PBECV: a fast resume extracting framework based on writing style recognition

ABSTRACT

In the information age, companies receive thousands of resumes from job seekers everyday. Most of resumes are wrote in different format, including font size, font color and cells. As a result, it's difficult to structure these data with a general extracting method. In this paper, we propose PBECV to extract the resume information from text file without format information. Our appoach consider the writing style of each resume as the latent pattern, which help to segment resume text into different blocks and easy to parse. The experimental results on the real world data-set of millions of resumes show that our approach can reach the performance of algorithms that trained with the structure information and the proposed approach's algorithm complexity is O(n).

General Terms

Algorithms, Design, Experimentation

Keywords

resume information extraction, structure resume data

1. INTRODUCTION

In the information age, head-hunting companies collect millions of resumes to occupy more market share. However, most of resumes are not wrote with the standard format or follow some special template file. In order to improve the success rate of recommand person to fit the requirement of employee, those resumes should be parsed exactly and detailed. This helps human resources to easily and quickly search for the right candidate. The challenge is how to analysis the different kinds of resume to get the content.

However, resumes are easier to structure than other texts, such as news. Different people have different writing style about personal resume, but the content of these work all around the same topic, their personal information, which contains contacts, educations, work experimences etc. As a result of this, resumes can be segmented into servel groups, which is one of the basic ideas to solve the problem. In other words, resumes share the document-level hierarchical contextual structure [2].

There are two main methods to deal with resumes in the practical engineering. One is mainly for resumes follows the same format, from some big recruiting platform like monster, linkedin. This kind of resumes can be parsed through special template file or regex rules. The advantage is very high accuracy, but not suitable for normal resume. Another one is keyword extraction from resume. This method use search technology to query keyword from resume to check whether it match the require. The advantage is suitable for every resume, but losing the accuracy. In the real application, resumes are mixed with those follow some template and normal ones. The system's ability to deal with resumes should be known to control the the robustness of the software.

In this paper, we want to build a baseline of paring resume for the system. Because when we meet a resume follows some template file, the specify parsing file will parse it exactly. We have made an effort to propose an easy implement and effective solution based on pattern recognition and the decision tree.

The rese of paper is organized as follows. In Section 2, we disscuss the related work. In Section 3, we explain our approach. In Section 4, experimental results are presented and analysised. Conclusion and future work are provided in Section 5.

2. RELATED WORKS

Jsoup[1] and POI[2] can be used to parse resumes that follows some template file. Jsoup is a Java library for working with real-world HTML. It provides a very convenient API for extracting and manipulating data, using the best of DOM, CSS, and jquery-like methods. It also implements the HTML5 specification, and parses HTML to the same DOM as modern browsers do. The Apache POI is a useful Java library for working with Office file, based on the Open XML standards which proposed by Microsoft company. These two open source tools help to extract resumes that follows some template file.

In [4], a cascaded information extraction framework was de-

[1]http://jsoup.org
[2]http://poi.apache.org

Table 1: heuristic rules for cleaning data

heuristic rules	operation
multiple continuous blank	short
value pair	short
begin with date pair	split
begin with part of date	merge
begin with block key words	split
begin with colon	merge
short text end with colon	merge

signed to support automatic resume management and routing. The first pass is used to segment the resume into cnsecutive blocks with labels indicating the information types. Then detailed information, such as Name and Address, are identified in certain blocks without searching globally in the whole resume.

In [3], a system that aids in the shortlisting of candidates for jobs was designed. The part of parsing resume combines three technologies. Table analysis is used to detect the type of values in table. CRF model is used to segment the resume text into different blocks. Content Recognizer mines the named entities salient to candidate profile.

In [1] and [2], only the specific data is extracted to filter the resume streams. Both of them are aim to accelerate the efficiency of search candicates for the job.

3. OUR APPROACH

In this section, we explain the details of our approach. The process can be devided into three part. First, the origin resume file is converted into text format no matter what the origin format is. Apache Tika[3] provides some api to support this operation. And some heuristic rules is created to regularize each line so as to remove some noises like continuous blank. Second, writing style is used to identify the appropriate block of each resume. Third, name entities are matched to the candicate profile based on the information statistics of the content of all the resumes in the data set.

3.1 Prepared processing

From the figure 1, it's clear to know that the raw resume text is not suitable to process directly. There are a lot of noises among the lines in each text file, such as continues blank, wrong newline, the necessary space missing. All these noised should be cleaned before the main part of extracting resume information module. We suppose the distribution of resume accordance with normal distribution, that most people will not cause serious errors on text format. Since we have millions of resume, it's easy to statistics the most common structure of sentence, especially the sentence begin with date or number. According to these rules, three kinds of operation are made ,shown in Table 1.

Merge means this line should be merged with the next line.

Split means this line should be split into two lines.

Short means the blanks in this line should be remove.

[3]http://tika.apache.org/

Table 2: Algorithm to extract the resume

Input: L: Set of n lines of each resume; Output: R: resume with structured data
1. for line in L if line match heuristic rules do operation 2. for line in L find pattern of line match the pattern to others if match record the block else continue return all blocks 3. for block in B match the name entities 4. return resume

We also defined three kinds of line type to facilitate the follow-up work. These three types provide the basic sentence structure which is helpful to identify the writing style.

Simple means this line is short text or contains few blanks.

$KeyValue$ means this line follows the key and value structure, with colon signal.

Complex means this line is a long text, which contains more than one signal.

3.2 Writing style recognition

After cleanning up the noise of raw lines, we need to devided these lines to blocks such as basic information, education, work experiments and so on. It's not hard to find that there are some latent pattern within each block. Entities are introduced into pattern recongined, however in this applicantion sence simple name entity is enough. For each line, we only detect whether this line contains date entity or some basic entity like school name, job position, company name. With the help of name entities, each line can be transfered to the pattern mode. Through loop the whole lines set, the latent block pattern can be found, which is the basic to extract the details content.

3.3 Name entities Match

Avoiding labled the data by human too much, we statist the whole set to find some useful information which may be the latent value for the profile. First, each resume is processed as the Prepared processing section 3.1 descripted. Second, those lines with key-value structure are considered to be the candicate attribute. Third, after removing some noises

4. EXPERIMENTS

In order to verify our approach, we did the experiment with one million resumes which provided by a commercial headhunting company. These resumes are contains different industry field people and different source. We use precision, recall and F value to evaluate this approach. Cause the dataset is huge, the standard precision and recall can not be compute without the label data. We involved a score function, which treat each field of the block as a unit, then

Table 3: The evaluation of results

block name	prcision	recall	F value
name	0.952	0.919	0.935
email	0.992	0.714	0.830
other basic information	0.923	0.75	0.823
education	0.912	0.701	0.792
work experimences	0.873	0.720	0.789
self evaluation	0.897	0.796	0.843

compute the total score of each resume. We supposed each resume text has basic information, education, work experiences and self evaluation things. This hyperspace is not match the real data, but as a result of the huge volume it does not matter to get the basic overview.

$$F - 1 = \frac{P + R}{2PR}$$

From the results, we can get an overview about the resume dataset that not all the resume is valid. Because person name has a strong feature, it's easy to detect and regconized. The email also has an obvious feature, which is contrustred by servel characters and only one @ singal. Other basic information concludes how many years he/she worked, address, sex, id number, phone num. Most resume contains the basic information but not sure each of them, which inflect the recall. Not each resume text is really a personal resume, this is noticed after the experient, for example, the presudo resume is a description about someone from a head-hunter or just an unfinished resume. This is the mainly reason of the low rate of education and work experience, whose block need carefully detected.

Compared to other approachs published in related works, our method is easy to implement and also gain a considerable result. Without too many human label data is another advantage.

5. CONCULSION AND FUTURE

In this paper, we propose an approach to extract the details from unstructual resume text. This work helps the human resource mangagement system clear that what's the baseline about resume parsing. The most contribution of our work is extract the details of resume without too much labeled data with simple model.

In the future, we will try to introduce our approach to English resumes and try to auto-generate the e-recuriting domain knowledge base in order to gain better extractor performance.

6. ADDITIONAL AUTHORS

7. REFERENCES

[1] S. K. Kopparapu. Automatic extraction of usable information from unstructured resumes to aid search. In *Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on*, volume 1, pages 99–103. IEEE, 2010.

[2] S. Maheshwari, A. Sainani, and P. K. Reddy. An approach to extract special skills to improve the performance of resume selection. In *Databases in Networked Information Systems*, pages 256–273. Springer, 2010.

[3] A. Singh, C. Rose, K. Visweswariah, V. Chenthamarakshan, and N. Kambhatla. Prospect: A system for screening candidates for recruitment. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 659–668, New York, NY, USA, 2010. ACM.

[4] K. Yu, G. Guan, and M. Zhou. Resume information extraction with cascaded hybrid model. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 499–506, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.