2012 International Conference on Computer and Software Modeling (ICCSM 2012)
IPCSIT vol. 54 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V54.06

Screener: a System for Extracting Education Related Information From Resumes Using Text Based Information Extraction System

Arunasish Sen[+], Anannya Das, Kuntal Ghosh and Soham Ghosh

Computer Science and Engineering, Institute of Engineering & Management, Salt Lake, Kolkata, India.

Abstract. A simple information extraction system which utilizes the natural paragraph of the document for segmentation, then applies segment specific rules to identified segments for data retrieval.

Keywords: Information Extraction, Resume, Skill Matching.

1. Introduction

Companies often receive thousands of resumes for each job posting and employ dedicated screeners to short list qualified applicants. In this research based project, we present SCREENER, a decision support tool to help these screeners shortlist resumes efficiently. Screener mines resumes to extract salient aspects of candidate profiles like skills, experience in each skill, education details and past experience. Extracted information is presented in the form of facets to aid recruiters in the task of screening. We also employ Lucene Scoring techniques to rank all applicants for a given job opening. In general it is found that automating the task of screening speeds up the process there by making screening task simpler and more efficient.

2. Related Work

There have been a lot of work in the field of Information Extraction and Resume Matching, all of which have suggested several techniques to automate the various aspects of a recruitment process.

For example [4] suggests several techniques of information retrieval from heterogeneous and unstructured Web data. [5] deals with Web User Profiling and employ a classification model to identify relevant documents for a user from the Web and propose a Tree-Structured Conditional Random Fields (TCRF) to extract the profile information from the identified documents.

[6] re-examines the problem at the multi resolution layers of phrase, clause and sentence using dependency and discourse relations. employs clausal relations in 2 ways: 1) to filter noisy dependency paths; and 2) to increase reliability of dependency path extraction.

In [8], a resume is segmented into a consecutive blocks attached with labels indicating the information types. Then in the second pass, the detailed information, such as Name and Address, are identified in certain blocks (e.g. blocks labeled with Personal Information), instead of searching globally in the entire resume. [7] proposes to use Hidden Markov Models to model text at the segment level, in which the extraction process consists of two steps: a segment retrieval step followed by an extraction step. [3] describes a method that uses relevance models to bridge the vocabulary divide between job descriptions and resumes. In their method, related job descriptions are identified by matching a given candidate job description with a database of job descriptions. Then, resumes matching those job descriptions are used to capture vocabulary that is not explicitly mentioned in the job descriptions. [1] introduces PROSPECT: For each job profile, the system automatically ranks candidates based on the similarity between job profile and the resume of the candidate.

[+] Corresponding author. Tel.: + (8377860190);
E-mail address: arunasish.sen@gmail.com

The screener can refine this ranking by adding multiple search and filtering criteria. As new conditions are entered by the screener, Prospect refines the ranking to reflect the new constraints.

3. Our Simpler Approach

After going through lots of contemporary resumes (especially job oriented ones), we realized that most follow a logical sequence of information groups, for e.g.

1. Personal Details.
2. Scholastic Performance
3. Technical knowhow
4. Projects and Research work/ Job experience

Statistically most resumes follow such an order, usually these information groups are separated by natural document separators such as ".", Line feeds, Table structures etc. our primary aim was to exploit these already established trends and segment the whole document into smaller logical groups of sentences.

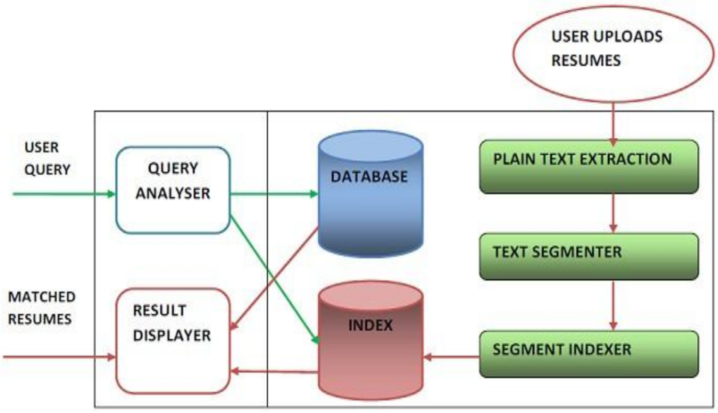


Fig. 1: Architecture of our system

Another striking feature of a resume is that whenever a new information group starts, it is marked by a heading. A heading can be a single word/ a group of words. But these headings also fall within a finite set of words and word combinations.

Now starting with a moderately sized set of such headings we can use a simple matching technique to identify segments which have smaller sizes. Now if we do encounter new words or word combinations which can possibly be new headings, we introduce such word/word combinations into the heading set by a probabilistic method. This helps us recognize segments irrespective of the order in which they are given.

After we recognize the two most important segments i.e. Educational Qualifications and Technical Knowhow/Projects and Internships, we apply specific rule based matching techniques to these segments to extract information.

We then index all the retrieved information along with the user profile in our database (We have developed a website which provides such a feature to its student users). In our website we provide detailed search forms to HR representatives of the company to match candidates to the required job description.

3.1. Plain Text Extraction

Given any format, i.e. PDF, MS-WORD etc., Apache Tika has been used to parse the document and extract the plain text from it. Apache Tika is a robust Apache project which can parse documents in different formats and provide a wide range of output types. The most useful are the plain text and the xml outputs. Here we use the plain text output due to the wide variety of Resume styles.

3.2. Text Segmentation

1. The whole plain text is segmented into smaller segments exploiting the natural paragraph of the document.

2. Segments which are most probable for becoming headings are chosen on the basis of the number of words and the associated probability of it. Also the case in which the segment is written is taken into

account. For e.g. A segment written in uppercase has more probability of being a heading than one written in lower case.

3. Then the chosen segments are matched against a set of probable headings, which have initially been created studying common terms used in resumes.

4. The matched segments and their indexes are stored, which in turn become the boundaries of the recognized segments

3.3. Segment Indexer

After having identified the segments and their corresponding tags, the only thing left is to index them for the purpose of searching. We have achieved this process of indexing using Apache Lucene Libraries. Lucene is Apache's open source project, which is a powerful Java search library that lets you easily add search to any application. Each of these Segments is first analyzed using Lucene's Standard Analyzer, which converts it into a string of tokens. After this using Lucene's IndexWriter each of these analyzed segments are indexed using their corresponding tags as Fieldnames.

3.4. Customised Resume Search Forms

The registered HRs are provided with a customized resume search form in jsp. Using this, the HRs can specify the skill, educational and other requirements for the job. On submitting this they can get their results.

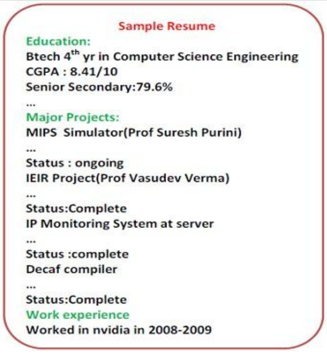
3.5. Query Parser

Firstly, all the submitted search criteria are parsed and tokenized individually using Lucene's QueryParser. Next all these individual queries are clubbed together into a composite query using BooleanQuery.

3.6. Search and Results

Using Lucene's IndexSearcher the composite query is used to search the indexed segment for any match. The results are returned as an array of Document. These results are then displayed to the HRs in the form of a jsp page along with links to the corresponding resumes. The results are displayed in the order using Lucene's Scoring technique. Lucene calculates a score for each document that measures the match between a query and a result.

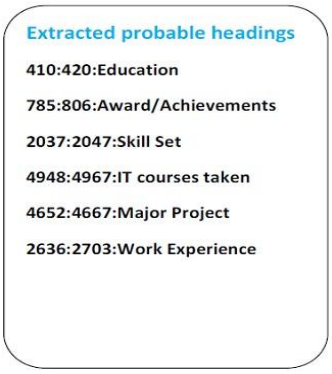
4. Experimental Results



Sample Resume

Education:
Btech 4th yr in Computer Science Engineering
CGPA : 8.41/10
Senior Secondary:79.6%
...
Major Projects:
MIPS Simulator(Prof Suresh Purini)
...
Status : ongoing
IEIR Project(Prof Vasudev Verma)
...
Status:Complete
IP Monitoring System at server
...
Status :complete
Decaf compiler
...
Status:Complete
Work experience
Worked in nvidia in 2008-2009

Fig. 2: Example of a Resume

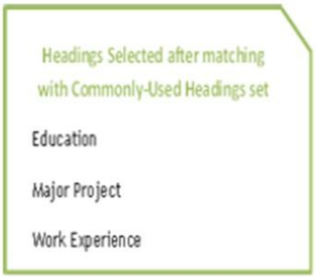


Extracted probable headings

410:420:Education
785:806:Award/Achievements
2037:2047:Skill Set
4948:4967:IT courses taken
4652:4667:Major Project
2636:2703:Work Experience

Fig. 3: Extracted Probable Headings

After applying the weighted heading extraction method (different weights for spaces, hyphens, semicolons, etc.) on the sample resume, the above probable headings are extracted.



Headings Selected after matching with Commonly-Used Headings set

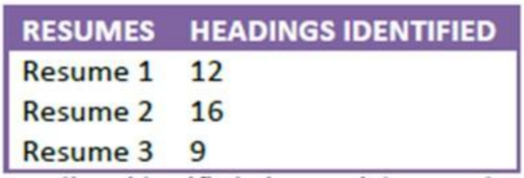
Education

Major Project

Work Experience

Fig. 4:Final Probable Heading

Table 1: Heading indentified given weightage only to spaces

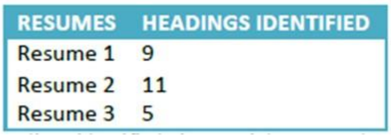


RESUMES	HEADINGS IDENTIFIED
Resume 1	12
Resume 2	16
Resume 3	9

The probable headings are now matched with the existing commonly encountered headings in a resume. After this process finally the above headings are selected for segmentation. Our text segmentation technique identifies headings by assigning probabilities to the number of words found in a line and then accepting or rejecting it as a heading based on that probability.

The above table(Fig: 3) shows the number of headings found in the resumes which were subjected to heading identification technique based on whole number weightage given solely to spaces for identifying number of words in a line. But this resulted in numerous spurious headings.

Table 2: Heading identified given weightage to other delimiters



RESUMES	HEADINGS IDENTIFIED
Resume 1	9
Resume 2	11
Resume 3	5

So to eliminate the false headings we modified the earlier heading identification technique to include fractional weightage provided to delimiters other than space like –hyphen, colon, semicolon, etc. Table 2 above shows the improved results, with almost all the wrongly identified headings eliminated.

5. Future Scope

SCREENER is a big step towards automating the screening process of job recruitment. But there is scope for much improvement to this system in the future. The first and foremost addition to this system can be the creation of a well-defined annotator. Such an annotator should be able to parse the resumes, thereby identifying segments and outputting the parsed document as a structured document in xml, where each segment will be enclosed under tags like <skill>, <education>, <project>, etc.

Secondly, we can integrate a learning model into the system. Introducing a learning model will enable the system to suggest new skill tags based on the submitted resumes and the submitted search criteria. Also a learning system will be able to identify the weights given to particular skills for a particular kind of job description. Thus the system can then boost CVs having experience and projects in those skills. Thirdly, often clever users may try to increase their probability for getting selected to a job by submitting copies of their CVs under different username and email-ids. But the system can be improved, such that it is not fooled by such tricks! We can use certain Structure Analyzers or Table Analyzers to analyze the structures of resumes. If multiple resumes are found to have identical structures then it can be thought to belong to the same user, and hence can be deleted.

6. Conclusion

In this paper we described SCREENER a system that aids in the short listing of candidates for jobs. The key technical component that makes Screening possible is the extraction of various pieces of information from resumes. The better defined segments are identified, the more accurate search results will be. Segmenting the text allows us to overcome the limitations inherent in purely keyword based matching.

It is estimated that by using the SCREENER tool will roughly speed up the process of screening by a factor of 20 as compared to manual screening. This speedup can be attributed mainly to a combination of two factors. Firstly, ranking the candidates by match to the job description and use of the filters provided based on various information extracted from the resume allows the screeners to inspect far fewer resumes to shortlist a given number of candidates. Secondly, providing direct links to matched resumes help to manage resumes efficiently thereby saving a lot of time and labour.

With the addition of future improvements as discussed in the above section the SCREENER system is expected to further save human labour, time and cost incurred during the screening process. We hope that in the future the Resume Matching system will be an indispensable tool in the industry for the purpose of screening candidates during job recruitment process.

7. References

[1] PROSPECT: A system for screening candidates for recruitment Amit Singh, Rose Catherine, Karthik Visweswariah, Vijil Chenthamarakshan, Nanda Kambhatla. In CIKM '10 Proceedings of the 19th ACM international conference on Information and knowledge management, Pages 659-668.

[2] PERSONALIZED INFORMATION RETRIEVAL IN WEB MINING, Subhabrata Sengupta, Sukalyan Goswami.

[3] X. Yi, J. Allan, and W. B. Croft. Matching resumes and jobs based on relevance models. In Proceedings of SIGIR, pages 809- 810, New York, NY, USA, 2007.ACM.

[4] Information Extraction: Methodologies and Applications Jie Tang, Mingcai Hong, Duo Zhang, Bangyong Liang, and Juanzi Li. In the book Emerging Technologies of Text Mining: Techniques and Applications, Chapter 1.

[5] A Combination Approach to Web User Profiling, Jie Tang, Limin Yao, Duo Zhang and Jing Zhang. In the journal of ACM Transactions on Knowledge Discovery from Data Volume 5 Issue 1, Article No. 2.

[6] A Multi-resolution Framework for Information Extraction from Free Text, Mstislav Maslennikov and Tat-Seng Chua Department of Computer Science National University of Singapore. In the Proceedings of the 45th Annual Meeting of the Association of ACL.

[7] Segment-based Hidden Markov Models for Information Extraction Zhenmei Gu, Nick Cercone. In the Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Pages 481-488.

[8] Resume Information Extraction with Cascaded Hybrid Model Kun Yu, Gang Guan, Ming Zhou. In the Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Pages 499-506.

[9] Shallow parsing with conditional random fields, F. Sha and F. Pereira. In Proceedings of NAACL, pages 134{141, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[10] Conditional random fields: Probabilistic models for segmenting and labeling sequence data, J. D. Lafferty, A. McCallum, and F. C. N. Pereira. In Proceedings of ICML, pages 282-289, San Francisco, CA, USA,2001. Morgan Kaufmann Publishers Inc.