, 2008). Before the Information Age, people used to collaborate by non-technological tools such as paper, flipchart, Post-it notes or whiteboards. Nowadays complex and web-based collaborative software like Wiki or SharePoint integrated in an agile work environment make us more efficient (Waggener et al., 2009).

Collaboration tools are classified into two categories[3]:

Asynchronous collaboration tools
A collaboration tool is asynchronous when its users are collaborating at a different time:

- **E-mail mailinglists and newsgroups**: E-Mail is the best known asynchronous collaboration tool and the most common used. It offers intuitive features for forwarding messages, creating mailing groups and attaching documents. Furthermore, information can be automatically chronologically sorted and assigned to task or calendar events.
- **Group calendar**: Through group calendars meetings can be scheduled, projects managed and people coordinated. It is a great tool to help you overlook your deliverables and deadlines. A group calendar includes functions such as the detection of conflicting schedules with other people in your team or organization or coordination of meeting times that suit everybody in your team. Besides the positive effects of group calendar there is also controversy about privacy and control that might influence your productivity (Tullio and Mynatt, 2007).
- **Workflow systems**: With workflow systems files or documents can be communicated to the organization by following a strict and organized process. They provide services for routing, development of forms and support for roles. As current workflow systems are controlled from one point, individuals within an organization normally do not have the permission to manage their own processes so far - this should be changed by implementing collaborative planning tools to current workflow systems (Swenson, 1994).
- **Hypertext**: Hypertext technology connects files to each other and makes sure that always the latest version is available to us. When people work on different documents the system automatically updates the information of other people (Kim, 2004).

[3]Wikipedia: https://en.wikipedia.org/wiki/Collaboration_tool

Synchronous collaboration tools
A collaboration tool is synchronous, when its users are collaborating at the same time:

- **Shared whiteboards**: Shared whiteboards give its users the capability to work efficiently on a task through a web-based platform. They can be used for informal discussions and also for communications that need structure, involve drawing or are in general more sophisticated. This might also be very useful in to realize virtual classrooms (Premchaiswadi, Tungkasthan, and Jongsawat, 2010).
- **Video communication systems**: Video communication systems offer two-way or multi-way calling with a live video stream. It can be best compared to a telephone system with an additional visual element (Kandola, 2009).
- **Chat systems**: Chat systems allow people to write and send messages in real-time. They are usually structured in chat rooms which show usernames, number of people, location, discussion topic and more.
- **Decision support systems**: Decision support systems allow groups to manage decision-making. They give you the ability to exchange your brainstorming, analyzing your ideas and even are used for voting (Druzdzel and Flynn, 1999). Decision-making is becoming more and more a core function of modern work. According to studies 50% of organizational decisions fail.
- **Multiplayer games**: Computer games are a good example of how a multi-user situation could look like in the future. They are constantly developed and expanded with features such as chat and video systems (Wendel et al., 2012).

Thanks to the above collaborative tools, people such as e-professionals are supported in a collaborative working environment (CWE) in their individual and cooperative work [4]. The concept of CWE is derived from the idea of virtual work-spaces (Prinz et al., 2006), and is related to the concept of e-work. It extends the traditional concept of the professional to include any type of knowledge worker who intensively uses Information and Communications Technology (ICT) environments and tools (Carreras and Skarmeta, 2006) in their working practices. Typically, a group of e-professionals conduct their collaborative work through the use of collaborative working environments (Ballesteros and Prinz, 2006).

Collaboration takes place when at least two persons communicate and interact to reach a goal. This is done frequently in most business operations and is increasingly the basic modus operandi of the modern business world. To increase value creation and goal achievement it becomes crucial to understand and improve the way people collaborate. As a basis for analyzing e-collaboration, Weiseth et al. (2006) define a framework consisting of collaboration environment, process and support. The collaboration process is performed in the context of a collaboration environment.

[4] Wikipedia: https://en.wikipedia.org/wiki/Collaborative_working_environment

The environment consists of the nature of the task and the organizational setting such as line of business, markets, actors, competencies, organizational structure, corporate information and cultural beliefs. Adopting a structuration theory perspective (Giddens, 1984), the collaboration process is constrained by the preexisting environment but the relationship evolves over time and appropriations will be made both to the environment and the process (Majchrzak et al., 2000b). Collaboration support consists of organizational measures, services and tools. The collaboration process is also constrained by the support and this relationship too will evolve over time and appropriations will be made both to the support and the process. The three elements of collaboration and the structuration process make up the collaboration framework as illustrated in Figure 2.1.

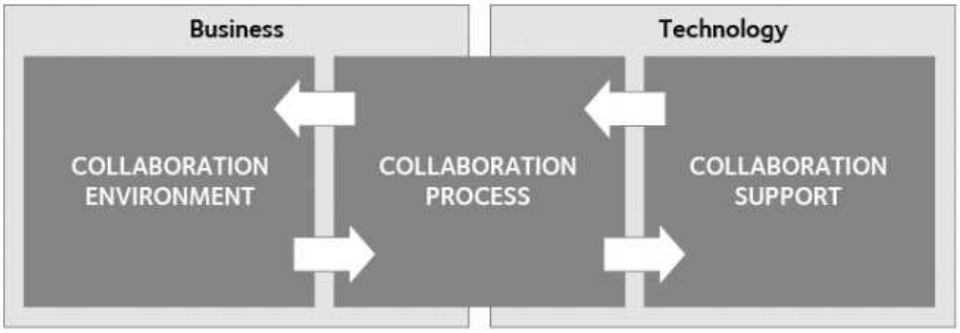


FIGURE 2.1: A collaboration framework (Weiseth et al., 2006).

This framework illustrates that a collaboration framework consists of three basic elements: collaboration environment (i.e. how the nature of task and organizational setting are defined), process (coordination, production and decision-making) and support (organizational measures, services and tools). Successful collaboration requires appropriate management of all the three elements and the related structuration process.

2.2.3 Community-based Question Answering Service

Among all the collaborative tools, the Community-based Question Answering Service (CQA) is the one flourishing recently. CQA sites, such as Yahoo! Answers [5], Baidu Knows [6], as well as more social-oriented newcomers such as Zhihu [7] and Quora [8], have gained substantial popularity over the recent years, effectively filling a niche left by the mainstream Web search engines. People around the globe resort to community help for a variety of reasons, from lack of proficiency in Web search to seeking an answer "with a human touch" (Liu et al., 2011). Although some of these sites allow for "monetary payments in exchange for answering questions (e.g., JustAnswer, or the now closed Google Answers)", answerers are usually attracted by social

[5] https://answers.yahoo.com/
[6] https://zhidao.baidu.com/
[7] http://www.zhihu.com/
[8] https://www.quora.com/

reward and less tangible incentives, such as reputation or points, as demonstrated by Raban (2009). The CQA communities are mainly volunteer-driven, and their openness and accessibility appeal to millions of users; for example, in 2009 Yahoo! Answers staff claimed 200 million users worldwide and 15 million users visiting daily[9]. Baidu Knows claimed having over 330 million answered questions as of September 2014[10].

The number of questions proposed and attempts to answer them on these sites are tremendous. Thus, this is a hard task both for a user who proposes an answer seeking a satisfying answer, or a user who explores already answered questions looking for helpful information. All CQA sites try to assist users searching information by classifying questions and answers by topics as shown in Figure 2.2.

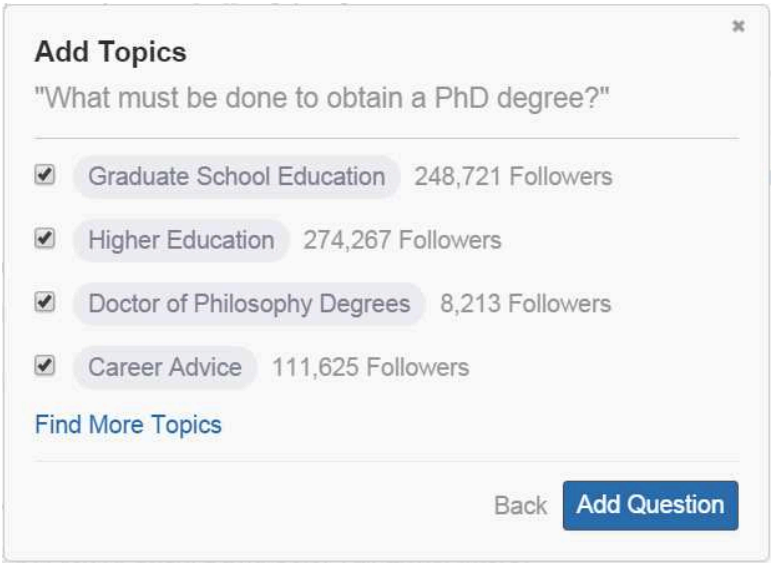


FIGURE 2.2: User is asked to classify the question he/she proposes to a certain topic.[11]

Another big issue CQA sites deal with is the authoritativeness of answerers. On one hand, users can judge directly an answerer's reputation by his/her profile. Unlike sites like Yahoo! Answers, users use their real identities on Quora due to the strict real-names policy. Furthermore, Quora is designed to be a persistent social network based on these real identities. The proponents of real identity use on social networks feel that real identities ensure accountability and safety online. Quora users feel that real identities ensure credibility to answers (Paul, Hong, and Chi, 2012).

[9] http://yanswersblog.com/index.php/archives/2009/12/14/yahoo-answers-hits-200-million-visitors-worldwide/

[10] http://zhidao.baidu.com/

[11] https://www.quora.com/unanswered/What-must-be-done-to-obtain-a-Phd-degree

On the other hand, users also judge the reputation of other users based on their past contributions. Users of Quora mainly performed the following activities (ordered by popularity according to the data captured from Quora[12] as in Figure 2.3):

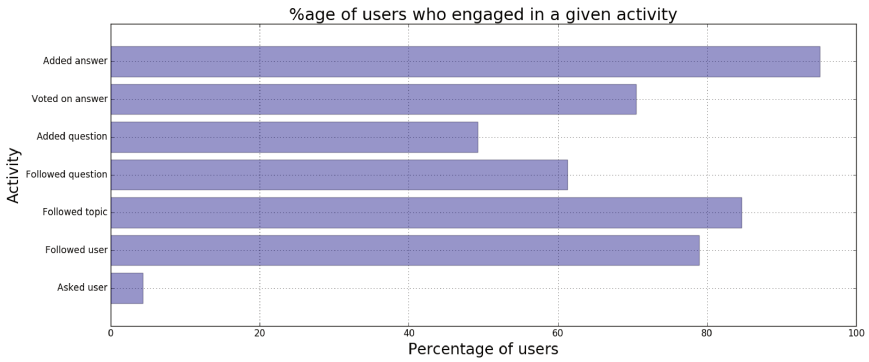


FIGURE 2.3: Percentage of users who participated in different activities on Quora (Paul, Hong, and Chi, 2012).

- **Adding answer**: User answers a question proposed by another user in this community;
- **Following topic**: User subscribes to the content about a topic. He receives an alert when there is an update about the topic, e.g. a new question classified to this topic is lately proposed;
- **Following user**: User *A* subscribes to the content of User *B*. User *A* receives an alert when there is an update about User *B*, e.g. User *B* just added an answer to a question;
- **Voting on answer**: User believes the answer well responded to the question;
- **Following question**: User subscribes to the content about a question. He receives an alert when there is an update about the question, e.g. a new answer is added to respond to this question;
- **Asking question**: User proposes a new question;
- **Being asked**: User *A* is nominated by User *B* to respond to a question User *B* proposed.

From the activity distribution of sample set in Figure 2.3, we can make two preliminary deductions. Firstly the most popular activity in CQA is answering a question which 95.1% users have ever participated. As the number of users responding to questions is numerous, judging the reliability of answers becomes crucial. Secondly, the proportion of asked users is very low (4.3%) indicating that users with competency in a specific domain are few, or at least they are not easily recognized by the public.

[12]Questions Log: http://www.quora.com/logquestions

2.2.4 Discussion

Obviously, collaboration is essential for Peter and his colleagues in the scenario detailed in Section 1.1 to exchange knowledge and improve their study. A good Collaborate Working Environment provides rich forms of collaboration including changing various types of documents, supporting a CQA system and so on.

2.3 Traces and Trace Modeling

2.3.1 Introduction

In the area of computer science, the issue related to the term "trace" or "digital footprint" has aroused more than ever researchers' interests and attentions, but sometimes overflows in the mainstream press refers to social issues (e.g. privacy and data securities). We focus on the "digital footprint" in the information system or more precisely, the trace in the collaborative working environment. We can regard a trace as an influence of the activity in the environment. Definitely, the scope of this environment depends on its context of utilization and "can range from a simple window application configuration until all tools available to the user at a given time" (Prié, 2006). Indeed, a digital trace not only contains the values from the environment properties but also the result of a systematic recording of user's interactions with the environment. According to distinct situations, a trace can be manipulated by the actor for different purposes. This is mainly from a single user's point of view and concentrates on the interactions between a human and an inanimate medium (e.g. a computer) (Lund and Mille, 2009).

Information may be intentionally or unintentionally left behind by the user; with it being either passively or actively collected by other interested parties. Depending on the amount of information left behind, it may be simple for other parties to gather large amounts of information on that individual using simple search engines. Internet footprints are used by interested parties for several reasons; including *cyber-vetting* (Berkelaar, 2014), where interviewers could research applicants based on their online activities. Internet footprints are also used by law enforcement agencies, to provide information that would be unavailable otherwise due to a lack of probable cause. Social networking systems may record activities of individuals, with data becoming a life stream. Such usage of social media and roaming services allow digital tracing data to include individual interests, social groups, behaviors, and location. Such data can be gathered from sensors within devices, and collected and analyzed without user awareness.

In a web-based Collaborative Working Environment (CWE) interactions facilitate sharing information. Almost all the past interactions represent a kind of trace that can be regarded as the user's working experience (Laflaquiere, 2009). According to (Clauzel, Sehaba, and Prié, 2009), an interaction trace is defined as "histories of users' actions collected in real

time from their interaction with the software." In their project "Trace-based Management Systems (TBMS) (systems devoted to the management of modeled traces)," the researchers focused on the personal interaction trace. They mentioned the concept: "Synchronous Collaborative Traces," but do not offer further discussion of its definition. Champin, Prié, and Mille (2003) proposed an approach, MUSETTE (Modelling USEs and Tasks for Tracing Experience), to "capture a user trace conforming to a general use model, describing the objects and relations handled by the user of the computer system." MUSETTE considers the trace as "a task-neutral knowledge base" that can be reused by the system assistants. The researchers of the TRAILS project (Personalized and Collaborative Trails of Digital and Non-Digital Learning Objects) consider the trace in hypermedia as a sequence of actions and use them to identify the overall objective of the user. In a different way, Settouti et al. (2009a) defined a numerical trace as a "trace of the activity of a user who uses a tool to carry out this activity saved on a numerical medium." Zarka et al. (2012) defined an interaction trace as "a record of the actions performed by a user on a system, in other words, a trace is a story of the user's actions, step by step". These considerable research works emphasize the personal aspect, however, they provide little insight for answering the question on "how to share and reuse the users' experiences in a group" and do not provide an effective method for directing the practical design in a CWE.

To conclude, a digital trace can be considered as a set of information recording the user's interactions within the framework of the system. Traces can be considered as a type of resources in the information system. Consequently, it is necessary to build a model to analyze and exploit traces that could assist user's work in many possibilities, e.g. decision-making, recommending, etc.

2.3.2 Trace and Collaboration

A web-based collaborative platform is always available and stable in distinct operation systems and devices, for example: Windows/Linux, tablet/smartphone. Undoubtedly, it can be used as an ideal object to support both personal and collaborative work in a variety of devices. For CWE, almost all the collaborative interactions are taken in the group shared/collaborative workspace. In the early research period, a shared workspace is defined as "a form of an electronic whiteboard" that could assist users in drawing or writing (Whittaker, Geelhoed, and Robinson, 1993). As the most important component of CWE, the group members' collaborative activities are made and taken according to the practical work requirements in the collaborative workspace. This involves several sub-systems of Groupware: communication system (e.g. information sharing and exchanging), coordination system (modeling the interactions between collaborators, the group workflow) and conferencing system (e.g. real-time conferencing, or computer teleconferencing). Besides, knowledge management (e.g. document management, group wikis and task management) and social intercourse models (e.g. the

forum and public wall) are lately discussed and designed within the framework of CWE (Martínez-Carreras et al., 2007).

Obviously, in the sharing workspace, there exists various kinds of interactions based on the group formation. Normally, it relies on the study of group structure that comes from the analysis and modeling of virtual community (Majchrzak et al., 2000a) on the Internet. In order to completely understand how the "collaboration" process generates (e.g. who collaborates with whom and the result) and affects the group members (e.g. the relationships or the interactions in the groups), it is necessary to analyze all kinds of past or finished interactions in the group shared/ collaborative workspace. In consideration of the principal characteristics of collaborative working environment, especially a web-based CWE, a trace not just records the interactions between users and system but also reflects the potential relationships between collaborators.

2.3.3 Trace Modeling

To exploit and reuse traces, a trace model is with no doubt required. In this section we present some important trace models from previous work.

In the research domain of Knowledge-Based System (KBS), Settouti et al. (2009b) defined a trace model as a tuple :

$$Trace = (T, C, R, Att, dom_R, range_R, dom_{Att}, range_{Att}) \tag{2.1}$$

consisting of

- a temporal domain T,
- a finite set C of observed element types (or classes), with a partial order[13] $\leq$ C defined on it,
- a finite set R of relation types, disjoint from C, with a partial order $\leq$ R defined on it,
- a finite set Att of attributes, disjoint from C and R,
- two functions $\text{dom}_R : R \to C$ and $\text{range}_R : R \to C$ defining the domain range of relation types,
- two functions $\text{dom}_{Att} : Att \to C$ and $\text{range}_{Att} : Att \to D$ defining the domain and range of attributes.

Intuitively, a trace model defines a vocabulary for describing traces: how time is represented (T), how observed elements are categorized (C), what relations may exist between observed elements (R), what attributes further describe each observed elements (Att). The domain and range functions constrain the kind of relations and attributes that an observed element of a given type may have. With the domain and range functions, any types

[13] A (non-strict) partial order is a binary relation "$\leq$" over a set P which is reflexive, antisymmetric, and transitive, i.e., which satisfies for all a, b, and c in P: a $\leq$ a (reflexivity); if a $\leq$ b and b $\leq$ a, then a = b (antisymmetry); if a $\leq$ b and b $\leq$ c, then a $\leq$ c (transitivity).

of relations and attributes from the observed element could be constrained. According to this model, they defined a modeled trace as "a sequence of observed elements recorded from a user's interaction and navigation through a specific system." The objective of this model is to support reasoning about traces (represents user's knowledge and experiences of activities with the system) and their interpretation. Additionally, they proposed a language and a framework in order to build a Trace-Based System (TBS) that relies on this model.

Figure 2.4 shows the entire trace model. Each ellipse indicates an observed element type, e.g., a "MouseClick." Each flash with no note indicates an observed element type hierarchy, e.g., "Message" $\leq$ "Application." A flash with a title means an observed relation type, e.g., $dom_R(over)$ ="$MouseClick$" and $range_R(over)$ ="$Window$." An observed attribute is presented by a pair of ellipse and rounded rectangle, e.g., $dom_A(Button)$ ="$MouseClick$" and $range_A(Button)$ ="$String$."

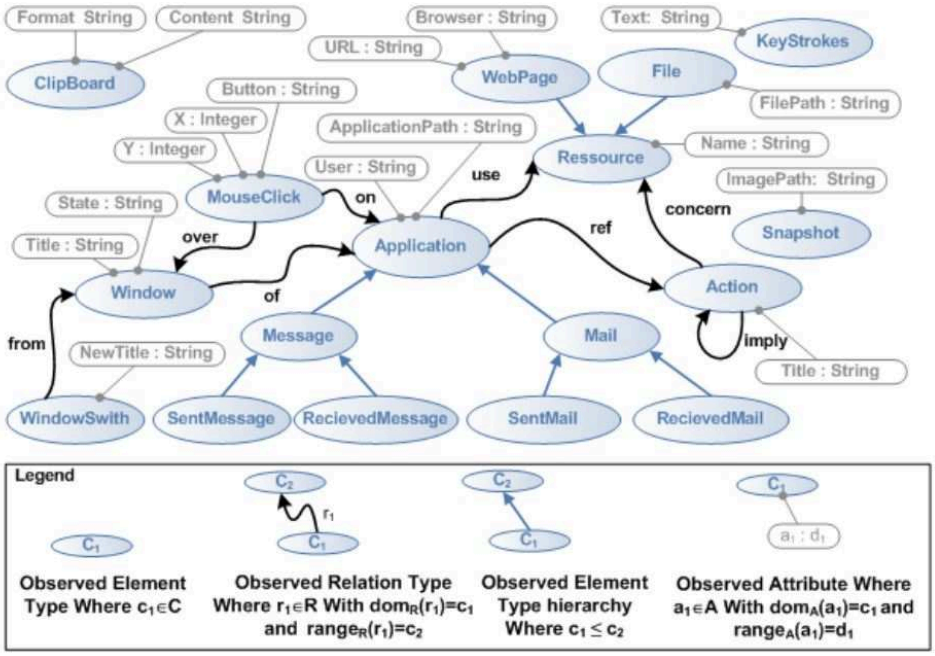


FIGURE 2.4: An example of the trace model proposed in (Settouti et al., 2009b).

To facilitate sharing experience among different users, Sehaba (2012) proposed a model for representing trace. A user who realizes a task acts by actions on one or several physical devices with the help of an interaction language. Formally a trace:

$$Trace =< u, task, o_i > \tag{2.2}$$

- u: the traced user,
- $task$: a description of the task of the user,
- o_i: an observation of the trace. Each o_i is a pair (A_i,Md_i), where

- A_i is an action of the user. For example, $A_i =< Typein\_Date(01, 01, 2015) >$
- Md_i is a modality of interaction such as $Md_i =< d, L >$ where d is a physical device and L is an interaction language.

By this approach, the author assumed that users' properties are sufficiently different, except for the properties related to the modality in question.

Nevertheless, these models above are weak on the group interactions in the domain of Computer-Supported Collaborative Work. For group work, collaboration always depends on shared "Knowledge" but more precisely, it requires collaborative "Experiences." Such "Experience" often comes from the past interactions among the actors themselves or between the actors and the system. Li (2013) proposes a definition of **Collaborative Trace** and introduces a general model that is based on this definition and a group model. Consider the means of collaboration and the correlation of group and individual, naturally a **Collaborative Trace** (CT) that is based on the definition of trace or trace of interaction, it can be defined as follows: **"A collaborative Trace is a set of traces that are produced by a user belonging to a group and is aimed at that group"** (Li, Abel, and Barthès, 2012). The trace is composed of three basic items:

- "Emitters" who leave the trace (the subject);
- "Receivers" who receive the trace or the object of the trace;
- "A property and a corresponding value," i.e. an original trace can generally be considered as a set of information having several properties. For each property, there exists a corresponding value.

With these three factors, for the j^{th} user in CWE, the k^{th} trace can be defined as:

$$Trace_j^k =< Emitter, Receiver, < Property, Value >>, \forall j, k \in N^+ \quad (2.3)$$

To illustrate the model, Li (2013) introduced a simple example. Suppose that in an established CWE, some engineers collaborate within a project. John finds a crucial problem that may be helpful for all the group members. First of all, he sends a mail to the group (every member in this group), then creates a new entry on this issue in group's wikis (every group member can edit and refine it) and his private wiki, and finally shares his solution (a .pdf document) in the group workspace. In the meantime, Tom and Peter from the scenario in the Section 1.1, whose views are similar but different from John's on this problem, both request a video conference with John in the reply email. John receives the emails and agrees on a video conference with Tom and Peter. At last, they obtain a satisfactory answer for this problem in the subgroup meeting and the group wiki is enriched by the new entry. Thus the interactions in the example can be presented as:

$$Trace_{John}^1 =< John, the\_group, < message, ' content' >>$$

and

$$Trace^{1}_{Tom} =< Tom, John, < message,' content' >>$$

and

$$Trace^{2}_{John} =< John, the\_group, < document\_type,' pdf' >>$$

as collaborative traces. In Web-based CWE, users may work in groups. A user may belong to no group, one group or several groups. Let U be the set of users: $U = \{u_j, \forall j \in N^+\}$. Let G be the set of groups: $G = \{g_z, \forall z \in N^+\}$. Each group g_z is defined as a set of users:

$$g_z = \{u_j, \forall z, j \in N^+\} \tag{2.4}$$

A group, g_z, is a set of users, and a subgroup of the set of groups. One can extend the concept of group by considering single users as belonging to a group of one person, namely $|g_z| \geqslant 1$.

A trace is the result of an action done by someone or by a set of individuals and is addressed to a group (which might be a set of one person). A trace is formally defined as:

$$t^{k}_{z,m} =< g_z, g_m, Q >, \forall z, m, k \in N^+ \tag{2.5}$$

where $t^{k}_{z,m}$ is the k^{th} trace emitted by a set of users, g_z (emitters), and sent to a set of users, g_m (receivers), and Q is a set of pairs of a property and a value.

In CWE, a collaborative process needs at least two persons to take a series of actions for a common object. Nevertheless, there exist other kinds of interaction not only among the actors (collaborative or collective activities) but also between actor and machine/system (e.g. private activities). Basically, from the formula definition of trace in CWE, Li (2013) classifies the various traces into four types: Private trace, Collaborative Trace, Collective Trace and Personal Trace.

1. **Private Trace**
   If $z = m, |g_m| = 1$, then the trace is the result of an action done by a user with destination this user. It is a private trace. With the consideration of privacy, additionally, it is decided that a private trace will not be visible by anybody except for its owner, e.g. edit private wikis.
2. **Collective Trace**
   If $|g_z| > 1$ then the trace is the result of a collective action and is defined as a collective trace, i.e. the trace emitted by a group action (e.g. every group member has voted for some candidates).
3. **Collaborative Trace**
   A collaborative trace can be regarded as a type of trace that satisfies

the conditions:

$$|g_z| = 1 \tag{2.6}$$

and

$$|g_m| > 1 \tag{2.7}$$

In accordance with the conditions above, this kind of trace is the result of an action that have been done by a user and addressed to another user or to a group.

4. **Personal Trace**
   If $|g_z| = 1$, then the trace is produced by one of the unique member in the group and aimed at a group. From the distinct cases of "g_m (receivers)", either $|g_m| = 1$ or $|g_m| > 1$. Thus it is concluded that the personal trace is either a private trace or a collaborative trace. This can effortlessly be understood since users' behaviors might be cooperative (social aspect) or private (secluded/unsocial aspect) in a collaborative environment.

2.3.4 Discussion

It is a good news for Peter from the scenario in the Section 1.1 that when he needs help on Java, he will have some solid evidence for his decision instead of vagueness. With the help of a Digital Collaboration Working Environment and a corresponding trace model, all activities realized within the study group will be collected, organized and presented to Peter. In this section we presented two previous work of traces models and they have their own merits. Nevertheless, these models above are weak on the group interactions in the domain of Computer-Supported Collaborative Work. We decide to continue to adapt the trace model of Li as it better responds to the needs of modeling group interactions.

2.4 Competency

Competency (also written as competence) is the ability of an individual to do a job properly. The concept of knowledge and competency are closely related. Once a person holds useful knowledge, he/she is capable to transfer knowledge to solve a problem or to face different situations. Allee (1997) defined competency as knowledge applied and enacted in work practice. Beyou (2003) also agreed to this notion defining competency as a capacity to mobilize efficiently knowledge in a given context. From this point of view, competency can also be defined as a way to put knowledge into practice in a specific context.

According to the model of Rothwell and Kazanas (2011), we can also define the notion of competency by linking with human performance. This model includes the following elements:

- **The situation of work** is the origin of specification of work which puts into practice the competency;
- **The individuals** should have knowledge, skillfulness, attitude for the goal of being capable to act in a given work situation;
- **The response** consists the realized action;
- **The consequence** is the result of action and is determinant if the standard performance is reached.

Different definitions of competency also agree with the three fundamental characteristics (Harzallah and Vernadat, 2002): the resource, the context and the objective.

- One kind of competency is composed of resource that we share by categories. Harzallah and Vernadat concluded there are three principal categories:
  - The competency is something that we acquire and store intellectually. It concerns all that should be learned in an educational system. For example, this category involves theoretical knowledge and procedural knowledge.
  - The "know-how" is related to personal experience and work condition. It is required by putting into practice the knowledge in a specific context.
  - The behavior is an individual characteristic which leads someone to act or react by a certain way in a certain circumstance.
- The context of competency is linked to the environment in which the competency is expressed. It represents the conditions and restrictions in which competency should be mobilized.
- The competency is motivated by an objective. It is characterized by the obtainment of a goal or accomplishment of one or several tasks.

In the following two subsections we will introduce some previous work about competency classification and how competency is evaluated.

2.4.1 Competency Classification

Competency, according to different resource context and objective, can be divided into different categories. Accordingly,[14]

- **Organizational competency:** The mission, vision, values, culture and core competency of the organization that sets the tone and/or context in which the work of the organization is carried out (e.g. customer-driven, risk taking and cutting edge). For example, how we treat the patient is part of the patient's treatment.
- **Core competency:** Capabilities and/or technical expertise unique to an organization, i.e. core competency differentiates an organization from its competition (e.g. the technologies, methodologies, strategies

[14]Wikipedia: https://en.wikipedia.org/wiki/Competence_(human_resources)_note-1

or processes of the organization that create competitive advantage in the marketplace). An organizational core competency is an organization's strategic strength.

- **Technical competency:** Depending on the position, both technical and performance capabilities should be weighed carefully as employment decisions are made. For example, organizations that tend to hire or promote solely on the basis of technical skills, i.e. to the exclusion of other competencies, may experience an increase in performance-related issues (e.g. systems software designs versus relationship management skills).
- **Behavioral competency:** Individual performance competency is more specific than organizational competencies and capabilities. As such, it is important that they be defined in a measurable behavioral context in order to validate applicability and the degree of expertise (e.g. development of talent).
- **Functional competency:** Functional competency is job-specific competency that drives proven high-performance, quality results for a given position. They are often technical or operational in nature (e.g., "backing up a database" is a functional competency) [15].
- **Management competency:** Management competency identifies the specific attributes and capabilities that illustrate an individual's management potential. Unlike leadership characteristics, management characteristics can be learned and developed with the proper training and resources. Competencies in this category should demonstrate pertinent behaviors for effective management to be effective. Such examples as:
  - **Initiative and Creativity**
    Plans work and carries out tasks without detailed instructions; makes constructive suggestions; prepares for problems or opportunities in advance; undertakes additional responsibilities.
  - **Judgment**
    Makes sound decisions; bases decisions on fact rather than emotion; analyzes problems skillfully; uses logic to reach solutions.
  - **Cooperation/Teamwork**
    Works harmoniously with others to get a job done; responds positively to instructions and procedures; able to work well with staff, co-workers, peers and managers; shares critical information with everyone involved in a project.
  - **Quality of Work**
    Maintains high standards despite pressing deadlines; does work right the first time; corrects own errors; regularly produces accurate, thorough, professional work.
  - **Commitment to Safety**
    Understands, encourages and carries out the principles of integrated safety management; completes all required Environment,

[15] Bersin: http://www.bersin.com/Lexicon/details.aspx?id=12840

Safety & Health (ES&H) training; takes personal responsibility for safety (Electrical Engineers, 1999).

- **Support of Diversity**
  Treats all people with respect; values diverse perspectives; participates in diversity training opportunities; provides a supportive work environment for the multicultural workforce.
- **Quantity of Work**
  Produces an appropriate quantity of work; does not get bogged down in unnecessary detail; able to manage multiple projects; organizes and schedules people and tasks.
- **Problem Solving**
  Anticipates problems; sees how a problem and its solution will affect other units; gathers information before making decisions; weighs alternatives against objectives and arrives at reasonable decisions; adapts well to changing priorities, deadlines and directions; works to eliminate all processes which do not add value; is willing to take action, even under pressure, criticism or tight deadlines.
- **Attention to Detail**
  Is alert in a high-risk environment; follows detailed procedures and ensures accuracy in documentation and data; carefully monitors gauges, instruments or processes; .
- **Flexibility**
  Remains open-minded and changes opinions on the basis of new information; performs a wide variety of tasks and changes focus quickly as demands change.
- **Organization**
  Able to manage multiple projects; able to determine project urgency in a practical way; uses goals to guide actions; creates detailed action plans; organizes and schedules people and tasks effectively.
- **Quality Control**
  Establishes high standards and measures; is able to maintain high standards despite pressing deadlines; does work right the first time and inspects work for flaws.
- **Responsiveness to requests for service**
  Responds to requests for service in a timely and thorough manner; does what is necessary to ensure customer satisfaction; prioritizes customer needs; follows up to evaluate customer satisfaction.
- **Innovation**
  Able to challenge conventional practices; adapts established methods for new uses; pursues ongoing system improvement; creates novel solutions to problems.

More concisely, Baugh (2000) distinguished two types of competency:

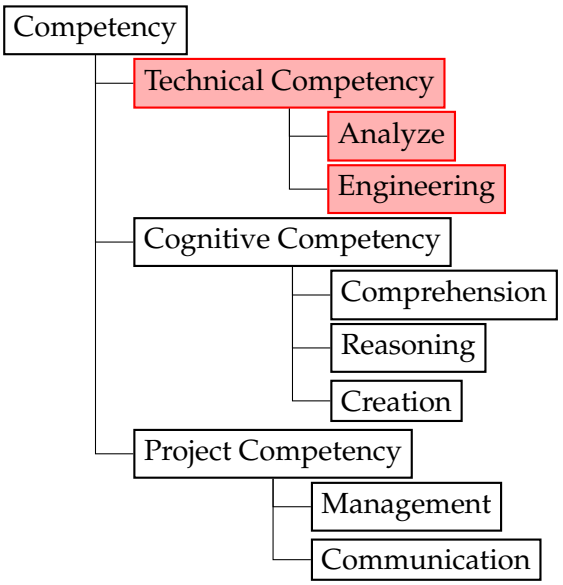


FIGURE 2.5: Extract of ontology of the domain of competency (Vasconcelos, Kimble, and Rocha, 2003).

- **Hard competency**: identifies the intellectual procedures need for the realization of an activity.
- **Soft competency**: is corresponding to personal behaviors, characteristics, for example, the tendency to work with others, leadership, etc.

For the following parts of this thesis when we speak of competency we mainly focus on the "hard competency."

As shown in Figure 2.5, Vasconcelos et al. defined an ontology of competency in order to represent the competencies used and necessary to complete a project within an organization. In this definition three types of competencies are defined:

- **Technical Competency** is the ability to analyze a problem and the engineering competencies. According to previous classification, it belongs to hard competency;
- **Cognitive Competency**: is the comprehension of a document or a problem, the reasoning and the creation of whatsoever document or more concrete object. It belongs to soft competency;
- **Project Competency**: it concerns the management of a group, project, or the aptitude whether to express a question or to transmit knowledge.

The model we propose will adapt this ontology of competency as it corresponds to both what is required for an engineer to work on group on a project, and at the same time represents the ability required for a manager to manage a project.

2.4.2 Competency Management

Competencies are measurable human capabilities that are required for effective work performance demands. Competency analysis and modeling identify those capabilities. The benefits of competency-based management systems include the following (Marrelli, 1998):

- Emphasizing human capital as essential to the organization's prosperity and longevity;
- Moving away from narrowly defined functions and jobs to integrated processes and teamwork;
- Creating the flexibility to quickly adapt to changing customer needs and business conditions through competency-based deployment of employees;
- Creating a culture of continuous learning;
- Substituting lateral growth for career ladders and promotions;
- Providing employees with opportunities to develop and apply new knowledge and skills in exchange for their work and commitment.

Marrelli (1998) observed over 20,000 engineers working activities and their performance in an anonymous aerospace and defense company. After several years of hard work she concluded the methods of identifying competencies:

FOCUS GROUPS
Through guided discussion, groups of individuals who are knowledgeable about the target job roles identify competencies they believe are required for success. The group may include incumbents of the target job roles, managers, and customers. Features of this
method are:

- Enables broad organizational input and thus promotes buy-in;
- Offers moderate validity;
- Can focus on competencies needed in the future;
- Relatively inexpensive for the large amount of data collected.

BEHAVIORAL EVENT INTERVIEWS
(Also called Critical Incident Interviews) Excellent performers are interviewed to identify the behaviors that were critical to their success in challenging situations. The interviewer asks the performers what they did, thought, said, felt and caused to happen. The competencies critical to their success are inferred from the information supplied. Often, average or below-average performers will also be interviewed for comparison. Features of this method are:

- A series of interviews provides an in-depth view of the challenges faced on the job and the competencies required to meet them; offers a high degree of validity;

- The data collected is subjective; the information may not be reliable but a large sample minimizes this problem;
- Extremely time and labor intensive;
- Requires a high degree of analytical ability and experience in competency analysis work to infer the competencies;
- The data may not be broadly accepted du to the small number of people included in the interviews;
- Emphasizes current and past job success factors that may differ from behaviors needed for the future.

INTERVIEWS WITH STAKEHOLDERS
Persons familiar with the target job or job role are individually interviewed to obtain their input on the competencies needed for success. This group can include job incumbents, managers, direct reports, and external and internal customers. Sometimes benchmarking interviews are also conducted with representatives of other successful organizations. Features of this method are:

- Provides for input from a wide range of stakeholders and promotes buy-in;
- Validity can be difficult to determine; the knowledge of the interviewees may vary widely;
- Care must be taken in applying the data collected from other organizations; the information may not be generalizable to a different work environment;
- Time and labor intensive.

SURVEYS
A written or electronic questionnaire is distributed to persons familiar with the target job role including incumbents, managers, direct reports, and customers. Typically the survey lists possible competencies required for the job and the respondents are asked to indicate the importance of each for success in the target job role. Respondents are also asked to add competencies that are not listed. Features of this method are:

- Validity and reliability vary with the selection of the sample of respondents and the quality of questionnaire construction;
- A lot of information can be collected inexpensively;
- Information can be obtained from a large number of geographically dispersed people;
- Facilitates buy-in through wide inclusion.

COMPETENCY MENUS AND DATABASES
Generic databases of competencies found to be important in many different organizations can be purchased from consulting firms and publishers. Some of these are formatted in menus so that the user selects the competencies appropriate for the target job from a list of possibilities. Some of these

databases focus on only one category of competencies, such as leadership competencies, while others cover a wide range of job roles. Features of this method are:

- Validity can be very low due to large differences in work environment, culture and specific job responsibilities among organizations;
- Inexpensive, quick, and easy to use;
- Can be useful as a first step to introduce an organization to competency modeling.

OBSERVATIONS
High performers are observed on the job. The tasks they perform and the actions they take to perform those tasks are recorded. Observations often include asking the performers to explain the reasons for their actions. Comparison samples of average and poor performers are also often included. The competencies needed for successful performance are inferred from the observations. Features of this method are:

- Validity is strong if representative samples are selected;
- Requires a high degree of analytical ability and experience in competency analysis work to infer the competencies;
- Extremely time and labor intensive;
- Buy-in can be low due to the small numbers of persons included in the observations.

Many projects have focused on this area. Competency management system (CMS) is a type of enterprise software used for evaluating and managing human resources. Over the last few years, there has been a push to improve and expand these systems. Many of the earliest efforts were custom developed by organizations such as National Aeronautics and Space Administration (NASA) and the U.S. Coast Guard.

NASA uses its system to anticipate human resources needs and manage recruitment activities. The system is a Web-based tool that allows users to edit their own information and to search for others. And, as of October 2011, it listed as future uses "employee development," "knowledge management," "integration of business processes," and searching for resources based on areas of expertise [16]. NASA publishes its own "Workforce Competency Dictionary," which defines specific categories of skills. For instance, it describes “Partnership & Business Development” as requiring “Knowledge, capabilities and practices associated with the effective targeting and acquisition of external partnerships and business opportunities, including funding opportunities for projects and programs.”

Similarly, the Coast Guard’s system maintains an online dictionary of competencies. The system is more of a set of procedures than a tool. The Coast Guard describes its use of software as being "in support of" the system.2 Its database and software "collects, stores, sorts and reports data required" by the system.

[16]NASA. Competency management system. http://ohcm.gsfc.nasa.gov/cms/home.htm

In this thesis, we actually adapt and expand the method "Observations" of a variety of performers to establish a benchmark for a certain competency. To achieve this goal and to obtain a satisfying validity, firstly, both good performers and poor performers are included as the activities are really diversified in a collaborative working environment. Furthermore, as we take a digital method to record traces, observation has been largely simplified. Thus "Observations" is no longer time and labor intensive. As for the analytical ability in competency analysis work to infer the competencies, we seek help from several mathematical tools which will be introduced in the following chapter.

2.4.3 Discussion

As presented in this section, competency is evaluated, according to different models, by a variety of characteristics and elements. It is also related to how a person is motivated and how the context he/she performs the competency is. Frankly speaking, the evaluation of competency itself is subjective. Thus we need to understand competency comprehensively. Some of the characteristics include quality of work, which can derive from evaluation by group members; quantity of work which we can conclude by the frequency of activities. As for competency management methods, previous work propose that we either make a survey to all members or to have an interview with the stakeholders. We prefer to observe the facts during the interaction. With the help of good competency management, "Peter" in our scenario can make an easier decision to reach for assistance.

2.5 Recommender Systems

Typically, recommender systems are a subclass of information filtering system that seek to predict the "rating" or "preference" that a user would give to an item (Ricci, Rokach, and Shapira, 2011). It is nowadays an active research topic in the data mining and machine learning fields. Many top-level conferences address recommender systems research including RecSys (The ACM Conference Series on Recommender Systems)[17], SIGIR (The ACM SIGIR Conference on Research and Development in Information Retrieval) [18], and KDD (The ACM SIGKDD Conference on Knowledge Discovery and Data Mining)[19]. With no doubt, in the field of business, recommender systems have also become extremely common in recent years. Recommender systems are changing from novelties used by a few e-commerce sites, to serious business tools that are re-shaping the world of e-commerce. Many of the largest commerce Web sites are already using recommender systems to help their customers find products to purchase. A recommender system learns from a customer and recommends products that he/she will find most valuable from among the available products.

[17] https://recsys.acm.org/
[18] http://sigir.org/
[19] http://www.kdd.org/

Recommender systems are not merely limited to the usage of providing a shopping list. They are based on user behaviors, watching people in their natural environment and making design decisions directly on the results. In the field of social network, it also facilitates and ameliorates users experience. Social Recommender Systems (SRSs) aim at alleviating information overload over social media users by presenting the most attractive and relevant content. SRSs also aim at increasing adoption, engagement, and participation of new and existing users of social media sites. Recommendations of content (blogs, wikis, etc.) (Guy et al., 2010), tags (Sigurbjörnsson and Van Zwol, 2008), people (Guy, Ronen, and Wilcox, 2009), and communities (Chen et al., 2009) often use personalization techniques adapted to the needs and interests of the individual user, or a set of users (Jannach et al., 2010). To conclude, no matter in what domain recommender systems are applied, they are always based on three things: users activities, features of "entities" to recommend to, and features of "items" to be recommended. Here "entities" and "items" are not limited to the pair of "customers" &"books" or "audience" &"films."

In the rest of this section, we introduce a typology of recommender systems. Generally, there are four types of recommender system: collaborative filtering (CF), content-based filtering, knowledge-based recommender systems and hybrid recommender systems. For each type, we review previous work, advantages and technical problems they may encounter.

2.5.1 Collaborative Filtering

Collaborative filtering (CF) is the process of filtering information or patterns using techniques which involves collaboration among various agents, viewpoints, data sources, etc. (Terveen and Hill, 2001). Applications of CF involve very large data sets. CF methods have been applied to many different kinds of data including: financial data, such as financial service institutions that integrate many financial sources; monitoring and sensing data, such as in mineral exploration, environmental sensing over large areas or multiple sensors; or in e-commerce and web applications where the focus is on user data, etc. CF can be used for making automatic predictions about the interests of a user by collecting preference or taste information from many users by means of collaboration. CF approach assumes that if a person X has the same opinion as a person Y on an issue, X is more likely to have Y's opinion on a different issue 'a' than to have the opinion on 'a' of a person chosen randomly. For example, a CF recommender system for laptop tastes could make predictions about which laptop brand a user should like given a partial list of that user's tastes (likes or dislikes).

The goal of this approach is trying to predict the opinion a user will hold on different items and to recommend the "best" item to each user based on previous opinions and those of similar users (Negre, 2016). Typical workflow of a CF system is as the following:

- a quantity of users' preference are registered;

TABLE 2.1: An example representing users and their interest of subjects.

u(c,i)	Python	Java	C++	VBA
Peter		1	0	1
Marie	1	1	0	1
Glory	0	1	1	

- a subgroup of users are recognized whose preference are similar with that of the user that is looking for recommendation;
- an average preference of this subgroup is calculated;
- a preference function is used to for recommending opinions/items to the user looking for recommendation.

The similarity needs to be defined on primarily two different aspects: similarity between items (Item-to-Item similarity) and similarity between users (User-to-User similarity).

Table 2.1 shows an example of users' interest on different subjects. Each value (1/0) indicates that user c is interested in item i or not. In the Item-to-Item approach, recommendation is made by looking for items that interest many users. Peter and Marie are both interested in "Java" and "VBA". This indicates that generally users who have interest in "Java" also have interest in "VBA". Thus "VBA" can also appeal to Glory as he has interest in "Java."

In the User-to-User approach, recommendation is made by looking for users holding the same opinions. Peter and Marie are both interested in "Java" and "VBA," and they are not interested in "C++." This indicates that these two users have the same interests. Thus "Python" can be a good recommendation to Peter as it is of interest to Marie.

The CF recommender system has the following advantages (Negre, 2016):

- It uses the score of other users to evaluate a current user's interest;
- It tries to find users or group of users that have corresponding interests with current user;
- The more there are users and scores, the better the recommending result is.

However it also has the following disadvantages:

- Finding users or group of users having mutual interests is difficult;
- The recommender system works badly when the scores of users on items are sparse (large amount of users and items, each user merely scores a few items);
- There exists a "cold-start" problem meaning that when a user starts to use the recommender system, his/her interest is unknown. Likewise,

if a new item is included into the system, no users have ever given it a score.

For a collaborative filtering, users where data is collected from do not really collaborate with each other. It is the data that describes user preferences and behaviors that actually "collaborate," namely be compared and exploited by different algorithms.

2.5.2 Content-based Filtering

Content-based filtering methods are based on description of items and a profile of the user's preference (Brusilovsky and Maybury, 2002). These algorithms try to recommend items that are similar to those that a user liked in the past or is examining in the present. In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended. This approach has its roots in information retrieval research. Basically, these methods use an item profile characterizing items within the system. The system creates a content-based profile of users based on a weighted vector of item features. The weights denote the importance of each feature to the user and can be computed from individually rated content vectors using a variety of techniques. Simple approaches use the average values of the rated item vector while other sophisticated methods use machine learning techniques such as Bayesian Classifiers, cluster analysis, decision trees, and artificial neural networks in order to estimate the probability that the user is going to favor the item.

Compared to collaborative filtering, content-based method exploit solely ratings provided by the active user to build his/her own profile. Instead, collaborative filtering methods need ratings from other users in order to find the 'nearest neighbors' of the active user, i.e. users that have similar tastes since they rated the same items similarly. Then, only the items that are most liked by the neighbors of the active user will be recommended. Thus a content-based method is relatively user independent.

The standard of recommended items proposed by content-based methods has clearer explanations on how the recommender system works, since they can be provided by explicitly listing content features or descriptions. Those features are indicators to consult in order to decide whether to trust a recommendation. Conversely, collaborative systems are different since the only explanation for an item recommendation is that unknown users with similar tastes were in favor of that item.

Content-based methods are capable of recommending items not yet rated by any user. As a consequence, they do not suffer from the "cold start" problem, which affects collaborative recommender which rely solely on users' preferences to make recommendations. Therefore, until the new item is rated by a substantial number of users, the system would not be able to recommend it.

As for shortcomings, a key issue with content-based filtering is whether the system is able to learn user preferences from user's actions regarding one content source and use them across other content types. Content-based recommenders have no inherent method for finding something unexpected. The system suggests items whose scores are high when matched against the user profile; hence the user is going to be recommended items similar to those already rated. This drawback is also called serendipity problem to highlight the tendency of the content-based systems to produce recommendations with a limited degree of novelty. Thus it is probable that an online store will recommend you another model of dust cleaner only because you have recently brought one.

2.5.3 Knowledge-based Recommender Systems

The knowledge-based recommender systems are a specific type of recommender system that are based on explicit knowledge about the item assortment, user preferences, and recommendation criteria (i.e., which item should be recommended in which context?) (Burke, 1999). These systems are applied in scenarios where alternative approaches such as collaborative filtering and content-based filtering cannot be applied. A major strength of knowledge-based recommender systems is the non-existence of cold-start (ramp-up) problems since its recommendations do not depend on a base of user ratings. A corresponding drawback are potential knowledge acquisition bottlenecks triggered by the need of defining recommendation knowledge in an explicit fashion.

More precisely, there exist two types of approaches on knowledge-based recommender system: case-based approach (Burke, 2000; Mirzadeh, Ricci, and Bansal, 2005; Ricci and Nguyen, 2007; Smyth et al., 2004) and constraint-based approach (Felfernig et al., 2007; Thompson, Goker, and Langley, 2004). Case-based approach treats recommendation as a problem of evaluating similarity. Looking for an item that is the most similar to what the current user considers to be desirable needs knowledge and preoccupation of a domain. Constraint-based recommendation requires the explicit definition of questions, product properties and constraints. These elements constitute a recommender knowledge base which can be represented as a constraint network (Felfernig and Burke, 2008).

2.5.4 Hybrid Recommender Systems

Hybrid recommender systems are based on the combination of collaborative filtering and content-based filtering. These overcome the limitations of native CF approaches. It improves the prediction performance. Importantly, it overcomes the CF problems such as sparsity and cold-start problem. Given two or more basic recommender system techniques, several ways have been proposed for combining them to create a new hybrid system (Burke, 2007). However, they have increased complexity and are expensive to implementation (Ghazanfar, Prügel-Bennett, and Szedmak,

2012). Usually most of the commercial recommender systems are hybrid, for example, Google news recommender system (Das et al., 2007).

2.5.5 Discussion

To conclude, a typical recommender system recommends items to users or customers. But we can borrow this method to propose a hybrid competency recommender system. For one part of the collaborative filtering, we can apply user traces on different concepts and user rates on resources instead of typical previous behavioral history and rating. For the other part of content-based approach, semantic relations between concepts and user profiles are to be considered instead of item features.

2.6 Chapter Summary

In this chapter, firstly we introduced a variety of types of collaborative working environment and applications and how they help collaboration. This is the environment which the proposal of this thesis is based on. Then we had a retrospect of traces and trace modeling. We emphasized the trace modeling proposed by Li (2013), which we will adopt in our system. Afterwards the notion was introduced. We reviewed the state-of-the-art of previous work on competency management and pointed out the method we will apply for competency modeling. Finally came the part of recommender systems distinguished respectively by four different types. In the following chapter, we state the main part of this thesis.

Chapter 3

Our Competency Recommender System

To respond to the need of a competency recommender system (CRS), two main pieces of work are required. Firstly, we need a model that is capable of representing traces and competencies. Secondly, we apply mathematical methods to measure and capitalize what we represent from the model and return recommendations accordingly.

The chapter is organized as follows: Section 3.1 presents our work for modeling traces and competencies. This work is based on the MEMORAe approach and realized on MEMORAe-core 2 platform. In Section 3.2, based on the semantic model we will then present how we apply mathematical methods (TF-IDF, Bayes Classifier, and Logistic Regression) to analyze traces of our scenario accordingly.

3.1 MEMORAe-CRS Ontology Model and Its modules

In chapter 2, we present the necessities to record and analyze users' traces from collaborative working environment (CWE). The work in this section is modeling traces and competencies accordingly in a CWE. This model covers all collaboration needs that were identified in the introduction, namely:

- Be based on semantic web standards: Semantic web models plays an important role for supporting collaboration;
- Represent and distinguish various types of activities in the collaboration, as different types of activities may have different importance for reasoning competency;
- Represent competency by ontology and integrate this ontology with other concepts to make the reasoning on competency possible.

Finally we develop a digital tool that has features based on a model conform to the above points, and offer a competency recommender system (CRS) in order to facilitate and improve collaboration. MEMORAe is an approach applied in a CWE. Based on this approach, we propose a MEMORAe-CRS ontology model that meets our requirement to realize the proposal. In the next section, we give a brief description of the MEMORAe approach then

we justify the choice of such an approach.

3.1.1 The MEMORAe Approach

3.1.1.1 A Brief Introduction of the MEMORAe Approach

The MEMORAe approach is a combination of a semantic model and a web platform sharing the same name to manage heterogeneous resources of knowledge in an organization. Semantic model is a conceptual data model in which semantic information is included. This means that the model describes the meaning of its instances. Such a semantic data model is an abstraction that defines how the stored symbols (the instance data) relate to the real world. Semantic model enables interpreting the semantic expressions in multiple databases and messages and that different databases can be treated as if they are one distributed database. Such inter-operation of databases enables verification and management of the consistency as well as a combination of their content.

Within MEMORAe approach, MEMORAe-core 2 (mc2) is a semantic model built using OWL (Ontology Web Language) and based on semantic web standards (FOAF, SIOC, BIBO, etc.). Regarding the typology of ontologies, MEMORAe-core 2 contains a core ontology representing collaboration in organizations. The model focuses on modeling resource sharing and indexing between individuals and groups of individuals within an organization (Deparis, 2013). There are two main aspects in MEMORAe-core 2 model:

- Modeling individuals and groups of individuals: MEMORAe-core 2 regards an organization as a set of users belonging to groups. Each group has its own sharing space in which users can share or access resources. All resources are indexed by an index key which is visible to a certain sharing space;
- Modeling resources: resources in MEMORAe-core 2 are defined as "vectors of information." The resources are divided into two main categories: simple and composed. A document, an agent, can be direct examples of simple resources. Composed resources are composed of other resources. Each resource is indexed by an index key which is visible for a certain sharing space. The model supports documentary resources (e.g. documents) and social resources (e.g. chat, forum, wiki).

E-MEMORAe web platform is based on MEMORAe-core 2 model. The platform is developed using web 2.0 technologies. Based on MEMORAe-core 2, the platform is dedicated to collaboration and resource sharing between members of an organization.

MEMORAe approach along with its model and web platform seems a good for a competency recommender system for two reasons. Firstly, MEMORAe model is a semantic model based on semantic web standards. The model

was re-designed using a modular approach. This would facilitate removing or adding new information resource types as being modules. This allows semantic expression of different databases according to different collaboration group and context without changing the information model. Secondly, MEMORAe-core 2 model represents collaboration and information sharing within an organization, which facilitates recording traces of user interactions. In his thesis Atrash Atrash (2015) improves the MEMORAe approach taking into account small and micro business needs to support organizational learning. My work is also based on this approach while we focus on the evaluation of users' competency based on traces. Nevertheless, the MEMORAe-core 2 model does not take into consideration the needs for modeling and recommending competency. In addition, the previous activity module lacks specifications of interactive activities within the platform. As a consequence, the model and the platform should be adapted to answer the needs of building a recommender system as identified in the introduction. The idea of adding votes as a type of resource and the representation of competency lead us to rethink the development of MEMORAe-core 2 model. We continue using the modular approach if a standard ontology corresponds to our need already exists.

3.1.1.2 Modularity of Ontology

The modularity approach is primarily used in the software engineering domain. Modules in software engineering are independent and reusable units. Recently, the use of this approach is more and more adopted for modeling semantic web ontology models. The modularity of ontologies is considered a crucial task to enable ontology reuse on the semantic web. Ontology modularization main objective is to structure and organize ontologies. Pathak, Johnson, and Chute (2009) define the module as being a subset of a "whole" that makes sense (i.e., is not an arbitrary subset randomly built) and can somehow exist separated from the whole. An ontology module is therefore (according to (Pathak, Johnson, and Chute, 2009)) a sub-ontology of a "whole" that makes sense. Doran, Tamma, and Iannone (2007) define the ontology module as being "a reusable component of a larger or more complex ontology, which is self-contained but bears a definite association to other ontology modules, including the original ontology." The connections in this case belong to source module.

There are two main approaches to construct modular ontologies. The first is ontology *decomposition*. In this case, there is an integrated ontology and the objective is to extract modules from this ontology to support a particular use case. The second is ontology *composition*. In this case, each ontology is independently developed and then integrated to the main ontology in a coherent and uniform manner.

The method proposed in this work is based on the ontology composition approach. In this method, there is a generic ontology which is considered as a base ontology. The generic ontology is called the "Abstract Ontology" i.e. an ontology that should be completed by one or more modules (according

to the need) in order to be used. We use owl:import to import the modules. However, we consider that owl:import is not enough to integrate a module. We must define an integration method, i.e., the properties and the axioms that we must add to the ontology when a module is integrated. The method is the following:

- We duplicate the "Abstract Ontology" and keep its namespace. We obtain an "Implementing Ontology (IO)" i.e., an ontology which is ready to import modules;
- We import a module (M) (without keeping its namespace) to IO;
- We add the required elements and modify the required axioms.

If a module M_i needs to integrate another module M_j, we follow the same method beginning with the M_j module as the first abstract ontology.

3.1.1.3 Existing Ontology Modules in MEMORAe-core 2

The modular ontology MEMORAe-core 2 is the modular version of MEMORAe-core 2 model. Semantic web standards are integrated as being modules. MEMORAe core 2 modular ontology integrates 4 modules: FOAF, SIOC, BIBO, VCARD. FOAF and SIOC modules are the main parts for modeling user organizations. The integration of a module is based on the method already presented. The examples in the following sections will be based on the scenario presented in Section 1.1:

"Peter is a college student and he joins a study group composed of his peers. Ordinarily, they discuss and share information concerning the courses they take. When Peter meets with difficulties on a certain issue, he tends to ask for help from group members."

Using the *composition* method, MEMORAe-core 2 integrates FOAF and SIOC modules as follows

- `mc2:Agent rdfs:subClassOf foaf:Agent`
- `mc2:Person rdfs:subClassOf foaf:Person`
- `mc2:Person rdfs:subClassOf mc2:Agent`
- `mc2:Organization rdfs:subClassOf foaf:Organization`
- `mc2:Organization rdfs:subClassOf mc2:Agent`
- `mc2:Group rdfs:subClassOf foaf:Group`
- `mc2:Group rdfs:subClassOf mc2:Agent`

We can add the knowledge base (`kb` as a prefix) in the following triples. Looking back to the scenario we mentioned in this section, we create `Peter` as a person agent:

- `kb:peter a mc2:Person`
- `kb:peter a foaf:Person (BY INFERENCE)`
- `kb:peter foaf:firstName "Peter"`
- `kb:peter foaf:lastName "Pan"`

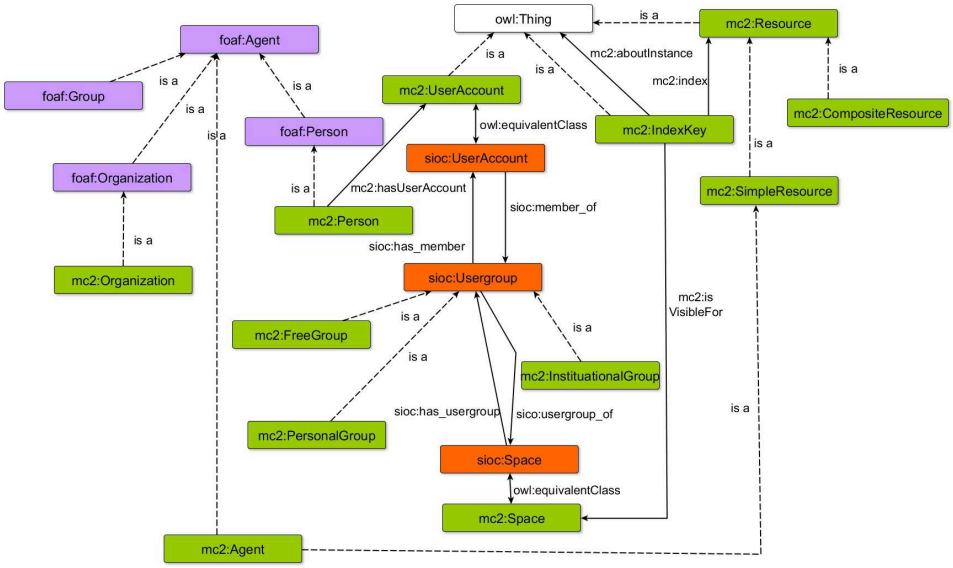


FIGURE 3.1: MEMORAe-core 2 with its modules (partial).

To create a group agent for the studying group:
- `kb:studyGroup a mc2:Group`
- `kb:studyGroup a foaf:Group (BY INFERENCE)`

To create an organization agent:
- `kb:utc a mc2:Organization`
- `kb:utc a foaf:Organization (BY INFERENCE)`
- `mc2:UserAccount owl:equivalentClass sioc:UserAccount`
- `mc2:Space owl:equivalentClass sioc:Space`
- `mc2:InstituationalGroup rdfs:subClassOf sioc:Usergroup`
- `mc2:FreeGroup rdfs:subClassOf sioc:Usergroup`
- `mc2:PersonalGroup rdfs:subClassOf sioc:Usergroup`

To integrate, we can add the knowledge base (`kb` as a prefix) in the following triples. We create an instance of person `kb:peter` for Peter and assign a user account to this person:

- `kb:peterAccount a mc2:UserAccount`
- `kb:peterAccount a sioc:UserAccount (BY INFERENCE)`
- `kb:peter mc2:hasUserAccount kb:peterAccount`

To create a personal group and sharing space for Peter:
- `kb:peterGroup a mc2:PersonalGroup`
- `kb:peterGroup a sioc:UserGroup (BY INFERENCE)`
- `kb:peterSpaceOfGroup a mc2:Space`
- `kb:peterSpaceOfGroup a sioc:Space (BY INFERENCE)`

To assign the user account of Peter to his personal group:
- `kb:peterAccount sioc:member_of kb:peterGroup`

```
- kb:peterGroup sioc:has_member kb:peterAccount
```

To assign the sharing space to the group:

```
- kb:peterGroup sioc:usergroup_of kb:peterSpaceOfGroup
- kb:peterSpaceOfGroup sioc:has_usergroup kb:peterGroup
```

Every person has a personal group which holds a personal sharing space. Peter can use this group to add and review personal information resources. Peter also belongs to another group which is an Institutional Group (which has all the members of the study group).

To create the institutional group and its sharing space:

```
- kb:studyGroup a mc2:InstituationalGroup
- kb:studyGroup a sioc:UserGroup (BY INFERENCE)
- kb:studySpaceOfGroup a mc2:Space
- kb:studySpaceOfGroup a sioc:Space (BY INFERENCE)
```

To assign the sharing space to the group:

```
- kb:studyGroup sioc:usergroup_of kb:studySpaceOfGroup
- kb:studySpaceOfGroup sioc:has_usergroup kb:studyGroup
```

To assign the user account of Peter to the study group:

```
- kb:peterAccount sioc:member_of kb:enterpriseGroup
- kb:studyGroup sioc:has_member kb:peterAccount
```

So Peter belongs to two groups, one personal and the other shared with his study group colleagues. According to his needs, he could choose between the two spaces in which to share his information resources.

3.1.2 MEMORAe-CRS Ontology and Modules

MEMORAe-CRS ontology is an ontology model for knowledge capitalization within collaboration group to propose a Competency Recommender System (CRS). The ontology takes into account the results of the discussion presented in chapter 1. The model should allow the following:

- Identify various types of interaction **activities** in the digital platform;
- Organize users' **votes** on different resources;
- Present and reason users' **competency**.

MEMORAe-CRS ontology is built from the modular ontology MEMORAe-core 2 as a base. The added modules are permits to respond to CRS need.

3.1.2.1 Activity Module

Previously in the model MEMORAe-core 2, the activity module represents the processes and procedures done over time. This old module focused only on procedural activities in the real life such as `mc2:BuyActivity`, `mc2:SellActivity`, and `mc2:ManufacturingActivity`. Meanwhile, to respond to the need of representing interaction activities in the virtual environment, we import the PROV Ontology (PROV-O)[1] and propose a specification of activity: `mc2:InteractionActivity` (Figure 3.3) . PROV-O expresses the PROV Data Model (PROV-DM) using the OWL2 Web Ontology Language. It provides a set of classes, properties, and restrictions that can be used to represent and interchange provenance information generated in different systems and under different contexts. It can also be specialized to create new classes and properties to model provenance information for different applications and domains. Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness.

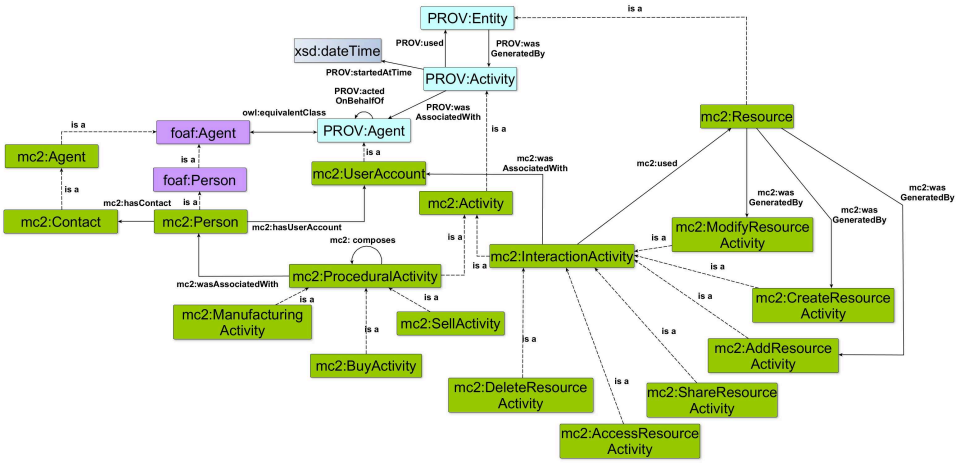


FIGURE 3.2: Integrating the activity module to MEMORAe-CRS ontology.

PROV-DM is the conceptual data model that forms a basis for the W3C provenance (PROV) family of specifications. PROV-DM distinguishes core structures, forming the essence of provenance information, from extended structures catering for more specific uses of provenance. PROV-DM is organized in three components, respectively dealing with: (1) entities; (2) activities, and the time at which they were created, used, or ended; and (3) agents bearing responsibility for entities that were generated and activities that happened. Each component is defined as follows:

- **Entity**: In PROV, things we want to describe the provenance of are called entities and have some fixed aspects. The term "things" encompasses a broad diversity of notions, including digital objects such

[1] PROV-O: The PROV Ontology, https://www.w3.org/TR/2013/REC-prov-o-20130430/

as a file or web page, physical things such as a mountain, a building, a printed book, or a car as well as abstract concepts and ideas. An entity is a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary. For each concept that could be instantiated from the PROV module, a concept is created as a specialization

- `mc2:Resource rdfs:subClassOf PROV:Entity`

- **Activity**: In PROV, an activity is something that occurs over a period of time and acts upon or with entities; it may include consuming, processing, transforming, modifying, relocating, using, or generating entities. Just as entities cover a broad range of notions, activities can cover a broad range of notions: information processing activities may for example move, copy, or duplicate digital entities; physical activities can include driving a car between two locations or printing a book. In MEMORAe-CRS ontology, we define digital activities as `mc2:InteractionActivity` and physical activities in real life as `mc2:ProceduralActivity`. For each concept that could be instantiated from the PROV module, a concept is created as a specialization

  - `mc2:Activity rdfs:subClassOf PROV:Activity`
  - `mc2:InteractionActivity rdfs:subClassOf mc2:Activity`
  - `mc2:ProceduralActivity rdfs:subClassOf mc2:Activity`

  Activities and entities are associated with each other in two different ways: activities utilize entities and activities produce entities. The act of utilizing or producing an entity may have a duration. The term 'generation' refers to the completion of the act of producing; likewise, the term 'usage' is the beginning of utilizing an entity by an activity. Generation is the completion of production of a new entity by an activity. This entity did not exist before generation and becomes available for usage after this generation. Before usage, the activity had not begun to utilize this entity and could not have been affected by the entity. Thus, we import the following object property:

  - `mc2:Activity PROV:used mc2:Resource`
  - `mc2:Resource PROV:wasGeneratedBy mc2:Activity`

  The `PROV:Activity` has the following data properties:

  - `PROV:Activity PROV:startedAtTime xsd:dateTime`
  - `PROV:Activity PROV:endedAtTime xsd:dateTime`

  Some activities are marked by both two properties. When you get access to a document, the activity is marked by a start time and an end time. Meanwhile, most activities are instanenous, i.e., they start and

end at approximately the same time. For instance, the activity of sharing a document only lasted between the time point that you make a request to the server and the time that the server responds with a success. The duration does not help our analysis. For the preliminary simplification, we suppose that all activities are instaneous and we import only the data property:

```
- PROV:Activity PROV:startedAtTime xsd:dateTime
```

- **Agent** is something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity. An agent may be a particular type of entity or activity. It means that the model can be used to express provenance of the agents themselves. Agents are defined as having some kind of responsibility for activities. The object property `PROV:wasAssociatedWith` is an assignment of responsibility to an agent for an activity, indicating that the agent had a role in the activity:

```
- mc2:UserAccount rdfs:subClassOf PROV:Agent
- mc2:Activity PROV:wasAssociatedWith mc2:UserAccount
(BY INFERENCE)
```

The specifications of mc2:InteractionActivity are as follows:

- CreateActivity: The activity of creating an original resource in the platform, e.g., creating a note;
- DeleteActivity: The activity of deleting a resource in the platform;
- ModifyActivity: The activity of modifying a resource in the platform;
- AccessActivity: The activity of accessing a resource in the platform;
- AddActivity: The activity of adding a resource in the platform which does not exist before, but is not created originally by the user who adds it;
- ShareActivity: The activity of sharing a resource in the platform, e.g., sharing with another group a document which is already added to the platform.

The creation by relation:

```
- mc2:Resource mc2:wasGeneratedBy mc2:CreateActivity
```

The addition by relation:

```
- mc2:Resource mc2:wasGeneratedBy mc2:AddActivity
```

The deletion by relation:

```
- mc2:DeleteActivity mc2:used mc2:Resource
```

The modification has two relations as we take into consideration the versioning of resources. Based on an old version of resource, a user modifies this resource by creating a new version:

```
- mc2:ModifyActivity mc2:used mc2:Resource
- mc2:Resource mc2:wasGeneratedBy mc2:ModifyActivity
```

The access by relation:

```
- mc2:AccessActivity mc2:used mc2:Resource
```

The sharing by relation:

```
- mc2:ShareActivity mc2:used mc2:Resource
```

Coming back to Peter's story:

"Peter adds a document named 'Advanced Java' in his personal sharing space."

We represent by triplets firstly, Peter acts an `mc2:AddResourceActivity` by his user account `kb:peterAccount`:

```
- kb:addAdv_Java mc2:wasAssociatedWith kb:peterAccount
```

This activity `kb:addAdv_Java` has effect on a resource `kb:docAdv_Java`:

```
- kb:docAdv_Java mc2:wasGeneratedBy addAdv_Java
```

The resource `kb:docAdv_Java` is indexed by `kb:indexkey_1` which is about subject "Java" and visible in his own personal sharing space:

```
- kb:indexkey_1 mc2:index kb:docAdv_Java
- kb:indexkey_1 mc2:aboutClass owl:Java
- kb:indexkey_1 mc2:isVisibleFor kb:peterSpaceOfGroup
```

"Peter shares it with his study group. Another member of the group, John, finds the document helpful and write an annotation 'Very inspiring to me!"'

```
- kb:shareAdv_Java mc2:wasAssociatedWith kb:peterAccount
- kb:shareAdv_Java mc2:used kb:docAdv_Java
```

The resource kb:docAdv_Java then is indexed by another indexkey kb:indexkey_2 which is about subject "Java" and visible in the public sharing space of the study group:

```
- kb:indexkey_2 mc2:index kb:docAdv_Java
- kb:indexkey_2 mc2:aboutClass owl:Java
- kb:indexkey_2 mc2:isVisibleFor kb:studySpaceOfGroup
```

"Another member of the group, John, finds the document helpful and write an annotation 'Very inspiring to me!"'

Firstly, John should possess a user account which belongs to the group kb:studyGroup

```
- kb:john mc2:hasUserAccount kb:johnAccount
- kb:johnAccount sioc:member_of kb:studyGroup
- kb:studyGroup sioc:has_member kb:johnAccount
```

Then he creates an annotation on the document kb:docAdv_Java:

```
- kb:createAnno_Adv_Java mc2:wasAssociatedWith kb:johnAccount
- kb:annoAdv_Java mc2:wasGeneratedBy kb:createAnno_Adv_Java
```

This annotation kb:annoAdv_Java aims at the resource kb:docAdv_Java:

```
- kb:annoAdv_Java mc2:hasTarget kb:docAdv_Java
```

3.1.2.2 Voting for a Resource

Users have different preferences on resources for different purposes. They express this preferences on the resources by giving a value between 1 and 5, the higher, the more they appreciate this resource. We model a vote as a mc2:Resource for mainly three reasons:

- Vote, as a mc2:SimpleResource, is indexed by a mc2:IndexKey which is about a owl:Class and visible in a certain mc2:Space;
- A vote is created by a user which is the result of a mc2:CreateResourceActivity;
- Vote can further be used for reasoning the competency of a user who has created, shared or modified the voted resource.

Users vote according to the relevance and suitability of a resource on a subject in a group. Even resulting from a vote of the same user on an identical subject, the results can also vary for different sharing space, as the viewers in different groups have a different level of cognition. Moreover, each sharing space may have a different concept focus which also differentiates the vote.

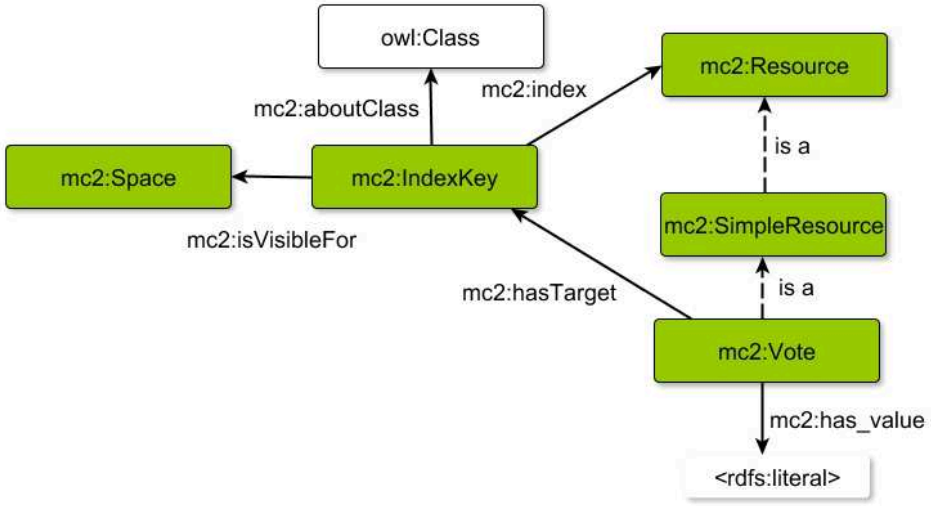


FIGURE 3.3: Integrating Vote to MEMORAe-CRS ontology.

The Vote has the following data property:

- Value: A value indicating the vote value.

Vote is a simple resource. It is indexed and aims at a `mc2:IndexKey`. The integration results in the creation of the following triplets:

```
- mc2:Vote rdfs:subClassOf mc2:SimpleResource
- mc2:Vote mc2:hasTarget mc2:IndexKey
- mc2:IndexKey mc2:index mc2:Vote
```

Picking up Peter's story:

"Peter votes 3 of 5 to 'Advanced Java' in his personal sharing space, as he holds it is already not suitable for his level."

As 'Advanced Java' is indexed by `voteAdv_Java_1` in the sharing space `kb:peterSpaceOfGroup`, this indexkey is the target of `kb:voteAdv_Java_1`

```
- kb:indexkey_1 mc2:index kb:docAdv_Java
- kb:indexkey_1 mc2:aboutClass owl:Java
- kb:indexkey_1 mc2:isVisibleFor kb:peterSpaceOfGroup
- kb:voteAdv_Java_1 mc2:hasTarget kb:indexkey_1
- kb:creVote_Adv_1 mc2:wasAssociatedWith kb:peterAccount
- kb:creVote_Adv_1 mc2:createResource kb:voteAdv_Java_1
- kb:voteAdv_Java_1 a mc2:Vote
- kb:voteAdv_Java_1 mc2:has_value "3"
```

The vote `voteAdv_Java_1` is indexed by `kb:indexkey_3` which is about subject "Java" and visible in his own personal sharing space:

```
- kb:indexkey_3 mc2:index kb:voteAdv_Java_1
- kb:indexkey_3 mc2:aboutClass owl:Java
- kb:indexkey_3 mc2:isVisibleFor kb:peterSpaceOfGroup
```

"In the study group. Peter votes 5/5 to 'Advanced Java' in the public sharing space, as he holds it is really useful to members of the group."

As 'Advanced Java' is indexed by `indexkey_2` in the sharing space `kb:studySpaceOfGroup`, this indexkey is the target of `kb:creVote_Adv_2`

```
- kb:indexkey_2 mc2:index kb:docAdv_Java
- kb:indexkey_2 mc2:aboutClass owl:Java
- kb:indexkey_2 mc2:isVisibleFor kb:studySpaceOfGroup
- kb:voteAdv_Java_2 mc2:hasTarget kb:indexkey_2
- kb:creVote_Adv_2 mc2:wasAssociatedWith kb:peterAccount
- kb:creVote_Adv_2 mc2:createResource kb:voteAdv_Java_2
- kb:voteAdv_Java_2 mc2:has_value "5"
- kb:indexkey_4 mc2:index kb:voteAdv_Java_2
- kb:indexkey_4 mc2:aboutClass owl:Java
- kb:indexkey_4 mc2:isVisibleFor kb:studySpaceOfGroup
```

3.1.2.3 Representing Competency in MEMORAe-CRS ontology

The part of competency represents the competency held by persons on different subjects. Competency has the following data property:

- Value: A value indicating the level of competency.

As presented in Chapter 2, competency can be specified as Project Competency, Cognitive Competency and Technical Competency (Figure 3.4). We represent this taxonomy in the semantic model to integrate the part of competency. The extension results in the creation of the following triplets:

```
- mc2:Competency rdfs:subClassOf owl:Thing
- mc2:Competency mc2:requires mc2:Competency

- mc2:ProjectCompetency rdfs:subClassOf mc2:Competency

- mc2:CognitiveCompetency rdfs:subClassOf mc2:Competency

- mc2:TechnicalCompetency rdfs:subClassOf mc2:Competency
```

Now Peter's story has a new plot:

"Peter's colleague, Jenifer, needs help for his Java programming. Among the study group, Julie has declared a good competency on C++ in her profile. Peter has realized a lot of activities on Java. Whom should Jenifer contact for help?"

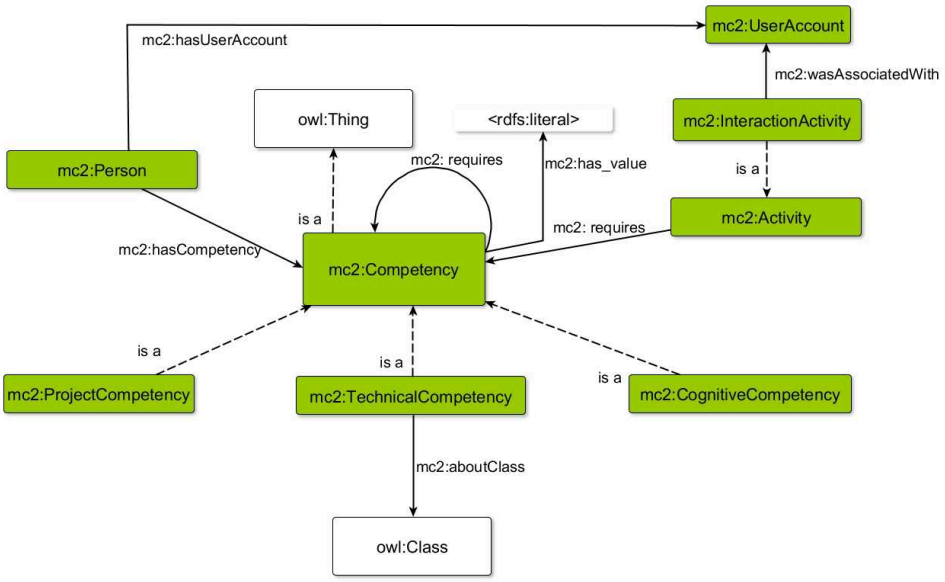


FIGURE 3.4: Representing competency in MEMORAe-CRS ontology.

Thanks to the representation of competency, we can present in the knowledge base (that has `kb` as a prefix) the above scenario by the following triples:

C++ and Java are all subclasses of `owl:Object-orientedProgrammingLanguage`. Julie's competency on Java requires her competency on C++:

```
- mc2:CompetencyC++ mc2:aboutClass owl:C++
- mc2:CompetencyJava mc2:aboutClass owl:Java
- owl:Java rdfs:subClassOf
owl:Object-orientedProgrammingLanguage
- owl:C++ rdfs:subClassOf
owl:Object-orientedProgrammingLanguage
- mc2:CompetencyJava mc2:requires mc2:CompetencyC++
- kb:Julie mc2:hasCompetency kb:CompetencyC++
```

At the same time we know that Peter has realized a lot of activities

```
{kb:Activity_Java_Peter}:- {kb:Activity_Java_Peter} mc2:aboutClass
owl:Java
- {kb:Activity_Java_Peter} mc2:wasAssociatedWith kb:Peter
- mc2:CompetencyJava mc2:requires {kb:Activity_Java_Peter}
```

To conclude, from the above example we can see that competency inference comes from two resources. One is from competency on semantically close subject declared by users. The other is from the activities realized on this subject. As for how to balance and quantify different features for reasoning users' competency, we propose to use different methods in the following section.

3.2 Applying Mathematical Methods for Competency Measurement

In the previous section, we presented our work for modeling the competency of users and we indicated what features we can extract from user traces to reason on users' competency. To deeply exploit these features, we need mathematical methods to quantify competency with the help of these features.

3.2.1 Time-decay Effect on Trace

Before discussing these methods, it is necessary to discuss the time-decay effect on the importance of trace. In a collaborative environment, the date when an action is carried out is also recorded. The forgetting curve hypothesizes the decline of memory retention in time. This curve shows how information is lost over time when there is no attempt to retain it as mentioned by Averell and Heathcote (2011). A typical graph of the forgetting curve purports to show that humans tend to halve their memory of newly learned knowledge in a matter of days or weeks unless they consciously review the learned material. So, it is necessary to take the decay of knowledge with time into consideration. In general, a recent action has more weight than a previous action. One of the common methods is to apply the impact of time on the importance of trace by a decay function. It is often used in time-sensitive recommender systems claimed such as in (Koren, 2010; Zhang and Liu, 2010; Chen, Jiang, and Zhao, 2010).

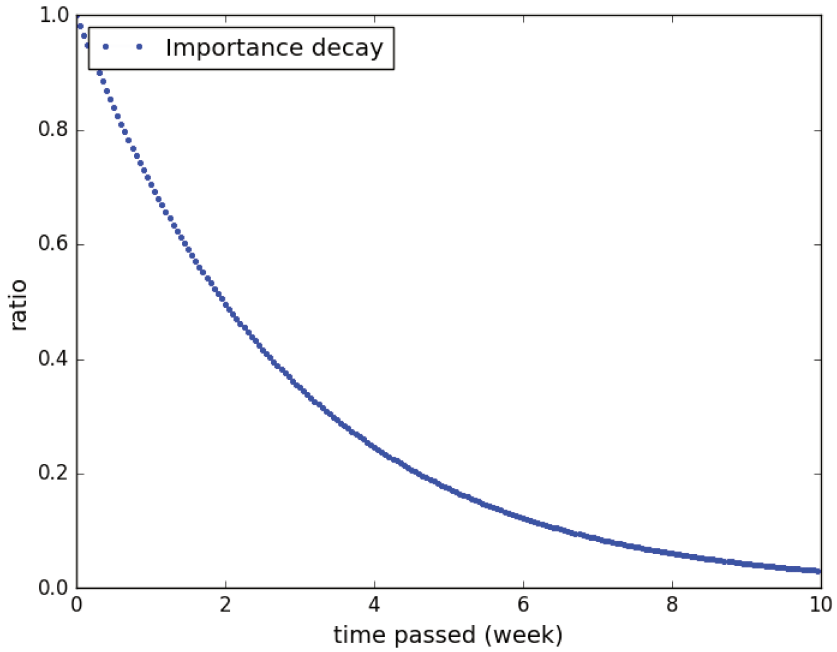


FIGURE 3.5: Time-decaying effects on importance of trace.

Ebbinghaus (1913) extrapolated the hypothesis of the exponential nature of forgetting. The decay function in Function 3.1 indicates that the trace we observe and analyze is less and less important as time goes by.

$$f(t) = e^{-\lambda t} \tag{3.1}$$

Its importance decays fast at first and then decays more slowly. The parameter λ controls the speed of decaying as t changes. It is indicated in (Baugh, 2000) that the regular pattern fits better the psychological pattern of humanity. Figure 3.5 shows the image of time-decaying effects on importance of trace. Ebbinghaus hypothesized that the speed of forgetting depends on a number of factors such as the difficulty of the learned material (e.g. how meaningful it is), its representation and physiological factors such as stress and sleep. In Figure 3.5, λ=0.15 and the importance of trace decreases to 50% of its original value after five weeks.

3.2.2 TF-IDF

3.2.2.1 Introduction and Previous Usage Scenarios

TF-IDF is short for **Term Frequency-Inverse Document Frequency**, which is almost the most classical method for domains like information retrieval and text mining. It is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus (Rajaraman et al., 2012). It measures the correlation of a term in presenting a document from a corpus. The TF-IDF value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which could adjust for the fact that some words appear more frequently in general. The values are used for evaluating the pertinence of a document.

As a document can be regarded as a combination of terms, we represent a document d as a set of terms $d = \{t_1, t_2, ..., t_m\}$. A corpus is a set of documents, namely $D = \{d_1, d_2, ...d_n\}$. In the following we explain TF-IDF in details.

Term Frequency
In terms of **Term Frequency**, the simplest way is to directly use the raw frequency of a term in a document, i.e. the number of times that term t occurs in document d. If we denote the raw frequency of t by $f_{t,d}$, thus the simplest TF form is $tf(t, d) = f_{t,d}$. There exist other more complicated forms which are adapted to different scenarios (Manning, Raghavan, and Schütze, 2008) as in Table 3.1:

- Boolean "frequencies": $tf(t, d) = 1$ if t occurs in d and $tf(t, d) = 0$ otherwise;

TABLE 3.1: Variants of TF weight.

weighting scheme	**tf weight**
Binary	$0, 1$
Raw frequency	$f_{t,d}$
Log normalization	$1 + log(f_{t,d})$
Double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{max\{f_{t',d}:t' \in d\}}$

TABLE 3.2: Variants of idf weight.

Weighting scheme	**idf weight** $(n_t = \|\{d \in D : t \in d\}\|, N = \|D\|)$
Unary	1
Inverse document frequency	$log\frac{N}{n_t}$
Inverse document frequency smooth	$log(1 + \frac{N}{n_t})$
Inverse document frequency max	$log(1 + \frac{max_{\{t' \in d\}} n_{t'}}{n_t})$
Probabilistic inverse document frequency	$log\frac{N-n_t}{n_t}$

- Logarithmically scaled frequency: $tf(t,d) = 1 + log(f_{t,d})$, or zero if $f_{t,d}$ is zero;
- Augmented frequency, to prevent a bias towards longer documents, e.g. raw frequency divided by the maximum raw frequency of any term in the document: $0.5 + 0.5 \cdot \frac{f_{t,d}}{max\{f_{t',d}:t' \in d\}}$. t' indicates the word that appears most frequently in the document.

Inverse document frequency

The **Inverse Document Frequency** measures how much information the word provides, namely, whether the term is common or rare across all documents. It was introduced as "term specificity" in a paper by Sparck Jones (1972). We divide the total number of documents by the number of documents containing the term, and then take the logarithm of that quotient.

$$idf(t, D) = log\frac{N}{|\{d \in D : t \in d\}|} \tag{3.2}$$

where:

- N: the total number of documents in the corpus $N = |D|$;
- $|\{d \in D : t \in d\}|$: number of documents where the term t appears (i.e. $tf(t,d) \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. Therefore it is common to adjust the denominator to $1 + |\{d \in D : t \in d\}|$.

Likewise, there also exist variants of idf weight as shown in Table 3.2. Then the TF-IDF is calculated as:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D) \tag{3.3}$$

TABLE 3.3: An example of calculating TF-IDF for "term, document, corpus"

Document 1		Document 2	
Term	**Term Count**	**Term**	**Term Count**
virtual	1	virtual	1
real	1	real	1
space	2	environment	2
technology	1	engineer	3

A high weight in TF-IDF is reached by two parts: a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents. Hence the weights tend to filter out common terms. Since the ratio inside the idf's log function is always greater than or equal to 1, the value of idf (and TF-IDF) is greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the idf and TF-IDF closer to 0.

Suppose we have a table containing term frequency of two documents as in Table 3.3 based on which we calculate TF-IDF of the term "virtual." Term frequency, if taken its basic form, is just the frequency that it appears in the document which is 1 in this case. As for idf, based on

$$idf(virtual, D) = log\frac{N}{|\{d \in D : t \in d\}|} \tag{3.4}$$

The numerator of the fraction N is 2 which is the number of documents. "virtual" appears in both documents, giving

$$idf(virtual, D) = log\frac{2}{2} = 0 \tag{3.5}$$

so TF-IDF is zero for the term "virtual" and all terms that appears in all documents in this corpus. This result indicates that the term "virtual" makes no contribution in distinguishing Document 1 from Document 2. Now we focus on another term "engineer." It occurs three times only in Document 2. For Document 2, TF-IDF of term "engineer" is:

$$tf(engineer, d_2) = 3 \tag{3.6}$$

$$idf(engineer, D) = log\frac{2}{1} \approx 0.301 \tag{3.7}$$

$$tfidf(engineer, d_2) = tf(engineer, d_2) \times idf(engineer, D) \approx 0.903 \tag{3.8}$$

Likewise, $tfidf(environment, d_2) \approx 0.602$. Apparently, compared to "environment", "engineer" better represents Document 2. In the next section we discuss the rationality of TF-IDF and why it is adaptable to our case.

3.2.2.2 TF-IDF and Information Entropy

To understand the form of TF-IDF, especially the denominator of the fraction in idf, information theory is a prerequisite. TF-IDF is actually a good example applying self-information and information entropy.

By definition, the amount of self-information contained in a probabilistic event depends only on the probability of that event: the smaller its probability, the larger the self-information associated with receiving the information that the event indeed occurred. In information theory, self-information or surprisal is a measure of the information content associated with an event in a probability space or with the value of a discrete random variable. This measure has also been called surprisal, as it represents the "surprise" of seeing the outcome (a highly improbable outcome is very surprising). This term was coined in (Tribus, 1961). It is expressed in a unit of information, for example bits, nats, or hartleys, depending on the base of the logarithm used in its calculation (Cover and Thomas, 2012). By definition, the amount of self-information contained in a probabilistic event depends only on the probability of that event. The self-information $I(\omega)$ associated with an event ω and its probability $P(\omega)$ is:

$$I(\omega) = log(\frac{1}{P(\omega)}) = -log(P(\omega)) \tag{3.9}$$

The smaller its probability, the larger the self-information associated with receiving the information that the event indeed occurred.

On the other hand, information entropy is a measure of unpredictability of information content. It can also be comprehended as how much unpredictability could be brought to us after an event takes place. Named after Boltzmann's H-theorem, Shannon defined the entropy H (Greek letter η) of a discrete random variable X with n possible values $\{x_1, ..., x_n\}$ and probability mass function $P(X)$ as the expectation of self-information of variable X:

$$H(X) = E[I(P(X))] \tag{3.10}$$

To get an informal, intuitive understanding, consider the example of a coin toss. When the coin is ideally fair, that is to say, when the probability of tossing heads is the same as the probability of tossing tails, then the entropy of tossing the coin is as high as it could be. This is because there is no way to predict the outcome of the coin toss ahead of time. The best we can do is predict that the coin will come up heads, and our prediction will be correct with probability $P(x_{head}) = P(x_{tail}) = \frac{1}{2}$.

$$H(X_{fair}) = E[-log(P(X_{fair}))] = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = 0.3 \tag{3.11}$$

Such a coin toss has one bit of entropy since there are two possible outcomes that occur with equal probability. That is to say, the outcome contains one bit of information. On the contrary, a coin toss with a coin that has two heads and no tails has zero entropy since the coin will always come up

heads, and the outcome could not bring us any new information.

$$H(X_{unfair}) = E[-log(P(X_{unfair}))] = -log(1) = 0 \tag{3.12}$$

Now let us look back to the TF-IDF. Among all the explanations, Liang (2007) explains TF-IDF by its origin of information theory and information entropy. Suppose that a document is a source of information (as tossing a coin in the previous example). Along with this, we also have the following assumptions:

- A document includes a list of n terms as $t_1, t_2, ..., t_n$;
- Each term appears $N_1, N_2, ..., N_n$ times, we also define $K = \sum_{i=1}^{n} N_i$;
- The frequency with which each term appears in the corpus is $Freq_1, Freq_2, ..., Freq_n$;
- The appearance of each term is independent and we ignore the order between terms (i.e. a document is considered as a bag of words).

Thus to compose such a document the probability is:

$$X = Freq_1^{N_1} * Freq_2^{N_2} * ... * Freq_n^{N_n} = \prod_{i=1}^{n} Freq_i^{N_i} \tag{3.13}$$

Its self-information can be presented as:

$$I(X) = -log(\prod_{i=1}^{n} Freq_i^{N_i}) = \sum_{i=1}^{n}(-N_i * log(Freq_i)) \tag{3.14}$$

We can also regard $I(X)$ as the minimum code length for coding the probability to compose this document. As stated in Equation 3.10, entropy is the expectation of self-information, thus the average self-information of every term is:

$$H(X) = \frac{I(X)}{K} = \frac{\sum_{i=1}^{n}(-N_i * log(Freq_i))}{K} \tag{3.15}$$

In this average code length, every term has a different contribution. If we quantify the importance of a term as the contribution to the code length, it is easy to reach the conclusion that: the more each term appears in a document and the less it appears in the corpus, the more it contributes to the coding of this document. For the term t_i , its contribution to average term coding is

$$-N_i * \frac{log(Freq_i)}{K} = \frac{N_i}{K} * log(\frac{1}{Freq_i}) \tag{3.16}$$

in which $\frac{N_i}{K}$ is the term frequency of t_i in the document, $log(\frac{1}{Freq_i})$ equals to $log\frac{|D|}{(n_t=|\{d \in D: t \in d\}|)}$ which is the inverse of document frequency of term t_i in corpus D, namely the inverse document frequency.

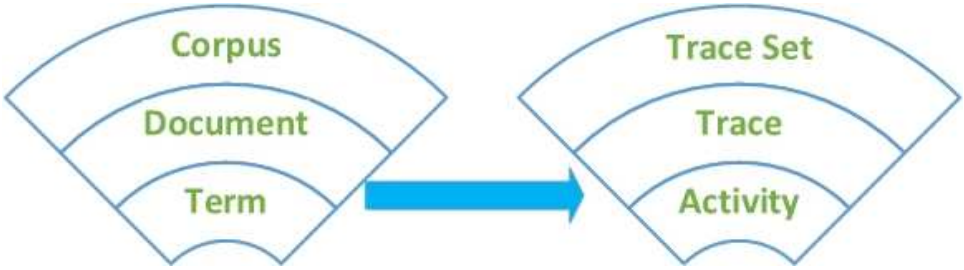


FIGURE 3.6: An analogy of concepts between classic TF-IDF and our scenario.

3.2.2.3 Adapting TF-IDF for Measuring Competency

In previous subsections, we presented TF-IDF and its typical usage scenarios. We also presented that TF-IDF is an application from information theory, especially self-information and entropy. In this subsection, we apply this method to our scenario and propose a method for measuring a user's trace and subjects. As presented in Section 2.3, a user's trace is composed of all activities that he/she acted and restored by us. Traces of all users compose a set of traces. Based on this, we can make an analogy between "term, document, corpus" and "activity, trace, trace set" as shown in Figure 3.6. If a word appears more often in a document and at the same time less often in the other documents of the same corpus, it could better represent this document. For our research, we are interested in evaluating the correlation between a trace of a given user and a certain subject. We propose to consider that if the activities of a user are more pertinent concerning a subject, the user has more knowledge about it. So we are able to recommend this user as an expert in this domain. In our case, we study the relation between activities, traces and the set of traces in a group of users working in the same environment.

To adapt the equation of TF-IDF, we have:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{3.17}$$

$$idf_i = log\frac{|U|}{|\{u_l : n_{i,l} > 0\}|} \tag{3.18}$$

The index of TF-IDF, $C_{i,j}$ indicating the competency of user j from group g on subject i, is defined as follows, which can be regarded as the relevance between a subject and a user:

$$C_{i,j} = tf_{i,j} * idf_i \tag{3.19}$$

where:

- $n_{i,j}$ is the number of activities concerning the subject i performed by user j;
- $\sum_k n_{k,j}$ is the number of activities concerning all k subjects performed by user j;

TABLE 3.4: An example of calculating TF-IDF for "activity, trace, trace set"

John's trace in group of 10 users		
Subject	**Frequency**	**Frequency in user group**
Java	10	7
Python	16	5
C++	2	8
PHP	7	3

- $|U|$ is the number of users in group g;
- $|\{u_l : n_{i,l} > 0\}|$ is the number of users in the set $|\{u_l\}|$ in group g who have performed at least one activity on the subject i.

In order to demonstrate how TF-IDF is applied, it is necessary to create a scenario. Suppose that in a CWE John has different activities on a set of subjects such as Java, Python, C++, and PHP. The frequencies of activities on each subject is shown in Table 3.4, John has realized in total 35 activities among which 10 activities concern Java and 7 activities concern PHP. The number of users is 10 among which 7 have realized at least one activity about Java and 3 about PHP. According to Equation 3.19, we obtain:

$$C_{Java,John} = \frac{10}{35} \times log\frac{10}{7} = 0.147 \tag{3.20}$$

$$C_{PHP,John} = \frac{7}{35} \times log\frac{10}{3} = 0.347 \tag{3.21}$$

From this simple example it is easy to come to a preliminary conclusion. Although the absolute frequencies of activities that John did on Java is more than that of PHP (10 to 7), as the idf also influences the result and the number of users who acts on PHP is less than that of Java, the results indicates that $C_{PHP,John}$ is bigger than $C_{Java,John}$ indicating the subject PHP better represents the traces of John.

3.2.3 Bayes Classifier

Previously, we focused on analyzing traces using TF-IDF. As a trace is composed of activities on a set of concepts, we need a method that better handles multi-dimension factors. The Naïve Bayes classifier is based on Bayes theorem with a strong (Naive) independence assumption, and is suitable for cases having high input dimensions (Ghazanfar and Prugel-Bennett, 2010). In statistical classification the Bayes classifier minimizes the probability of misclassification. In the following, we elaborate on adapting the method to our purposes.

Naïve Bayes is a conditional probability model. Given a problem instance to be classified, represented by a vector of features $F = (F_1, ..., F_n)$, we tend to calculate the probability that it belongs to class Cls. Using Bayes' classic

theorem, we have:

$$p(Cls|F_1, ..., F_n) = \frac{p(Cls)p(F_1, ..., F_n|Cls)}{p(F_1, ..., F_n)} \tag{3.22}$$

To simplify, we use the naïve Bayes classifier so that features $F_1, ..., F_n$ are independent. Here we still adapt the classic bag-of-words theory proposed by Mooney and Roy (2000) and regard a trace as an independent bag of activities, neglecting the logical relationship among the activities. Based on this assumption we have:

$$p(F_1, ..., F_n|Cls) = p(F_1|Cls)p(F_2|Cls), ..., p(F_n|Cls) \tag{3.23}$$

$$p(F_1, ..., F_n) = p(F_1)p(F_2), ..., p(F_n) \tag{3.24}$$

thus Equation 3.22 is reformulated as:

$$p(Cls|F_1, ..., F_n) = \frac{p(C)p(F_1, ..., F_n|Cls)}{p(F_1)p(F_2), ..., p(F_n)} \tag{3.25}$$

In our case, we aim at evaluating a user's competency on a certain concept with a trace he/she left on a set of concepts. So we adapt Equation 3.22 as:

$$p(Comp_j|Tra_i) = \frac{p(Comp_j)p(Tra_i|Comp_j)}{p(Tra_i)} \tag{3.26}$$

where $p(Comp_j)$ is defined as the a priori probability that a random user has the highest competency on concept $j \in J$. $p(Comp_j|Tra_i)$ represents the probability that a user $i \in I$ with trace Tra_i in the platform has the highest competency on concept j. $p(Tra_i)$ is the probability of composing Tra_i. As described previously, a trace is a combination of activities on a variety of concepts. We define $p(Tra_i)$ as:

$$p(Tra_i) = p(A_{i,1})p(A_{i,2}), ..., p(A_{i,n}) = \prod_k p(A_{i,k}) \tag{3.27}$$

where $p(A_{i,k})$ represents the probability that activities of trace i on concept k happen. Tra_i is composed of activities on n concepts respectively. So Equation 3.26 becomes:

$$p(Comp_j|Tra_i) = \frac{p(Comp_j)p(Tra_i|Comp_j)}{\prod_k p(A_{i,k})} \tag{3.28}$$

$p(Comp_j)$ is a constant because with no other conditions, all users have the same probability to perform the best for a concept. With no prior information, the probability of being the most competent among $|I|$ users equals to randomly drawing lot from N users. Thus an estimation of $p(Comp_j)$ is:

$$\hat{p}(Comp_j) = \frac{1}{|I|} \tag{3.29}$$

In our proposition, user competency is measured by the frequency of activities. We define $p(A_{i,k})$ as rank of frequency from the top among all users.

Thus the more frequent user i acts on concept k, the smaller $p(A_{i,k})$ is. For example, John realizes activities on concept $Java$ of which the frequency ranks second out of 10 users, then $p(A_{John,Java}) = 2/10 = 0.2$. It can be explained that if we randomly choose a user i from this set of users, the probability that i performed as much as John on $Java$ is 0.2.

$p(Tra_i|Comp_j)$ represents the probability that user i has a trace Tra_i if user i has the most competency on concept j. Two factors influence this value. Firstly, if a user has the most competency on j, it is highly probable that user i has much competency on semantically related concepts. As Tra_i is composed of a set of activities $\{A_{i,k}|A_{i,k} \in Tra_i\}$, we evaluate the semantic distance between j and each k. We use $\omega_{k,j}$ to represent the weight of concept k on j. Figure 3.7 shows a part of ontology of a use case for developing a semantic website. In view of complexity of calculations, we consider only the concepts semantically 2 edges away from j. Suppose j is the concept "Ontologic_request." Obviously, "Language" and "SQL" are two edges from j and we put their weight of influence to j as ω. "Request" and "SPARQL" are given 2ω and finally for the concept j itself we allocate 4ω. The sum of weights of concepts is $10\omega = 1$. Secondly, given the weight between concept k and j, the higher user i ranks on concept k, the bigger $p(Tra_i|Comp_j)$ is. We define:

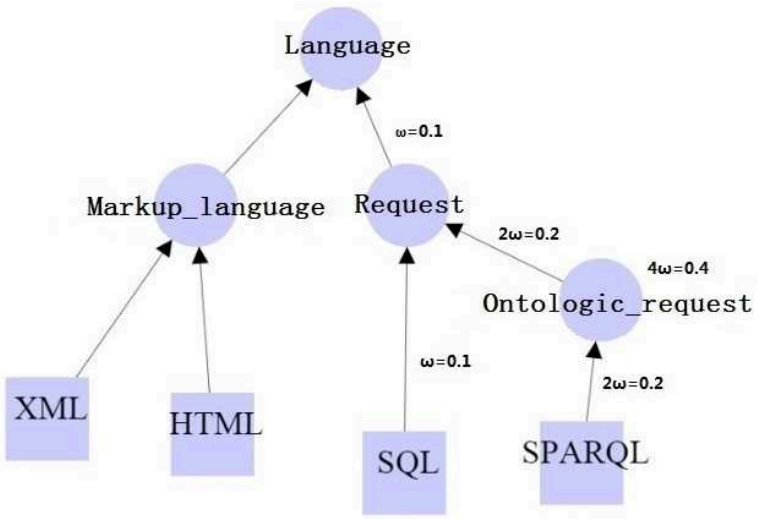


FIGURE 3.7: A part of ontology of a use case for developing a semantic website.

$$p(Tra_i|Comp_j) = \frac{1}{Z} \sum_{\{k|A_{i,k} \in Tra_i\}} [1 - p(A_{i,k})] \times \omega_{k,j} \tag{3.30}$$

in which Z is a scaling normalizing factor depending only on $\{A_{i,k}|A_{i,k} \in Tra_i\}$, that is, a constant if the values of the feature variables are known. We get:

$$p(Comp_j|Tra_i) = \frac{\sum_{\{k|A_{i,k} \in Tra_i\}} [1 - p(A_{i,k})] \times \omega_{k,j}}{N \times Z \times \prod_{k=1}^{n} p(A_{i,k})} \tag{3.31}$$

Finally, we obtain $p(Comp_j|Tra_i)$ and by comparing the probability of all users on the concept, we can finally give a recommendation about who is most probably the "best" at a concept given his/her trace.

3.2.4 Logistic Regression

Previously we applied a probabilistic method, Bayes Classifier, for measuring competency. However, this method suffers from a problem as it contains parameters assigned subjectively. When deciding the weight of semantically related concepts, we give them a value according to our expertise and experience. Inevitably this imports subjectivity to the results of the recommender system. To avoid subjectivity, we apply the Logistic Regression. This method is an important branch of machine learning methods which groups samples based on each sample's features in the database.

3.2.4.1 Introduction and Previous Usage Scenarios

In statistics, logistic regression is a regression model where the dependent variable is categorical. It was developed by statistician Walker and Duncan (1967). The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features).

Logistic regression is used widely in many fields, including the medical and social sciences. For example, Boyd, Tolson, and Copes (1987) used logistic regression to develop the Trauma and Injury Severity Score (TISS), which is widely used to predict mortality in injured patients. Logistic regression may be used to predict whether a patient has a given disease (e.g. diabetes; coronary heart disease), based on observed characteristics of the patient (age, sex, body mass index, results of various blood tests, etc.) (Freedman, 2009). Another example might be to predict whether an American voter will vote Democratic or Republican, based on age, income, sex, race, state of residence, votes in previous elections, etc (Harrell, 2014). In economics it can be used to predict the likelihood of a person's choosing to be in the labor force, and a business application would be to predict the likelihood of a homeowner defaulting on a mortgage. Conditional random fields, an extension of logistic regression to sequential data, are used in natural language processing.

Comparing with other regression algorithms that could be applied to our purpose, Logistic Regression is adaptable due to its high variability and non-linear distribution of a variety of input features. Whatever the input t is, the output $H(t)$ is always restricted to a rational set $(0, 1)$. It uses the logistic function to model an output variable:

$$H(t) = exp(t)/(1 + exp(t)) \tag{3.32}$$

From the image of logistic function in Figure 3.8, it is clear that $H(t)$ regresses to the extreme values (0 and 1) very fast if t deviates from 0. This feature well fits the need for a binary classifier.

Each example $Xple_i$ is presented by a pair $\{(x_{ti}, y_{ti})\}, \forall i \in N^+$ ($x_i \in R^n$, $y_i \in [0, 1]$) for the training set. n is the total number of features. In the training set, for each x_i the corresponding y_i is equal to 0 or 1 indicating whether x_i belongs to a certain class or not. The parameter vector of model $w \in R^n$ determines the weight of each dimension of vector x_i:

$$H_w(x_i) = exp(w^T x_i)/(1 + exp(w^T x_i)) \tag{3.33}$$

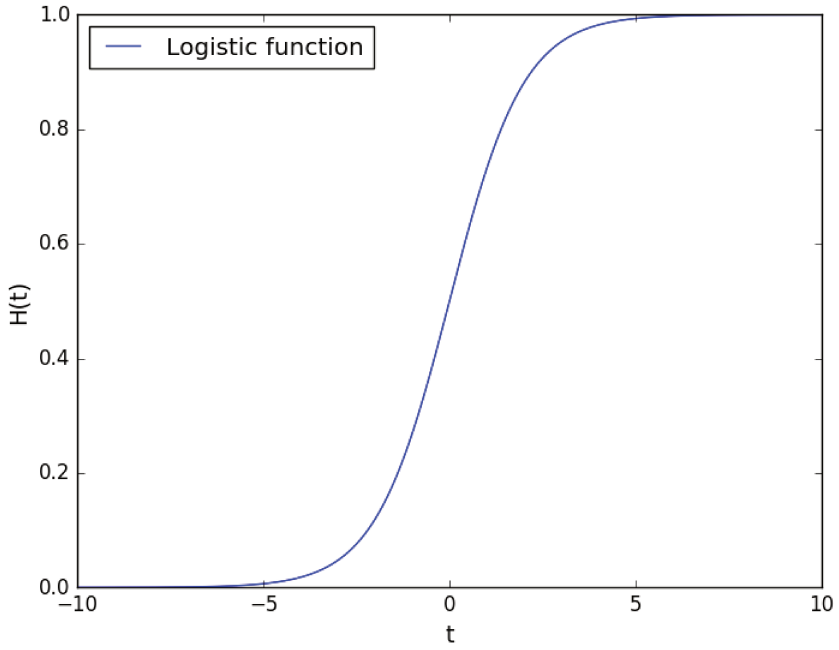


FIGURE 3.8: Image of the logistic function.

$H_w(x_i)$ also equals to the probability that $y_i = 1$ given x_i and w.

$$H_w(x_i) = P(y_i = 1|x_i, w) \tag{3.34}$$

w is determined by minimizing a loss function substituted by the training set of m samples:

$$L(w) = \sum_{i=1}^{m} (H_w(x_i) - y_{i,j})^2 + \|w\|^2 \tag{3.35}$$

where $\|w\|^2$ is applied for regularization to avoid over-fitting. Now that the Logistic Regression model is trained, we obtain the parameter set w. For each $Xple'_i$ in the testing set $\{Xple'_i\}$, we calculate and compare $H_w(x'_i)$ for classification.

TABLE 3.5: Features by roles of asker and answerer in CQA.(Liu et al., 2011)

Features	Description
Asker User Traces	
Answer/Question Ratio	Ratio of # of answers to # of questions
Self-voted best answer Ratio	Ratio of # of best answers that asker himself also voted to # of best answers in the questions posted
Total Questions Posted	# of questions proposed in the past
Answerer User Traces	
Best Answer Ratio	Ratio of # of best answers to # of total answered questions
Percentage of Vote	Percentage of users who voted his answers as the best
Average of Vote for past questions	Average # of votes for each answer he received for all questions responded
Average Vote by Asker	Average # of vote of this answers by the questions' askers
Average Words per Answer	# of characters in the answer
Remark: # is short for number	

3.2.4.2 Adapting Logistic Regression for Measuring Competency

To classify whether user i is competent on concept j ($Uy_{i,j} = 1$) or not ($Uy_{i,j} = 0$), we define the vector of features of user i on concept j as $Ux_{i,j} = [Ux_{i,j,(1)}, ..., Ux_{i,j,(n)}]$. Therefore Equation 3.33 becomes:

$$H_w(Ux_{i,j}) = exp(w^T Ux_{i,j})/(1 + exp(w^T Ux_{i,j})) \tag{3.36}$$

where $H_w(Ux_{i,j})$ equals to the probability that $Uy_{i,j} = 1$ given $Ux_{i,j}$ and w.

$$H_w(Ux_{i,j}) = P(Uy_{i,j} = 1|Ux_{i,j}, w) \tag{3.37}$$

Each of the n dimensions corresponds to one feature. Table 3.5 is a list of features that describe user performance in the CQA environment. In this table, user features in CQA are divided into three parts: features concerning activities that a user asks questions and features concerning activities that a user responds to questions. Answer/Question Ratio represents whether a user proposes more questions comparing to questions he/she answers. Self-voted best answer ratio indicates whether the user has the same opinion with the rest voting the best answer (but it does not necessarily mean that being different impairs a user's competency).

For the answerer part, best answer ratio and percentage of vote directly reflect whether a user's response is highly appreciated. Average of vote for the past questions evaluates an answerer's past performance. Average vote by asker tells whether the answerer well comprehends the question and satisfies the need of the asker.

Other features include frequencies of users activities on a concept, i.e., creating resource 5 times on "Java" and so on. We will explain in detail the features we take into account in Chapter 5.

3.3 Chapter Summary

In this Chapter we mainly focused on two parts of work to respond to the needs of a competency recommender system (CRS). Firstly, we proposed a semantic model (MEMORAe-CRS) that is capable of representing traces and competency. On the other hand, we adapted different mathematical methods (TF-IDF, Bayes Classifier, and Logistic Regression) for the usage of our scenario to measure and capitalize what we represent in the semantic model.

Chapter 4

Experiments and Evaluation

In previous chapter, we proposed three mathematical methods for processing trace data and providing recommendations (TF-IDF, Bayes Classifier, and Logistic Regression). Each of them has its merits and drawbacks from the aspects of efficiency, accuracy, etc. In order to compare these methods, we apply them to a dataset that is prepared from real life and used non-commercially by academics and scientists. With the results we discuss and conclude the scenario that each method best fits to the balance of efficiency and accuracy.

Evaluation of recommender systems can be divided into three types: online experiment, user study, and offline experiment (Shani and Gunawardana, 2011). Often it is easiest to perform offline experiments using existing data sets and a protocol that models user behavior to estimate recommender performance measures such as prediction accuracy. A more expensive option is a user study, where a small set of users is asked to perform a set of tasks using the system, typically answering questions afterwards about their experience. Finally, we can run large scale experiments on a deployed system, which we call online experiments. Such experiments evaluate the performance of the recommenders on real users which are oblivious to the conducted experiment.

We decide to apply an offline experiment. An offline experiment is performed by using a pre-collected data set of users choosing or rating items. Using this data set we can try to simulate the behavior of users that interact with a recommender system. In doing so, we assume that the user behavior when the data was collected will be similar enough to the user behavior when the recommender system is deployed, so that we can make reliable decisions based on the simulation. Offline experiments are attractive because they require no interaction with real users, and thus allow us to compare a wide range of candidate algorithms at a low cost. The downside of offline experiments is that they can answer a very narrow set of questions, typically questions about the prediction power of an algorithm. In particular, we must assume that users' behavior when interacting with a system including the recommender system chosen will be modeled well by the users' behavior prior to that system's deployment. Thus we cannot directly measure the recommender's influence on user behavior in this setting. Therefore, the goal of the offline experiments is to filter out inappropriate approaches, leaving a relatively small set of candidate algorithms

to be tested by the more costly user studies or online experiments. A typical example of this process is when the parameters of the algorithms are tuned in an offline experiment, and then the algorithm with the best tuned parameters continues to the next phase.

The rest of this chapter is organized as follows. Section 4.1 introduces the dataset on which we test our proposition. Section 4.2 then apply separately each method on the dataset. Based on the results, we also discuss advantages and disadvantages of each method and conclude the scenarios that each method performs best in Section 4.2. At the end of this chapter comes the conclusion.

4.1 Dataset

In order to test the performance of each method for real case. We seek for large dataset that is extracted from real life. Our method is tested on the dataset from Yahoo Webscope Program [1]. The Yahoo Webscope Program is a reference library of interesting and scientifically useful datasets for non-commercial use by academics and other scientists. All datasets have been reviewed to conform to Yahoo's data protection standards, including strict controls on privacy. Yahoo is pleased to make these datasets available to researchers who are advancing the state of knowledge and understanding in web sciences.

We choose to use the dataset "A4 - Yahoo Data Targeting User Modeling, Version 1.0". This data set contains a small sample of user profiles and their interests generated from several months of user activities at Yahoo webpages [2]. Each user is represented as one feature vector and its associated labels, where all user identifiers were removed. Feature vectors are derived from user activities during a training period of 90 days, and labels from a test period of 2 weeks that immediately followed the training period. Each dimension of the feature vector quantifies a user activity with a certain interest category from an internal Yahoo taxonomy (e.g., "Sports/Baseball", "Travel/Europe"), calculated from user interactions with pages, ads, and search results, all of which are internally classified into these interest categories. The labels are derived in a similar way, based on user interactions with classified pages, ads, and search results during the test period. It is important to note that there exists a hierarchical structure among the labels, which is also provided in the data set. All user IDs in the data set are anonymized. All feature and label names are replaced with meaningless anonymous numbers so that no identifying information is revealed. The dataset is of particular interest to machine learning and data mining communities, as it may serve as a testbed for classification and multi-label algorithms, as well as for classifiers that account for structure among labels. The dataset package (its directory as shown in Figure 4.1) mainly includes

[1] Yahoo Webscope Program http://webscope.sandbox.yahoo.com/

[2] Yahoo Labs Webscope dataset ydata-ytargeting-user-modeling-v1_0 [https://labs.yahoo.com/outreach]

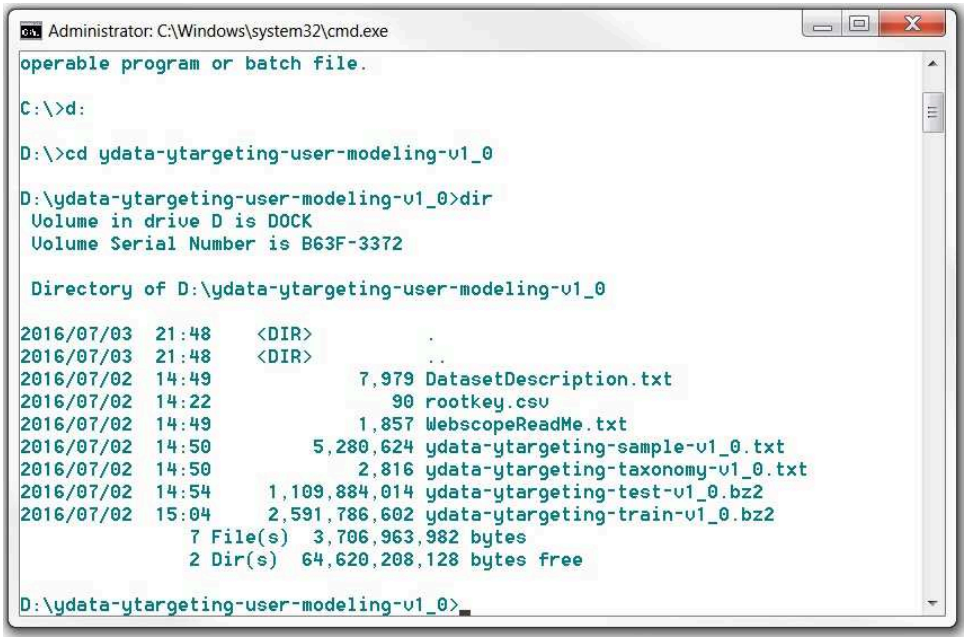


FIGURE 4.1: Directory of Yahoo Data Targeting User Modeling dataset.

the following files:

- "ydata-ytargeting-taxonomy-v1_0.txt" contains an interest taxonomy used at Yahoo (e.g., "Sports/Baseball", "Travel/Europe"), where the actual IDs and names of taxonomy categories have been anonymized. The format of the data is the following:

  ```
  child_node:parent_node
  ```

  A snippet of the taxonomy is as follows:

  ```
  8:209 233:209 120:209 253:-1
  ```

  The node with ID equal to -1 denotes the root node. There is a total of 380 categories. All categories follow a hierarchical structure. To summarize, there are in total 15 categories (nodes) in the first hierarchical level. For each lower hierarchical level, there are respectively 15, 119, 151, 78, 10, 5, and 2 categories (nodes) of interest;

- "ydata-ytargeting-train-v1_0.bz2" contains a training data set of Yahoo user profiles. The data contains 1,589,113 rows (i.e., user profiles), represented by a total of 13,346 features and 380 labels (each label corresponds to one category in the taxonomy).

  Features are extracted from a snapshot of user profiles on 2014/07/06 that contains previous 90 days of activity logs, where the activities include user events from the following groups: 1) page views, 2) search queries, 3) search result clicks, 4) sponsored search clicks, 5) ad views, and 6) ad clicks. Events from these six groups are all categorized into

the hierarchical taxonomy by an automatic categorization system and human editors. Each event is assigned to a category from a leaf of the taxonomy, and then propagated upwards toward parent categories.

Following the event categorization step, recency and intensity features are computed for each interest category in each of the six groups, where recency is defined as the number of days since the last user event in the group-category pair, and intensity is defined as exponentially time-decayed count of all events (with decay parameter set to 0.99; for detailed explanation of the feature generation process see Section "Dataset" in (Bi and Kwok, 2011)). If there was no activity in a group-category in last 90 days default intensity and recency are set to 0. The total number of features generated in this way is 13,346. All original feature IDs are anonymized in the data set, and have been assigned a random integer as an ID.

Labels are extracted from user activities that occurred during two-week period following 2014/07/06. Label is equal to +1 if user had an ad click in a specific interest category, and -1 if user had an ad view but no ad click in the interest category. 380 categories were kept for labels (for which there was enough positive examples). Only those users who had positive labels in at least two paths in the taxonomy tree were include in the data set. All original label IDs are anonymized in the data set, and have been assigned a random integer as an ID.

The data has the following sparse format (only non-zero valued features/labels are included in the profile):

```
space-separated list of featureID:value pairs <tab>
space-separated list of labelID:value pairs
```

A snippet of the training dataset is as follows:

```
83:0.84294 967:68.63747 1106:5 1133:0.86006
1237:0.49984 1527:0.58704 1535:6 12966:0.41712<tab>
32:-1 45:-1 51:-1 57:-1 198:-1 209:-1 211:-1 223:1
263:-1 268:-1 272:1 279:1 280:-1 290:-1 298:-1
313:6:0.50999 10837:5.33449 10886:16.8626
10911:0.57517 10945:0.41295 10967:47 <tab> 10:-1
17:-1 236:1 245:-1 248:-1 253:-1 270:-1 279:1 281:1
293:-1 316:-1 336:-1 350:1 370:-1 372:-1 373:-1
380:-1;
```

- "ydata-ytargeting-test-v1_0.bz2" contains a testing data set of Yahoo user profiles. The data contains 680,528 rows (i.e., user profiles), represented by a total of 13,346 features and 380 labels. The data is generated in the same way as file "ydata-ytargeting-train-v1_0.txt", with

only difference that it has a smaller subset of the Yahoo users;

- "ydata-ytargeting-sample-v1_0.txt" contains a sample of Yahoo user profiles. The data contains 1,000 rows (i.e., user profiles) that are a subset of the file "ydata-ytargeting-train-v1_0.bz2", represented by a total of 13,346 features and 380 labels.

This dataset well fits our need for testing the algorithms and for simulating our model for the following reasons:

- In the dataset, users are presented by vectors of features extracted from activities collected from the usage of Yahoo website. Further more, the activities include user events from the following groups: 1) page views, 2) search queries, 3) search result clicks, 4) sponsored search clicks, 5) ad views, and 6) ad clicks. Events from these six groups are all categorized into the hierarchical taxonomy by an automatic categorization system and human editors. This corresponds to our model of classified activities;
- Each activity is with a certain interest category from an internal Yahoo taxonomy (e.g., "Sports/Baseball", "Travel/Europe"), corresponding to our model that an activity is indexed by a concept in the ontology;
- The internal Yahoo taxonomy follows a hierarchical structure, corresponding to the subordinative relationships between concepts in our model;
- There are 13,346 features describing 1,589,113 user profiles in the training dataset and 680,528 user profiles in the test dataset. Features concerns about 380 labels. From the aspect of volume, this dataset is capable of evaluating the algorithm.

Based on the reasons above, it is reasonable to assume that results from the experiment we take on this dataset is also applicable to our case.

4.2 Experiments

4.2.1 Evaluation Methods

Before applying our methods on the dataset, we need to be clear of the evaluating standards. Indeed, recommender systems have a variety of properties that may affect user experience, such as accuracy, robustness, scalability, and so forth. Initially most recommenders have been evaluated and ranked on their prediction power — their ability to accurately predict the user's choices. As for our special case of recommending user competency, for the present we only seek for accuracy and time efficiency.

We use the **Root Mean Squared Error (RMSE)** metric, which is popular in evaluating accuracy. The methods we propose generate predicted ratings $\hat{r}_{ui}$ for a test set $\mathcal{T}$ of user-category pairs (u, i) for which the true ratings r_{ui} are known. Typically, r_{ui} are known because they are hidden in an offline

experiment, or because they were obtained through a user study or online experiment. In our dataset, r_{ui} are defined as label ID on the categories. The RMSE between the predicted and actual labels is given by:

$$RMSE = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} (\hat{r}_{ui} - r_{ui})^2} \tag{4.1}$$

Normalized RMSE (NRMSE) is the version of RMSE that has been normalized by the range of the ratings (i.e., $r_{max} - r_{min}$). r_{max} and r_{min} are respectively the highest rate and lowest rate in the training data set. Since it is simply the scaled version of RMSE, the resulting ranking of algorithms is the same as the ranking given by the unnormalized measures.

$$NRMSE = \frac{1}{r_{max} - r_{min}} \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} (\hat{r}_{ui} - r_{ui})^2} \tag{4.2}$$

We apply the NRMSE on the following application platform:

Operating System: Windows 7 Ultimate 64-bit
Processor: Inter(R) Core(TM) i5-3210M CPU @ 2.50GHz
Installed Memory: 8.00 GB
Software: Canopy 1.6.2 + Python 2.7.10

Besides accuracy, time efficiency is also an important aspect in evaluating algorithms (Miller et al., 2003). Whether or not to provide in-time recommendations when dealing with huge amount of data is a main task for algorithms. Therefore in the following experiments we also take into account operation time of each algorithms.

4.2.2 Experiment with Different User profile Volume

In this experiment, we change the number of rows of data (numbers of user profile) calculated by the three algorithms. For each algorithm, we take on five experiments. Each experiment uses a different percentage of scale of the whole "ydata-ytargeting-train-v1_0.bz2" file. Then NRMSE and operation time are measured by running the same experiment on the "ydata-ytargeting-test-v1_0.bz2" file. This is to compare the accuracy and operation time between three algorithms with different volumes of dataset. Results are shown in Figure 4.2 and Figure 4.3.

To conclude, it is clear that Logistic Regression exceeds the other two methods in nearly all the experiments. When the scale of dataset is small, the difference of accuracy between Logistic Regression, Bayes Classifier, and Logistic Regression is slight. But the NRMSE of Logistic Regression drops and meanwhile its accuracy grows fast when the number of user profiles calculated increases. The accuracy of this training-based model relies on

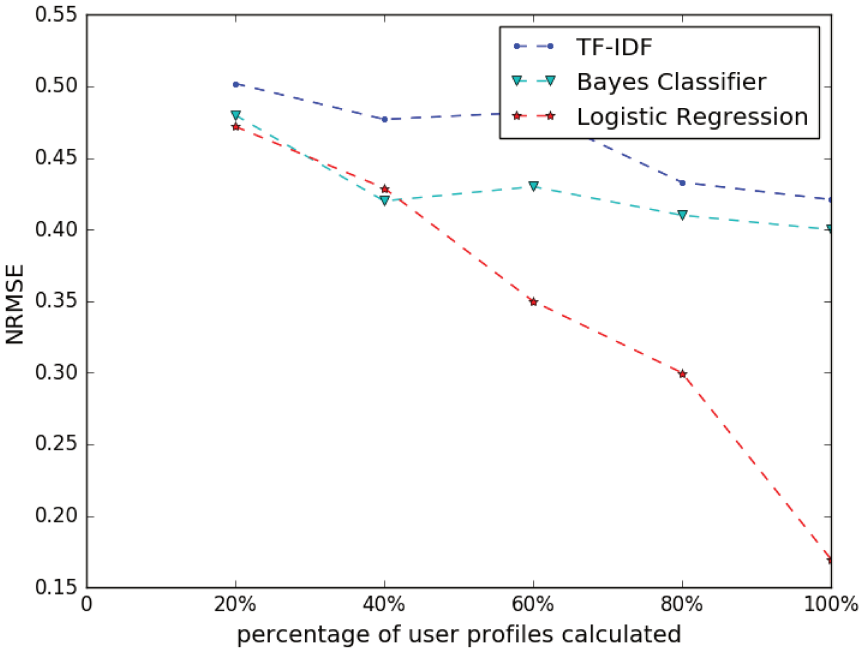


FIGURE 4.2: NRMSE of three methods changing the number of user profiles calculated.

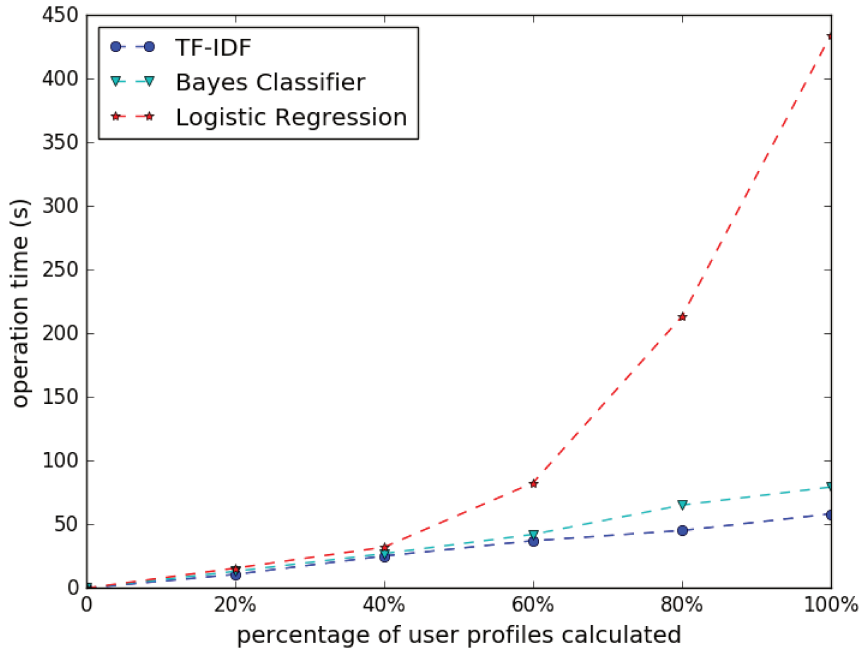


FIGURE 4.3: Operation time of three methods changing the number of user profiles calculated.

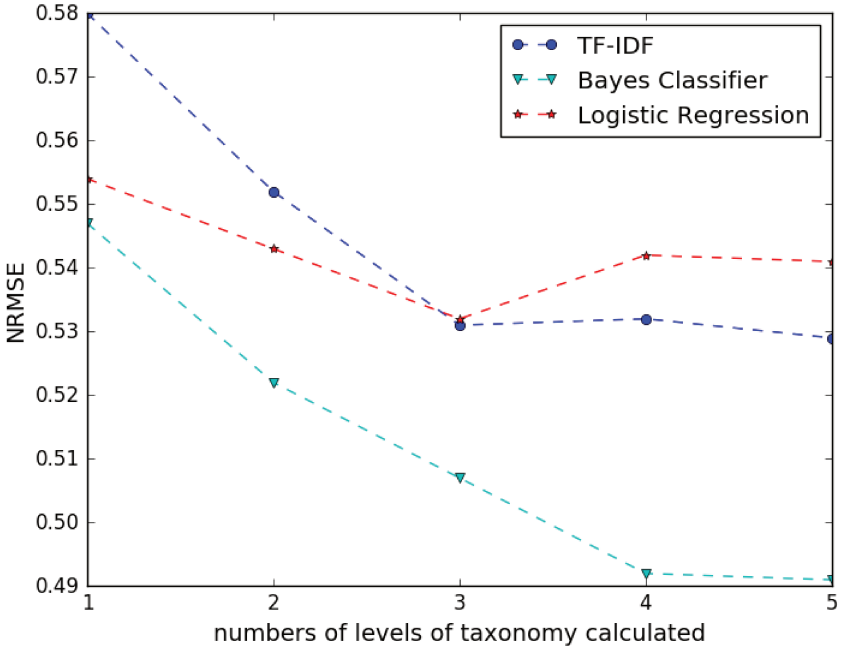


FIGURE 4.4: NRMSE of three methods changing the levels of taxonomy calculated.

number of rows of data calculated. On the contrary, as its scale of training dataset increases, the operation time of Logistic Regression also rises enormously due to its time complexity of $O(N^2)$. This phenomenon hinders the time efficiency of Logistic Regression as a candidate for our recommender system. Thus when the scale of dataset is small, choosing Logistic Regression seems perfect. Meanwhile, with a large scale of dataset make us encountered with a trade-off between accuracy and time efficiency.

4.2.3 Experiment with Different Levels of Taxonomy

In this experiment, we change the number of levels of taxonomy (numbers of categories) calculated by the three algorithms. Our algorithm will traverse not the target concept, but also its parent category and son category. Thus the hierarchical levels taken into consider will largely influence the accuracy and operation time. As there are in total seven hierarchical levels in the taxonomy, and the last two levels only contain a small amount of categories, thus we only implement experiments on the first five levels of categories. For each algorithm, we take on five experiments. Each experiment respectively considers different levels of the whole "ydata-ytargeting-taxonomy-v1_0.txt" taxonomy. Then NRMSE and operation time are measured by running the same experiment on the "ydata-ytargeting-test-v1_0.bz2" file. This is to compare the accuracy and operation time between three algorithms with different levels of taxonomy. Results are shown in Figure 4.4 and Figure 4.5.

To conclude, in this serie of experiments Bayes Classifier stands out as the number of hierarchical levels increase. When the number of levels is small,

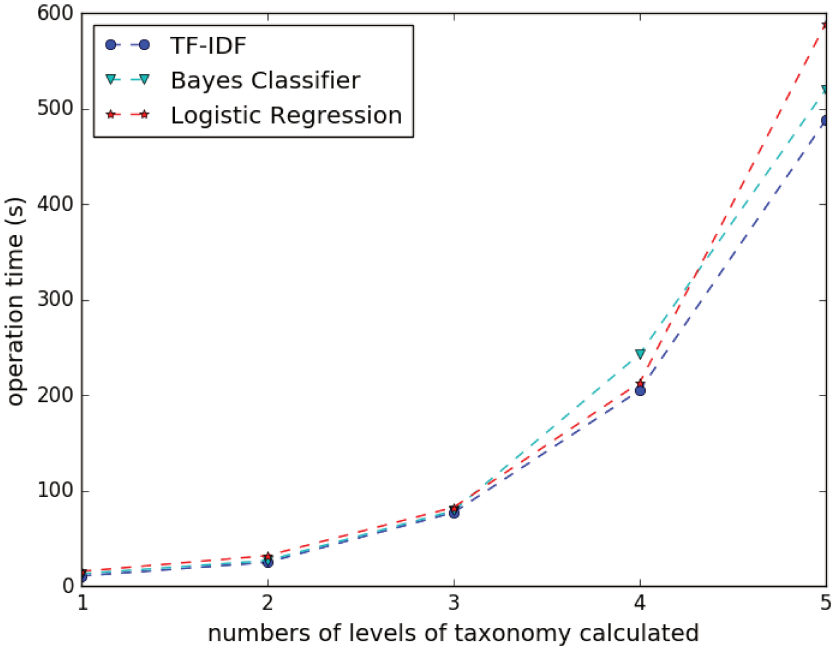


FIGURE 4.5: Operation time of three methods changing the levels of taxonomy calculated.

the difference of accuracy between Logistic Regression, Bayes Classifier, and Logistic Regression is slight. But the NRMSE of Bayes Classifier drops and meanwhile its accuracy grows fast when the number hierarchical levels of taxonomy calculated increases. As presented in Chapter 3, Bayes classifier relies on cause and effect relations between related categories. This might explain why Bayes Classifier outstands the other two algorithms in this test.

On the contrary, as the number of hierarchical levels of taxonomy increases, the operation time of all three methods rises enormously due to the time complexity of $O(N^2)$ according to levels. This phenomenon reminds us that, if not necessary, there is no need in calculating bigger numbers of levels especially concepts that is "far" from the target concept. On one hand it will not help increase a lot on the accuracy. On the other hand this will largely increase cost of time.

4.3 Chapter Summary

In order to test the performance of each method for real case, in this chapter, we seek for large dataset that is extracted from real life. Our method is tested on the dataset from Yahoo Webscope Program. This data set contains 1,589,113 user profiles and their interests generated from several months of user activities at Yahoo webpages. Results show that each method has its advantages and disadvantages under different conditions. Generally, Logistic Regression stands out on accuracy, especially when the number of

userfiles used for training the model is large. But it suffers from time complexity and thus lacks temporal efficiency. Bayes Classifier is more accurate when dealing with complex hierarchy of concepts. TF-IDF is moreover a compromise solution.

Chapter 5

Prototype

5.1 Introduction

Collaboration is a crucial element integrating skills and competencies of all members. Whether in real workspace or in virtual workspace, people always need collaborative tools to accomplish tasks, e.g., a blackboard on the floor or an online chat room. In modern digital era, a web-based Collaboration Working Environment (CWE) makes collaboration more convenient and accessible regardless of barriers of time and space. It involves several sub-systems with various tools in order to facilitate different levels of collaboration (e.g., communication or coordination) in groups, e.g., document management systems, electronic conferencing systems, working-flow systems, or knowledge management systems. In this chapter we present a prototype of platform that aims not only at supporting collaboration, but also at collecting user interaction traces, analyzing traces, and finally suggesting and recommending in order to facilitate and ameliorate collaboration.

In this chapter we will apply our method including competency model on a web-based collaborative platform E-MEMORAe2.0. The reminder of this chapter is organized as follows: Section 5.2 introduces our prototype: MEMORAe-CRS Web Application based on E-MEMORAe approach. Several collaborative tools will be explained with some explicit figures. At last comes the result of our recommender system.

5.2 E-MEMORAe-CRS Web Application

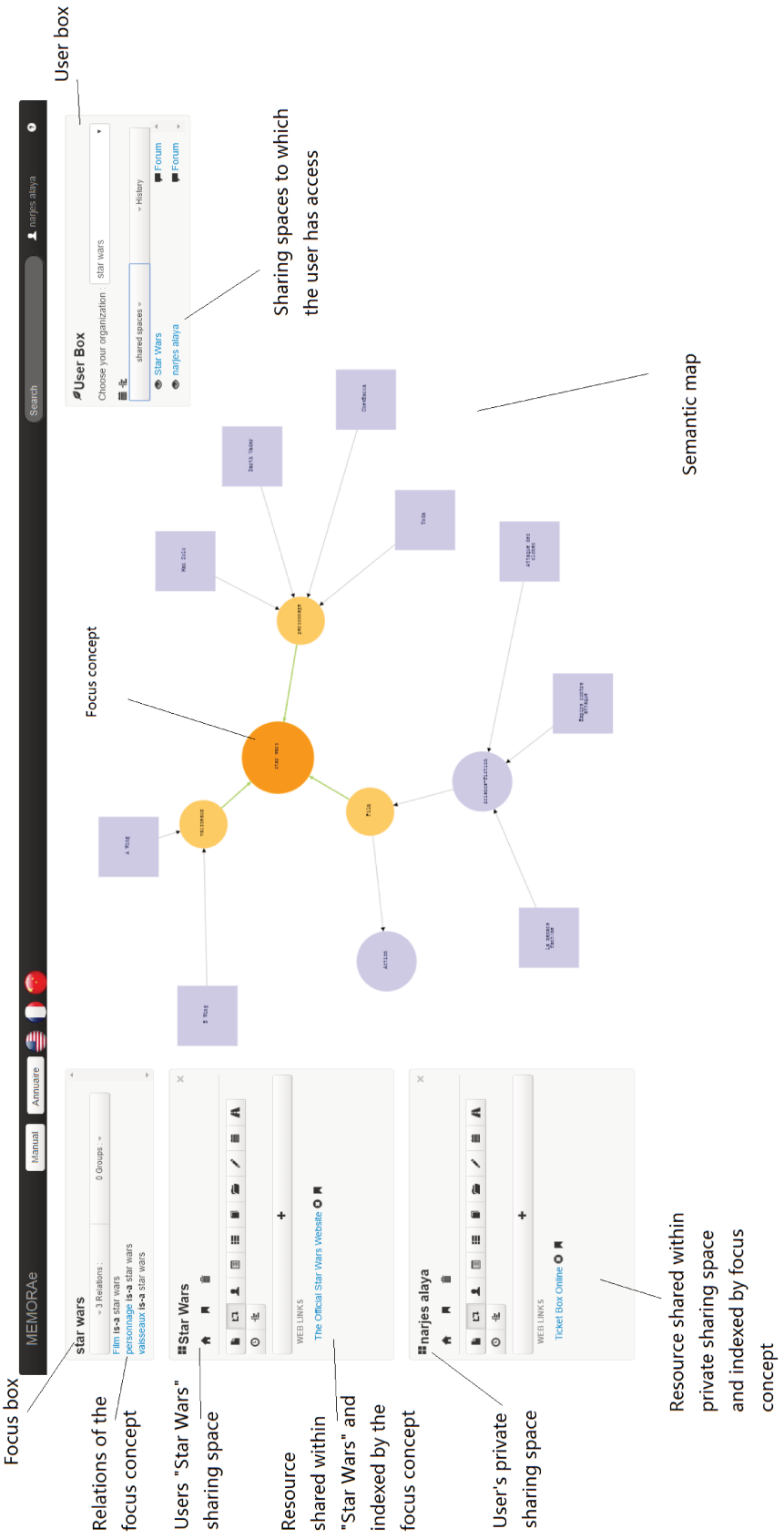


FIGURE 5.1: E-MEMORAe web platform.

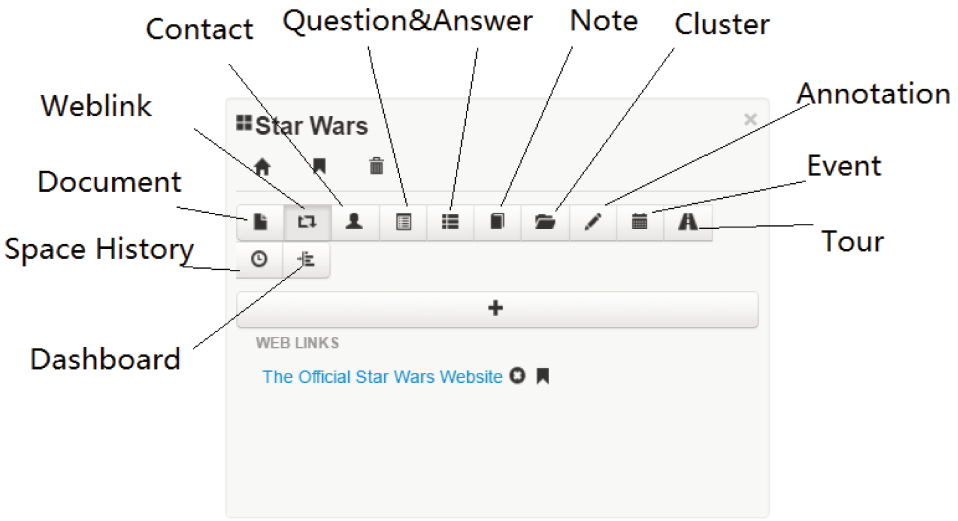


FIGURE 5.2: Resources of the sharing space.

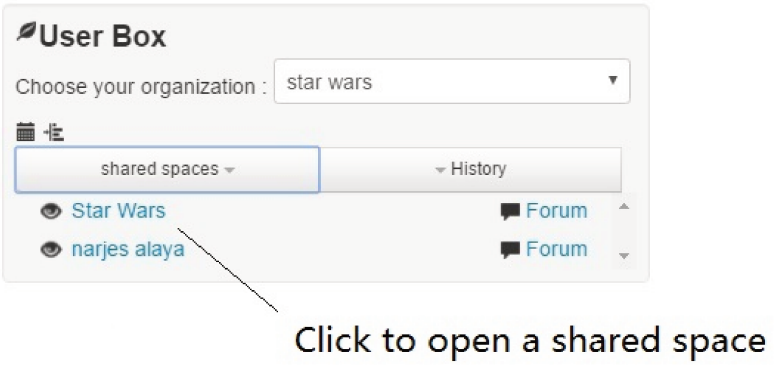


FIGURE 5.3: User box (sharing spaces page) in E-MEMORAe web platform.

MEMORAe-CRS web application is part of MEMORAe approach developed using web 2.0 technologies. This platform is based on MEMORAe-CRS model. It aims at facilitating knowledge sharing and capitalization (Figure 5.1). E-MEMORAe web platform supports sharing the resources modeled in the MEMORAe-CRS model as presented in Section 3.1 (documents, weblinks, contacts, Question&Answers, clusters, notes, annotations, agenda, tours) (Figure 5.2). All these resources are indexed by the concepts of an ontology that represents either the application ontology of an organization (e.g. the application ontology for to be completed), or another semantic reference based on the knowledge base (e.g. clients, prospects). This ontology is presented as a semantic map in the middle of the web page.

The semantic map defines a common reference shared among all users. A user can navigate through the map to view shared resources in the sharing spaces to which he/she has access. When the user clicks on a concept, this concept becomes the focus concept. Then he/she could open a sharing space to see different resources indexed by this focus concept which are shared in this sharing space. On the top-right corner of the platform, a user box is set where current user can open one or several sharing spaces that he/she belongs to (Figure 5.3). Sharing spaces can be opened on parallel while navigating through the map. This parallel view is advantageous because the user can see the resources indexed by the same focus concept and shared in different sharing spaces.

E-MEMORAe web platform can manage the fields of expertise of the organization and favor collaboration. For the purpose of defining, structuring and capitalizing explicit knowledge, the learning organizational memory is structured by means of ontologies that define knowledge within the organization on this platform (Abel and Leblanc, 2009). On the platform, generally users can:

- Manage users and user groups (transactions only by the administrator);
- Manage memories, private spaces and group workspace: these spaces associated with the memories to which the user has access are simultaneously visible, and it is easy to transfer content from one space to another;
- Access to knowledge map (ontology) and content (resources) based on the active sharing space: i.e., individual, group, and organizational spaces;
- Add and share the resources, e.g., PDF files or images;
- View and navigate through the concept map;
- Annotate concepts and resources;
- Utilize the concepts and the individuals of the knowledge map to index the resources;
- Collaborate by means of Web 2.0 tools to support informal communication and spontaneous production of knowledge, e.g., semantic wiki, chat or forum.

As the E-MEMORAe web platform is based on MEMORAe-CRS model, this platform is developed using a modular approach. When a module is removed from/ integrated to MEMORAe-CRS model, corresponding functions related to this module should also be removed from/ integrated to the MEMORAe web platform. In the following sections we will present in detail functionalities of MEMORAe-CRS web application.

5.2.1 Voting Resource

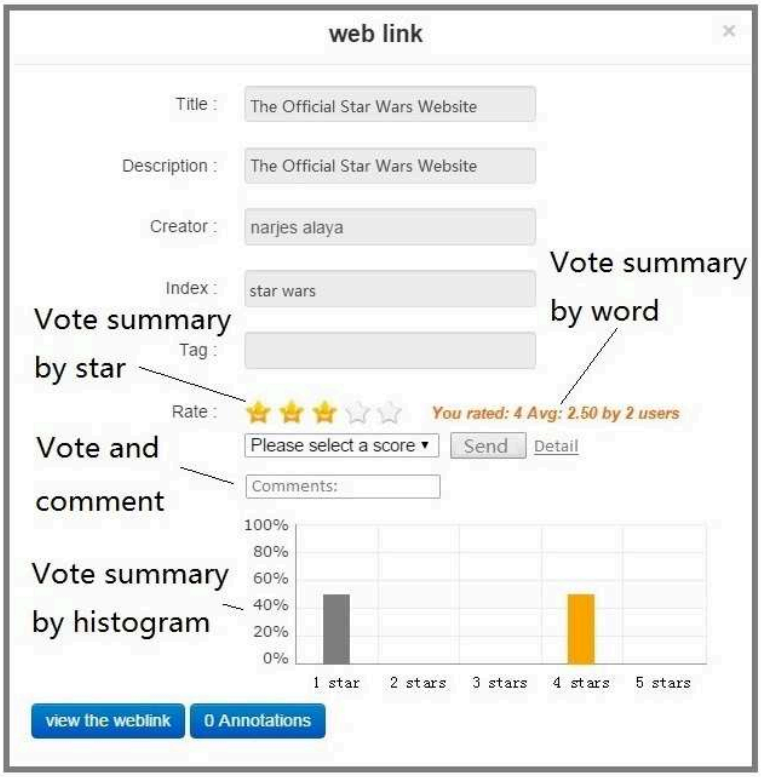


FIGURE 5.4: Voting a resource and showing results.

Users have different preferences on resources according to the quality, utility and relevance of it. They express these preferences on the resource by assigning it a value between 1 and 5, the higher, the more they appreciate this resource. Thus to extend the functions of the platform, we implement the voting system. Every resource can be voted upon regardless of its type. Users vote for the resource by opening it, choosing a corresponding value of vote, writing down a piece of comment and then sending it as in Figure 5.4. At the same time, voting results are spontaneously shown along with the resource. The current user has access to votes of all users who have access to this resource. Vote summary is shown by three way. Firstly the average vote value is shown by stars. After this graph follows a phase introducing the vote of current user, average vote, and the number of participating users. At the bottom a histogram is shown showing vote percentage of each vote value. Clicking the "detail" button opens a new page (Figure 5.5) displaying details of all votes and comments.

FIGURE 5.5: The vote detail of a resource.

This function also assists judging a users competency. Generally, if the resource that a user acts on has a high score, it indicates that the user is more competent on the concept the resource is about. For example, a user "Peter" writes a note on Java and it is positively reviewed as "5 stars", then the activity of writing this note promotes the reliability of his competency on Java as it is confirmed by other users.

5.2.2 Trace Dashboard

FIGURE 5.6: User box (history page) in E-MEMORAe web platform.

A user's activities on E-MEMORAe platform are stored in real time in the server. Figure 5.6 shows the history page of user box. The upper field of history page shows the concepts that the current user lately focused. As is shown, the current user previously focused on concepts such as "star

wars," "Film" and "Action." The lower field of this page indicates previous activities of the current user. In this example, user "Alaya" added a weblink entitled "Ticket Box Online."

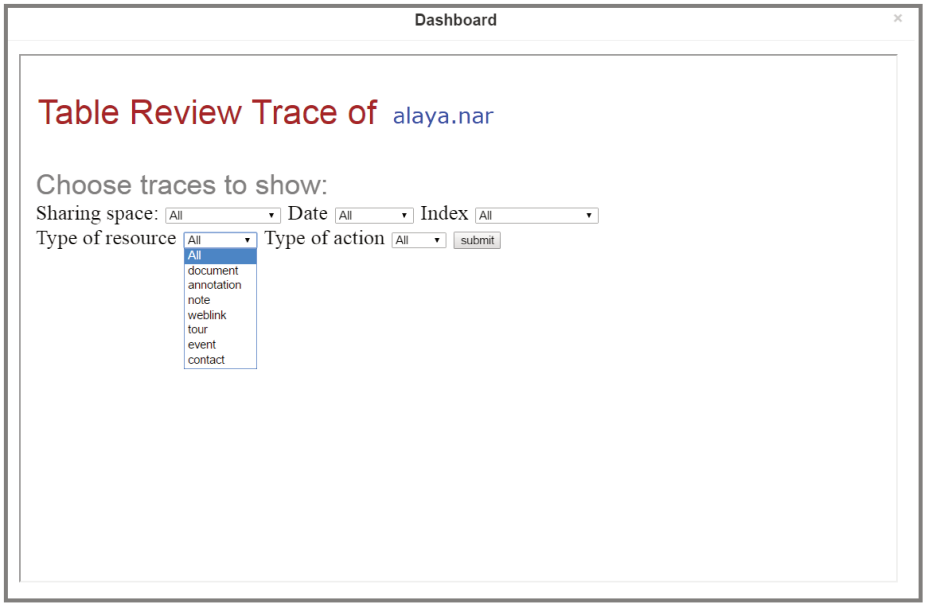


FIGURE 5.7: Filters of dashboard from user box in E-MEMORAe web platform.

The dashboard button in the user box directs the current user to the dashboard of his/her own. As shown in Figure 5.7, this dashboard can visualize and summarize all current user's activities regardless of sharing space and index concepts. The dashboard provides a set of filters to specify what activities must be shown. These filters include the activity taking place in which sharing space, on which date, indexed by which concept, concerns about what type of resource, and belongs to what kind of action.

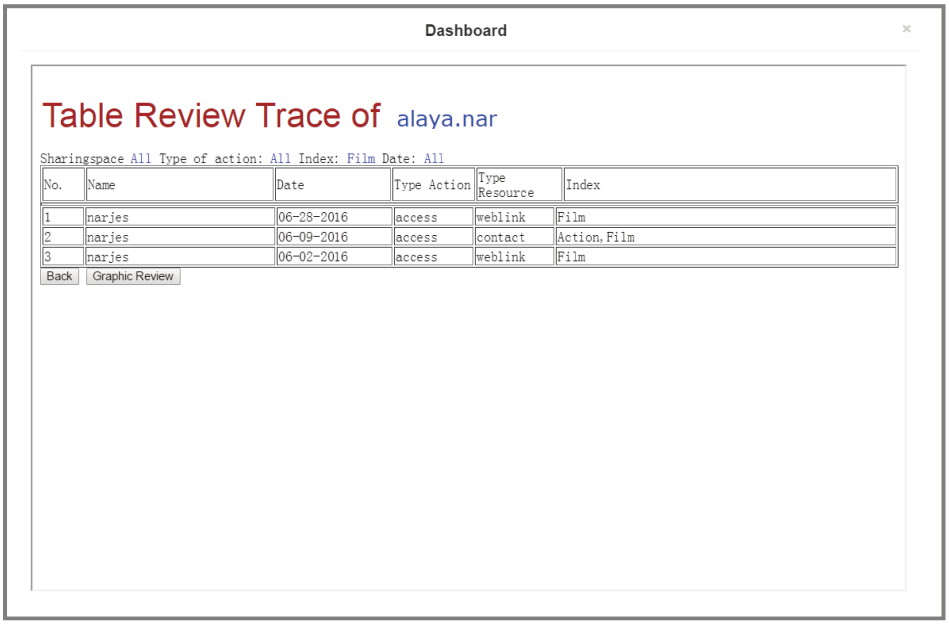


FIGURE 5.8: Table review of trace from dashboard in user box.

After submitting the filter conditions, dashboard shows a table review of traces ordered anti-chronologically (Figure 5.8). Here the current user uses index filter and only activities indexed by the concept "Film" are shown as a table.

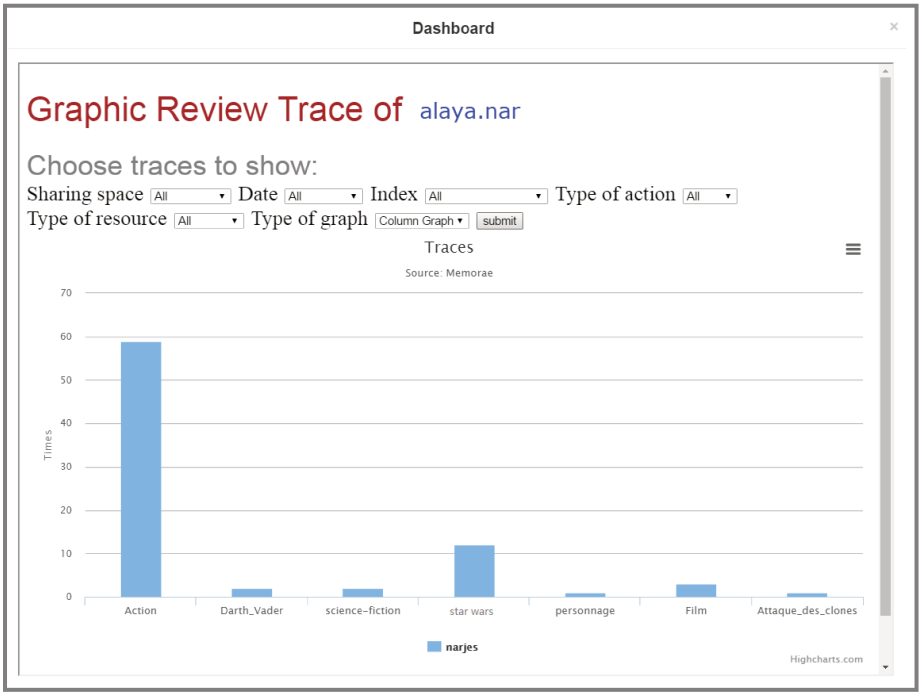


FIGURE 5.9: Presenting user traces by column graph in the dashboard of user box.

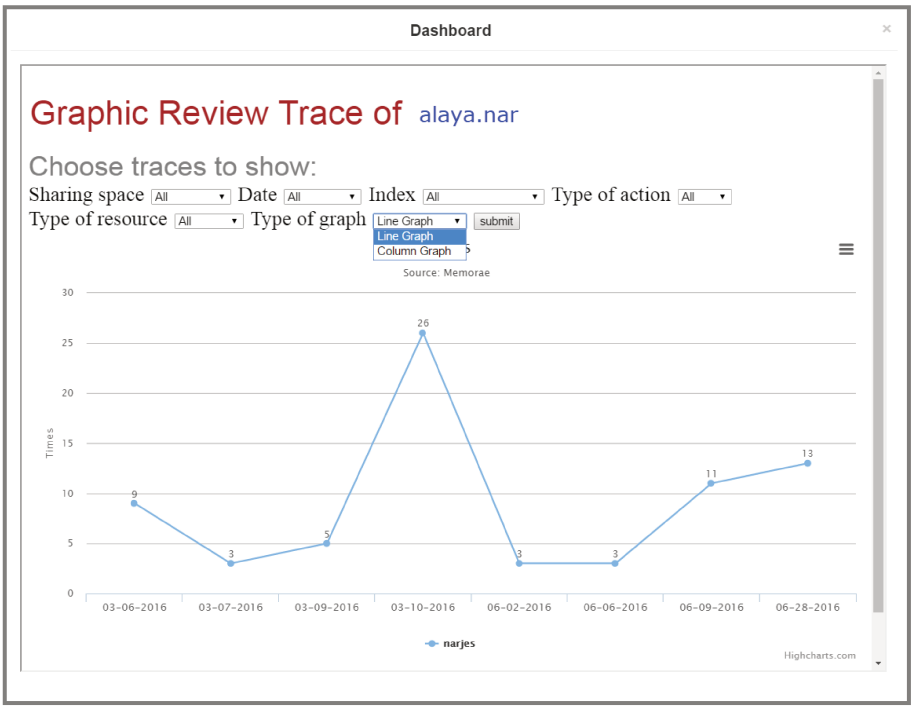


FIGURE 5.10: Presenting user traces by line graph in the dashboard of user box.

Clicking on the "Graphic Review" button directs the user to graphs demonstrating trace using Highcharts [1]. Here the user can choose either a column graph or a line graph to demonstrate the trace. In a column graph, the user activities are classified by indexed concepts. This form intuitively shows on which concept the user acts the most. In the example (Figure 5.9), the user has the most activities on concept "Action." On the other hand, line graph allows a visualization of activities by the order of time. This form intuitively shows at which time the user is most active on the platform. In the example (Figure 5.10), the user is most active on 10, Mars, 2016.

[1]Highcharts is a product that was created by the Norway-based company, Highsoft. Highcharts was released in 2009, and it is a charting library written in pure JavaScript. The product is developed in Vik, Norway and has been regularly featured in the national media, such as Finansavisen and Dagsrevyen.

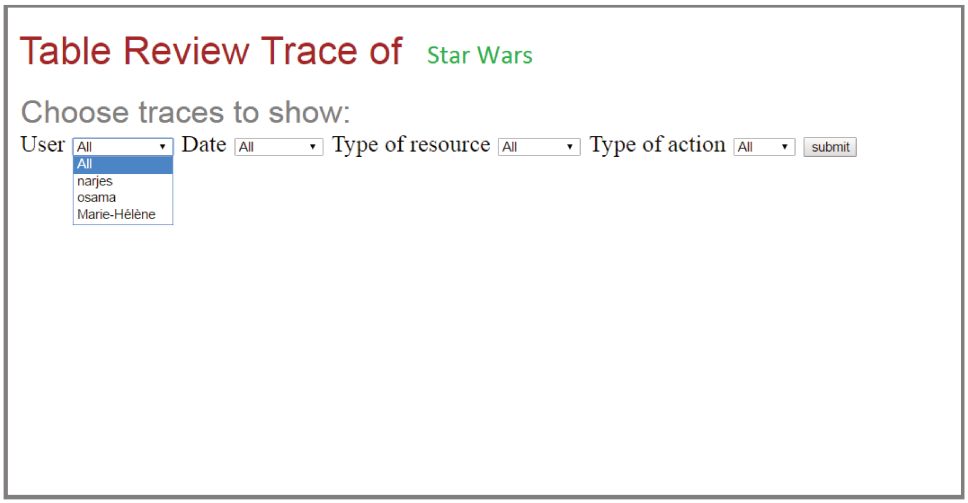


FIGURE 5.11: Filters of dashboard from memory box.

Previously shown in Figure 5.2, we also have a dashboard in the memory box of sharing space. This dashboard button in memory box directs the current user to the dashboard of activities indexed by the focus concept of the whole group which the sharing space belongs to. As shown in Figure 5.11, this dashboard can visualize and summarize all users' activities of the sharing space and focus index. It also provides a set of filters to precise what kinds of activities to be shown. These filters include the activity taking place by which member of the group, on which date, concerns about what type of resource, and belongs to what kind of action.

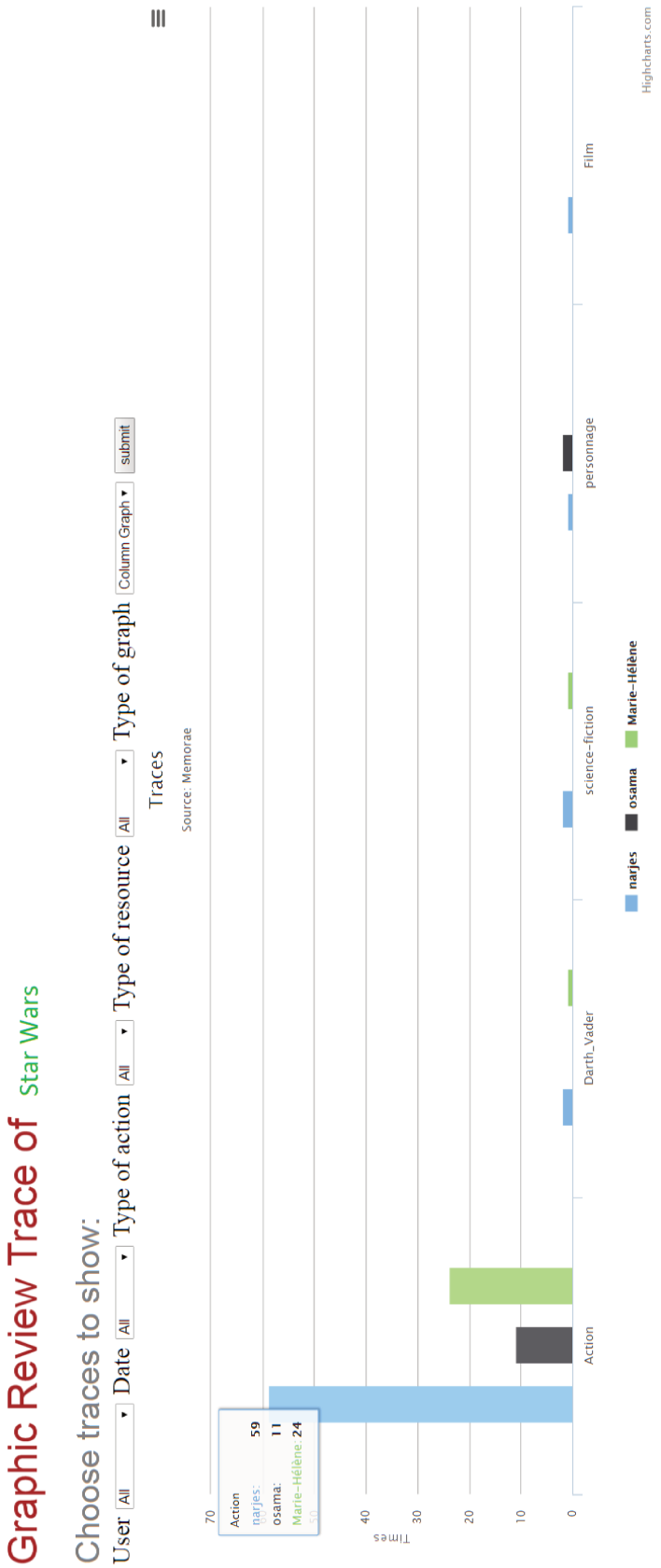


FIGURE 5.12: Presenting user traces by column graph in the dashboard of memory box.

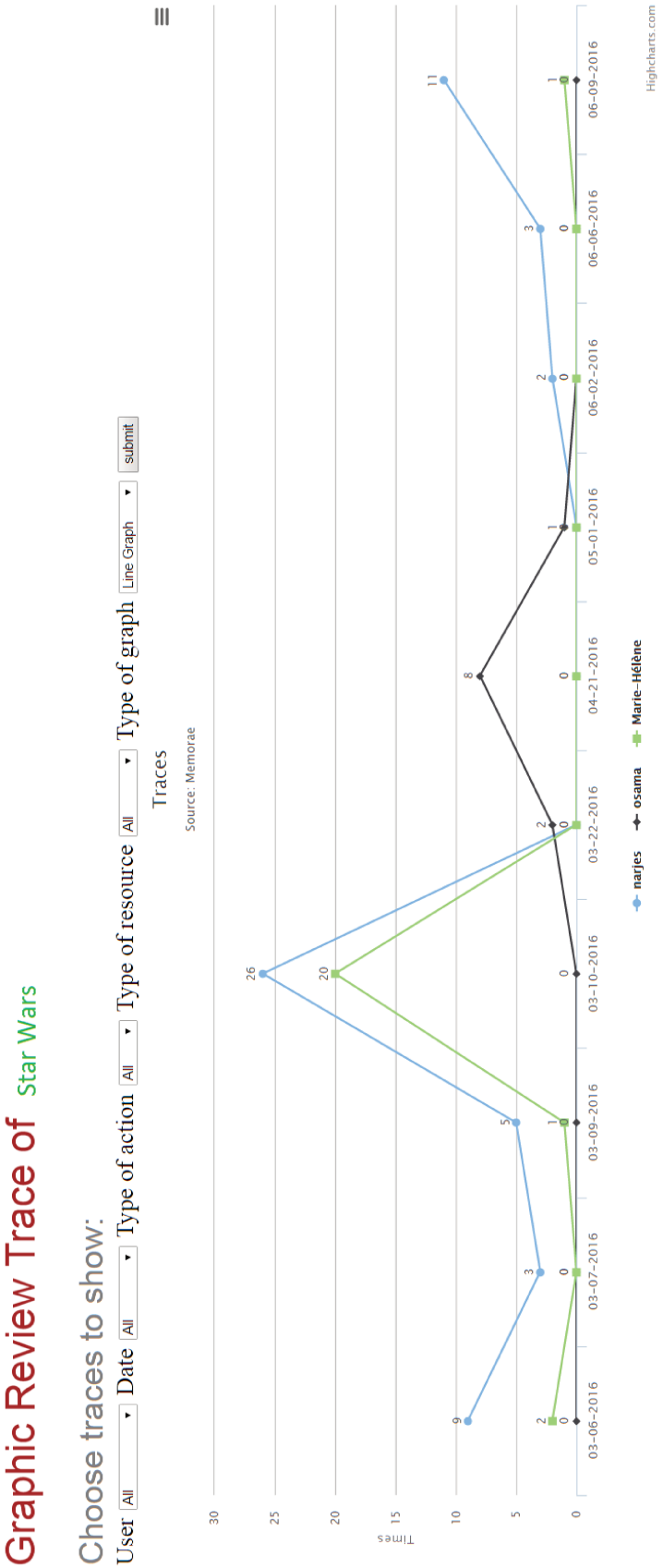


FIGURE 5.13: Presenting user traces by line graph in the dashboard of memory box.

Similarly, Clicking on the "Graphic Review" button directs user to graphs demonstrating trace. In a column graph, user activities are classified by indexed concept (Figure 5.12). This graph not only shows on which concept the user acts the most, but also indicates the comparison of frequencies of activities between users. On the other hand, line graph (Figure 5.13) allows a visualization of activities according to time. Likewise, this form compares at which time and which user is most active on the platform.

5.2.3 Recommendation

In Section 3.2, we presented in detail how to adapt TF-IDF, Bayes Classifier, and Logistic Regression to quantify the competency. Thus in Chapter 4, we applied these three methods on the dataset from Yahoo Webscope Program. The results we obtain show that when user profiles exceeds 40% of total test dataset, the accuracy of Logistic Regression overcomes the rest. But at the same time, it suffers from a long operation time. Meanwhile, when the number of levels of taxonomy increases, Bayes Classifier has an advantage over the rest.

Until now this prototype is not applied into practice with a big crowd of users and we suffer from a lack of the volume of dataset. As now in the prototype the number of user profiles and level of ontologies are small, we apply the method TF-IDF in our propositions which has the smallest complexity. Personal Recommendations based on traces of user "alaya.nar" shown in Figure 5.8 are shown in Figure 5.14. The recommender system proposes to the current user that his/her top competent concept is "Film" and the least competent concept is "Attaque des clones". According to this result, user "alaya.nar" can correspondingly assign the time of work on the platform, e.g., work more on "Attaque des clones".

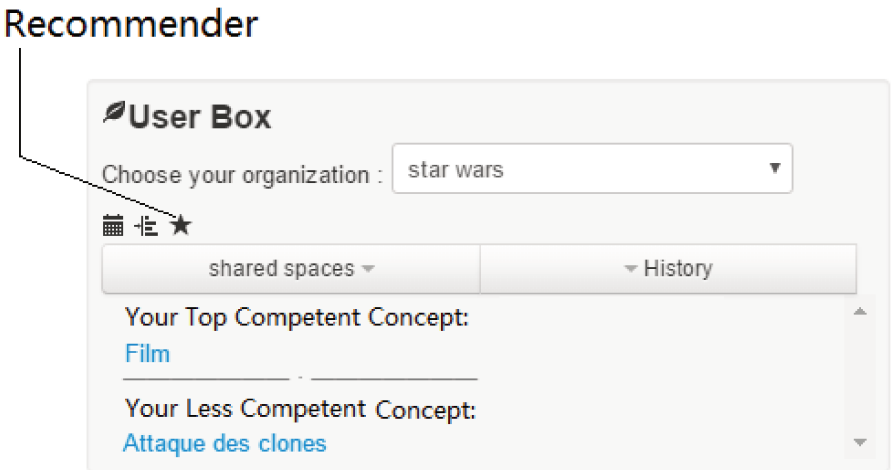


FIGURE 5.14: Recommendation on current user for most and least competent concept.

Similarly, recommendations are given to the whole group "Star Wars" according to traces of each member (Figure 5.15). Group members get aware of their well-done concept by referring to "Group Top Competent Concept." In this case, the top competent concept for the group is "Film". Now as we are focused on the concept "Action" on the semantic map, the recommender system indicates that "narjes alaya" is the most competent according to her previous traces on the platform. Other users can take this as a reference and choose "narjes alaya" as a partner when working on "Action".

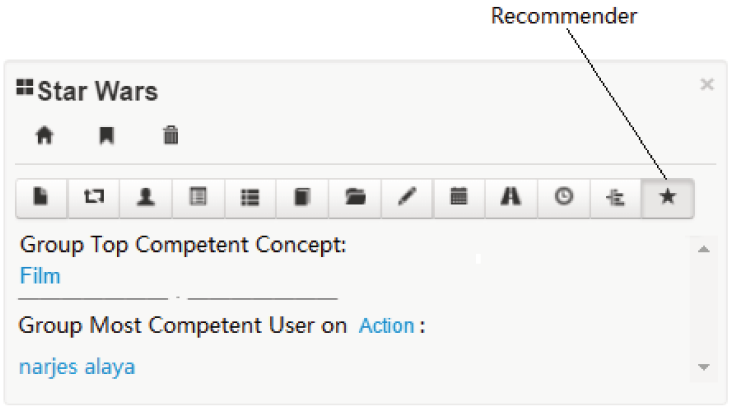


FIGURE 5.15: Recommendation on the group of current user.

5.3 Chapter Summary

In this section, we implement our proposal on the E-MEMORAe web platform to obtain a MEMORAe-CRS Web Application. Several parts of work is done. Firstly we integrate voting system which well represents opinions of users. We show trace dashboard for both current user box and for the whole organization which the group of users belong to. Finally, based on the semantic model of traces and competency, we use the proposal algorithms to give recommendations. These recommendations are divided to current user, and to the whole group he/she belongs to.

Chapter 6

Conclusion, Perspectives and Future Work

This chapter aims to make a conclusion of our work. Looking to the future, at the end we present some perspectives and some future extension of our thesis.

6.1 Conclusion

The goal of this work is to help ameliorate organizing collaboration by the means of a competency recommender system. To organize a competitive group, it is necessary to evaluate group members' competencies in order to better allocate task and resource. While competencies are not new to most organizations and companies, what is new is their increased application across varied human resource functions (i.e., recruitment/selection; learning and development, performance management, career development and succession planning, human resource planning). At the same time, measuring competency is never an easy task because of its intangibility. What's more, the results of competency assessment is inclined to be influenced by unneuturalty and subjectivity.

We try to mediate the conflict between the need of assessing user competency and a lack of proper methods by applying recommender system. Typical recommender system proposes items to potential buyers. However adapting this method can also help recommending competent person to the collaboration group with a certain need. Our proposal is based on analyzing user activities on collaborative platforms.

Firstly, definition and modeling of collaborative traces are necessary. A digital trace can be considered as a set of information recording the user's interactions within the framework of the system. Trace can be considered as a type of resources in the information system. Consequently, it is necessary to build a model to analyze and exploit traces that could assist user's work in many possibilities, e.g. decision-making, recommending, etc. To exploit and reuse traces, a trace model is with no doubt required. We retake the model of Li (2013) and further classify interactive activities as six different types. The advantage of doing so is to distinguish that a different type of

activity has its own weight and importance on competency inference. Competency is inferred by the activities that a user realizes in a platform. Based on this, we build a competency model. Competency inference comes from two resources. One is from competency on semantically closed subject declared by users. The other is from the activities realized on this subject.

In order to balance and quantify different features for reasoning users' competency, we propose to try different mathematical methods. We adapt TF-IDF from the domain of information retrieval, Bayes Classifier and Logistic Regression from machine learning. A piece of theoretical work is firstly done for the adaptation. In order to test the performance of each method for real case. We seek for large dataset that is extracted from real life. Our method is tested on the dataset from Yahoo Webscope Program. This data set contains 1,589,113 user profiles and their interests generated from several months of user activities at Yahoo webpages. This dataset well fits our need for testing algorithms and for simulating our model for several reasons:

- In the dataset, users are presented by vectors of features extracted from activities collected from the usage of Yahoo website. Further more, the activities include user events from the following groups: 1) page views, 2) search queries, 3) search result clicks, 4) sponsored search clicks, 5) ad views, and 6) ad clicks. Events from these six groups are all categorized into the hierarchical taxonomy by an automatic categorization system and human editors. This corresponds to our model of classified activities;
- Each activity is with a certain interest category from an internal Yahoo taxonomy (e.g., "Sports/Baseball", "Travel/Europe"), corresponding to our model that an activity is indexed by a concept in the ontology;
- The internal Yahoo taxonomy follows a hierarchical structure, corresponding to the subordinative relationships between concepts in our model;
- There are 13,346 features describing 1,589,113 user profiles in the training dataset and 680,528 user profiles in the test dataset. Features concerns about 380 labels. From the aspect of volume, this dataset is capable of evaluating the algorithm.

Results show that each method has its advantage and disadvantage under different conditions. Generally, Logistic Regression stands out on accuracy, especially when the number of user files used for training the model is large. But it suffers from time complexity and thus lacks temporal efficiency. Bayes Classifier is more accurate when dealing with complex hierarchy of concepts. TF-IDF is moreover a compromise solution.

Now that we obtain the model for traces and competency, as well as the methods for analysis and quantification, we propose the competency recommender system based on E-MEMORAe web platform. E-MEMORAe web platform is based on MEMORAe-core 2 model. The platform is developed using web 2.0 technologies. Based on MEMORAe-core 2, the platform

is dedicated for collaboration and resource sharing between members of an organization. We ameliorate this collaborative platform by adding two kinds of resources: voting and question answering. Users have different preference on resources for different purposes. They express this preference on the resource by voting. Community-based Question Answering service is a flourishing type of resource. Users resort to community help for a variety of reasons, from lack of proficiency in web search to seeking an answer from experience of other users. We implement our proposals and demonstrate the features in the platform, including trace visualization, and corresponding recommender results. It is a pity that

In the next section, we will present possible directions for future work based on work done in this thesis.

6.2 Perspectives and Future Work

To construct a good recommender system is never merely a technical problem. To achieve this goal, a wider range of issues should be considered. For example, recommendation should not go against users' innate preference. Several perspectives, in our opinion, represent a natural continuation of this work. This section is dedicated for a description of different aspects of such perspectives. Future work can be divided by industrial parts and scientific parts. Scientific parts include:

- Naturally, it is noticed that different types of activities have different weights on the inference of competency. For an apparent instance, creating a technical document that is highly appraised is of a great importance while deleting a resource hardly means anything. As presented in Section 3.2.4, the method Logistic Regression actually calculates the weights of all features (not limited to different types of activities, but also features in the CQA service, etc.) when dealing with user traces and proposing recommendations. Meanwhile, weights are yet not included in our semantic model. In the future, we expect to complete the model of activities with weights which can be updated by our recommender system.
- Apart from activities, one more important part of assigning weight is to assess weights of different users in the platform. As users competency vary, their activities and opinions (sharing, voting) also means different importance. A newcomer appreciates a high vote from an expert rather than a peer. Thus our future work also includes assigning, updating and taking into account the weights of users of different expertise.

- When we propose recommendations to users, these recommendations should never be dissociated with user preference. Completing user profiles with their preference, we could propose recommendations

more pertinent.

On the other hand, industrial parts include:

- In this thesis, user activities are divided in six categories based on their actions on resources in the platform. According to the state of the art, many researchers already extend the analysis of user activities by the level physical devices. This specification is oriented to a deeper level, for example to register the history of mouse click and typing on the keyboard. Traces like this will be much more enriched. For example, recording the pattern of typing on the keyboard, e.g., the delay between each type, the frequencies that a user corrects errors, we can further analyze whether a user is confident or not about the input of information he/she makes to the platform compared to his/her previous record/pattern. But at the same time, these traces need more sophisticated method to exploit. This is one of the directions for the future development of our semantic model as well as our analysis methods.

- Our method is applied on a open source dataset with which we try to simulate our ideal scenario of a collaborative working environment. However, applying our platform to obtain real dataset is a must for the future. Only in real scenario could a recommender system be tested and verified. For example, this platform can be used for education in a specific course in the university. By the time knowledge resources are shared, students activities and performance are registered in the platform. Student scores of this course can be compared with the competency evaluation results of our system. We can cooperate with famous MOOC websites such as Coursera and mooc.ca and provide corresponding service and help improve their quality of education.

- Many useful functions and applications can be further developed according to the competency recommender system to extend our platform. For instance, a resource allocation system can be build based on the result of recommendation. Further, we can imagine branching an access control system to our platform. Users are only allowed an access to certain resources according to his/her competency evaluated based on previous activities. Meanwhile, as a users participation on the platform goes on and his/her competency updated, higher and more sophisticated levels of resources will be gradually unlocked. This method will on the other hand encourage users participation and contribution

- When evaluating competency, our proposal only takes into consideration activities and their related concepts. In the future, as a variety of

techniques and strategies from domains are applied such as linguistics (e.g., natural language processing for transformation or implementation) and deep learning (e.g., Markov tree, pattern recognition), the recommendation results are expected to be largely improved.

Chapter 7

Publications

7.1 International Publications

- Ning Wang, Marie-Hélène Abel, Jean-Paul Barthès, and Elsa Negre. "An Answerer Recommender System Exploiting Collaboration in CQA Services." In IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2016), regular paper, pp. 198-203. Nanchang, China, May 2016.
- Ning Wang, Marie-Hélène Abel, Jean-Paul Barthès, and Elsa Negre. "Recommending Competent Person in a Digital Ecosystem." In IEEE International Conference on Industrial Informatics and Computer Systems (CIICS 2016), regular paper, pp. 37-43. Sharjah, United Arab Emirates, Mar 2016.
- Ning Wang, Marie-Hélène Abel, Jean-Paul Barthès, and Elsa Negre. "Recommending Competent Users from Semantic Traces Using a Bayes Classifier." In IEEE International Conference on Systems, Man, and Cybernetics (SMC 2015), poster, pp. 1351-1356. Hong Kong, China, Oct 2015.
- Ning Wang, Marie-Hélène Abel, Jean-Paul Barthès, and Elsa Negre. "Mining user competency from semantic trace." In IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2015), regular paper, pp. 48-53. Calabria, Italy, May 2015.
- Ning Wang, Marie-Hélène Abel, Jean-Paul Barthès, and Elsa Negre. "A Recommender System from Semantic Traces Based on Bayes Classifier." In International Conference on Knowledge Management, Information and Knowledge Systems (KMIKS 2015), regular paper, pp. 49-60. Hammamet, Tunisia, Apr 2015.
- Ning Wang, Marie-Hélène Abel, Jean-Paul Barthès, and Elsa Negre. "Towards a recommender system from semantic traces for decision aid." In International conference on Knowledge Management and Information Sharing (KMIS 2014), position paper, pp. 274-279. Rome, Italy, Oct 2014.

7.2 National Publications

- Wang Ning, Marie-Hélène Abel, Jean-Paul Barthès, and Elsa Negre. "Vers un Système de Recommandation à Partir de Traces Sémantiques pour l'Aide à la Prise de Décision." In INFormatique des ORganisation et Systèmes d'Information et de Décision (INFORSID 2014), poster, pp. 29-32. Lyon, France, May 2014.

Bibliography

Abel, Marie-Hélène, Adeline Leblanc, et al. (2009). *Knowledge sharing via the E-MEMORAe2. 0 platform*. Tech. rep.

Allee, Verna (1997). *The knowledge evolution: Expanding organizational intelligence*. Routledge.

Atrash, Ala (2015). "Modeling a System of Expertise Capitalization to Support Organizational Learning Within Small and Medium-sized Enterprises". PhD thesis. Université de Technologie de Compiègne.

Averell, Lee and Andrew Heathcote (2011). "The form of the forgetting curve and the fate of memories". In: *Journal of Mathematical Psychology* 55.1, pp. 25–35.

Ballesteros, I Laso and W Prinz (2006). "New collaborative working environments 2020". In: *Report on industry-led FP7 consultations and 3rd Report of the Experts Group on Collaboration@ Work, European Commission*.

Baugh, Jeremy (2000). "Rewarding competencies in flatter organizations". In: *Competency: the journal*.

Berkelaar, Brenda L (2014). "Cybervetting, Online Information, and Personnel Selection New Transparency Expectations and the Emergence of a Digital Social Contract". In: *Management Communication Quarterly*.

Beyou, Claire (2003). *Manager les connaissances:[du knowledge management au développement des compétences dans l'organisation]*. Liaisons.

Bi, Wei and James T Kwok (2011). "Multi-label classification on tree-and dag-structured hierarchies". In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 17–24.

Boyd, Carl R, Mary Ann Tolson, and Wayne S Copes (1987). "Evaluating trauma care: the TRISS method." In: *Journal of Trauma and Acute Care Surgery* 27.4, pp. 370–378.

Brusilovsky, Peter and Mark T Maybury (2002). "From adaptive hypermedia to the adaptive web". In: *Communications of the ACM* 45.5, pp. 30–33.

Burke, Robin (1999). "Integrating knowledge-based and collaborative-filtering recommender systems". In: *Proceedings of the Workshop on AI and Electronic Commerce*, pp. 69–72.

– (2000). "Knowledge-based recommender systems". In: *Encyclopedia of library and information science* 69.Supplement 32, p. 180.

– (2007). "Hybrid web recommender systems". In: *The adaptive web*. Springer, pp. 377–408.

Carreras, Martínez and AF Gómez Skarmeta (2006). "Towards interoperability in collaborative environments". In: *Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006. International Conference on*. IEEE, pp. 1–5.

Champin, Pierre-Antoine, Yannick Prié, and Alain Mille (2003). "Musette: Modelling uses and tasks for tracing experience". In: *ICCBR*. Vol. 3, pp. 279–286.

Chen, Wen-Yen et al. (2009). "Collaborative filtering for orkut communities: discovery of user latent behavior". In: *Proceedings of the 18th international conference on World wide web*. ACM, pp. 681–690.

Chen, Zhimin, Yi Jiang, Yao Zhao, et al. (2010). "A Collaborative Filtering Recommendation Algorithm Based on User Interest Change and Trust Evaluation." In: *JDCTA* 4.9, pp. 106–113.

Clauzel, Damien, Karim Sehaba, and Yannick Prié (2009). "Modelling and visualising traces for reflexivity in synchronous collaborative systems". In: *Intelligent Networking and Collaborative Systems, 2009. INCOS'09. International Conference on*. IEEE, pp. 16–23.

Cover, Thomas M and Joy A Thomas (2012). *Elements of information theory*. John Wiley & Sons.

Das, Abhinandan S et al. (2007). "Google news personalization: scalable online collaborative filtering". In: *Proceedings of the 16th international conference on World Wide Web*. ACM, pp. 271–280.

Deparis, Etienne (2013). "Création de nouvelles connaissances décisionnelles pour une organisation via ses ressources sociales et documentaires". PhD thesis. Compiègne.

Doran, Paul, Valentina Tamma, and Luigi Iannone (2007). "Ontology module extraction for ontology reuse: an ontology engineering perspective". In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, pp. 61–70.

Druzdzel, Marek J and Roger R Flynn (1999). "Decision support systems. Encyclopedia of library and information science. A. Kent". In: *Marcel Dekker, Inc. Last Login* 10.03, p. 2010.

Ebbinghaus, Hermann (1913). *Memory: A contribution to experimental psychology*. 3. University Microfilms.

Electrical Engineers, London (United Kingdom); Institution of (1999). *Safety, competency and commitment Competency guidelines for safety-related system practitioners*.

Felfernig, Alexander and Robin Burke (2008). "Constraint-based recommender systems: technologies and research issues". In: *Proceedings of the 10th international conference on Electronic commerce*. ACM, p. 3.

Felfernig, Alexander et al. (2007). "The VITA financial services sales support environment". In: *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*. Vol. 22. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 1692.

Freedman, David A (2009). *Statistical models: theory and practice*. cambridge university press.

Ghazanfar, Mustansar and Adam Prugel-Bennett (2010). "An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering". In:

Ghazanfar, Mustansar Ali, Adam Prügel-Bennett, and Sandor Szedmak (2012). "Kernel-mapping recommender system algorithms". In: *Information Sciences* 208, pp. 81–104.

Giddens, Anthony (1984). *The constitution of society: Outline of the theory of structuration*. Univ of California Press.

Gray, Barbara (1989). "Collaborating: Finding common ground for multiparty problems". In:

Guy, Ido, Inbal Ronen, and Eric Wilcox (2009). "Do you know?: recommending people to invite into your social network". In: *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, pp. 77–86.

Guy, Ido et al. (2010). "Social media recommendation based on people and tags". In: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 194–201.

Harrell, Frank E (2014). "Regression Modeling Strategies." In: *as implemented in R package 'rms' version* 3.3.

Harzallah, Mounira and François Vernadat (2002). "IT-based competency modeling and management: from theory to practice in enterprise engineering and operations". In: *Computers in industry* 48.2, pp. 157–179.

Jannach, Dietmar et al. (2010). *Recommender systems: an introduction*. Cambridge University Press.

Kandola, Pearn (2009). "Successful Video Communication". In: pp. 6–8.

Kim, Eugene E (2004). "A Manifesto for Collaborative Tools". In: *Doctor Dobbs Journal* 29.5, pp. 38–40.

Koren, Yehuda (2010). "Collaborative filtering with temporal dynamics". In: *Communications of the ACM* 53.4, pp. 89–97.

Laflaquiere, Julien (2009). "Conception de système à base de traces numériques dans les environnements informatiques documentaires". PhD thesis. Troyes.

Li, Qiang (2013). "Modeling and exploitation of the traces of interactions in the collaborative working environment". PhD thesis. Université de Technologie de Compiègne.

Li, Qiang, Marie-Hélène Abel, and Jean-Paul Barthès (2012). "Sharing working experience: Using a model of Collaborative Traces". In: *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*. IEEE, pp. 221–227.

Liang, Bin (2007). *Stepping in to search engine*. Publishing House of Electronics Industry.

Liu, Qiaoling et al. (2011). "Predicting web searcher satisfaction with existing community-based answers". In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, pp. 415–424.

Lomas, Cyprien, Michael Burke, and Carie L Page (2008). "Collaboration tools". In:

London, Scott (1995). "Building collaborative communities". In: *Pew Partnership for Civic Change*.

Lund, Kris and Alain Mille (2009). "Traces, traces d'interactions, traces d'apprentissages: définitions, modèles informatiques, structurations, traitements et usages". In: *Analyse de traces et personnalisation des environnements informatiques pour l'apprentissage humain. Hermès*, pp. 21–66.

Majchrzak, Ann et al. (2000a). "Computer-mediated inter-organizational knowledge-sharing: Insights from a virtual team innovating using a collaborative tool". In: *Information Resources Management Journal* 13.1.

Majchrzak, Ann et al. (2000b). "Technology adaptation: The case of a computer-supported inter-organizational virtual team". In: *MIS quarterly* 24.4, pp. 569–600.

Manning, Christopher D, Prabhakar Raghavan, Hinrich Schütze, et al. (2008). *Introduction to information retrieval*. Vol. 1. 1. Cambridge university press Cambridge.

Marrelli, Anne F (1998). "An introduction to competency analysis and modeling". In: *Performance Improvement* 37.5, pp. 8–17.

Martínez-Carreras, M Antonia et al. (2007). "Designing a generic collaborative working environment". In: *Web Services, 2007. ICWS 2007. IEEE International Conference on*. IEEE, pp. 1080–1087.

Martinez-Moyano, I (2006). "Exploring the dynamics of collaboration in interorganizational settings". In: *Creating a Culture of Collaboration: The International Association of Facilitators Handbook* 4, p. 69.

Miller, Bradley N et al. (2003). "Movielens unplugged: Experiences with a recommender system on four mobile devices". In: *In Proceedings of the 2003 Conference on Intelligent User Interfaces*. Citeseer.

Mirzadeh, Nader, Francesco Ricci, and Mukesh Bansal (2005). "Feature selection methods for conversational recommender systems". In: *2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*. IEEE, pp. 772–777.

Mooney, Raymond J and Loriene Roy (2000). "Content-based book recommending using learning for text categorization". In: *Proceedings of the fifth ACM conference on Digital libraries*. ACM, pp. 195–204.

Negre, Elsa (2016). *Information and Recommender Systems*. John Wiley & Son.

Pathak, Jyotishman, Thomas M Johnson, and Christopher G Chute (2009). "Survey of modular ontology techniques and their applications in the biomedical domain". In: *Integrated computer-aided engineering* 16.3, pp. 225–242.

Paul, Sharoda A, Lichan Hong, and Ed H Chi (2012). "Who is authoritative? understanding reputation mechanisms in quora". In: *arXiv preprint arXiv:1204.3724*.

Premchaiswadi, Wichian, Anucha Tungkasthan, and Nipat Jongsawat (2010). "Enhancing learning systems by using virtual interactive classrooms and web-based collaborative work". In: *Education Engineering (EDUCON), 2010 IEEE*. IEEE, pp. 1531–1537.

Prié, Alain Mille-Yannick (2006). "Une théorie de la trace informatique pour faciliter l'adaptation dans la confrontation logique d'utilisation/logique de conception". In: *Cité en*, p. 29.

Prinz, Wolfgang et al. (2006). "ECOSPACE? Towards an Integrated Collaboration Space for eProfessionals". In: *2006 International Conference on Collaborative Computing: Networking, Applications and Worksharing*. IEEE, p. 39.

Raban, Daphne Ruth (2009). "Self-presentation and the value of information in Q&A websites". In: *Journal of the American society for information science and technology* 60.12, pp. 2465–2473.

Rajaraman, Anand et al. (2012). *Mining of massive datasets*. Vol. 1. Cambridge University Press Cambridge.

Ricci, Francesco and Quang Nhat Nguyen (2007). "Acquiring and revising preferences in a critique-based mobile recommender system". In: *IEEE Intelligent systems* 22.3, pp. 22–29.

Ricci, Francesco, Lior Rokach, and Bracha Shapira (2011). *Introduction to recommender systems handbook*. Springer.

Rothwell, William J and Hercules C Kazanas (2011). *Mastering the instructional design process: A systematic approach*. John Wiley & Sons.

Sehaba, Karim (2012). "Partage d'expériences entre utilisateurs différents: Adaptation des modalités d'interaction". In: *IC 2011, 22èmes Journées francophones d'Ingénierie des Connaissances*, pp. 639–656.

Settouti, Lotfi Sofiane et al. (2009a). "A trace-based system for technology-enhanced learning systems personalisation". In: *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on*. IEEE, pp. 93–97.

Settouti, Lotfi Sofiane et al. (2009b). "A trace-based systems framework: Models, languages and semantics". In:

Shani, Guy and Asela Gunawardana (2011). "Evaluating recommendation systems". In: *Recommender systems handbook*. Springer, pp. 257–297.

Sigurbjörnsson, Börkur and Roelof Van Zwol (2008). "Flickr tag recommendation based on collective knowledge". In: *Proceedings of the 17th international conference on World Wide Web*. ACM, pp. 327–336.

Smyth, Barry et al. (2004). "Compound critiques for conversational recommender systems". In: *Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on*. IEEE, pp. 145–151.

Sparck Jones, Karen (1972). "A statistical interpretation of term specificity and its application in retrieval". In: *Journal of documentation* 28.1, pp. 11–21.

Swenson, Keith D (1994). "The Future of Workflow Technology: Collaborative Planning". In: *Proc. Conf. on Groupware (Groupware'94), San Jose, Kalifornien*.

Terveen, Loren and Will Hill (2001). "Beyond recommender systems: Helping people help each other". In: *HCI in the New Millennium* 1, pp. 487–509.

Thompson, Cynthia A, Mehmet H Goker, and Pat Langley (2004). "A personalized system for conversational recommendations". In: *Journal of Artificial Intelligence Research* 21, pp. 393–428.

Thompson, John N, Scott L Nuismer, and Richard Gomulkiewicz (2002). "Coevolution and maladaptation". In: *Integrative and Comparative Biology* 42.2, pp. 381–387.

Tribus, Myron (1961). *Thermostatics and thermodynamics*. Center for Advanced Engineering Study, Massachusetts Institute of Technology.

Tullio, Joe and Elizabeth D Mynatt (2007). "Use and implications of a shared, forecasting calendar". In: *Human-Computer Interaction–INTERACT 2007*. Springer, pp. 269–282.

Vasconcelos, J, Chris Kimble, and A Rocha (2003). "Ontologies and the dynamics of organisational environments: an example of a group memory system for the management of group competencies". In: *Proceedings of I-Know*. Vol. 3, pp. 2–4.

Waggener, Shelton et al. (2009). "Collaborative Tools Strategy". In:

Walker, Strother H and David B Duncan (1967). "Estimation of the probability of an event as a function of several independent variables". In: *Biometrika* 54.1-2, pp. 167–179.

Weiseth, Per Einar et al. (2006). "The wheel of collaboration tools: a typology for analysis within a holistic framework". In: *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. ACM, pp. 239–248.

Wendel, Viktor et al. (2012). "Designing collaborative multiplayer serious games for collaborative learning". In: *Proceedings of the CSEDU* 2012.

Whittaker, Steve, Erik Geelhoed, and Elizabeth Robinson (1993). "Shared workspaces: how do they work and when are they useful?" In: *International Journal of Man-Machine Studies* 39.5, pp. 813–842.

Zarka, Raafat et al. (2012). "Trace replay with change propagation impact in client/server applications". In: *IC 2011, 22èmes Journées francophones d'Ingénierie des Connaissances*, pp. 607–622.

Zhang, Yuchuan and Yuzhao Liu (2010). "A collaborative filtering algorithm based on time period partition". In: *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on*. IEEE, pp. 777–780.