SKILL/JOB RECOMMENDER APPLICATION

LITERATURE SURVEY

TEAM ID : PNT2022TMID04854

S.NO	PAPER	AUTHOR	YEAR	SHORT DESCRIPTION	RESULT	FUTURE WORK AND ANALYSIS
1	Job Recommen-dation based on job seekers skills	Jorge Valverde-Rebaza, Ricardo Puma, Paul Bustios, Nathalia C. Silva	2018	1.Based on the person-job fit premise, proposed a framework for job recommendation based on professional skills of job seekers. 2.Automatically extracted the skills from the job seeker profiles using a variety of text processing techniques. 3., we performed the job recommendation using TF-IDF and four different configurations of Word2vec over a dataset of job seeker profiles and job vacancies collected .	we observed that not all the profiles were given a good recommendation as the maximum value of the average minimum effectiveness is 0.96 (48 out of 50 profiles). This last metric is highly dependent on the quality of the filtering process and the variety of job offers since there can be a shortage of offers for some specific profiles.	Future directions of our work will focus on performing a more exhaustive evaluation considering a greater amount of methods and data as well as a comprehensive evaluation of the impact of each professional skill of a job seeker on the received job recommendation .

2	Job Recommen-der Systems: A Review	Corné de Ruijt , Sandjai Bhulai	2021	1.Content-based recommender systems (CBRs) in the context of JRS are models which, to construct a recommendation , only use a semantic similarity measure between the user profile and the set of available vacancies. 2.In collaborative filtering (CF), recommendations are based solely on behavioral data, typically stored in a user × items rating matrix. 3.Knowledge-based recommender systems as recommender systems having the conceptual goal to “Give me recommendations based on my explicit specifications of the kind of content (attributes) I want”.	An advantage of using job ontologies is that this also simplifies the implementation of keyword-based search 7 engines and simplifies filtering.	1.we managed to further split JRS hybrids into smaller categories, still some classes comprise similar methods.

3	A systematic review and research perspective on recomme-nder systems	Deepjyoti Roy and Mala Dutta	2022	This paper aims to fulfil this signifcant gap by reviewing and comparing existing articles on recommender systems based on a defned classifcation framework, their algorithmic categorization, simulation platforms used, applications focused, their features and challenges, dataset description and system performance.	From the set of research works, 35% of the works use recall measure, 16% of the works employ Mean Absolute Error (MAE), 11% of the works take Root Mean Square Error (RMSE), 41% of the papers consider precision, 30% of the contributions analyse F1-measure, 31% of the works apply accuracy and 6% of the works employ coverage measure to validate the performance of the recommender systems.	New research may extend this study to cover other journals and non-English papers.
4	Large-scale e-learning recommender system based on Spark and Hadoop	Karim Dahdouh1 q2, Ahmed Dakkak1 , Lahcen Oughdir1 and Abdelali Ibri	2019	Our system uses association rules for extracting more interesting relationships between learners' behaviours. Indeed, it aims to find similarities between courses enrollments in the transaction database.	It is clear that our recommender system using FP-growth run faster rather than when we choose to work with the Apriori algorithm.	In our future work will focus on incorporating Spark Streaming in the e-learning field. Actually, through real-time analysis of spark, it is possible to quickly extract value from live data streams.
5	CPRS: A cloud-based program	hin-Feng Lai a , Jui-Hung	2010	k-means and k nearest neighbour	In this phase, the average viewing time of each	More user profiles and longer histories

	recommendation system for digital TV platforms.	Chang a , Chia-Cheng Hub , Yueh-Min Huang a,∗ , Han-Chieh Chaoc		(kNN) are used to group users into clusters and to add a new user into a grouped cluster, respectively. Each program is assigned a weight, which is the sum of time periods that users have watched it. The weight of the program is used to indicate its popularity.	program is calculated and a program list sorted by the average viewing time is produced for recommendation. Subclasses kNNResMap and kNNResReduce are implemented to handle this task. In the map function of kNNResMap, all total viewing times of a program are extracted, and average_viewing_time is calculated as the mean of all total_viewing_time to a program_name. Output is collected as .	of a set of users should improve the recommendation results. With the power of cloud-computing, it is possible to recommend programs using a very large dataset.