Information Extraction from Resume Documents in PDF Format

Jiaze Chen, Liangcai Gao*, Zhi Tang; Institute of Computer Science and Technology of Peking University; Beijing, P.R. China

Abstract

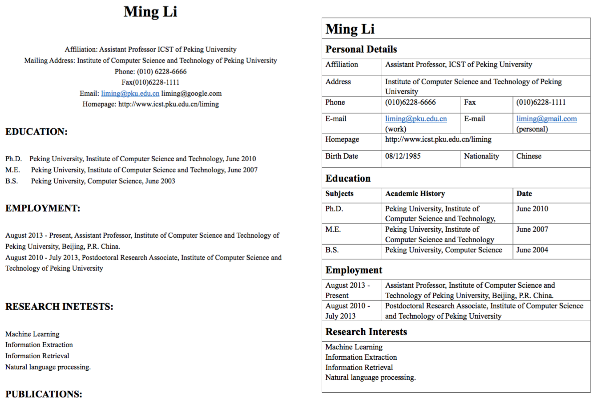
Now more and more people release their resumes through the Internet, and PDF is a wide adopted format of resume documents which contain lots of valuable information for recruitment, personal profile mining,etc. However, only a few studies have been down in this direction. Therefore, this paper focuses on the task–information extraction from resume documents in PDF format, and proposes a hierarchical extraction method. At first, this method segments a page into blocks according to heuristic rules. And then each block is classified by a Conditional Random Field (CRF) model. To take advantage of the structure and layout information of PDF documents, the classification model employs two kinds of features:content-based features and layout-based features which are parsed from PDF documents. The experimental results show that the effectiveness of the proposed method. Especially, the layout-based features are proved to be very useful for the task, improving more than 20 percent of the average F1-score in the experiments.

Introduction

With the development of the Internet, more and more people publish their resumes on the Internet. To extracting the information existing in resumes, a database of job seekers can be efficiently and automatically built to help the human resources department in big companies or headhunters. In addition, researcher profiles could be constructed by extracting the information of researchers' resumes.

However, extracting information from resume documents with high precision and recall is still a challenge, because resumes often differ in data formats, layout formats and writing styles. Furthermore, the existing researches on the challenge mainly focus on resume documents in plain text. Those researches get the text from resume documents at first, then extract metadata from plain text. But plain text loses format and layout information of the document which is useful for document understanding and information extraction. Other researchers focus on resume documents in HTML format such as researchers' homepage or introduction pages. However resume documents in PDF format often contains more detailed information than resume documents in HTML format. Therefore, this paper mainly processes the resume documents in PDF format.

A resume usually includes a summary of personal, educational and academic backgrounds, as well as working and research experiences, publications, awards and honors, research interests and other details. Normally a resume document has a hierarchical logical structure. At first the document can be separated into high-level blocks. Each high-level block contains the relevant detailed block in low-level. For example, a resume can be segmented into personal information, education, publications, etc. Personal information block contains name, affiliation, address, e-mail, fax and homepage as detailed information blocks in low-level. Education block contains the university major, date and degree information of PhD, master and bachelor as detailed information block in low-level.



Ming Li

Affiliation: Assistant Professor ICST of Peking University
Mailing Address: Institute of Computer Science and Technology of Peking University
Phone: (010) 6228-6666
Fax(010)6228-1111
Email: liming@pku.edu.cn liming@google.com
Homepage: http://www.icst.pku.edu.cn/liming

EDUCATION:

Ph.D. Peking University, Institute of Computer Science and Technology, June 2010
M.E. Peking University, Institute of Computer Science and Technology, June 2007
B.S. Peking University, Computer Science, June 2003

EMPLOYMENT:

August 2013 - Present, Assistant Professor, Institute of Computer Science and Technology of Peking University, Beijing, P.R. China.
August 2010 - July 2013, Postdoctoral Research Associate, Institute of Computer Science and Technology of Peking University

RESEARCH INETESTS:

Machine Learning
Information Extraction
Information Retrieval
Natural language processing.

PUBLICATIONS:

(a) A list-style resume

Ming Li			
Personal Details			
Affiliation	Assistant Professor, ICST of Peking University		
Address	Institute of Computer Science and Technology of Peking University		
Phone	(010)6228-6666	Fax	(010)6228-1111
E-mail	liming@pku.edu.cn (work)	E-mail	liming@gmail.com (personal)
Homepage	http://www.icst.pku.edu.cn/liming		
Birth Date	08/12/1985	Nationality	Chinese
Education			
Subjects	**Academic History**		**Date**
Ph.D.	Peking University, Institute of Computer Science and Technology,		June 2010
M.E.	Peking University, Institute of Computer Science and Technology		June 2007
B.S.	Peking University, Computer Science		June 2004
Employment			
August 2013 - Present	Assistant Professor, Institute of Computer Science and Technology of Peking University, Beijing, P.R. China.		
August 2010 - July 2013	Postdoctoral Research Associate, Institute of Computer Science and Technology of Peking University		
Research Interests			
Machine Learning Information Extraction Information Retrieval Natural language processing.			

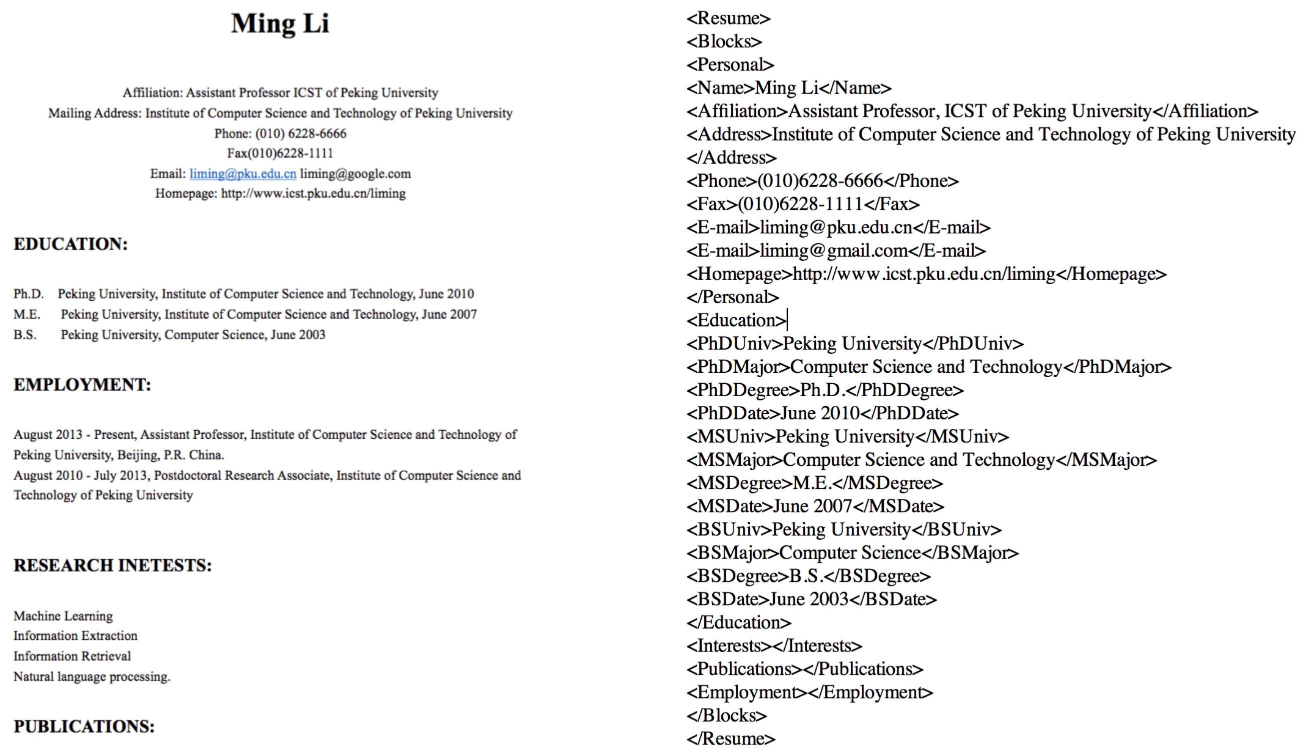
(b) A table-style resume

Figure 1. *Two typical styles for resume documents*

As figure 1 shows, there are two typical and common styles of PDF resumes. The first one is table-style and the second one is list-style. A table-style resume is a structured table for the entire document. But for list-style resume, information is arranged in highly formatted regions and is represented as a list item in each region. Extracting information from a structured table document can be transformed into a table metadata extraction problem. D Pinto[5] and other researchers have done a lot of research on this problem and already achieves a impressive result. Besides, in the Internet, there are fewer table-style resume documents. Therefore, this paper mainly deals with list-style resume documents.

In this paper, we propose a two-layer model for information extraction from resume documents in PDF format. To take advantage of the layout and the content information of PDF documents. This paper integrates various kinds of features of both content and layout. In the first layer, the resume documents are segmented into blocking by using heuristic rules. Then a well-trained classification model is employed to classify each block into pre-defined categories. In the second layer, the detailed information extraction task is regarded as a sequence labeling problem. A Conditional Random Fields (CRF) is utilized to finish sequence labeling.

*glc@pku.edu.cn (Liangcai Gao is the corresponding author)



Ming Li

Affiliation: Assistant Professor ICST of Peking University
Mailing Address: Institute of Computer Science and Technology of Peking University
Phone: (010) 6228-6666
Fax(010)6228-1111
Email: liming@pku.edu.cn liming@google.com
Homepage: http://www.icst.pku.edu.cn/liming

EDUCATION:

Ph.D. Peking University, Institute of Computer Science and Technology, June 2010
M.E. Peking University, Institute of Computer Science and Technology, June 2007
B.S. Peking University, Computer Science, June 2003

EMPLOYMENT:

August 2013 - Present, Assistant Professor, Institute of Computer Science and Technology of Peking University, Beijing, P.R. China.
August 2010 - July 2013, Postdoctoral Research Associate, Institute of Computer Science and Technology of Peking University

RESEARCH INETESTS:

Machine Learning
Information Extraction
Information Retrieval
Natural language processing.

PUBLICATIONS:

(a) a resume document

```
<Resume>
<Blocks>
<Personal>
<Name>Ming Li</Name>
<Affiliation>Assistant Professor, ICST of Peking University</Affiliation>
<Address>Institute of Computer Science and Technology of Peking University
</Address>
<Phone>(010)6228-6666</Phone>
<Fax>(010)6228-1111</Fax>
<E-mail>liming@pku.edu.cn</E-mail>
<E-mail>liming@gmail.com</E-mail>
<Homepage>http://www.icst.pku.edu.cn/liming</Homepage>
</Personal>
<Education>
<PhDUniv>Peking University</PhDUniv>
<PhDMajor>Computer Science and Technology</PhDMajor>
<PhDDegree>Ph.D.</PhDDegree>
<PhDDate>June 2010</PhDDate>
<MSUniv>Peking University</MSUniv>
<MSMajor>Computer Science and Technology</MSMajor>
<MSDegree>M.E.</MSDegree>
<MSDate>June 2007</MSDate>
<BSUniv>Peking University</BSUniv>
<BSMajor>Computer Science</BSMajor>
<BSDegree>B.S.</BSDegree>
<BSDate>June 2003</BSDate>
</Education>
<Interests></Interests>
<Publications></Publications>
<Employment></Employment>
</Blocks>
</Resume>
```

(b) the metadata of the resume

Figure 2. *the metadata of the resume document*

The remaining part of this paper is structured as follows. Section 2 introduces the related work. Section 3 presents the proposed approach. Experimental results are shown in Section 4. In section 5, we provide an analysis and discussion about the experimental results. Section 6 presents the conclusion and the future work.

Related Work

Several research efforts have been made to resume information extraction tasks.

Previous work of resume information extraction is mainly for resume documents in plain text. Cravegna and Lavelli[7] use $(LP)^2$[24], an algorithm for adaptive information extraction, to learn information extraction rules for resume documents. The information they extract includes Name, Street, City, Province, Email, Telephone, Fax and Zip code. Their work is based on plain text and only a flat structure is used, other than a hierarchical structure. Jun yu and et al. [3] proposes a cascaded hybrid model for resume information extraction. They segment the resume into blocks at first, then extract detailed information in blocks. Their two-layer model is similar to our work. But they only use content features such as and named entity, while we use not only content features but also PDF document specific layout features.

Other research work focuses on resume document in HTML format. Limin Yao and et al. [4] proposes a unified method to researcher profiling. They employ a Conditional Random Fields model to extract information from researchers' homepage or introduction page. The information they extract includes photo, position, affiliation, phone and other data about personal information or education background. Differ from their work, our work mainly focuses on resume documents in PDF format. The media of HTML documents are much different from PDF documents. DOM (Document Object Model) tree structure is used to organize the HTML document. It makes document understanding and information extracting in HTML document easier than in the PDF document. In addition, the metadata in the PDF document is contained in highly formatted regions, while most of the information is presented in natural language of resume documents in HTML format.

Many information extraction models have been proposed. Support Vector Machines, [11], Hidden Markov Model[9], Maximum Entropy Markov Model[10], Conditional Random Fields[2] are widely used extraction algorithms. The SVM is a classification model and the others are generative models. Among these models, Conditional Random Fields is a state-of-art probabilistic model for information extraction. It is first proposed by [2] for segmenting and labeling sequence data and then widely used in many tasks such as named entity recognition[23], shallow parsing[21], relational learning[22], scholarly document information extraction[1] and table extraction[5].

Information extraction from resume documents in PDF format is a typical semi-structure information extraction problem. Information extraction has been studied for many years and is mostly used for unstructured text. In that case, linguistic knowledge(lexicons and grammars) can be useful. But recently, layout information is learned to be useful for semi-structure information extraction problem. Several researchers have been focused on the layout structure and logical structure of PDF documents. Anewierden[17] recovers the logical structure of the technical manuals in PDF. Kushmerick[18] uses text classification model to extract information in business cards.

Some research work focuses on scholarly document information extraction. Their tasks are similar to us. The main focus of their work is on the header (title, authors, institutions, venue, etc.) and citation metadata extraction. In [12], Han, Hui, et al. treat document metadata extraction problem as a classification task. They use Support Vector Machines to extract metadata of documents. The features they use are mainly word specific features. In [1], Peng et al. apply Conditional Random Fields Model to extract various fields from the header part of researcher papers. In [19], IG Councill and et al. parses the reference strings that appear in scholarly documents' reference section and uses a knowledge-based approach and a CRF model. E Cortez and et al.[20] conduct the same work using an unsupervised method. Resumes are similar to scholarly documents that most of the information is contained in highly formatted regions. But there are also differences. Metadata in research paper only appears in the fixed section, for example header information in the header section and citation information in the last section. But the information on resumes may be contained in any places of documents. Besides, papers often use standard document layouts (e.g. Latex Template) while resume documents differ in forms and layouts.

Our Approach

Problem Definition

As the figure 3 shows, we can use a hierarchical structure to describe the layout and logical structure of the resume document. A two-layer model is employed in our paper. The first layer describes high-level blocks. The second layer describes relevant detailed information in low-level blocks. High-level blocks and detailed information are defined in table 1.

As figure 4 shows, there are four steps in our experiments: 1) PreProcessing 2) Block Segmenting and Classifying 3) Detailed Information Extraction 4) Post Processing.

In the preprocessing step, the resume documents in PDF format are parsed to the text lines at first. Then each text line is segmented into separated words with recording style and position information such as font name, font size and bounding box.

In step 2, the document is segmented into blocks via content and layout clues of the document. A well-trained Support Vector Machines (SVM) model is used to classify each block into a predefined category. Then layout blocks are mapped to the logical structure of the resume document.

In the tagging step, the detailed information extraction problem is transferred into a sequence tagging problem. Only the education block and the personal block are selected for further extraction instead of tagging in the entire document. A Conditional Random Fields model is used to determine the most likely corresponding sequence of tags.

In the post processing step, heuristic rules are used to merge tokens or split tokens to prevent from mis-tagging.

Preprocessing

PDF document provides information about all characters used in the text with their positions and attributes, such as font size and font name. Therefore, this paper parses the resume document to a character list. Characters are sorted by their y/x positions. Characters with nearby y positions are merged into one line later. Considering the characteristic of the resume documents in PDF format, a simplified word segmentation algorithm based on the word gaps and punctuations is employed.

All characters are traversed in each line from left to right. Neighboring characters are merged by some rules. Rules are described as follows.

1. If the successive character has a different font size or font style, then the character is separated from the successive character and if the gap between two characters is larger than a fixed threshold, they are separated either.

2. If the current character is one of the punctuations(e.g. double quotes, parentheses, colons, commas), the current character is separated from the preceding character and the following character. But there are some exceptions. When the current character is a dash, the character is not separated from its preceding character and following character. When the current character is a period, if the preceding word is a single capital letter(often be a part of the name) or in a handcrafted list (often a word like a Ph.D., Dr., Prof. etc.), it is not separated either.

After merging characters into tokens, the layout information of the token such as the main font size, main font style, bounding box and line positions is calculated.

Block Segmenting and Classifying

Segmenting

As we have defined the hierarchical logical structure of resume documents previously, the document can be segmented into blocks at first. Each block is mapped into the high-level block in the logical structure.

We use a simple method based on a recursive bottom-up algorithm. Blank spaces between lines are sorted by their size. Little blocks are merged into larger blocks vertically or horizontally. A threshold is set to determine the size of the block and end the recursion.

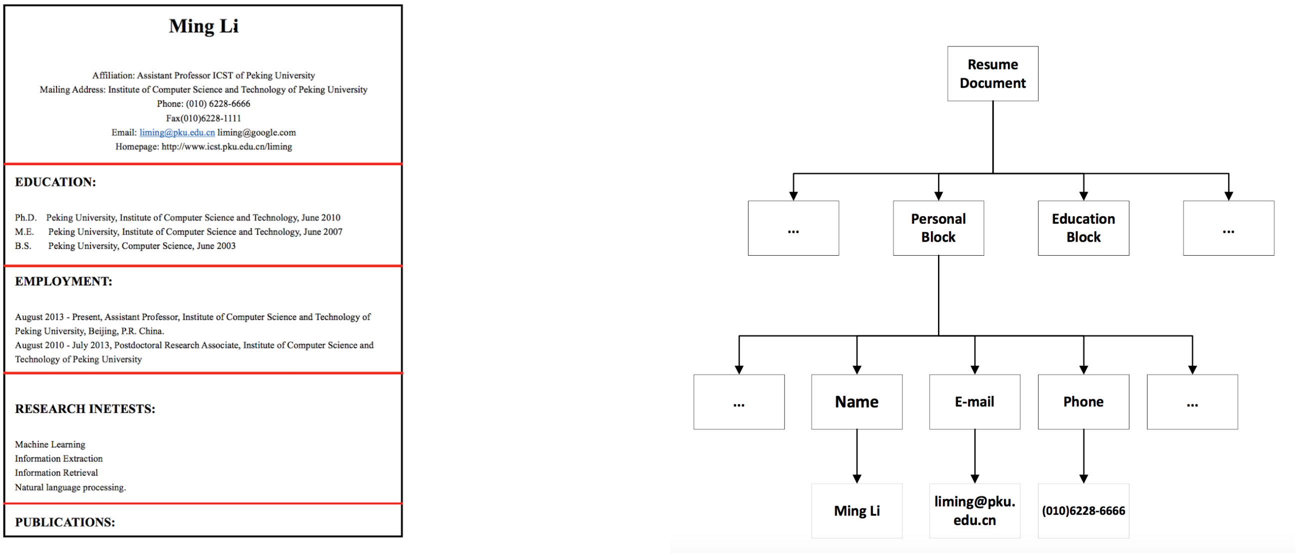
Besides, we add some constraints on our implementation of page segmentation. Some heuristic rules based on document layout information (e.g. font size, font style, blank space, alignment) are used to prevent from mis-segmenting and over-segmenting. For example, a text object of the large bold font maybe the title of the blocks, then the current line is not merged with the preceding block.

Because the block segmenting results will be used in the detailed information extraction steps, we need to explore the affections of different block sizes. This paper tests different experimental results by using both tiny block and large block. It will be discussed in the later section.

Block classification using a Support Vector Machine model

Deciding which category a block belongs to is a classical multi classification problem. In this step, a SVM classifier is trained on the labeled datasets. Then, the trained model is used to predict which category a new coming block belongs to.

Support Vector Machine[6] is well known for its generalization performance. A support vector machine can construct a hyperplane in a high dimensional space which can be considered as the optimal classification hyperplane. In a binary classification



Ming Li

Affiliation: Assistant Professor ICST of Peking University
Mailing Address: Institute of Computer Science and Technology of Peking University
Phone: (010) 6228-6666
Fax(010)6228-1111
Email: liming@pku.edu.cn liming@google.com
Homepage: http://www.icst.pku.edu.cn/liming

EDUCATION:

Ph.D. Peking University, Institute of Computer Science and Technology, June 2010
M.E. Peking University, Institute of Computer Science and Technology, June 2007
B.S. Peking University, Computer Science, June 2003

EMPLOYMENT:

August 2013 - Present, Assistant Professor, Institute of Computer Science and Technology of Peking University, Beijing, P.R. China.
August 2010 - July 2013, Postdoctoral Research Associate, Institute of Computer Science and Technology of Peking University

RESEARCH INETESTS:

Machine Learning
Information Extraction
Information Retrieval
Natural language processing.

PUBLICATIONS:

(a) layout hierarchical structure

(b) logical hierarchical structure

Figure 3. *The layout and logical hierarchical structure of the resume document*

Pre-defined Categories

Items	Categories
Blocks	Personal, Education, Publications, Employments, Honors, Interests, Projects
Personal Block Attributes	Name, Phone, E-mail, Address, Affiliation, Website, Fax
Education Block Attributes	Phduniv, Phdmajor, Phddate, Phddegree, Msuniv, Msmajor, Msdegree, Bsuniv, Bsmajor, Bsdegree, Bsdate

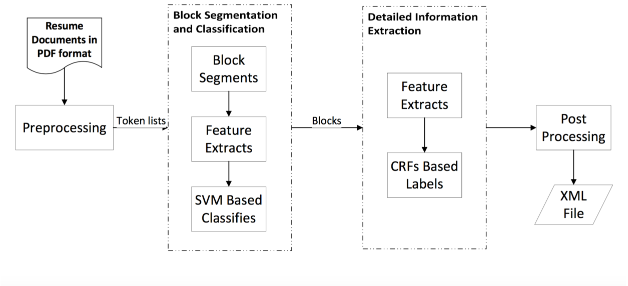


Figure 4. *Workflow of our method*

problem, the training dataset is defined as $(x_1,y_1),(x_2,y_2),...,(x_N, y_N)$, where $x_i \in R^N$ is a feature vector and $y_i \in -1,+1$ is the class label responding to x_i. Support Vector Machine attempts to find an optimal hyperplane to maximize the margin of two classes. The kernel function of an SVM is written as $K(x_i,x_j)$.

We extend the former SVM mode to multi-class classifiers through ”one versus one”.

In our implementation, LIBSVM[15], an optimized implementation of Support Vector Machine, is used to build the SVM classifier.

Feature Extraction

This paper uses both content-based features and layout-based features in our experiments. The features defined in our SVM model are described as follows:

Block keywords: Whether the block contains keywords as ”education”, ”honors”, ”contact”, ”publications” or others. These words are considered as important clues to block classifying.

Special patterns: Whether the block contains some special patterns such as e-mail, date, location or website. Regular expressions and pre-defined dictionaries are used to identify such special patterns.

Words with the largest font size: Words with the largest font size of the block are considered as features. They may be the section title of the resume document.

Words of the first line: Words with the largest font size of the block are considered as features. They may contain import clues of blocks.

Geometrical information: The geometrical information of the block such as the dimensions, relative positions and areas of the blocks.

Detailed Information Extraction

In the detailed information extraction step, the extraction problem is transferred into a sequence labeling problem. Only the education block and the personal block are selected in this step.

Content-based features and layout-based features are extracted of each token. A Conditional Random Fields (CRF) model is employed to extract detailed information.

In our experiment, CRFsuite[16], a fast implementation of Conditional Random Fields, is used for tagging.

Feature Extraction

As table 2 shows, the features we use in our Conditional Random Fields model can be classified into two categories: Content-based features and Layout-based features. Content-based features represent linguistic and grammar information of the tokens and they are similar to previous information extraction tasks. Layout-based features are PDF specific features which contain structure, format information of the PDF documents.

Features of a token

Content-Based Features	Orthographic Case
	Punctuation
	Number
	Dictionary
	DomainSpecific
	Conj & Prep
	SpecialItem
Layout-Based Features	Leftmost Token
	Long Gap
	Font Size
	Font Format
	Vertical Alignment
	Horizontal Alignment
	Single Line
	Format Change

Content-Based Features:

Orthographic Case: Whether the token is ICAP(Initial with uppercase), MCAP(mixed with uppercases and lower cases), ALLCAPS (all cases are uppercases), SCAP(single uppercase letter) or others.

Punctuation: This feature has two categories. The first one is that the token is a punctuation such as quote, dash, comma, semicolon, period. The second one is that the token contains a punctuation such as a word with periods, digits with a hashtag.

Number: This feature describes the numeric information of token. The values of this feature are: long-year (all digits and the value between 1900 to 2100), 4digit, 3digit, 2digit, 1digit, 4+digits(the token consists of only digits and the length of digits is more than 4), hasDigit(the token contains digits and other symbols), noDigit(the token does not contain digits).

Dictionary: Whether the token is in pre-defined keywords dictionaries. Such dictionaries include affiliation keywords, address keywords.

DomainSpecific: Whether the current token is in a domain specific list. The list is as 3 shows.

Conj & Prep: Whether the current token is one of the conjunctions or prepositions such as ”in”, ”and”, ”or”, ”of” or ”at”.

Special Items: Regular expressions are used to identify whether the current token is in E-mail format, URL format or none.

Domain list

Domain	**Definition**
Name list	First name (Male and Female) Last name
University list	A university list in the US and around the word.
US location list	A list of US Zip Codes, State, City and County
Degree list	A list contains aliases or abbreviations of PhD., Master, Bachelor such as Ph.D., M.E.
Month list	month names and month abbreviations

Layout-Based Features:

Font Size: This feature describes about the font size of the token. Relative font size is used in our experiment. Our method calculates the main font (the most frequent font) of the entire document. Then, if the font size of the token is larger than the main font, the value of this feature is ”Bigger”, if the font size of the token is smaller than the main font, the value of the feature is ”Smaller”. Otherwise, it will be set to ”Normal”.

Font Format: This feature describes the font format of the token. The value of this feature is bold, italic or others.

Horizontal Alignment: This feature describes whether the current token horizontally aligns with another token.

Vertical Alignment: This feature describes whether the current token vertically aligns with another token.

Leftmost Token: This feature describes whether the current token is the leftmost token of the line.

Long Gap: This feature describes the distance of the gap between the current token and the preceding token. It is a binary feature. A threshold is set before. If the distance is larger than the threshold, the value of the feature is set to 1, otherwise the value is set to 0.

Single Line: This feature describes whether the current token is a single line or in the same line with the preceding token and the following token.

Format Change: This feature describes whether the current token and the preceding token are different from font formats. The font format includes font name, font style and font size.

Experiments

Datasets

A crawler is used to collect resume documents in PDF format from Google Search Engine. Resume documents which are non-English or cannot be properly parsed are discarded at first. Finally, there are 400 resume documents in PDF format for our experiment.

Each resume document has been manually annotated for the fields described before. Human annotators conduct annotation on the tokens and blocks. A spec is created to guide the annotation process. We use 3/4 of them for training and others for testing. Four-fold cross-validation is used in experiments.

Evaluation Measures

In the experiments, we use an instance-based evaluation. It means that an extracted instance is considered correct only if it is totally identical to a hand-annotated instance. Instance precision is the percentage of extracted instances of attribute that are identical to annotated instance. Instance recall is the percentage of extracted instances of total extracted by CRF. Instance F1-score is defined according to standard measure methods[25].

Experimental Results

The result of the experiments

Attributes	Precision	Recall	F1-score
Name	87.00%	57.14%	68.88%
Phone	78.49%	77.66%	78.07%
Homepage	88.46%	92.00%	90.20%
Fax	94.59%	83.33%	88.61%
Address	52.04%	56.67%	54.26%
Affiliation	61.2%	42.86%	50.42%
PhD University	49.47%	55.29%	52.22%
PhD Major	67.36%	76.19%	67.04%
PhD Date	73.68%	83.33%	78.21%
PhD Degree	84.21%	94.11%	88.89%
MS University	50.52%	56.32%	53.27%
MS Major	63.92%	72.94%	68.13%
MS Date	73.20%	84.52%	78.45%
MS Degree	83.51%	93.10%	88.04%
BS University	57.45%	66.67%	61.71%
BS Major	64.89%	77.22%	70.52%
BS Date	76.60%	87.80%	81.82%
BS Degree	80.85%	84.44%	82.61%
Average	70.70%	75.44%	72.78%

As table 4 shows, our method achieves average 72.78% F1-score of the detailed information extraction. However, our method suffers in distinguishing affiliation and address, missing or major or university information of educational background.

Analysis and Discuss

Contribution of Features

The features we use can be divided into two categories. One is content-based features which are similar as other information extraction works. The other is layout-based features which are PDF-specific features.

To analyze the contribution of the PDF-specific features, we train two different models. 1) Only content-based features. 2) Both content-based and layout-based features.

Contribution of features

Features	Precision	Recall	F1-score
content	60.89%	57.91%	59.18%
content+layout	70.70%	75.44%	72.78%

As table 5 shows, we can see that layout-based features are very effective in our experiments. Layout-based features improve both averaged precision and averaged recall. Compared with only using content-based feature, the averaged F1-score with both content and layout features increases 20 percent.

Effectiveness of Hierarchical Structure

In our experiment, we use a hierarchical structure to represent the information of the resume documents. In this section, we test whether the hierarchical structure performs better than the flat model or not. Besides, we also want to test whether the hierarchical structure with tiny block sizes performs better than the hierarchical structure with large block or not.

The effect of different block size

Block Size	Precision	Recall	F1-score
Flat Structure	45.98%	78.32%	57.94%
Tiny Blocks	71.82%	65.32%	68.42%
Large Blocks	70.70%	75.44%	72.78%

As shown in table 6, we test the effects of three different block sizes of the experimental results. The first one is the flat structure(the block size is the entire document). The second one is the hierarchical structure of tiny blocks(each paragraph as a text block) and the third one is our previous method.

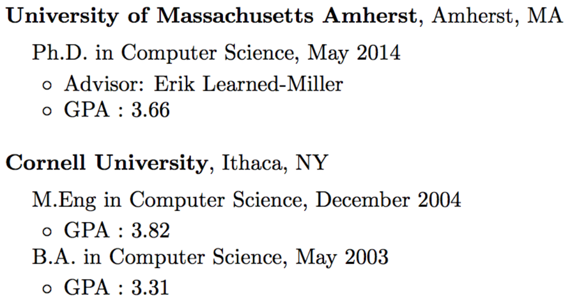
We can see that the hierarchical structure performs much better in the precision and F1-score with a little loss in the recall. It proves that hierarchical structure achieves a better F1-score than flat model. In addition, the model with hierarchical structure is more efficient than the flat model. It cost less time for training and testing.

Besides, experimental results show that the hierarchical model with large block is more efficient in F1-score. Because using tiny block increases errors in classifying block step.

Error analysis for education background extraction

In detailed information extraction step, we found that some information such as university or major are lost in our experiments.

Figure 5 shows the education background information of a person. His under-graduated university is same as his graduated university. But in our extraction result, we cannot get his Bsuniv information. Because the ”Cornell University” in the figure should be tagged as Msuniv and Bsuniv, but in our method, it is only tagged as Msuniv.



University of Massachusetts Amherst, Amherst, MA

Ph.D. in Computer Science, May 2014

- Advisor: Erik Learned-Miller
- GPA : 3.66

Cornell University, Ithaca, NY

M.Eng in Computer Science, December 2004

- GPA : 3.82

B.A. in Computer Science, May 2003

- GPA : 3.31

Figure 5. *Error extraction in education background block*

To solve this problem, a rule-based adjustment is used in our post-processing step. After the detailed information extraction step, if some tokens are tagged as degree information while major information or university information are lost, the missing information would be filled with adjacent information. For example, in figure 5, we will regard his Msuniv value ”Cornell University” as his Bsuniv value.

Rule-based adjustment for education background extraction

Process	Precision	Recall	F1-score
Rule-Based Adjustment	65.89%	89.13%	75.77%
No Adjustement	70.56%	79.70%	74.83%

As table 7 shows, the recall of the experiment has been greatly improved. But our method also resulted in the wrong filing. So the improvement on F1-score is not too much.

Error analysis for name extraction

From the experimental results, we can see that name extraction result has lower recall than other extraction results. We find that in some cases the name attribute does not appear in the personal block, but appears as the title of the documents. So our previous method does not work well on name extraction.

Considering the special nature of the name extraction, we designed a new method to extract names. According to the results of our statistics, the name which is as the title of the resume document often appears on a single line. Therefore, we treat name extraction problem as a text line classification problem.

Based on the obvious content and layout clues to the name line, a rule-based algorithm is used to filter out most of non-name lines. Then, a classification model is employed to classify whether the text line is a name line or not. At last, we do some post processing and check whether the extracted information is the name attribute or not.

The features used in our method are as follows:

Name line classification

Features	Definition
Line number	The line number of the text line.
Number of words	The total number of the words in the text line.
Capitalized words	The percent of the Capitalized words in the text line.
Bold Font Words	The percent of the words with the bold font in the text line.
Center Alignment	Whether the text is in the center of the line.
Left Alignment	Whether the text is in the left of the line.
Domain Words	The percent of the words in pre-defined name list.

As the table 9 shows, our text line classification method is a good complement to the previous one. It improves more than 20 percent on the F1-score.

Result of Name line classification

Methods	Precision	Recall	F1-score
Previous Algorithm	87.00%	57.14%	68.88%
Text line Classification	84.00%	86.60%	85.28%

Conclusion and Future Work

In this paper, we propose a hierarchical model for information extraction from resume documents in PDF format. According to the layout and format information on a resume document, the document is first segmented into blocks. Then the detailed information inside each block is extracted. The experimental results show that the average F1-score of the hierarchical extraction model achieves 72.78%, which is 25 percent higher than the flat model. Besides, layout-based features are verified to be useful with a 22 percent improvement in average F1-score. The experiments demonstrate that the method could achieve a high extraction accuracy with well adaptability to various document layouts.

In the future, we plan to update our method by trying other page segmentation algorithms to achieve a more accurate understanding of the layout and the content of the document. Also, we plan to combine different machine learning methods to further improve the results of the experiments.

Acknowledgments

This work is supported by the Projects of National Natural Science Foundation of China (No.61202232 and No.61573028).

References

[1] Peng, Fuchun, and Andrew McCallum. ”Information extraction from research papers using conditional random fields.” Information processing & management 42.4 (2006): 963-979.

[2] Lafferty, John, Andrew McCallum, and Fernando CN Pereira. ”Conditional random fields: Probabilistic models for segmenting and labeling sequence data.” (2001).

[3] Yu, Kun, Gang Guan, and Ming Zhou. ”Resume information extraction with cascaded hybrid model.” Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005.

[4] Yao, Limin, Jie Tang, and Juanzi Li. ”A unified approach to researcher profiling.” Proceedings of the ieee/wic/acm international conference on web intelligence. IEEE Computer Society, 2007.

[5] Pinto, David, et al. ”Table extraction using conditional random fields.” Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. ACM, 2003.

[6] Vapnik, Vladimir Naumovich, and Vlamimir Vapnik. Statistical learning theory. Vol. 1. New York: Wiley, 1998.

[7] Ciravegna, Fabio, and Alberto Lavelli. ”LearningPinocchio: Adaptive information extraction for real world applications.” Natural Language Engineering 10.02 (2004): 145-165.

[8] Smith, Dan, and Mauricio Lopez. ”Information extraction for semi-structured documents.” Proc. Workshop on Management of Semistructured Data. 1997.

[9] Ghahramani, Zoubin, and Michael I. Jordan. ”Factorial hidden Markov models.” Machine learning 29.2-3 (1997): 245-273.APA

[10] McCallum, Andrew, Dayne Freitag, and Fernando CN Pereira. ”Maximum Entropy Markov Models for Information Extraction and Segmentation.” ICML. Vol. 17. 2000.

[11] Cortes, C. and Vapnik, V. Support Vector Networks. Machine

Learning, 1995, 20: 273-297.

[12] Han, Hui, et al. ”Automatic document metadata extraction using support vector machines.” Digital Libraries, 2003. Proceedings. 2003 Joint Conference on. IEEE, 2003.

[13] Sha, Fei, and Fernando Pereira. ”Shallow parsing with conditional random fields.” Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics, 2003.

[14] Pietra, Stephen Della, Vincent Della Pietra, and John Lafferty. ”Inducing features of random fields.” Pattern Analysis and Machine Intelligence, IEEE Transactions on 19.4 (1997): 380-393.

[15] Chang, Chih-Chung, and Chih-Jen Lin. ”LIBSVM: A library for support vector machines.” ACM Transactions on Intelligent Systems and Technology (TIST) 2.3 (2011): 27.

[16] Okazaki, Naoaki. ”CRFsuite: a fast implementation of conditional random fields (CRFs).” URL http://www. chokkan. org/software/crfsuite (2007).

[17] Anjewierden, Anjo. ”AIDAS: Incremental logical structure discovery in PDF documents.” icdar. IEEE, 2001.

[18] Kushmerick, Nicholas, Edward Johnston, and Stephen McGuinness. ”Information extraction by text classification.” In The IJCAI-2001 Workshop on Adaptive Text Extraction and Mining. 2001.

[19] Councill, Isaac G., C. Lee Giles, and Min-Yen Kan. ”ParsCit: an Open-source CRF Reference String Parsing Package.” LREC. 2008.

[20] Cortez, Eli, et al. ”FLUX-CIM: flexible unsupervised extraction of citation metadata.” Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries. ACM, 2007.

[21] Sha, Fei, and Fernando Pereira. ”Shallow parsing with conditional random fields.” Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics, 2003.

[22] Sutton, Charles, and Andrew McCallum. ”An introduction to conditional random fields for relational learning.” Introduction to statistical relational learning (2006): 93-128.

[23] McCallum, Andrew, and Wei Li. ”Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons.” Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. Association for Computational Linguistics, 2003.

[24] Ciravegna, Fabio. ”lp2, an adaptive algorithm for information extraction from web-related texts.” In Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining. 2001.

[25] Rijsbergen, C. Information Retrieval. 1979.

Author Biography

Jiaze Chen received his BS in Computer Science from the Beijing University of Posts and Telecommunications(2014). Then he has been studying in Peking University for his Master Degree up till now. His work mainly focuses on information extraction and retrieval.