AOBD Final Examination - Career Path and Skills recommendation

Kishan Raval, 1401117

***Abstract*—We work on one of the current problems of social networking services, where we try to recommend different skills and career path based on available details of user, profile of the user and other user's data. We work on initially given raw data and try to clean data and convert it in such a form that it can be used easily and efficiently for our later implementation. Different techniques of data cleaning have been employed. We use Natural Language Processing techniques such as Latent Semantic Indexing for finding the relationship between the data and based on which, we try to create clusters of the given data. We approach to solve both the given problem through a generalized algorithm, which can be used for solving both the problems.**

***Keywords*—Latent Semantic Indexing, Vector Space, Data Cleaning, Skills Recommendation, Online Latent Semantic Indexing, Clustering**

I. Introduction

With increase connection to the social networking services, more and more data is available, which can be used for solving many real life problems. We work on one of those problem of recommendation. In general, every person has a goal in his or her life, such as working on a big MNC or being a data science researcher, etc. For achieving such milestone, they try to follow similar people and try to employ similar skills. This helps to ensure that the person will reach to his milestone successfully.

Based on other users data, we try to do the same and help the users by recommending him skills and other things required to achieve a particular goal. This is an open problem, where a good amount of research work is going on for building a prediction engine. People have proposed different ideas to solve such problems.

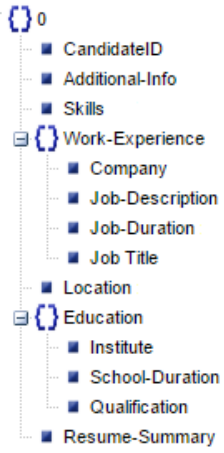
The data which is available to us is in the raw format. Which contains different aspects of person's profile. Different languages have been used by different users. There are many difficulties for getting accurate solution of the problem as different people use different words for similar meaning. Such scenario occurs in cases of position in job, skills and company names. Some people prefer to write *C* as a skill, whereas some prefer to use *C Programming* as a skill. Normally computer cannot understand similarity for such skills. We use Natural Language Processing for making the data better, and try to cluster the data based on the similarity of the words.

Rest of the report is as following: We discuss about the data and the techniques used for cleaning the data in Section 2. In section 3, Model for the problem is discussed with some information of LSA. We discuss other approaches in section 4, future work in section 5 and Bibliography in section 6.

This report is a part of final examination of Algorithms and Optimization for Big Data course at Ahmedabad University.

II. Data

The data that is already given to us priory is in JSON format with different 39 files, separated by major field of the person. Data contains total of 792 users whose different details are given though JSON Object. Each JSON file contains a JSON Array which stores all details of users in an array. The architecture of each user is displayed in the image below.



Some of data that is available is filled in Spanish Language, whereas the other users' details are in English language. Language that is used in resumes are general and different punctuation has been used by different users. There is no much unanimity in terms of data. We try to overcome such problems by cleaning the data.

A. Data Cleaning

Data cleaning is conversion of the given raw data to some structured way such that it can be used easily and efficiently for our algorithms. Our initial approach to clean the data is as following: we first take the JSON files. Then add one parameter to each user's object which shows name of the file. Then, we combine all the files together and create a single JSON array which contains all the users' objects. After which we remove the hierarchy from user's data. This will help us to make our implementation more simpler an we overcome the problem of handling all the files.

After the combine process, we create two parts of our users' profile. One as a training data and other as testing data. We

create 650 training data set and remaining 142 data set have been used for testing purpose.

After the separation of testing and training data, we apply some of the Natural Language Processing techniques to make the data more robust. First, we remove all the special characters. Then, we remove all the stop words from all the objects. After which we convert all the characters of alphabet to lower case. This would help us while matching the string. **Stop Words**: A stop word is a commonly used word (such as "the") that we ignore it for training as well as testing the data. Stop words are unnecessary.

Above processed data is used for our algorithm. Use of data cleaning helps to get better data, which leads to more efficient results.

III. Model

After the cleaning of data, there are still some problems while using the data. One cannot cluster the data without knowing similarity between the words. As discussed earlier. some people use some methodology to write some word, whereas some tend to use some other approach. To analyze a person's profile, use of Natural Language Processing is must.

Our major idea behind the model is to create clusters based on similarity of words. That is, we create clusters such that similar values tend to come closer and create clusters. For finding similarity between different entities, many different approaches have been discussed in literature. Where, one of the frequent solution requires to use Word-Net for finding semantical similarity between two words. But in this scenario, if a person knows C programming, then he is more likely to know C++ or Java Programming. But in Word-Net, the similarity in such cases does not show correct results as they are not similar.

We use Latent Semantic Analysis for finding similarity between words through their occurrence in documents. This approach seems fruitful as it is language independent and can be efficiently in our case. LSA is discussed in the following section.

A. LSA - Latent Semantic Analysis

Latent Semantic Indexing (LSI) is a method for discovering hidden concepts in document data. Each document and term (word) is then expressed as a vector with elements corresponding to these concepts. Our goal is to find relationship between any two different words.

We first create a term-document matrix C by putting documents in columns and words on rows. We put 1 at place (i, j), if the word i is present in document no j. Once we create this matrix, then we apply Singular Value Decomposition to it.

$$C = U * \Sigma * V'$$

Then we lower the rank by taking only first k latent dimensions and ignoring the others.This low rank matrix yields a new representation for each document in the collection. We keep value of k far smaller than the rank of C. Thus, We thus map each row/column (respectively corresponding to a term/document) to a k dimensional vector space.

$$C_k = U_k * \Sigma_k * V_k'$$

B. Online Latent Semantic Analysis

As discussed above, we are finding the similarity matrix by k rank approximation of term-document Matrix C, as new words or document arise, we not need to compute whole Singular Value Decomposition again. Incremental Singular Value Decomposition can be applied to make Latent Semantic Analysis online. This would help very much for updating similarity matrix quickly and efficiently.

C. Using LSA for our problem

We treat each user data as a document and each word is labeled by their type. So, that it can be used while classifying them as skills, company etc. Based on which we apply LSA to our model and find a similarity matrix which gives similarity distance based on words' occurrence in documents. Having similarity near to 1 shows that those two words are related and occur almost in similar documents. This distance can be used for creating clusters.

We create a matrix which contains similarity between all other skills, company and other labeled data. We try to use a generalized approach for both the problems.

Once a user asks for recommendation, we first create labeled words from each each users and separate them, so that they can be treated as separate document. We then apply LSA to these words and get a similarity matrix. This matrix would have all the possible words and their relationship.

Now, for module 1, use whole profile as our database for generating similarity matrix. Using this matrix, when new user's data comes, we split all the words and label them accordingly. Now, most of those words are present in our similarity matrix. For all the words, we find 3 nearest words having required label (in first module, the label should be skill-set). We keep counter of all possible skills and increment respective counter for these 3 labelled words. Similarly, we apply this process for all words present in test string.

in module 2, we simply restrict our input set to user's career goal and other related information. Here also, the relationship matrix remains the same as above case. Only the change is that we will find 3 nearest neighbours using given information (i.e. career goal and other information).

whereas in the module 2, we only take carrier goal and other related information.

D. Another Possible Approach

Data Extraction - Put profile to table form. Each column is converted into a list of unique elements, whose relationship with user data table is defined through a binary relationship matrix.

Career Path Suggestion - Augment relationship matrices of all columns to form feature matrix. Using list and relationship matrix of job titles as labels, train as many classifiers as there are job titles. Use these classifiers to predict jobs for new user.

IV. Algorithms

Algorithm 1 Pre-Processing

Result: Relationship Matrix

1) Generate term-document matrix C
2) Apply SVD on C
3) Get k rank estimation of C
4) Generate Relationship Matrix
5) Store Relationship Matrix

Algorithm 2 Recommendation Algorithm

1) Read Relationship Matrix
2) Initialize counter for each word from 0
3) Read user profile
4) Separate words and apply labels
5) Find nearest 3 values from relationship matrix
6) Increment respective counters for those 3 words
7) Suggest top m words from counter having apropriatte label

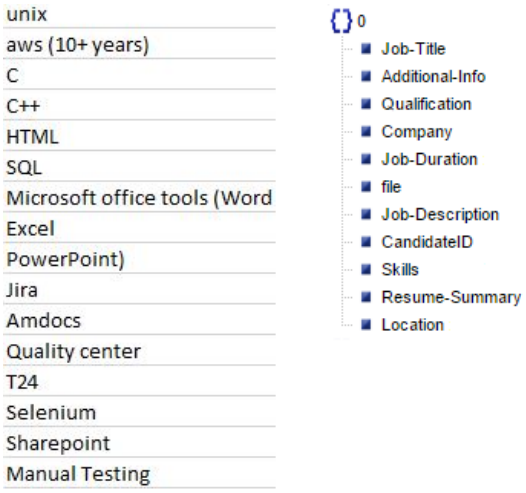
For pre-processing algorithm, most costly step is line number 2. That is applying SVD on Matrix C. Computational Complexity for applying SVD on $m * n$ matrix is $O(m^n + mn^2 + n^3)$. Which comes out to be of $O(n^3)$.

Similarly, in Algorithm 2, most costly step is to find out 3 respective values from relationship matrix. Which has Communication Complexity $O(n^2)$. Hence the overall complexity is $O(n^3 + n^2) = O(n^3)$, where n is number of unique words in all the files.

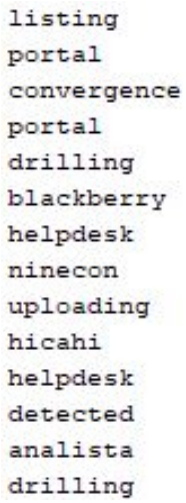
The problem is open problem, and yet accurate and correct solution has not been made. So, proof of correctness would not be possible to make.

V. Results and Interpretation

After the separation of skills and other data through splitting using delimiters, we generate a file which contains all such possible skill, company name and job-titles. A snippet of such output is as following in the left, and the image on the right side side shows the simplified hierarchy during data cleaning:



Following image shows the output of recommendation for one of the test user based on the discussed algorithm.



VI. Future Work

- LSA approach doesn't take similarity through meaning of the words into the consideration. In such cases, Word-Net can be used. We can incorporate both LSA and Word-Net approach to get more better results.
- The result would be more accurate if the data available is more clean. For more better data, website which takes data should suggest inputs and put some restriction while taking the data.
- Implementing Online Version
- Better Data Cleaning such that accurate separation of skills, companies and others is possible.

References

[1] Brand, Matthew. "Fast low-rank modifications of the thin singular value decomposition." Linear algebra and its applications 415.1 (2006): 20-30.

[2] Lou, Yu, Ran Ren, and Yiyang Zhao. A Machine Learning Approach for Future Career Planning.

[3] Malinowski, Jochen, et al. Matching people and jobs: A bilateral recommendation approach. System Sciences, 2006. HICSS06. Proceedings of the 39th Annual Hawaii International Conference on. Vol. 6. IEEE, 2006.

[4] Dumais, Susan T., et al. Using latent semantic analysis to improve access to textual information. Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 1988.

[5] "An Introduction To Latent Semantic Analysis". http://lsa.colorado.edu.