ONTOLOGY-DRIVEN PERSONALIZED EXPERT RECOMMENDER SYSTEM FOR IT SERVICE MANAGEMENT

Karuna Sagara Wasala Kelaniya Bandarage Lihini Chamalka

199462V

Dissertation submitted in partial fulfillment of the requirements for the degree Master of Science in Artificial Intelligence

Department of Computational Mathematics

University of Moratuwa

Sri Lanka

July 2022

Declaration

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name of the Student: K.S.W.K.B.L. Chamalka Signature of the Student

Date:

The above candidate has carried out research for the Masters dissertation under my supervision.

Name of the Supervisor: Dr. A.T.P. Silva Signature of the Supervisor

Date:

Dedication

I dedicate this work to my dear parents who are always supporting and encouraging me to achieve better in academics.

Acknowledgements

I would like to express my special gratitude and thanks to my supervisor Dr. Thushari De Silva for guiding me throughout this research work and encouraging me to improve as a research scientist. Also, I would like to thank Professor Asoka S. Karunananda for his constructive advice that had a positive impact on both my research as well as career.

My special thanks go to all the other lecturers and non-academic staff who have helped me in many ways during this MSc in AI course.

Then I would like to thank the University of Moratuwa for giving me an opportunity to engage in this research work and to produce this dissertation.

Last but not least, I would like to thank my parents, friends, and colleagues, without them it would not possible.

Abstract

Finding experts related to a given query in an industrial environment is a time-consuming manual task. Much research has been conducted in this area using multiple intelligent techniques, but still, there are research gaps with personalizing the recommendation accurately. In this context, an expert recommender system should consider the expert's preference, experience, and other factors as well as complex organizational processes involved in the recommendation task. Also achieving high accuracy with other conflicting conditions simultaneously is a popular topic in recent research related to recommender systems.

This thesis presents our hybrid approach to enhance the personalized expert recommendation problem in enterprise context. We integrate semantic-based ontology with the TOPSIS based Artificial Bee Colony algorithm to achieve high accuracy in this problem domain. Ontology is used for knowledge modeling of the expert profiles and the TOPSIS-ABC algorithm is used for ranking the profiles for a given query based on the distance to the ideal solution.

Key words: Expert Recommender Systems, Multiobjective Optimization, ABC Algorithm, TOPSIS Method

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
ABC	Arificial Bee Colony
ACM	Association for Computing Machinery
ACO	Ant Colony Optimization
ACT	Author Conference Topic
ADJ	Adjective
ADV	Adverb
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
API	Application Programming Interface
APT	Author Persona Topic
ARM	Author and Co-Author Relationship Model
ARS	Answerer Recommendation System
ASN	Academic Social Network
ATM	Author Topic Model
AWS	Amazon Web Services
BAT	Bat Algorithm
BOW	Bag of Words
CONJ	Conjunctions
CPU	Central Processing Unit
DBLP	Digital Bibliography & Library Project
DE	Differential Evaluation
DEMOIR	Dynamic Expertise Modeling from Organizational Information Resources
DL	Description Logic
ELK	ELK is a reasoner for OWL 2 ontologies that currently supports a part of the OWL EL ontology language
ERS	Expert Recommender System
FSS	Fish School Search
GSA	Gravitational Search Algorithm
HITS	Hyperlink-Induced Topic Search
ID	Identity
TF-IDF	Term Frequency–Inverse Document Frequency
IT	Information Technology
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
IWO	Invasive Weed Optimization
JIRA	Proprietary Bug Tracking Tool
JQL	Jira Query Language
L1	Level 1 support

L2	Level 2 support
L3	Level 3 support
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
LSI	Latent Semantic Index
MAP	Mean Average Precision
MITRE	American not-for-profit organization
NER	Named Entity Recognition
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NMF	Non-negative Matrix Factorization
ORG	Organization
OS	Operating System
OWL	Web Ontology Language
PCA	Principal Component Analysis
PLSA	Probabilistic Latent Semantic Analysis
POC	Proof of Concept
POS	Part-of-speech
PREP	Prepositions
PSO	Particle Swarm Optimization
RAF	Research Analytics Framework
RAM	Random Access Memory
RDF	Resource Description Framework
RP	Random Projection
RR	Reciprocal Rank
RRF	Reciprocal Rank Fusion
RTM	Rough Threshold Model
SI	Swarm Intelligence
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
TRM	Topic-document Relationship Model

CHAPTER 1

1. INTRODUCTION

1.1 Prolegomena

One of the most challenging problems faced by large-scale IT service management in the industry is the very high number of support tickets being raised daily in the helpdesk. There is a limited number of support agents available to attend to the incidents and they have different domain knowledge, skills, and expertise levels. Currently, it is a manual task to find the best support agent who is capable of resolving the incident, when a fresh support ticket is raised in the helpdesk. Sometimes support tickets are routed between multiple support agents and it becomes a tedious task. This results in a high mean time to resolution and ultimately lower customer satisfaction.

Expert recommender systems help users to find the best experts for a given topic or query. However, Expert recommendation in a personalized manner has still been a research challenge in industrial application domains. Also, there are research opportunities identified in evaluating the expertise under multiple attributes of the expert profiles in the recommendation process. Our solution for this problem is to integrate an expert recommender system with the helpdesk to recommend the best support agent in a personalized manner using their profile details, ticket details, and similar past tickets for a faster resolution. We mainly focus on improving the accuracy and computational efficiency of the expert recommendation by using a hybrid approach combining ontology-driven semantic expert profile modeling with artificial bee colony optimized ranking.

The rest of the chapters are organized with the sections, namely, Aim & Objectives, Background & Motivation, Problem Definition, and A Novel Approach to ERS, Resource Requirement and Structure of the thesis with the summary for the chapter.

1.2 Aims and Objectives

This research aims to address the challenges in extending the personalized expert recommendation problem in the IT service management domain and improve the accuracy and efficiency of the algorithm.

Objectives:

1. Critically review the literature related to Expert Recommender Systems
2. In-depth study of technology, algorithms, and theory to solve Personalized Expert Recommendation
3. Design and develop a hybrid approach by integrating Ontology and Artificial Bee Colony algorithm for Personalized Expert Recommendation
4. Evaluate the solution using appropriate metrics

1.3 Background & Motivation

The expert recommendation is a popular topic among academia to find researchers and related articles. For an instance, ArnetMiner, ExpertSeer are public expert search systems related to computer science research and we can find the experts for a given topic. Also, some commercial-level applications are available such as LinkedIn, Google Careers, CareerBuilder, Indeed, etc. that allow job seekers to search for jobs. Currently, potential employers are finding suitable job candidates by searching on these platforms as well [1].

ACM RecSys community organizes an annual challenge to motivate the researchers under recommender systems since 2007 and it also has contributed to the progress of this area considerably.

1.4 Problem Definition

Identifying the most suitable experts considering multiple factors such as expertise, experience, workload and preference to do a given task with minimal resources has been a research challenge in industrial domains.

First, we will give a brief overview of the incident management procedure under the IT service management as shown in figure 1.1. When an application issue is observed by the end-user, he will report the problem in the helpdesk tool provided by the organization. For this purpose, he will create an issue ticket under the relevant system category and the issue type. Sometimes the categorization of the issue will be irrelevant due to the lack of knowledge of the end-user. After that, he will be provided with a form and he reports the summary, description, and other details required, attach images and other content and finally submit the ticket.

After that L1 support agents assess the issue, fix the customer incident, and assign the ticket to a domain expert in L2/L3/Design team for a permanent resolution. The L1 task of assessing the issues and identifying the relevant expert is a time-consuming manual task which is the problem of our interest.

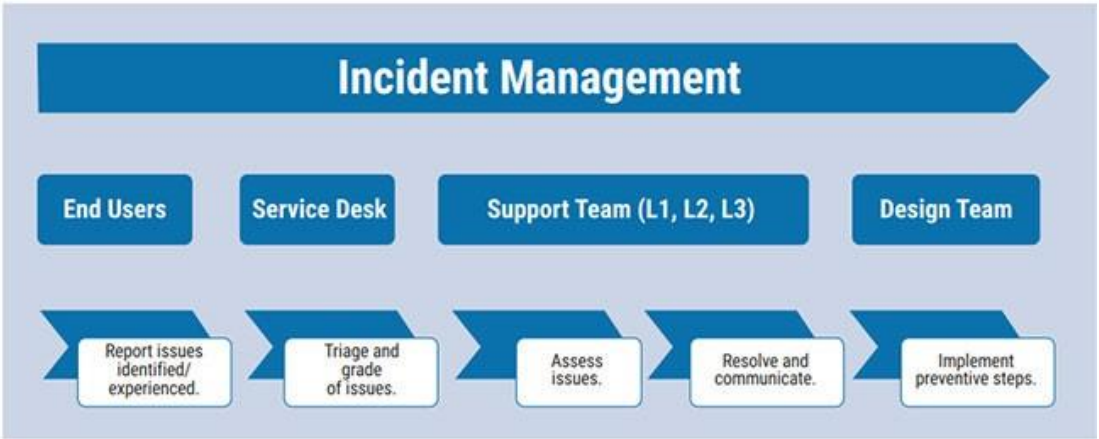


Figure 1.1: Incident Management Process of ITSM

Source: https://www.isaca.org/resources/isaca-journal/issues/2019/volume-3/incident-management-for-erp-projects

Therefore this research focuses on how to solve the expert recommendation problem in a personalized manner with high accuracy and efficiency in the IT service management domain. Every application domain has a specific set of problems that are required to be addressed in the recommendation task, hence it is impossible to apply a generalized recommendation approach to each area. In this research, we study the challenges in extending the personalized expert recommender system approach for IT Service Management. The solution requires high accuracy, efficiency and there is risk associated with inaccurate recommendations which lead to customer dissatisfaction.

1.5 A Novel Approach to Expert Recommender Systems

This research hypothesizes that ontology-driven semantic-based expert profiling combined with TOPSIS-ABC optimized expert ranking can provide a better solution with high accuracy and efficiency in the personalized expert recommendation task.

The proposed solution is to apply a hybrid filtering approach to recommend the best experts. Expert profiles including skills, domain knowledge, and attributes of the tickets will be the inputs to the model and the output will be a ranked list of the N best experts. An ontology-driven semantic-based approach will be used for knowledge modeling and profiling of the experts. Domain ontology reasoning will reduce the search space based on the semantic similarity of the agent and ticket profiles and will improve the accuracy as well. Then TOPSIS-ABC optimization algorithm will be used for ranking the best expert profiles in a personalized way. Considering only the neighborhood expert profiles in the recommendation process will result in faster results and higher efficiency.

1.6 Resource Requirement

- Software Requirement:
  Python libraries, Protégé framework, Jupyter Notebook, PyCharm IDE

- Hardware Requirement:
  Computer with Windows OS, 2.6 GHz Core, 8GB RAM for POC setup
  Google Colaboratory or AWS Cloud Computing Services depending on the processing power requirement

- Dataset Requirement:
  Helpdesk tickets dataset, Expert skills dataset

1.7 Structure of the Thesis

This introduction chapter provided an overview of this research and chapter 2 will present the literature review of the expert recommender systems, existing techniques, challenges, and the research problem. Then chapter 3 will explain the technologies in-depth applicable for expert recommendation and chapter 4 will present the novel hybrid approach proposed in this research. Then chapter 5 presents the design including the architecture and modules, chapter 7 and chapter 8 present the evaluation and conclusion respectively.

CHAPTER 2

2. LITERATURE REVIEW

2.1 Introduction

In Chapter 1, we provided an overall picture of the research presented in this thesis. This chapter gives a critical review of literature in Expert Recommender Systems (ERS). In doing so, we have structured the literature review under several subheadings, namely, State-of-the-art Techniques in ERS, ERS and Early Developments, and Latest Trends in ERS. This chapter also summarizes challenges in ERS and defines the research problem. The research gap identified is the inadequate application of ERS in industrial domains which depends on multiple factors and complex organizational processes. Also Improving the accuracy and efficiency of the ERS algorithm for such domains in a personalized manner is the research problem.

2.2 State-of-the-art Techniques in ERS

An expert is a person who has mastered a specific knowledge area and is capable of guiding or answering other people. The expert recommendation is a subcategory of recommender systems, that recommend people-to-people. In other words, the purpose of the expert recommendation is to predict experts to users relevant to their queries. The experts are modeled as a weighted set of skills and knowledge in their domains. The ERS takes the input query from the user and matches the relevant past queries. Then it predicts similar experts using the relevancy and expert classification is provided in the expertise model. Finally, the candidate experts are ranked according to their competency levels [2].

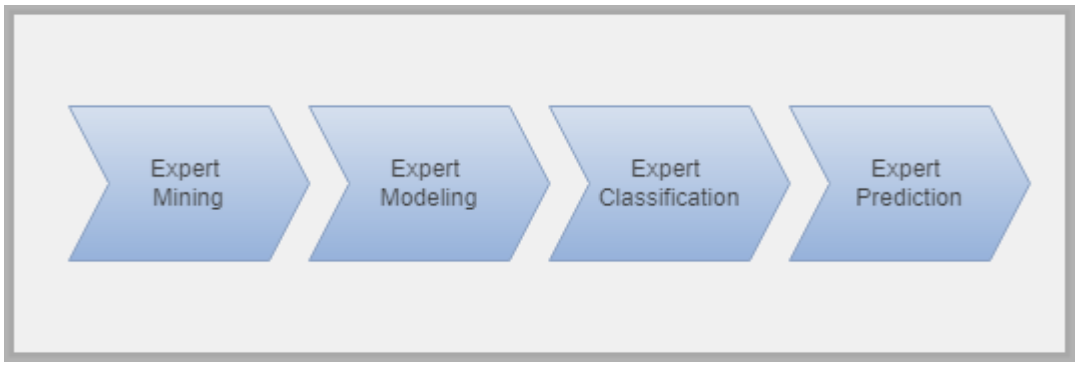


Figure 2.1: Major Steps in Expert Recommender System Design

There were only very few surveys conducted on the state-of-the-art techniques used in expert recommender systems compared to recommender systems in general. We have studied the classification, techniques used in different phases, applications, evaluation metrics, and current challenges in this research area and the subsequent paragraphs will discuss the points in detail.

According to the review of Lin and others, expert finding is a four-step procedure namely expertise resource selection, expertise modeling, query matching, and result recommendation. They have discussed the techniques, retrieval algorithms, and evaluation metrics of the first two steps in this ERS process [3]. The challenges encountered in areas of weighing different expertise levels, clustering experts based on similarity, discovering experts knowledgeable in cross-domain, and classifying knowledge into subdomains. Also, they have emphasized that evaluation of ERS highly depends on the test data used. These datasets which are publicly available or taken from other parties are not reliable hence evaluating retrieval results accurately is a problem.

Husain and others had reviewed the literature on ERS under five main research questions, domains applied, expertise source used, expertise retrieval versus expertise seeking and theories integrated, methods used for expert finding, and dataset used in these systems [4]. They have identified main domains such as enterprise, academia, knowledge-sharing platforms, social networks, online forums, and medicine. The majority of the research was limited to the academic domain under various tasks to

find research supervisors, paper reviewers, and collaborators from the industry, thus encouraging researchers to extend ERS in other domains like medicine. They have proposed publishing benchmark data sets to avoid the issues in evaluating the accuracy of the ERS models while using unreliable datasets.

The authors have done a comprehensive classification of expertise recommendation models mainly including probabilistic, voting, network-based (mentioned as graph-based), and other models [2]–[4] as shown in Figure 2.1. Further, the Generative Probabilistic Models were classified into Candidate and Topic Generation models. Other models include author-topic models which represent experts as a distribution of topics and determine the association of the new query topic to the contributed topics of the experts. In addition to this authors [4] have mentioned another type called Hybrid models for combined approaches. Hybrid models are used to overcome the drawbacks of using a single technique in the expert recommendation process. For instance, integrating content-based and collaborative filtering helps in finding experts via social networks connections, and enables access to a wide range of heterogeneous information sources.

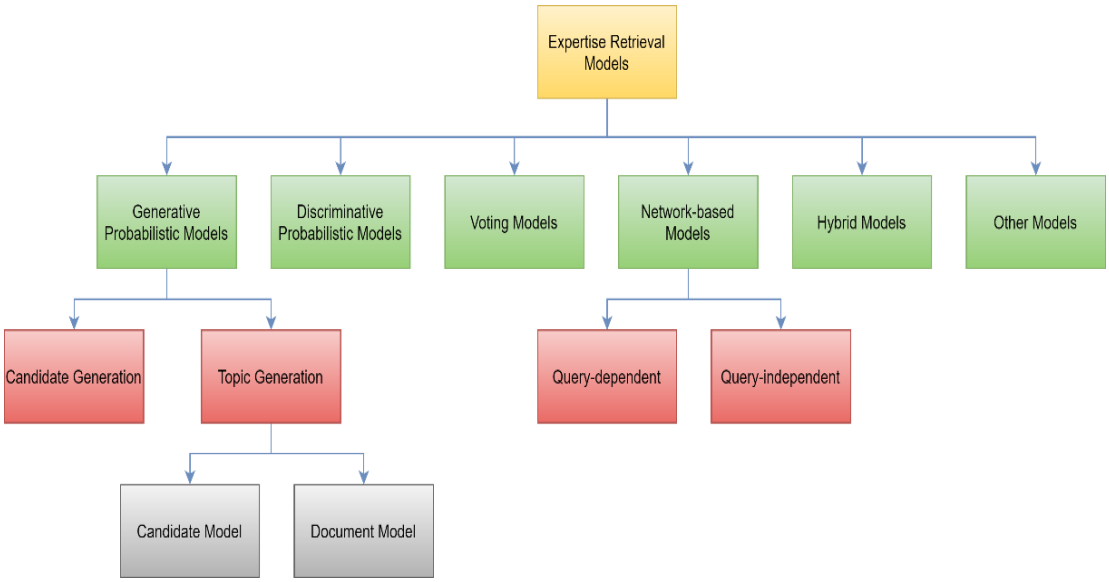


Figure 2.2: Classification of Expert Recommendation Techniques

Now we provide a brief overview of the state-of-the-art expert recommender models found in the literature.

2.2.1. Generative probabilistic models

We use probability to rank the experts in generative probabilistic models. Probability is taken as the likelihood of a candidate expert related to the query. There are two main types of generative probabilistic models namely, candidate generation and topic generation. Most of the approaches have adapted generative probabilistic models due to their promising performance in mining expertise from documents. The generative probabilistic method is the foundation of many language models used in expert recommender tasks.

2.2.2. Voting models

In voting models, we use data fusion techniques to aggregate the votes for experts and provide the final ranked list. Reciprocal Rank (RR), CombSUM, and CombMNZ are popular fusion techniques used in the expert search. Researchers have shown that voting models are also similar to probabilistic models. HITS, PageRank algorithms, and random walk propagation methods are used in voting models. In the voting model scores are not normalized as probabilities in generative models.

2.2.3. Network-based models

Network-based models collect expert information from the web and social networks connections. This is also known as the graph-based method and exploits the idea that experts with similar skills are closely related to each other. There are two subcategories namely query dependant graph-based model and query-independent graph-based model. In the query-dependent process, the documents are searched within the most related communities to the expert's scope and in the query-independent process, a particular expert knowing a specific area is assumed.

2.2.4. Other models

In other models, existing information retrieval techniques were discussed such as author-topic modeling, vector space modeling, and cluster-based retrieval. For instance, the Latent Dirichlet Allocation (LDA) based author-topic-model (ATM), author-persona-topic (APT) model and author-conference-topic (ACT) model are extensions of author-topic modeling which are based on probability distributions of authors and documents. In vector space modeling, experts are modeled as vectors that consist of a linear combination of documents. Cluster-based retrieval is used to group similar experts and ranking, which is the same as the aggregation techniques [2].

2.2.5. Hybrid models

The hybrid models are combinations of content-based filtering and collaborative filtering algorithms. Most of the recent research has combined the techniques explained earlier and proposed hybrid expert recommender systems as per the literature review [4]. For example, some hybrid approaches included the techniques such as,

a) Content-based recommendation and network-based collaborative filtering
b) Social network analysis and semantic web
c) Document-based model and community-sensitive AuthorRank
d) Cluster-based expert selection and collaborative filtering

Nikzad–Khasmakhi had proposed two phases in the expert recommendation process, information retrieval of the expert and predicting the expertise level of the expert and ranking [1]. The survey was done under key areas related to ERS like information source types, methods of information retrieval, approaches used in expert prediction, and evaluation metrics. It has presented a critical review on different techniques used in predicting expertise scores such as rule-based model, propagation, link analysis, language model, K-means, and deep learning as shown in table 2.1. Application of

other technologies like evolutionary algorithms, fuzzy logic, semantic web & ontology, swarm optimization, and multi-agent techniques were not adequately discussed in this review. Hence we have extended the list under these techniques by reviewing the pros and cons of the applicability in ERS.

Limited usage and analysis of multimedia content in expert mining and extending ERS into various domains were the major challenges identified in this survey. According to the type of application, the importance of the expertise scores varies. In that sense, researchers have proposed approaches that use a non-linear combination of these scores to improve the accuracy of expert prediction. Therefore expert recommendation process can be considered as a multi-objective optimization problem. This is another finding emphasized in this review.

Table 2.1: Existing Expert Recommendation Techniques

Technique	Advantages	Disadvantages
Rule-based model	Accurate results, less error rate, cost-efficient	Less learning capacity, complex pattern discovery, need for manual changes
Propagation	Handling finding the top-ranked experts by setting a relevance threshold	Time-consuming, a heavy overhead because of redundant score messages
Link analysis	Modeling users' activities and their relationships in social network	The cost of building a dynamic user-user graph, an extensive amount of data preparation
Language model	Solving the issue of text representation and term weighting, working with little corpus data	Computational complexity
K-Means	Easy to implement, grouping experts	Difficult to predict the number of clusters, sensitive to scale
Matrix completion	Estimating missing values in the rating matrix	The high computational cost for the determinant matrix calculation
Vector Space Model	Simple in implementation, normalizing the resulting probabilities	Not considering semantic similarities, the relationship between terms

Reputation score	Considering the quality of user's answers in ranking experts	More efforts are needed to determine the best answer
Deep learning	Automatic feature extraction, good accuracy in question answering matching	The high computational cost of training, complex error space
Ontology	Describe various features and attributes in a particular domain with a set of properties, that can be reused and shared by other systems	Inability to provide well optimized best answer from a list
Fuzzy Logic	Handle uncertainty and ambiguity in the relevance feedback	Not suitable for unstructured data or undefined processes

Sources: [1], [5], [6]

2.3 ERS and Early Developments

Many researchers have contributed to the enhancement of the aforementioned stages in the expert recommendation process and we critically reviewed these approaches under the main fields of enterprise, academia question answering platforms. The following sections are divided as the early developments and the latest trends in expert recommender systems considering the published years of the research papers or the journal articles.

Research in expert recommender systems for organizational environments emerged in the early 2000s. The researchers were more focusing on identifying the proper information sources for expertise mining and a generalized architecture for expert finding in the enterprise domains. They have discussed the technical features of such a design and the challenges in implementing it at an operational level.

McDonald and Ackerman had notable work in Expertise Recommenders, ER-Arch for organizational environments which is adaptable for different structures [7]. It is proved that a general architecture can be used in any industrial domain with proper expert profiling and selecting techniques. Also, it clusters the expert profiles using work products and can provide alternative recommendations. However, the

implementation was more tedious due to a large number of Java classes used in the code for simplification of adding new profiling and selection methods according to the organizational hierarchies.

As stated above the first step in the expert recommendation process is to model the profiles of experts in required knowledge areas. Yimam and Kobsa have categorized expert profiling based on query generation time and location. The DEMOIR hybrid architecture proposed by them uses different information sources to collect data for expertise modeling and stores them in a centralized location in a form of organizational ontology [8]. It used documents and content analysis as well, but the classification of documents into three main types was a manual task. Also, the efficiency of the algorithms in mapping the matrices was not adequately discussed.

MITRE's Technical Report on Expert Finding Systems discusses tools that were commercially available at that time under five main criteria namely source of information, type of processing, method of searching, results format, and system properties [9]. Among the challenges listed in the report, limited access to the experience of past experts, lack of knowledge of new experts, privacy concerns related to sharing the expert details, and difficulty in finding the most suitable expert are the most significant.

This research has answered challenges in extracting expertise from the document collections of an organization [10]. It has represented experts using documents in the enterprise corpora and determined candidate-document associations. Matching the expert's demographic information such as ID, Name, and Email was done using a binary rule-based method. It compares two generative probabilistic language models centered on documents and candidates under metrics like Mean Average Precision (MAP) by changing the β parameter in Bayes smoothing. Still, there is an opportunity for improvements in text preprocessing of the document extraction.

Reichling has designed a more sound expert recommender framework for a networking company and proposed a list of technical functionalities of the system

design to address the requirement and challenges in extending expert recommendation in a complex organizational setting [6,7]. Compared to other ERS frameworks, a feedback component is introduced to enhance the reliability of the expert recommendation. They have applied three matching strategies, Latent Semantic Indexing (LSI) for expertise mining from the textual content, simple keyword matching, and TF/IDF matching for personal data and feedback data respectively. The major drawback of this framework is that the most skilled experts will get an additional workload out of their scope due to high competency.

A structural way of modeling academic content available on the web (DBLP) using a domain ontology was proposed in this research [5]. This method is reusable and can be extended in other domains as well for expert finding and ranking. The authors present a comprehensive approach in applying ontological features including classes, properties, and instances to model concepts, relations, and individuals respectively in the research field. Also, they have constructed an Academic Social Network (ASN) which consists of two models, Topic-document Relationship Model (TRM) and Author and Co-Author Relationship Model (ARM) to measure the level of relationships between documents and authors.

The main focus of this paper is to mine the skills and competencies of experts and maintain the changes of the expert profiles over time using content created by them. The information sources considered are research papers, wiki pages, blogs, comments in online question-answering forums, and enrollment in academic or professional courses [13]. They show the importance of using labels (novice, advanced beginner, competent, proficient, and expert) for different proficiency levels rather than ratings that have no explainable facilities. Different sources are integrated and the activities are monitored, then the change in information is stored in a skills ontology along with the organization ontology.

The authors have presented a framework for extracting author information from wiki pages using domain ontologies [14]. Three scores are computed expertise, relevance, and reputation based on the contribution given for a topic, semantic similarity for the

available topics in the ontology, and the results of peer-reviewing respectively. Part-of-speech (POS) tagging and stemmers were applied in text processing and TF-IDF to capture the relevance of the documents. In the ontology owl:subClassOf and owl:ObjectProperties were used to match the similar concepts. They have proposed a trust function to determine the reputation score which increases in the beginning and then remains constant with time. This solution has proved promisingly accurate results in evaluation but is limited only to enterprises maintaining a proper Wiki system. Further enhancements are proposed in modeling the experts in multi-dimensions.

2.4 Latest Trends in ERS

In this section, we are discussing research papers published since 2010 in the expert recommendation research area that has contributed noticeable work for the progress of the field. Most of the work has employed hybrid approaches combining multiple factors and various aspects for expert profiling. The following paragraphs comprise a summary of these hybrid approaches formulated according to their application domains.

Silva proposed a social network empowered research analytics framework (RAF) to assign the most suitable reviewer in the research proposal evaluation process, considering multiple aspects of quality, quantity, and social relationships [15]. They have modeled the experts in three dimensions using relevance, productivity, and connectivity indices. Also, presented a detailed approach for researcher and proposal profiling using the self-declared discipline codes, title, keywords, and abstract of the proposals. For textual content analysis, they have employed the Rough Threshold Model (RTM) and Database Tomography in the algorithm development. This solution has proved cost reduction and efficiency in the reviewer finding process of the largest project funding organization, which has been a time-consuming manual task otherwise. In this approach, the reviewer's preference was not considered and there was no mechanism to validate the quality of reviews, although the authors have

concluded that this assignment problem can be solved as a multi-objective decision problem.

The framework in the previous study was further extended in the profile boosted RAF for recommending journals for manuscripts in the scientific community [16]. This is the first research work contributed in the field of scientific product recommendation. They have enhanced the Rough Threshold Model with a key phrase analysis algorithm to handle the semantic ambiguity observed in the original version. To obtain the journal quality the Analytical Hierarchy Process (AHP) was implemented and compared the importance of journals pairwise. Also, formulated the journal recommendation problem as a multiobjective optimization problem combined with a heuristic approach. The particle swarm optimization technique has evidence of low computational costs and fast convergence which is beneficial in a large candidate journal base. Since the journal selection is a highly subjective matter user's preference should be taken into account, but this approach has not demonstrated that aspect. Also, the cold start problem was not properly handled in the connectivity index for new users.

The authors have broadened the big data perspective for the researcher modeling in this study and presented Leverage RAF to find domain experts on research social network services [17]. The architecture has three main stages for profiling, modeling, and ranking. The modeling stage consists of three modules for relevance, quality, and connectivity analysis. The collected expert details are represented using correlation matrices Keyword-Document matrix, Article-Journal matrix, Project-Type matrix, and researcher-researcher matrix respectively. In the ranking stage, Analytic Hierarchy Process (AHP) is employed for aggregating the weighted scores. The MapReduce algorithm presented in this work is used to compute the similarity between the documents and authors and the results have proved its better performance in efficiency and scalability in large-scale datasets. The authors have suggested using other data fusion techniques than AHP, considering more subjective factors, and integrating a domain ontology for further improvements in the framework.

The authors in [18] have applied the Evidential Reasoning (ER) rule-based method as the aggregating technique in the research project evaluation and selection process. Peer reviews on research projects include highly subjective and qualitative information so, they have introduced a belief structure to represent evaluation information of the projects and a confusion matrix to generate expert reliabilities. As per the study, simple weighted sum techniques provide only a single score to differentiate the candidate solutions. Therefore they state the necessity of more compelling aggregation techniques to utilize the richness of the profile details available in multiple perspectives. This work has contributed to the research area by providing a rational approach in multi-criteria decision making with conflicting interests still, the expert preferences were not incorporated in the approach.

The performance of ERS heavily relies on the soundness of the expert profiling mechanism. Hence a lot of expert data needs to be gathered from various sources to accommodate the dynamics of expert behavior which ultimately require big data-based approaches for generating meaningful insights. The authors have proposed a method for real-time expert profiling using big data based on Map-Reduce technology for a research social network ScholarMate [19]. They have applied Author Topic Model (ATM) to extract expertise information from a huge collection of textual documents. Also, crowdsourcing techniques were utilized in mining social reviews of expertise. In the ranking phase, the entropy-based aggregation method was used to merge two profiles, Entity-Profile and Social-Profile. The limitation in this study is using the sampling method for the evaluation of the prototype, where big data solutions should experiment with large-scale datasets.

A Research Analytics Framework for Education (RAF-E) was proposed by the authors in this paper, which has modeled experts in three dimensions by the scores obtained in relevance, connectivity, and quality [20]. The solution was developed for identifying the most suitable research supervisor for novice students and they have integrated both direct and indirect means of relevance with past students. In the proposed approach there are two stages filtering and ranking. The relevance is computed in the first stage, then the connectivity and quality were obtained in the ranking stage. Finally, the

entropy method was used to combine all three scores and a ranked list is provided. Supervisor profiles were modeled using a novel keyword correlation matrix using their research interest details and publications. Despite that, the evaluation of the solution is not well assuring since it was merely dependent on the feedback of the students, and the flexibility of the framework was not justified.

A multi-level profile-based approach was introduced by the authors to find the most suitable experts in academic research collaboration [21] that combined research relevancy with the network proximity details. This network proximity is a new measure proposed in this work that measures the degree of association between the researcher and the institutes interacted. The expert profile was represented as a combination of three aspects, expertise relevance, individual connectivity, and institutional connectivity. In the relevance computation, the Author-Topic model based on Latent Dirichlet Allocation (LDA) was used to identify the relation between the authors and the latent topics. For ranking the experts a score-based method Comb-MNZ was applied and weights of the scores are computed by greedy optimization. As per the writers, there was a discrepancy due to de-normalized keywords in the Author-Topic modeling and they proposed to use a domain ontology to avoid this problem.

Table 2.2: Comparative Analysis of Aggregating Techniques used in ERS

Technique	Advantages	Disadvantages
Evidential Reasoning (ER Rule-Based)	▪ Aggregate evaluation information on multiple criteria using the ER rule ▪ Consistent and informative support to make informed decisions ▪ Different pieces of evidence are associated with weights and reliabilities in the aggregation process ▪ Utilizes historical data to measure the reliability of an expert ▪ Consider the constraints in the selection process	▪ Selection is done using review information from multiple peer experts, hence the decision is more subjective, and objective information is not used ▪ Does not capture the richness of the actual evaluation information of reviewers ▪ Using an adding up the score is not effective to differentiate the true quality ▪ Decision maker's preferences are not modeled

Entropy-based	▪ Matrices are used to aggregate information in multiple dimensions ▪ Preferences and subjective information were considered ▪ Indirect relevance is used with previous students	▪ Only used in the process of weight computation ▪ The influence of subjective information is low
Analytic Hierarchy Process (AHP)	▪ Determine the relative importance of different criteria using pair-wise comparison matrices ▪ Provides a simple and very flexible model for a given problem ▪ Can use either subjective or objective information	▪ The computational requirement is tremendous even for a small problem ▪ The methodology cannot guarantee the solutions as definitely true ▪ When the number of the levels in the hierarchy increase, the number of pair comparisons also increase ▪ Qualitative components are not convincing
Random Forest	▪ Determined by the probability of the expert being captured in the positive class of confusion matrix ▪ Time efficiency ▪ Suitable for large scale datasets	▪ High computational resource requirement

Sources: [17], [18], [20], [22]

Chen and others have proposed an answerer recommendation system (ARS) for a question answering platform that operates promptly and improves peer interactions by instant answering [22]. They have analyzed past answers and categorized domain experts by keyword tags and determined the online availability by the last login time. Then estimated the likelihood of answering as a binary classification problem using the random forest algorithm and route the question to the expert considering their preference in answering as well. The ranking of the experts was determined by the probabilities of them being captured in the positive class of the confusion matrix.

Liu and others have proposed a simple aggregation method to predict the most suitable experts for peer review by considering various constraints in the context of big data [23]. They have employed three modules for relevance, connectivity, and quality in the second stage of the expert recommendation algorithm. In this aggregation model, the importance of the connectivity score is reduced by considering the remainder from

subtracting one. Then all three scores were multiplied to make the impact of relevance and quality measures high.

Hoang and others have implemented a framework for recommending a group of experts to review a specific problem [24]. They have employed three main modules, one is to collect data from open-source databases like DBLP, ResearchGate, and Google Scholar, an expert detection module to determine and classify expertise, and a prediction module to match the problem with clusters of experts. They have used machine learning to extract expert features and classified them into similar groups. Cosine similarity was used to predict the matching experts for a new query considering three criteria diversity, competence, and suitableness. Finally used a simple weighted sum method to aggregate the three scores. The limitation of this work is not considering the reputation of the expert in the recommendation.

In this paper, the authors have presented a detailed approach on how to apply the expert recommendation for trouble ticket routing in a time-efficient manner with minimal human efforts, which is the same problem of interest for our research as well [25]. They have focused on a collaborative network for the expert assignment process using historical ticket data through maximum likelihood estimation. Also, both experts and tickets were represented using vectors and modeled in the same dimension via transformation matrices. The specialty in this approach is that they are providing two recommendations consecutively using two algorithms. If the initial recommendation fails it considers the closest experts in the network for the next recommendation. The recommendation process continues until the ticket is resolved by an expert. They have utilized an objective function optimization for detecting the similarity between experts and historical tickets and focused on applying deep learning for model improvements. Anyhow they have not considered other factors like workload and the expert's ability to attend to the tickets on time.

Another hybrid approach was introduced in this paper for expert retrieval in community question answering services which used two segments for computing knowledge and authority scores [26]. The novelty in this contribution is determining

the question hardness and using the Reciprocal Rank Fusion (RRF) technique in the aggregation stage. They have evaluated the solution under five metrics over eighteen state-of-the-art algorithms on four real-world datasets. However, there was a limitation in the efficiency of the algorithm that required processing of large-scale text in the archival, and obtaining the similarity with a new query at each instance was time-consuming.

2.5 Challenges in ERS

As with any traditional recommender system, expert recommender systems also have similar challenges like modeling user preference, cold start, and sparsity problems. Still, expert mining is mostly limited to the document and network information of the experts. Studies do have not adequate evidence in utilizing multimedia content, images, audio, video, etc for expertise extraction. With the rapid evolution of modern technology, expertise is also changing fast. To address these dynamics in the real world we need more time-efficient approaches in expert recommender systems. Also, there are privacy-related concerns in sharing and manipulating personal expert details without their consent. So the solutions should be developed adhering to these data privacy guidelines.

As well as there are many limitations with the existing techniques like high complexity, time inefficiency, huge computational cost, and higher error rates. Achieving high accuracy for the expert recommendation with other conditions simultaneously is a hot topic among the research community. When evaluating the implemented ERS solutions the researchers have shown the necessity of benchmarked datasets with reliable information. Usually, most organizations do not publish their expert details publicly which has limited research work in the enterprise domain.

Modeling the expert's preference in the recommendation process has space for improvements. When the system is recommending the most qualified or well-reputed expert for a new query each time, there is a tendency of work getting overloaded for

that person. Most of the research work has not considered experts' viewpoints such as their availability to attend to the problem of the end-user. There is a need for an effective approach that produces alternative expert recommendations within the problem contexts given all these situational conflicts.

2.6 Problem Definition

Identifying the most suitable experts considering multiple factors such as relevance, expertise, experience, and the workload to do a given task with minimal resources has been a research challenge in the expert recommender system area.

Every application domain has a specific set of problems that need to be addressed in the recommendation task, hence it is impossible to apply a generalized recommendation approach to each area. This research focuses on how to solve the expert recommendation problem in a personalized manner with high accuracy and efficiency in the IT service management domain because there is a risk associated with inaccurate recommendations, which leads to customer dissatisfaction. We study how to extract the expertise details from the resources available at the industry level and how to integrate our approach in the organizational settings.

According to the literature the latest trends in expert recommender systems heading towards modeling the preference of the experts by integrating subjective, objective, and social network information. Obtaining similarity measures consider both direct and indirect relevance of the users to past users. Consolidating all the information has resulted in expert profiling in multiple dimensions. Therefore in this study, we have perceived the expert recommendation task as a multiobjective optimization problem and attempted on improving the accuracy and efficiency of the recommendation algorithm.

2.7 Summary

This chapter presented a critical review of literature in expert recommender systems and identified achievements in the field as well as the research gaps in the current trends. We have presented the evolution of the expert recommender system approaches from a multiobjective optimization viewpoint and provided a comparative analysis of the existing aggregation techniques used in the expert recommendation literature. In the next chapter, we give a detailed discussion on the technologies adapted for our problem domain.

CHAPTER 3

3. TECHNOLOGIES ADAPTED

3.1 Introduction

In chapter 2 we have critically reviewed the literature in the expert recommendation research area and identified the research gaps. In this chapter 3, we provide an in-depth study of technologies to solve the expert recommendation problem more accurately and efficiently. The purpose of this study is to adapt the technologies from an IT service management perspective. The technologies identified are under four subfields of artificial intelligence, namely, Natural Language Processing, Topic Modeling, Semantic Web & Ontology, and Swarm Optimization.

3.2 NLP Text Preprocessing

In organizations, expert knowledge is mostly maintained as documents or text content. Therefore expertise can be extracted from emails, knowledge bases, wiki pages, codebases, user manuals, resumes, etc. In IT service management we use helpdesk tools to track incidents and their resolutions. When a user submits a ticket with issue details in the helpdesk tool, then a domain expert is assigned to handle the issue until the closure. Therefore the summary, description, and comments of the issue tickets contain useful information for expert mining.

We can employ natural language processing (NLP) techniques to process these huge collections of unstructured text data and generate hidden insights related to experts. These documents are in human language, mostly in English, hence first we need to apply text pre-processing techniques to convert them into a machine-readable format.

The authors have comprehensively reviewed literature related to NLP text pre-processing in text mining [27]–[29] and the methods include, tokenization, stemming, lemmatizing, stop word removal, case folding, punctuation removal, removing non-alphanumeric characters, Part of speech tagging, Named Entity Recognition, N-grams, etc. We provide a brief description of these basic methods below.

3.2.1. Tokenizing

A text document consists of chapters, paragraphs, sentences, phrases, words, characters, etc. In NLP a word is considered to be the smallest element in semantic analysis. Tokenizing is the method used to segment the text content into semantic elements. It reduces the space required for document storage and improves the efficiency of information retrieval. Whitespace-based and regular expression-based tokenizations are the widely used methods. Most of the NLP toolkits provide tokenization facilities for English and other popular languages. NLTK Regexp tokenizer, TreebankWord tokenizer, and WordPunct tokenizer are a few examples.

Machine learning or language models require a bag of words (BOW) input for text mining. Hence we can use tokenization to produce a dictionary of words from our problem domain and improve the expert mining process.

3.2.2. Stemming

Stemmer converts words in any grammatical form into their roots by removing prefixes and suffixes. Apart from affix removal (truncating), there are other stemming algorithms used such as statistical and mixed. In English, we can reduce words in count (dog, dogs), tense (run, ran, running), gender (waiter, waitress), etc using stemmers. . Therefore, in the helpdesk domain, the words check, checked, checking, checks all can be stemmed from the word “check”. It enhances the efficiency of the expert mining process by decreasing the memory requirement and reducing the distinct word count in the dictionary.

Porter’s stemmer and Lancaster stemmer are the widely used stemming technique in NLP applications. In real-world language processing, two scenarios occur with stemmers namely, over stemming and under stemming which should be handled according to the language structure.

3.2.3. Lemmatizing

Lemmatization reduces the word to its “lemma” which is similar to the stemming method, but it is a step ahead of that. In other words, a lemmatizer provides the dictionary form of a word and it considers the position of the word in a sentence for this purpose. When stemming converts the word “shared” as “shar” by removing the suffix “ed”, lemmatizing converts it to the dictionary term “share”. WordNet lemmatizer is a popular technique used in lemmatization and it is used for searching and indexing documents. However computationally, lemmatization is more time-consuming and susceptible than stemming, hence for practical applications stemming is preferred by the community.

3.2.4. Stop word removal

This is used to remove the common words in the language such as articles and prepositions, and conjunctions in English. For instance words in English like “a”, “the”, “as”, “of” carry no semantic meaning in the expert mining. Removing such words with high occurrences enhances the frequency of more important words. Apart from the default stop words lists provided by the NLP toolkits, we can define custom stop words within the problem context. For example words like “hi”, “please”,” check”, “reminder” are more frequent in ticket data but do not have any hint on the expert’s knowledge. Hence using stop word removal we can further improve the expert mining and generate meaningful information.

3.2.5. POS tagging

Part-of-speech (POS) tagging provides the part of the speech tag of the tokens, in other words, classify the words as nouns, verbs, adjectives, etc. The tags are also known as labels, for example, N for nouns, V for verbs, ADJ for descriptive words, ADV for adverbs, PREP for prepositions, and CONJ for conjunctions. NLP toolkits provide POS taggers such as Penn Treebank tagset based on maximum entropy. When processing issue ticket data, we can use POS tagger labels to extract the semantics in terms of nouns and verbs, for instance, verbs are related to the task assigned to the

expert. However, there are limitations with semantic ambiguity and ungrammatical errors when using POS taggers.

3.2.6. Named entity recognition

Named Entity Recognition (NER) is applied after the POS tagging. It removes unstructured content and extracts meaningful phrases related to people, organizations, places, time keys, etc. In line with that NER, can generate labels such as PERSON for people, ORG for organizations, DATE for dates, etc. Therefore can use Named Entity Recognition to improve the accuracy of the expert mining and classify the phrases based on the labels to identify experts and their tasks.

3.2.7. N-grams

N-grams extract the words that occur consecutively in the text and determine the relevant content. Bigrams and Trigrams are the widely used N-gram combinations in practical applications. The concept behind Bigrams is that if two words occur together more frequently in a document, the content is most probably related to the topic represented by that phrase. This technique is more effective for documents with short text since the significance of the Bigram is high compared to long text. Therefore we can use Bigrams to discover topics in the issue data which consists of short text.

3.3 Topic Modeling for Documents

In the expert recommendation literature, language models play a major role in mining expertise from documents. Topic modeling is the widely used technique to determine the degree of association between people to documents and vice versa. For instance, ExpertSeer is an open-source project for digital libraries based on language models to recommend experts in a specific scientific field. They have used keyword mining and Baye's rule to determine author-to-document and document-to-document associations in the expert recommendation process [30].

The state-of-the-art techniques in topic modeling include Latent Semantic Indexing (LSI), Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF), Random Projection (RP), and Principal Component Analysis (PCA) [31], [32]. In the next section, we discuss a few popular topic modeling techniques.

3.3.1. Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is one of the basic topic modeling techniques in the literature and the documents are modeled as vectors. It is an enhancement of the Latent Semantic Indexing (LSI) method and is based on the similarity between words and their meaning. It determines the latent topics by computing matrix decomposition on the term to document matrix. This can be used for keyword matching and it solves data sparsity issues. However, it does not measure the correlation between the topics.

3.3.2. Probabilistic Latent Semantic Analysis (PLSA)

Probabilistic Latent Semantic Analysis (PLSA) is an extension of the LSA method. It uses a generative probabilistic model to enhance the LSA algorithm. The technique used in PLSA is determining the similarity of text data without referring to any dictionary. It is capable of handling semantic ambiguities and was used in real-world applications such as recommender systems. When the number of documents increases PLSA suffers from model overfitting and is unable to generate the distribution at a document level.

3.3.3. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is the most popular topic modeling method used in practical applications. In LDA each document is represented as a distribution of topics, and each topic as a distribution of words. Therefore it employs a generative probabilistic model based on the Bayesian method. It is an unsupervised model, hence learning can be done without providing any training text data, which is the main advantage of LDA. Providing more meaningful data and the ability to handle

documents with different lengths are among the other advantages. However, it needs to combine short text data to get a better insight. Figure 3.1 shows the structure and parameters of the LDA model.

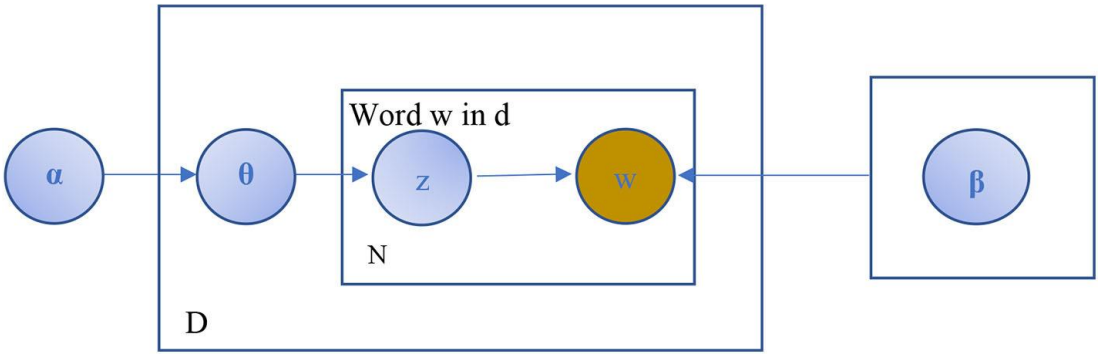


Figure 3.1:Structure of the LDA Topic Modeling

Source: [32]

Parameters of the LDA model are,

α - Dirichlet prior for the document topic distribution
β - Dirichlet prior for the word distribution
θ - Vector for topic distribution over document *d*
z - Topic for a chosen word in a document
w - specific word in N
D - length of the document
N - number of words in the document

Jelodar and others [33] have discussed the practical applications of LDA-based topic modeling in recommender systems in different areas such as scientific papers, music and video, locations, friends, travel and tour, apps, etc. Also, they have studied seven challenges [34] of LDA-based methods in these application areas and suggested the future direction. This paper presents the detailed process of applying NLP preprocessing techniques and LDA topic modeling to develop a recommender system for scholarly articles [35].

The authors have compared the performance of topic modeling methods in short textual content such as comments, reviews, emails, and short messages on the web and social media [32], [36]. According to Albalawi and others, Latent Dirichlet Allocation (LDA) and non-negative matrix factorization had performed better in terms of precision, recall, topic coherence, and F-score in short text analysis. Therefore we can apply LDA-based topic modeling techniques in our issue data analysis, due to their short text nature.

3.3.4. LDA-based Author-Topic Modeling (ATM)

LDA-based author topic modeling was first introduced by Rosen and others, where each author is represented as a topic distribution and, each topic is represented as a word distribution [37]. This is used to model the interests of authors, in other words, we can determine the topics that the author has contributed. Also, we can discover authors who have done similar work relevant to a given document. The advantage of ATM over other topic modeling methods is that we can get recommendations at the author level or document level. Therefore ATM can be used to recommend experts related to an issue ticket by analyzing its summary, description, and comment text data.

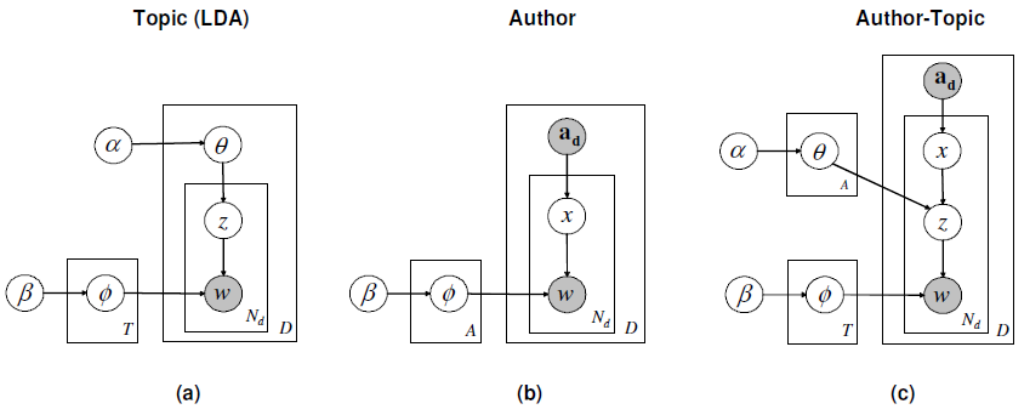


Figure 3.2: Comparison of topic models (a) LDA a topic model. (b) An author model. (c) The author-topic model.
Source: [37]

3.4 Ontology-based Expert Modeling

The growth of the web and social media content has emerged the latest trend of applying semantic and ontology techniques in recommender systems. Semantic web technology is an extension to the document-based world wide web. It was first proposed by Berners Lee in 2000 and enables human-machine collaboration by ontological concept mapping. In other words, semantic recommender systems employ knowledge bases defined in the form of ontology to improve the performance of recommendations [38]. In an ontology, we have the features such as class hierarchies, object properties, and data type properties to define the concepts of a domain and their relationships. Also, many reasoning techniques namely Hermit, Pallet, etc. to determine the similarity.

As per the ERS literature, ontologies are used to model the knowledge about the expert profiles and their preference under the interested domain [5], [13]. We can use the reasoning capability of ontologies to group similar experts under their properties and relations [14], [39], [40]. The performance of recommendation systems can be enhanced in terms of cold start and sparsity problems when combined with a domain ontology [41]. Also, ontologies improve the precision and time efficiency of the recommendation [42], [43]. Further, the authors [44] have discussed the benefits of using a rule-based personalization approach to improve the reasoning of the ontology beyond DL and OWL.

Therefore, we can use an ontology to model the personal data of experts such as demographics, skills, domain knowledge, and preferences within the ITSM context. Also, we can define rules to classify similar experts according to their ratings using the inference methods on the ontology. There are many reasoning techniques like Hermit, Pellet, ELK, etc. provided with the ontology designing tools.

3.5 Swarm Intelligence in Optimization

In our problem scope, experts are modeled in a multidimensional space having discrete values in each criterion. Also, it is difficult to represent complex organizational processes in the continuous domain. Hence the traditional optimization techniques cannot be used in the recommendation step to rank the overall score of experts. Swarm Intelligence techniques are capable of optimizing multi-objective problems. These are nature-inspired techniques and resolve the problem in a decentralized manner. In real-world recommender system applications, usage of the swarm intelligence technique was insignificant. Peska [45] suggested that combining SI methods with other approaches as pre or post-processing can achieve promising solutions.

Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are the widely used SI optimization methods in the literature. In addition to that Artificial Bee Colony (ABC), Invasive Weed Optimization (IWO), Fish School Search (FSS), Bat algorithm (BAT), and Gravitational search algorithm (GSA) are the new techniques emerging in this area. The authors [46], [47] have done comparative analysis on the performance of different SI algorithms under benchmark functions such as Sphere, Rosenbrock, Schwefel, etc.

In the below section we are comparing several popular swarm optimization techniques and their applicability to our problem.

3.5.1. Particle swarm optimization (PSO)

The Particle swarm optimization technique was introduced by Kennedy and Eberhart in 1995 and it imitates the bird flocking behavior in the search space to find the optimal solution. In PSO, a particle is the smallest individual and it has a velocity and position to represent the candidate solutions. The basic steps of the original PSO include particle initialization, velocity updating, particle position updating, memory updating, and termination checking. The PSO algorithm has many merits such as easy

implementation and having only a few parameters. As per the latest trend, researchers are proposing enhanced versions of PSO with faster convergence and efficiency.

Silva has applied the PSO algorithm combined with crowd distance for the assignment problem in ERS literature [16]. It has reduced the slow convergence of the algorithm and improved the exploration in the search space.

3.5.2. Ant colony optimization (ACO)

Ant colony optimization (ACO) is proposed by Marco Dorigo in 1992 which is based on the food-seeking behavior of ants. The major components of ACO are ants, pheromone, daemon action, and decentralized action. The ants use pheromones to mark the paths to the optimal solution. The high-intensity paths are chosen by the ants as the direction. The daemon maintains the global paths and their intensity while the decentralized action controls the dynamic environment. Hence this is more suitable for routing or path planning problems than for assignment problems. The ACO also has the slow convergence problem when the search space is large.

Ahmad and Srivastava [48] have proposed ACO based approach for expert identification and routing the queries to them in social networks. They have improved the algorithm in terms of time efficiency and addressed the cold start for problem for new emerging topics.

3.5.3. Artificial bee colony optimization (ABC)

Artificial bee colony algorithm (ABC) was proposed by Dervis Karaboga in 2005 and it is one of the latest swarm intelligence algorithms which has proved significant performance in many areas [49]. It is based on the behavior of honey bees finding nectar and informing about the food source to other bees in the hive. It consists of employed bees, onlooker bees, and scout bees, with their respective roles. The food sources represent the candidate solutions in the search space and the nectar amount shows their fitness value. The phases of the ABC algorithm briefly are [47],

a) Initialization Phase: Population is initialized
b) Employed bees Phase: Searching for new food sources
c) Onlooker Bees Phase: Selecting food sources for local search
d) Scout Bees Phase: Unemployed bees who got their food source abandoned
e) Memorizing the best fitness value and the position
f) Termination Checking Phase: Checking if the termination condition is met

ABC algorithm is simple and easy to implement such as PSO because it requires only two control parameters namely, colony size and maximum cycle number. Also, the authors have compared the performance of ABC with PSO and Differential Evaluation (DE) algorithms in solving constrained optimization problems [50]. According to the results enhanced versions of ABC proved to be more efficient than the state-of-the-art algorithms.

When comparing appropriate swarm intelligence techniques for the problem domain Artificial Bee Colony Algorithm proved to be more efficient, simple and consumes less resources than Ant Colony Optimization [51]. The authors have done a comparative analysis of both algorithms in the movie recommendation task and ABC outer performed ACO in CPU time and behavior. Since our problem has an assignment nature rather than routing, ABC is selected over ACO under swarm optimization algorithms.

3.6 Summary

In Chapter 3 we have presented an in-depth study of technologies adaptable for the expert recommendation problem. According to the reviewed literature, it is more convenient to adopt a hybrid approach to eliminate the drawbacks of different recommendation techniques and to produce better results. The LDA-based Author Topic Modeling can be used for expert mining from the issue text dataset. Also, the discussed NLP preprocessing techniques will improve the expert mining process. The ontology approach is the best way to model already available knowledge of the expert profiles and classify similar experts. Then the Artificial Bee Colony algorithm, a

variant of swarm intelligence techniques will be more effective in multi-objective optimization, to rank the experts under multiple criteria in this assignment problem. In the next chapter, we introduce our hybrid approach for personalized expert recommendation in the IT service management domain.

CHAPTER 4

4. APPROACH

4.1 Introduction

Chapter 3 provided an overview of AI technologies that can be adapted to solve the personalized expert recommendation task accurately and efficiently. In this Chapter 4, we are presenting our novel framework for expert recommendation in the IT service management context. This study is adopting a hybrid approach combining semantic-based ontology and TOPSIS-ABC optimization for the personalized expert recommendation task. In the coming sections, we present the hypothesis, inputs, outputs, process, users, and features of the novel hybrid expert recommendation approach.

4.2 Hypothesis

This research hypothesizes that ontology-driven expert profiling combined with TOPSIS-ABC optimized expert ranking can provide better solutions with high accuracy and efficiency in the personalized expert recommendation task considering multiple factors.

4.3 Input

There are two main inputs of this approach namely, expert data and issue data. Expert data means the personal details of the experts including their identification details, contact information, skills, domain knowledge, pending workload, etc. The issue data consists of issue ticket id, summary, description, comments, reporter, system category, issue type, etc. Also for the training of the author topic model, we provide historical issue data with the labels of the assigned expert.

4.4 Output

The output of this approach will be a ranked list of the most suitable experts who are capable of resolving the given issue with minimal resources. Also, it will display the identification and contact details of the experts along with their overall scores.

4.5 Process

The proposed framework mainly consists of three phases, expert mining, expert matching, and TOPSIS-ABC optimization. For expert profiling, we consider both subjective and objective information of the experts and compute the expertise index. Subjective information includes their self-declared qualifications and interests, while objective information such as teams, domains, etc. are according to the organizational structure and procedures. Then the domain ontology is used to build the expert profiles including their preferences. After that, the classification of similar experts is done by the reasoning methods of the ontology.

For relevancy computation, we use both direct and indirect matching. When a new issue ticket is raised the LDA-based ATM model will be used to predict similar experts by indirect matching with historical data. Then the ontology-based semantic similarity will be used to determine the direct matching of the experts based on the system category and issue types. The combination of these two methods will address the cold start and sparsity problems by providing alternative solutions.

To indicate the availability of the expert we use the productivity index, which measures the pending workload of the expert. As stated above, in this framework experts are modeled in three-dimensional space using expertise, relevancy, and productivity indexes. Then the Artificial Bee Colony algorithm determines the ideal solution from the candidate expert scores as a conflicted constrained optimization problem. Finally, the TOPSIS method is used to obtain the ranked list of the most suitable experts by computing the distance to the ideal solution.

4.6 Users

The intended users of this system are support agents and the end-users (customers) of the respective applications. Also, this system is capable of reducing manual intervention in finding relevant experts. Hence it will capture the interest of line managers and supervisors in terms of cost reduction. Moreover, human resource managers can benefit from this system by managing the expert knowledge within the organization and identifying the areas that need new recruitment.

4.7 Features

The simulation design of this approach contains a simple web interface that facilitates inputting an issue ticket id and getting the top-n expert list with their details and overall scores.

4.8 Summary

In this chapter, we have given an overview of our hybrid expert recommendation approach and presented the hypothesis. We are modeling the experts in three dimensions using the indexes relevance, expertise, and productivity. The novelty of this approach is using a combined technique for aggregating, namely, TOPSIS based ABC algorithm. This approach produces the most matching alternatives that are closely related to the ideal solution. In the next chapter, we provide a detailed description of the proposed design.

CHAPTER 5

5. DESIGN

5.1. Introduction

Chapter 4 presented our novel hybrid approach ExRecSys for expert recommendation in the IT service management domain. This chapter discusses the design of the proposed framework. We present the top-level architecture, modular architecture, and detailed descriptions of the modules and their inter-connectivities.

5.2. Novel Framework ExRecSys

The ExRecSys framework has three main phases for expert mining, expert matching, and TOPSIS-ABC optimization. In expert mining, we have considered both subjective and objective details of the experts and included their preferences as well. Experts are allowed to self declare their expertise and state their domains of interest. In this framework as shown in figure 5.2, we model the experts using three scores namely, expertise, relevance, and productivity. Obtaining these scores is a sequential process and at the final step, it recommends the top-N list of the best experts related to a given issue ticket. In the following sections, we present the architecture diagrams of the design.

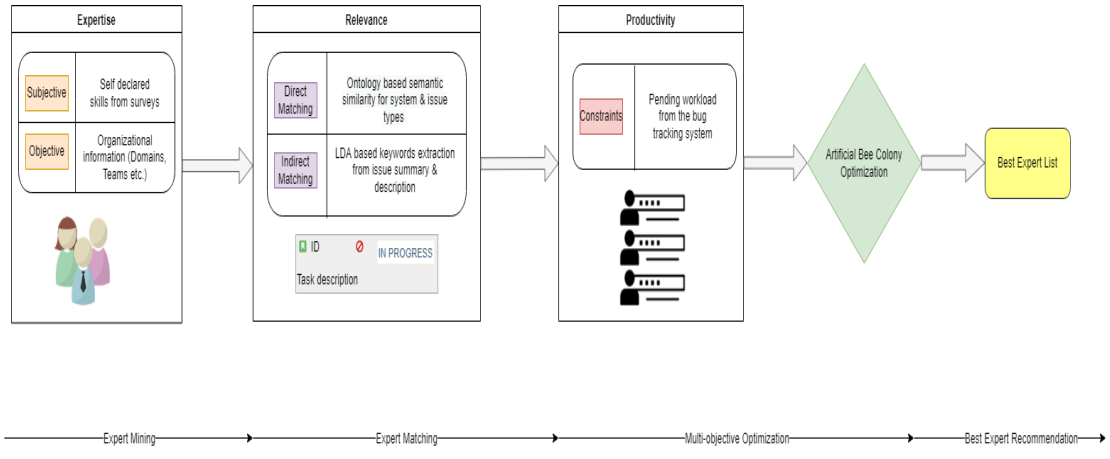


Figure 5.1: Proposed Framework for Personalized Expert Recommendation

5.3. Top-level Architecture

Figure 5.2 illustrates the top-level architecture of the personalized expert recommender engine. It consists of three main modules, Preprocessor & ATM Predictor, Ontology Reasoner, and TOPSIS-ABC Optimizer. It has three inputs expert data, issue data, and new query. The issue data is preprocessed by the Preprocessor module and expert data is entered into the ontology. The output is the ranked list of best experts with their overall score. In the next section, we discuss the modular architecture and the detailed processes of these modules.

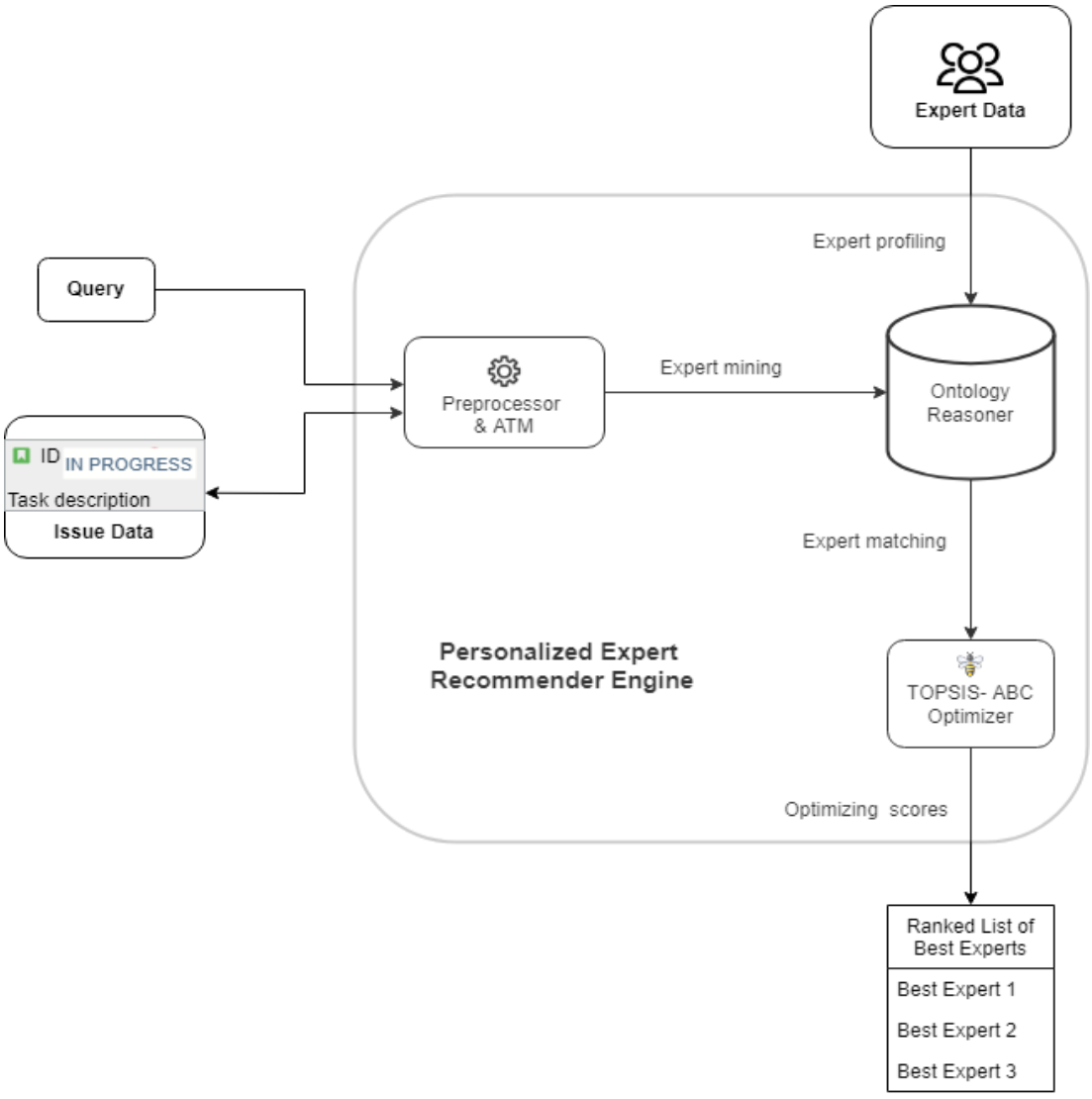


Figure 5.2: Top Level Architecture of the Proposed System

5.4. Modular Architecture

Figure 5.3 shows the modular architecture of the design and the inter-connections between the modules. It also shows the interactions of the end-users with the system modules. Among the three main modules, the Text Preprocessing and ATM module and TOPSIS-ABC optimizer module belong to the application layer while the Ontology Reasoner module is in the business logic layer. In the presentation layer, the end-user is allowed to submit issue tickets using the service desk web interface, and experts can update the ontology with their skill ratings via the ontology interface.

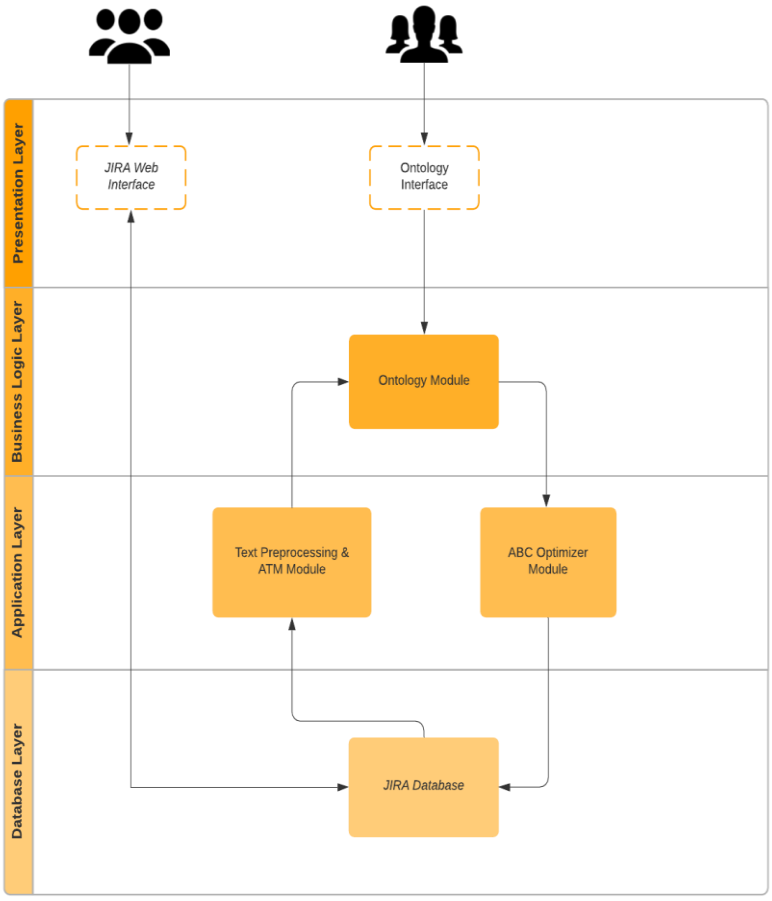


Figure 5.3: Modular Architecture of the Proposed System

Then the information updated in the service desk database is taken for text pre-processing and expert prediction via the Preprocessor & ATM module. After that, the top-n candidate expert list goes through the Ontology Reasoner module and computes their expert ratings based on semantic similarity and business rules. At the final stage, the TOPSIS-ABC optimizer recommends the most suitable top-n experts to assign the issue ticket.

5.4.1. Text Preprocessor & ATM

In this module, the summary and description of the issue ticket are combined and short text is generated. Then the text is preprocessed using the NLP techniques discussed in the technology-adapted chapter. After that, the short text content is fed into the ATM module to predict similar experts for the given issue ticket. The ATM module is trained on the historical issue data and experience of the experts for this expert prediction.

5.4.2. Ontology Reasoner

The ontology contains the expert profile details under demographics, skills, domain knowledge, and the relationships between the experts and the applications. Also, the Ontology Reasoner classifies similar expert groups based on the business logic definitions. When the system category of an issue ticket is provided it will recommend the relevant experts using the inference mechanisms.

5.4.3. TOPSIS-ABC Optimizer

The TOPSIS-ABC optimizer module takes the candidate experts with expertise, relevance, and productivity scores and computes the score combination of the ideal expert using the ABC algorithm. Then for each candidate expert, it determines the distance from the ideal solution based on the TOPSIS method. Finally, it provides the ranked list of top-n experts with minimal distance to the ideal expert.

5.5. Simulation Design

The simulation design includes a web interface that the end-user can input an issue ticket id and get the best-recommended expert list with the scores. When the ticket id is submitted the issue details and the workload information of the experts are taken from the helpdesk database and provided to the ExRecSys. Then the data flow happens through the modules as explained in the modular architecture section and the best expert list is produced.

5.6. Summary

In chapter 5 we have discussed the architecture of the proposed design, main modules, their internal processes, and interconnectivity with other modules. Also, we have described how the end-user communicates with the system and the flow of data for a new query. In the next chapter, we will describe the implementation of each module and how the expertise, relevance, and productivity scores are obtained for the experts.

CHAPTER 6

6. IMPLEMENTATION

6.1. Introduction

In Chapter 5 we have discussed the proposed system design in terms of the architecture and modules. In the subsequent chapter, we present the implementation details of the modules such as algorithms, flowcharts, pseudo-codes, tools, software, and hardware utilization. Also, we explain how the expertise, relevance, and productivity indexes are computed.

6.2. Expert Profiling

An expert profile can be represented using a vector under multiple areas of expertise or other factors related to the expert. This is known as the topical profile and the expertise is measured against the level of skills or knowledge areas for each expert [2]. According to the equation below, the *k* weighted score obtained by the expert *e* in the area a_1 is represented by the term *score(e,ka1)*. The overall expert profile is the sum of these terms under all areas.

$$profile(e) = < score(e,ka_1), score(e,ka_2), \ldots, score(e,ka_n) > \text{ , where } i=1,2,3,\ldots,n$$

Influenced by this, our approach represents the expert profile using three scores Relevancy (R), Expertise (E), and Productivity (P) for the i^{th} expert for a given query q. The weights of the scores are manually determined by a domain expert which adds up to one. The following equation 6.1 shows the mathematical representation of the expert profile in three-dimensional space.

$$profile(e_i, q) = \alpha R_i + \beta E_i + \gamma P_i, \qquad where\ \alpha + \beta + \gamma = 1$$

Equation 6.1: Expert Profile

6.2.1. Relevancy index using Author-Topic Modeling

The Relevancy Index (R) is obtained from the prediction score of the ATM module for a given query q as shown in equation 6.2 below. In the next section, we describe the steps of the text preprocessing and author-topic model training.

$$R_i = prediction_{ATM}(e_i, q)$$

Equation 6.2: Relevance Index

We have referred to historical issue data to train the Author-Topic model. For this purpose, we have taken 85000 past issue tickets submitted over 6 months from the helpdesk of a large-scale IT support organization. The dataset included the fields, namely issue key, issue id, summary, description, issue type, and the assignee.

First, we removed the issue ticket records which had no assignee details and selected the experts who only contributed to resolving more than ten tickets. We have categorized the remaining 55000 tickets according to the assignee. Then we have split the dataset into training and testing sets. After that, we have selected the ticket summary and description for the expert mining process and created short text documents concatenating the two fields under each expert.

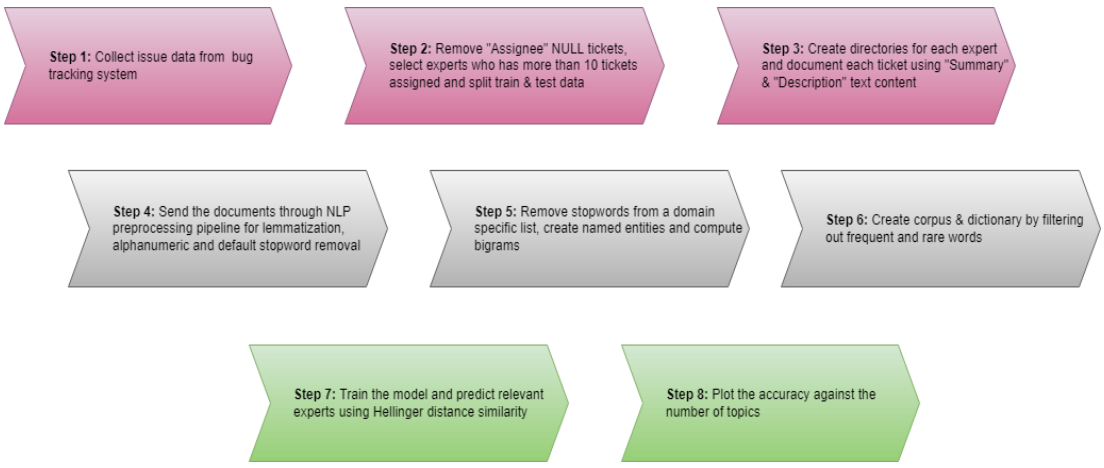


Figure 6.1: Text Preprocessing & Author-Topic Modeling Steps in brief

Figure 6.1 illustrates the text preprocessing and ATM model training steps briefly. For text preprocessing, we have used the Spacy preprocessing pipeline. Spacy is a standard NLP toolkit written in Python and Cython specifically for industrial applications due to its efficiency [28]. It provides tokenization, POS tagging, NER, word vector facilities. We applied Spacy's pre-trained English model optimized for CPU, namely en_core_web_sm. The preprocessing steps include lemmatizing tokens, removing punctuation and numeric characters, removing default and custom stopwords, POS Tagging for Verbs, Nouns, and Proper Nouns, and Named Entity Recognition.

Thereafter, we used the Gensim library for computing Bigrams for words that appear twenty times or more. Gensim is also an open-source Python library supporting unsupervised topic modeling and natural language processing. Then we created a corpus and dictionary from the training dataset and trained the author-topic model. The accuracy of the model was computed against the test dataset and plotted under different number of topics. The model accuracy results are presented in the evaluation chapter more comprehensively [52]. We have used a computer with Windows OS, 2.6 GHz Core, 8GB RAM for training this model, and text preprocessing took around 20 mins for the training dataset and 8 mins for the testing dataset. The model training took around 24 mins for 200 topics with 50 iterations and symmetric alpha and eta parameters.

When a new query is raised by an end-user the issue text data goes through the same preprocessing steps and the trained ATM model is used to predict the relevancy of the experts. In this expert prediction, Hellinger distance is used for similarity computation [53], [54].

6.2.2. Expertise index using Personalized Ontology

The Expertise index (E) is computed using the below equation 6.3 for the similar experts inferred from the ontology.

$$E_i = \sum_{j=1}^{n} w_j s_j$$

$where,$

$s_j = the\ rating\ of\ the\ j^{th}\ skill\ of\ the\ i^{th}\ expert,$

$w_j = the\ weight\ assigned\ for\ the\ j^{th}\ skill,$

$n =\ number\ of\ skills\ ratings\ the\ i^{th}\ expert\ is\ measured$

Equation 6.3: Expertise Index

According to the ITILv.3 framework, the ontology for the IT service management domain was developed [55], [56]. Application, Issue Type, Team & Agent are the concepts of ITSM which we have represented via classes, then the relations are designed as shown in figure 6.2 via object properties and the demographic and skill ratings are indicated by the datatype properties.

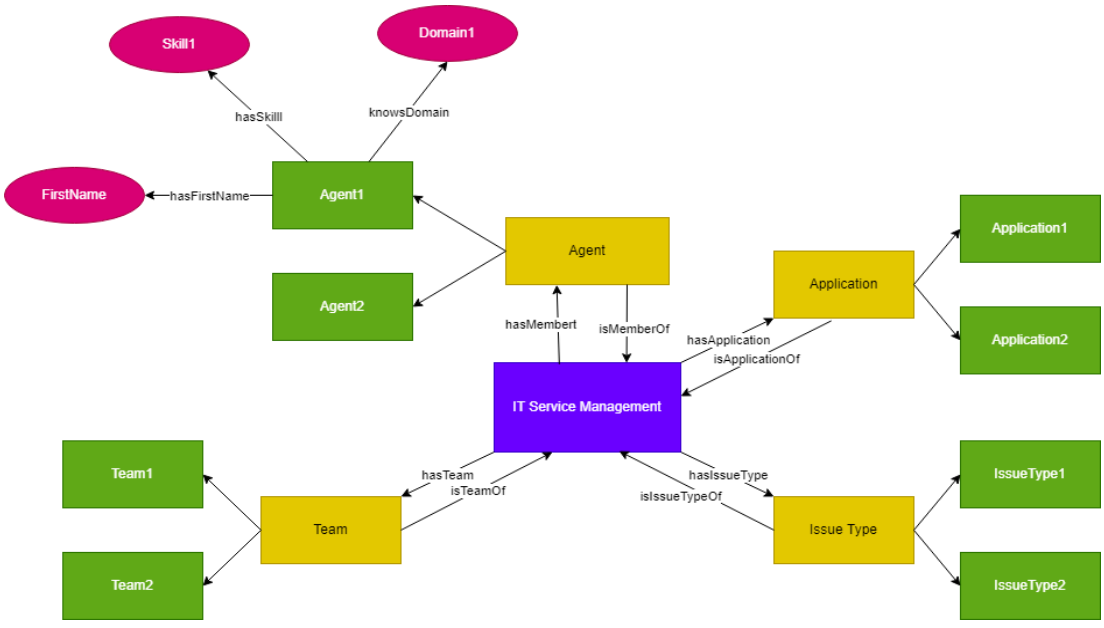


Figure 6.2: ITSM ontology adapted from the ITILv.3 framework

Table 6.1: ITSM Ontology Classes and Property Definitions

Name	Description	Type
ITSM	IT Service Management core concept	Ontology
Application	Types of applications managed by the organization	Class
Issue Type	Types of issues that occur in applications	Class
Agent	Support agent who resolves the issues	Class
Team	Group of similar support agents	Class
hasApplication	The relation between ITSM and Application	Object Property
hasMember	The relation between ITSM and Agent	Object Property
hasTeam	The relation between ITSM and Team	Object Property
hasFirstName	Describes the first name of an Agent	Datatype Property
hasSkill	Describes the skill rating of an Agent	Datatype Property
hasDomain	Describes the domain rating of an Agent	Datatype Property

We have defined equivalent classes for specific teams according to skills and domain ratings, then the reasoner engine classifies the similar experts by inferencing as per the rules. Then we used SPARQL queries to retrieve the candidate experts for a given query based on the application type of the ticket. Then the Expertise index is calculated using the datatype property values of the candidate experts as per equation 6.3.

For the ontology design, we have used the ontology development tool Protégé 5.5.0 and Hermit reasoning engine for the inference. Further, we used python libraries OWLReady2, RDFLib, OS, Numpy for implementing this module. Figures 6.3, 6.4, and 6.5 illustrate the class hierarchy, object property hierarchy, and datatype property hierarchy respectively.

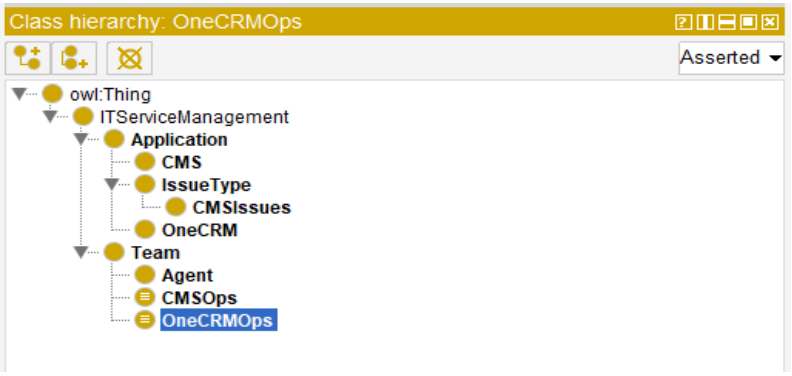


Figure 6.3: Class Hierarchy

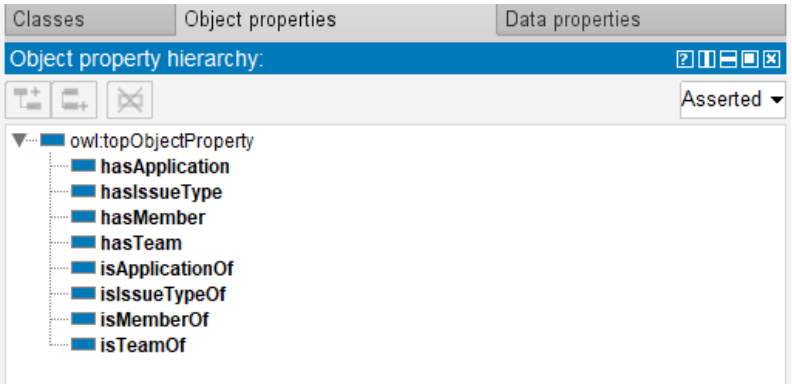


Figure 6.4: Object Property Hierarchy

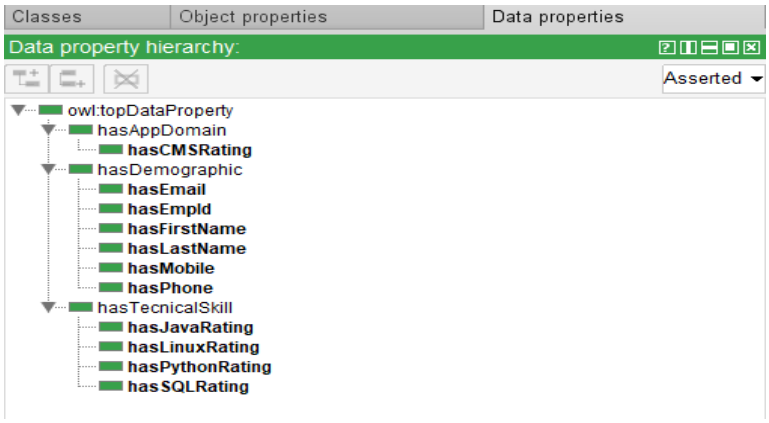


Figure 6.5: Datatype Property Hierarchy

6.2.3. Productivity index using Workload

The Productivity Index (P) is computed from the number of issue tickets $|T|$, assigned to the i^{th} expert at the time the query is raised t_q.

$$P_i = |T|, where\ T = \{Tickets\ assigned\ to\ e_i\ at\ time\ t_q\}$$

Equation 6.4: Productivity Index

For this purpose I have used the search_issues( ) method of the JIRA Atlassian API and JQL queries to get the issue ticket list assigned to the candidate experts and counted the number of ticket ids.

6.3. Expert Ranking

In this approach, we rank the candidate experts using the computed scores for Relevance (R), Expertise (E), and Productivity (P) indexes. This section gives the detailed process of the expert ranking using the TOPSIS-ABC optimizer module.

6.3.1. TOPSIS-ABC Optimization

Table 6.2 shows the functions we used to compute the scores and the candidate experts are represented as 3-D vectors in the solution space. Then all the scores are normalized in the range [0,1] and the scores of the ideal expert are obtained among the candidate experts. After that based on the TOPSIS-ABC method [57], [58], candidate experts are ranked by minimizing the distance from the ideal expert, where the distance is obtained by the sum of squared errors.

Objective Function 1 → $(e_i, q) = \alpha E_i + \beta R_i + \gamma P_i$

Weighted sum → $\alpha + \beta + \gamma = 1$

Objective Function 2 → Sum of the squared errors to the ideal solution

Table 6.2: Candidate Experts are represented in three-dimensions

	Relevance (R)	Expertise (E)	Productivity (P)
Function	$prediction_{ATM}(e_i, q)$	$\sum_{j=1}^{n} w_j s_j$	$\lvert T \rvert$, $where\ T = \{Tickets\ assigned\ to\ e_i\ at\ time\ t_q\}$
Candidate Expert 1: Employee A	0.770243	47	4
Candidate Expert 2: Employee B	0.681070	32	10

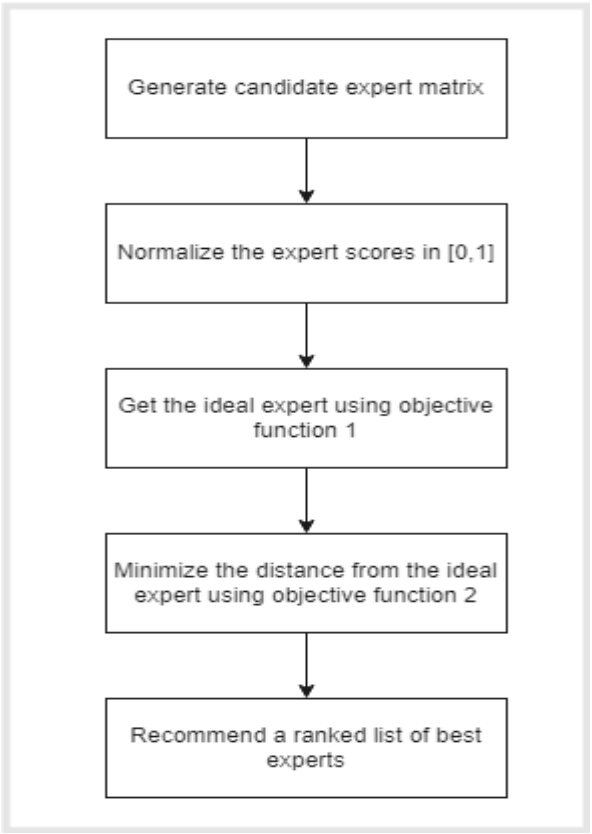


Figure 6.6: Flow chart of the TOPSIS-ABC Optimizer

For the implementation of this module we have used Tensorflow, Sklearn Numpy, Pandas, Matplotlib and Swarm Intelligence Python libraries.

6.4. Summary

In this Chapter 6, we have discussed the implementation of the three modules in detail with algorithms, flowcharts, pseudo-codes, tools, software, and hardware. In the next chapter, we evaluate the ATM module, TOPSIS-ABC algorithm, and the final recommendation in terms of accuracy and efficiency. The evaluation metrics we used for this purpose are precision and computation time.

CHAPTER 7

7. EVALUATION

7.1. Introduction

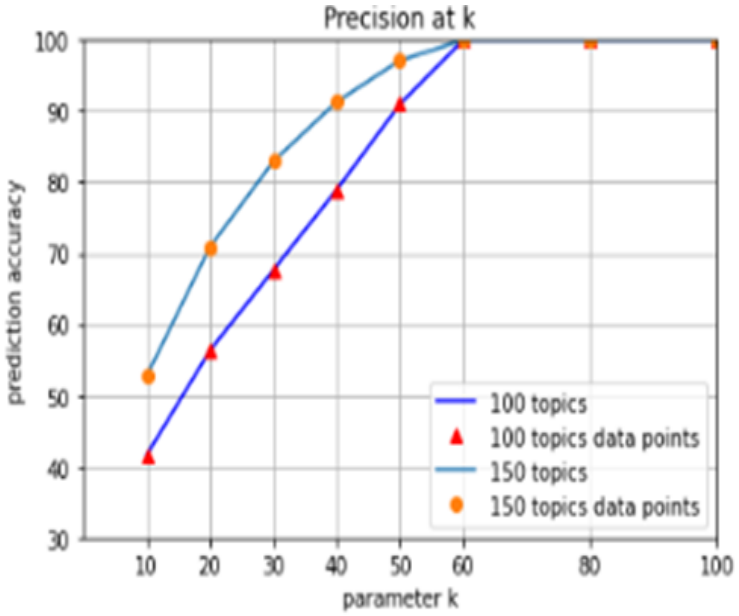
In Chapter 6 we have discussed the implementation of the proposed system in detail using the algorithms, flowcharts, pseudo-codes, tools, software, and hardware. In this chapter, we are discussing the evaluation methods adapted to evaluate this approach in terms of accuracy and efficiency.

The proposed system was evaluated using a real-world dataset related to a large-scale IT Service Management organization in the industry. The performance of the algorithms were measured using the below metrics.

- Precision
- Computation Time

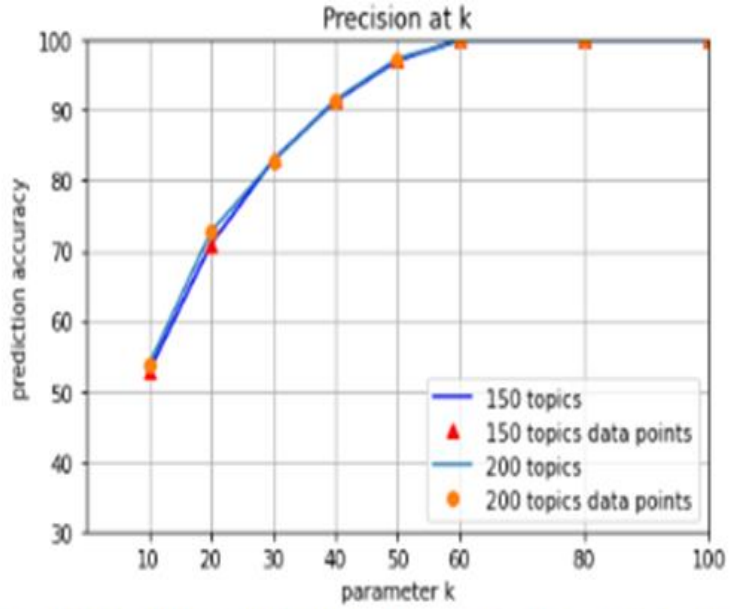
7.2. Evaluation of Text Preprocessing & Author-Topic Model

The Author-Topic model was evaluated using 55000 issue tickets resolved by 58 authors taken from the IT Service Desk of a large-scale ITSM organization. We used symmetric alpha and eta parameters with 50 iterations and plotted the precision against the number of topics. The model converged between 150-200 topics. Following figures 7.1 and 7.2 show the variation of accuracy against the number of topics.




```
CPU times: user 15min 29s, sys: 15.7 s, total: 15min 44s
Wall time: 15min 29s
```

Figure 7.1: AT Model Accuracy Plot for 100-150 Topics




```
CPU times: user 22min 10s, sys: 6min 10s, total: 28min 21s
Wall time: 21min 51s
```

Figure 7.2: AT Model Accuracy Plot for 150-200 Topics

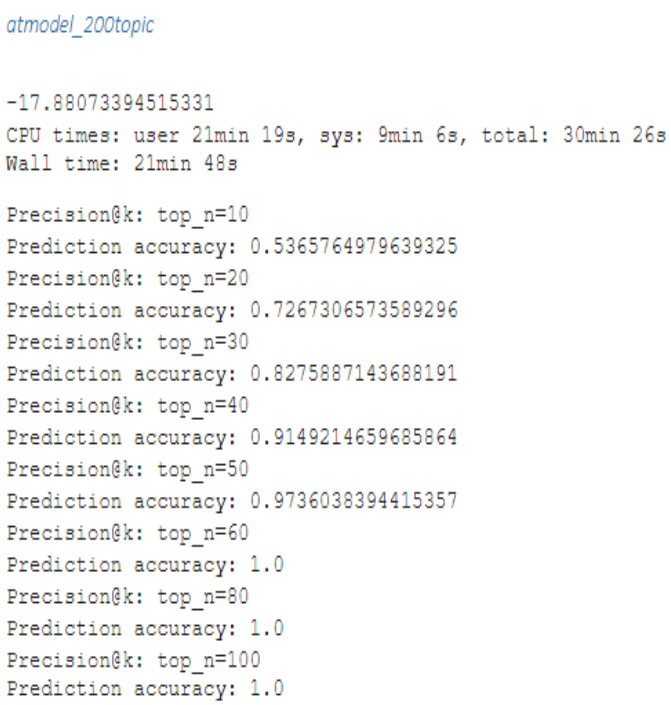


Figure 7.3: Results of the 200 Topic ATM Evaluation

7.3. Evaluation of the Expert Recommendation

In the previous section, we evaluated the precision of the proposed expert recommender system with a real-world scenario. This section presents the results of expert ranking using the proposed TOPSIS-ABC approach.

Figure 7.4 shows the position of the ideal expert in the two-dimensional space which was generated for illustration purposes. Then according to figure 7.5, the final scores of the experts are computed and sorted according to the distance to the ideal expert.

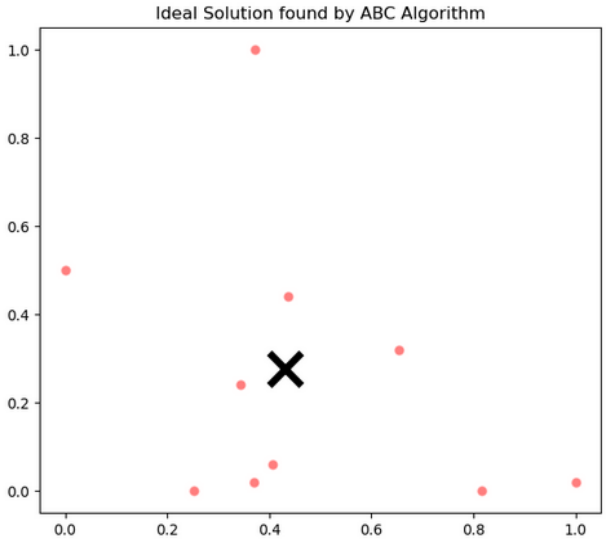


Figure 7.4 Position of the ideal expert in the two-dimensional space

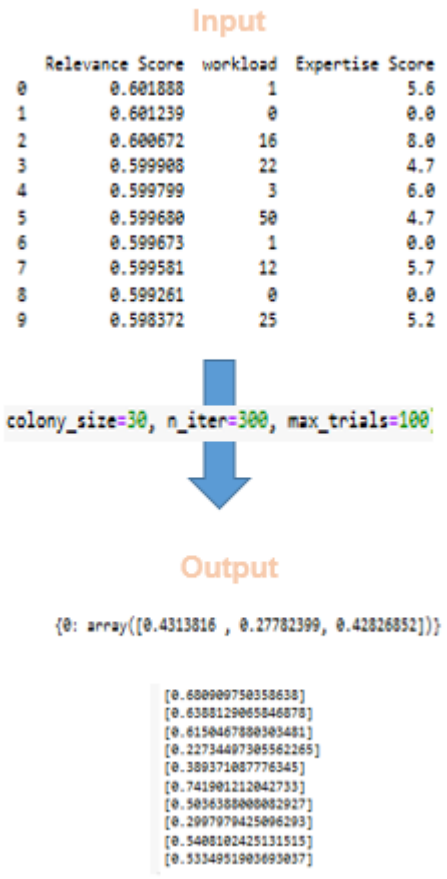


Figure 7.5 Input, Process & Output of the TOPSIS-ABC Expert Ranking

7.4. Summary

In Chapter 7, we have presented the evaluation of the Author-topic model trained for the real-world dataset of an organization. In the next chapter, we discuss the conclusion and further work for this research.

CHAPTER 8

8. CONCLUSION AND FURTHER WORK

In this chapter, we discuss conclusions and further work related to this expert recommender systems research area in IT service management.

The conclusions are,

- Custom stop word removal, POS tagging (Verbs, Nouns, and Proper Nouns), and Bigrams in text preprocessing have improved the accuracy of recommending the best experts
- Author-Topic modeling combined with ontology address the cold start and sparcity problems related to finding experts for new issues
- TOPSIS-ABC optimizer has improved the efficiency of the expert ranking considering multiple factors

Further work includes,

- Use code mapping to recommend developers or L3 support agents
- Integrate a feedback module to improve reliability
- Address dynamics of people resigning/ changing teams/ upgrading skills and ticket routing among multiple agents

9. REFERENCES

[1] N. Nikzad–Khasmakhi, M. A. Balafar, and M. Reza Feizi–Derakhshi, "The state-of-the-art in expert recommendation systems," *Eng. Appl. Artif. Intell.*, vol. 82, pp. 126–147, Jun. 2019, doi: 10.1016/j.engappai.2019.03.020.

[2] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si, "Expertise Retrieval," *Found. Trends Inf. Retr.*, vol. 6, no. 2–3, pp. 127–256, Feb. 2012, doi: 10.1561/1500000024.

[3] S. Lin, W. Hong, D. Wang, and T. Li, "A survey on expert finding techniques," *J. Intell. Inf. Syst.*, vol. 49, no. 2, pp. 255–279, Oct. 2017, doi: 10.1007/s10844-016-0440-5.

[4] O. Husain, N. Salim, R. A. Alias, S. Abdelsalam, and A. Hassan, "Expert Finding Systems: A Systematic Review," *Appl. Sci.*, vol. 9, no. 20, Art. no. 20, Jan. 2019, doi: 10.3390/app9204250.

[5] M. N. Uddin, T. H. Duong, K. Oh, and G.-S. Jo, "An Ontology Based Model for Experts Search and Ranking," in *Intelligent Information and Database Systems*, Berlin, Heidelberg, 2011, pp. 150–160. doi: 10.1007/978-3-642-20042-7_16.

[6] O. Alhabashneh, R. Iqbal, F. Doctor, and A. James, "Fuzzy rule based profiling approach for enterprise information seeking and retrieval," *Inf. Sci.*, vol. 394–395, pp. 18–37, Jul. 2017, doi: 10.1016/j.ins.2016.12.040.

[7] D. W. McDonald and M. S. Ackerman, "Expertise recommender: a flexible recommendation system and architecture," in *Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00*, Philadelphia, Pennsylvania, United States, 2000, pp. 231–240. doi: 10.1145/358916.358994.

[8] D. Yimam and A. Kobsa, "DEMOIR: a hybrid architecture for expertise modeling and recommender systems," in *Proceedings IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2000)*, Jun. 2000, pp. 67–74. doi: 10.1109/ENABL.2000.883706.

[9] *Expert Finding Systems*.

[10] K. Balog, L. Azzopardi, and M. de Rijke, "Formal models for expert finding in enterprise corpora," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, Aug. 2006, pp. 43–50. doi: 10.1145/1148170.1148181.

[11] T. Reichling, M. Veith, and V. Wulf, "Expert Recommender: Designing for a Network Organization," *Comput. Support. Coop. Work CSCW*, vol. 16, no. 4, pp. 431–465, Oct. 2007, doi: 10.1007/s10606-007-9055-2.

[12] T. Reichling and V. Wulf, *Expert recommender systems in practice: Evaluating semi-automatic profile generation*. 2009, p. 68. doi: 10.1145/1518701.1518712.

[13] M. Fazel-Zarandi and M. S. Fox, "Constructing expert profiles over time for skills management and expert finding," in *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, New York, NY, USA, Sep. 2011, pp. 1–6. doi: 10.1145/2024288.2024295.

[14] R. Schäfermeier and A. Paschke, "Using domain ontologies for finding experts in corporate wikis," in *Proceedings of the 7th International Conference on Semantic Systems*, New York, NY, USA, Sep. 2011, pp. 63–70. doi: 10.1145/2063518.2063527.

[15] T. Silva, Z. Guo, J. Ma, H. Jiang, and H. Chen, "A social network-empowered research analytics framework for project selection," *Decis. Support Syst.*, vol. 55, no. 4, pp. 957–968, Nov. 2013, doi: 10.1016/j.dss.2013.01.005.

[16] T. Silva, J. Ma, C. Yang, and H. Liang, "A profile-boosted research analytics framework to recommend journals for manuscripts," *J. Assoc. Inf. Sci. Technol.*, vol. 66, no. 1, pp. 180–200, 2015, doi: 10.1002/asi.23150.

[17] J. Sun, W. Xu, J. Ma, and J. Sun, "Leverage RAF to find domain experts on research social network services: A big data analytics methodology with MapReduce framework," *Int. J. Prod. Econ.*, vol. 165, pp. 185–193, Jul. 2015, doi: 10.1016/j.ijpe.2014.12.038.

[18] W. Zhu, F. Liu, Y. Chen, J. Yang, D. Xu, and D. Wang, "Research project evaluation and selection: an evidential reasoning rule-based method for aggregating peer review information with reliabilities," *Scientometrics*, vol. 105, no. 3, pp. 1469–1490, Dec. 2015, doi: 10.1007/s11192-015-1770-8.

[19] T. Silva and J. Ma, "Expert profiling for collaborative innovation: big data perspective," *Inf. Discov. Deliv.*, vol. 45, no. 4, pp. 169–180, Jan. 2017, doi: 10.1108/IDD-03-2017-0021.

[20] M. Zhang, J. Ma, Z. Liu, J. Sun, and T. Silva, "A Research Analytics Framework-Supported Recommendation Approach for Supervisor Selection," *Br. J. Educ. Technol.*, vol. 47, no. 2, pp. 403–420, Mar. 2016, doi: 10.1111/bjet.12244.

[21] C. Yang, J. Ma, T. Silva, X. Liu, and Z. Hua, "A Multilevel Information Mining Approach for Expert Recommendation in Online Scientific Communities," *Comput. J.*, May 2014, doi: 10.1093/comjnl/bxu033.

[22] T. Chen, J. Cai, H. Wang, and Y. Dong, "Instant expert hunting: building an answerer recommender system for a large scale Q&A website," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, New York, NY, USA, Mar. 2014, pp. 260–265. doi: 10.1145/2554850.2554915.

[23] D. Liu, W. Xu, W. Du, and F. Wang, “How to Choose Appropriate Experts for Peer Review: An Intelligent Recommendation Method in a Big Data Context,” *Data Sci. J.*, vol. 14, no. 0, Art. no. 0, May 2015, doi: 10.5334/dsj-2015-016.

[24] D. T. Hoang, N. T. Nguyen, and D. Hwang, “A Group Recommender System for Selecting Experts to Review a Specific Problem,” in *Computational Collective Intelligence*, Cham, 2018, pp. 270–280. doi: 10.1007/978-3-319-98443-8_25.

[25] J. Xu and R. He, “Expert recommendation for trouble ticket routing,” *Data Knowl. Eng.*, vol. 116, pp. 205–218, Jul. 2018, doi: 10.1016/j.datak.2018.06.004.

[26] D. Kundu, R. K. Pal, and D. P. Mandal, “Topic sensitive hybrid expertise retrieval system in community question answering services,” *Knowl.-Based Syst.*, vol. 211, p. 106535, Jan. 2021, doi: 10.1016/j.knosys.2020.106535.

[27] A. Kadhim, “An Evaluation of Preprocessing Techniques for Text Classification,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 16, pp. 22–32, Jun. 2018.

[28] D. Yogish, T. N. Manjunath, and R. S. Hegadi, “Review on Natural Language Processing Trends and Techniques Using NLTK,” in *Recent Trends in Image Processing and Pattern Recognition*, Singapore, 2019, pp. 589–606. doi: 10.1007/978-981-13-9187-3_53.

[29] M. Işik and H. Dağ, “The impact of text preprocessing on the prediction of review ratings,” *Turk. J. Electr. Eng. Comput. Sci.*, vol. 28, no. 3, pp. 1405–1421, May 2020.

[30] H.-H. Chen, A. G. Ororbia II, and C. L. Giles, “ExpertSeer: a Keyphrase Based Expert Recommender for Digital Libraries,” *ArXiv151102058 Cs*, Nov. 2015, Accessed: Jun. 06, 2021. [Online]. Available: http://arxiv.org/abs/1511.02058

[31] R. Alghamdi and K. Alfalqi, “A Survey of Topic Modeling in Text Mining,” *Int. J. Adv. Comput. Sci. Appl. IJACSA*, vol. 6, no. 1, Art. no. 1, Dec. 2015, doi: 10.14569/IJACSA.2015.060121.

[32] R. Albalawi, T. H. Yeap, and M. Benyoucef, “Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis,” *Front. Artif. Intell.*, vol. 3, 2020, Accessed: Feb. 23, 2022. [Online]. Available: https://www.frontiersin.org/article/10.3389/frai.2020.00042

[33] H. Jelodar, Y. Wang, M. Rabbani, and S. Ayobi, “Natural Language Processing via LDA Topic Model in Recommendation Systems,” *ArXiv190909551 Cs*, Sep. 2019, Accessed: Nov. 09, 2021. [Online]. Available: http://arxiv.org/abs/1909.09551

[34] H. Jelodar *et al.*, “Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey,” *Multimed. Tools Appl.*, vol. 78, no. 11, pp. 15169–15211, Jun. 2019, doi: 10.1007/s11042-018-6894-4.

[35] H. Jelodar *et al.*, "Recommendation system based on semantic scholar mining and topic modeling on conference publications," *Soft Comput.*, vol. 25, no. 5, pp. 3675–3696, Mar. 2021, doi: 10.1007/s00500-020-05397-3.

[36] S. Likhitha, B. S. Harish, and H. M. Keerthi Kumar, "A Detailed Survey on Topic Modeling for Document and Short Text Data," *Int. J. Comput. Appl.*, vol. 178, pp. 975–8887, Aug. 2019, doi: 10.5120/ijca2019919265.

[37] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The Author-Topic Model for Authors and Documents," *ArXiv12074169 Cs Stat*, Jul. 2012, Accessed: Nov. 18, 2021. [Online]. Available: http://arxiv.org/abs/1207.4169

[38] E. Peis, J. Morales-del-Castillo, and J. Delgado-López, "Semantic Recommender Systems. Analysis of the state of the topic," *Hipertextnet Núm 6 Ed. En Anglès*, Jan. 2008.

[39] M. I. Martín-Vicente, A. Gil-Solla, M. Ramos-Cabrer, J. J. Pazos-Arias, Y. Blanco-Fernández, and M. López-Nores, "A semantic approach to improve neighborhood formation in collaborative recommender systems," *Expert Syst. Appl.*, vol. 41, no. 17, pp. 7776–7788, Dec. 2014, doi: 10.1016/j.eswa.2014.06.038.

[40] J. M. Ruiz-Martínez, J. A. Miñarro-Giménez, and R. Martínez-Béjar, "An ontological model for managing professional expertise," *Knowl. Manag. Res. Pract.*, vol. 14, no. 3, pp. 390–400, Aug. 2016, doi: 10.1057/kmrp.2015.3.

[41] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," *Expert Syst. Appl.*, vol. 92, pp. 507–520, Feb. 2018, doi: 10.1016/j.eswa.2017.09.058.

[42] R.-Q. Wang and F.-S. Kong, "Semantic-Enhanced Personalized Recommender System," in *2007 International Conference on Machine Learning and Cybernetics*, Aug. 2007, vol. 7, pp. 4069–4074. doi: 10.1109/ICMLC.2007.4370858.

[43] V. Codina and L. Ceccaroni, "A Recommendation System for the Semantic Web," Jan. 2010, vol. 79, pp. 45–52. doi: 10.1007/978-3-642-14883-5_6.

[44] A. Ameen and Head, "Knowledge based Recommendation System in Semantic Web-A Survey," *Int. J. Comput. Appl.*, vol. 182 – No. 43, pp. 975–8887, Mar. 2019, doi: 10.5120/ijca2019918538.

[45] L. Peška, T. M. Tashu, and T. Horváth, "Swarm intelligence techniques in recommender systems - A review of recent research," *Swarm Evol. Comput.*, vol. 48, pp. 201–219, Aug. 2019, doi: 10.1016/j.swevo.2019.04.003.

[46] S.-C. Chu, H.-C. Huang, J. F. Roddick, and J.-S. Pan, "Overview of Algorithms for Swarm Intelligence," in *Computational Collective Intelligence. Technologies*

and Applications, Berlin, Heidelberg, 2011, pp. 28–41. doi: 10.1007/978-3-642-23935-9_3.

[47] M. N. A. Wahab, S. Nefti-Meziani, and A. Atyabi, "A Comprehensive Review of Swarm Optimization Algorithms," *PLOS ONE*, vol. 10, no. 5, p. e0122827, May 2015, doi: 10.1371/journal.pone.0122827.

[48] M. A. Ahmad and J. Srivastava, "An Ant Colony Optimization Approach to Expert Identification in Social Networks," in *Social Computing, Behavioral Modeling, and Prediction*, Boston, MA, 2008, pp. 120–128. doi: 10.1007/978-0-387-77672-9_14.

[49] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, Jun. 2014, doi: 10.1007/s10462-012-9328-0.

[50] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems," in *Foundations of Fuzzy Logic and Soft Computing*, Berlin, Heidelberg, 2007, pp. 789–798. doi: 10.1007/978-3-540-72950-1_77.

[51] D. Sethi and A. Singhal, "Comparative analysis of a recommender system based on ant colony optimization and artificial bee colony optimization algorithms," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2017, pp. 1–4. doi: 10.1109/ICCCNT.2017.8204106.

[52] "Documentation — gensim." https://radimrehurek.com/gensim/auto_examples/index.html#documentation (accessed Feb. 27, 2022).

[53] "Jupyter Notebook Viewer." https://nbviewer.org/github/rare-technologies/gensim/blob/develop/docs/notebooks/atmodel_tutorial.ipynb (accessed Jan. 03, 2022).

[54] "| notebook.community." https://notebook.community/gojomo/gensim/docs/notebooks/atmodel_prediction_tutorial (accessed Jan. 03, 2022).

[55] M. T. Dharmawan, H. T. Sukmana, L. K. Wardhani, Y. Ichsani, and I. Subchi, "The Ontology of IT Service Management by Using ITILv.3 Framework: A Case Study for Incident Management," in *2018 Third International Conference on Informatics and Computing (ICIC)*, Oct. 2018, pp. 1–5. doi: 10.1109/IAC.2018.8780478.

[56] M.-C. Valiente, E. Garcia-Barriocanal, and M.-A. Sicilia, "Applying an ontology approach to IT service management for business-IT integration," *Knowl.-Based Syst.*, vol. 28, pp. 76–87, Apr. 2012, doi: 10.1016/j.knosys.2011.12.003.

[57] "A Hotel Recommender System for Tourists Using the Artificial Bee Colony Algorithm and Fuzzy TOPSIS Model: A Case Study of TripAdvisor." https://www.worldscientific.com/doi/epdf/10.1142/S0219622020500522 (accessed Jan. 20, 2022).

[58] S. Forouzandeh, M. Rostami, and K. Berahmand, "A Hybrid Method for Recommendation Systems based on Tourism with an Evolutionary Algorithm and Topsis Model," *Fuzzy Inf. Eng.*, vol. 0, no. 0, pp. 1–25, Jan. 2022, doi: 10.1080/16168658.2021.2019430.

Appendix 01: System Front End