A Map-based Job Recommender Model

Manal Alghieth[1], Amal A. Shargabi[2]

Information Technology Department
College of Computer, Qassim University, Qassim, Saudi Arabia

Abstract—**Location is one of the most important factors to consider when looking for offering a new job. Currently, there exist many job recommender systems to help match the right candidate with the right job. A review of the existing recommender systems, included within this article, reveals that there is an absence of appropriate mapping support offering for job recommendation. This article aims to propose a general map-based job recommender model, which is implemented and applied within a system for job seekers in Saudi Arabia. The system adapts content-based technique to recommend jobs using the cosine similarity and will help Saudi job seekers finding their desired job in an efficient way using interactive maps. This ultimately will contribute to Saudi Arabia moving forward to the digital transformation which is one of the major objectives to fulfill the Saudi vision 2030.**

Keywords—Recommender systems; content-based recommendation; location-based search; maps

I. INTRODUCTION

Finding a job in today's market is a major challenge. A common way to look for a job is to use job search websites. Rather than taking the time to search newspapers, company web sites, and other traditional job postings. A job search website can do it all with the click of a button. A job search engine facilitates the matching and communication of job opportunities between job seekers and employers.

The location of a job has the potential to significantly affect an individual's lifestyle. People often tend to focus their job search on a particular area and the job location can play an important part in the decision to apply for a job. Thus, location is one of the most important factors to consider when looking for a new job. As such, every job search website worth using has the ability to search for jobs based on location.

In the most popular job search websites, e.g. Indeed (www.indeed.com) and Monster (www.monster.com), the representation for the retrieved information may not be appropriate for job seekers with respect to the job location. The reason is because the results returned to a job seeker about the job locations are in textual form with no provision of a map for the employers' geospatial location.

As such, users of such websites may find difficult searching for their desired job, and not efficient because the lack of the website support in displaying the available jobs on a map might lead to them spending a significant amount of their time reading and reviewing their options regarding job location.

Two job search websites, namely Glassdoor (https://www.glassdoor.com/Job/explorer/index.htm) and Pathwayjobs (www.pathwayjobs.com), provide mapping tools

for job search but their tools are specifically designed for American job seekers only and they need to be personalized. They require recommender systems to meet job seekers preferences as thousands of jobs are posted on these websites daily, and it takes a great deal of effort to find the right position.

Currently, there exist many personalized job search systems, i.e. job recommender systems to help match the right candidate with the right job. Examples include CASPER [1], Proactive [2], FES [3], PROSPECT [4], eRecruiter [5], iHR [6], RésuMatcher [7] and the work of [8]. The work of [9] and [10] provides comprehensive review on job recommender systems.

Table I shows a general comparison among these systems based on the following characteristics: recommendation input, recommendation technique, and the offering of mapping support in these systems.

All the recommender systems of Table I support personalization in the job search, although they differ in the recommendation technique used. Most of the system use collaborative filtering recommendation (CFR) and content-based recommendation (CBR). Unfortunately, most of these systems are poor in personalized search because their search functionality is limited to keyword-based search, often resulting in poor, irrelevant search results. For example, a job search using the keyword "Java" to search for jobs within a limited geographical location (New York, NY) on www.indeed.com returned over 8000 jobs.

In the context of this research, it is worth mentioning that none of them offer mapping support tools. There are a number of international job search web sites, including Indeed (www.indeed.com), Monster (www.monster.com), Glassdoor (www.glassdoor.com),CareerBuilder(www.careerbuilder.com), SimplyHired(www.simplyhired.com),Pathwayjobs(www.path wayjobs.com), and LinkedIn (www.linkedin.com), that provide geospatial search. To the best of our knowledge, only two of these websites, namely Glassdoor and Pathwayjobs, provide mapping tools for job search and both of them are for American job seekers only. Anyway, none of them are recommender systems.

This research aims is to bridge the above two mentioned gaps in the current job search websites and systems, and proposes a personalized model based on job seeker preferences and also support location-based search with interactive mapping tool. The proposed model and prototype will help job seekers, especially in Saudi Arabia, to find the right job that meets their qualifications in an efficient way using interactive maps.

TABLE. I.   A SUMMARY OF EXISTING JOB RECOMMENDATION SYSTEMS

Job Recommender System	Year	Reference	Recommendation Input (User Profile)	Recommendation Technique	Mapping Support
CASPER	2000	[1]	Personal information User behaviour (revisited data, read time data, activity data, feedback) Query (job description, salary, location, education background)	CFR CBR	No
Proactive	2000	[2]	Personal information (includes preferences and interests either predefined or general)	CBR (Ontology) KBR	No
FES	2007	[3]	Personal information (Qualification, class of degree, years of experience, certifications, age, course of study)	Fuzzy-based	No
PROSPECT	2010	[4]	Personal information	CBR (Resume Mining)	No
eRecruiter	2011	[5]	Personal information (include interest) User behaviour (include feedback in real-time)	CBR KBR (Ontology)	No
iHR	2013	[6]	Personal information User behaviour	CBR CFR Hybrid Cluster-based	No
Social JRS	2013	[11]	Users data, users friends data, users profile	SVM	No
iHR+	2015	[12]	Basic information only	Lucene IKAnalyzer TF-ADF	No
SKILL	2015	[13]	Skills in a resume , requirements in job poster	Skills taxonomy, skills tagging	No
RésuMatcher	2016	[7]	Resume document	Statistical similarity FST	No
Graph-based recommender	2017	[8]	Personal information User behaviour (active and passive users)	CBR Graph-based	No
DNN Model for Job Matching	2018	[14]	Personal information (age, gender, major, profession, educational level and employment history)	DNN	No

These research work main research contributions are:

- Construction of a map-based job recommender model.

- Evaluation of the constructed job model using a simple prototype.

The rest of the article is organized as follows: Section II presents the proposed map-based job recommender model. The section starts with a subsection on the theoretical base of the proposed model and then explains the proposed model elements in detail. Section III presents the several modules of a prototype implemented based on the proposed model and applied for Saudi job seekers. Finally, Section IV provides a simple comparison between the proposed prototype and other Saudi job recruitment systems. Finally Section V summarizes and concludes the article.

II.   A PROPOSED MAP-BASED JOB RECOMMENDER MODEL

Despite the differences among the different job recommender systems, most of them have the same general architecture, as the proposed model does. Therefore, before describing in detail the proposed model, its general architecture is first introduced in the following subsection.

A. The General Architecture of Recommendation Systems

The general architecture of recommendation systems consists of three basic elements: the input data, the recommendation technique, and the recommendation output (Fig. 1).

The input is the users' preferences used for recommendation such as user profile. The input captures the main preferences of users and is the content of the user profile, which can contain one or more of three types of data: 1) individual information such as educational experience, working experience and skills; 2) user's historical behaviors regarding job application and collecting job posts, e.g. CASPER and eRecruiter; and 3) user preference captured based on a description of a preferred job, e.g. CASPER, or based on mining the resume of the users, e.g. PROSPECT and eRecruiter. The input data also include the available jobs provided by recruiters.

Fig. 1.   The General Architecture of Recommendation System.

The recommendation technique is the core of the recommender system, and it refers to the recommendation strategy to use. In the literature, several techniques have been introduced based on the following filtering techniques:

1) Collaborative filtering makes a user-to-user comparison in order to suggest preferred items from similar users. For suggesting items, collaborative filtering recommender systems search for users with similar taste. Most often, such recommender systems are based on ratings. These ratings are then used to match similar users; for example, two users have the same rating for the same article. Afterwards, the preferences of similar users are re-used for recommending items. An example of collaborative filtering recommenders is [15].

2) Content-based filtering tries to find items that are similar to items the user likes. Content-based recommender systems analyze item descriptions to identify items of particular interest to the user. This type of recommender system may be used in a variety of different domains, such as web page recommendations, television programs, news articles, and social media content. This approach has its roots in the information retrieval area, as methods for searching for documents are involved. Compared to traditional information retrieval applications, these recommender systems require user profiles, which encode the user's preferences. An example of collaborative filtering recommenders is [4].

3) Knowledge-based uses additional knowledge in order to infer items that best match the user's needs. Knowledge-based recommender systems use knowledge about the items, the users and on how to map users' needs to items' features, the so-called functional knowledge. Knowledge-based recommender systems suggest products based on inferences about a user's needs and preferences that are derived explicitly from their mapping to product features [14]. An example of such recommenders is [16].

4) Hybrid filtering combines two or more of the above techniques to achieve better performance. An example of these filtering is [17].

Different systems may employ different recommendation strategies approaches based on their own user profiles. For example, PROSPECT uses content filtering techniques while CASPER applies hybrid techniques based on content and collaborative filtering techniques.

After taking the user profile as the input and applying the recommendation technique, the recommendation system outputs the recommendation results that satisfy the desires of users. The output is usually in the form of a list of recommended jobs. It may also include "You Maybe Also Like" and "What Others Looking" jobs.

B. The Proposed Model

The proposed model is based on the general architecture of a recommendation system presented in the previous section (see Fig. 2). The details of its three elements are provided below.

- Input

This element consists of the job seeker and job vacancies data. The job seekers data are the preferences of the people who are looking for jobs. In the proposed model two main preferences have been identified: job title and location. The location in the proposed model is central because the proposed model is a map-based model. The job vacancies data are the jobs postings to be retrieved from common job boards' sites.

- Recommender

This element gets the job seeker preferences, i.e., job title and location as well as the job retrieved from job boards' sites and recommend the most similar jobs. The recommendation technique in the proposed model is content-based.

The basic approach in content-based filtering is to match similarity between two texts based on the count of common words between the two documents. This provides a measure of the 'Euclidean distance' between the two documents. As the size of documents increases, the number of common words would increase, even if the documents are related to the same topic/idea. The Euclidean distance, therefore, stops being a good measure of the match between the documents. However, a measure that can indicate the closeness of two documents irrespective of their size is the cosine of the angle between two associated vectors, i.e. arrays having word counts in the two documents, projected in a multidimensional space (Fig. 3). The axes correspond to search words being compared. It is obvious that we might have a large Euclidean distance between very similar vectors, i.e. between vectors with a small angle (hence a larger value of cosθ). The cosine gives a normalized measure of the match: 0 for no match to 1 for a perfect match.

- Output

The output is a visualizer element which is responsible for providing a map support for the recommended jobs. The visualizer gets the recommended jobs locations and processes them and then shows them on an interactive map.
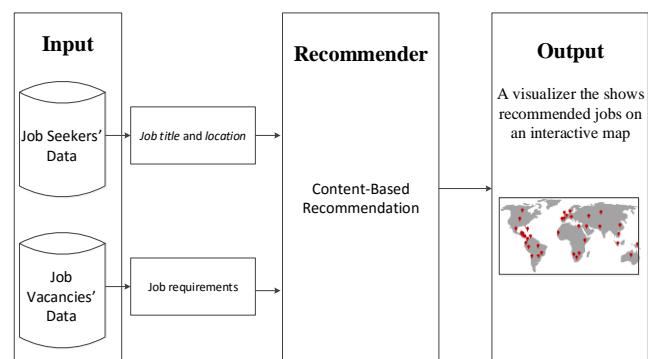

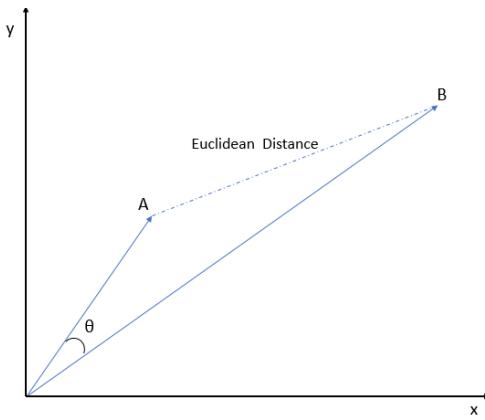Fig. 2.   The Architecture of the Proposed Model.

Fig. 3.   Cosine Similarity.

III.   A Prototype based on the Proposed Model

To realize and evaluate the proposed model in the previous section, a corresponding prototype is developed. The prototype was implemented for Saudi job seekers and consists of three modules (Fig. 4): job grabber, job recommender, location geocoder and geometry visualizer.

As shown in the figure, there are three modules: job grabber, job recommender, location geocoder and geometry visualizer. The following subsections describe each module in detail.

A.   The Job Grabber

The proposed system is meant for Saudi job seekers; thus, the module extracts a list of available jobs posted online in Saudi Arabia. Specifically, the module is based on Mihnati.com, one of the most popular job boards in Saudi Arabia and Gulf countries. This module scrapes the jobs posted on Mihnati.com and saves then save them as a Comma-separated values file, i.e., CSV file. The jobs posted are grabbed either they were in English or Arabic languages.

Three main elements of each job are extracted: job title, address and URL. The address here is essential to be used later to the recommended jobs on a map.

To implement this module, a web scrapper is developed using Python language. Specifically, the Beautiful Soup library is used for pulling jobs out. A sample of the grabbed jobs is shown in Fig. 5.

B.   Job Recommender

The job recommender module is a content-based recommender system. This module reads the scrapped jobs from the CSV file produced from the job extractor module and recommends jobs based on job seeker filters. These filters are job title and location.

The entered job title filter is checked for cosine similarity against jobs saved in the CSV file and shortlisted in decreasing order based on their cosine-similarity values, with job location being used as a secondary ordering criterion of the sorting. The jobs within a certain distance from the preferred location are included in the shortlist, and sorting is done based on increasing distance from the preferred location. The algorithm used in this module is described.

Algorithm 1: Recommendation Algorithm used in job recommender module

```
Define the minimum cosine-similarity threshold for a
match;
Define the maximum distance threshold between job
location and preferred location;
Read user job title and preferred location;
Define a data-frame with the columns Job title, Job
location and Job link;
Load job records from the CSV file into the jobdata
data frame;
Define a job_shortlist data frame with columns:
jobdata-index, cosine similarity and distance;
For each job title in the data-frame
        Create a vector of common words between job-
        search title and job title on file;
        Calculate the cosine-similarity;
For records with cosine >= cosine threshold
        Calculate distance between location preference
        and job location;
If distance <= distance threshold
        Move job-data index, cosine-similarity, distance
        to shortlist data frame;
Sort the shortlist data-frame on decreasing cosine-
similarity and increasing distance;
Use the index in the short-list record to retrieve the
job-data record;
Display the job title, job location and job link in the
sorted order;
```

The cosine similarity was chosen as the metric to rank the recommendations, as the job-search text is usually small and hence the number of common words (Euclidean distance) is not a good indicator of a match in this content-based recommendation. A minimum threshold of 0.8 was used for the cosine similarity, although this is customizable. The vector orientation is more important in this work than the magnitude. The search is further narrowed down by job-location comparison. The results from the recommendation can also be plotted on a map as highlighted locations within a radius of the specified maximum distance from the preferred location.

The module is implemented in Python and uses packages CSV to read comma-separated values for the database file, Counter and CountVectorizer to create vectors (word count arrays), and cosine_similarity to calculate cosine similarity. To calculate the distance between two cities, the package Nominatim is imported to get geocoordinates, i.e., latitude and longitude, of a place, and a package distance is imported to calculate the distance between two geo-coordinates. Package pandas are imported to use the Python data-frame structure.

The module can be further extended by including cosine-similarity calculations on other parameters like work experience, expected vs offered compensation, etc. It can also be adapted to take input directly from a resume rather than a user.
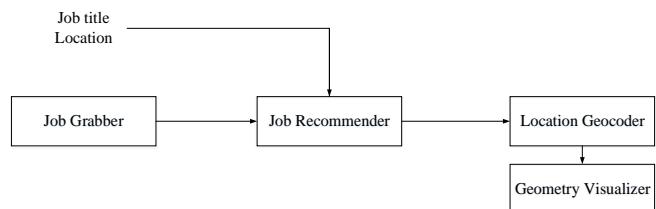
The recommended jobs are shown in Fig. 6.

Fig. 4.   Prototype Architecture.

C. Location Geocoder

Geocoding is the process of converting addresses into geographic coordinates, i.e. latitude and longitude.

This module reads the addresses of the recommended jobs and formats the address to be complete if they are not. The geographic coordinates of the formatted addresses are then found. As an example, the address of the third recommended job in Fig. 6 was extracted as:

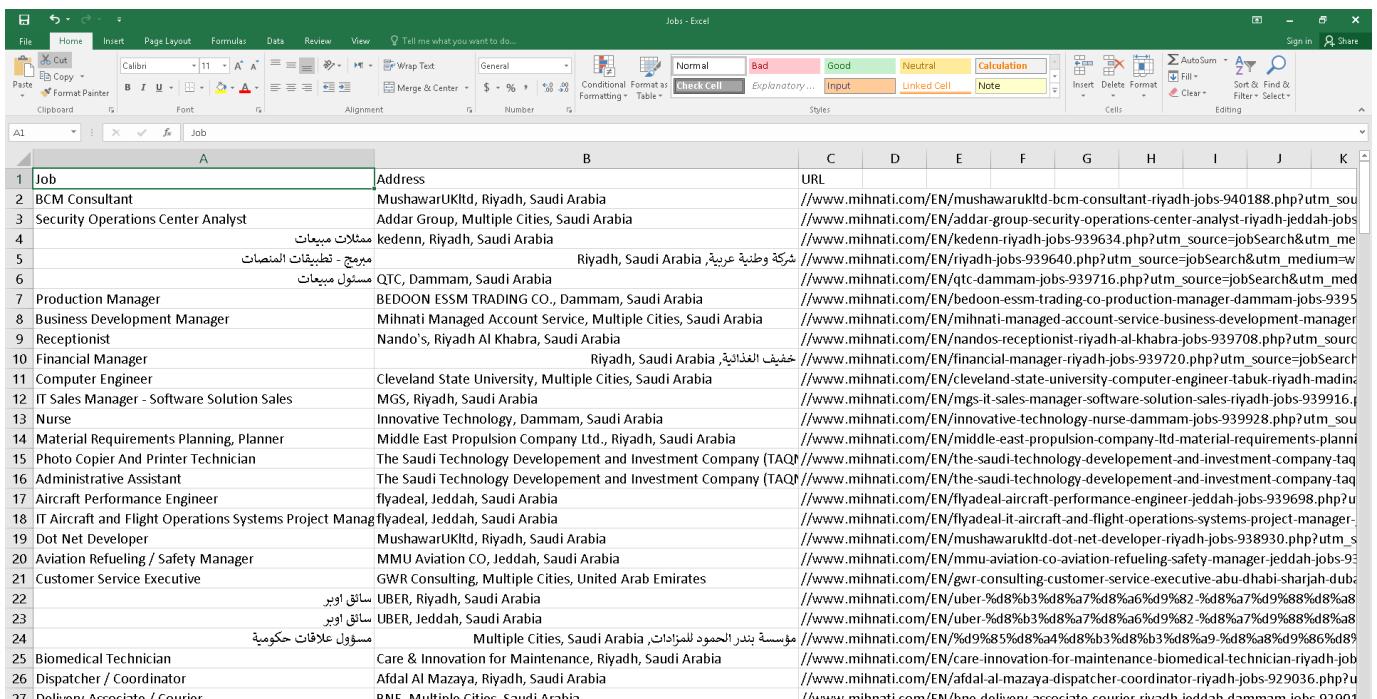Al Othman Agriculture Production and Processing Company (NADA), Khobar, Saudi Arabia

In order to be able to visualize the address on a map, the Location Geocoder module finds the complete address, i.e. formatted address as well as the geographic coordinates, i.e. geometry, in terms of latitude and longitude.

The results got form; this module is shown in Fig. 7. To implement this module, Google Geocoding API is used. In order to use Google APIs, an API key is needed. This key should be later embedded in the code.

D. Geometry Visualizer

Markers are one of the most common ways to visualize locations on a map. In this module, the geocoded addresses of the recommended jobs are visualized on a map using markers. Fig. 8 shows on a map the locations of the six recommended jobs shown in Fig. 6.

To implement this module, Google Maps JavaScript API is used.

Fig. 5.   Jobs Grabbed from Mihnati.com.

Fig. 6.   Recommended Jobs.

```
                                                                                              all_1_1_1.html:49
▼Object 🔢
 ▶config: {adapter: f, transformRequest: {…}, transformResponse: {…}, timeout: 0, xsrfCookieName: "XSRF-TOKEN", …}
 ▼data:
   ▼results: Array(1)
     ▼0:
       ▶address_components: (6) [{…}, {…}, {…}, {…}, {…}, {…}]
        formatted_address: "Prince Sultan Road, Al Jawharah, Al Khobar 34434, Saudi Arabia"   ⟵
       ▼geometry:
         ▼location:
             lat: 26.3338725    ⟸
             lng: 50.19116
           ▶__proto__: Object
          location_type: "GEOMETRIC_CENTER"
         ▶viewport: {northeast: {…}, southwest: {…}}
         ▶__proto__: Object
        place_id: "ChIJXc0ipqXoST4RJ4GpVGMQTh8"
       ▶plus_code: {compound_code: "85MR+GF Al Khobar Saudi Arabia", global_code: "7HRG85MR+GF"}
       ▶types: (3) ["establishment", "finance", "point_of_interest"]
       ▶__proto__: Object
       length: 1
     ▶__proto__: Array(0)
     status: "OK"
   ▶__proto__: Object
 ▶headers: {content-type: "application/json; charset=UTF-8", cache-control: "public, max-age=86400", expires: "Fri, 31 May 2019 13:13:36 GMT"}
 ▶request: XMLHttpRequest {onreadystatechange: f, readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, …}
  status: 200
  statusText: ""
 ▶__proto__: Object
```

Fig. 7. Geocoded Address of a Selected Recommended Job, Arrows Shows the Formatted Address, Latitude and Longitude of a given Address.
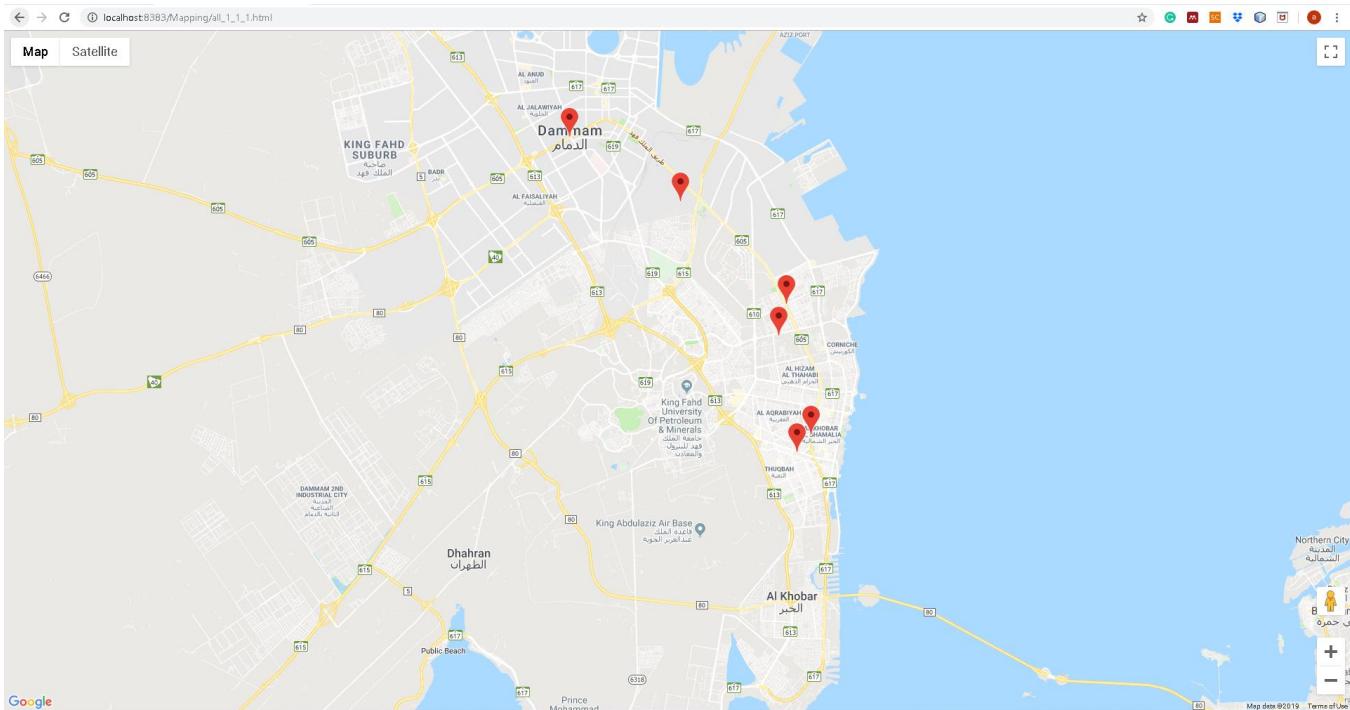

Fig. 8. The Recommended Jobs are Shown on a Map.

IV. A COMPARISON BETWEEN THE MAP-BASED JOB RECOMMENDER PROTOTYPE AND OTHER SAUDI JOB SYSTEMS

As an initial evaluation for the proposed model, the prototype that has been implemented based on that model is compared with JADARAH. The most commonly used system by Saudi job seekers. JADARAH was developed on 2011 by the ministry of civil service to help Saudi job seekers find jobs within Saudi government sectors. The comparison is made in terms personalization, map-based support, and type of jobs offered.

For personalization criterion, both systems are recommender systems. That is, the jobs are personalized based on job seeker's qualifications and preferences. However, in our prototype, the location is highly considered during the recommendation process. We believe that, in Saudi Arabia, job location is essential when looking for a new job. This is due to the large geographical area of the country and the long distances among its several regions. Unlike our system, JADARAH does not pay great attention to job seeker location. That is, a person who lives in Qassim which is located in the central region may be got a job in Abha which is located in the southern region.

The proposed system is map-based as the recommended jobs are displayed on a map. Unlike our system, JADARAH displays the recommended job in textual form.

In terms of types of jobs, our system considers jobs from private companies as well as governmental sectors. On the other hand, JADARAH is only for governmental jobs.

It is worth mentioning that there are other job requirement systems in Saudi Arabia such as bayt.com, wadhefa.com, and tanqeeb.com, however, these systems were not included in the comparison as these systems do not provide personalized search, i.e. not recommender systems.

V. CONCLUSION

Although there are many job recommendation systems, these systems do not offer mapping support. In line with the digital transformation objectives of the new Saudi vision 2030, and in order to improve job search in general and in Saudi Arabia in particular, this work proposed a personalized and map-based job search model. The model was theoretically based on the existing recommender systems in the literature and used content-based recommendation with integration of mapping feature for location-based search which has never been used in the previous systems. The cosine similarity was used for the content-based recommendation with a minimum threshold of 0.8 for job title search and further narrowed down by job location comparison. As a proof of concept, a prototype was implemented based on the proposed model. The proposed system provides better features compared with JADARAH, the most common recruitment system used by Saudi job seekers. The proposed system in this research work will help Saudi job seekers finding the desired job in an efficient way using maps. In this work, the proposed system is meant by Saudi job seekers as it is based on Mihanti.com as a main source for jobs. In the future, we plan make more general and grab jobs from several international job's boards.

ACKNOWLEDGMENT

REFERENCES

[1] R. Rafter, K. Bradley, and B. Smyth, "Personalised Retrieval for Online Recruitment Services," 2000.

[2] D. H. Lee and P. Brusilovsky, "Fighting Information Overflow with Personalized Comprehensive Information Access: A Proactive Job Recommender," in Third International Conference on Autonomic and Autonomous Systems (ICAS'07), 2007, pp. 21–21.

[3] J. O. Daramola, O. O. Oladipupo, and A. G. Musa, "A fuzzy expert system (FES) tool for online personnel recruitments," Int. J. Bus. Inf. Syst., vol. 6, no. 4, p. 444, 2010.

[4] A. Singh, C. Rose, K. Visweswariah, V. Chenthamarakshan, and N. Kambhatla, "PROSPECT: A system for screening candidates for recruitment," in Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10, 2010, p. 659.

[5] M. Hutterer, "Enhancing a Job Recommender with Implicit User Feedback," vol. 2011, 2011.

[6] W. Hong, S. Zheng, H. Wang, and J. Shi, "A Job Recommender System Based on User Clustering," J. Comput., vol. 8, no. 8, Aug. 2013.

[7] S. Guo, F. Alamudun, and T. Hammond, "RésuMatcher: A personalized résumé-job matching system," Expert Syst. Appl., vol. 60, pp. 169–182, Oct. 2016.

[8] W. Shalaby et al., "Help Me Find a Job: A Graph-based Approach for Job Recommendation at Scale," Dec. 2017.

[9] Z. Siting, H. Wenxing, Z. Ning, and Y. Fan, "Job recommender systems: A survey," in 2012 7th International Conference on Computer Science & Education (ICCSE), 2012, pp. 920–924.

[10] S. T. Al-Otaibi and M. Ykhlef, "A survey of job recommender systems," Int. J. Phys. Sci., vol. 7, no. 29, pp. 5127–5142, Jul. 2012.

[11] M. Diaby, E. Viennet, and T. Launay, "Toward the next generation of recruitment tools: An online social network-based job recommender system," in Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2013, 2013, pp. 821–828.

[12] H. Wenxing, C. Yiwei, Q. Jianwei, and H. Yin, "iHR+: A mobile reciprocal job recommender system," in 2015 10th International Conference on Computer Science & Education (ICCSE), 2015, pp. 492–495.

[13] M. Zhao, F. Javed, F. Jacob, and M. McNair, "SKILL: A system for skill identification and normalization," in Proceedings of the National Conference on Artificial Intelligence, 2015, vol. 5, pp. 4012–4017.

[14] S. Maheshwary and H. Misra, "Matching Resumes to Jobs via Deep Siamese Network," in WWW '18 Companion Proceedings of the The Web Conference 2018, 2018, pp. 87–88.

[15] K. Haruna, M. Akmar Ismail, D. Damiasih, J. Sutopo, and T. Herawan, "A collaborative approach for research paper recommender system," PLoS One, vol. 12, no. 10, p. e0184516, Oct. 2017.

[16] Y. Balachander and T.-S. Moh, "Ontology Based Similarity for Information Technology Skills," in 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2018, pp. 302–305.

[17] Y. Wang, M. Wang, and W. Xu, "A Sentiment-Enhanced Hybrid Recommender System for Movie Recommendation: A Big Data Analytics Framework," 2018.