Automatic Human Resources Ontology Generation from the Data of an E-Recruitment Platform

Sabrina Boudjedar¹, Sihem Bouhenniche¹, Hakim Mokeddem¹(✉), and Hamid Benachour²

¹ Ecole Nationale Supérieure d’Informatique, BP 68M, 16309 Oued-Smar, Algiers, Algeria
{fs_boudjedar, fs_bouhenniche, h_mokeddem}@esi.dz

² Laboratoire de Recherche En Intelligence Artificielle (LRIA), USTHB, El Alia BP 32, Bab Ezzouar, Algiers, Algeria
hamid.benachour@gmail.com

Abstract. Over the last decade, several e-recruitment platforms have been developed, allowing users to publish their professional information (training, work history, career summary, etc.). However, representing this huge quantity of knowledge still limited. In this work, we present a method based on community detection and natural language processing techniques in order to generate a human resources “HR” ontology. The data used in the generation process is user’s profiles retrieved from the Algerian e-recruitment platform Emploitic.com (www.emploitic.com). Data includes occupations, skills and professional domains. Our main contribution appears in the identification of new relationships between these concepts using community detection in each area of work. The generated ontology has hierarchical relationships between skills, professions and professional domains. In order to evaluate the relevance of this ontology, we used both the manual method with experts in human resources domain and the automatic method through comparisons with existing HR-ontologies. The evaluation has shown promising results.

Keywords: Ontology · E-recruitment · Occupation · Competence · Professional domain · Community detection

1 Introduction

The recent development of web technologies has revolutionized the world by offering permanent accessibility and high availability of data at the same time. Nevertheless, existing knowledge representation tools have shown their limits to fit with this huge revolution [7]. An ontology which is defined as “a formal and explicit specification of a shared conceptualization that is characterized by high semantic expressiveness” [2], can provide better knowledge representation that improves data exploitation.
In this context, using ontologies in human resources domain can be useful for both recruiters and candidates. In fact, an ontology can be used to create semantic search engines of job offers and candidate profiles. In addition, it can make easier for recruiters to select relevant candidates by matching between their profiles and job requirements.

The human resources domain is characterized by a huge quantity of concepts. For example, ESCO ontology (European Skills, Competences, qualifications and Occupations) created by the European Commission, provides 2942 occupations and 13.485 skills linked to these occupations [10]. In fact, creating a human resources ontology manually is a time-consuming task because it requires managing huge quantities of data that need to be processed and exploited automatically. In addition to that, human resources domain is a field that grows quickly. For this reason, using an automatic process to generate an HR ontology can help to manage its evolution. Thus, many research works studied the generation of HR ontologies such as AGOHRA (Automatic Generation of an Ontology for Human Resource Applications) [6] and HOLA (HR Ontology Learned Automatically) [1] -ontologies. However, these ontologies contain many information gaps and do not include many concepts.

In this paper, we propose a new method based on community detection and natural language processing techniques, for automatic generation of our HR-ontology using data provided by an Algerian e-recruitment platform named “Emploitic.com”. It is structured as follows: in Sect. 2, we present some related works regarding HR ontologies and their generation process. Then, we describe our ontology generation process and its evaluation in Sect. 3.

2 Related Works

Several research works have been performed in order to create a valid HR-ontology. In this section, we go through the most interesting HR-ontologies. We present their structures and generation process.

ESCO [10], an ontology resulting from a European project, which brings together occupations, skills and qualifications. This ontology was created manually in order to provide a common European repository to be used in recruitment. The ontology includes 2942 occupations, 13485 skills/competences and 9455 qualifications available in 26 different languages. This knowledge is organized in a structure of three pillars:

1. Occupation pillar organizes occupation concepts. Each occupation concept is accompanied by a description, its related skills and essential knowledge for this occupation.
2. Skill pillar organizes skill concepts. Each skill concept is accompanied by a description and its occupations.
3. Qualification pillar organizes qualification concepts. Qualifications displayed in ESCO come from databases of national qualifications that are owned and managed by the European Member States.
Authors of [6] proposed a semi-automatic generation approach of the HR-ontology AGOHRA based on job offers collected from the Internet. The generation process of this ontology follows these steps: firstly, a standardization on job offers is performed to separate offers according to the language used and also to remove duplicate offers. Secondly, occupations are extracted from job titles of each offer and only occupations having a significant frequency are maintained. Thirdly, skills linked to each occupation are extracted and represented according to the n-gram word format. In order to filter skills, a comparison is made with a prefix tree containing about 25,000 skills from social media profiles. Finally, the ontology is generated in RDF (Resource Description Framework) format where each universe contains a set of occupations and each occupation contains a set of skills. For its validation, a manual evaluation was performed by experts in HR domain. The detailed analysis showed results of a good quality (average precision of 0.79).

More recently, authors of [1] generated automatically a human resource ontology named HOLA from Professional Social-Network Data (LinkedIn). Authors used data extracted from user’s profiles and maintained only occupations and skills in the generation process of the HOLA ontology. This process follows these steps: first, data are extracted and represented in a graph where nodes represent either occupations or skills and edges represent the existing relationships between them. The weight of the edge between these two concepts represents the importance of the skill in a given occupation. Second, since data is entered by users, the noise level is probably very high. This step eliminates the noise and cleans data to keep only those that are useful in the creation. At the end, nodes are grouped into communities using Louvain algorithm in order to have more relationships in the graph. For the evaluation and validation, authors performed an automatic evaluation by doing a comparison between HOLA and ESCO. The evaluation results showed that HOLA contains more nodes and relations with 19,756 nodes and 154,259 edges.

3 Generation Process

AHROGA\textsuperscript{1} is an HR-ontology generated automatically from data of the Algerian e-recruitment platform emploitic.com. We can define our ontology through the following characteristics:

1. The Domain and The Goal of The Ontology
   Our ontology is a human resources ontology. This ontology aims to provide useful services that enable to evaluate whether a candidate has required skills for an occupation, suggest to a candidate how to highlight his skills and propose most relevant profiles for recruiters.

2. Data Used for Ontology Generation
   For the ontology generation, we use candidate profiles of emploitic.com, that include occupations, skills and professional domains.

\textsuperscript{1} Algerian Human Resources Ontology Generated Automatically.
3. Concepts (classes)

The concepts of our ontology are: Professional domain, Occupation, Community and Skill.

The generation process of our HR-Ontology is based on these three main aspects:

- Cleaning and validating data.
- Using professional domains in order to split our data in different clusters (representing sub-domains) and perform community detection on every cluster separately.
- Merging the resulting clusters into one cluster.

Our generation process is inspired from the process proposed by the authors of [2]. In addition to that, our generation process brings a new way to validate and evaluate the resulted ontology and its concepts using both automatic and manual methods. This process is described in the following pipeline (see Fig.1).

![Fig. 1. Generation process](image)

3.1 Extraction

This first step aims to extract necessary data to build AHROGA ontology and to perform preprocessing and cleaning.
**Data Extraction.** Our data corpus contains 300,000 profiles extracted in JSON format from *Emploitic.com* database. We identified for each profile three concepts that we consider essential to create AHROGA: Professional domain, occupation and skill.

There are many reasons to justify this choice. The most important are:

- The availability of these attributes in each profile.
- The strong semantic link between occupations, skills and professional domains.

**Data Preprocessing.** The data preprocessing phase is necessary in order to clean data and eliminate all types of noise.

Preprocessing concerns only occupations and skills because they are filled manually by users, which increases the probability of providing erroneous data, unlike professional domains which are proposed by the system.

**Occupations Preprocessing.** In order to standardize and validate occupations, we performed a preprocessing stage. This stage enables to identify the following problems:

- Existence of words/sentences that do not represent valid occupations, for example: curriculum vitae, without experience, 2019, never worked, etc.
- Use of locations inside occupations, for example: teacher at *bab ezzouar*, secretary *algiers*.
- Existence of occupations in multilingual, for example: web developer, *développeur web*.
- Using surnames and first names in occupations.
- Spelling errors.

In order to standardize and validate occupations, a preprocessing stage was done. This one goes through the following steps:

- Unifying the occupation format in lower case.
- Detecting the language of occupations to keep only ones written in French because 80% of job seekers use it.
- Removing special characters, accents, punctuation, numbers, stop words, verbs and proper names such as first and last names and Algerian localizations.
- Eliminating non-significant names that are very used by job seekers such as: CV, job search, profile, experience, curriculum vitae, internship.
- Grouping occupations that have the same stem.

After performing these tasks, many insignificant occupations which represent 3% of the data corpus were eliminated. In addition, we have identified through this step 290,458 occupations.
Skills Preprocessing. Preprocessing skills enables to identify two types of problems:

- Syntax problems such as wrong spelling.
- The validity problem of the skill itself. For example, internet is not a valid skill.

Among syntax problems we encountered we find: misspellings and multilingual skills, expressing skills in very long paragraphs (more than 6 words), using words that cannot be considered as skills like pc, internet, computer.

In order to fix these problems, we performed the same tasks used in occupations preprocessing.

3.2 Analysis

The purpose of the analysis stage is to extract existing relationships between data concepts. It goes through the three following steps.

Similar Jobs Detection. Similar jobs detection enables to create new relationships between occupations based on their syntax similarity in order to improve our final ontological structure. We link two occupations by using a distance calculated by Jaro-Winkler metric \([4]\), which is a string metric measuring an edit distance between two sequences. This distance is a variant of Jaro distance metric proposed by William E Winkler \([5]\). It is based on the number and order of common characters between the two strings. It uses the length \(P\) of the longest common prefix of two strings \(s\) and \(t\). Letting \(P' = \max(P, 4)\) we define:

\[
\text{Jaro} - \text{Winkler}(s, t) = \text{Jaro}(s, t) + \frac{P'}{10}(1 - \text{Jaro}(s, t))
\] (1)

For example, through this step we create syntax links between “Architect” and these occupations: “Urban planning architect”, “Project manager architect”, “Designer Architect”, “Architect designer”.

Initial Graph Generation. Before performing the community detection algorithm, we firstly represent cleaned data in a weighted graph. This graph is structured as follows:

- **Nodes**: to represent domains, occupations and skills.
- **Edges**: pairs of relations between two different nodes. There are three types of relations:
  - Domain-Occupation.
  - Occupation-Occupation.
  - Occupation – Skill.
- **Edge Weight**: a numerical value assigned to an edge representing the importance of an occupation in its domain or the importance of a skill in its occupation, it is calculated by the occurrence frequency.
Community Detection. We had initially 27 large separated domain graphs that include a lot of occupations. In order to create small communities that have high semantic links, we splitted occupations into smaller communities by performing community detection on each of the 27 graphs separately.

We have tested two different community detection algorithms: Louvain [3] and Label Propagation Algorithm (LPA) [11]. A performance comparison we performed have shown that Louvain algorithm gives more exact results. The performance metric is calculated by using a confusion matrix. Each element $m(i,j)$ of this matrix represents the number of nodes in the community $i$. The performance formula is as follows [8]:

$$I = -2 \frac{\sum_i \sum_j n \cdot m_{ij} \log \frac{n \cdot m_{ij}}{m_i + m_j}}{\sum_i \log \frac{n \cdot m_i}{n} + \sum_j \log \frac{m_j}{n}}$$

(2)

If the estimated community is correct, the measure takes 1 otherwise it takes 0.

In Table 1, we can see an example of obtained performances by applying Louvain and LPA algorithms on 3 different domains.

**Table 1.** Louvain and LPA Community detection performance on 3 domains

| Professional domain                        | Louvain algorithm | LPA algorithm |
|--------------------------------------------|-------------------|---------------|
| IT, information systems, internet         | 0.876             | 0.875         |
| Construction site, construction jobs, architecture | 0.859             | 0.844         |
| Engineering, studies, project, r&d        | 0.86              | 0.556         |

A second comparison based on a quantity and a quality of generated communities is shown in Table 2.

According to these results, we used Louvain algorithm for community detection because:

1. It generates partitions (set of communities) with better performance compared to LPA algorithm.
2. It generates fewer communities with better grouping of occupations.
3. It considers the importance of skills for each occupation as a parameter for grouping, which helps to group together occupations with significant numbers of common skills.
Table 2. Comparison between Louvain and LPA algorithms based on generated communities

| Professional domain                     | Louvain algorithm | LPA algorithm |
|-----------------------------------------|-------------------|---------------|
|                                         | Communities       | Communities   | Some generated communities | Some generated communities |
| IT, information systems, internet       | 16                | 38            | Community 0: developer, web developer, IT developer, software developer, programmer Community 1: software architect, software engineer | Community 0: IT developer, developer, web developer, Community 1: software architect |
| Construction site, construction jobs, architecture | 17                | 33            | Community 0: project director, project manager, works manager Community 1: architect, project architect, state architect | Community 0: project manager, site manager, architect, project manager, state architect Community 1: Project leader |

3.3 Generation

To enable computer programs to exploit the ontology and provide useful features to users, we transformed the resulting graph after the detection of communities into Ontology Web Language (OWL)\(^2\) through the creation of ontological components which are: axioms, classes, instances and relations.

Listing 1.1. Structure of ontology owl file

```
<owl:ObjectProperty rdf:about="#has_level">  
  <rdfs:domain rdf:resource="#Domain"/>  
  <rdfs:range rdf:resource="#Community"/>  
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#has_job">  
  <rdfs:domain rdf:resource="#Community"/>  
  <rdfs:range rdf:resource="#Job"/>  
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#has_skill">  
  <rdfs:domain rdf:resource="#Job"/>  
  <rdfs:domain rdf:resource="#Community"/>  
  <rdfs:range rdf:resource="#Skill"/>  
</owl:ObjectProperty>
```

\(^2\) https://www.w3.org/OWL/.
3.4 Validation

The validation is performed at the end of each previous step. It verifies that relationships and concepts are correct and checks if the generated ontology is consistent.

Confirmed Occupations. To validate occupations, we used only those having an occurrence frequency greater than a certain threshold, which is determined according to the percentage of valid occupations in each domain. Using a threshold of 1% allowed us to validate 43,110 profiles, which represent 15% of the total number of profiles.

Confirmed Skills. The purpose of this validation is to keep only correct skills. The approach that we followed is based on text classification. For that, we have firstly created the dataset by retrieving 5000 skills from profiles. Then, we used ESCO ontology API to get their descriptions. In addition, we have used Wikipedia for skills that did not have descriptions in ESCO. After that, we added a third attribute about the validity. The dataset that we have used in the skills classification is presented in the table below.

| Skill       | Description                                                                 | Validity |
|-------------|------------------------------------------------------------------------------|----------|
| Apache Maven| The tool Apache Maven is a software program to perform configuration identification, control, status accounting and audit of software during its development and maintenance. (from ESCO) | 1        |

As second step, we have used naïve Bayes classifier [9], which is recommended in text classification in order to classify skills into two different classes (valid, not valid). By applying this approach on our dataset, we got a model accuracy of 95% and we validated 77.10% of all skills.

Experts’ Validation: The purpose of this evaluation is to validate with experts the ontology concepts of three professional domains: (1) IT, information systems, internet (2) construction site, construction jobs, architecture and (3) telecommunications, networks.

This evaluation was conducted with five experts in the human resources domain.

The evaluation is based on a questionnaire that has three sections where each section represents one professional domain. Questions of each domain concern the generated communities and the relationships between occupations and skills.
After analyzing experts’ responses, we calculated for each occupation the average number of valid skills and the percentage of valid domain skills. The results are presented in the table below.

**Table 4. Experts’ evaluation**

| Domain                          | Occupation                   | Number of generated skills | Average number of valid skills | Valid domain skills |
|---------------------------------|------------------------------|-----------------------------|--------------------------------|---------------------|
| IT, information systems, internet | Computer science engineer    | 60                          | 43                             | 74%                 |
|                                 | Web Developer                | 37                          | 33                             |                     |
|                                 | Mobile Developer             | 20                          | 17                             |                     |
|                                 | Database Administrator       | 20                          | 9                              |                     |
| Construction site, construction jobs, architecture | Urban Architects            | 8                           | 4                              | 64%                 |
|                                 | Civil engineer               | 14                          | 9                              |                     |
|                                 | Planning engineer            | 13                          | 10                             |                     |
|                                 | Site supervisor              | 10                          | 6                              |                     |
| Telecommunications, networks    | Network engineer             | 61                          | 41                             | 70%                 |
|                                 | Telecommunication systems engineer | 45                        | 32                             |                     |

We then calculated for each domain the percentage of community validation. The results are presented in the figure below.

**Fig. 2. Community validation**
3.5 Evaluation

In order to evaluate the quality of our results, a comparison of our ontology was made with ESCO ontology. It is based on common concepts between the two ontologies. We have found that 68.20% of total skills and 73.61% of total occupations generated using our method exist in ESCO, which affirm that these instances are valid.

In order to ensure that our results are relevant, we applied the same approach used to generate HOLA ontology on our dataset and we compared it with our approach. (See Table 5)

|                      | HOLA                  | AHROGA               |
|----------------------|-----------------------|----------------------|
| Number of communities of level 1 | 20                    | 27                   |
| Number of communities of level 2 | 108                   | 392                  |
| Occupations of the same communities | Dentist doctor, veterinary doctor, bailiff, legal officer, personal representative, call center operator, waiter, receptionist, microbiologist, technical officer, professor, operator, nurse, machine operator | Dentist doctor, veterinarian doctor, doctor |

This comparison shows that our approach performed better compared to HOLA’s approach. Using professional domains as a concept in AHROGA allows better grouping of occupations in communities. For example, two occupations which are not in the same domain but share the same skills (depending on users) are assigned to the same community with HOLA approach.

4 Conclusion and Future Work

In this paper, we have proposed an automatic ontology generation method using data of the Algerian e-recruitment platform Emploitic.com. From 300,000 profiles collected, we have retrieved common data to build a structured hierarchy of concepts representing each professional domain. Our dataset contained many noisy data, we have started by a preprocessing step in order to clean it and keep only valid data. Then, we have represented these cleaned data in a weighted graph, on which we have applied the community detection using Louvain algorithm in order to create new relationships between concepts based on shared skills. Finally, this hierarchical structure was represented in OWL format in order to be used in useful use cases. For its evaluation and validation, we have used both of manual
methods like validating ontology concepts by human resources domain experts, and automatic methods like comparing our ontology concepts with ESCO.

AHROGA contains 13606 instances and 113612 relationships, including 27 domains, 716 communities, 1437 occupations and 11,426 skills. The approach we proposed is generic and can be applied on every dataset that contains occupations, skills and professional domains.

As future works, we have proposed to make several improvements on our generation pipeline. Firstly, in order to enrich the ontology, we have planned to use a combination of job offers and profiles data rather than only using profiles data. Secondly, we believe that we can enrich also our ontology by adding multilingual instances since that most of data are provided in three languages (French, Arabic and English) and by considering more concepts in the creation of our ontology, such as professional experience and education qualifications. Finally, useful services based on the ontology need to be implemented. For example, matching automatically job offers and candidate profiles.

References

1. Alfonso-Hermelo, D., Langlais, P., Bourg, L.: Automatically learning a human-resource ontology from professional social-network data. In: Meurs, M.-J., Rudzicz, F. (eds.) Canadian AI 2019. LNCS (LNAI), vol. 11489, pp. 132–145. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-18305-9_11
2. Bedini, I., Nguyen, B.: Automatic ontology generation: state of the art. PRiSM Laboratory Technical report. University of Versailles (2007)
3. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. 2008(10), P10008 (2008)
4. Cohen, W.W., Ravikumar, P., Fienberg, S.E., et al.: A comparison of string distance metrics for name-matching tasks. In: IIWeb 2003, pp. 73–78 (2003)
5. Jaro, M.A.: Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. J. Am. Stat. Assoc. 84(406), 414–420 (1989)
6. Kessler, R., Lapalme, G.: Agohra: generation of an ontology in the field of human resources. Traitement Automatique des Langues 58(1), 39–62 (2017)
7. Omelayenko, B., Fensel, D.: A two-layered integration approach for product information in B2B E-commerce. In: Bauknecht, K., Madria, S.K., Pernul, G. (eds.) EC-Web 2001. LNCS, vol. 2115, pp. 226–239. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44700-8_22
8. Orman, G.K., Labatut, V.: A comparison of community detection algorithms on artificial networks. In: Gama, J., Costa, V.S., Jorge, A.M., Brazdil, P.B. (eds.) DS 2009. LNCS (LNAI), vol. 5808, pp. 242–256. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04747-3_20
9. Rish, I., et al.: An empirical study of the Naive Bayes classifier. In: IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, vol. 3, pp. 41–46 (2001)
10. le Vrang, M., Papantoniou, A., Pauwels, E., Fannes, P., Vandensteen, D., De Smedt, J.: ESCO: Boosting job matching in Europe with semantic interoperability. Computer 47(10), 57–64 (2014)

11. Xie, J., Szymanski, B.K., Liu, X.: SLPA: uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In: 2011 IEEE 11th International Conference on Data Mining Workshops, pp. 344–349. IEEE (2011)