

Resume Information Extraction with Named Entity Clustering based on Relationships

Ertuğ Karamatlı, Selim Akyokuş
Doğuş University, İstanbul, Turkey
ertug@karamatli.com, sakyokus@dogus.edu.tr

Abstract

This paper presents an effective method for resume information extraction. The information extraction process consists of 4 phases. In the first step, a resume is segmented into blocks according to their information types. In the second step, named entities are found by using special chunkers for each information type. In the third step, found named entities are clustered according to their distance in text and information type. In the fourth step, normalization methods are applied to the text.

1. Introduction

Large companies and recruitment agencies receive, process and manage hundreds of resumes from job applicants. Besides, many people publish their resumes on the web. These resumes can be automatically retrieved and processed by a resume information extraction system. Extracted information such as name, education, work experience can be stored as a structured information in a database and then can be used in many different areas.

In contrast to many unstructured document types, information in resumes is in a semi-structured form where information is stored in blocks. Each block contains related information about a person's contact, education or work experience.

Even if it is in a restricted domain and semi-structured form, resume documents are not easy to parse automatically. They tend to differ in information types, information order, containing full sentences or not, etc. Also, conversion from other document formats (e.g. PDF, DOC, ODT, etc.) to text yields unexpected layout of information. To parse resumes effectively, the system should be independent of the order and form of information in the document.

This paper presents a resume information extraction system. In order to design an effective system, many types of resumes are analyzed to

find a robust, generalized and efficient way for information extraction. We assumed that resumes have a three level hierarchical structure where top most level contains segments. Segments consist of blocks that contain related information. Each block can contain several chunks which are named entities.

2. Related Work

Resume Information Extraction is an application area of Information Extraction (IE) which is a process that automatically extracts predefined types of information from unstructured documents. Information Extraction also includes structuring, grouping and preparing found data to populate a database. [1,2,3,4].

Resume information extraction, also called resume parsing, enables extraction of relevant information from resumes which have relatively structured form. Although, there are many commercial products on resume information extraction, there has been surprisingly little published research work on this area. Some of the commercial products include Sovren Resume/CV Parser [5], Akken Staffing [6], ALEX Resume parsing [7], ResumeGrabber Suite [8] and Daxtra CVX [9]. There is a little information in specification of these products about methods and algorithms used for information extraction.

There are four types of methods used in resume information extraction: Named-entity-based, rule-based, statistical and learning-based methods. Usually a combination of these methods is used in many applications. Named-entity-based information extraction methods try to identify certain words, phrases and patterns usually using regular expressions [2] or dictionaries [9]. This is usually used as a second step after lexical analysis of a given document [2,3]. Rule-based information extraction is based on grammars. Rule-based information extraction methods include a large number of grammatical rules to extract information from a given document [4,10,14]. Statistical information extraction methods apply

numerical models to identify structures in given documents [4]. The learning-based methods employ classification algorithms to extract information from a document. In these methods, a classifier is trained and then the classifier is used to extract relevant information [11].

Many resume information extraction systems employ a hybrid approach by using a combination of different methods. For example, the study in [12] uses Hidden Markov Model and Support Vector Machines which are statistical and learning-based methods respectively. Another work in [13] uses rule-based and statistical methods.

3. Information Extraction Process

The designed Information Extraction System consists of 4 phases: Text Segmentation, Named Entity Recognition, Named Entity Clustering and Text Normalization. In the end, the extracted information is produced in JSON or XML format.

Table 1. Extracted Information Types

Segment Types	Related Information Types
Contact	Name
	Phone
	Email
	Web
Education	Degree
	Program
	Institution
	DateRange
Experience	Position
	Company
	DateRange

3.1. Text Segmentation

In the first phase, a resume document is separated into several segments such as contact information segment and education information segment as shown in Figure 1.

Segmentation algorithm relies on the fact that each heading in a resume contains a block of related information following it. A dictionary is used to store common headings in a resume. These headings are searched in a given resume to find segments of related information. All of the text between the heading and the start of the next heading is accepted as a segment. An exception to this is the first segment which contains the name of the person and generally the contact information. It is found by extracting the text

between the top of the document and the first heading.

For each segment there is a group of named entity recognizers, called chunkers, that works only for that segment. This improves the performance and the simplicity of the system since a certain group of chunkers only works for a given segment.

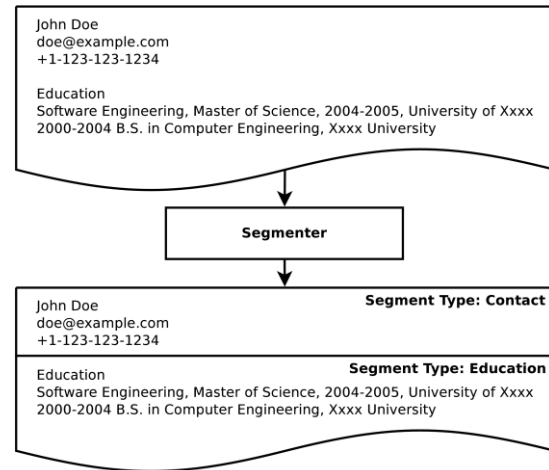


Figure 1. Text Segmenter

Segmentation is a crucial phase. If there is an error in the segmentation phase, chunkers will run on a wrong context. This will produce unexpected results.

3.2. Named Entity Recognition

Named entities are atomic elements that have predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. Resumes consist of mostly named entities and some full sentences. Because of this nature of the resumes, the most important task is to recognize the named entities. We have a set of modules called chunkers to perform named entity recognition. For each type of information, there is a specially designed chunker. Information types are shown in Table 1. Each chunker is run independently as shown in Figure 2.

Chunkers use four types of information to find named entities:

- *Known names*; through dictionaries of well-known institutions, companies, academic degrees, etc.
- *Characteristic prefixes and suffixes*; for institutions (e.g. University of, College, etc.) and companies (e.g. Corp., Associates, etc.)

- *Clue words*; like prepositions (e.g. in the work experience information segment the word after “at” most probably a company name)
- *Known patterns*; names of people (e.g. capitalization of letters and forms like John Bob Doe, J. Bob Doe, etc.)

The chunkers produce an output that contains information about named entities as shown in Table 2.

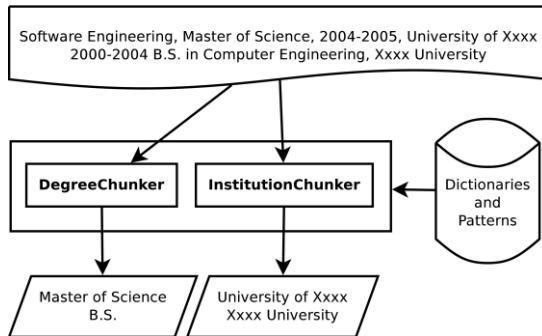


Figure 2. Chunks

3.3. Named Entity Clustering

Each segment (e.g. education information) contains a block of related information. For example, an education segment will have a number of blocks of information about educational institutions that a person attended. Information about each educational institution will be stored in a block. For example, an education information block can contain institution name, degree, major, and date information.

Table 2. Found named entities

Named Entity	Start	End	Type
Master of Science	22	39	Degree
University of Xxxx	52	70	Institution
B.S.	80	84	Degree
Xxxx University	110	125	Institution

In the previous step we obtain many independent named entities as shown in Table 2. They need to be grouped together to form a block of information. Related information is defined as shown in Table 1. Named entities (chunks) are grouped according to their proximity and type. The algorithm in Figure 3, tries to associate related entities into a group depending on their type and how much they are close to each other.

```

GroupChunks(chunks, text_length) {
  sorted_chunks := Sort all the chunks by their start
  positions in the text
  max_same_type := The maximum number of chunks in the
  type
  chunk_type_count := Number of different chunk types in
  chunks
  max_distance := text_length / max_of_chunks /
  chunk_type_count
  groups := list()
  current_group := list()
  for each chunk in sorted_chunks {
    previous_chunk := Last item in current_group
    has_chunk := length(current_group) > 0
    is_too_far := chunk.start_pos -
    previous_chunk.end_pos > max_distance
    is_same_type := is chunk has same type as a chunk in
    current_group
    if has_chunk AND is_too_far OR same_type {
      groups.append(current_group)
      current_group := list()
    }
    current_group.append(chunk)
  }
  groups.append(current_group)
}

```

Figure 3. Named Entity Clustering Algorithm

3.4. Text Normalization

In text normalization, some of the named entities are transformed to make it consistent. Table 3 shows some of the transformations performed on several text phrases.

In normalization phase, we expand some of abbreviations using dictionaries similar to the dictionary given in Table 4. For example, the abbreviation “B.S.” is expanded as “Bachelor of Science”. We also convert some of the text into a new form. For example, the first letters in a person’s name is capitalized as shown in Table 3. Some of the phrases are also converted to its most common form. For example, “University of Yale” is converted to “Yale University” as shown in Table 3.

Table 3. Applying text normalization

Input	Output	Type
B.S.	Bachelor of Science	Degree
JOHN DOE	John Doe	Name
University of Yale	Yale University	Institution

Table 4. Contents of the degree dictionary

Term	Abbreviations / Other forms
Bachelor of Science	B.S., BS, BSc
Master of Science	M.S., MS, MSc
Bachelor of Arts	B.A., BA
Doctor of Philosophy	Ph.D., PhD
Doctor of Medicine	Medicinae Doctor, M.D.

4. Performance Evaluation

We used precision, recall and F-measure metrics for performance evaluation [15]. Precision measures the number of relevant items retrieved as a percentage of the total number of items retrieved. Recall measures the number of relevant items retrieved as a percentage of the number of relevant items in collection. The F-measure is the harmonic mean of precision and recall.

$$precision = \frac{\text{number of relevant items retrieved}}{\text{total number of items retrieved}} \quad (1)$$

$$recall = \frac{\text{number of relevant items retrieved}}{\text{number of relevant items in collection}} \quad (2)$$

$$F - \text{measure} = 2 \times \frac{precision \times recall}{precision + recall} \quad (3)$$

To test the performance of the system, a collection of 20 resumes is gathered. These 20 resumes are analyzed by proposed system and the relevant information is extracted. On these 20 resumes, the identification success rates of segment types and some of the named entities are evaluated. Table 5 gives the precision, recall and F-measure rates for segments. Segments are usually recognized well by the system. F-measure for segments is between %95 and %100. This is because of the fact that the segment dictionary includes almost all of the common headers used in the resumes.

Table 5. Precision, recall and F-measure rates for segments

Segment Type	Precision (%)	Recall (%)	F-measure (%)
Title	100	100	100
Contact	100	100	100
Education	100	100	100
Experience	95	95	95

Table 6 shows the precision, recall and F-measure rates for named entities. The identification rates of named entities such as name, e-mail, date range and education program are in acceptable ranges. Because these named entities has specific format and include well known words in them. On the other hand, the recognition rates for company names are low. Because there aren't sufficient clue words that helps identification of company names. The success rate of company names can be increased by using a company names dictionary.

Table 6. Precision, recall and F-measure rates for named entities

Named Entity Type	Precision (%)	Recall (%)	F-measure (%)
Name	95.0	95.0	95.0
Email	100.0	100.0	100.0
Education Institution	100.0	83.33	90.91
Education Degree	100.0	85.71	92.31
Education Program	100.0	88.24	93.75
Education DateRange	88.24	88.24	88.24
Experience Company	78.57	64.71	70.97
Experience DateRange	94.74	90.0	92.31
Experience Position	92.86	76.47	83.87

5. Conclusion and Future Work

We presented a resume information extraction system based on named entity recognition and clustering of named entities by their relationships. The designed system is implemented with Python programming language. The output format of the system can be in JSON or XML [16]. The developed system has a flexible and modular structure that can be extended easily. A web interface is developed to test the system.

The resume extraction process consists of 4 phases. In the first step, a resume is segmented into blocks according to their information types. In the second step, named entities are found using special chunkers for each information type. In the third step, the found named entities are clustered into groups according to their distance in text and information type. In the fourth step, normalization methods are applied to the text.

We tested our system on a small number of resumes. Even though we use simple rules and dictionaries, the results are promising in comparison to other studies that use complicated methods.

As a future work, we will expand the resume data collection set and improve the performance of the proposed system. The developed system's performance can be improved by adding new types of chunkers, adding new rules and expanding the dictionaries. We also plan to incorporate other clustering algorithms and learning-based approaches in the future.

6. References

- [1] D. Appelt and D. Israel, "Tutorial: An Introduction to Information Extraction Technology", *IJCAI-99 Conference*, Stockholm, Sweden, 1999.
- [2] R. Grishman, "Information Extraction: Techniques and Challenges", *Lecture Notes In Computer Science*, vol. 1299, Springer-Verlag, London, 1997, pp. 10-27.
- [3] C. Siefkes and P. Siniakov, "An Overview and Classification of Adaptive Approaches to Information Extraction", *Journal On Data Semantics*, vol. 4, Springer, 2005, pp. 172–212.
- [4] S. Sarawagi, "Information Extraction", *Foundations and Trends in Databases*, vol. 1, 2008, pp 261-377.
- [5] Akken Stuffing, <http://www.akken.com/> (Accessed on March 12, 2010).
- [6] Sovren Resume/CV Parser, <http://www.sovren.com/> (Accessed on March 12, 2010).
- [7] ALEX Resume Parsing, <http://www.hireability.com/ALEX/> (Accessed on March 12, 2010).
- [8] ResumeGrabber Suite, <http://www.egrabber.com/resumegrabbersuite/> (Accessed on March 12, 2010).
- [9] Daxtra CVX, <http://www.daxtra.com/> (Accessed on March 12, 2010).
- [10] F. Reiss and S. Raghavan, R. Krishnamurthy, H. Zhu and S. Vaithyanathan, "An Algebraic Approach to Rule-Based Information Extraction" , *Proceedings of the 24th IEEE International Conference on Data Engineering*, Cancun, Mexico, 2008.
- [11] H.L. Chieu, H.T. Ng and Y.K. Lee, "Closing the Gap: Learning-Based Information Extraction Rivaling", *In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, 2003.
- [12] K. Yu, G. Guan and M. Zhou, "Resume information extraction with cascaded hybrid model", *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, 2005, pp. 499-506.
- [13] Z. Jiang, C. Zhang, B. Xiao and Z. Lin, "Research and Implementation of Intelligent Chinese Resume Parsing", *Proceedings of the 2009 WRI international Conference on Communications and Mobile Computing*, vol. 3, 2009.
- [14] J. Piskorski, M. Kowalkiewicz and T. Kaczmarek, "Information Extraction from CV", *Information Retrieval and Filtering*, 2005, pp. 185-192.
- [15] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Wokingham, UK, 1999.
- [16] D. Crockford, "JSON: The Fat-Free Alternative to XML", 2006, <http://www.json.org/fatfree.html> (Accessed on March 12, 2010).