Addis Ababa University

College of Natural and Computational Sciences

Context-Aware Personalized Job Recommendation

Selam Abebe Abera

A Thesis Submitted to the Department of Computer Science in Partial Fulfilment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

June, 2020

Addis Ababa University

College of Natural and Computational Sciences

Selam Abebe Abera

Advisor: *Fekade Getahun (PhD)*

This is to certify that the thesis prepared by *Selam Abebe Abera*, titled: *Context-Aware Personalized Job Recommendation* and submitted in partial fulfilment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name	Signature	Date
Advisor: Dr. Fekade Getahun	_____	_____
Examiner: _____	_____	_____
Examiner: _____	_____	_____

Abstract

Companies often receive thousands of resumes for each job posting and employ dedicated screeners to short list quailed applicants. Searching for jobs online is an information intensive activity, because thousands of jobs are posted on the Web daily and it takes a great deal of effort to find the right position. Job search sites require recommender systems to meet diversified information needs. In this thesis work, we introduce a context aware job recommender which not only produces recommendation based on resume and job description, it also had integrated the preferences of the job seeker to enhance the recommendation. From the user's perspective, three different kinds of recommenders are implemented collaborative filtering based, content based and context/ preference based. Users of this system can retrieve jobs with different methods. From the recruiters' perspective, two different kinds of recommenders are implemented content based and context/ preference based. Recruiter can retrieve candidate job seekers based on their resumes or likelihood of the job seeker with the job based on the job seekers preferences. A challenge lies on the design of recommendation approaches since different job seekers might have diverse features and interests. To address the above-mentioned problem we integrate context/preferences of a user with their respective profile.

In our evaluation we show that personalized recommendation can be enhanced by integrating contextual information to a user profile.

Keywords: Recommendation system, Job recommendation, Personalized recommendation

Acknowledgments

First and for most I would like to thank my advisor Dr. Fekade Getahun for his time, guidance, patience and support throughout this thesis. And I would like to thank my family and friends for their support and love.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Abbreviations

CF	Collaborative Filtering
CB	Content Based
CBF	Content Based Filtering
CAPJR	Context Aware Personalized Job Recommender
ER	Entity Relationship
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
KNN	K Nearest Neighbors
UML	Unified Modeling Language

Chapter One –Introduction

1.1 Background

Recommender systems can help users deal with information overload problem efficiently by suggesting items (e.g., information and products) that match users' personal interests. The recommender technology has been successfully employed in many applications such as recommending films, music, books, etc. [1]. Recommender systems are being broadly recognized in wide range of applications to recommend products, services, and information items to users. Many e-commerce applications use recommender systems in order to increase customer services, increase selling rates and decrease customers search time. Context-aware applications utilize contextual information, such as location, preference, display medium and user profile, in order to provide tailored functionality [2]. The context of the user, such as the task he/she is working on, time of the day, location and device used, has a direct impact on the relevance of the recommended items [3]. Context is measured and represented by actors, activity spaces and assets in learning environments as well as explicit interest parameters such as tags and queries of the user. The context is defined as a set of factors of the objective environment, which affects the whole recommendation process including the selection of user profiles, the application of recommendation approaches and the output of recommendation results. For example, one kind of the context is the factor formed in the peak season and the off season. It affects directly the desire of a job applicant [4].

The recommender system approaches are categorized into the following main four categories: Collaborative filtering (CF) approach, Contend-based (CBF) filtering approach, Knowledge-based approach and Hybrid approaches. **Collaborative filtering** is one of the most popular approaches for construction of recommender systems. This approach makes use of the recognized preferences of a set of users to build the unknown preferences for new users. The essential assumption of collaborative filtering is that if users u and y rate x items in the same manner, or have similar actions. In future, they will rate other items in the same manner [5]. Content-based filtering in information retrieval method, the representation of a document have to be matched to representations of users on textual similarity. While, in machine learning problem, the textual content of the representations are joined as feature vectors, which are used for teaching a prediction algorithm [6]. In a content-based system, the objects of interest are defined by their associated features. For example, text recommendation systems like the newsgroup filtering system NewsWeeder [7] uses the words of their texts as features. A content-based recommender learns a profile of the user's interests based on the features present in

objects the user has rated. The type of user profile derived by a content-based recommender depends on the learning method employed. Decision trees, neural nets, and vector-based representations have all been used [8].

The content-based filtering recommends services or products whose content is alike to the content that the user has previously visited or selected [9]. Content based recommendation has been functional in various fields ranging from recommending product sale, web pages, news articles, television programs, movies and restaurants. [10] **Knowledge-based** recommender systems challenges to suggest items based on inferences about user's desires and preferences [8] . This method supports users in the determination of appropriate results from complex product and service assortments. These approaches based on using profound knowledge about the product dominion to figure out the best desires of the customer. In Knowledge-based recommendation methods, the association between customer requirements and products can be clearly modeled in an underlying knowledge base [11]. In Hybrid recommender **the** above three recommendation approaches be integrated in various ways to a hybrid recommender system to acquire improved performance.

Most companies put the attention on their own e-recruiting platforms as the main recruitment channels. Job advertisements are published automatically on the job portal as soon as they are arrived into the system. On the other hand, the job seeker creates a user profile to apply for one of the recorded job positions. The user profiles are stored in the system, allowing the job seeker to reuse it for other job position. This functionality gives the companies opportunity to create the applicants pool. Hence, the companies achieved a constant view for all applicants' profile and data in a candidate pool. The pool can be used by the recruitment subdivision to find the applicant documents. Suitable applicants' documents are engaged to the human resource subdivision for additional processing. In addition, the system supports all essential communication procedures as well as tracks applicant status inside the application process [12].

Job recommendation problem is bidirectional recommendation between applicant and job. The recommendation process can be separated into two parts: job recommendation and applicant recommendation. The design idea of these parts is more or less similar. For n applicant, the job with greater matching degree shall be recommended to the applicant. In the same way, for a job, the applicant with greater matching degree should be recommended to it. In general, the ranking items either are the top n candidates that best meet the job in consideration or the top n job profiles that best meet the candidates' preferences.

Currently the Internet-based hiring platforms is not a main recruitment channel in most organizations. While such platforms decline the recruitment time and commercial cost, they ache from an inappropriateness of traditional information retrieval techniques like the Boolean search methods [12]. Accordingly, a huge amount of applicants missed the chance of getting hired [13]. The recommender system technology wishes to help users in finding things that match their personnel needs; it has an effective usage in e-commerce domains to deal with problems associated to information overload efficiently.

For several years, information system provisions in human resource management have been mostly limited in storing and tracking job seekers data through the applicant management systems. These systems support the work flow and communication processes between the human resource management subdivision and the other subdivisions. Recently, the amplified amount of digital information and the appearance of e-business change the way organizations conduct business in different aspects. Initially, modest solutions are applied such as posting the job advertisements on the occupation unit of the company website. Then, based on the knowledges gained from these first implementations, the opportunities are understood, establishing other modifications and hence, implementing improved e-recruitment platforms. Numerous researches have been conducted to discuss diverse issues related to the recruiting problem as well as, the application of recommender system technologies.

The aim of this proposal is to design a Context Aware Personalized Job Recommender that uses better user profiling by integrating jobseekers preferences and assist online recruiting functions. While job-seekers search jobs by category such as location, job category or preference. And recruiters search applicants' by education, experience, preference. The proposed system recommend job openings by observing what kind of jobs and organizations the job seeker desires. It take into account additional contextual information such as location, job category, sector (the industry field of the hiring organization is in) and date. The system will also help the recruiting company to find the best fit candidate for the job.

1.2 Motivation

The importance of Information System support in the recruitment procedure can be observed when seeing the phases of the recruitment such as the management of candidates' applications and the pre-selection of job seekers. However, a best fit between job and job seers depends on fundamental aspects that are hard to measure. These fundamental aspects are a significant reason why information

systems have not been broadly used in the area of personnel selection so far. The main motivation to do the research work is the need to provide accurate recommendations for job-seekers that are searching for a job and for recruiting companies that are searching for the best candidates for a job. And this needs deep understanding of job characteristics and user's real demands.

1.3 Statement of the Problem

Recruiting process is an important function of human resource management giving the labor as one of the important factors of production. The main objective of the recruiting process is to hire job seekers who are valuable for the organization. Two perspectives are distinguished: from recruiters' and job seekers. The recruiters produce the job description by determining the set of requirements and constraints on education, skills and expertise levels. The job-seeker, on the other hand, produces his/her CV by postulating the academic background, previous work experience and skills. The information technology support for the recruiting processes range from appealing and finding the talent to select and keep candidates.

Making accurate recommendations requires deep understanding of item characteristics and user's real demands. This can be achieved by exploiting the abundant auxiliary information. For example, context information tailors services and products according to user's circumstances and surroundings [14], and mitigate cold start influence; Implicit feedback indicates users' implicit intention and is easier to collect while gathering explicit feedback is a resource-demanding task. Although existing works have investigated the efficiency of deep learning model in mining user and item profiles [15] [16], implicit feedback, contextual information, and review texts for recommendation, they do not utilize these various side information in a comprehensive manner and take the full advantages of the available data [17]. Without sufficient knowledge about users, even the most sophisticated recommendation strategy will not be able to make satisfactory recommendations. The cold start, sparseness, malicious rating are difficult problems for user profiling. They cause user profiles to become the weakest link in the whole recommendation process.

The data sparsity challenge give the impression in several conditions, specifically, the cold start problem take place whenever a new user or item has just arrived the system, it is difficult to find similar ones because there is not sufficient information (in some literature, the cold start problem is also called the new user problem or new item problem [18] [19]). New items cannot be recommended until some users rate it, and new users are unlikely given good recommendations because of the lack of their rating or purchase history [5]. The cold start problem, also known as the bootstrapping

problem, when there are not sufficient, or no prior history associated to a user's interaction with items within the recommendation system. In a condition like this, the capability of giving meaningful recommendations to the user is no possible [20].

The intention of this thesis is to come up with a new user profiling and personalized context aware job recommendation methodology. It is expected to improve the problems of the existing collaborative filtering cold start problem by making use of different new user profiling and analyzing methods. Thus the key research questions associated to this thesis work are:

- How can user profiles can be enhanced and used to improve personalized recommendation?
- How can preferences and contexts of users extracted and applied in the process of personalized recommendation and what advantages can it bring?

1.4 Objectives

General Objective

The general objective of this research is to model and develop context aware personalized job recommender that supports job-seekers and recruiting companies.

Specific Objectives

Specific objectives of this research are:

- ✓ Review researches that have been done in the area.
- ✓ Analyze personal context parameters in a job searching environment
- ✓ Propose a system architecture that suits the development of context aware personalized job recommender system.
- ✓ Enhance user profiling
- ✓ Produce capable recommendation
- ✓ Build a prototype of the system
- ✓ Test and evaluate the performance of the system.

1.5 Methodology

Different methodologies are employed in this research in order to accomplish the general and specific objectives of this study. The following phases will be conducted in the course of the research.

a. Literature Review

Conduct an extensive review of relevant literatures to acquire a deeper understanding of the research area and its problem domains. Previous researches in the areas of personalized recommendation, context awareness and related issues are investigated to visualize their importance towards this research. Existing works related to this research work are also assessed to identify and point direction in providing solution to identified problems. Moreover, limitations of those studies and the approaches they have used will be reviewed.

b. Resource gathering

Using the inputs from conducted review of literatures, useful models and design requirements are then identified to arrive at practical solutions. Based on these requirements, a model of context and design of architecture is proposed which can guide implementation of application. Tools and techniques that can fulfill the design of the architecture are then labeled for the implementation of the prototype.

c. Prototype development

The prototype considers the necessary identified services to prove the significance of the proposed solution so that it should carefully follow the specification provided in the model and architecture. The prototype application will be developed python programming language

d. Testing and Evaluation

To assess the performance of the proposed job recommender, the prototype is tested using jobs information (job profile) and job seekers information (job seeker profile and context). The significance of the retrieved recommendations will be checked. The result of the test will be evaluated by applying precision and technique.

1.6 Scope and Limitations

The scope of this work is to design and build a web based context aware personalized job recommender. It takes into account the contextual information of users to recommend job. Also the system recommends the best fit candidates to a job opening based on the predicted score.

Implementation is limited to developing a prototype. The proto type is limited to viewing recommendations.

1.7 Application of Results

The context aware personalized job recommender can be implemented in organizations which wants to hire new personnel and solve the problem of resource wastage caused by labor-intensive activities. And job- seekers who are looking for a job can use this system to find thir desigerd jobs based on their resume and personal taste. The result of this research work will have an application on assisting both the job-seeker and the hiring company because it adds the necessary context information of users to find a job.

1.8 Organization of the Rest of the Thesis

The rest of the document is organized as follows. In Chapter Two review of literature about our thesis is presented. Key concepts in personalized recommendation, approaches to recommendation, context-awareness, profile modeling and context modeling is described.

In Chapter Three related works that have been done in the past are thoroughly discussed. Architectures in a personalized job recommendation are presented with their strength and weaknesses and summary of these related works is also briefly outlined.

Chapter Four deals with the design of an architecture for modeling context context-aware application for mobile phone. The design goals that should be achieved in designing an architecture for the thesis are clearly stated and an architecture that fulfills those design goals is designed. Components of the architecture are described in detail with algorithms that show how they can be implemented.

In Chapter Five the implementation of the prototype for the research is presented. Tools and techniques that have been used for developing the prototype are stated and description about how each component is implemented is presented. An implementation scenario that can show the implementation is also presented along with some user interfaces for demonstration. Lastly, conclusion is given in Chapter Six. The main contributions of this research are outlined and possible future works are indicated.

Chapter Two - Literature Review

2.1 Personalized Recommendation

Recommender system (RS) is a useful information filtering tool for guiding users in a personalized way of discovering products or services they might be interested in from a large space of possible options. Recommender system has been playing a more vital and essential role in various information access systems to boost business and facilitate decision-making process. In general, the recommendation lists are generated based on user preferences, item features, user-item past interactions and some other additional information such as temporal and spatial data. Recommendation models are mainly categorized into collaborative filtering, content-based recommender system and hybrid recommender system based on the types of input data [1] . However, these models have their own limitations in dealing with data scarcity and cold-start problems, as well as balancing the recommendation qualities in terms of different evaluation metrics [17]. Recommender systems are being applied in various applications to suggest products, services, and information items to customers. Many e-commerce applications join recommender systems in demand to increase customer services, increase selling rates and decrease customers search time.

Recommender systems combine ideas from user profiling, information filtering and machine learning to deliver users a more intelligent and proactive information service by making concrete product or service recommendations that match their learned user preferences and needs. The recommender technology is superior to other information filtering applications because of its ability to provide personalized and meaningful information recommendations. For example, while standard search engines are very likely to generate the same results to different users entering identical search queries, recommender systems are able to generate results to each user that are personalized and more relevant because they take into account each user’s personal interests [1].

2.2 Approaches to Recommendation

The four main recommender system approaches are classified the following four categories: Collaborative filtering, Contend-based filtering, Knowledge-based and Hybrid approaches [6] .

2.2.1 Collaborative Filtering (CF)

Collaborative filtering one of the most fruitful approaches for construction of recommender systems. This approach applies the known preferences of a set of users to predicate the unknown preferences for new users. The fundamental assumption of collaborative filtering is that if users' u and y rate n items in the same way, or have similar behaviors. They will rate other items in the same way [5]. The ratings can either be explicit that states to a user expressing his/her preference for an item using the numerical scale such 1–5, or implicit that states to concluding the user behavior or selection to assign the user preference [21] . Collaborative filtering approaches have the ability of working in areas where items contents are challenging to acquire or cannot be parsed automatically. However, collaborative filtering techniques can deliver unexpected recommendations, which are not similar to the items in the user's profile, but concern him/her [22] [23]. Examples of recommender systems that established on collaborative filtering techniques are presented by [24]. The collaborative filtering approaches are categorized into two main types: Memory-based and Model-based methods [21] [18].

Memory-Based CF methods

This type makes use of an example of user-item database to produce predictions. Each user is part of a group of users with similar needs. When recognizing the neighbors of a user, the user's prediction for preferences of new items can be formed [21]. Users are compared against each other directly using correlation or other methods [8]. Additionally, The Memory-based collaborative filtering methods embrace the user-based and item-based correlation/similarity measures. The user based measures forecast a target user's upcoming preferences by combining the detected preferences of similar users. The procedure first computes a user similarity score which is calculated based on the vector similarity function. A great similarity score shows that the two users have alike preferences [21] [24]. On the other hand, the item based measures are unlike from the user-based measures only in that item resemblances are computed instead of user resemblances. A great similarity score indicates that the two items are alike because they have been selected by numerous users [24].

Model-based CF methods

This is a technique in which a model is created from the historical rating and used to infer the predictions [21]. The improvement of models allows the system to learn and recognize difficult patterns using the training data, and then produce forecasts for test data. Model-based collaborative

filtering approaches applied techniques such as Bayesian models, clustering models, and dependency network to resolve the weaknesses of memory-based CF methods [5].

Characteristics and challenges of CF

The foremost characteristic of collaborative filtering approaches is that they are completely independent of any machine-readable symbol of the objects being recommended, and they work well for difficult objects such as sounds and movies where variants in flavor are affected the variation in preferences. On the other side, there are numerous main challenges suffered by collaborative filtering such as cold-start difficulties that include data scarceness and ramp-up problems. In the data insufficiency problem, there is lack of historical data. For example, in many real world applications, users' historical data, such as what they have liked, viewed, bought or rated, is sparse by nature because the website is in its early operational stage. Therefore, it is extremely possible that either the resemblance between any two users is nearly zero or the measures are undependable. In the ramp-up problem, while there is a great amount of users whose likings are identified, the system cannot be valuable for new users till an adequate amount of items' rating has been composed [25]. The second challenge is the scalability, when the amount of existing users and items rise extremely, the collaborative filtering techniques will undergo serious scalability difficulties, with computational resources going past practical or acceptable levels [5]. Dimensionality reduction techniques such as Singular Value Decomposition (SVD) [26] can deal with the scalability problem and quickly produce decent recommendations, but they have costly matrix factorization processing [5].

2.2.2 Content-based filtering (CBF)

In information retrieval method, the document representations have to be matched to user representations on textual resemblance while, in machine learning problem, the textual content of the representations are joint as feature vectors, which are used for teaching a prediction algorithm [6] . The content-based filtering recommends items whose content is alike to the content that the user has previously viewed or liked [9]. Content-based filtering has been applied in various areas ranging from recommending movies, music, web pages, news articles and items for trade [10].

There are two foremost tasks associated to content-based filtering recommender systems, the User profiling and the item representation. User profiling is one of greatest puzzling tasks in CBF recommender systems that deal with obtaining, extracting and express the features of users. User profile is regularly created automatically in response to user feedback on the importance of items that have been offered to the user. This profile may contain different types of information such as the

selected items, ratings of items, and user's demographic data, etc. [27]. However, the user interface can straightforwardly be created to support users building their profiles. Authors of [10] categorized the profile information into two kinds: (1) the user's likings such as item description that concern the user. There are numerous possible representations of this description, but the common representation is using a function to forecast the possibility of user is attracted in that item. (2) The user's interactions history with the recommendation system that contains saving the items that a user has viewed with information about user's interaction. Item representation is also a significant issue in content-based filtering recommender systems. Items can be a structural data represented by the same set of attributes, and there are explicit values that the attributes may have. Numerous methods for learning a structural data used such as machine learning techniques. Furthermore, unstructured data may take place in some applications such as unrestricted texts in news articles. In this type, there are no attribute names with well-defined values. A communal approach to deal with free text fields is to interchange the text to a structured representation. Each word can be treated as an attribute, associated by Boolean value representing the accessibility of the word in the article with an integer value representing the number of incidences of the word in the article [10].

Characteristics and challenges of CBF

The strong characteristics of content-based filtering approaches are that there is no need of domain knowledge, and they are satisfactory to collect implicit feedback from users about their item preferences. This make CBF the finest algorithm in domains where acquiring clear ratings from users is difficult or unwieldy, and where area knowledge is hard to study. CBF techniques have a ramp-up problem in that they must gather sufficient ratings to construct a dependable classifier. Moreover, they are limited by the features that are explicitly associated to the objects that they recommend [22].

1.2.3 Knowledge-based

Knowledge –based recommender systems efforts to propose objects based on inferences about user's needs and preferences [8]. This approach supports users in the determination of appropriate solutions from difficult product and service assortments. These solutions based on exploiting deep knowledge about the product domain to figure out the best requirements of the customer. In Knowledge-based recommendation techniques, the association between customer desires and products can be explicitly showed in an underlying knowledge base [11]. They can use rules and patterns to recommend items based on functional knowledge of how an explicit item meets an individual user need [8].

Knowledge-based recommendations achieve reasoning about what products meet the user's requirements by engaging techniques such as a quantitative decision support tools [28].

Characteristics and challenges of knowledge-based approach

Knowledge-based approach does not need to gather information about a detailed user because its judgments are free of individual flavors. They do not have ramp-up problem because its recommendations do not subject to user ratings [25]. These characteristics make knowledge-based recommenders appreciated systems on their own, as well as, greatly complementary to other types of recommender systems [29]. The core challenges as all knowledge-based systems are they need knowledge acquisition and knowledge engineering with all of their associated difficulties [25].

1.2.4 Hybrid recommender

Collaborative filtering, content-based filtering and knowledge-based approaches can be combined in diverse ways to a hybrid recommender system to acquire improved performance. All recommendation approaches have characteristics and challenges. To acquire improved performance and overcome challenges, these approaches can be joined. In overall, collaborative filtering is integrated with other methods in an effort to avoid the prior mentioned challenges [8] . As authors of [8] [29] presented different ways to incorporate collaborative filtering, content-based filtering and knowledge-based methods into a hybrid recommender system that classified as follows:

- Weighted hybrid recommender: The score of item recommendation is calculated from the results of all of used recommendation approaches that are available in the system.
- Switching hybrid recommender: The system uses some measure to switch between recommendation techniques.
- Mixed: In which huge number of recommendations are applied at once.
- Feature Combination uses the collaborative information as further feature data for each example and use content-based methods over this improved data set.
- Cascade: It comprises a staged process. In this method, one recommendation approach is used first to yield a rough ranking of candidates and a second approach refines the recommendation.
- Feature augmentation: One approach is used to yield rating or classification of items and that information is then combined into the processing of the next recommendation approach.
- Model: Where an output of one approach is used as an input for another approach.

2.3 User Profiling for Recommender Systems

The essential recommendation making mechanism of current recommender systems is to firstly identify the target user's neighbors based on user profile similarity, and then suggest the target user items that the neighbors have liked in the past. User profiling is the process of acquiring, extracting and representing the features of users. The profile data may include users' selected items, ratings for specific items, and demographic data etc. User profiling is one of most challenging tasks. In current recommender systems, user profiles are usually generated based on data with limited relevance that are too simple to produce quality recommendations [18]. Job seekers' profile that is used in our work includes demographic data, educational background, experience, skills, users selected and weighted preferences.

Massive quantities of User Generated Content (UGC) on social networks are now available from blogs, tags, item reviews, knowledge-sharing sites, collaborative filtering systems, online gaming, newsgroups, chat rooms, etc. A warehouse of UGC can be mined and analyzed to expand user profiles based on which more reliable recommendations can be made to users. The richness of the online UGC challenges the current personalization techniques and also provides new possibilities for accurately profiling users [1].

The user profile is used in most disciplines whose main objective is the personalization, customization and adaptation. However, it is far from being defined as standard. This constitutes all information explicitly provided by the user himself or by all traces left on any computer system [30]. User profiles can also be created through consumer information explicitly specified when they initially register or subscribe to service [31].

A. Information defining the user profile

The user profile is far from being defined as standard. This constitutes all information explicitly provided by the user himself or by all traces left on any computer system. This information is usually of a different area of use to another. However, they allow in all cases to answer the following five questions: "Who are you?", "Where are you?", "What are you doing?", "Why do you do it?", "When do you do it? " and "how are you doing it?". The first three questions correspond to its properly known identity. Regarding the next three questions, they correspond to the active identity [31]. According to the authors of [32], the user profile can be divided into two main dimensions. The first dimension represents the interests of the long term user or what they called "persistent dimension", on the second dimension, it represents short-term user interests "evolutionary dimension". The first dimension

includes all the information for defining the identity of the user. These change very rarely and are composed of personal data (name, surname, date of birth ...), field of interest represented by a set of terms describing them. The second dimension, by cons, contains all the information to describe the user for a short term. These are all information by which the user is interested during its interactions with the system.

The information contained in a user profile is different from a domain to another. The authors [30] have proposed a generic model of the user profile. The latter is divided into six dimensions defined by a set of attributes that we categorized in Table2.1. The general point of view, the user profile is created for a specific objective. In other-words, it is oriented context. However, it possible to define the profile so that it will be possible to reuse the (at least part) in various contexts. This is the converse into two different parts. The first part consists of information that is independent of the context (name, surname, date of birth, etc.), as to the second part, it comprises information specific to each application.

B. User profile representation

Collection and selection of information that define the user profile is a key step in the customization process. However, the definition of a representation of its information model is also important. Several approaches have been developed for this purpose. The most used is the one that allows structuring the user profile as a vector model. In this model, each point of interest is represented by a set of terms. However, there are three main approaches: ensemblist, semantic and multidimensional.

- **Ensemblist representation**: In this approach, the user profile is formalized in the form of weighted terms of vectors.
- **Semantic representation**: the user profile is shown in this case as a set of terms related to each other by semantic relationships. These terms correspond generally to the concepts of the domain ontology in which the user is interested.
- **Multidimensional representation**: in this approach, the information in the user profile is grouped together as dimensions. Each dimension is an attribute class. In this context, the authors of [33] have attempted to define a set of dimension. This is personal data, focus, expected quality, delivery preferences, and security and interaction history.

C. Acquiring information of a user profile

The acquisition of information about a user is a key process in the construction of the profile. This process has become easier now through the Internet and especially social networks and search engines. This information can, as we mentioned earlier in this chapter, be provided explicitly by the

user himself or collected implicitly by the system during the sessions of use. It is also possible to combine these two approaches in most systems.

2.4 User Profiles Learning Issues in Recommender Systems

A crucial factor for the success of recommender systems is the availability and quality of the user profiles. The user profile information can be input explicitly by users or implicitly gathered by software agents that monitor user activity [34]. For explicit acquisition, users are required to rate or select items. For implicit acquisition, the users' behaviors will be passively observed as they interact with the system and then the users' interests will be inferred from these interactions. Currently the user profile information for online recommendation is mainly obtained by analyzing usage log data such as users' click streams and navigation patterns etc. Both the explicit and implicit methods have their respective strengths and weaknesses. Explicit method depends on a user's active participation for describing the user's interest to information items, forcing the user to accept extra load, it may place an increased cognitive burden on the users [35] . The implicit acquisition places little or no burden on the users. However, inferences drawn from the user interaction are not always valid because of the indicators of the user interests are often unpredictable [36].

The user profiles are often difficult to obtain and their quality is also hard to ensure. Current existing user profiling for recommender systems is mainly using user rating data. Usually, hundreds of thousands of users and items are involved in a recommender system, but only a few items are viewed, selected or rated by users. As Sarwar et.al reported in [37], the density of the available ratings in commercial recommender systems is often less than 1%. Moreover, as for new users, they will start with a blank profile without selecting or rating any items at all. These situations are commonly referred to as the data sparseness and cold start problem [38]. The current recommender algorithms are impeded by the scarcity and cold start problems. With the increasing use of recommender systems in e-commerce and social networks, maliciously or unfairly influences to the outcomes of recommender systems by creating false user rating data are also intensified. For example: a simple but effective attack to recommender system is to deliberately create a bunch of fake users with pseudo ratings favor or disfavor to some particular products. With the fake information, user profile data becomes unreal and not reliable.

Without sufficient knowledge about users, even the most sophisticated recommendation strategy will not be able to make satisfactory recommendations. The cold start, sparseness, malicious rating are

formidable problems for user profiling. They cause user profiles to become the weakest link in the whole recommendation process.

2.5 Discovering Features of Documents

TF-IDF

Tf-idf is a numerical statistic used in information retrieval to represent how important a specific word or phrase is to a given document. The tf-idf value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

Term Frequency – TF is about how often the term appears in this document. The more often, the higher the weight. A field containing five mentions of the same term is more likely to be relevant than a field containing just one mention.

Inverse Document Frequency – IDF is about how often does the term appears in all documents in the collection. The more often, the lower the weight. Common terms like **and** or **the** contribute little to relevance, as they appear in most documents, while uncommon terms like **future** or **SEO** help us zoom in on the most interesting documents.

Tokens include punctuation and other specialized symbols that have been engineered out of the structure found in the article headers. Using noun phrases as tokens has also proven [39] to be useful. In NewsWeeder [7], raw text is first parsed into generalized words, called tokens. Next, NewsWeeder creates a vector of token counts for the document. This vector is the size of the total vocabulary with zeros for tokens not occurring in the document. Using this type of vector is sometimes called the bag-of-words model, and is the basis for their representation. While the bag-of-words model does not capture the order of the tokens in the document, which is necessary for linguistic or syntactic analysis, we assume it captures most of the information needed for filtering purposes. As a next step, it is common in IR systems to group the tokens together by their common linguistic roots, a procedure called stemming. In NewsWeeder, the tokens are left in their unstemmed form. While better performance could probably be achieved by stemming in the short-term due to the larger statistical samples it creates, the long-term research goal is to not waste the extra information contained in the use of exact tokens. We plan to eventually try to use this extra information by looking at larger-size samples of unrated text to get more reliable statistics on rare words. This technique would attempt to use the large amounts of available unrated text in an unsupervised learning approach to bias the supervised learning on the smaller set of rated articles. Further, there is evidence [40] that stemming

can hurt performance if the approach used is able to make strong enough statistical inferences about the unstemmed tokens.

The assumptions behind tf-idf are based on two empirical observations regarding text. First, the more times a token t appears in a document d (called the term frequency), the more likely it is that t is relevant to the topic of d. Second, the more times t occurs throughout all documents (called the document frequency), the more poorly t discriminates between documents. For a given document, these two terms are combined into weights by multiplying the tf by the inverse of the df for each token. Often the logarithm of tf or idf are taken in order to de-emphasize the increases in weight for larger values. The way in which tf-idf vectors are compared also takes advantage of the domain. Because documents usually contain only a small fraction of the total vocabulary, the significance of a word appearing is much greater than of it not appearing. To emphasize the stronger information content in a word appearing, the cosine of the angle between vectors is used to measure the similarity between them. TF-IDF is widely used to construct a weight vector for predefined terms by using content information [18].

2.6 Measuring similarity

Cosine similarity

An important step of achieving matching jobs and job seekers is calculating the similarity or relevance based on their profiles. Recommender systems use numerous various similarity functions to compute similarity between users, between items or between users and items: some similarity functions are heuristic and others are learnt models from underlying data using machine learning techniques [41]. Two well-known similarity measures [18] [42] are cosine similarity and Pearson Correlation Coefficient (PCC). Cosine similarity measures the cosine of the angle between two vectors. Common similarity measures work on some specific problems but do not work on others [41]: cosine similarity yields better results in item-item filtering systems [42] but in content-based recommender systems, if the set of terms used by users is different from the set of terms used by jobs, the computed similarities between users and jobs using cosine similarity or PCC are almost always 0 [41].

K-Nearest Neighbors (KNN)

KNN algorithm is called K nearest neighbor classification algorithm. The fundamental idea of the KNN algorithm is: if the majority of the k most similar neighbors of sample in the feature space belongs to a certain category, then the sample is considered to belong to this category [43]. Since the significant amount of events, the application of collaborative filtering is straightforward. As the user

item matrix is sparse in some dataset, nearest neighbor methods [37] are preferred over matrix factorization. Their application has a goals, one of the goal is based on the interactions, to find the best next items [44]. The KNN algorithm doesn't make any assumptions on the underlying data distribution, but it relies on job feature similarity. When a KNN makes a prediction about a job, it will calculate the distance between the target job and every other job in its database. It then ranks its distances and returns the top k nearest neighbor jobs as the most similar job recommendations. The weight of the KNN approach increases as more interaction data of the user is available [45].

2.7 Prediction

The most important step in recommendation is to generate the output in terms of prediction. Once we separate the set of most similar jobs based on the similarity measures, the next step is to look into the target job seekers ratings and use a method to get predictions.

Weighted Sum

This method calculates the prediction on a job i for a job seeker u by calculating the sum of the ratings given by the job seeker on the jobs similar to i. Each ratings is weighted by the corresponding similarity between items i and j. This method tries to capture how the job seeker rates the similar jobs. The weighted sum is scaled by the sum of the similarity terms to make sure the prediction is within the predefined range.

2.8 Context and Context Awareness

The idea of contexts is the modelling of context-aware information system, which acts as a kind of assistant providing information needed in the current situation. Hereby, the context represents a part of the user's situation [46]. Context aware system, should consider designing context information wisely for its better use in the computing world. The authors of [47] describe context information as that exhibits a range of temporal characteristics which is highly interrelated, imperfect and can be represented in many ways. Context-aware systems should deal with such characteristics by representing aspects of context information such as how the information representing the context is obtained; how it is represented; and, for what objectives and purposes the context information is used for [48]. This is where context acquisition techniques, context model and its modeling approaches should come in. The reason for modeling context is to better understand users' behavior, to find out their desires related with the personal activity and to respond to them with the essential information required.

Context Modeling Approach

Context aware modeling approach is required to design model of context to support context reasoning and other requirements of context data manipulation. Modeling approaches to model context are presented in different researches [49], [50], [51] based on the data structures used for representing and exchanging contextual information in the respective system. The authors in [49] present relevant context modeling approaches and evaluated the approaches with requirements. The requirements include distributed composition, partial validation, richness and quality of information, incompleteness and ambiguity, level of formality, and applicability to existing environments. Most researches classify context modeling approaches into six categories [49] [52] as key value models, markup scheme models, graphical models, object oriented models, logic based models and ontology based models. The context modeling approaches investigated are discussed below in this section.

A. Key-Value Models

Key-Value models are frequently used and represent the most simple data structure for context modeling which employ matching algorithms using key-value pairs for service discovery. Key-value pairs are easy to manage but lack capabilities for sophisticated structuring for enabling efficient context retrieval algorithms.

B. Markup Scheme Models

Markup scheme models use a hierarchical data structure consisting of markup tags with attributes and content which is usually recursively defined by other markup tags. The strengths of markup scheme context modeling approach are partial validation and applicability to existing markup-centric infrastructures of pervasive computing environment. Scheme definition and a set of validation tools exists which enable type checking for complex types and range checking to some degree for numerical values. However, it doesn't satisfy the requirements such as incompleteness and ambiguity, and quality of information while distributed composition is addressed to some level. A comprehensive scheme definition is a step towards a high level formality and thus may be used to determine interoperability.

C. Graphical Models

Graphical models use a modeling instrument called Unified Modeling Language (UML) which has a strong graphical component to model context since it is appropriate due to its generic structure. Applicability requirement is addressed well in graphical models due to its strength on the structure

level which used to describe structure of context knowledge and to drive a model. Partial validation and distributed composition are possible with the latter one having some constraints at structure level where as level of computer evaluable formality is relatively low. UML is appropriate to model context because of its generic nature. Examples of this approach can be found in [46].

D. Object Oriented Models

Object oriented models use object oriented techniques. It incorporates advantage of object orientation comprising encapsulation, re-usability and inheritance to cover parts of the problems arising from the dynamics of the context in pervasive environments. The details of context processing is encapsulated on an object level and hence hidden to other components. Access to contextual information is provided through specified interfaces only.

Object oriented context modeling approach is strong regarding the distributed composition requirement. New types of contextual information as well as new or updated instance can be handled in systems in a distributed fashion. Partial validation, and handling incompleteness and ambiguity are possible. A higher level of formality can be reached but with few application drawbacks. Applicability to existing object-oriented ubiquitous computing runtime environment is given with some requirements which often cannot be fulfilled in ubiquitous computing systems.

E. Logic Based Models

Logic based models use facts, expressions and rules in a high degree formality to define a context model. A formal system is applied to describe the conditions in a set of rules. Inference (also called reasoning) process can be used to derive new facts based on existing rules in the systems. Usually contextual information is added to, updated in and deleted from a logic based system in terms of facts or inferred rules in the system. Logic based context models have distributed composition as their strength, but partial validation is difficult to maintain. This makes the specification of contextual knowledge very error-prone though the level of formality is high. Quality of information, and incompleteness and ambiguity are not addressed while applicability to existing ubiquitous computing environments seems to remain as a major issue.

F. Ontology Based Models

Ontology based models represent contextual information using description of concepts and relationships with high and formal expressiveness and reasoning techniques. Concepts and their interrelations can be defined using ontologies. It is suitable to project parts of the information

describing our daily life onto a data structure utilizable by computers. Ontology based context modeling approaches are strong regarding the distributed composition requirement. Incompleteness and ambiguity are also addressed. Partial validation is possible, and a comprehensive set of validation tools do exist.

2.9 The Concept of Context in Recommendation

As the authors of [2] context-aware applications utilize contextual information, such as location, display medium and user profile, in order to provide tailored functionality. The context of the user, such as the task he/she is working on, time of the day, location and device used, has a direct impact on the relevance of the recommended items [3]. For our job recommender location interests of the job seeker is taken as a context. Context is measured and represented by actors, activity spaces and assets in learning environments as well as explicit interest parameters such as tags and queries of the user. The context is defined as a set of factors of the objective environment, which affects the whole recommendation process including the selection of user profiles, the application of recommendation approaches and the output of recommendation results [4].

Not much research has been done to incorporate contextual information of the user in the recommendation process [53]. The context of the user, such as the task she is working on, time of the day, location and device used, has a direct impact on the relevance of the recommended items [3]. New challenges emerge for capturing and understanding the context of the user and exploiting such contextual information for creating intelligent recommendations adapted to the current, contextual, needs of the user.

El Helou, Salzmann and Gillet [54] present the 3A recommender system that targets context-aware recommendation in personal learning environments. The authors give interesting insights into technology that can be used to extract contextualized user profiles from emerging information systems. Context is measured and represented by actors, activity spaces and assets in learning environments as well as explicit interest parameters such as tags and queries of the user. The authors propose a contextual and multi-relational ranking mechanism that adapts a version of Google's PageRank algorithm to the particular modelling framework, recommending to users not only assets (content), but also relevant activities and actors to interact with. This approach is an interesting alternative to existing conceptualizations of the various actors and components in knowledge sharing environments, offering contextualized recommendations that consider all types of relations between actors and components.

Butoianu, Vidal, Verbert, Duval [55] and Broisin and Niemann, Scheffel, Friedrich, Kirschenmann, Schmitz and Wolpers [56] also base their work on the monitoring of usage interactions with tools and resources as a basis to capture context information. A comparative analysis of approaches to capture such interactions is presented by [55]. The authors have compared several modelling approaches and architectures for managing such data against several criteria, such as flexibility, extensibility and scalability. Then, they present a framework that takes into account the advantages of the presented approaches and that tackles their limitations.

Niemann, Scheffel, Friedrich, Kirschenmann, Schmitz and Wolpers [56] present a new approach for calculating item-based similarity in order to support collaborative recommendation. A context-based usage similarity measure is presented and contrasted with the results of classic content-based item similarity. The authors then discuss scenarios of how recommendations may be supported by exploiting the pre- and post-context in which an item was used.

2.10 Job recommendation

Regularly, information system technology is used to pre-select job seekers based on Boolean search method. This method used queries contain a mixture of key words that define skill requirements in order to determine those job seekers that match with search standards. Such type of skill matching is practical in numerous e-recruiting applications. Though, the modest filter approach such as Boolean search method cannot be enough to recognize the complexity of a person-job fit as selection decisions often hang on on underlying attributes such as personal features or social skills that cannot be put into an operational way easily [57]. Moreover, the need to recognize the job requirements, in terms of the skills that are required and those that are optional but preferable, the experience criteria if any, preference for the location of the candidate etc. Consequently, the main challenge faced e-recruiting applications as recognized by the literature analysis is the great number of low qualification of applicants that match the search criteria [58] .

The recommender systems approaches can be used to address the problem of information overload by prioritize the supply of information for individual users grounded on their learned preferences [59]. In addition, the achievement of personalization technologies depends critically on the presence of comprehensive user profiles that precisely capture user's interests [60] and the matching approach. Furthermore, the recommender systems could use past rating information to determine which type of job required which type of job seeker characteristics in the past in order to be rated positively by the

recruiter. This information could then be used to forecast the match between job and previously not rated job seekers. The need of applying the recommender system approaches for selection process can be inspired from different viewpoints [57].

The Recruiting Process

Recruiting process is a central function of human resource management treating the labor as one of the important aspects of production [61]. The main objective of the recruiting process is to hire job seekers who are valuable for the organization [62]. Two perspectives are distinguished: from recruiters and job seekers. The recruiters generate the job description by determining the set of requirements and constraints. The job-seeker, on the other hand, generates his/her CV. The information technology support for the recruiting activities is ranging from appealing and finding talent to choose and retain job seekers [63]. The degree of process combination represents the complexity of using e-recruitment solutions [64].

As authors of [61] demonstrated in their proposed model, the association between recruiting tasks and divided the recruiting process into two main phases: The attraction phase and the selection phase, both phases cover a planning and an execution part. The planning part regulates the overall strategy and definite measures to attract valuable job seekers as well as, the explicit selection approaches. The execution part involves the employer branding activities that contain all long-term marketing measures that appeal qualified job seeker. The attraction phase aims to generate a description for open job positions. The selection phase starts with the pre-screening of resumes and additional submitted resources. Then, the final selection of candidates is accompanied by comparing the remaining set of job seekers that has not been filtered out in the screening phase. Lastly, the applicant management works as a secondary function; it consists of the contact of applicants, the management of applicant data and related processes such as directing applications to organization's members involved in the selection decision [12].

Moreover, [65] presented four phases of the recruiting process: an assessment of job position that needs to be occupied, a description job profile, the production of a job description and a candidate specification. Also, [66] composed the recruiting process into five main tasks: short-term and long-term candidate attraction, applicant management, pre-selection as well as the final selection of candidates. Short-term and long-term marketing measures are creating the attractive employer image that intended to attract qualified candidates.

2.11 Job recommendation techniques

Numerous recommender system approaches applied in job seeker/job matching problem, started by the personnel selection approach that proposed by [61] who developed a probabilistic hybrid recommendation method for job seeker/job matching. Then, their model used and extended by [57] [67] [68].

2.11.1 Hybrid job recommender systems

A probabilistic hybrid approach

In [61] applied a recommendation system originally used to recommend items to users such as movies or books to similar partners. The recommendation approach used both concepts: content-based filtering and collaborative filtering at once. This assists moderately to overcome the problem of data insufficiency. Alternative concept that they applied is the latent aspect model described by [69]. It recognizes the individual preferences as a convex combination of preference factors. In a straightforward method for collaborative filtering, we look at each value of user/item pairs *(x, y)*, where *x* is a set of users and *y* is a set of items. The aspect model can then be symbolized as a variable *z* which is related with each value of *(x, y)*, presumptuous that *x* and *y* are independent conditioned on *z*. The model parameters are then assessed using the Expectation Maximization (EM) algorithm. This model produced a rating matrix that assigns assessed values to candidate's profile holding the chance that recruiter *x* rates candidate *y* with value *v*. Latter, they defined *v* = {"qualified", "not qualified"}. Then, they converted the rating matrix by substituting variable *y* with a variable *a* to characterize the attributes that was extracted from the candidate resumes. As numerous attributes are assigned to several profiles, we will see the attribute *a* numerous times with different values *v*. The entries of the transformed matrix are essentially not either 0 or 1 but take values in the interval [0;1] depending on the relative frequency of value *v* being allocated to attribute *a* by recruiter *x*.

Grounded on the aforementioned model proposed by [61] and [57] applying this model into two separate recommendation systems in order to expand the match between job seekers and jobs: a CV recommender and a job recommender, distinctly. In the first stage, they built a system recommending CVs that are alike to resumes formerly selected by the recruiter for a specific job profile. In the second stage, they developed a second recommendation system that recommends jobs to job seekers grounded on their preference profiles which are in turn grounded on previous preference ratings. Moreover, [70] integrates these previous researches into a united multilayer framework to support the matching of job seekers for job and team member who will collaborate with them. Later, [68] applied and prolonged a decision support system for team building using the probabilistic hybrid method. They integrate a trust into the recommender-based approach. They argue that a decision

support system for team building needs to consider relational attributes such as trust in order to determine a fit between the candidate and current team members.

A proactive job recommender system

The proactive recommender system is an adaptive system that make an effort to incorporate the idea of recommender systems [71] and adaptive hypermedia [72]. This system holds five components: web spider, ontology checker, profile analyzer, preference analyzer, and user interface generator. Web spider is a parser that occasionally acquires job information from an external source. The ontology checker matches information with ontologies and implements the classification. Then, the job data is kept in a pre-designated form. The profile analyzer creates the recommendations, whenever the users modify the group of favorites by associating the weight differences with current open jobs. Then, a list of recommended jobs is produced. Lastly, the preference analyzer infers the explicitly defined user's preferences and provides a recommendation for favored jobs after calculating the resemblance of jobs to user's preference [59].

Semantic matchmaking for job recruitment

In [73] the authors tried to increase the matching process by given that an adaptive job offering and discovery atmosphere. They joined different matchmaking approaches in a hybrid approach for matching job seekers and jobs using logic-based and similarity-based matching. Primary, they applied a deductive approach to determine the match between individual and job, and then they used a resemblance measure to rank the applicants with fractional match.

A fuzzy multiple criteria method for recruitment

It is a method that attempts to determine the appropriate personality behaviors and key specialized skills through information statistics and Analytic Hierarchy Process. The Analytic Hierarchy Process is a multi-objective decision making method that is applied in improbability of decision-making matters with other assessment standards. Author implemented a study started by questionnaire survey and criteria assessment. Then, the weight of appropriate factors was determined based on Analytic Hierarchy Process, and by means of fuzzy multiple criteria algorithm. This algorithm is based on triangular fuzzy number and linguistic variable, which is used to assess the significance and acceptable level of certain criteria. Finally, based on the comprehensive assessment the applicants' scores were calculated as a foundation of recruitment [74].

2.11.2 Content-based job recommender systems

Machine learned recommender system

The recommendation problem preserved as a supervised machine learning problem. They construct an automated system that can recommend jobs to job seekers based on their past job histories, in order to facilitate the process of selecting a new job. An item in this learning approach represents a person who is hired in a company. Each item is described by set of features extracted from the candidates' resumes. Given a person who is presently working in a company, they want to forecast the next company. If the accuracy of such forecasts is sufficiently high, the approach can be used to recommend companies to employees who are looking for for jobs. This approach uses all past job changeovers as well as the data of both employees and companies to predict an employee's next job transition. They train a machine learning approach using a huge amount of job transitions extracted from person profiles obtainable in the web [70].

A system for screening candidates

Authors of [58] have presented the PROSPECT system, which is a decision support tool supporting recruiters to shortlist candidate resumes list. It mines resumes to extract features of candidate profiles such as skills, education, and experience. It used information retrieval methods to rank job seekers for a given job position. For every job profile, the system ranks job seekers based on the resemblance between job profile and job seekers resumes. The ranking can be refined by adding filtering measures. These measures grounded on the job seeker meta-data, as well as on the information that is automatically extracted from the job seekers resumes. This system contains of three main modules: Batch processor, Query processor and Resume matcher. New job seekers are initially processed by the batch processor. It stores the job seekers meta-data in the central database and extracts data from the job seekers resumes, which in turn saved in extracted database. This information is used by the query processor and the resume matcher to deliver the ranking job seeker list for a given user request.

Reciprocal recommendation for recruitment

As authors of [75] proposed a preference approach based on user's interaction history and a new resemblance measurement approaches. The recommendation procedure divided into two parts: job recommendation and job-seeker recommendation. For both parts, the recommendations should be the objects which are the most reliable with their likings. The valuable information is extracted from users' resumes. Then, they find the obvious likings of users and obtain the implicit likings indirectly depending on the situation of sending and receiving resumes. The resemblance of different preferences is calculated using different approaches. Lastly, the comprehensive resemblance is

calculated and the recommendation is produced. The phases of reciprocal recommendation are as follows: (1) users' likings extracted from the content of users' resumes and then, the attribute is determined and changed to vector. (2) The resemblance calculated among users in turn then calculates the reciprocal score. (3) Lastly, the recommendation is made by ranking the reciprocal scores to present the top-n recommendations.

Chapter Three - Related Work

In this Chapter, related researches conducted on personalized job recommendation are reviewed and discussed with their strengths and weaknesses. First, the reviews on the existing research works on the area of personalized job recommendation will be presented. Then, the weakness of those research works are summarized and presented. Finally issues that make our research different from all those researches will be pointed out.

3.1 A Ranker Ensemble for Multi-objective Job Recommendation in an Item Cold Start Setting

On [76] the researchers discuss their findings and solution for the multi-objective job recommendation problem in an item cold start setting. This research was conducted for ACM Recsys 2017 challenge .They show that interaction-based rankers quickly outperform the content-based baseline with respect to the evaluation measure. And obtain the best recommendation candidates using a hybrid of rankers based on reciprocal interactions as well as content features. The proposed top N selection strategies further refine the candidates for obtaining the final recommendations considering the problem constraints and objectives.

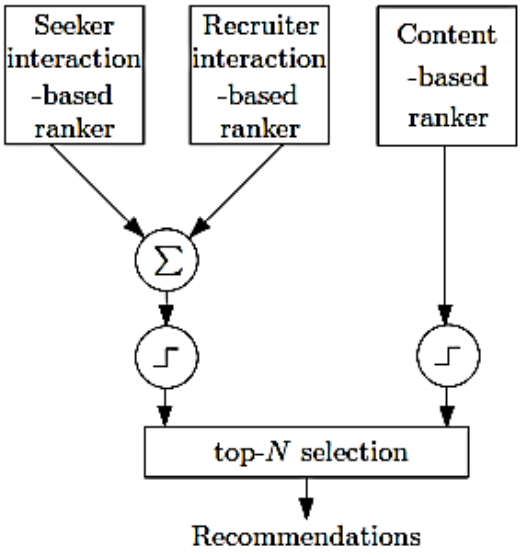


Figure 3.1*: Ranker ensemble for multi-objective job recommendation*

The high level components of the solution are shown in Figure3.1. First, two rankers based on interaction profiles are blended. Then, a content-based ranker similar to the challenge baseline is

trained. Best recommendations from both recommenders are further refined through different top-N selection strategies to produce final recommendations. A recommender solely based on content features is the baseline of this challenge. It is observe that relying on a job seeker's behavior (interaction data) in the system is often more reliable, in case such data exists. The main reason is that the authors believe that the content features based on a resume or form-based data can be misleading due to several reasons like being outdated or incomplete. On the other hand, job ad content is often more elaborate and reliable. Based on this observation, they use a job seeker's positive interaction data to build the user profile. The drawback of this research is that the user is not represented well enough, also negative interactions not represented effectively.

3.2 A Bottom-Up Approach to Job Recommendation System

In this paper, the work on developing a job recommendation system for XING is presented. The work is a part of a competition by ACM RecSys 2016. The authors believe the design of any good system is solely based on how deeply the developers understand the system and the data [77]. Therefore they take a bottom-up approach in which they first deal with each dataset individually and analyze how each of them can help in providing quality recommendations to users. Once equipped with a deeper insight into data, the individual components are combined to build a very powerful recommendation system. They also consider traditional approaches like collaborative filtering (CF). The best model that was produced is based on Gradient Boosting algorithm. The idea is that if the user has interacted positively with a particular job item, then they are more likely to interact positively again. Jobs items with interaction value 3 are on top, followed by 2 and 1. Items with interaction type 4 are ignored, because they represent that the user disliked the item. The drawback of this approach is that the quality depends on the number of clusters that need to be chosen manually. And delete interactions are not represented rather these interactions are ignored assuming that we do not want to show the job items that the user deleted.

3.3 A Scalable, High-performance Algorithm for Hybrid Job Recommendations

This solution is a hybrid algorithm combining a content-based and KNN approach. The content based algorithm matches features of candidate recommendation and job postings of historical interactions. The KNN approach searches for the job postings that are the most similar to the postings the user interacted with in the past. The resulting combination is a lightweight algorithm that is fast and scalable, generating recommendations with a proper evaluation score [45].

The visibility of a job item is estimated by the number of interactions in the dataset. Jobs with a limited visibility get a penalty. Since job items with a low visibility have a low probability of interaction, these job items get a penalty in the algorithm. This penalty favors the popular items, and is not always desirable in a real-world recommender. Also if the combination of the user's explicit profile, interactions, and impressions is still insufficient information for generating recommendations, the system falls back on recommending the most popular items.

3.4 On Recommending Job Openings

The basic model regards (1) two job records of the same company but different job titles and (2) those of the same job title but different companies as four distinct items. As the number of distinct items increases the new item and data sparsity problems get more serious. In order to solve the problems, [78] propose a new modeling that considers a company and a job title as distinct items instead of job records. The two problems can be successfully alleviated by this approach as follows.

- New item problem: Even when a job opening with a new <company, job title> is offered, this is not a new item any longer if both of a company and a job title are found in some (different) job records existing within the past data. This makes the new item problem considerably relieved.
- Data sparsity problem: Since a company and a job title are regarded as separate items, the job records with the same company but different job titles will be associated with the same company, and also those with the same job title but different companies will be associated with the same job title. Because the number of distinct items decreases in this way, the data sparsity problem could be significantly alleviated.

Using this approach, recommendation is performed as follows: Step 1 predicts the preferences of an active job seeker for each company and also for each job title; Step 2 predicts the preference for each job opening by combining the preferences for its company and job title; Step 3 finds the top-N job openings based on combined preferences. There have been a lot of CF algorithms primarily based on user ratings over items. In this domain, however, there are no user ratings but only positive-or-not records over items. But only positive or negative values cannot capture users' behavior and interest properly.

3.5 Field Selection for Job Categorization and Recommendation to Social Network Users

[79] Presents a novel method to formalize the textual content of job offers that aims at identifying the most relevant information and fields expressed by them and leverage this compact formalization for job recommendation and profile matching in social network environments. In this study three classes of documents are considered: job offers, job categories and social network user profiles (as potential job candidates); each class contains several fields with textual information. The proposed representation method permits to dynamically identify those text fields, for each class, that could help a cross-matching strategy in order to preserve, from one hand, the matching/recommendation performances and, on the other hand, reduce the cost of these operations (due to a straightforward dimensionality reduction mechanism). The contributions of this paper are: firstly propose a formal representation of job offers, categories and social user profiles that focuses on field-to-field matching strategy. Then make use of this representation to improve similarity functions for cross-matching purposes.

The drawback of this research is that association between user and job is labeled 1 when they match each other, 0 otherwise. Which we believe is not proper to label the association this way. A user and a job relationship can be between 0 and 1. One cannot completely dislike or like a job.

3.6 Job Recommendation in AskStory: Experiences, Methods, and Evaluation

The authors of this research employ the collaborative filtering approach as their basic framework for job recommendation. They propose data modeling methods (for CF) that alleviate the new item and data sparsity problems. In spite of this alleviation, the data sparsity problem in job recommendation is still serious compared with other domains such as movie recommendation [80]. This makes the authors exploit content information in addition to collaborative information. As a result, the proposed approach is a hybrid one that is mainly based on the CF approach supplemented by the content information. This approach models collaborative and content information together as a graph, and then performs recommendations by analyzing the graph with graph analysis algorithms such as belief propagation (BP and random walk with restart (RWR). Figure 3.2 shows the overview of the proposed recommendation approach. The input is a resume of a target job seeker and the output is a list of top-N job openings to be recommended to her/him.

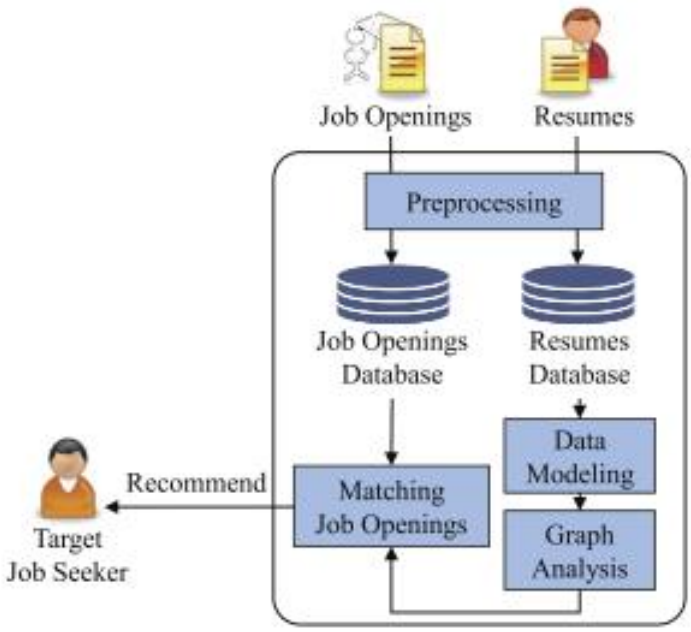


Figure 3.2*: Overview of the proposed approach of Job Recommendation in AskStory*

3.7 Toward the Next Generation of Recruitment Tools: An Online Social Network-based Job Recommender System

[41]This paper presents a content-based recommender system which suggests jobs to LinkedIn users. Experiments recommend that to forecast the users interests for jobs, using basic resemblance measures together with their relations data collected by Work4 can be upgraded upon. The second portion of this study presents a method to estimate the significance of each field of users and jobs in the task of job recommendation. Lastly, the third part is devoted to the use of a machine learning algorithm in order to increase the results gained with resemblance measures: the authors trained a linear SVM (Support Vector Machines). The results indicate that using this supervised learning procedure increases the performance of the content based recommender system. The proposed recommender systems only use users' interactions data and jobs descriptions to predict users' interests for jobs

The personal information posted by users of a social network (which may involve personal description, posts, ratings, but also social links) can be exploited by a recommender system. Due to privacy concerns the proposed recommender system only uses the data that social networks users explicitly granted access to. A vast majority of Facebook users' fields are almost empty. This has raised a big problem: they cannot accurately make recommendations to users whose profiles are almost empty using the proposed recommender system.

3.8 iHR: An Online Recruiting System for Xiamen Talent Service Center

The iHR shown in Figure 2, employ practical techniques in data mining and recommendation to help strengthen the system experience [81].

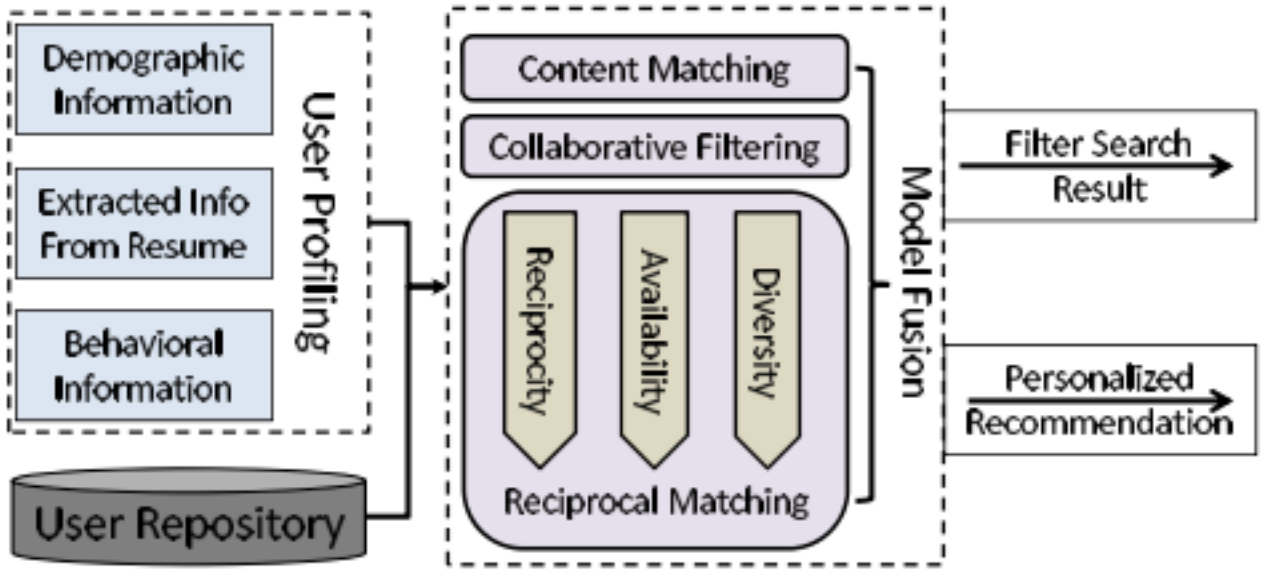


Figure 3.3*: The system overview of iHR.*

To effectively construct the users' profiles, the authors synthesize information from three different channels to maximally discover and understand users' preferences. To satisfy different in formation needs, they integrate search and recommendation into a mutually promoted framework. To enhance the recommendation for more effective information refining, a comprehensive strategy to simultaneously consider the appetite of job seekers and recruiters is proposed. To construct user profiles for jobseekers and recruiters, iHR considers multiple information resources, including users' basic information, extracted text from uploaded files or links and user's behavioral information. The fusion of different types of information enables one to comprehensively understand a user's exact interest.

3.9 Summary of Relate Works

This chapter reviewed a number of related works as summarized in Table 3.1.

Table 3.1: *Summary of Related Works*

No.	Research	User Profile	Approach	Drawback
1	A Ranker Ensemble for Multi-objective Job Recommendation in an Item Cold Start Setting	Individual information and historical behaviors	• CBF • CF	• User representation. • Insufficient user profile information (job seeker is not represented well) • Negative interactions not represented effectively.
2	A Bottom-Up Approach to Job Recommendation System	Individual information and historical behaviors	• CF • K-Means clustering on users and items	• The quality depends on the number of clusters that need to be chosen manually. • Delete interactions are not represented.
3	A Scalable, High-performance Algorithm for Hybrid Job Recommendations	Individual information and historical behaviors	• CBF • KNN	• Jobs with a limited visibility get a penalty, this penalty favors the popular items. • The system falls back on recommending the most popular items. If the combination of the user's explicit profile, interactions, and impressions is insufficient information.
4	On Recommending Job Openings	Individual information (resume) and	• One-class collaborative filtering (OCCF)	• Positive and negative (binary) values cannot capture users' behavior and interest properly.

		historical behaviors		
5	Field Selection for Job Categorization and Recommendation to Social Network Users	Social network user profile	• CBF • Focuses on the field to field similarity, computed from cosine similarity	• Insufficient user profile (from social network user profiles) • Association of user and job is labeled by 0 or 1, which can't represent their correspondence/association well.
6	Job Recommendation in AskStory: Experiences, Methods, and Evaluation	Resume Individual information	• Hybrid • Use a graph-based CF approach as a main framework and supplement it by content information for further refinement. • Collaborative and content information is modeled together as a graph and is analyzed by graph analysis.	• Insufficient user profile – resume(education and experience) is the main resource for analysis
7	Toward the Next Generation of Recruitment Tools: An Online Social Network-based Job Recommender System	Social network user profile	• CBF • Use basic similarity measures together with their interactions data • Used Basic heuristic-based method • Supervised learning: SVM	• The personal information posted by users of a social network cannot be enough for job recommendation. • Fields from a social network cannot describe/ represent a user efficiently. • All fields are assumed to have the same importance on recommendation scores, not all the users or jobs fields have the same importance on

				the recommendation scores: some are more important than others • Facebook fields are almost empty which leads to improper user representation
8	iHR: An Online Recruiting System for Xiamen Talent Service Center	Individual, extracted and behavior	• CBF • CF • Reciprocal Recommendation	• Insufficient user profile information (job seeker is not represented well) • Insufficient user behavior and preference information.

Recommender systems rely on user profiles in some shape or form. Although the above researches utilize the individual information, their origins are not the same. Most of them capture the profile from resume, some rely on social network profiles while others rely on the behavior of user towards the systems. We believe that these researches conducted on personalized job recommendation lack on capturing the interest/ preference of the user. And using only individual information and historical behaviors (ratings) or social network user profile is not enough to represent a job seeker and to produce recommendations based on those information.

All of the above related works on job recommendation lack capturing users individual preferences regarding location, job category, sector (industry field of the hiring company, and dates (expire dates) the job) in the job recommendation domain. In our work we enhance user representation by adding this preferences of the job seeker.

Our work is fundamentally different from other related works in two main ways. Firstly some of the information is given a value as a preference level by recording their level of interest in specific information items they provide into the system. Thus generating user profiles that are more detailed in a way of expressing the users' interest/ preference better which leads to a better recommendation. This is done in addition to the traditional rating method used in the previous related works. Secondly, unlike the previous researches our system takes contextual information such as location, job category and the sector which the hiring company is in into consideration with their preference value/ weight.

And make use of those contextual information in combination with the user profile. By capturing most of the user behavior and interest we believe we can understand the users need well and make relevant recommendations (jobs to the job seeker and resumes for the recruiter).

Chapter Four – Proposed Solution

In this Chapter the proposed architecture a context-aware personalized job recommender will be described. The main design goals considered in designing a context-aware personalized job recommender as well as the different components along with techniques for realizing them are presented in detail. The high-level components of the proposed solution are presented in Figure 4.1.

4.1 Overview

Personalized Recommendation offers a way of providing services based on one's personal interest or preference. Personalized Recommendation are based on user behavior, it filters information and can learn users interest according to their profile and historical behavior and predict their interest or preference for a given item.

A context-aware personalized recommender offers a way of providing services that enable users to access and use information that might be of interested on. It captures the behavior of a user in order to be able to generate the recommendation that suits. Context awareness plays a vital for acquiring the goal of personalized recommendation by identifying any relevant information on individuals' personal information and their personal interests.

Job management, job seeker and job recommender can be seen as the overall components in the proposed architecture. It collects raw context data from the job seeker, converts context information, and provides relevant service according to the given context. Producing context information not only helps systems to simply provide their services to the intended user, but also enables them to give intelligent decision when necessary before delivering those services.

An overview of the proposed architecture is shown in Figure. 4.1. The Context-Aware Personalized Job Recommender is consist of components such as job management, job seekers, job recommendation. The Job Management component is the one that is responsible for uploading new jobs and edit existing one and searching for candidate job seeker that fits to the requirements of the job using the job- person recommender. The Job seekers sub-system manages job seeker profile and contexts, and handles the request of job seeker. The job recommender has two sub- systems: job to person recommender and person to job. The job-person recommender is responsible to make use of

the profile/context in the repositories and the two recommendation approaches to generate top 10 candidate job seekers based on those information. These top candidates are recommended based on the similarity score (which holds a numerical value that represents how a job is similar to each job seeker) that is generated from the two recommendation strategies. The person-job recommender is responsible to make use of the profile/context in the repositories and the three recommendation approaches then to generate top 10 candidate jobs. The recommendations are generated by using the similarity score (holds a numerical value that represents how a job seeker is similar to each job). We believe presenting/ displaying the top 10 job candidates/job seeker candidates is enough. Recommendations more than that won't be relevant recommendation because of lower similarity score. All of these tasks are discussed in detail in the following sections.

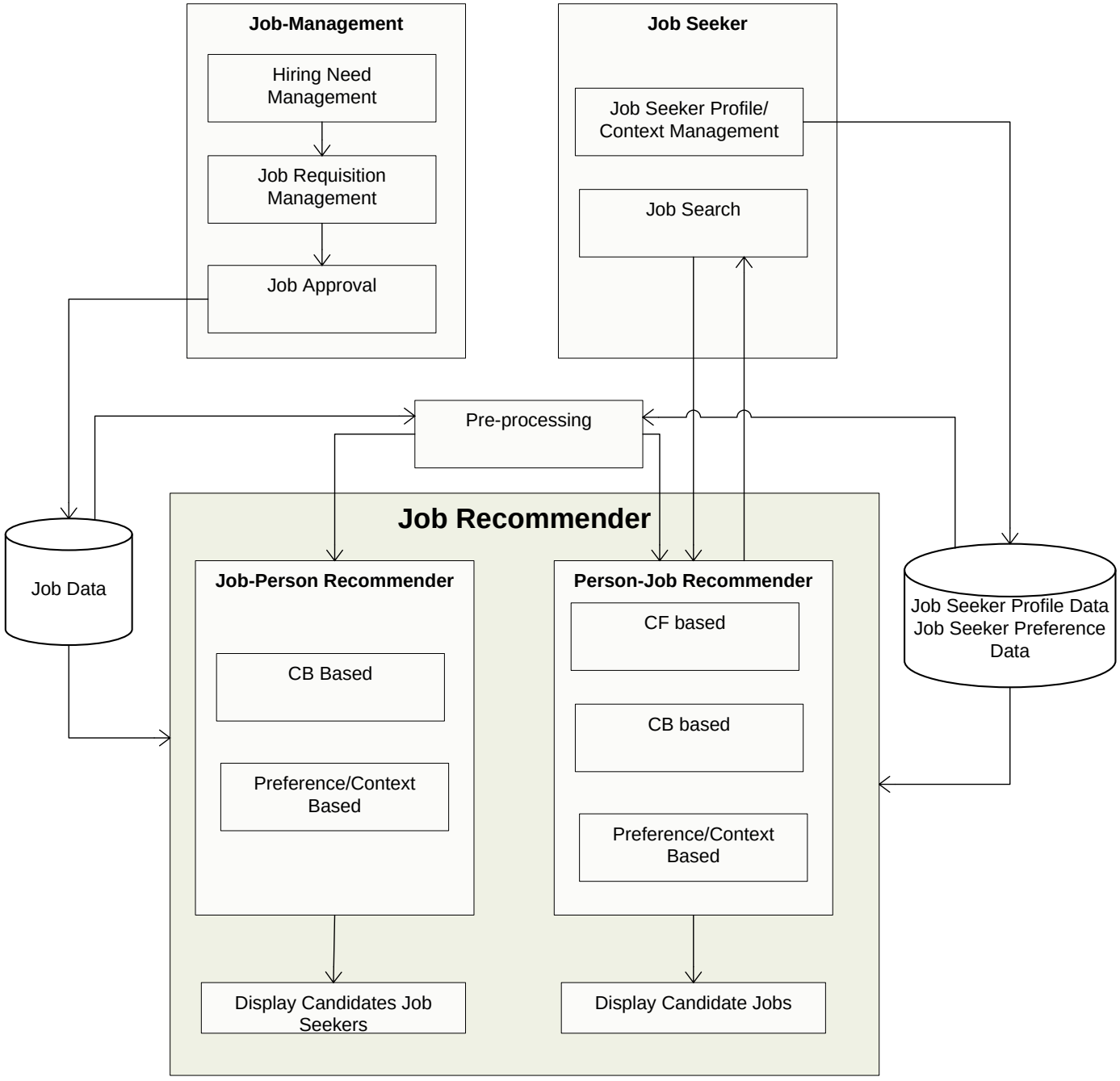


Figure 4.1*: Context Aware Personalized Job Recommender Architecture*

4.2 User Profile and Context Modeling

In this section, we propose a new customer profile model reflecting job seeker individual behavior information, then we submit a job recommendation technique using the proposed job seeker profiles and job profiles.

The recommendation process starts with the job seeker, who both provides the resume and performs actions, like ranking jobs and providing contexts. These actions again influence the job seekers profile and lead to an adaption of the recommendations. The data and action is then used by the recommender to find the best matching candidate and job. A better recommendation system needs an expressive representation, which is able to capture users' interests and preferences.

For providing personalized recommendations, job seekers profiles and context are to be used. For creating such a profile, job seekers provide the resume, containing various demographic data and other relevant information such as educational background and work experience. These job seeker information are represented in relational scheme data model. Job-seeker, education, experience, skill, job category preference, location preference, sector preference and date preference are relational tables in the repository representing job seeker. The resume helps to generate recommendations from the very beginning on and enables the recommender to suggest jobs according to the capabilities of the job seeker. Preference/ context is also used as input for our recommender. The job seeker provides the system which jobs and job attributes are interesting and which ones are not by weighing (e.g. For location preference the job seeker provides their preference by weighting them from one to five).

The profile and context modeling module is responsible for taking the preprocessed job seeker and job data and generate job seeker profile, job profile, context information and company profile. Then this module stores each of these profiles in the repositories.

In the job seekers context representation each job seeker is represented by the value/ importance of location, and other properties he or she has. Not only data from the resume has been included in the profile, preferences of the job seeker towards each job attribute is included here as context. Job seekers profile fields contain values, associated with a weight, which represents the job seeker’s degree of interest.

Job Seeker Profile

User profiles can be created through consumer information explicitly specified when they initially register or subscribe to service. We use same profiling method. Each user can be defined by a set of following general attributes:

- Single-valued attribute: first name, last name, sex, date of birth, age and other attribute has only one value.

- Multiple-valued attribute: experience, education, skill the attribute has two or more values.

On this thesis to capture an individual job seeker's interest/ preference we propose a new user profile through extending the job seeker profile by accumulating the preferences of the job seeker in four aspects (location, job category, industry(sector) and dates) to the profile. This model consists of the following steps.

- To capture the job seeker's preference explicitly from the user.
- Construct customer profiles using the job seeker personal information and preferences/ interests.

The representation of the job seeker profile is the first task to do. The user profile contains information that represent job seekers' basic information (name, age, email address, address), education, series of job records representing her/his career work experience and skills. The context of job seeker represents the preference of the job seeker in terms of their interest on job category, location, sector and dates. In our solution job seekers are represented by a set of attributes that can be considered as a structured summary of the curriculum vitae. And characterized by a set of preferences they have towards job attributes. After interaction with the system we add a job seekers ratings data to build a history of the job seekers interest. More specifically, every rated job and job contributes to the profile. This job seeker history assist the person job recommender to produce more personalized predictions by finding other job seekers with similar interest, and recommend jobs that other job seeker is interested in.

Job Profile

A job profile is a description of job features. We recommend jobs by computing the similarities between the job and job seeker profiles (person-job recommender) and recommend candidates by computing the similarities between the job and job seeker profiles (job person recommender) defined in Section 4.2.5. This profile contains information about the jobs. The fields provided are: job ID, job category, title, sector, type of employment, location, company, required skills, posting date, ending date etc. These attributes are to be provided by the recruiters.

Preference/ Context Representation

As [46] the context representation depends on the utilization of context. If the context is used to express the state of an entity, it may be sufficient to express the context is a class diagram. But if the context is used to characterize context dependent information delivery and following user interaction,

the context shall be expressed in the proper diagrams. UML diagrams are be used to represent the context in the domain model as shown in Figure 4.2.

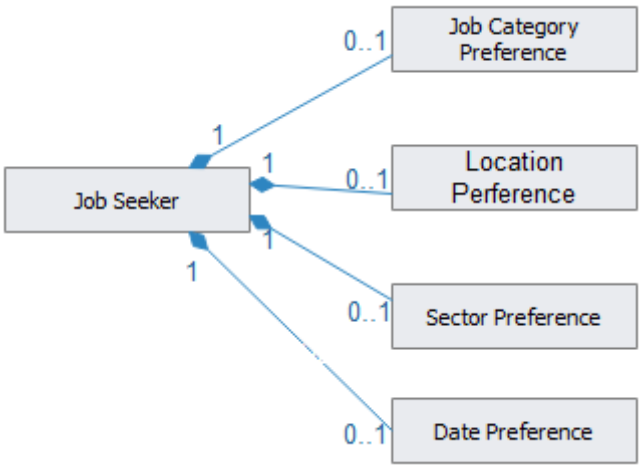


Figure 4.2*: Conceptual class diagram for the preferences*

Job category preference – three job categories (and their weight) which interests the job seeker.
Location preference – these preference is about where does the job seeker wants to find a job.
Sector preference – three industry fields that the job seeker wishes the hiring company is in.
Date preference – at which three preferred dates the job seeker is looking (to apply) for a job.

As stated in chapter two context can be defined as the elements of the user's environment which the computer knows about and as ―location, identity, environment, and time. Context information can be categorized as static context or dynamic context. Static context is information about the user and changes rarely. We have identified 4 contexts for the job seeker. These are location preference, date preference, sector preference, and job category preference. The location, date, sector type, and job category preferences are used to capture and understand the behavior and needs of job seeker. These context contains the value of weighted preference of the job seeker regarding of the desired jobs location, sector, date and company. For example for location context, a job seeker evaluates his/her preferred locations to find a job in. All the five contexts are weighted by the job seeker. The location context/ preference is represented as shown in Table 4.1. Preference is represented by weighting the desired location name one to three. The same applies for the other four preferences. These static contexts are stored in a SQL relational database management system. Whenever the user wants to change one or more of his/her profile information (for example address) or preference the database will be updated to have that change.

Table 4.1*: Location Preference Sample*

Job Seeker ID	Weight	Location Name
0001	1	Addis Ababa
	2	Bahir Dar
	3	Hawassa
. . .		
0087		
0088	1	Gondar
	2	Addis Ababa
	3	Adama
. . .		

Company Profile

This profile contains the information of companies (Name of the company, the classifications of the industry (sector) that company is in and location)

4.3 Major Components of the Architecture

4.3.1 Job Seeker

This component is responsible to handle the request of job seeker. It has two sub-components profile/context management and job search. Profile management deals with profile information: upload new profile and edit existing one. Job search is responsible searching for job that fits to the profile of the person using the person-job recommender. The management of job profile demands preprocessing resume using the preprocessing sub-component.

4.3.2 Job Seekers Profile/ Context Management

This component is responsible to manage the profile and context of a job seeker. In this work, Job seeker provide resume having personal information, work experience and educational experience.

Also the job seekers provide their interests regarding location, job category, sector, and dates of their interest.

b) Personal information section contains basic profile data such as the first name, last name, contact address and location.
c) Work experience section contains Information about the current and past professional positions held by the job seeker such as company names, positions, company descriptions, job start dates, and job finish dates. The company description field may additionally contain information about the company (sector).
d) Information about educational experiences, such as university names, degrees, fields of study, start and finish dates
e) Context/ preference section contains list of job-categories the job-seeker interested to apply for. Also it contains a list of locations, job category, sector and dates which are preferred by the job seeker. This information is collected explicitly from the job seeker.

In addition, each job-seeker provides her context or preference information. In this thesis, context is dedicated to job-categories, locations, sectors and dates in which the seekers want to apply for (interested in). Detail of the context and profile modelling is presented in the previous section (4.2)

4.3.3 Preprocessing

The user (both job seeker and the recruiter) provides data in structured and unstructured format to the system. These data need to be preprocessed to make it ready for further manipulation by removing unnecessary words and changing characters and words into their common form. The preprocessing module is responsible for this task. Preprocessing is one of the major steps when we are dealing with any kind of text models. This module is responsible for cleaning the resume and data that is provided by the user. And passes the cleaned data to the job recommender. This involves:-

- Lowercase - During the text processing each sentence is split to words and each word is considered as a token after preprocessing.

- Stop words removal - Stop words are low information bearing words such as “is” or “the”, typically appearing with high frequency. Stop words may be context dependent. High frequency words have higher variance and effective weight in many methods, causing them to be erroneously selected as features due to sample noise. Stop words are the most commonly

occurring words which don't give any additional value to the document vector. In-fact removing these will increase computation and space efficiency.

- Stemming - This is the final and most important part of the preprocessing. Stemming converts words to its stem. Stemmer reduces the word to its stem. We are going to use a library called porter-stemmer which is a rule based stemmer. Porter-Stemmer identifies and removes the suffix or affix of a word.

4.3.4 Repositories

Data Modeling

All information that is provided by the users' needs to be modeled and stored in the repositories. These information from job seekers is passed to the job seeker profile/ context module to be modeled and stored in the repository. The job recommender (both job-person and person job recommender) uses these stored data/ profiles to produce recommendations. All the profiles generated in the profile modeling are represented in Relational Scheme. Whenever any of the content of the profile is required to change, one or more of the information (for example experience) the database will be updated to have that change. The data model is represented using chicken feet notation representing parent and child relationship as shown in Figure 4.3.

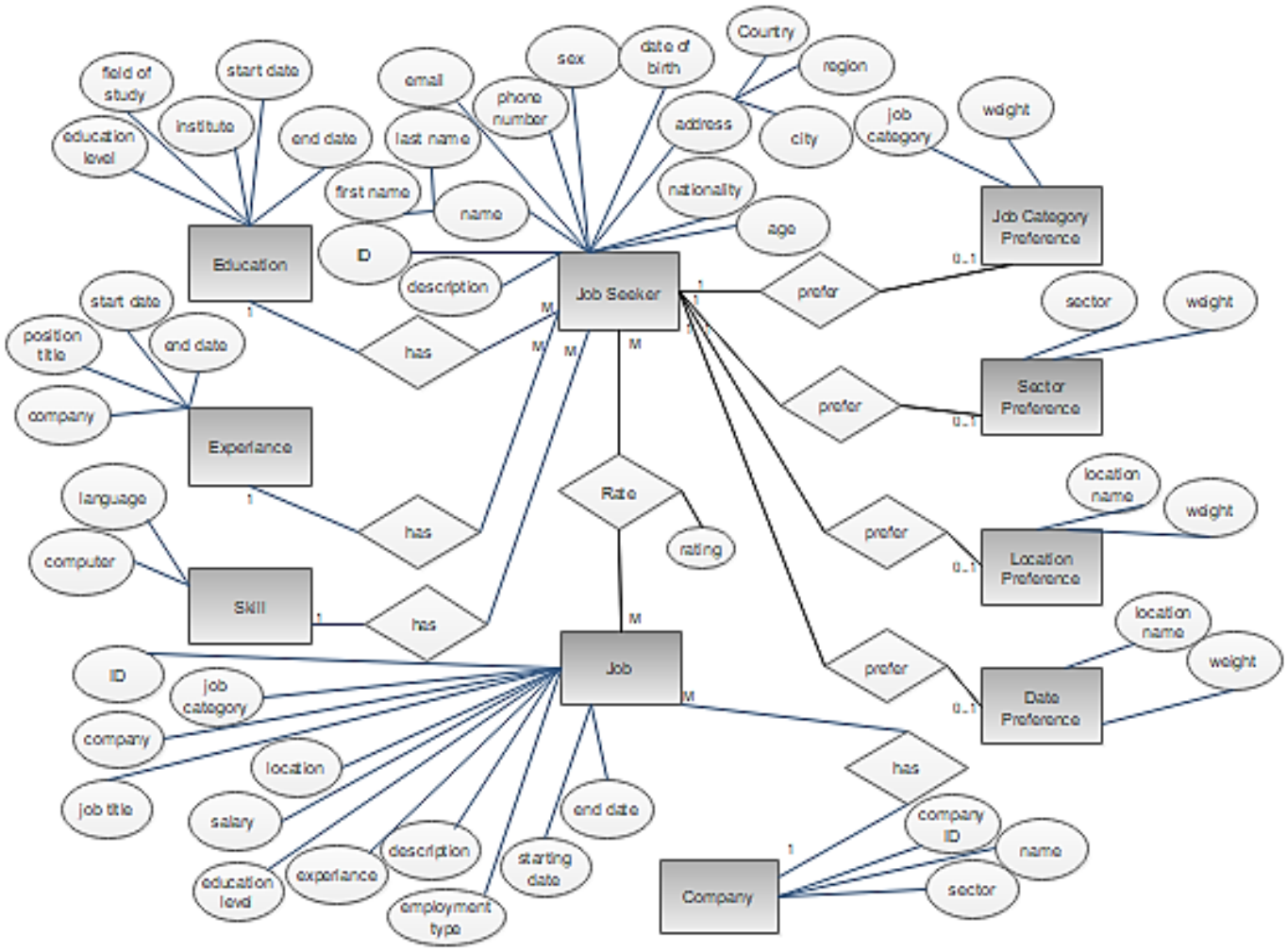


Figure 4.3*: Entity Relationship Model Diagram for CAPJR*

4.3.5 Job Recommender

The job recommendation is responsible for making use of the profiles and preferences of the job seeker and job postings of companies then generate recommendations that fits to a person (case of person-job recommendation) or list of persons that fits to a specific job (by job-person recommendation). All of these tasks are discussed in detail in the following sections.

A. Person Job Recommender

The person-job recommender component of the system is responsible make the appropriate predictions based on those information. It uses the stored information and performs matching between job seekers and jobs then produce list of jobs that fit the job seekers' interest. There are three recommendation models inside this module, content based filtering, collaborative filtering and

context based filtering to compute the best matches for a job seeker. We uses these three recommendation approaches with all having their own drawbacks and strengths, by using them we can tackle the drawback on one approach with another. Personalized job recommendation is presented in three modes:

1. The first one is based on the personal information provided from the job seeker – achieved by content based approach.
2. Based on the four job seeker preferences - achieved by context based approach.
3. And based on the ratings of the jobs by user (this is through when the user has interacted with the system by rating jobs) - achieved by collaborative filtering approach.
4. And the last one is based on all available parameters (personal information, preferences and ratings) - achieved by merging the above three (in the case of absence of user interaction i.e. rating, achieved by merging the first two).

With the above approaches the job we have the option to select from these approaches which is suitable for the available job seeker and job data. If the user have only preference information available, job is recommended based on the available preference information. If the user have only resume available also recommendation is based on the resume (job seeker profile). And if the job seeker have ratings available recommendations are made based on that interaction. A job seeker can have all information (profile, preference and rating) available all recommendation strategies/ modes are used to produce recommendation. With all the approaches the job seeker finds 10 relevant jobs that best suits their interest.

1. Content based recommendation

The base of the content-based approach is a user profile with the clearly specified user information and job description. This module is responsible for producing predictions based on only the contents of the job seeker profile and job profile. In this work we use an approach that capture similar jobs and job seekers in content. The TF-IDF algorithm is used to weigh a keyword in any profile and assign the importance (how much is a word significant to a document) to that keyword based on the number of times it appears in the document. The higher the TF-IDF score (weight), the rarer and more important the term, and vice versa. Algorithm 4.1 presents the implementation detail of this module.

We use the education, experience and skill on job seekers, job description on job as our content information. Content-based features are extracted from the descriptions by using the term

frequency/inverse document frequency (TF-IDF). We compute the similarity between job seeker and jobs by the cosine similarity between their TF-IDF vectors.

// ***Content Based Person-Job Recommendation***

READ Job seeker profile, Jobs profile

Preprocess

BEGIN

a. Calculate tf - Weigh (count number of occurrence) of each word in any profile job and job seeker
b. Calculate idf - Assign the importance to that word based on the number of times it appears in the profile (the more a word appears the more important that word is).
c. Measure how important that term is in all profiles (job seeker and job). More times a word appears through all profiles, the more poorly discriminates between profiles.
d. Tf and idf are combined into weights - multiply tf and idf
e. Compute the similarity(cosine similarity) of a job seeker profile with every job profile in the repositories
f. Calculate cosine of the angle between vectors to measure the similarity between job seeker profile and every job profile in the repositories.
g. Store the similarity score

END

Algorithm 4.1*:* *Content Based person – job recommendation*

2. Context based recommendation

This sub module is responsible to make recommendations based on the users explicitly defined preferences. This contextual recommendation algorithms recommend jobs that match the user's current preference/ context to the job seeker. This allows them to be more flexible and adaptive to current job seekers' needs. The algorithm 4.2 find jobs that match the job seekers location, job category, sector and dates with the job profile information such as location, job category, sector and job expire date. If these job seeker preferences match with those information from the job profile a job is recommended for the job seeker.

// ***Context/ preference based Person-Job Recommendation***

READ Job seekers profile, Job profile

BEGIN

a. Extract jobs location, job category, sector and expire date

   i. IF jobs' location is found in job seeker preference
      1. Increment the similarity score by the weight of that preferred location

   ii. IF jobs' job category is found in job seeker preference
      1. Increment the similarity score by the weight of that preferred location

   iii. IF jobs' sector is found in job seeker preference
      1. Increment the similarity score by the weight of that preferred sector

   iv. IF jobs' expire date is found in job seeker preference
      1. Increment the similarity score by the weight of that preferred date

b. Store similarity score for all jobs

END

Algorithm 4.2*: Context Based person – job Recommendation*

3. Collaborative filtering based recommendation

This module is responsible to make job recommendations for the job seeker based on the ratings. It searches for the jobs that are the most similar to the job the job seeker rated. This approach does not make any assumptions on the underlying data distribution but it relies on job similarity. When we makes inference about a job, we will calculate the distance (similarity) between the target job and every other job in our repository, then rank using distances and return the top 10 nearest jobs as the most similar jobs recommendations. In this algorithm, the similarities between different jobs in are calculated by using cosine similarity measures, and then these similarity values are used to find nearest neighbors of a job. The perception behind this approach is that a job seeker would be interested

in jobs that are similar to the jobs the job seeker liked prior and would incline to avoid jobs that are alike to the jobs the job seeker didn't like earlier.

The first phase in this collaborative filtering algorithm is to compute the similarity between jobs and then to select the most similar job. The basic idea in similarity computation between two jobs is to first isolate the job seekers who have rated both of the jobs and then to apply a similarity computation technique to determine the similarity the two jobs. To compute the similarity between jobs we used cosine-based similarity. In this case, two jobs are thought of as two vectors. The similarity between them is measured by computing the cosine of the angle between these two vectors. After calculating the similarity between jobs, then the algorithm selects a number of jobs with the highest similarity as neighbors. Set a value K = 10 for the neighbor selection, select the most 10 high similarity as neighbors.

Algorithm 4.3 shows the details of collaborative person-job recommender. The goal of this algorithm is to suggest new jobs for a particular user based on the users' ratings. The algorithm returns a list of 10 jobs that a job seeker will be interested the most.

// ***Collaborative Filtering Based Person-Job Recommendation***

READ Job seeker ratings on jobs, Jobs

BEGIN

a. Find/isolate all the job seekers who have rated jobs
b. Compute similarity between these jobs - cosine based similarity
c. Select the set of most similar jobs based on the cosine similarity measure
d. Look into the target job seekers ratings
e. Compute prediction on job j for a job seeker s
    1. compute the sum of the ratings given by the job seeker on the jobs similar to j
f. Store the rating predictions for the jobs

END

Algorithm 4.3*: Collaborative Filtering person – job Recommendation*

Display Candidate Jobs

This module is responsible to display the recommendation of jobs to the job seeker based on the similarity score obtained from the above three modules. All the above three recommenders (content based, context based and collaborative filtering based recommenders) return similarity score which is a measure of how much jobs match a job seeker. The similarity score is a list of (job id, score). So this module accept the score and recommend those jobs with the top 10 score and present them for the job seeker.

B. Job Person Recommender

The job-person recommender is the one responsible for generating candidate job seeker recommendation for a job opening. It uses the profiles and performs matching between job and job seekers then produce list of jobseekers that fit the job requirements. There are two recommendation models inside this module, content based filtering and context/ preference based filtering to compute the best candidates for a job. Personalized job recommendation is presented in three modes:

1. The first one is based on the job profile and job seekers profile – achieved by content based approach.
2. Based on the four job seeker preferences - achieved by context/ preference based approach.

1. Content Based Recommendation

This module is responsible for producing predictions based on only the contents of the job profile and jobseekers profile. It works in the same manner except that it is the other way around. It recommends the most similar job seekers for a job based on the matching score of those profiles. Algorithm 4.4 shows the implementation of this module.

// ***Content Based Job-Person Recommendation***
READ Job profile, Job seekers profile
Preprocess
BEGIN

a. Calculate tf - Weigh (count number of occurrence) of each word in any profile job and job seeker
b. Calculate idf - Assign the importance to that word based on the number of times it appears in the profile (the more a word appears the more important that word is).

c. Measure how important that term is in all profiles (job seeker and job). More times a word appears through all profiles, the more poorly discriminates between profiles.
d. Tf and idf are combined into weights - multiply tf and idf
e. Compute the similarity(cosine similarity) of a job profile with every job seekers profile in the repositories
f. Calculate cosine of the angle between vectors to measure the similarity between job profile and every job seekers profile in the repositories.
g. Store the similarity score

END

Algorithm 4.4*: Content Based job-person recommendation*

2. Context based recommendation

This sub module make job seeker recommendation for a job based on the preference of the job seeker. It finds job seekers that are more likely to be interested on a job. The algorithm 4.5 find job seekers with preferences that match the job profile.

// ***Context/ preference based Job-Person Recommendation***

READ

Job profile

Job seekers profile

BEGIN

a. Extract job seekers preferences

i. IF jobs' location is found in job seeker preference
   2. Increment the similarity score by the weight of that preferred location

ii. IF jobs' job category is found in job seeker preference
   1. Increment the similarity score by the weight of that preferred location

iii. IF jobs' sector is found in job seeker preference
   2. Increment the similarity score by the weight of that preferred sector

iv. IF jobs' expire date is found in job seeker preference

2. Increment the similarity score by the weight of that preferred date

c. Store score for all jobs

END

Output

Similarity score

Algorithm 4.5*:* *Context Based job-person Recommendation*

The context aware personalized recommender accepts the generated top 10 recommendations from all three modules and merges these results and compute the average for each recommendation using the candidate generator. Based on the computed average this module selects the top 10 job recommendations for a job seeker also selects the top 10 candidates for a job.

Display Candidate Job seekers

Both the above job-person recommenders (content based and preference/ context based) return similarity score (how much a job is similar to each job seekers). So this module accept the score and recommend those jobseekers with the top 10 score. The display candidate job seekers is responsible to present/ display the recommended job seekers to the job/ recruiter based on the similarity score that is obtained from the above two modules(CB based and Preference/ Context Based recommenders).

Chapter Five – Implementation and Evaluation

In chapter four, we describe in detail the design of the proposed CAPJR system. Following the concepts described on the design of the architecture for CAPJR, we also develop a prototype. In this chapter we will discuss the implementation of the proposed architecture by presenting the prototype, tools utilized in developing the prototype and the implementation detail and the evaluation result.

5.1 Overview

The prototype developed in this study is purposed on checking the functionality of the proposed context aware personalized job recommender (CAPJR) architecture which is discussed in chapter four. Majority of the components in each modules of the architecture are implemented as per the theoretical specification and functionalities described during the design of the proposed system. The prototype implemented utilizes job seeker profile and preference, and jobs profile to provide personalized recommendation.

5.2 Tools and Technology used for Implementation

During the development of the prototype, a number of tools are utilized which are related to operation environment, programming, communication, database management, context/preference modeling and representation. Lists of tools used are described below:

- **Python programming language -** Python programming language has been used to program the prototype implementation. . Python is a programming language that lets you work quickly and integrate systems more effectively. Python packages such as Numpy, Scipy, Sklearn and Matplotlib can be installed in a program running on Python. These packages cater to machine learning and help developers detect patterns in big sets of data.
- **Pycharm Integrated Development Environment (IDE) -** Pycharm is an open source Integrated Development Environment (IDE). It is a hybrid-platform developed by JetBrains as an IDE for Python. PyCharm is a dedicated IDE for Python. It incorporates some state of the art features and is *currently* available in two editions: professional (paid) and community (unpaid). It supports two python versions: v2.x and v3.x. The Community Edition is used for developing the prototype with Python.
- **MySQL Relational Database Management System -** MySQL is an Oracle-backed open source relational database management system (). The database structures are organized into physical files. The logical model, with objects such as databases, tables, rows and columns, offers a flexible

programming environment. MySQL is offered under two different editions: the MySQL Community Server and the proprietary. For our work we use the open source edition.

- **Ubuntu 16 OS, Intel(R) Core(TM) i5 CPU @ 2.4GHz, 4GB RAM and 500GB HD -** is used as an operating and development environment Ubuntu is an open source computer operating system (OS). Ubuntu is the best Linux distro for developers for many reasons. The first reason relates to the support for different emerging technologies such as deep learning, artificial intelligence, and machine learning.
- **HTML** computer language: Is used to implement sample user interfaces for the prototype.
- **Flask:** Flask is a lightweight web frame of Python. It provides the user with libraries, modules and tools to help build Web-Applications such as a blog or wiki.

5.3 Implementation detail

In this sub section, we will describe the detail implementation of each modules. The modules are implemented with the python programming language to take the language‘s advantages of machine independency and its ease in interacting with other open source tools. All modules are implemented in Ubuntu 16 environment. The prototype contains two components which run both in the client side and server side. Client side is implemented with HTML, to accept user information. The communication between the server and client is handled by a standard hypertext protocol (HTTP). Table 5.1 illustrates the implementation detail for the modules.

Table 5.1*:* *Module implementation detail*

Component name	**Implementation detail**
Job seeker profile/ context management	• Provide an interface where the user input resume and personal preferences • Used to accept user information • Implemented using flask and python • Annex B shows the sample user interface
Job Search	• Implemented by python method • Used to search jobs based on the provided parameters

Profile data	• Implemented in MySQL server database and use to store the registered job seekers and jobs information
CF based person-job recommender	• Implemented using KNN in by python sklearn • Uses to recommend jobs to job seeker
CB based person-job recommender	• Implemented by python method • Uses to recommend jobs to a job seeker based on their ratings • Implemented using tf-idf vectorizer • Annex A illustrates the python code utilized for the purpose of implementation
Context based person-job recommender	• Implemented by python method • Uses to recommend jobs to job seeker based on the profile contents
Generate candidate jobs	• Implemented by python method • Used to present the recommendation result to user
CB based job-person recommender	• Implemented by python method • Uses to recommend jobseekers to a job based on their job seekers ratings • Implemented using tf-idf vectorizer • Annex A illustrates the python code utilized for the purpose of implementation
Context based job-person recommender	• Implemented by python method • Uses to recommend jobs to job seeker based on the profile contents
Display candidate job seekers	• Implemented by python method

	• Used to display/present the (recommended) candidate job seekers to the recruiter based on the similarity score
Display candidate jobs	• Implemented by python method • Used to display/present the (recommended) candidate jobs recommendation to the job seeker based on the similarity score

Jobs Based on Your Preference

Job Category	Company	Job Title
Information Technology, Telecommunications	Amref Health Africa	Help Desk Officer
Information Technology	ELNET Technology P.L.C	ull Stack Developer
Information Technology	CTO	None
Information Technology	Flora IT Solution PLC	Data Encoder
Information Technology	Nared General Trading Plc	IT expert
nformation Technology, Social Sciences	UNHCR	Registration Assistant
Information Technology	UNHCR	PRIME Business Analyst
Information Technology	UNHCR	Senior Data Management Associate
Information Technology	Dan Church Aid	IT and Database Assistant
Information Technology	Dashen Bank S.C	Senior System Support Officer

Jobs Based on Your Profile

Job Category	Company	Job Title
Information Technology	CTO	None
Information Technology, Telecommunications	Amref Health Africa	Help Desk Officer
Information Technology	UNHCR	PRIME Business Analyst
Information Technology	Dan Church Aid	IT and Database Assistant
Information Technology	Nared General Trading Plc	IT expert
Information Technology	UNHCR	Senior Data Management Associate
Information Technology	ELNET Technology P.L.C	ull Stack Developer
Information Technology	Dashen Bank S.C	Senior System Support Officer
Information Technology	Flora IT Solution PLC	Data Encoder
Information Technology	Nared General Trading Plc	IT expert

a b

Figure 5.1*: sample result of person-job recommender: based on preference (a) and base on content (b)*

5.4 Experiment and Evaluation

As it is mentioned in the beginning of this chapter, a prototype has been developed as a proof of the concepts described in in the architecture of CAPJR. Therefore, in this chapter the test and experiments conducted to validate the proposed context-aware personalized job recommender is presented. In

order to conduct the experiment end evaluate the prototype, 50 jobs and 50 resumes are used. In order to test the system we have followed some form of procedures.

The testing is done manually by checking the compatibility of the job seeker profile and job profile. We use precision to evaluate this prototype. First, for each job and job seeker profile evaluation has been conducted by measuring the precision of the prototype. Finally the result is merged to get the final result.

To evaluate the result of the recommenders we test the prototype by checking the resulting job recommendations for each CV. Collaborative filtering based approach is not evaluated for this thesis due to the lack of user interaction which is rating in our case. The test produces the following result for person-job recommender.

Table 5.2*: Job recommender evaluation result*

Recommendation models	Precision
Content Based	0.64
Context/ Preference Based	0.86
Combined usage	0.76

The evaluation result shows there is an improvement in personalized job recommender by applying preferences in the recommendation process.

Chapter Six – Conclusion and Future Works

6.1 Conclusions

In this thesis, we have proposed and developed a model for a context-aware personalized job recommender. We enhance user profiling method by applying context information with the user profile. Context information has first to be collected from the job seeker directly. Once context information is collected the next step is context modeling. Context modeling deals with a way to store, process and reason over it to make the service context-aware. A suitable context modeling approach has been studied and identified and context modeling has been done. Appropriate tools and techniques have been identified to develop a prototype for the thesis. The prototype has been tested under different with a precision and the results are presented.

The following are the main contributions of this thesis:

- Provide an architecture for context aware personalized job recommender with capabilities of preference representation.
- Enhance user profiling by adding preferences as context.
- Produce better personalized recommendations by integrating preference.

6.2 Future Works

The following are some of the potential future works to the continuation of this work.

- By integrating social network profiles enhance user profiling for personalized job recommendation process,
- Experiment the capability of this work by adding the user interaction history.

References

[1] X. Zhou, Y. Xu, Y. Li, A. Josang and C. Cox, "The State-of-the-Art in Personalized Recommender Systems for Social Networking," *Artificial Intelligence Review,* pp. 119-132.

[2] N. D. K. M. A. F. C. E. Keith Cheverst, "Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences".

[3] E. D. S. N. L. D. G. K. Verbert, "Context -aware Recommender Systems," *Journal of Universal Computer Science,* vol. 16, pp. 2175-2178, 2010.

[4] B. K. h. a. S. Barry, "Personalized information ordering: a case study in online recruit ment," *Knowledge Based Systems,* vol. 15, pp. 269-275, 2003.

[5] T. M. K. Xiaoyuan Su, "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence,* pp. 421-425, 2009.

[6] H. J. F. S. Wei K., "A Survey of E-Commerce Recommender Systems," in *International Conference on Service Systems and Service Management*.

[7] L. K, "Newsweeder: Learning to filter news," in *12th International Conference on Machine Learning*, Lake Tahoe, 1995.

[8] R. Burke, "Hybrid Recommender Systems:Survey and Experiments," *User Modeling and User-Adapted Interaction,* pp. 331-370, 2002.

[9] R. L. Mooney RJ, "Content -Based Book Recommending Using Learning for Text Categorization," in *ift h ACM Conference on Digital Libraries*.

[10] B. D. Pazzani MJ, "Content -Based Recommendation Systems. The Adaptive Web: Methods and Strategies of Web Personalization".

[11] K. Felfernig A., "selling complex products and services using knowledge-based recommender technologies," in *7th IEEE International Conference on E-Commerce Technology* .

[12] S. T. A.-O. a. M.Ykhlef, "A survey of job recommender systems," *Internat ional Journal of the Physical Sciences,* vol. 7, pp. 421-425, 2012.

[13] L. S. M. C. E. Lang S, "Drivers, Challenges and Consequences of E-Recruiting – A Literature Review," pp. 26-35. , 2011.

[14] A. B. B. S. a. L. R. Moshe Unger, "Towards latent context aware recommendation systems. KnowledgeBased Systems," p. 165–178, 2016.

[15] F. Z. X. X. a. G. Jianxun Lian, "A Content Boosted Collaborative Filtering Neural Network for Cross Domain Recommender Systems," in *26th International Conference on World Wide Web Companion*, 2017.

[16] L. Y. a. X. X. Shuai Zhang, "An Effcient Hybrid Collaborative Filtering Model via Contractive Autoencoders," in *40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017.

[17] L. Y. A. S. Shuai Zhang, "Deep Learning based Recommender System: A Survey and New Perspectives," 2017.

[18] T. Adomavicius G, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," in *IEEE Transactions on Knowledge and Data Engineering*, 2005.

[19] A. S. V. T. X. X. H. K. K. Yu, "Probabilistic memory-based collaborative filtering," *IEEE Transactions on Knowledge and Data Engineering,* vol. 16, p. 56–69, 2004.

[20] S. T. P. e. W. Chu, "Pairwise preference regression for cold-start recommendation," in *RECSYS '09: Proceedings of the third ACM Conference on Recommender Systems, ,* New York, 2009.

[21] H. D. K. C. Breese JS, "Empirical analysis of predictive algorithms for collaborative filtering," in *Fourteenth Conference on Uncertainty in Artificial Intelligence,*, 1998.

[22] P. P. Hu R, "Enhancing Collaborative Filtering Systems with Personality Information," in *RecSys'11*, chicago, 2011.

[23] S. B. Y. J. Linden G, " Amazon.com Recommendations: Item-to-Item Collaborative Filtering," in *IEEE Computer Society, IEEE Internet Comput*, 2003.

[24] Z. D. C. H. Huang Z, "A Comparative Study of Recommendation Algorithms in ECommerce.," in *IEEE Intell. Syst*, 2007.

[25] B. R., " Integrating Knowledge-Based and Collaborative Filtering Recommender Systems," in *AAAI Workshop on AI in Electronic Commerce*, florida, 1999.

[26] K. G. K. J. R. J. Sarwar B, "Incremental SVDBased Algorithms for Highly scaleable Recommender Systems," in *5th International Conference on Computer and Information Technology* , 2002.

[27] S. M. M. M. Felfernig A, "Recommendation and Decision Technologies For Requirements Engineering," in *RSSE* , cape town, 2010.

[28] S. S. Bhargava HK, "Tools for Building Decision Support Systems," in *IEEE Computer* , 1999.

[29] B. R, "Knowledge-Based Recommender Systems.," *Encyclopedia of Library and Information Encyclopedia of Library and Information*, 2000.

[30] M. H. E. Y. A. Z. ,. A. I. Samir Anter, "Towards a generic model of a user profile," in *Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS)*, 2016.

[31] K.-N. C. You-Jin Park, "Individual and group behavior-based customer profile model for personalized product recommendation," *Expert System Application* , 2009.

[32] S. e. D. H. KECHID, "Habiba. Accès personnalisé à de multiples serveurs d'informations," pp. 249-255, 2006.

[33] U. S. G. Amato, "User profile modeling and applications to digital," in *3rd european conf. Research and advanced technology*, 1999.

[34] S. M. C. A. M. Gauch S, "A User profiles for personalized information access.," *The Adaptive Web*, pp. 54-89, 2007.

[35] S. Y. Morita Morita, "Informat ion filtering based on user behavior analysis and best match text retrieval," in *SIGIR '94 ACM*, 1994.

[36] T. J. Kelly D, "Implicit feedback for inferring user preference: a bibliography," p. 18–28, 2003.

[37] K. G. K. J. R. J. Sarwar B, "Item-based collaborative filtering recommendation algorithms," in *10th international conference on World Wide Web*, Hong Kong, 2001.

[38] P. A. U. L. P. D. Schein AI, "Methods and metrics for cold-start recommendations," in *25th annual internat ional ACM SIGIR conference on Research and development in information retrieval,* , New York, 2002.

[39] D. Evans, "A Summary of the CLARIT Project, Project, Technical Report, Laboratory for Computational Computational," Carnegie-Mellon University, 1991.

[40] Y. Yang, "An Example-Based Mapping Method for Text Categorization and Retrieval," *ACM Transactions on Information Systems,* Vols. Vol. 12, No. 3, pp. 252-277, 1994.

[41] E. V. T. L. Mamadou Diaby, "Toward the Next Generation of Recruitment Tools: An Online Social Network-based Job Recommender System," *ASONAM'13,* 2013.

[42] M. Z. A. F. a. G. F. D. Jannach, "Recommender Systems: An Introduction," Cambridge University Press, 2011.

[43] C. K. I. Y. I. G. H. P. ALPER BILGE, "A survey of privacy-preserving collaborative filtering schemes," *International Journal of Software Engineering and Knowledge Engineering* , vol. 23, pp. 1085-1108, 2013.

[44] D. Zibriczky, "A Combination of Simple Models by Forward Predictor Selection for Job Recommendation," in *RecSys Challenge ’16,* , Boston, 2016.

[45] K. V. L. M. Toon De Pessemier, "A Scalable, High-performance Algorithm for Hybrid Job," *RecSys Challenge ’16,* no. RecSys, 2016.

[46] J. Bauer, "Identification and Modeling of Contexts for Different Information Scenarios in," Technical University of Berlin, 2003.

[47] J. I. A. R. Karen Henricksen, "Modeling Context Information in Pervasive Computing Systems," *Pervasive,* no. 2002, pp. 167-180.

[48] G. Fischer, "Context-Aware Systems—The ‘Right’ Information, at the ‘Right’ Time, in the ‘Right’ Place, in the ‘Right’ Way, to the ‘Right’ Person," p. pp.287294, 2012.

[49] T. S. a. C. L. Popien, "A Context Modeling Survey," in *Workshop on Advanced Context Modeling, Reasoning and Management, The Sixth International Conference on Ubiquitous Computing*, Nottingham,, 2004.

[50] O. B. K. H. J. I. D. N. A. R. D. R. Claudio Bettini, "A survey of context modeling and reasoning techniques," *Pervasive and Mobile Computing,* vol. 6, pp. 161-180, 2010.

[51] S. D. a. F. R. Matthias Baldauf, " A survey on context aware systems," *Int. J. Ad Hoc and Ubiquitous Computing,* vol. 2(4), pp. 263-277, 2007.

[52] D. Ejigu, "Context Modeling and Collaborative Context-aware Services for Pervasive Computing," *Journal of Computers,* vol. 3, pp. 40-50, 2008.

[53] S. S. M. B. Anand, " Contextual recommendation," *Lecture Notes In Artificial Intelligence,* vol. 4737, p. 142–160, 2007.

[54] C. S. D. G. Sandy El Helou, "The 3A Personalized, Contextual and Relation-based Recommender System,," *Journal of Universal Computer Science,* vol. 16, 2010.

[55] P. V. K. V. E. D. a. J. B. V. Butoianu, "User Context and Personalized Learning: a Federation of Contextualized Attention Metadata," *Journal of Universal Computer Science,* vol. 16, pp. 252-2271, 2010.

[56] M. S. M. F. U. K. H. S. a. M. W. K. Niemann, "Usage-based Object Similarity," *Journal of Universal Computer Science,* vol. 16, pp. 2272-2290, 2010.

[57] K. W. O. W. T. Malinowski J, "Matching People and Jobs: A Bilateral Recommendation Approach," in *39th Annual Hawaii International Conference on System Sciences*, Hawaii, 2006.

[58] C. R. K. V. V. C. a. N. K. l. A. Singh, "PROSPECT: a system for screening candidates for recruitment," in *19th ACM International Conference on Information and Knowledge Management*, Toronto, 2010.

[59] D. H. L. &. P. Brusilovsky, "Fight ing Information Overflow wit h Personalized Comprehensive Informat ion Access: A Proactive Job Recommender”," in *Third International Conference on Autonomic and Autonomous Systems* , 2007.

[60] S. B. Rafter R, " Passive Profiling from Server Logs in an Online Recruitment Environment," in *IJCAI Workshop on Intelligent Techniques for Web Personalization (ITWP 2001)*, Washington, 2001.

[61] W. T. K. T. (. Färber F, "An Automated Recommendation Approach to Selection in Personnel Recruitment," in *AMCIS*, FL, 2003.

[62] E. A. Laumer S, "Help to Find the Needle in a Haystack: Integrating Recommender Systems in an IT Supported Staff Recruitment System," in *special interest group on management information system's 47th annual conference on Computer personnel Research*, Limerick, 2009.

[63] E. A. W. T. Laumer S, "Electronic Human Resources Management In An Ebusiness Environment.," in *J. Electron. Commer. Res.* , 2010.

[64] K. W. T. Malinowski J, " Analyzing the Impact of IS Support on Recruitment Processes: An E-recruitment Phase Model," in *ninth Pacific Asia conference on information systems (PACIS-2005)*, Bangkok, 2005.

[65] M. M. E. J. T. S. Carroll M, "Recruitment in Small Firms: Processes, Methods and Problems. Employee Relations," vol. 21, pp. 236-250, 1999.

[66] S. M. Breaugh JA, "Research on Employee Recruitment: So Many Studies, So Many Remaining Questions," vol. 26, 2000.

[67] K. T, "Extending the Applicability of Recommender Systems:A Multilayer Framework for Matching Human Resources," in *40th Annual Hawaii International Conference on System Sci. (HICSS'07)*, Hawaii, 2007.

[68] W. T. K. T. Malinowski J, "Decision Support for Team Staffing: An Automated Relational Recommendation Approach," vol. 45, pp. 429-447, 2008.

[69] P. J. Hofmann T, "Latent Class Models for Collaborative Filtering," in *16th International Joint Conference on Artificial Intelligence*, Stockholm, 1999.

[70] C. B. G. A. Paparrizos I, "Machine Learned Job Recommendation," in *fifth ACM conference on Recommender systems, RecSys''11*, Chicago, 2011.

[71] K. J. R. J. Schafer J, " Recommender Systems in ECommerce," in *First ACM Conference on Electronic Commerce (EC 1999)*, Denver, 1999.

[72] P. Brusilovsky, "Adaptive hypermedia. User Model. User Adapt. Interact," vol. 11, pp. 87-110, 2001.

[73] F. M. Fazel-Zarandi M, "Semantic Matchmaking for Job Recruitment: An Ontolgy Based Hybrid Approach," in *3rd International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web at the 8th International Semantic Web Conference.*, Washington D.C., 2010.

[74] P. Chen, "A Fuzzy Multiple Criteria Decision Making Model in Employee Recruitment," . *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 9, pp. 113-117, 2009.

[75] L. C. Z. F. Yu H, "Reciprocal Recommendation Algorithm for the Field of Recruitment," *Inform. Comput. Sci*, vol. 8, 2011.

[76] F. G. Murat Yagci, "A Ranker Ensemble for Multi-objective Job Recommendation," *RecSys Challenge*, 2017.

[77] M. R. Sonu K. Mishra, "A Bottom-Up Approach to Job Recommendation System," *RecSys Challenge '16,,* 2016.

[78] J. H. S.-W. K. Yeon-Chang Lee, "On Recommending Job Openings," 2015.

[79] M. D. M. C. E. V. M.-A. A. Emmanuel Malherbe, "Field Selection for Job Categorization and Recommendation to Social Network Users," *ASONAM 2014*, 2014.

[80] J. H. S.-W. K. Yeon-Chang Lee, "Job Recommendation in AskStory: Experiences, Methods, and Evaluation," *SAC 2016*, 2016.

[81] L. L. T. L. W. P. Wenxing Hong, "iHR: An Online Recruiting System for Xiamen Talent Service Center," *KDD'13,* 2013.

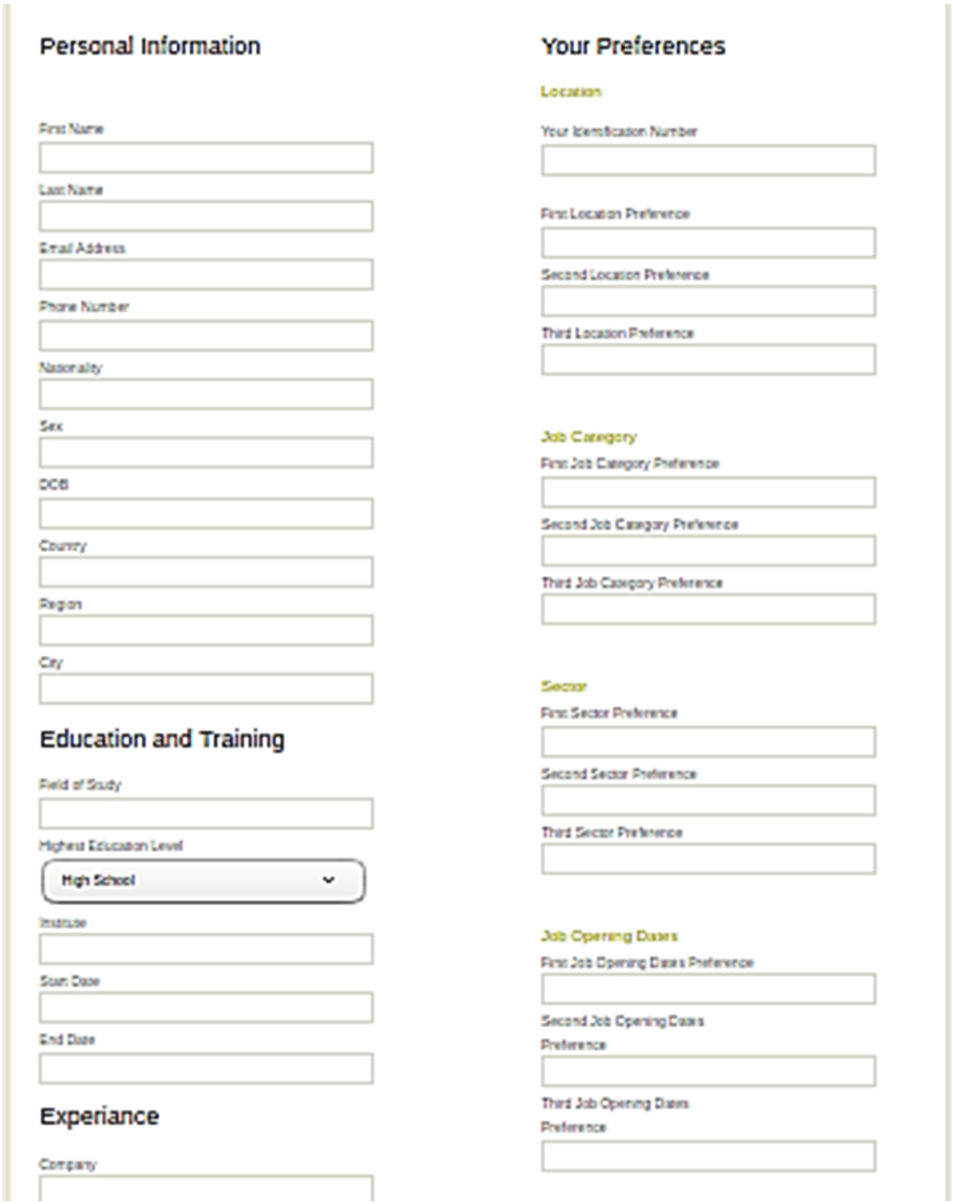
Annexes

Annex A: Sample Code – CB

Annex A

```
def tf_idf_CB(reccommend_for,ds):
    cb_score = []
    tf = TfidfVectorizer(analyzer='word', ngram_range=(1, 3), min_df=0, stop_words='english')
    tfidf_matrix = tf.fit_transform(ds['description'])
    cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
    results = {}
    for idx, row in ds.iterrows():
        similar_indices = cosine_similarities[idx].argsort()[:-100:-1]
        similar_items = [(cosine_similarities[idx][i], ds['id'][i]) for i in similar_indices]
        results[row['id']] = similar_items[1:]
    def item(id):
        return ds.loc[ds['id'] == id]['description'].tolist()[0].split(' - ')[0]
    def recommend(item_id, num):
        score = []
        print("Recommending " + str(num) + " jobs similar to " + item(item_id) + "...")
        print("-------")
        recs = results[item_id][:num]
        score = recs
        for rec in recs:
            # print("Recommended: " + item(rec[1]) + " (score:" + str(rec[0]) + ")")
            K = []
        return score
    cb_score = recommend(item_id=reccommend_for, num=10)
    return cb_score
```

Annex B: User Interface



DECLARATION

I, the undersigned, declare that this research is my original work and has not been presented for a degree in any other university, and that all sources of materials used for the thesis have been duly acknowledged.

Declared by:

Name: Selam Abebe Abera

Signature: _____

Date: _____

Confirmed by advisor:

Name: Fekade Getahun (PhD)

Signature: _____

Date: _____