

# Information Extraction for E-Job Marketplace

Dwi H. Widyantoro<sup>1</sup>, Yudi Wibisono<sup>2</sup>

1: School of Informatics and Electrical Engineering ITB, [dwi@if.itb.ac.id](mailto:dwi@if.itb.ac.id), 62 22 2508135

2: School of Informatics and Electrical Engineering ITB, [yudi@upi.edu](mailto:yudi@upi.edu)

**Abstract**—Indonesian job postings become widely available on the Internet recently. Information gathered from those sites is a great asset for developing an intelligent e-job marketplace. Job postings are usually presented in text document and each site uses different format or even cross lingual. In this paper, information extraction technique is employed to locate and extract specific and important data about available job opportunities from job postings on the Internet. Our early experiment with limited corpus shows promising result.

**Index Term**—information extraction, intelligent e-job marketplace.

## I. INTRODUCTION

Indonesian job postings become more and more available on the Internet recently. Several Websites such as karir.com present job information in a structured format. However, a lot more Websites use free or semi-structured format. This includes job posting on blog, online forum and mailing list.

Merging job information from Websites that use free or semi-structured document is very useful for developing intelligent e-job marketplace that supports high precision information retrieval, job recommendation and job summarization. Job postings are usually presented in natural language. Although providing convenience for human readers, this representation in general is unreadable by machines. In addition, each Website has different format and is sometimes cross lingual. This makes high precision retrieval from gathered job information a difficult task.

This situation has made information extraction task an attractive technique. The task is basically to extract specific data from free or semi structured job posting. Once extracted, data can be handled in a similar way as traditional database.

This paper discuss our first step on developing information extraction system for e-job market domain. In the next section we briefly provide overview of related work. In Section 3 we discuss the background and technique we use for information extraction. In particular, we define rule representation and manually construct extraction rule. Preliminary experiment and discussion of results are presented in Sections 4 and 5 respectively, followed by conclusions in Section 6.

## II. RELATED WORK

Several works in the information extraction for job domain have been done before (Soderland, 1999) (Califf et al., 2003) (Ciravegna et al., 2002) (Freitag, 2000).

WHISK (Soderland, 1999) is a system that can generate extraction rule automatically. They tested the system on job domain and used seventeen slots: ID, Title, Salary, Company, Recruiter, State, City, Country, Language, Platform, Application, Area, Req\_years\_experience, Desired\_years\_experience, Req\_degree, Desired\_degree, and Post\_date.

RAPIER (Califf et al., 2003) is a bottom-up relational learner. It employed pairs of sample document and filled template to induce extraction rule. For job domain, RAPIER used newsgroup at [misc.jobs.offered](http://misc.jobs.offered). RAPIER slots for job domain are: Job Title, Salary, State, City, Platform and Area.

Ciravegna et al. (Ciravegna et al., 2002) used information extraction as an annotation tool for semantic web on job posting. They found information extraction contribution to annotation can be quite high.

Freitag used hidden markov model to extract phrases (Freitag, 2000). They trained separated HMM for each slot with stochastic optimization. For job domain, they used 298 usenet job announcements with two slots: company, and job title.

## III. INFORMATION EXTRACTION SYSTEM

This section provides background on the task of information extraction from job posting. It also describes our techniques in the task.

### A. Information Extraction

Information extraction is a process to search specific and important data from natural language document. The attribute that we want to extract is generally called slot.

Slots for job domain that we will use are:

1. Job Title
2. Industry
3. Level of Education
4. Work Experience (in years)
5. Language Skill

As an illustration, Figure 1 shows an example of job posting for accounting position and Figure 2 shows expected slot value after extraction process. It is possible that one or more slots

are not filled after the extraction process.

PT. Qdc Technologies is The Fast Growing Company, specialized in Telecommunication infrastructure development and construction, invites you to be a part of our team for the following immediate opening:

ACCOUNTING STAFF (SURABAYA BASE)

Job requirements:

- Female
- Domicile in Surabaya
- Educational background min. D3 in Accounting is a must
- Having minimum experience for 1 year, preferably in project area
- Good command in English, both active and passive
- Dynamic and willing to work under target

**Figure 1: Job Posting**

1. Job Title: Accounting Staff
2. Industry: telecommunication infrastructure
3. Level of Education: D3
4. Work Experience: 1 year
5. Language Skill: English

**Figure 2: Expected slot value after extraction**

### B. Rule Construction and Representation

The important element for information extraction is a set of extraction rules, which identifies relevant information to be extracted. Every domain has different set of rules. An extraction rule can be constructed manually or generated automatically. Manual, hand crafted extraction rule construction is known as knowledge engineering approach and the automatic one is automatic training approach (Applet, 1999). The advantages and disadvantages between two methods are discussed in (Applet, 1999).

The advantages of knowledge engineering approach are:

1. Good performing system is not hard to build.
2. Best performing system can be achieved by hand crafted rules.

The advantages of automatic training approach are

1. Portable to other domains.
2. No expertise is required for customisation.
3. Full coverage of examples can be ensured by data driven rule acquisition.

The drawback of knowledge engineering approach are:

1. Very laborious development process.
2. Some changes in specification can be hard to accommodate.
3. Required expertise may not be available.

Disadvantages of automatic training are:

1. Training data may not exist and may be very expensive to acquire.
2. Large volume of training data may be required.

3. Changes to specifications may require reannotation of large quantities of training data.

In this paper, we use knowledge engineering approach as a baseline. We will use machine learning technique to build extraction rule automatically in future work.

The knowledge engineering approach is characterized by the development and formalism of the extraction rule by a “knowledge engineer,” i.e. a person who is familiar with the information extraction system. The knowledge engineer either on his own, or in consultation with an expert in the domain of application, develops extraction rules. Typically, the knowledge engineer will have access to a moderate-size corpus of domain-relevant texts; he/she will also use his or her own intuitions. The latter is very important. It is obvious that the skill of the knowledge engineer plays a major role in the level of performance that will be achieved by the overall system.

For rule representation, we employ an adapted regular expression pattern for identifying relevant phrases and exact delimiters of those phrases. Figure 3 provides an example of pattern for “industry” slot.

```
Pattern1:: ("we are work at" | "company in" |
"specialized in") <industry> "."
Pattern2:: "growing" <industry> "looking for"
Pattern3:: "berkembang dibidang" <industry> "di"
<work_location>
```

**Figure 3: Extraction Rule for Industry Slot**

Slot is represented by word in < >. An extraction rule could contain more than one slot. For example, in pattern3, there are two slots: industry and work\_location. A pattern could also contain cross lingual word.

Extraction rule allows a form of disjunction, which is simply a set of terms that are considered to be an equivalence class. Pattern1 in figure 3 shows example of disjunction: (“we are work at” | “company in” | “specialized in”).

Wildcard “\*” can be used to accept any word. Unlike regular expression, the wildcard in extraction rule is limited to enforce proximity between matched tokens. In Figure 4, the wildcard between “from” and “university” doesn’t match every possible strings between those literals. Allowing an unlimited number of matches on each wildcard would also make the pattern matching unacceptably slow.

```
Pattern4:: "min" <degree> "from" * "university"
```

**Figure 4: Wildcard example in an extraction rule**

There are three types of wildcard:

1. Exactly one word.
2. Short sequence: one to three words.
3. Long sequence: one to ten words.
4. Optional word. This wildcard uses “?” sign.

Therefore, pattern4 in Figure 4 could be re-written as:

```
Pattern4B:: "min" *one-word <degree> "from"
*short-word "university"
```

Figure 5: Detailed wildcard example in rule

In pattern4B, we expect exactly **one** word between “min” and “from” and expected **one to three** word between “from” and “university”. For instance, pattern4B would match this sentence: “min bachelor from any reputable technology university”. First wildcard would match: “bachelor” and the second wildcard match: “any reputable technology”.

The extraction rules are then transformed into regular expression format. So, pattern4B in Figure 5 will be transformed into an extraction rule as shown in Figure 6.

```
Pattern4C:: min (\s+[\s&&[^\.]]+\s*)
from (\s+[\s&&[^\.]]+\s*){1,3}
university
```

Figure 6: Extraction rule in regular expression format

Based on the examination of 25 job postings, we manually crafted 15 extraction rules for 5 slots: Job Title, Industry, Level of Education, Work Experience, and Language Skill.

Figures 7,8,9,10 give extraction rules for each slot (except for industri slot shown in Figure 3).

```
Pattern5A :: ("position for"|"urgently
required") <job_title>
Pattern5B :: "talented person(s)?" <job_title>
"as"
```

Figure 7: Example of extraction rules for "Job Title" slot

```
Pattern6A :: "education background"
("min"|"minimum") <education>
Pattern6B :: <education> "from" *shortWord
"university"
```

Figure 8: Example of extraction rules for "Level of Education" slot

```
Pattern7A :: <year_experience> "year(s)?"
*shortWord "experience"
Pattern7B :: "experience" <year_experience>
year(s)?"
```

Figure 9: Example of extraction rules for "Work Experience" slot

```
Pattern8A ::
("writing"|"speaking"|"speak"|"write"|"written")
("in"|"good")? <language>
Pattern8B :: <language> *shortWord
("oral"|"written")
```

Figure 10: Example of extraction rules for "Language Skill" slot

### C. E-Job Marketplace

E-Job marketplace is a web-based application providing services application. Ideally, e-job marketplace has following

features (Widyantoro, 2007):

1. Automatically collecting job posting from the Internet.
2. Applying state-the-art of machine learning and techniques to better recommend job information (good candidates) to job seekers (job providers),
3. Providing adaptive user interface to use in mobile devices (cellular phone, PDA, or Pocket PC) using automatic job summarization.

Using information extraction, we can convert semi-structured job posting from various sources into structured information in traditional database. This is important element to create e-job marketplace.

## IV. EXPERIMENT

This section provides experiment setting for information extraction on Indonesian job posting.

### A. Preprocessing

For the experiment, we use a small corpus collected from <http://karirkerja.blogspot.com>. All header, footer and advertising section are removed. Some tags like <BR> and <P> are converted to “.”. Figure 11 shows a document after preprocessing.

```
PT. Qdc Technologies is The Fast Growing Company, specialized in.
telecommunication infrastructure development and construction, invites
you. to be a part of our team for the following immediate opening:
ACCOUNTING STAFF (SURABAYA BASE). Job Responsibilities:
· Manage petty cash project.
· Collecting & checking timesheet for all project staff.
· Coordinating company operational vehicle.
· Distribute Daily Worker Salary each week.
· Responsible for all duties and matters relates to administration project. office.
Job requirements:
· Female.
· Domicile in Surabaya.
· Educational background min. D3 in Accounting is a must.
· Having minimum experience for 1 year, preferably in project area.
· Good command in English, both active and passive.
· Dynamic and willing to work under target.
Application letters can be sent by e-mail to: hr@qdc.co.id
```

Figure 11: Preprocessed Document

### B. Extraction Process

In the extraction process, each rule is applied on text document sequentially. Slot is filled when matched string pattern is found. It is possible that a slot is filled by more than one extraction rules. In this case, the slot will be overwritten by the latest matched extraction rule.

### C. Evaluation Method

Information extraction evaluation typically uses two performance measures: precision (see Equation 1) and recall (Equation 2).

$$precision = \frac{\#ofCorrectValueExtracted}{\#ofValueExtracted} \quad (1)$$

$$recall = \frac{\#ofCorrectValueExtracted}{\#ofAllCorrectValueInSlot} \quad (2)$$

F-measure was introduced to combine precision and recall and is computed as in Equation 3 (same weight is given to precision and recall)

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

These performance metrics are generally more informative than the accuracy of predicting the presence/absence across all slot-value pairs (Nahm et al., 2000).

## V. RESULT AND DISCUSSION

The result of our experiment shows that our system can still be improved. Although we have precision of 88.9%, recall was much lower at 37.5% and F-measure was 52.74. Table 1 shows detailed precision and recall for each slot.

Low recall is common in information extraction. Using job domain document, RAPIER (Calif et al., 2003) had precision 84% and recall 53%. WHISK (Soderland, 1999) had precision 80% and recall 57%.

**Table 1: Experiment result for each slot**

	Job title	degree	work-exp	industry	Lang.
Prec.	75.00	100.00	100.00	100.00	33.33
Recall	20.00	50.00	58.33	50.00	9.09
F-meas.	31.58	66.67	73.68	66.67	14.29

“Work-experience” slot provides the best performance. It might be due to the fact that work experience in job posting usually uses the same pattern:

`<work-experience> years * experience`

For example, “5 years related work experience”, “7 years of relevant experience”, “5-10 years experiences”.

The worst performer was “language” slot. The main reason was over generalization in one of extraction rule:

`"good" * "communication"`

This rule is meant to match “good English communication” but it also matches the wrong sentence like “good quality in communication skill”. We need to evaluate more job documents in order to refine extraction rule and to improve our system performance.

## VI. CONCLUSIONS AND FUTURE WORK

Information extraction can be used to extract semi-structured, natural language job posting effectively and efficiently. In the future we need to compare the performance between knowledge engineering approach and automatic training approach.

## ACKNOWLEDGMENT

This work is supported by Incentive research grant from the Ministry of Research and Technology of the Republic of Indonesia under contract number 75/RT/Insentif/PPK/I/2007.

## REFERENCES

- [1] Appelt, D.E (1999).. Introduction to Information Extraction Technology, IJCAI-99
- [2] Califf, M.E., Mooney, R.J. (2003). Bottom-Up Relational Learning of Pattern Matching Rules for Information Extraction.
- [3] Ciravegna, F., Dingli, A., Petrelli D., Wilks, Y. (2002) UserSystem Cooperation in Document Annotation based on Information Extraction, In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management.
- [4] Freitag, D., McCallum (2000). A. Information Extraction with HMM Structures Learned by Stochastic Optimization Proceedings of the Eighteenth Conference on Artificial Intelligence (AAAI-2000)
- [5] Nahm, U.Y., Mooney, R.J (2000). A Mutually Beneficial Integration of Data Mining and Information Extraction. In the Proceedings of 17<sup>th</sup> National Conference on Artificial Intelligence (AAI-2000), pp 627 – 632.
- [6] Soderland, S. (1999). Learning Information Extraction Rules for Semi-Structured and Free Text. Machine Learning volume 34 number 1-3 pages 233-272.
- [7] Widyantoro, D.H., Khodra, M.L., Wibisono, Y. (2007) Toward the Development of Intelligent E-Job Marketplace. Submitted to ICTS 2007