Research Article
A Two-Step Resume Information Extraction Algorithm

Jie Chen,[1] **Chunxia Zhang,**[2] **and Zhendong Niu** (iD)[1,3,4]

[1] *School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China*
[2] *School of Software, Beijing Institute of Technology, Beijing 100081, China*
[3] *Beijing Engineering Research Center of Massive Language Information Processing and Cloud Computing Application, Beijing Institute of Technology, Beijing 100081, China*
[4] *School of Computing & Information, University of Pittsburgh, Pittsburgh, PA 15260, USA*

Correspondence should be addressed to Zhendong Niu; zniu@bit.edu.cn

With the rapid growth of Internet-based recruiting, there are a great number of personal resumes among recruiting systems. To gain more attention from the recruiters, most resumes are written in diverse formats, including varying font size, font colour, and table cells. However, the diversity of format is harmful to data mining, such as resume information extraction, automatic job matching, and candidates ranking. Supervised methods and rule-based methods have been proposed to extract facts from resumes, but they strongly rely on hierarchical structure information and large amounts of labelled data, which are hard to collect in reality. In this paper, we propose a two-step resume information extraction approach. In the first step, raw text of resume is identified as different resume blocks. To achieve the goal, we design a novel feature, Writing Style, to model sentence syntax information. Besides word index and punctuation index, word lexical attribute and prediction results of classifiers are included in Writing Style. In the second step, multiple classifiers are employed to identify different attributes of fact information in resumes. Experimental results on a real-world dataset show that the algorithm is feasible and effective.

1. Introduction

The Internet-based recruiting platforms play an important role in the recruitment channel [1] with the rapid growth of the Internet. Nowadays, almost every company or department posts its job requirements on various online recruiting platforms. There are more than one thousand job requirements uploaded per minute in Monster.com (http://www.monster.com/). Online recruiting is immensely useful for saving time to both employers and employees. It allows the job seekers to submit their resumes to many employees at the same time without travelling to the office and it also saves employees' time to organize a job fair. Meanwhile, there are also many portals acting as a third-party service between job seekers and company human resources, so that lots of resumes are collected by these portals. For instance, LinkedIn.com (http://www.linkedin.com) has collected more than 300 million personal resumes uploaded by users. Because of the increasing amount of data, how to effectively analyze each resume is a severe problem, which attracted the attention of researchers.

In the real world, job seekers usually use diverse resume text formats and various typesetting to gain more attention. Lots of resumes are not written in accordance with a standard format or a specific template file. This phenomenon means that the structure of resume data has a great deal of uncertainty. It decreases the success rate of recommending recruits who meet most of the employer's requirements and take up too much time of human resources to do job matching. In order to improve the efficiency of job matching, exploring an effective method to match jobs and candidates is important and necessary. In addition, the resume mining is also helpful to do user modeling of the recruitment platform [2].

According to its usage scenarios, personal resume data has some properties as follows. First, job seekers write their resumes with varying typesetting, but most of the resumes involve general text blocks, such as personal information, contacts, educations, and work experiences. Second, personal

resumes share the document-level hierarchical contextual structure [3], which is shared among different items in the corresponding text block of each resume. The main reason for this phenomenon is that items in a text block sharing the similar hierarchical information can make the whole resume more comfortable for readers. Above all, a resume can be segmented into several text blocks; then facts can be identified based on the specific hierarchical contextual information.

In recent years, many e-recruitment tools are developed for resume information extraction. Although basic theories and processing methods for web data extraction exist, most of the tools for e-recruitment still suffer from text processing and candidate matching with the job requirements [4]. There are three main extraction approaches to deal with resumes in previous research, including keyword search based method, rule-based method, and semantic-based method. Since the details of resume are hard to extract, it is an alternative way to achieve the goal of job matching with keywords search approach [3, 5]. Inspired by the way of extracting the news web page [6–10], several rule-based extraction approaches [11–13] treat the resume text as a web page and then extract detailed facts based on the DOM tree structure. For the last kind of methods, researchers treat the resume extracting task as a semantic-based entity extraction problem. Some researchers use sequence labelling process [14–17] or text classification methods [18] to predict the tags for segments of each line. However, most of these methods strongly rely on hierarchical structure information in resume text and large amounts of labelled data. In reality, learning of text extraction models often relies on data that are labelled/annotated by a human expert. Moreover, the more expertise and time the labelling process requires, the more costly it is to label the data. In addition, there may be constraints on how many data instances one expert can feasibly label. More details about these three kinds of methods will be introduced in Section 2.

This paper focuses on the extraction algorithm that is proposed for resume facts extraction. Our contributions are as follows. (1) We propose a novel two-step information extraction algorithm. (2) A new sentence syntax information, Writing Style, for each line in the resume is designed in this paper. It is used to get semistructured data for identifying the detailed fact information. (3) We give an empirical verification of the effectiveness of the proposed extraction algorithm.

The remainder of this paper is organized as follows. The related work about other methods on this problem is reported in Section 2. In Section 3, the detailed processing steps and the data pipeline used in our algorithm are described. In Section 4, we introduce what is the Writing Style and how to process and identify it. In Section 5, experimental results are presented and analyzed. Conclusions and future work are provided in Section 6.

2. Related Works

To find relevant literature on e-recruiting and data mining from resumes, we summarized the methods of previous research and carefully selected the articles that are most relevant to our research. According to the adopted features, there are three kinds of popular methods about resume information extraction in previous research, which can be described as follows.

The first group of methods takes keywords retrieval in consideration. In [3, 5], only the specific data are selected to filter resume streams. Both of them aim to accelerate the efficiency of search candidates for the job. Some of the important queries were created to filter the resume set so that they can help to improve the work efficiency of the staff. Although these kinds of methods are easy to implement, the raw text content brings too many noises into the index, leading to low precision and unsatisfactory ranking results.

The second group of methods based on the DOM (Document Object Model) tree structure, in which tags are internal nodes and the detailed text, hyperlink, or images, are leaf nodes. Ji et al. [19] proposed a tag tree algorithm, in which they detected and removed the shared part among web pages with the same template, and then the main text is retained. Also some other methods extract the knowledge with Regex rules from the HTML pages. Jsoup (http://jsoup.org) and Apache POI (http://poi.apache.org) can be used to parse resumes that follow some specific template file. Jsoup is a Java library for working with real-world HTML. It provides a very convenient application interface for extracting and manipulating data based on the DOM structure. Moreover, POI is a useful Java library for working with Office file, focused on extracting the file content. It is easy to create a specific program to extract the information from those resumes which follow the specific template file. In [20], the system performed the information extraction by annotating texts using XML tags to identify elements such as name, street, city, province, and email. These methods based on template file with DOM tree are limited by human efforts. Since it is impossible to know how many groups of resumes follow the same template, these methods are hard to scale out in big data.

The third group of methods treats extracting knowledge as a semantic-based entity extraction task. In [17], a cascaded information extraction framework was designed to support automatic resume management and routing. The first pass is used to segment the resume into consecutive 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 [16], a system that aids in the shortlisting of candidates for jobs was designed. Their system integrates table analyzer, CRF predictor, and content recognizer into the whole part of parsing resumes. The layout of table cells in the file was considered by the table analyzer, and the CRF predictor was used to predict the label of the text sequence; then the content recognizer was used to mine named entities in the candidate resume text. In [14], they proposed an ontology-driven information parsing system that was designed to operate on millions of resumes to convert their structured format for the purpose of expert finding through the semantic web approach. In [15], researchers presented EXPERT, an intelligent tool for screening candidates for recruitment using ontology mapping. EXPERT has three phases in screening

Raw Resume Text	Semi-structured Output	Structured Output
Allen Yan(Male) Phone: 1580133XXXX Email: allenXXXX@gmail.com *Education Background* 2004/9/1–2008/6/1, Tsinghua University, Bachelor Computer Science *Work Experience* 2013/6/1–Now, Tencent, Project Manager Product search and development department I am in charge of the department and do research in the user generate content. 2012/1/1–2013/6/1, Baidu, Project Manger Business Search Department My duty is control the progress of our team and the quality of the program.	Simple: Name(Gener) KeyValue: Phone: Number KeyValue: Email: String Simple: String Complex: Date-Date, {University, Company}, Degree Simple: Major Simple: String Complex: Date-Date, {Company, Software}, Position Simple: Department Simple: String Complex: Date-Date, {Company}, Position Simple: Department Simple: String . . .	Name: Allen Yan Gender: Male Phone: 1580133XXXX Education Background: Edu1: Date: 2004/9/1–2008/6/1 University: Tsinghua University Degree: Bachelor Major: Computer Science Work1: Date: 2013/6/1–Now Company: Tencent Position: Project Manager Work2: Date: 2012/1/1–2013/6/1 Company: Baidu Position: Project Manger . . .

Origin File(Doc, Docx, Pdf, Html etc.) → Raw text → Identify the composition of each line → Adaptive Segmentation — Text Block Classification

Semi-structured data → Multiple category text classifier → Structured data — Resume Facts Identification
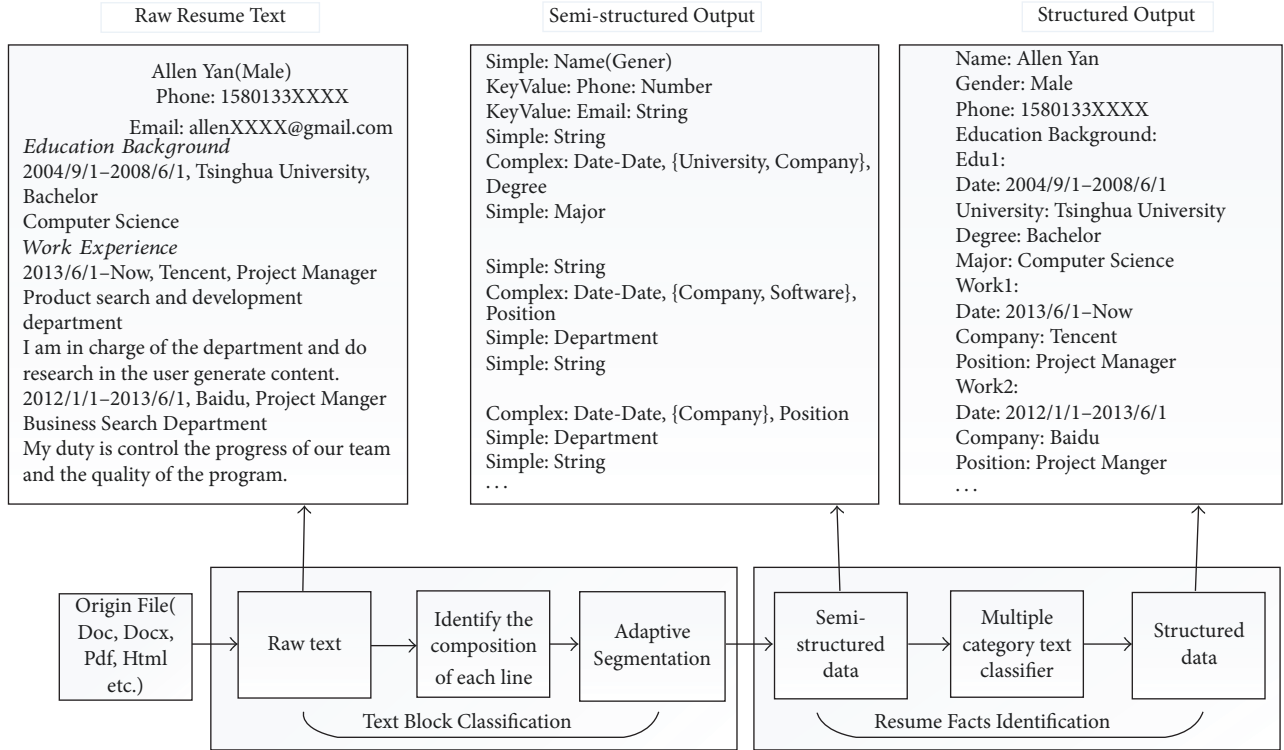
FIGURE 1: The pipeline of our algorithm and an example.

candidates for recruitment. In the first phase, the system collects candidates' resumes and constructs ontology document for the features of the candidates. Job requirements are represented as ontology in the second phase. And in the third phase, EXPERT maps the job requirement ontology onto the candidate ontology document and retrieves the eligible candidates. Tang et al. [21] also employ CRF as the tagging model. DOM tree structure is used to infer the hierarchical structure; then content features, pattern features, and term features are combined to train the model. Uldis Bojars introduced ResumeRDF (http://rdfs.org/resume-rdf) ontology to model resume. Further, he extended FOAF (http://xmlns.com/foaf/spec) to support more description of resume. Chen et al. [18] proposed a framework based on text classifiers, which are trained with data corpus from the Internet instead of manual annotation. However, these works are limited by file formats and the huge human efforts, which cost in labeling the sequence data for ontology, CRF, or semantic web model.

3. Two-Step Resume Information Extraction Algorithm

In this section, we first introduce the inputs, outputs, and the architecture in our algorithm. Then, the pipeline of our algorithm is explained and an example is used to make it clear. The details of each part are shown after the pipeline.

3.1. Inputs & Outputs. In this algorithm, we focus on extracting information without hierarchical structure information.

The definition of *Input* and *Output* in our algorithm is as follows.

3.1.1. Input. Given a set of resumes, with different file types, such as doc, Docx, and pdf, those files will be processed by Tika (http://tika.apache.org) to get the raw text, where table layouts, font type, and font colors will be removed.

3.1.2. Output. The structured output resume data should contain the facts about a person written in the resume file. Moreover, most of the personal facts should be stored in key-value pairs.

The architecture of our two-step resume information extraction algorithm and an example are shown in Figure 1. During the data pretreatment process, raw text content is extracted from the origin resume files, and some prepared processing work is used to clean data noises brought with Tika, including remove images, background colour, and watermark. In the first step, lines of the text are segmented into semistructure phase based on text block classification, which will be introduced in the next section. A multiple-class classifier is used to predict the label for each phrase, such as university, date, and number. We design a new feature, Writing Style, to model the syntax of each line. The Writing Style feature of each line is constructed with word index, punctuation index, word lexical attribute, and prediction results of classifiers. More details of Writing Style will be introduced in Section 4. In the second step, Writing Style is used to identify the appropriate block of each semistructured text and identify different items of the same module. Meanwhile,

name entities are matched to the candidates' profile based on the information statistics and phrase classifier.

3.2. Text Block Classification. Text block classification is an important step in this extracting process because the follow-up work is based on it. Most people like to write a caption at the beginning of each block, such as "Education," "Project Experiments," and "Interests and Hobbies." In intuition, raw text of each resume can be separated into different blocks based on these words. However, there are lots of synonyms and word combinations which bring a big challenge to build a dictionary to do keyword match.

We defined three types of different lines to facilitate the follow-up work as follows:

(i) *Simple* means this line is a short text and may contain few blanks.

(ii) *KeyValue* means this line follows the key and value structure, with comma punctuation.

(iii) *Complex* means this line is a long text, which contains more than one punctuation.

These three types provide the basic sentence structure which is helpful to classify the block and further identify the block with Writing Style.

Each sentence with the *Simple* tag is treated as one word to compute its frequency in the whole dataset since most caption always occupies the whole line. A probability formula, used to find the potential caption words, is defined as

$$p\left(\text{caption}_i\right) = \frac{\text{Count}_{\text{sentence}_i}}{\text{Count}_{\text{resume}}}, \qquad (1)$$

where sentence$_i$ is the count of sentence i appearing in the dataset and Count$_{\text{resume}}$ is the total number of the resume dataset. After removing stop words and some text modifier, the synonyms were easy to find and group into the different cluster with different block's title.

3.3. Resume Facts Identification. Instead of labelling too much data, a lot of statistical work needs to be done for collecting the name entity candidate keys, often shown in the text with key-value pair as the attribute name. The similarity of the entity can help to do attribute cluster; then they can be labelled with the standard attribute name. For different blocks of the resume, we used the different corpus to train the text classifier. And in our algorithm, the naive Bayes classifier is used to do text classification.

The detailed process is as follows. First, each resume is processed as Section 4.2 described. Second, those lines with key-value structure are considered to be the candidate attribute. Third, after removing some noises in the text, cosine similarity is computed based on *TFIDF*, and the *Kmeans* cluster algorithm shows the attribute cluster. Fourth, these clusters are matched to the profile attribute. Algorithm 1 summarizes the proposed text-free extraction method.

```
(1)  for each line ∈ lines do
(2)     if line match heuristic rules then
(3)        do operation
(4)     end if
(5)  end for
(6)  for each line ∈ lines do
(7)     find pattern of line
(8)     match the pattern to others
(9)     if match then
(10)       record the block
(11)    else
(12)       continue
(13)    end if
(14) end for
(15) record all blocks
(16) for each block ∈ blocks do
(17)    match the name entities attribute
(18)    if match then
(19)       save the name entities
(20)    end if
(21) end for
```

ALGORITHM 1: Extracting facts from raw resume text.

4. Writing Style

In this section, we will focus on the Writing Style feature, which is designed to model the syntactic information of each sentence. Firstly we will give the definition of Writing Style. Secondly, how to process the raw resume text in practice is described in detail, and three kinds of operations are proposed to aid segmenting the text. Thirdly, how to identify each sentence's Writing Style is introduced.

4.1. Definition of Writing Style. For each resume, there is some hidden syntax information about the structure, which is different from the surface information, such as font size, font colour, and cells. Further, within the scenario of Chinese resume, spaces are used to separate different tags, which is a very clear Writing Style feature. In other words, everyone who writes his/her resume will follow its local format, such as "2005–2010 [company name] [job position]," "[company name] [job position] [working time]," and "[university] [major] [degree] [time range]." This local format forms the writer's Writing Style, and the writer will follow the same format during the same block, which is a kind of hidden syntax information. Inspired by this, the Writing Style is defined as follows and the samples of Writing Style are shown in the middle of Figure 1.

4.1.1. Writing Style. The Writing Style includes the prediction of classification, the location, and the punctuation of phrases in the line. It combines the predict results of text classifier and the literal information. In other words, the Writing Style is a kind of syntax feature about the structure of a line in a resume.

4.2. Writing Style Processing. Due to the capabilities of Tika, the raw text is not in accordance with the original layout.

TABLE 1: Heuristic rules for cleaning data.

Heuristic rules	Operation
Multiple continuous blanks	Trim
Value pair	Trim
Begin with date pair	Split
Begin with part of date	Merge
Begin with block key words	Split
Begin with comma	Merge
Short text ends with comma	Merge

There are a lot of noise among the lines in each text file, such as continues blank, wrong newline, and the necessary space missing.

Based on enough data observation about the raw text, three kinds of operation are defined as follows:

(i) *Merge* means this line should be merged with the next line.

(ii) *Split* means this line should be split into two lines.

(iii) *Trim* means the blanks in this line should be removed.

Data cleansing rules are made for different lines in Table 1.

4.3. Writing Style Recognition. After cleaning up the noise of raw text, lines of resume text are prepared to identify the Writing Style. A lot of name entities are collected, such as university name, company name, job positions, and department, which are easy to extract from different media on the Internet. Some sample data, translated into English, are as shown in Table 2. The data used to train these classifiers is easier to obtain from the Internet. For example, university names can be easily obtained from the ministry of education's official website, and job position names can be extracted from the portal of Internet-based recruiting platforms. These data are used to train a basic multiclass classifier, including university name, job position name, department name, ID number, address, and date.

With the help of classifier, each phrase in the line can gain a probability distribution on a different class. The position of the phrase, the symbols, and the probability are combined to be the Writing Style of a line. For each line, we only detect whether this line contains date entity or some basic entity like university name, job position, company name, or date. Each line can be transferred into entities pattern mode, as shown in the middle of Figure 1.

5. Experiments

For evaluating the performance of our algorithm, we tested it on a real-world dataset. Because the rule-based method will gain a full score of precision, we will not do experiments about it. Moreover, the generalization ability of rule-based method is very bad. In other words, the experiments in this paper focus on the free text extraction method, which is worth evaluation and research. Comparative analysis is carried out on the text block classification and detailed

knowledge extraction on three modules including the education experience, work experience, and basic information for each resume. We compared the proposed framework with PROSPECT [16] and CHM [17], which also treat extracting as a nature language processing task as introduced in Section 2.

5.1. Dataset and Measures

5.1.1. Dataset. In order to verify the proposed algorithm, an experiment was conducted involving fifteen thousand resumes in Chinese which provided by Kanzhun.com (http://www.kanzhun.com), the biggest company review website in China, similar to Glassdoor (http://www.glassdoor.com). All the resumes are well labelled for information extraction, including beginning position, end position, and attribute name of each tag. These resumes involve multiple industries, and most of them are created by job seekers that it is hard to find a common template to match. About ten thousand resumes were in Microsoft Word format and five thousand in Html format. Apache Tika is used to parse these documents in Word format and extract the whole text without any visual format information. Jsoup was used to parse those documents in HTML format and both the HTML tags and scripts were removed.

5.1.2. Measures. Standard precision, recall, and *F*-measure are used to evaluate experimental results. Precision and recall metrics are adopted from the IR research community. Precision reports how well a system can identify information from a resume and recall reports what a system actually tries to extract. Thus, these two metrics can be seen as a measure of completeness and correctness. In order to define them formally, we define that #key denote the total number of attributes expected to be filled about each resume and let #correct(#incorrect) be the number of correctly (incorrectly) filled attributes in the extraction results. *F*-measure is used as a weighted harmonic mean of precision and recall. These three metrics are defined as follows:

$$\text{precision} = \frac{\#correct}{\#correct + \#incorrect}$$

$$\text{recall} = \frac{\#correct}{\#key}$$

$$F = \frac{(\beta^2 + 1) * \text{precision} * \text{recall}}{(\beta^2 * \text{precision}) + \text{recall}},$$

(2)

where β is set as 1 in our experiments and *F*-1 is used to represent *F* measure.

The overlap criteria [16] (match if > 90% overlap) was also used in our experiment to match ground truth with extracted data.

5.2. Evaluation of Text Block Classification. We extract four main blocks from each resume, basic information, education, work experiences, and self-evaluation things. As a result of that, the extracting algorithms focusing on the field of resume are independent of the test corpus; we used the experiment results from their paper directly. Moreover, only two blocks'

TABLE 2: Data Samples of multiclass classifier.

University name	Company name	Job position name
Tsinghua	Baidu	Java Developer
Peking University	DiDi	Sales
Beijing Institute of Technology	San Kuai Technology	System Administrator
Shandong University	Tencent	Test Engineer
North China Electric Power University	Alibaba	Supply Chain Solution Architect

TABLE 3: Education block classification.

	PROSPECT	CHM	Our approach
Precision	0.94	0.71	0.912
Recall	0.902	0.77	0.701
F-1	**0.921**	0.73	0.792

TABLE 4: Work experiences block classification.

	PROSPECT	Our approach
Precision	0.790	0.873
Recall	0.780	0.720
F-1	0.785	**0.789**

TABLE 5: Basic info block classification.

	CHM	Our approach
Precision	0.868	0.923
Recall	0.769	0.75
F-1	0.804	**0.823**

TABLE 6: The results about education extraction.

Module name	Precision	Recall	F-1
School name	0.950	0.853	0.898
Degree	0.947	0.821	0.879
Major	0.796	0.891	0.840
Graduation time	0.764	0.877	0.817

TABLE 7: The results about work experience extraction.

Module name	Precision	Recall	F-1
Company name	0.914	0.811	0.859
Job title	0.831	0.849	0.840
Description	0.948	0.790	0.861
Work time	0.813	0.878	0.844

data were provided by these two models; we compared them, respectively.

Table 3 shows the results about education block classification, Table 4 shows the results about work experiences block, and Table 5 shows the results about basic info block. From the results, we can get an overview of the resume dataset that most resumes can be detected by our approach and the precision and the recall are acceptable. The PROSPECT's precision and recall are higher than our free text extraction method in the education block classification; the main reason is the difference of application scene. The application scene of PROSPECT focuses on the resumes of software engineers with IT professionals, but there is no qualified professional in our application scene. Resumes of IT professionals always cover a limited major, which help to increase the precision and recall of the classifier. This is a kind of classification advantage for them. When facing the work experience block, this kind of advantage is very small, which explain the reason for low precision and recall.

5.3. Evaluation of Resume Facts Identification

5.3.1. Extraction Results on Education Experience Module. Table 6 shows the extraction results about education module. Since the school name and degree are relatively fixed, the precision of them is high. However, the text format for these is more than others; for example, the school name has abbreviations and some other names which are known by people. The name of majors may be different in different universities, and this is hard to prepare the prior data. The text format of graduation time is totally out of control because there are so many possibilities, such as 1985-11-04, 1989/4/2, and 15/01/12.

5.3.2. Extraction Results on Work Experiences Module. The results of detailed knowledge extracted from work experiments are shown in Table 7. Most job seekers write the full name of employer company or its famous website, which provided enough information to match this word to be a company name, while the name of job title is hard to identify because it depends on the industry of company.

A different company may use different job titles for the same position at the same level, which is harmful to training text classifier. The description is made up of sentences around the details of work in the ex-employer company. It is easy to find these description sentences because the successive lines are always full of kinds of symbols and the length is longer than others. But the beginning and ending of description are hard to determine; the line next to the end of description always is the beginning of next work experience item. The reason for low precision about work time is the same as graduation time.

5.3.3. Extraction Results on Basic Information Module. Table 8 shows the details of specific values in the basic information. From the table, we know that most resumes contain the name and email information, which is consistent with our intuition since job seekers must leave their contact

TABLE 8: The results about detailed basic information extraction.

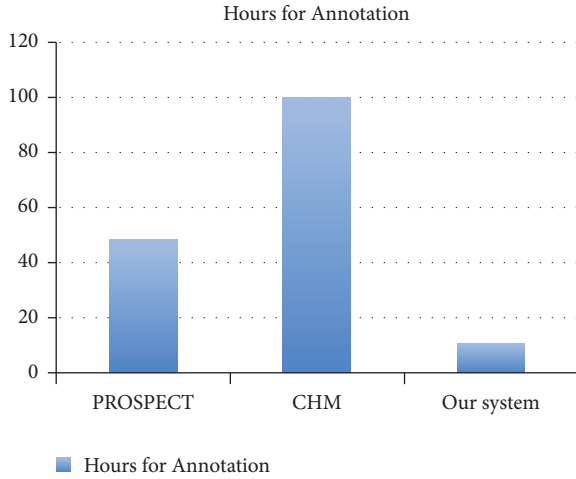| Item name	Precision	Recall	F-1
Name	0.952	0.919	0.935
Email	0.992	0.714	0.830
Other basic information	0.923	0.75	0.823
Self-evaluation	0.897	0.796	0.843

FIGURE 2: Hours for annotation.

information in their resumes. Also, person name has obvious characteristics so that it is easy to detect and recognize which explain the high precision and recall. The email also has an obvious feature, which is constructed by several characters and only one @ symbol. Other basic information concludes how many years he/she worked, address, sex, id number, and phone number. These kinds of information have different features such as length and spelling habits, which reflect the recall.

5.4. Evaluation of Efficiency. In order to compare the effectiveness of different algorithms, the cost of human labour investment should be considered. We compared the time used in preparing training data for each algorithm. The results are shown in Figure 2.

In PROSPECT, they annotated around 110 English resumes using GATE [22] and collected 7200 annotations from 3 volunteers. We followed the guiding document to label one resume with GATE, which takes about 5 minutes on average and annotates that each one instance takes 20 seconds on average. In other words, PROSPECT will cost about nearly 49.2 hours in total.

In CHM, they annotated 1200 Chinese resumes. On average, each resume takes 5 minutes to label all the attribute and value. As a result, CHM take about 100 hours to annotate all the training instances.

In our experiments, we need to prepare the dataset for each classifier. The big difference with PROSPECT is that the data can be collected from different websites. For example, we collect nearly 2300 university names in the Ministry of

Education website which is official and well prepared. We prepared seven classifiers for different blocks, including person name block, phone number, address, university, job position, certificate, and technology skill. Each training instance takes 1.5 hours on average; that is, the whole training dataset cost us 10.5 hours in total.

5.4.1. Discussion. From the experimental results above, the values of precision and recall are competitive to those complex machine learning methods. Compared to other approaches published in related works, our method is easy to implement and also gain a considerable result. Further, our approach can omit lots of manual annotation work which can save a lot of cost and time.

6. Conclusion and Future Work

In this paper, knowledge facts are extracted from resumes with different text formats and file types in our algorithm. The algorithm consists of two processing step, which are text block identification and name entity recognition. This work aims to improve the accuracy of extracting information from personal resumes. It is useful to build resume repository for head-hunters and companies focus on Internet-based recruiting. In the second processing step, we propose a Writing Style to distinguish different lines. Compared to those extracting algorithms, based on either HMM or CRF, our approach does not need too much manually annotated training set, which can save lots of human efforts and time. Meanwhile, experimental results on real-world dataset indicate that the precision and recall of free text resume extracting are better than them.

We hope to continue this work in the future and to explore social relations among people, similar to community discovery. As our future works, we will apply multiple label predication [23] and coreference resolution [24, 25] to improve the recall rate of name entity classification, and other classification algorithms will be tested.

Conflicts of Interest

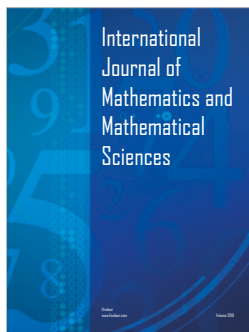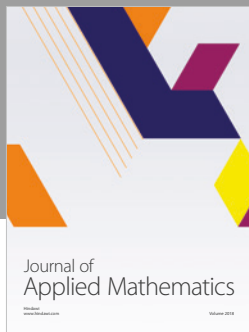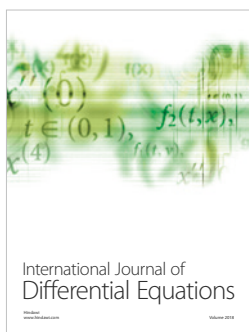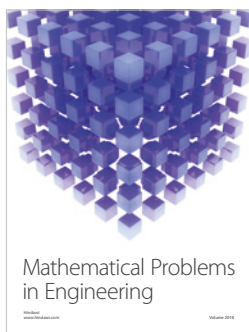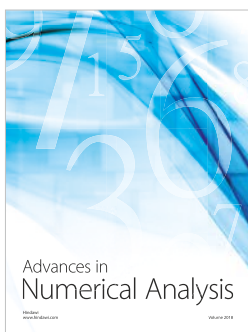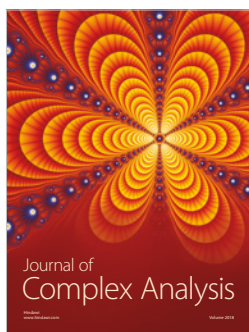The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] S. T. Al-Otaibi and M. Ykhlef, "A survey of job recommender systems," *International Journal of Physical Sciences*, vol. 7, no. 29, pp. 5127–5142, 2012.

[2] J. K. Tarus, Z. Niu, and A. Yousif, "A hybrid knowledge-based recommender system for e-learning based on ontology

and sequential pattern mining," *Future Generation Computer Systems*, vol. 72, pp. 37–48, 2017.

[3] 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*, vol. 5999 of *Lecture Notes in Computer Science*, pp. 256–273, Springer, Berlin, Germany, 2010.

[4] C. Bizer, R. Heese, M. Mochol, R. Oldakowski, R. Tolksdorf, and R. Eckstein, "The impact of semantic web technologies on job recruitment processes," in *Wirtschaftsinformatik 2005: eEconomy, eGovernment, eSociety*, pp. 1367–1383, 2005.

[5] S. K. Kopparapu, "Automatic extraction of usable information from unstructured resumes to aid search," in *Proceedings of the 1st IEEE International Conference on Progress in Informatics and Computing, (PIC '10)*, vol. 1, pp. 99–103, China, December 2010.

[6] Z. Bar-Yossef and S. Rajagopalan, "Template detection via data mining and its applications," in *Proceedings of the 11th International Conference on World Wide Web, (WWW '02)*, pp. 580–591, ACM, NY, USA, May 2002.

[7] S. Lin, J. Chen, and Z. Niu, "Combining a segmentation-like approach and a density-based approach in content extraction," *Tsinghua Science and Technology*, vol. 17, no. 3, Article ID 6216755, pp. 256–264, 2012.

[8] S. Lin and J. Ho, "Discovering informative content blocks from Web documents," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, (KDD '02)*, pp. 588–593, Edmonton, Alberta, Canada, July 2002.

[9] B. Sluban and M. Grcar, "Url tree: Efficient unsupervised content extraction from streams of web documents," in *Proceedings of the 22nd ACM International Conference on Information Knowledge Management (IKM '13)*, pp. 2267–2272, ACM, NY, USA, 2013.

[10] X. Song, J. Liu, Y. Cao, C. Lin, and H. Hon, "Automatic extraction of web data records containing user-generated content," in *Proceedings of the the 19th ACM international conference on Information and Knowledge Management, (CIKM '10)*, pp. 39–48, ACM, Toronto, ON, Canada, October 2010.

[11] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm, "DOM-based Content Extraction of HTML Documents," Defense Technical Information Center, 2003.

[12] P. M. Joshi and S. Liu, "Web document text and images extraction using DOM analysis and natural language processing," in *Proceedings of the 9th ACM Symposium on Document Engineering, (DocEng '09)*, pp. 218–221, Germany, September 2009.

[13] F. Sun, D. Song, and L. Liao, "DOM based content extraction via text density," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR '11)*, pp. 245–254, China, July 2011.

[14] D. Celik and A. Elci, "An ontology-based information extraction approach for resumes," in *Proceedings of the 7th International Pervasive Computing and the Networked World, (ICPCA/SWS '12)*, pp. 165–179, Springer-Verlag, Berlin, Germany, 2012.

[15] V. S. Kumaran and A. Sankar, "Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT)," *International Journal of Metadata, Semantics and Ontologies*, vol. 8, no. 1, pp. 56–64, 2013.

[16] 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)*, pp. 659–668, Toronto, ON, Canada, October 2010.

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

[18] J. Chen, Z. Niu, and H. Fu, "A novel knowledge extraction framework for resumes based on text classifier," in *Proceedings of the 16th International Conference on Web-Age Information Management*, vol. 9098 of *Lecture Notes in Computer Science*, pp. 540–543, Springer International Publishing, Berlin, Germany, 2015.

[19] X. Ji, J. Zeng, S. Zhang, and C. Wu, "Tag tree template for Web information and schema extraction," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8492–8498, 2010.

[20] F. Ciravegna and A. Lavelli, "Learning Pinocchio: Adaptive information extraction for real world applications," *Natural Language Engineering*, vol. 10, no. 2, pp. 145–165, 2004.

[21] J. Tang, L. Yao, D. Zhang, and J. Zhang, "A Combination Approach to Web User Profiling," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 1, pp. 1–44, 2010.

[22] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "Gate: an architecture for development of robust hlt applications," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, (ACL '02)*, pp. 168–175, Philadelphia, Pennsylvania, July 2002.

[23] W. Liu and I. W. Tsang, "Large margin metric learning for multi-label prediction," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2800–2806, Austin, Texas, USA, 2015.

[24] E. Bengtson and D. Roth, "Understanding the value of features for coreference resolution," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, (EMNLP '08)*, pp. 294–303, Honolulu, Hawaii, October 2008.

[25] P. Fragkou, "Applying named entity recognition and co-reference resolution for segmenting English texts," *Progress in Artificial Intelligence*, vol. 6, no. 4, pp. 325–346, 2017.