A Matching Approach based on Term Clusters for eRecruitment

Gülşen Bal[1], Aşkın Karakaş[2], Tunga Güngör[3], Fatmagül Süzen[4], Kemal Can Kara[5]

Kariyer.net, [1]gulsenb@kariyer.net, [2]askin@kariyer.net, [4]fatmaguls@kariyer.net, [5]cank@kariyer.net

Boğaziçi University, Computer Engineering Department, [3]gungort@boun.edu.tr

Abstract. As the Internet occupies our daily lives in all aspects, finding jobs/employees online has become an important issue for job seekers and companies. However, it is difficult for a job applicant to find the the best job that matches his/her qualifications and also it is difficult for a company to find the best qualified candidates based on the company's job advertisement. In this paper, we propose a system that extracts data from free-structured job advertisements in an ontological way in Turkish language. We describe a system that extracts data from resumés and jobs to generate a matching system that provides job applicants with the best jobs to match their qualifications. Moreover, the system also provides companies to find the best fit for their job advertisement.

1 Introduction

The Internet has affected our daily life activities in several ways and it enables us to perform many things easily, including shopping, reading news, online banking, etc. In the past, seeking for a job was requiring interviews and several visits to the offices and companies. However, currently online recruitment websites make it possible for job seekers to look for jobs based on their criteria. Additionally, companies are able to find appropriate candidates based on their job advertisement. However, it takes too much time for a company to examine hundreds of resumes and find the most qualified candidates among them. Moreover, it is also difficult for a candidate to find jobs that meet his/her education, qualifications, and work experiences.

There are some related works that match resumés and job advertisements for the English language. Crow and DeSanto proposed a system that identifies the basic parts of the resumés and creates search methods based on these identified parts [1]. They create ontologies for job applications and use rule-based approaches to extract concepts. Another similar study performs semantic search for job advertisements in online recruitment websites [2]. The system creates ontologies with the help of other ontologies that were built in the related area before. The contents of the texts are clustered thematically and the similarities between concepts are observed. Another study based on matching resumés and job applications aims at generating a system which is based on description

logics [3]. This study just focuses on skills in resumés and tries to analyze and match skills. A concept-based algorithm that gives an output of potential-match and partial-match was developed. In the literature, there are other studies that aim to match resumes and job advertisements and prepare ontologies with semi-automatic methods [4,5,6].

Another study examines ontology matching approaches in different ways [7]. These approaches include terminological, structural, and extensional matching techniques. In terminological matching, the system calculates the similarity between texts using names, labels, etc. and matches them. In the structural matching technique, the system compares entity descriptions for each ontology (internal structure) or the corresponding points that each entity may have with others (external structure). For extensional matching, the system compares the instance/extension or the length of the classes of ontologies: in other terms, class instantiations or objects. This study detemines whether the matching process composed of text-based string sequences followed by terminological matching technique presents more accurate results than the other techniques. The basic approach of these studies is to figure out concepts in a semantic way and analyze different matching processes. Because of that, instead of word-based analysis, concept-based approaches or items that consist of word groups and ontologies should be used. In this paper, the proposed system embraces the same idea as well.

2 Methodology

2.1 Compiling Term Lexicons

Before the process of matching resumés and job advertisements, the system extracts information from resumés and job advertisements to create a lexicon of terms for the positions in job advertisements. Since the information in the resumé is saved in a structured way in the database, there is no need for an extra process to extract information. However, companies use free format texts for job advertisements and there is no standard format for it. Job advertisements in our system (Kariyer.net) are composed of two parts: general qualifications and job description. General qualifications are used to specify the qualifications of a position; like required skills, educational background, foreign language, and work experience information. Job description is used for indicating responsibilities and duties of the jobs. Since the proposed system is interested only in qualifications required for the position, the job description part will not be analysed for information extraction. For analysis, the system first creates sentences from free format text. Note that the sentences in advertisements do not end with some punctuation marks as in ordinary text. Instead, each sentence is like a long phrase that emphasizes a qualification. We analyzed more than 100 job advertisements and tried to find the most common sentence structures and define the rules. At the final phase of the analysis, we generate “a lexicon of endings” to identify the end of the sentences. This lexicon contains the words

or word groups that are used at the end of the sentences, so we can extract them from the free format texts. In English, verbs are at the middle of the sentences, whereas in Turkish, they are at the end of the sentences. That is why we labeled it as the lexicon containing endings.

Examples of ending word phrases in English: *Experienced in, Having knowledge of, Hands-on experience with, Having experience in, Experienced in*, etc.

Examples of ending word groups in Turkish: *bilgili (well informed), tecrübeli (exprerienced), bilgi sahibi (has knowledge of), deneyimli (has hands-on experience), yeteneğine sahip (has the ability)*, etc.

After the sentences are found, we analyzed them and focused on how to identify the terms in the sentences. Based on this analysis, we created some pattern rules to find the terms. Examples of pattern rules are shown below:

- T + and/or + T + {SpecialWord} + {EndingWord}
- T, T, T {EndingWord}
- T(3) + {SpecialWord} + {EndingWord}
- T, T(2) + and + T + {EndingWord}
- T(3) + {SpecialWord} + … + {EndingWord}
- T + {SpecialWord} + (T, T) + {SpecialWord} + {EndingWord}

where

T: term, T(n): terms composed of *n* words

SpecialWord: a word in the set {konusunda (*about*), konularında (*in the field of*), üzerinde (*upon*), ...}

EndingWord: a word in the set {tecrübeli (*experienced*), bilgi sahibi (*having knowledge of*), ...}

We use the conjuctions "ve" (*and*) and "veya" (*or*) as well as commas to find the terms. Terms can be composed of one or more words. Moreover, we use the data containing SpecialWords and EndingWords, which are determined by analyzing the job advertisements. Based on these rules, we extract words or word groups.

The system finds the terms based on pattern rules, but we do not know wheter these terms are related to the position in the job advertisement or not. To eliminate the unnecessary terms, the system implements a morphological analysis process to the terms, which reveals the stems and suffixes of the words. Generally, using word stems instead of the surface forms of the words gives better results for agglutinative languages like Turkish. After morphological analysis, morphological disambiguation process is applied in order to find the correct sense of the word. In this work, we used a state-of-the-art Turkish morphological analysis and disambiguation package [8].

Table 1 shows examples of extracted terms from a job advertisement for software engineering and their morphological representation.

Term	Morphological Analysis
ASP	ASP[Noun]+[Acro]+[A3sg]+[Pnon]+[Nom]
C	C[Noun]+[Acro]+[A3sg]+[Pnon]+[Nom]
Microsoft .NET	Microsoft[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom] .Net[Unknown]
problem	problem[Noun]+[A3sg]+[Pnon]+[Nom]
Java J2EE	java[Noun]+[A3sg]+[Pnon]+[Nom] J2EE[Unknown]
Yazılım Mühendisliği	Yazılım[Unknown] Mühendisliği[Unknown]
Microsoft .NET C	Microsoft[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom] .NET[Unknown] C[Noun]+[Acro]+[A3sg]+[Pnon]+[Nom]

Table 1. Morphological representation of words

To identify the terms that give useful information and will be used in the process, we have analyzed job advertisements that are looking for a software engineer and found the terms manually. Then, we obtained the mophological analysis representation of these terms and found the most common mophological analysis representation of the software engineering terms. Finally, we defined these representations as rules to eliminate unnecessary terms. Morphological analysis rules and their explanations for the software engineering position are shown in Table 2.

This term extraction process was applied to 21 selected positions. These positions are: software engineer, accounting specialist, mechanical engineer, architect, electrical engineer, production engineer, graphic designer, lawyer, electrical and electronic engineer, project engineer, business analyst, quality engineer, planning engineer, financier, interior architect, research and development engineer, enviromental engineer, technical service engineer, project manager, and industrial engineer. We built morphologic analysis rules for each position by analyzing each position's job advertisements. After the elimination of terms with morphological analysis rules is completed, we implemented an additional process formed of three steps in order to determine which terms actually belong to the domain of that position. [9,10].

Rule	Rule Meaning
[Unknown]	Unknown word
[Noun]+[A3sg]+[Pnon]+[Nom]	Noun
[Noun]+[Acro]+[A3sg]+[Pnon]+[Nom]	Acronym Nominative
[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom]	Proper Noun Nominative
[Noun]+[A3sg]+[Pnon]+[Nom], [Noun]+[A3sg]+[Pnon]+[Nom]	Singular Noun Nominative, Singular Noun Nominative

Table 2. Morhologic analysis rules

The first step is the Domain Relevance (DR) process. DR indicates the amount of information captured within the target domain (position) with respect to the entire collection of domains. More precisely, given a set of n domains (D_1, …, D_n), the domain relevance of a term t (t may be a single word or a multiword term) is computed as:

$$DR_{D_i}(\mathrm{t}) = \frac{\hat{P}(t|D_i)}{max_j(\hat{P}(t|D_j))} = \frac{freq(t, D_i)}{max_j(freq(t, D_j))}$$

where

$P(t \mid D_i)$: Probability that term t occurs in domain D_i

Freq (t, D_i) : (number of times term t occurs in domain Di) / (number of times all terms occur in domain Di)

The second step is the Domain Consensus (DC) process. DC measures the distributed use of a term in a domain D_i. The distribution of a term t in documents $d_j \in D_i$ can be taken as a stochastic variable estimated throughout all $d_j \in D_i$. The entropy H of this distribution expresses the degree of consensus of t in D_i . More precisely, the domain consensus is expressed as follows:

$$DR_{D_i}(\mathrm{t}) = -\sum_{d_k \in D_i} \hat{P}(t|D_k)\log(\hat{P}(t|D_k)) = -freq(t, d_k)\log(freg(t, d_k))$$

where $P(t \mid D_k)$ and Freq (t, D_k) are defined as above.

The third step is the Lexicon Cohesion (LC) process. In this step, it is used to determine whether the words in the (multiword) term t occur in the documents separately or together.

$$LC_{D_i}(t) = \frac{n \cdot freq(t, D_i) \cdot \log(freq(t, D_i)}{\sum_{w_j} freq(w_j, D_i)}$$

where w_j denotes the j-th word in the n-word term t, and Freq (t, D_i) and Freq (w_j, D_i) are defined analogously as above.

We apply all the three metrics DR, DC and LC, each of which returns a number between 0 and 1. These figures are combined with equal weight to each metric:

$$DomainResult(t, D) = \alpha_1 DR + \alpha_2 DC + \alpha_3 LC$$

where t denotes a term, D denotes a domain, and $\alpha_1=\alpha_2=\alpha_3=1/3$.

Afterwards, the system eliminates terms based on the value of their results. We analyzed and defined cut off values for every domain to eliminate terms. Then, we manually check and delete the unnecessary terms and create a lexicon of terms for every domain (position). For instance, for the software engieering position, we built a lexicon formed of 887 terms.

2.2 Creation of Term Clusters

In order to use the terms in the term lexicons for the matching process, the system analyzes and sorts the terms which are used together frequently. To determine these term groups, the following data are used:

- How many times the term is used in job advertisements
- How many job advertisements include the term
- Terms which are used together and their frequency

Based on these criteria, the system finds the terms that are used together with a frequency of 30% or more. From these terms, we create term clusters that are composed of 2, 3, 4, 5, 6, and 7 terms. Table 3 shows examples of the term clusters.

Length	Terms
7	ajax, asp.net, CSS, HTML, Javascript, XML ,Web
6	asp.net, C# ,Javascrpit, SQL, XML, Web
5	.NET, asp.net, C#, Web, SQL
5	afnetworking, CoreData, CoreGraphics, CoreLocation, QuartzCore
5	ajax.CSS,HTML,Jquery,JavaScript
3	Amazon AWS, Bamboo, Microsoft Azure
3	Hibernate, J2EE, Spring
3	Cassandra, Hbase, Hadoop
3	JSP,struts, servlet
2	java,Oracle
2	MongoDB, noSQL
2	MS Visio , Ms Project
2	android, ios
2	ABAP, SAP

Table 3. Example term clusters

2.3 Matching of Resumés and Job Advertisements

First of all, to match resumés and job advertisements, the system finds related terms. Afterwards, the system uses the extracted terms from job advertisements and resumés as well as the term clusters to include term relations to the matching process. We implement the well-known cosine similarity measure to evaluate the similarity between job advertisements and resumés. To evaluate the similarity, we create vectors for each job advertisement and resumé. For example, in order to match the resumés and the job advertisements in the domain of software engineering position, the system creates vectors that are composed of the items in the term lexicon concerning software engineering (887 terms).

Cosine similarity formula is shown below, which returns a similarity score between 0 and 1. A score of 0 means two vectors are completely different, while 1 means they are exactly the same. If the result approaches to 1, it means the resumé and the job advertisement match more.

$$similarity(A,B) = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{i=1}^{N} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(a_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

2.3.1 Boolean Weighting

The vectors of the resumé and the job advertisement have values that are based on their domain's lexicon of terms. For instance, both vectors will have a dimension of 887 in the domain of software engineering. In this method, when assigning values to the vectors, we use boolean weighting. That is, an element will take a value of 0 if the term does not occur in the resumé/job advertisement and 1 otherwise. An example is shown in Fig. 1.

Term Lexicon	**Job Advertisement**	**Resume**
.NET	1	0
C#	1	0
ASP.NET	0	0
HTML	0	1
JAVA	0	1
Hibernate	0	0
XML	1	1
MS SQL Server	1	0
PL/SQL	0	1
Ireport	0	0
J2MEE	0	1

Figure 1. An example resumé and job advertisement

Based on these vectors of the resumé and the job advertisement, we see that the resumé has HTML, JAVA ,XML, PL/SQL and J2MEE terms, while the job advertisement has NET, C#, XML and MS SQL Server terms. According to these values, the similarity between job advertisement and resume is calculated by cosine similarity.

2.3.2 Assigning vector values with term clusters

In addition to taking into account the presence or absence of terms in the resumé or the job advertisement, the relationships between the terms may also be deemed as important, which may affect the matching results. To include term clusters in the matching process, the system assumes that the terms which are in the same group are related. In this way, if the resume/job description has a non-existing term which is in the same group with a term that the resume/job advertisement contains, instead of assigning the value of 0 to that term, the frequency of terms that are used together is assigned. Thereby, the vectors of the job advertisement and the resumé come closer and the distance between vectors decreases.

3 Experiments

The experiments are applied to the positions of Software Engineering and Accounting Specialist. We randomly selected 10 job advertisements and 100 resumés for each domain. Then the system calculates the matching score with cosine similarity. The system selects 5 highest scoring resumés for each one of the 10 job advertisements. Table 4 shows some examples of job advertisement–resumé matching results for three example job advertisements 112, 156 and 163. In the table, the method column denotes the method with boolean weighting (1-BW) or the method with term clusters (1-TC). Index indicates the sequence of the top 5 resumés for each job advertisement using that method.

To measure the accuracy of our system, we defined a gold standard which was determined by a Human Resources specialist in the company. We asked for the specialist to select 5 resumés that are the most suitable for the selected job advertisements. Table 5 shows the results of the resumé selection of the specialist for the same three job advertisements.

ID	Method	Job Ad. ID	Resume ID	Matching Score	Index
1	1-BW	112	106890930	0.3651483	1
2	1-BW	112	102318815	0.2637521	2
3	1-BW	112	110299507	0.2581988	3
4	1-BW	112	9463976	0.2581988	4
5	1-BW	112	11777808	0.2306328	5
6	1-BW	156	215689	0.3481553	1
7	1-BW	156	102318815	0.3405026	2
8	1-BW	156	2563345	0.3333333	3
9	1-BW	156	108766850	0.3333333	4
10	1-BW	156	105445360	0.3187883	5
11	1-BW	163	102318815	0.4865336	1
12	1-BW	163	108766850	0.4082482	2
13	1-BW	163	8624239	0.3644054	3
14	1-BW	163	215689	0.3553345	4
15	1-BW	163	11777808	0.3403516	5
16	1-TC	112	1307469	0.5207212	1
17	1-TC	112	308945	0.3693846	2
18	1-TC	112	106890930	0.3651483	3
19	1-TC	112	7134122	0.3132371	4
20	1-TC	112	102318815	0.2637521	5
21	1-TC	156	7134122	0.4235080	1
22	1-TC	156	215689	0.3481553	2
23	1-TC	156	102318815	0.3405026	3
24	1-TC	156	2563345	0.3333333	4
25	1-TC	156	108766850	0.3333333	5
26	1-TC	163	102318815	0.4865336	1
27	1-TC	163	108766850	0.4082482	2
28	1-TC	163	8624239	0.3644054	3
29	1-TC	163	215689	0.3553345	4
30	1-TC	163	11777808	0.3403516	5

Table 4. Matching results

Job Ad.	Resume 1	Resume 2	Resume 3	Resume 4	Resume 5
112	1321224	163083	41816	106890930	308945
156	41816	2563345	215689	102318815	106890930
163	102318815	41816	168821	215689	11777808

Table 5. Gold standard data for the job advertisements

To find the accuracy of our system, we compared our results with the gold standard data. The results are shown in Table 6. We obtain success rates up to 42% which is a significant success rate in the recruitment domain. As seen in the table, the matching method that uses the relationship between terms gives a more successful result than the Boolean weighting method. Thus, we can comment that analysing the relationship between terms improves the performance of the matching process.

Method	**Similarity**
1-BW	0.36
1-TC	0.42

Table 6. Performance results

4 Conclusions

Finding the best candidates that fit the requirements of the job is time consuming and necessitates human effort. Also, finding the best jobs that meet the qualifications and work experiences of the candidate has been an important research issue in recent years. The proposed system is the first system that works in Turkish. The system extracts the terms from job advertisements and creates a lexicon of terms. Afterwards, it implements resume and job advertisement matching with cosine similarity. Based on performance results, the matching method that uses term clusters gives higher success rates. For future work, we plan to develop other versions of the methods and calculate the performance results by taking into account the rankings for an advertisement.

5 Acknowledgements

This study is supported by TÜBİTAK TEYDEB programme with the project number 3130841.

References

1. Crow, D. and DeSanto, J. “A Hybrid Approach to Concept Extraction and Recognition-based Matching in the Domain of Human Resources”, Proc. of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2004.
2. Mochol, M., Wache, H. and Nixon, L. “Improving the Accuracy of Job Search with Semantic Techniques”, Proc. of BIS, LNCS 4439, Springer, s. 301–313, 2007.
3. Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F.M., Mongiello, M. and Mottola, M. “A Formal Approach to Ontology-based Semantic Match of Skills

Descriptions", Journal of Universal Computer Science, Cilt 9(12), s. 1437-1454, 2003.

4. Mochol, M., Paslaru, E. and Simperl, B. "Practical Guidelines for Building Semantic eRecruitment Applications", Proc. of International Conference on Knowledge Management, Special Track: Advanced Semantic Technologies (AST), 2006.
5. Le, B.T., Dieng-Kuntz, R. and Gandon, F. "On Ontology Matching Problems for Building a Corporate Semantic Web in a Multi-communities Organization", Proc. of the Sixth International Conference on Enterprise Information Systems, Porto, Kluwer, 2005.
6. Ehrig, M. and Sure, Y. "Ontology Mapping - An Integrated Approach", Proc. of ESWS, LNCS 3053, Springer, p. 76-91, 2004.
7. Hassan, F., Ghani, I., Faheem, M., and Hajji, A. "Ontology Matching Approaches for eRecruitment", Proc. of ESWS, LNCS 3053, Springer, p. 76-91, 2004.
8. Sak, H., Güngör, T. and Saraçlar, M. "Resources for Turkish Morphological Processing", Language Resources and Evaluation, Vol. 45(2), p. 249–261, 2011.
9. Navigli, R. and Velardi, P. "Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites", Association of the Computational Linguistics, 2004.
10. Sclano, F. and Velardi, P. "Term Extarctor: a Web Application to Learn the Common Terminology of Interest Groups and Research Communities", TIA, 2007.