1. INTRODUCTION
In recent years, the volume of data present online has grown exponentially. A major portion of this data is related to internet-based different platforms. The evaluation of such data and/or the extraction of information is difficult due to its huge volume. It is cumbersome for an individual or an organization to obtain the desired results in a timely manner. Hiring the right talent is a challenge faced by all companies. This challenge is amplified by the high volume of applicants if the business is labor intensive, growing and faces high attrition rates. One example of such a business is IT services run out of growth markets. In a typical services organization, professionals with varied technical skills and business domain expertise are hired and assigned to projects to solve customer problems. In the past few years, IT services including consulting, software development, technical support and IT outsourcing has witnessed explosive growth, especially in growth markets like India and China. For in-stance, according to a NASSCOM (National Association of Software and Services Companies of India) study, the total number of IT and IT enabled services professionals in India has grown from 284000 in 1999-2000 to over 1 million in 2004-2005 [12]. More recent estimates suggest that this industry employs more than 2 million professionals in India alone. For organizations in the IT Services business, growth in business is synonymous with growth in the number of employees and recruitment is a key function. Hiring large number of IT professionals in growth markets poses unique challenges. Most countries in growth markets have large populations of qualified technical people who all aspire to be part of the explosive growth in the IT Services industries. Thus, a job posting for a Java programmer can easily attract many tens of thousands of applications in a few weeks. Most IT Services companies are inundated with hundreds of thousands of applicants. For example, Infosys, one of the largest IT Outsourcing companies in India, received more than 1.3 million job applications in 2009. However, only 1% of them were hired. To give the context for work, consider a typical recruitment process. This is illustrated in Figure. The process starts when a business unit decides to hire employees to meet its business objectives. The business unit creates a job profile that specifies the role, job category, essential skills, location of the opening and a brief job description detailing the nature of work. It might also specify the total work experience that the prospective employee should possess, along with the desired experience level for each skill. The job openings are advertised through multiple channels like on line job portals, newspaper advertisements, etc. Candidates who are interested to apply for the job opening upload their profile through a designated web-site. The website typically provides an on-line form where the candidate enters details about her application like personal information,[5] education and experience details, skills, etc. We call this Candidate Meta data. The candidates can also upload their resumes through the website. The objective of allowing the candidate to enter meta data is to capture the information in a more structured format to facilitate automated analysis. However, real life experience suggests that most candidates do not specify a lot of information in the on line forms and hence Candidate Meta data is often incomplete] Once the applications of prospective candidates are received, they are subjected to careful scrutiny by a set of dedicated screeners.
This screening process[17] is crucial because it directly affects the quality of the intake and hence, the company profits.

A job recommender system is expected to provide recommendations in 2 ways: firstly recommending most eligible candidates for the specified job, to the recruiters and secondly, recommending jobs to the aspiring candidates according to their matching profiles. The focus of this paper is the second part only i.e. to recommend jobs to the candidates according to their matching profiles.

2. Objectives
The aim of recommender systems is to assist users in finding their way through huge databases and catalogues, by filtering and suggesting relevant items taking into account or inferring the users’ preferences (i.e., tastes, interests, or priorities). Based on this objectives for Job Recommendation systems are

1. Study of matching technique to match the suitable candidate for job position.
2. Use of Collaborative Filtering to find best suitable match.
3. Use content-based approach that takes into consideration an organization’s needs and the skills of candidate.
4. Comparison of matching technique with Collaborative Filtering and Content Based Recommendation.

3. Literature Review
Job Recommendation work resides in the domain of online recommender systems, which are widely adopted across many web applications, e.g., movie recommendations [12], e-commerce item recommendations [13], job recommendations [14] and so forth, where authors mainly concentrate on the relevance retrieval and ranking aspects of the recommendation system. There is insightful research and modeling of the hiring processes within job marketplaces. Such research includes work related to estimation of employee reputation for optimal hiring decisions [15], as well as work related to ranking and relevance aspects of job matching in labor marketplaces [16]. There has been work related to the theory of optimal hiring process, e.g., on the problem of finding the right hire for a job (the hiring problem), as well as on the classical secretary problem, where a growing company continuously interviews and decides whether to hire applicants [17,18]. Authors of [19] investigated job marketplace as a two-sided matching market using locally stable matching algorithms for solving the problem of finding a new job using social contacts. RS can be treated as one of the most efficient tools for business, aimed directly at increasing revenue and profitability as well as optimizing current product portfolio.

4. Proposed System
4.1 Brief Description
Proposed job recommendation system in this paper would take the input from tab separated files (TSV) to analyze the dataset. After analysis of the data the data would be transformed in the form of matrix by applying item-item collaborative filtering or user-user collaborative filtering. Once we have the data organized in the form of matrix we can apply different algorithms and machine learning techniques to find out the most suitable job for each user based on his or her analysis.

The central task of this challenge is to predict those job postings (items) that the user will interact with. Given a user, a job recommendation system should predict that job position that is likely to be relevant to the user. Proposed system makes use of Collaborative Filtering using cosine similarity and Content based recommendations using FirstText library.
4.2 Proposed System Design

The proposed Employment Recommendation System is divided mainly into four modules: Data acquisition module, Transformation Module, Computation module and Recommendation Module.

![Fig.4.2.1 Employment Recommendation System Modules](image)

Data acquisition module: In data acquisition module, data from different users are collected and stored in dataset. These data consist of user profile, skills, past activities, browsing history. For this purpose we are storing data in tab separated values (tsv) format files.

Transformation Module: Transformation stage deals with the different processes such as general information extraction and detail information extraction as per the requirement. If collaborative approach needs to be used in that case we are going to transform data present in dataset into the user-user and user-item matrices to be analyzed for recommendation.

Computation module: Computation module deals with the calculation part of the system. It mainly consists of two parts - Data filtering and Result set generation. In the data filtering we need to analyze the similarity between item-item and user-item.

Recommendation Module: The last stage is the Recommendation unit where recommendations to the users are made depending upon the filtered results set from computation module.

Matching based Recommendation design:

In the matching between people and jobs, the content is the personal information [20] and their job desires for people while for jobs, it’s the job description posted by recruiters, even including the background description of enterprises. In our case we have considered two parameters namely City and State for matching and recommending jobs to the users based on these two parameters based on the assumption that the user will prefer first the popular jobs in his city and followed by his state. Steps for recommending jobs to the users are as below:

Step 1: Read input data about all the jobs available from jobs.tsv file
Step 2: Read the data about job applications from apps.tsv file
Step 3: Sort the jobs based on popularity
Step 4: Create a directory of available jobs based on city, state and popularity
Step 5: Predict the top 10 matching jobs for each user

Content Based Recommendation design:

The principle of a content-based [3] recommender [9][10] is to suggest items that have similar content to ones the target user prefers. The process of content-based recommender is selecting the same feature type and comparing them by calculating their similarity for people and jobs [23]. The recommendatory result is a list of job positions or candidates sorted by the similarity index. In short, the two key components of content-based recommender [21] are feature selection and similarity calculation. During selecting feature, not only it’s need to select the common feature but also considering its influence on recommendation according to the target user’s preferences or the scientific analysis in the job recruiting market. Then the selected features should be represented in an appropriate form, for instance vector space model and their similarity can be calculated.

Our content-based recommender system uses the Vector Space Model (VSM) [32]. In the VSM, each job is represented by a vector in an n-dimensional space, where each dimension corresponds to a textual feature from the overall feature-set of the job collection.

Let \( J = (J_1, J_2, ..., J_N) \) denote a set of jobs and \( F = (f_1, f_2, ..., f_m) \) be the feature set. Formally, every job \( J_a \) is represented as a vector of feature weights, where each weight indicates the degree of association between the job and the feature:

\[
J_a = (w_{a1}^f, w_{a2}^f, ..., w_{am}^f),
\]

where \( w_{af}^f \) is the weight of feature \( f \) for job \( J_a \).

For feature weighting, we have used FirstText Deep learning library from Facebook.

Job Similarity Assessment: For a given feature representation [33], the similarity between two jobs, \( J_a \) and \( J_b \) is computed by the cosine similarity between their vector space representations, as follows:

\[
sim(J_a, J_b) = \frac{\sum_{i=1}^{m} w_{ai}^f \times w_{bi}^f}{\sqrt{\sum_{i=1}^{m} (w_{ai}^f)^2} \times \sqrt{\sum_{i=1}^{m} (w_{bi}^f)^2}},
\]

where \( w_{af}^f \) and \( w_{bf}^f \) are the weights of the feature \( f \) in job \( J_a \) and job \( J_b \) respectively.

Collaborative Filtering based recommendation Design:

The Collaborative filtering is one of the most successful approaches for building recommender systems. In Collaborative filtering important step of achieving matching jobs and people is calculating the similarity or relevance based on their profiles. Several common similarity calculation measures, namely, Constrained Pearson Correlation, Pearson Correlation, Spearman Rank Correlation, Cosine and Mean Squared Differences in the recommender system. For
example, with explicit rating information we can measure the similarity between two users or two jobs. We are going to use Cosine similarity and Mean Square differences in this employment recommendation system. The similarity measure is the measure of how much alike two data objects are. Similarity measure in a data mining context is a distance with dimensions representing features of the objects. If this distance is small, it will be the high degree of similarity where large distance will be the low degree of similarity. The similarity is subjective and is highly dependent on the domain and application. In this technique analysis of the user-item matrix to discover relations between different users or items and use them to compute the recommendations [35]. In our CF algorithm after the k most similar applicants have been identified, their corresponding rows in the jobs/applicants ratings matrix R are aggregated to identify the set of jobs, J, rated by the group together with their ratings. We predict ratings for the current applicant. Then, we recommend the top-N jobs depending on their ratings.[36]

**Computation process:**

**Cosine based Similarity:**
Moving forward to this similarity measure, any two things are taken as two items in the s dimensional client-space. The concept of angle is used here to calculate the similarity among the different items. The similarity between the two items [7] is calculated by finding out the cosine of the angle between the taken any two items. Formally, in the n × m ratings matrix (that is user-item matrix) , similarity between any items let suppose that we are taking the arbitrarily items i and j, denoted by

\[
sim(i,j) = \cos(i,j) = \frac{i \cdot j}{\|i\|^2 \cdot \|j\|^2}
\]

Steps for generation of recommendation list using different similarity measures are as follows:

**Phase 1**
Input: User-item matrix n×m that is R and k that indicates the count of job to job similarities that will be stored for each job.
Result: m×1 matrix M

**Step 1:** The user-job matrix is taken and job to job similarity is calculated using the different similarity measure.

**Step 2:** The value of M (i, j) is compared with the k most similar jobs. If it’s same then it is left as the value is else it is made zero.

This is what we get the output matrix M, which will be used in the next phase of the algorithm.

**Phase 2**
Input: The output matrix M from the previous phase, the matrix m×1 U which store the products that has been purchased beforehand by the users, and the variable N that specifies the number of items that will be recommended to the users.

Output: m×1 matrix x that stores number of jobs to be recommended. Its non-zero value indicates that the jobs that is in top n and is recommended to its users.

**5. Comparison between recommendation approaches**

All three approaches proposed in this paper that is matching based recommendation, collaborative based recommendation and content based recommendation have recommended / predicted different set of top 10 jobs recommendation for each user. If we compare all three approaches it is clear that matching algorithm will produce fastest results/recommendation but the recommendations might not be that much useful to the jobseekers as it only tries to match jobs based on certain parameters and does not contain any personalized information[23]. On the other hand both collaborative and content based recommendations will predict/recommend jobs that are personalized for that user and hence might be much relevant to the users. It is also clear that Collaborative filtering arrives at a recommendation that's based on a model of prior user behavior.[34] The model can be constructed solely from a single user’s behavior or more effectively also from the behavior of other users who have similar traits. When it takes other users' behavior into account, collaborative filtering uses group knowledge to form a recommendation based on like users. Comparison of these three approaches is summarized in below table:

<table>
<thead>
<tr>
<th>Parameters Considered</th>
<th>Matching based recommendation</th>
<th>Collaborative recommendation</th>
<th>Content based recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personalized recommenda tions</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Execution speed for generating recommendations</td>
<td>Fast</td>
<td>Slow</td>
<td>Very slow</td>
</tr>
<tr>
<td>Relevance to the user</td>
<td>May be</td>
<td>Based on ranking of similar jobs</td>
<td>Based on ranking of past behavior</td>
</tr>
<tr>
<td>Issues faced</td>
<td>No personalization</td>
<td>Cold start problem</td>
<td>Over specializa tion</td>
</tr>
</tbody>
</table>

Table 5.1: Comparison of different recommendation approaches
6. Results
The results of Cosine similarity based Collaborative filtering are promising in comparison to the results of matching technique. We have generated recommendations of collaborative filtering for different input data set and results generated are found to be more appropriate with the increased data set size.

Comparison of RMSE (Root Mean Square Error) for different size of input dataset for Collaborative filtering is shown in figure 6.1

![Figure 6.1 Relationship between user data set and RMSE](image)

Figure 6.1 Relationship between user data set and RMSE

Form the figure 6.1 it is clear that as we increase the size of dataset there is decrease in Root Mean square error and hence accuracy of prediction made by recommendation system is increasing with increase in input data samples

Comparison of User based RMSE and Job based RMSE is compared and results are shown in Figure 6.2

![Figure 6.2 Bar chart of User based RMSE, item based RMSE with Data set size](image)

Figure 6.2 Bar chart of User based RMSE, item based RMSE with Data set size

Both user based RMSE and item based RMSE performance is shown in graph for increasing size of input data set and RMSE is decreasing with increase in input data set and accuracy of prediction by recommendation system is increasing with increase in input data size.

7. Conclusion
We have designed the employment recommendation system using different recommendation techniques like simple matching and then collaborative and content based filtering. Similarity metrics are used to calculate how much similar all the items are to each other in the matrix. We implemented the algorithm and get the result accordingly. Comparison is made between different recommendation approaches. While the Matching techniques is the simplest one but recommendations that are generated might not be that useful to the jobseekers as they are not personalized. Both collaborative filtering and content based recommendation have generated the personalized recommendations and hence they are more useful to the jobseekers. But collaborative filtering suffers from cold start problem while content based recommendation might generate too specific results.

8. Future Scope
In this field of job recommendation system there is a large scope for future work such as:
- By using different similarity measure we can see which gives the most accurate answer when compared with the other similarity measures.
- If we take into consideration the recommendation of the recommender system in contrast with the real life preferences we can compare their mean absolute error.
- We can consider large number of parameters by giving them associated weights for more accurate Content Based recommendation
- Different Content based approaches can be compared like TF-IDF, Word to Vec with FirstText[8]
- Hybrid recommendations[11] can be generated either by combining the approaches or by combining the outputs generated by Content based and Collaborative recommendations.
- We can perform natural language processing to extract information from jobseekers resume and then recommending him the jobs

9. REFERENCES


