Ontology Based Job and Resume Matcher

Nimet Tülümen
Department of Research and Development
Talentra Recruitment Agency
Istanbul, Turkey
nimet.tulumen@talentra.net

Ali Öztaş
Department of Research and Development
Talentra Recruitment Agency
Istanbul, Turkey
ali.oztas@talentra.net

Ali Nohutcu
Department of Research and Development
Talentra Recruitment Agency
Istanbul, Turkey
ali.nohutcu@talentra.net

Günnur Sevgi Aktoros Genç
Department of Recruitment
Talentra Recruitment Agency
Istanbul, Turkey
gunnur.aktoros@talentra.net

Tunga Güngör
Department of Computer Engineering
Boğaziçi University
Istanbul, Turkey
gungort@boun.edu.tr

***Abstract*—The purpose of our work is to design a system that matches job ads and resumes, then assign them a scoring point. Thereafter, rank them according to their scores. The system composes mainly from two parts; information extraction and matching. We used natural language processing techniques for information extraction and ontology for matching and scoring. We chose to create our own ontology given that we can benefit from the knowledge of our HR experts regarding matching and scoring. Overall purpose of the study is to find the right candidates for the selected job within a large pool of resume dataset.**

Keywords—cv, job ad, ontology, matching, scoring, nlp, similarity

I. INTRODUCTION

For recruiters, seeking through all the resumes in order to create a shortlist for the candidate seekers is always a difficult and time-consuming occupation. Requirements for every job are different; for some candidate seekers, the experience is more important, for some, education background is more important and for some, skills are more important. Finding the right candidates within a big resume pool is not easy, especially when we search for specific abilities. The search process is one of the most time-consuming part and it can easily be automated.

In this paper, we create an intelligent system that ranks the candidates automatically and gives them a similarity score for the selected job. The system starts with extraction of the information from resumes, and by using the ontology we created, it ranks the candidates. When we began to build the system, we searched for an ontology designed especially for the human resource sector. However, we could not find the ontology that we were searching for and then we realized that we should design it ourselves. In the end, we created a detailed ontology for the sector from scratch to the best of our knowledge.

II. LITERATURE REVIEW

For information and entity extraction, used methods can be classified into two main classes which are rule-based methods and statistical methods. Moreover, relationships between entities can also be extracted in which part of speech (PoS) tags play an important role. These methods conclude that if input data is noisy, statistical methods do give better results since rule-based approaches are biased [1]. A new Hidden Markov Model proposed Maximum Entropy Hidden Markov Model which combines the features of Maximum-Entropy Models and Hidden Markov Models into a new model for information extraction and segmentation [2], [3]. Recent advances in Named Entity Recognition with Deep Learning Models show that in general, neural network models outperform feature-engineered models, while hybrid neural networks that combines characters and words generally outperform other representational choices [4].

Resumes that are used in this study are extracted from LinkedIn and kariyer.net. Since these resumes have a fix format, rule-based methods are used to extract information. Entities have been defined within the ontology. Relationships within the entities are also defined in the ontology.

Online job platforms and e-recruitment are trendy contemporary topics. Different kinds of approaches and systems have been constructed lately [5], [6], [7].

The main task of matching up job ads and resumes are based upon the usage of an ontology. Protege is one of the primary tool for designing an ontology. The ontology design, construction and Protege are explained in several detailed studies [8], [9], [10]. The main reason for using the ontology is to find the similarity scores. These scores depend on the structure of the constructed ontological model [11], [12]. Depending on the structure, the scores may vary. These scores are calculated by using the similarity functions. There are several different similarity functions and each of them is suitable for different applications [13], [14], [15].

III. SYSTEM OVERVIEW AND METHODS

A. System Architecture

The system build in this work aims to compute a statistical score for matching the given job ads and resumes according to the information extracted within them.

It consists of two parts; information extraction and the matcher. The information extraction is the first step of the process. For matching and giving a similarity score between the job ad and the resumes we should have all the information that we need. When different kinds of resume formats are considered, this work is quite challenging. After the information has been obtained from both job ad and resumes, the matching process is done by ruled-based methods and with the usage of the ontology. At the end of the process, a score for each resume can be calculated for each job ad.

Fig. 1 explains the pipeline of the system.

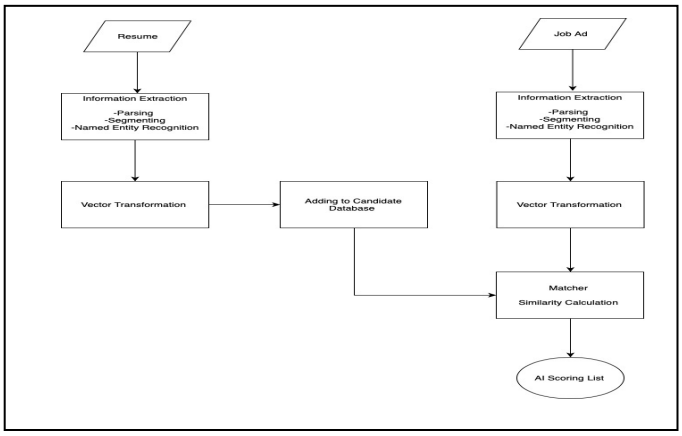

Fig. 1 System Architecture

B. *System Modeling and Methods*

1) Information Extraction: The first step is text extraction. Since the resumes can be of any document type such as .pdf, .docx, they must be extracted from the relevant file. This part does not need detailed explanation since it is just the extraction of text from a file by the use of tools. The next step is text pre-processing. To analyze any text data, it must be cleaned of unnecessary characters such as unicode symbols and it must be put into some standard format so that the process of information extraction can start. In our case, that standard format was lowercase characters in the English alphabet. Therefore, the irrelevant unicode characters were removed from the text. Then, information extraction from the resumes was done gradually. The first part of the information extraction from the resumes is segmentation. After the pre-processing part, for the information to be extracted, there must be a header for the information to align with. These are called segments. For example, the company name is aligned with an entry from experience segment. Therefore, segmentation is needed for accurate information extraction. After segmentation is completed, entity recognition is done. Labels must be put to the parts of the resume. For example, if "University of Texas" is seen within education segment, it needs to be labelled as university within related education entry. For this, ontology in the Neo4j database is used. To extract job titles within the experience segment, an algorithm which uses the obligation of having a person indicator in the title such as "java developer", "data analyst", "recruiter" was used. The last step of the information extraction from the resumes is semantic analysis. Within experience segment, each experience entry has a company, job title, date and a description which is optional. Moreover, companies represent an industry which is extracted with the help of the ontology, job titles do denote an expertise which belongs to the person related to the resume. This information is extracted with the help of the ontology and several algorithms and aligned with their respected parts. After these steps are finished, the resume is transformed into an object which is ready to be scored. The same steps are also done in order to extract information from the job advertisements. In a job advertisement, the client specifies the qualifications required for a candidate. This information is extracted mostly with the help of regular expressions since there is a pattern within sentences used to specify these qualities. For instance, desired majors, skills, or the related times can be seen in sentences such as "graduated from computer science, electrical engineer or related fields", "3+ years of experience in Java, Hibernate". Since the majors, skills and all the qualifications needed are defined within the ontology, majors of "computer science" and "electrical engineering" can be extracted as wanted majors. Moreover, from the second sentence, not only the skill "Java" is extracted, but the related time of "3+ years" is also extracted to be analyzed from the ontology and matched with the data from resume. Semantic Analysis is similar to the one applied for the resumes since the ontology is used for both. There are other similarities such as; in a resume, every experience entry has a duration and the skills extracted from the description of the mentioned entry is aligned with the duration. In the job advertisements, from the example above, whatever "3+ years" means within the ontology is matched with the Java and Hibernate skills. The main difference is the format of the returned object which is ready to be used in scoring.

2) Vector Transformation: We define our segments as education, experience, sector, skill and language. Each segment is formed by a different number of objects. For each segment, its vector space is constructed. Each segmental vector space has a different number of dimensions depending on their own object numbers. Objects are constructed by the extracted information from both resume and job ad. After construction of the vectors for the job ad and resume for every segment, we find the distance between every segment of the job ad and resume by comparing the distance between each object. Objects in the segments mathematically represent the dimensions of the vectors. The distance method for every dimension is constructed with the usage of the ontology.

Education segment is defined as follows; for resume:

$$S_r^1 = (S_{re}^1, S_{re}^2, S_{re}^3)$$

For job ad:

$$S_j^1 = (S_{je}^1, S_{je}^2, S_{je}^3)$$

$S_{re}^1, S_{re}^2, S_{re}^3$ are the objects of the education segment. They represent education level, major and institution name.

Experience segment is defined as follows; for resume:

$$S_r^2 = (S_{rx}^1, S_{rx}^2, S_{rx}^3)$$

For job ad:

$$S_j^2 = (S_{jx}^1, S_{jx}^2, S_{jx}^3)$$

$S_{rx}^1, S_{rx}^2, S_{rx}^3$ are the objects of the experience segment. They represent expertise, expertise level and company name.

Sector segment is defined as follows; for resume:

$$S_r^3 = (S_{rs}^1, S_{rs}^2)$$

For job ad:

$$S_j^3 = (S_{js}^1, S_{js}^2)$$

S_{rs}^1, S_{rs}^2 are the objects of the sector segment. They are present the sector name and its level.

Skill segment is defined as follows; for resume:

$$S_r^4 = (S_{rk}^1, S_{rk}^2)$$

For job ad:

$$S_j^4 = (S_{jk}^1, S_{jk}^2)$$

S_{rk}^1, S_{rk}^2 are the objects of the skill segment. They represent the skill name and its level.

And finally language segment is defined as follows; for resume:

$$S_r^5 = (S_{rl}^1)$$

$$S_j^5 = (S_{jl}^1)$$

S_{rl}^1 is the object of the language segment. It represents the language name.

The distance function for each segment is defined by the function Φ_n. For each n, the distance function is defined with a different method.

For example, for n=4, $\Phi_4(S_r^4, S_j^4)$ is calculated from the constructed skill ontology.

3) Matcher: After finding every Φ between job segments and resume segments, the final score (similarity score) is calculated by multiplying the determined weight of each segment γ_n. The determined weights are selected by the recruiters and depending on the search of the candidate seeker, for every job ad, all these weights are variables. The total sum of the weights should be equal to 1. Formula 1:

$$simscore = \sum_{n=1}^{5} \gamma_n * \Phi_n$$

C. Ontology

The main ontology consists of all the objects in the segments. Each possible object variable has been defined as nodes. Especially for expertise and skills, a very detailed and unique ontology has been created.

It has been seen that several different job titles can represent the same meaning. As a solution to this problem, 79 unique expertises are created. Every title is connected to at least one specific expertise with the algorithm that we created. The relations among expertises are created by our human resources specialists. The relations are the percentages that how similar they are to each other.

The next step was to create the skill ontology. The skill ontology is constructed by starting from expertises and created as a multi-inheritance ontology. The similarity scores between the skills are calculated using the cosines similarity function.

In total, 37493 unique nodes were created with 54632 defined relations.

D. Normalization

Every object can be written in numerous ways. Let's consider our company name; “Talentra İnsan Kaynakları”. One can say “Talentra”, “Talentra İnsan Kaynakları A.Ş.“, etc.

One of the most challenging part of this project was to reduce all the synonyms of a given name to only one main name. We call this process normalization.

In the created ontology, all of these names refer to only one node. In this case, before assigning the identity of the node, the normalization process is required. For every kind of object, a unique and tailor-made normalization method should be created. For every object, the tailor-made normalization algorithm is created.

For company normalization, the Jaro Winkler distance combined with Levenshtein distance is used. Jaro Winkler formula for two given strings (s_1, s_2) :

$$sim_w = sim_j + l*p(1 - sim_j)$$

where;

s_j : Jaro similarity between strings (s_1, s_2).

l: Length of common prefix at the start of the string up to a maximum of four characters.

p: Constant scaling factor for how much the score is adjusted upwards for having common prefixes and it is between 0 and 0.25.

We tried to find the optimum p-value for our company normalization problem. After several tests, the threshold is determined for every created normalization algorithm. If the string is under the threshold value, it is assigned automatically to its corresponding node value in the ontology. For every string that is over the threshold value, it is assigned to the “other” node. After that, the ontology will be updated manually and all the word entries which have been assigned to the “other” node will be added to the ontology periodically. Of course, putting every possible company name, university name, major name, etc. are extremely hard work. Collecting all the data for our matching problem is also one of the hardest parts. However, our ontology is being nourished and expanded periodically.

Table 1 demonstrates an example of company normalization. In this algorithm, our threshold for the distance value is 0.00900. For every value under it will be assigned automatically to the corresponding node and the rest will be assigned to the “other” node and they will be updated in the ontology.

TABLE 1—COMPANY NORMALIZATION

Unknown Company Name	Corresponding Node Name in the Ontology	Distance Value	Assigned Node
tradesoft	tradesoft ltd	0.0047	tradesoft ltd.
turkcell technology research and development	turkcell technology	0.0075	turkcell technology
gittigidiyor.com / ebay inc	gittigidiyor	0.0076	gittigidiyor
onedio bilişim yazılım medya teknoloji aş	onedio bilişim yazılım medya teknoloji sanayi ve ticaret a.ş.	0.0043	onedio bilişim yazılım medya teknoloji sanayi ve ticaret a.ş.
oyak yatırım	oyak	0.0088	oyak
pamukbank	pamukcu hukuk bürosu	0.0135	other
vepa group	pa group	0.0666	other

IV. TECHNOLOGIES

Another challenging part was to find the environment and the database for the creation and the storage of the ontology. Neo4j, a graph database, was used. All the ontology is created and stored in Neo4j. Right now, we have approximately 38.000 unique nodes and 54.632 relations between these nodes.

Fig. 2 shows the structure of a part of our skill ontology.

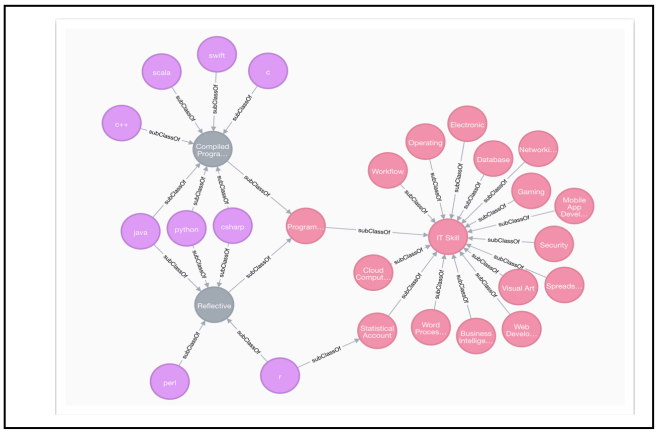

Fig. 2 Neo4j Skill Ontology

MongoDB is used as a storage for the vectorized candidate and job ad information.

Fig. 3 shows the structure of a part of our candidate and job ad storage.

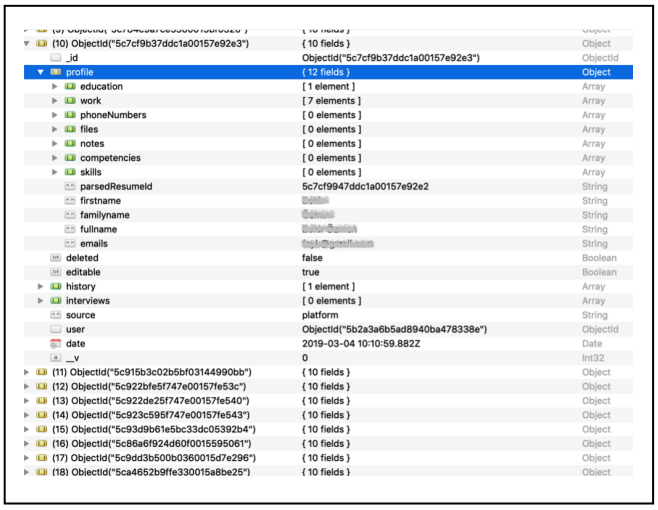

Fig. 3 MongoDB Candidate and Job Ad Storage

V. CASE STUDY

We ran our system for real case studies to test the validity of the artificial intelligence scoring and all the rest.

For a real open job position, the recruiters and the system started to search for the candidates simultaneously. After running the system, we made a shortlist of the top-ranked candidates.

A. Job Ad Requirements

- Graduated from Computer Science, Industrial Engineering, Mathematics, MIS, Statistics or related technical departments.
- 4+ years of experience in data science / data analytics in leading companies.
- Proficient in SQL, preferably MySQL, knowledge of R language is a plus.
- Understanding of fundamental statistical tools and techniques.
- Advanced in English language skills.

B. Candidates

Four candidates that have been associated with the job ad were used in this process to increase the test quality. The associations were previously done by the consultants that were working on the position. Therefore, the tested candidates are all related to the job.

C. Scoring the Candidates

All the scores for the defined segments (university type, major, university level, expertise, expertise level, company, sector, skill, language) are calculated by the distance formula that is defined in Formula 1. Distances are calculated using the created ontology. Segment scores are multiplied by their given weights to find the final matching scores.

TABLE 2—CANDIDATE SCORING

	Candidate 1	Candidate 2	Candidate 3	Candidate 4
University Type	0.75	1	1	1
Major	1	0.8	1	1
University Level	1	0	0	1
Expertise	1	1	1	0.2
Expertise Level	0	0.5	0	0.75
Company	1	1	0.9	1
Sector	1	1	0.9	0
Skill	1	0.8	1	0
Language				
Matching Score	0.85	0.80	0.79	0.71

This example was an ongoing process. After the interviews and investigations of the candidates that were on the shortlist of our recruiters, the two candidates picked for the elimination round by our customer were also the top two candidates in our list.

VI. CONCLUSION

After the creation of all the pipeline and the system, we are working on implementing an online platform for the usage of the recruiters. The recently formed system, with all NLP and matching algorithms, is a complicated one and needs longer time to calculate. During our work we encountered specific problems about the ontology creation, similarity calculation by ontology, normalization, etc. We found the solution to these problems by using our own knowledge and we created our own algorithms, methods and designs for them. In the future, the potential and the performance of the system should be improved.

ACKNOWLEDGMENT

This work is supported by Tübitak with a project number 7170337. We would like to thank them.

REFERENCES

1. Sarawagi, S., Information Extraction, Now, Boston, 2008
2. McCallum, A., Freitag, D., Pereira, F., Maximum Entropy Markov Models for Information Extraction and Segmentation 8, 2000
3. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 1989, pp.257-286
4. Yadav, V., Bethard, S., A Survey on Recent Advances in Named Entity Recognition from Deep Learning models, August 2018
5. Melike Sah, Vincent P Wade, Jinwu Li, Developing Knowledge Models of Social Media: A Case Study on LinkedIn, July 2014
6. D. Crow; J. DeSanto, A hybrid approach to concept extraction and recognition-based matching in the domain of human resources, Nov. 2004
7. Vijil Chenthamarakshan, Nanda Kambhatla, PROSPECT: A system for screening candidates for recruitment, January 2010
8. Matthew Horridge, A Practical Guide To Building OWL Ontologies Using Prot´eg´e 4 and CO-ODE Tools Edition 1.3 , March 2011
9. Jérôme Euzenat, Pavel Shvaiko, Ontology Matching , Springer-Verlag Berlin Heidelberg 2007
10. Keith Bradley, Rachael Rafter & Barry Smyth, Case-Based User Profiling for Content Personalisation, June 2000
11. Folami Alamudun, Tracy Anne Hammond, RésuMatcher: A Personalized Résumé-Job Matching System, Expert Systems with Applications, April 2016
12. Malgorzata Mochol, Ralf Heese, Radoslaw Oldakowski, Ontology-based Recruitment Process, July 2004
13. MingxinGan, XueDou, and RuiJiang, From Ontology to Semantic Similarity: Calculation of Ontology-Based Semantic Similarity, The Scientific World Journal, Volume 2013, Article ID 793091, 16 January 2013
14. Abdeslem Dennai, Sidi Mohamed Benslimane, A New Measure of the Calculation of Semantic Distance between Ontology Concepts, June 2015
15. Hisham Al-Mubaid, Hoa A. Nguyen, Measuring Semantic Similarity Between Biomedical Concepts Within Multiple Ontologies, August 2009
16. Iulia Andrada Ungureanu, Matchmaking Skills Using OWL Ontologies, April 2013
17. Henrik Eriksson, William E. Grosso, Ray W. Fergerson, John H Gennari, Knowledge Modeling at the Millennium, July 1999