PAWAN KUMAR AMAT

IIT BOMBAY

RESUME PARSING SYSTEM USING TEXT ANALYTICS

1. INTRODUCTION:

This document is Text Analytics can be defined as a set of statistical, linguistic and machine learning techniques which let us analyses textual content in a structured manner so that it can be used for deriving higher quality information from unstructured data. It is also referred as Text Mining.The process involves structuring of text, using it to derive different patterns and evaluating them to get some useful output from it. Various methods of Natural Language Processing (NLP) are involved in Text Analysis like Lexical Analysis, Pattern Recognition, Information Retrieval, Data Mining, Parsing, Sentiment Analysis and Information Extraction. All these techniques help in enabling the computers to understand human language and analyses it like a human.

Resumes are a great source of unstructured data which can be usefully analyzed by the companies to shortlist the right candidate. Various qualities of a candidate can be identified based on the content of his resume. Just like humans, a computer can analyses the resume by finding the right keywords which will categorize the level of every candidate on a scale of 3, Low, Average and High. A learning algorithm can be created which would extract useful keywords from each and every resume which will be analyzed by the system.

Learning technique used can be either Supervised or Unsupervised. Supervised learning would involve training data set for each class of levels defined on the scale. The techniques involved in classification under supervised learning are Support Vector Machine, K-nearestneighborr and Naïve Bayes. Unsupervised learning don't use any training set data rather the use of clustering algorithms like K-means clustering can be used to classify data into various categories or levels.

2. METHODOLOGY:

The model we propose has four steps:

- Collection of resumes.
- Searching for keywords stored in knowledge base in the resume text.
- Fetching new keywords from the resumes to build the knowledge base further.
- Ranking and Categorization of candidate based on a rating score

Figure (1) shows the proposed model and all the steps are explained in subsequent sub sections.

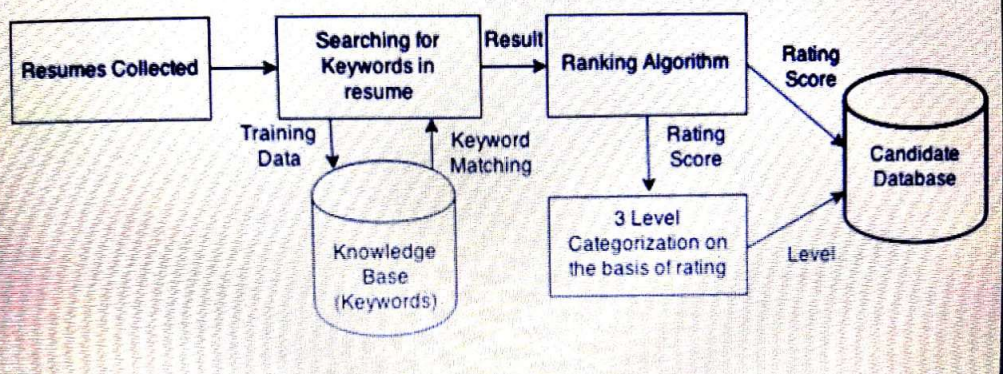

Figure 1: Proposed Model

2.1 RESUME COLLECTION:

This step involves the collection of various resumes uploaded by the candidates. A simple web interface has been designed in our prototype model which will make the user fill a form having the fields which would be required to be filled by the job seeker. The candidates will specify the languages they know along with the projects on which they have worked. This will help the hiring company as they can easily filter out the candidates who do not have the knowledge of the language which is demanded by the company. Most websites use this as their filter method by searching with a keyword. For example, if they want a candidate who knows Java, they can simply search for 'Java' in the resume to filter out candidates who do not know Java. But this technique does not tell the company anything about the proficiency level of the particular candidate in the language he or she knows. There is no way to tell how good the candidate is in Java.

2.2 KEYWORD SEARCHING:

This is one of the most crucial steps of our model. A knowledge base consisting of various keywords is made from the initial training data. The input text which is received needs some pre-processing before it can be used. For this purpose, we use a POS tagger and a chunker which are used to split the text into sentences, which are then analyzed by a syntactic parser which labels all the words with their part of speech information.

The keywords are extracted from the analyzed set of words. The nouns, verbs and adverbs are the part of speech tags which are targeted for extraction while others can be dropped. Extracted words are then compared with the keywords stored in the knowledge base. Every word stored in the knowledge base has a value associated with it. These values are defined based on the importance of the word. Since our prototype deals only with resumes for jobs in IT companies, we have used various keywords which are extracted from the description of projects in which the candidate was involved. A large set of valued keywords can be made and used to rate the candidates on the basis of words extracted from their resumes. The sum of all the keyword values is calculated to obtain a rating score which will be used further to rank the resume and categorize the candidate on the basis of rating.

2.3 ADDITION IN KNOWLEDGE BASE:

While the keywords found in resume text will be matched, the words which are not found in knowledge base are further analyzed and if found relevant, is added to the knowledge base. Since the data from which knowledge extraction has to be done is unstructured, we follow traditional methods of information extraction.

2.4 RANKING AND CATEGORIZATION:

After getting the rating score of the resume, a candidate can be ranked on the basis of his resume's score. This will be useful in comparing two candidates while shortlisting them. Whenever the company searches for a candidate keeping in mind certain requirements, the candidate who is ranked above will be presented to the company first which would be adding to his advantage in cases where the vacancies available may not be high. More important procedure which has to be followed is of categorization. The sentiment analysis categorizes the people's opinions as Positive, Negative or Neutral to derive results. Similar to that, our model would categorize candidates as Low, Average or High on the basis of their resume.

In this way, the efficiency of recruitment process of a company could be significantly improved as better candidates would be picked up without needing to give a lot of time in going through the resumes manually

3. IMPLEMENTATION:

The algorithms for matching of keywords have been implemented on Python facilitated by MySQL connector which fetches data required for matching from the table of Keywords and their associated values which form the Knowledge Base. The algorithm matches the extracted keywords with the keywords present in the knowledge base and stores them in a different list along with their rating values. The rating scores of individual keywords after being added are returned to the candidates table for the purpose of their ranking on the basis of score. Categorization is performed on the basis of rating score of each candidate. The rules for rating and categorization followed in the prototype are as follows –

Rating scale for individual keywords – 1: Low 2: Average 3: High

Rating Score = Sum of ratings of all keywords matched

Categorization on the basis of Rating Score – Below 10: Low 10 to 20: Average Above 20: High

The candidates and the company will use a website based interface to interact. Both of them after getting registered as users shall be added in the database, separate for candidates and companies. The candidate database consists of various fields the candidate would have to fill in while registering which includes the programming languages known and projects in which the candidate has been involved along with its description. These are the crucial fields which will be used to determine the expertise level of the candidate. The fields for expertise level and rating score shall be automatically filled for every candidate once the resume is analyzed.