Career Path Recommender System

Palash Hariyani (1401101)

***Abstract*—Until recently, suggesting a career path based on the skills of a person and his preferences has been a tricky, tedious and time consuming process because a candidate often faces a dilemma when searching for a career based on his acquired skills and skills which he needs to acquire to be eligible for a particular job. Recommender Systems have been proposed in order to automate this task and also increase the effectiveness. The work here aims at extracting relevant skills from a users profile and classifying them according to the career he has chosen using clustering methods. Whenever a new user looking for a career uploads his profile, skills and other relevant information are extracted from his profile and matched with the database using Fuzzy logic. This basically matches based on a certain probability defined previously in the logic and compares the skills of the user with the skill dataset using the concept of Levenshtein distance which finds the distance between two strings.**

Keywords*—*k clustering, fuzzy logic, Levenshtein distance, DB-SCAN, data cleaning.

I. Introduction

Career path suggestion refers to the process of candidates looking for a career based on their current skills. The output of the process usually provides the student either with the list of skills he needs to acquire to achieve a career goal or suggests a career path based on his provided goals. The concept of skills is crucial in both the tasks, because it could help pointing out people capabilities even better than in the state of the art approaches, which only focuses on either academic degree or preceding job positions. Both tasks are of the same general problem, namely extracting relevant information from a users profile and matching it with that of the profiles present in the database. Since manual search among a huge amount of data is quite tedious and time consuming, an algorithm which extracts relevant information directly from the profile, classifying it and then suggesting career path would be be helpful in the practical scenario.

II. Design

Algorithms will be used for designing the system. It will involve:

a) Data Collection
b) Data Segregation
c) Data Cleaning
d) Data Appending
e) Clustering using k means and Fuzzy Logic
f) Labelling of Data
g) Extracting skills and String Matching
h) Suggesting Career Path

Following is the brief of methods used in each of the above tasks.

A. Data Collection

We are initially provided with a large amount of data that contains the user profiles of different people from several professions. The data is basically in json format which could easily be accomodated into python. A copy of the data is also converted into csv format which would be required for segregation and cleaning. It is done using online converters.

B. Data Segregation

The data which we are provided contains different types of skills a user possesses in one cell. We need to segregate these skills into different columns so that individual skills could be later used for clustering and fuzzy logic matching.

C. Data Cleaning

A java script code has been written that basically cleans the available data and separates out the different skills, the previous jobs and companies worked under. It also creates different matrices for storing this data.

D. Data Appending

The skills obtained are currently stored as it is in the file. However, directly applying fuzzy logic over this data might not be helpful. This is explained later under [3]. We modify this data by appending the skills with a series of similar characters which could be easily detected and used by the fuzzy logic concept for clustering and string matching. This has been done using a python code that appends the data.

E. Clustering using k-means and fuzzy logic

We now have a data that contains the skills of different users which have been appended with a series of characters. We now apply the fuzzy logic and the k means clustering. Here, we provide the code with an input json file and mention the column name as well as the optimum number of clusters that we want(found using trial and error). The output is a text file that contains the various clusters into which the skills have been classified.

F. Labelling of Data

We now label the clusters according to the profession to which they correctly match with the help the appended characters at the beginning of each skill.

G. Extracting Skills and String Matching

We are now provided with a user profile as an input. We now extract the different skills of the user and store it in a file. These skills are then compared with the different clusters using string matching(fuzzy logic). Different skills are then labelled corresponding to the respective clusters to which they match and are numbered based on majority.

H. Suggesting Career Path

Now the final task of the project is to output either the skills that the person needs to acquire in order to achieve a goal or it provides the career path based on the goal selected by the person. This depends on the user input.

III. Fuzzy Logic

A recommendation system can employ data mining, statistical and predicts correctly. The general output is observed comparing the difference with the clustering output to make the prediction more correctly. The classification based on Fuzzy set theory and Rough Set is applied. Fuzzy set has been used for representing knowledge and decision making. Fuzzy system constitutes of four basic modules which are fuzzification unit, inference module, fuzzy rule based and de-fuzzification unit. The process, data is then chosen through selective sampling process and the classification rules are deduced from the training set.

For example, a simple fuzzy rule is written as: If Degree = Bachelors and Skill1 =C and Skill2 =C++ and Skill3 = ”Java”, then recommend=Programmer.

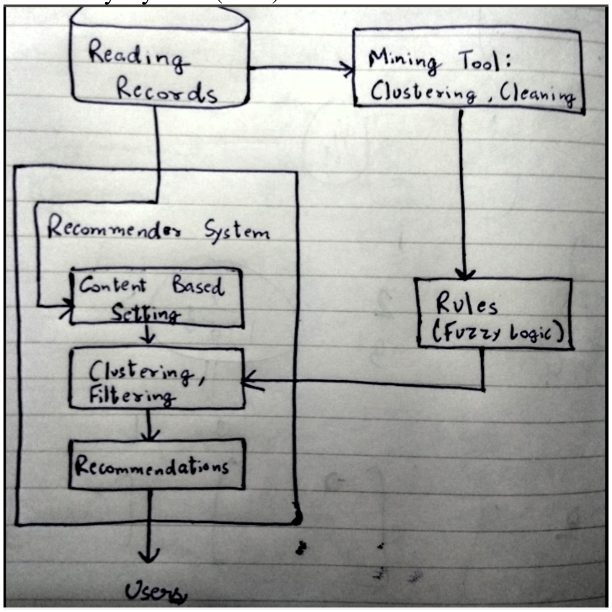
To validate the accuracy of the rules from the training data set, an independent testing data set has been applied. The important thing at this stage is the requirements to gather efficient training set to create classified rules. Rough set is used to find the minimal subsets of attributes and they are applied to data with different categorical values. Various parameters used in the data analysis were identified and the data not found is managed.

IV. Levenshtein Distance

The Levenshtein distance algorithm is a popular method of fuzzy string matching. Levenshtein distance algorithm has implementations in SQL Server also. The term Levenshtein distance between two strings means the number of character replacements or character insert or character deletion required to transform one string to other. Levenshtein distance is also known as Edit Distance. If two strings are equal the Levenshtein distance is 0, zero. A zero value for Levenshtein distance between two string variables in SQL Server means, these two string variables are identical. The higher the value of Levenshtein distance between two varchar or nonvarchar string variables means the strings are more different than each other. As the Levenshtein distance algorithm counts each character edition to transform one string to other, if strings are completely different then the Levenshein distance function will result high values. The name Levenshtein is for the memory of Vladimir Levenshtein who is the developer of this idea.

V. Comparison between Fuzzy Logic and Artificial Neural Network

Fuzzy logic allows making definite decisions based on imprecise or ambiguous data, whereas ANN tries to incorporate human thinking process to solve problems without mathematically modeling them. Even though both of these methods can be used to solve nonlinear problems, and problems that are not properly specified, they are not related. In contrast to Fuzzy logic, ANN tries to apply the thinking process in the human brain to solve problems. Further, ANN includes a learning process that involves learning algorithms and requires training data. But there are hybrid intelligent systems developed using these two methods called Fuzzy Neural Network (FNN) or Neuro-Fuzzy System (NFS).



General Architecture for Recommendation System

VI. Results

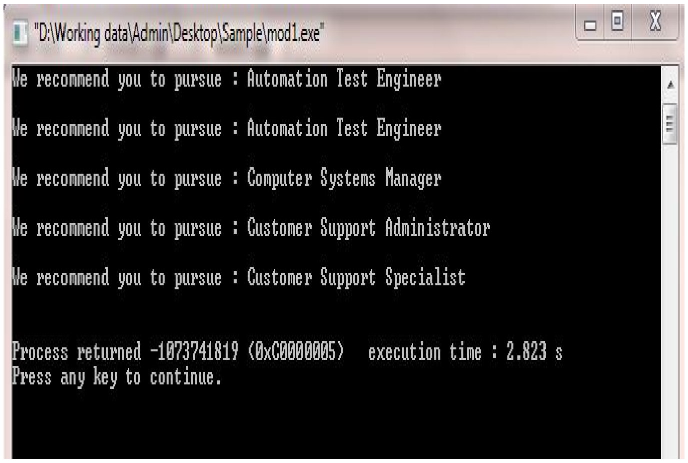
For the given task, I have basically implemented two different methods. The first one is the naive method of string matching in C++ that gives an output accurately. But this method has a drawback of time complexity and redundancy. The second method is using fuzzy logic for string matching and k-means clustering. This reduces the amount of time and also helps to get rid of redundant data. This method was run several times to bring about an optimum value of k which was found to be most accurate at k=25. Below this value, accuracy was quite low. As we increase the value of k, the accuracy increases slightly(till k= 39, after which it decreases) but their is an overlapping of skills among different professions. Thus, I am running my code at k=25.

The output of the dataset of the first implementation accurately predicts the skills as well as the career path. The second implementation classifies the data into different clusters. Fuzzy logic could then be applied on this dataset for string matching.

Time Complexity of the algorithm: $O(n^2)$ (Here n is the number of user profiles previously fetched into the input.)

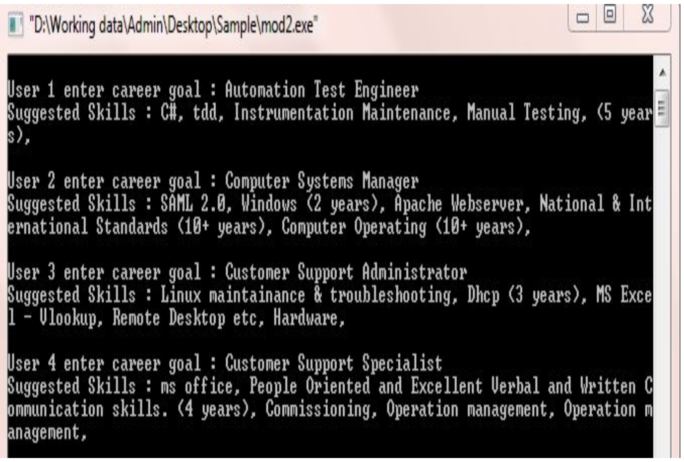
VII. Running Code and Output

Module 1 : The skills of multiple users are extracted from his profile and fetched into a text file. Each line in the text file corresponds to the skills of each individual user. The output suggests the career path that the user should pursue.



```
We recommend you to pursue : Automation Test Engineer

We recommend you to pursue : Automation Test Engineer

We recommend you to pursue : Computer Systems Manager

We recommend you to pursue : Customer Support Administrator

We recommend you to pursue : Customer Support Specialist


Process returned -1073741819 (0xC0000005)   execution time : 2.823 s
Press any key to continue.
```

Module 2 : The user enters his career goal upon prompt. The output shows the skills that he should achieve which he does not previously possess in order to achieve his goal. This basically depends on the number of skills which could be displayed ranging from 0 to 100. Currently, set to 5 skills.



```
User 1 enter career goal : Automation Test Engineer
Suggested Skills : C#, tdd, Instrumentation Maintenance, Manual Testing, (5 year
s),

User 2 enter career goal : Computer Systems Manager
Suggested Skills : SAML 2.0, Windows (2 years), Apache Webserver, National & Int
ernational Standards (10+ years), Computer Operating (10+ years),

User 3 enter career goal : Customer Support Administrator
Suggested Skills : Linux maintainance & troubleshooting, Dhcp (3 years), MS Exce
l - Vlookup, Remote Desktop etc, Hardware,

User 4 enter career goal : Customer Support Specialist
Suggested Skills : ms office, People Oriented and Excellent Verbal and Written C
ommunication skills. (4 years), Commissioning, Operation management, Operation m
anagement,
```

VIII. Conclusion

The above method basically accurately predicts the career path of the user based on his relevant skills. The fuzzy logic reduces the time required compared to the naive string matching algorithm. This method would be helpful in the practical world scenario where a candidate would be suggested skills which he needs to acquire. Currently, it works for an offline implementation and could later be extended for the online part.

IX. Further Research and Improvement

Several other methods could be implemented for various parts of the project which would be helpful and might make the work less tedious and time-consuming.

-The task of Data Cleaning could be performed using the DBSCAN technique which would work both as a crawler as well as for cleaning data.

-Hierarchical clustering could be used instead of k-means clustering which be an accurate measure for classification in this case.

-LSA(Latent Semantic Analysis) technique could be used which could classify the skills which are similar and would also give the count of the occurance of every skill in a given matrix.

-The technique could be made online using advanced machine learning algorithms that could accomodate newer skills which are previously not present in the dataset.

-Larger number of features could be accomodated which would reduce bias that arises from focusing on skills.

Acknowledgment

I would like to thank prof. Ratnik Gandhi to assign me on this project.