Automatically Learning a Human-Resource Ontology from Professional Social-Network Data

David Alfonso-Hermelo[1(✉)], Philippe Langlais[1], and Ludovic Bourg[2]

[1] Université de Montréal, Montreal, Quebec H3C 3J7, Canada
david.alfonso.hermelo@umontreal.ca, felipe@iro.umontreal.ca
[2] LittleBIGJob, Montreal, Quebec H3B 4W5, Canada
ludovic.bourg@lorenzandhamilton.com
http://rali.iro.umontreal.ca/rali/

Abstract. In this work, we build an ontology (automatically learned) in the domain of Human Ressources by using a simple, efficient and undemanding procedure. Our principal challenge is to tackle the problem of automatically grouping human-provided job titles into a hierarchy and by similarity (as they are presented in human-made HR ontologies). We use the Louvain algorithm, a greedy optimization method that, given a sufficient amount of data, interconnects domain-specific jobs that have more skills in common than jobs from different domains. In our case, we used publicly available profiles from LinkedIn (written in English by users in France). An automatic evaluation was performed and shows that the resulting ontology is similar in size and structure to ESCO (one of the most complete human-made ontology for HR). The whole procedure allows recruitment professionals to easily generate and update this ontology with virtually no human intervention.

Keywords: Automatic Ontology Learning · E-recruitment · Occupations and skills ontology · Community detection · Relational model · Natural language processing · Taxonomy · Data mining · Artificial intelligence

1 Introduction

In computer science and according to the authors of [9], an ontology is a formal, explicit specification of a shared conceptualization that is characterized by high semantic expressiveness. It typically encompasses sectors of knowledge (more or less limited) in a fast computer-accessible way. With the development of AI technology mimicking human understanding of natural language, the number of technologies that can benefit from ontology-based artificial understanding has increased greatly. The problem is that, in addition to requiring multiple domain specialists, the elaboration of an ontology is labor and cost intensive.

M.-J. Meurs and F. Rudzicz (Eds.): Canadian AI 2019, LNAI 11489, pp. 132–145, 2019.
https://doi.org/10.1007/978-3-030-18305-9_11

In spite of these obstacles, multiple ontologies have been manually created in the domain of Human-Resources (HR), as, for instance, the ESCO ontology that we describe in Sect. 2.2. Those ontologies encode a hierarchy of categories and relations between occupations and skills. They serve as administrative tools in the recruitment domain to conduct operations such as grasping the skills needed for a given occupation, pinpointing the right jobs for a given set of skills, or guiding post assignation.

Still, even the most covering and detailed of these ontologies is incomplete and even if it is an important tool for HR workers and guidance counsellors, there are usually information gaps: brands are usually absent from skills (e.g., using "*text processor program*" instead of "*Microsoft Word*"), some occupation categories are constantly evolving, occupations are generally limited to one standardized denomination and so on. This was our first motivation to use a different strategy.

The fact that governments and institutions have gathered the necessary resources to produce human-made HR ontologies leads us to believe that there is great interest in having, updating and expanding this domain-specific resource.

In this paper, we explore the first version of a system designed to build an HR Ontology using the publicly available user data of a professional social network (LinkedIn in this case). This system is named HOLA for **H**R **O**ntology **L**earned **A**utomatically. In Sect. 2, we take a quick look at the most relevant related works. We present our system in Sect. 3. We describe the evaluation we conducted in Sect. 4 and conclude in Sect. 5.

2 Related Work

2.1 Automatic Ontology Learning

Every Automatic Ontology Learning (AOL) research approach we have encountered in the literature is divided in two main parts: a semantic triple[1] extractor from free text, and an ontology generator (e.g., the works of [13], or [5]). The appeal of unstructured data is evident: it is widely available, abundant and far richer than structured data.

2.2 HR Ontologies

As mentioned in Sect. 1, there have been some notable efforts to build human-made taxonomies and ontologies in the HR sector. We will mention two of the largest and publicly available ontologies: ROME and ESCO.

ROME is the Operational Directory of Trades and Jobs (Répertoire Opérationnel des Métiers et des Emplois), an HR ontology in French whose last version dates to 2009. It contains 532 job titles and 12,099 skills (divided in competences, activities and environments).

[1] Three entities that codify a statement in the form of subject-predicate-object expressions.

ESCO is the European Skills/Competences, qualifications and Occupations ontology project developed by the European Commission since 2010 and described in [17]. In its current state, ESCO stands as the most comprehensive free-of-charge ontology with regard to the number of languages (26), the number of job titles (2,942), the number of skills/competences and knowledges (13,485) and the number of job title variants (23,281 in English). ESCO is intended to serve as a multicultural and multilingual unifying resource between job seekers and job providers.

We now briefly present the structure of ESCO since our system design was mostly inspired by it. ESCO shows three different types of concepts: *occupation*, *knowledge* and *skill/competence*. All concepts have a preferred label, a list of alternative labels, a human-understandable description and are uniquely identified by a Uniform Ressource Identifier (URI). The occupation concepts are classified with a deep hierarchy based on the ISCO (International Standard Classification of Occupations) structure[2]. An excerpt of ESCO is provided in Table 1.

Table 1. Excerpt of ESCO entries.

Concept type	ISCO group	Preferred label	Alternative labels	Description
Occupation	2512	Software analyst	Software analysts, software requirements analyst, etc.	Software analysts elicit and prioritise user [...]
Skill		Load animals for transportation	Load animals safely, load animal, etc.	Load and unload animals safely into containers or [...]
Knowledge		Animal transport regulations	Animal welfare during transportation, etc.	The legal requirements relating to safe and [...]

2.3 Automatic HR Ontology Learning

The authors of [12] use automatic extraction methods on a large corpus of job offers to obtain job titles and their corresponding skills. After the extraction, the job titles are grouped into 27 clusters that represent the domain-specific categories. The final result is an ontology generated without human supervision, containing 440 job titles (128 in English, 312 in French) and 6,226 skills (4,059 in English, 2,167 in French). This system describes very interesting methods for HR AOL, but because of the unstructured nature of job offer texts, the number of reliable extracted data is relatively small and the resulting ontology is therefore incomplete.

[2] The hierarchy for the skill/competence and knowledge concepts is not yet available in full.

Google has also been developing an occupation and skill ontology for their recently released service Google Jobs [15]. According to the author of the blog, this ontology is a machine learning enrichment of the O*NET Standard Occupational Classification combined with a proprietary skill ontology containing 50,000 skills. The result is an HR ontology of 30 job categories, 1,100 job 'families' and 250,000 job titles; all connected to specific sets of skills from the skill ontology. Unfortunately, this ontology is not available to the public.

2.4 Ontology Evaluation

One of the problems in AOL is the evaluation of the resulting ontology. Building on the works of the authors of [4], we present the following grouping of the evaluation approaches usually used in AOL, for which there are four main strategies.

A ***human inspection*** is the most commonly used strategy. According to some authors (like [14] and [10]) three human annotators (not necessarily experts) can offer a good precision score if the sample is representative and if a protocol is uniformly followed.

A ***comparison to a gold standard*** measures the overlap between a candidate ontology and a reference one, at the risk of being penalized by an incomplete reference. Some researchers (see, for instance, [8] and [7]) go further and compare the *structural* similarity between ontologies. This measure depends on having an ontology and a gold standard whose structure is similar enough to be compared.

When an ontology is designed with a task in mind, a ***benchmark assessment*** measures its ability to perform this task (see, for instance, [18]).

It is also possible to use ***quality scores*** based on a collection of indicative proxies. The authors of [6] introduce a tool to detect inconsistencies, redundancies and incompleteness in taxonomies. In [1] the authors present a ranking system that uses 4 metrics (class match, density, semantic similarity, and betweenness) to compare multiple ontologies and rank them according to a quality score.

One such system, OntoQA [16] caught our attention. It is based on a structure-oriented score that uses 8 different metrics. We will analyze it in some details in Sect. 4.

3 System Design

The aim of Hola is to automatically learn a domain-specific ontology for the HR sector. The human-made ontology already encompass the organizational knowledge HR specialists can bring forth. In those ontologies however, it is common for the more specific information to go unnoticed (i.e. the latest professional tools; the rare, new or specialized job titles; the diversification of skills; etc.).

Our claim is that professionals have a greater understanding of their own occupational fields and, by extension, of the kind of detailed knowledge our ideal ontology seeks to capture. One way to do this is to use professional social networks. Just like HR ontologies, these social networks are meant as a link

between hiring entities and professionals. They use semi-structured data to allow an easy data query and they allow unlimited modifications and updates in order to be constantly up-to-date.

One such network is LinkedIn, for which we have collected the occupation and skill data from publicly available profiles made in France in 2016 and written in English. It should be fairly simple to continuously update this LinkedIn data collection and transform HOLA into a dynamic ontology. This is however left as future work.

3.1 Input Data, Term and Triple Extraction

As we discussed in Sect. 2, the state-of-the-art AOL projects we have encountered mine unstructured data. Because mining free text is a noisy process, we use the semi-structured data of LinkedIn user profiles.

Some HR ontologies make a distinction between different types of skills (i.e., skill, competence, qualification, knowledge, etc., see Table 1). To simplify the task for its users, LinkedIn allows listing skills in a separate section of their profile. LinkedIn does not expect its users to be HR specialists capable of differentiating the distinct types of skills, and neither do we. This is why, of all the available LinkedIn profile data, we only extract the *job title* and *skill* data for our use in HOLA.

The HOLA system depends on a fairly simple ontology design inspired by ESCO described in Subsect. 2.2. The ontology is encoded as a graph characterized as:

- **Node**: Fundamental unit of the graph. In HOLA, every concept, whether it is a job title, a skill or a community is represented as a node.
- **Community**: Cluster or group of nodes densely connected. Each node is assigned to the cluster with which it shares the most connections.
- **Edge**: Ordered pair of nodes. In HOLA, there are four different types of edges: job title - skill, community - job title, community - skill, community - community.
- **Edge weight**: Numerical value assigned to an edge. In HOLA, the edge weight is either null (for community - community, community - job title and community - skill edges) or it represents the skill frequency (for job title - skill edges).

At this point, our 2016 corpus has 5,110,768 profiles. Each profile contains an average of 1.1 job title and 2.1 skills. We can see an example of a LinkedIn profile's data in Table 2.

Of these LinkedIn profiles, more than half (55.9% of our corpus) have no job titles[3] and 18.8% have multiple job titles. LinkedIn profiles are presented with a Resumé structure that allows and encourages the chronological listing of each job title, instead of replacing it with the most current. The profiles with

[3] These profiles most probably correspond to LinkedIn users who have not yet any experience in the professional world or who have not updated their profile.

Table 2. Excerpt of a LinkedIn profile.

Country code	FR	**Language code**	en
Personal Branding Pitch	Prove organizational, analytical and process oriented skills and can demonstrate the ability to serve clients in a professional manner [...]		
Experience	– Experience 2 • **Function** : SAP SMB Delivery Manager • **Start Date** : 2014-04 • **Missions** : SAP ByD and B1 Projects Delivery Improve Team [...] – Experience 1 • **Function** : SAP Technical Lead • **Start Date** : 2010-07 • **End Date** : 2013-04 • **Missions** : Functionnal and Technical implementation of [...]		
Skills	SAP, ERP, Microsoft Office, Business Analysis, IT Strategy [...]		

multiple job titles, as the one in Table 2, pose a problem because of the difficulty to pinpoint what skills correspond to which job title. For this first version of HOLA, we chose to limit ourselves to the profiles containing exactly one job title and at least one skill. The implicit assumption is that the skills given in these profiles are pertinent to that single job title. By doing so, we are left with 1,293,082 profiles (25.3% of our initial corpus).

3.2 Filtering

Many of the job titles and skills appear several times in our profiles. After suppressing all duplicates, we are left with 57,655 job title nodes and 50,557 skill nodes, and 795,044 relationship edges linking them. Unfortunately, as it is often the case with real user data, we observe a rather high level of noise in both job title and skill nodes. A careful inspection led us to distinguish five types of noise:

- Lack of usefulness: the skill is not useful to the job title to which it is associated (e.g., skill "*powerpoint*" for job title "*Sushi Chef*").
- Foreign words: a word or set of words written in a different language than the one specified in the metadata (e.g., "中国市场负责人 *Marketing Development*"[4]).
- Occupational over-specification: the job title is exclusively used inside a particular company and is not recognized by a wider group (e.g., "*sandwich artist at Subway Boulevard Voltaire*" instead of "*fast-food cook*").
- Incoherent content: nonsensical or misplaced job titles or skills (e.g., "*XXxxxXX*").
- Spelling error: (e.g., "*Research Ingenieer*").

[4] Loosely translated as: "head of China market/person who is in charge of the Chinese market".

To filter out noisy job titles and skills, we use five heuristics[5] that are very different in nature.

- **Language detector**: By using Google's language detection library *langdetect*[6] we detect the language for each job title and skill and select those who correspond to our targeted language (English). This heuristic leaves out 56.8% of the job titles and 46.69% of the skills.
- **Gibberish detector**: We remove meaningless strings of characters (e.g., *XXXXXX*, *****). This heuristic filters out 0.08% of the job titles and 1.5% of the skills.
- **Token counter**: In human-made ontologies, job titles and skills rarely exceed 5 tokens (e.g., less than 1% of ESCO job titles have 6 or more tokens). We speculate that filtering out the nodes containing 6 or more tokens (not counting stop-words), will leave out most over-specific job titles (e.g., "*Internship as Assistant Product Manager and Business Developer (3D Printing Systems)*", "*Consultant food and beverage and culinary for hotels and restaurants opening*", "*Senior Administrative Assistant European Sales Management*"). This heuristic leaves out 5% of the job titles and 2.31% of the skills.
- **2 in 1 label detector**: As we explained in Subsect. 3.1, we select the profiles containing only one job title. Nevertheless, we sometimes observe that those still contain two (or more) job titles in the form of an enumeration (e.g., "*Owner/Partner & Managing Director*", "*director / film researcher*"), which defeats our goal. If we detect this type of job title, we remove it and its connected skill nodes. This heuristic eliminates 8.27% of the job titles and 2.91% of the skills.
- **Isolated trees eliminator**: We want to avoid capturing job titles whose skills are exclusive to one job title and that are so specialized that they only appear once in the whole dataset. We thus filter out a) the trees isolated from the rest of the ontology forest, and b) the job titles and skills appearing in only one or two profiles. This heuristic leaves out 4.98% of the job titles and 35.78% of the skills.

By applying the filters, we obtain a data reduction of 75.16% of the job title, 89.26% of the skill nodes, and 80.60% of the edges, leaving our graph with 14,326 job title nodes, 5,430 skill nodes and 154,258 edges. Figure 1 shows in gray the concepts and relations that are removed by our filters and, in bold and red, the ones remaining.

Evaluating such a pipeline of filters is, of course, tricky. But since our goal was to eradicate not only aberrant noise but also spelling errors, gender versus singular-plural form variations and the like, we can measure the percentage of nodes in the filtered resource that have a label close to the label of another node. Intuitively, a good filter would significantly reduce the number of close labels. We used a simple Levenshtein edit-distance at the character level to detect,

[5] We did not include a spelling checker because the data contains terminology, acronyms, neologisms and variations that do not correspond to dictionary forms.

[6] http://code.google.com/p/language-detection/.

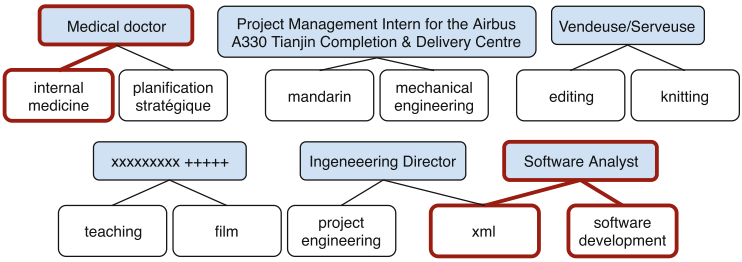

Fig. 1. Excerpt of job title and skill nodes in the graph before (in gray) and after applying the filters (in bold and red). (Color figure online)

for each node, neighbors within a distance between 1 and 3. Before filtering, 39% of the nodes have a neighbor. After filtering, this percentage falls to less than 33% (as Table 3 shows). This represents a small decrease after eliminating so many nodes during filtering, which goes against out initial intuition. After analyzing the captured neighbors, we observe that the greater part of what we managed to capture are node labels with distant meanings who happen to have similar spellings (e.g., "*heat**l**ing*" and "*heal**i**ng*", "***Ge**ologist*" and "***Ur**ologist*", "***Tut**or*" and "*T**en**or*") and a rare amount of the variations we aimed (and failed) to eradicate (e.g., "*System**s** **e**ngineer*" and "*System **E**ngineer*", "*Nav**y** Officer*" and "*Nav**al** Officer*"). We have no automatic way to determine which is which but since our heuristics focused on eliminating the noisy nodes and not the paronymic nodes, the decrease of neighbors from 39 to 33% suggests that there actually is a measurable reduction of noise.

3.3 Community Identification Process

At this point we have obtained a graph composed of skill and job title nodes connected between them. This graph still lacks the necessary structure to be a functional ontology. The nodes still need to be grouped according to their similarities and those groups must be placed in a hierarchy. This can be achieved with the Louvain algorithm, a community identification algorithm presented in [3]. It is a non-deterministic algorithm that is considered one of the fastest (with a complexity of $O(n \log n)$ according to the analysis presented in [11]).

Table 3. Number of nodes in HOLA at a given edit-distance of a neighbor after filtering.

Edit distance	Nb of nodes after filtering	Percentage of nodes after filtering	Example	
1	2365	12%	**Heating**	Healing
1-2	4208	21.3%	**Systems engineer**	System engineer
1-3	6505	32.9%	**Java engineer**	Naval engineer

In a nutshell, each node of the graph is assigned a different community label so there are as many communities as there are nodes. Then, for each node i we consider the neighbors j of i. We calculate the modularity gain of removing i from its community and placing it in the community of j. The node i is then placed in the community for which the modularity gain is maximum and positive. If no positive gain is possible, i stays in its original community.

This reduces the total amount community labels from one per node to one per community cluster. This process is applied repeatedly and sequentially for all nodes until there is no more modularity gain. After calculating the modularity gain for all the nodes, we iterate the whole process but, this time, reassigning a community label to each cluster of nodes instead of each node. For each iteration over i, we obtain a hierarchical ordering of the communities where the communities after the first iteration can be found inside the communities found after the second iteration and so forth.

After applying the Louvain algorithm to our graph, we obtain a 3-level classification for each node. At the first, less restrictive level we have 6 *parent* communities (level-1), these can be further divided into 8 *children* communities (level-2), which can be even further divided into 88 *grand-children* communities (level-3).

The community detection algorithm does not offer a perfect match to the usual domain segmentation and hierarchy. The main reason is that the current classification of occupations is the result of centuries of arbitrary categorization while the community detection is purely based on the shared adjacent nodes (the shared skills to determine job title categories, and vice versa). Nevertheless, if we observe the nodes composing each community and sub-community we can see in Table 4 that they show certain affinities.

Table 4. Excerpt of HOLA nodes and edges grouped in the 3rd-level community 6.6.17.

Job Titles	Yoga Teacher, Health Coach, complementary therapist, homeopath, Ayurveda Therapist, Mandataire Syner J Health, body psychotherapist, Osteopath
Skills	Physical therapy, smoking cessation, exercise physiology, energy healing, craniosacral therapy, wellness coaching, body massage, hatha yoga, holistic health
Edges	Ayurveda Therapist - wellness coaching, Health Coach - wellness coaching, Osteopath - craniosacral therapy, Ayurveda Therapist - holistic health

At this point, the outcome we obtain is a graph of 14,325 job title nodes, 5,431 skills, a 3-level node hierarchy and 154,259 job title-skill edges. Even after filtering, in pure numbers of nodes and edges, this is still more than ESCO or ROME (as we can see in Table 5).

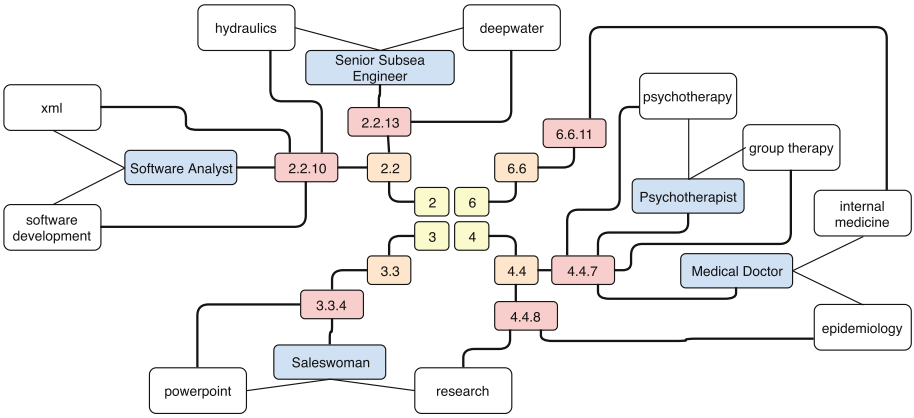

Fig. 2. Excerpt of the HOLA ontology.

In Fig. 2, we observe that the job title and skill nodes in our graph are connected to multiple community nodes, which are ordered in a hierarchical way. In this figure, the nodes in white represent skills, in blue are job titles, in red are level-3 communities, in orange are level-2 communities, in yellow are level-1 communities. We feel comfortable calling this graph an ontology[7].

4 Evaluation

As mentioned in Sect. 2.4 we opt for an evaluation based on quality scores and retain the OntoQA System described in [16] for comparing the ESCO ontology and the HOLA one. The OntoQA system is based on 8 metrics, but only 5 are actually useful in our setting (that is, for comparing ontologies):

- **Relationship Richness**: ratio of the number of non-inheritance relationships, divided by the total number of relationships defined in the schema. The closer this normalized score is to 1.0, the more diverse types of relations in the ontology. Compared to HOLA, ESCO uses three more non-inheritance relationships (*Alternative Label*, *Description*, *Skill Type*), which we do not implement yet. This is why we expect ESCO to surpass HOLA for this score. Yet, since we can successfully emulate many non-inheritance relationships we anticipate a high score approaching 1.0.
- **Class Richness**[8]: ratio of the number of classes in the ontology divided by the total number of classes defined in the ontology schema (non-empty classes/all classes). HR ontologies usually get the maximum score of 1.0 since

[7] The complete HOLA graph is available for a dynamic consultation at http://www-etud.iro.umontreal.ca/~alfonsda/project/holaOntology/index.html.

[8] In our case, the classes correspond to the automatically detected communities.

the HR classification evolves slowly and, therefore, it is rare to see a schema class that is not represented in the ontology[9].

- **Attribute Richness**: average number of attributes (nodes and edges) per class. The assumption behind this metric is that more attributes equals more knowledge conveyed. Since we have quite a high number of nodes and edges for an HR ontology, we expect to get a higher score than the human-made ontology.
- **Inheritance Richness**: average number of subclasses per class. A low Inheritance Richness score indicates a deep (or vertical) ontology that covers a specific domain in a detailed manner, while a high value denotes a shallow (or horizontal) ontology indicating a lower level of details. LinkedIn profiles tend to denominate specific job titles and skills but cover many professional sectors. We aim to get an ontology that would mirror these qualities and produce a corresponding middle ground score: not so high as to indicate a broad and general classification and not so low as to indicate a restrictive coverage.
- **Cohesion**: number of connected components. If the ontology is a connected graph, then the Cohesion value should be 1, if not, it should have a value of 2 or more. Since we specifically eliminated isolated data (as described in Subsect. 3.2) we expect to get a score of 1.

Evidently, there is still room for improvement (particularly for the Relationship Richness score) but, in general, we are quite pleased with the results obtained and reported in Table 5. These depict an ontology coinciding with the properties we aim to obtain: whose structure profile is (metrically) comparable to one of a human-made ontology, conveying more knowledge than its counterparts and indicating a compromise in inheritance coverage.

Table 5. OntoQA metric scores obtained for ESCO, ROME and Hola.

Ontology name	Nodes	Edges	Relationship Richness	Class Richness	Attribute Richness	Inheritance Richness	Cohesion
ESCO	16,427	114,406	**0.929**	1.0	11,894	54	1
ROME	12,631	31,099	0.65	1.0	2,082	5	1
Hola	**19,756**	**154,259**	0.909	1.0	**29,002**	16	1

Another point of comparison between the two ontologies is the number of job titles and skills they share. Actually, we were surprised to measure that only 286 job titles and 300 skills are shared among those ontologies, which represents only 3% of Hola and 3.6% of ESCO nodes. We can see some examples in Table 6.

[9] Some ontologies from other domains evolve so quickly that conventional classes at the time of the schema conception might become obsolete, e.g., the smartphone sensor network ontologies analyzed in [2], like OntoSensor or CESN have a Class Richness score of 0.59 and 0.71 respectively.

While we anticipated a larger intersection, we feel this is a good sign. Our aim is not to replicate the ESCO ontology (or any other ontology) but to imitate its structure while using more dynamic data in nature[10].

Table 6. Excerpt of job titles and skills appearing in ESCO or HOLA. The nodes common to both ontologies appear in bold.

ESCO		HOLA	
Job Title	Skills	Job Title	Skills
webmaster	**css**, **php**, python (computer programming), pascal (computer programming), javascript, etc.	**webmaster**	html, xhtml, html5, **css**, css3, **php**, xml, python, mysql, flash, etc.
specialised doctor	general surgery, plastic surgery, radiology, **emergency medicine**, clinical biology, cardiology, etc.	medical doctor	medicine, surgery, emr, **emergency medicine**, clinical research, etc.

5 Conclusion and Future Work

We proposed a procedure to create an HR Ontology automatically, using a selection of the public semi-structured information found in LinkedIn. Even though there is still much to improve, this is a first step towards a system that resembles human-made HR ontologies but needs less time and effort. We report encouraging results both in terms of number of nodes, and level of structuring. While using user profiles to populate an ontology gives the hope to obtain a more up-to-date ontology for various activities in the recruitment domain, we showed that dealing with real user data is in itself a challenge that needs to be addressed.

There are several avenues of this work we plan to address in future research. First, we proposed filters in the HOLA pipeline that are not without limits: we do not detect all the noise in the data, and we do remove profiles that could contribute to populate the HOLA ontology. In particular, we discarded profiles with more than one professional experience, in order to associate skills to the mentioned job title. We should definitely leverage the data we discarded in this first version of our system. Second, we believe we could automatically identify descriptions and variants of job titles and skills, very similarly to what is manually done in the ESCO ontology. Third, the community detection algorithm is not naming a found community, a challenge we want to look at. Last, one motivation in conducting this work was to be able to update and increase the ontology with new job titles or skills emerging from professional social networks. This remains to be investigated.

[10] Even thought the enrichment of the ESCO ontology was not among this work's objectives, we do not discard this possible application of the HOLA procedure.

References

1. Alani, H., Brewster, C., Shadbolt, N.: Ranking ontologies with AKTiveRank. In: Cruz, I., et al. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 1–15. Springer, Heidelberg (2006). https://doi.org/10.1007/11926078_1
2. Ali, S., Khusro, S., Ullah, I., Khan, A., Khan, I.: Smartontosensor: ontology for semantic interpretation of smartphone sensors data for context-aware applications. J. Sens. **2017** (2017)
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. Brank, J., Grobelnik, M., Mladenić, D.: A survey of ontology evaluation techniques. In: Slovenian KDD Conference (2005)
5. Cimiano, P., Völker, J.: Text2Onto. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 227–238. Springer, Heidelberg (2005). https://doi.org/10.1007/11428817_21
6. Corcho, Ó., Gómez-Pérez, A., González-Cabero, R., Suárez-Figueroa, M.C.: ODEval: a tool for evaluating RDF(S), DAML+OIL, and OWL concept taxonomies. In: Bramer, M., Devedzic, V. (eds.) AIAI 2004. IIFIP, vol. 154, pp. 369–382. Springer, Boston, MA (2004). https://doi.org/10.1007/1-4020-8151-0_32
7. Dasgupta, S., Padia, A., Maheshwari, G., Trivedi, P., Lehmann, J.: Formal ontology learning from English is-a sentences. arXiv preprint arXiv:1802.03701 (2018)
8. Dellschaft, K., Staab, S.: On how to perform a gold standard based evaluation of ontology learning. In: Cruz, I., et al. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 228–241. Springer, Heidelberg (2006). https://doi.org/10.1007/11926078_17
9. Feilmayr, C., Wöß, W.: An analysis of ontologies and their success factors for application to business. Data Knowl. Eng. **101**, 1–23 (2016)
10. Gupta, A., Piccinno, F., Kozhevnikov, M., Pasca, M., Pighin, D.: Revisiting taxonomy induction over wikipedia. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, December 11–17 2016, pp. 2300–2309. EPFL-CONF-227401 (2016)
11. Kanawati, R.: Détection de communautés dans les grands graphes d'interactions (multiplexes): état de l'art. In: HAL archives ouvertes (2013)
12. Kessler, R., Lapalme, G.: Agohra: Génération d'une ontologie dans le domaine des ressources humaines. Traitement Automatique des Langues **58**(1), 39–62 (2017)
13. Mukherjee, S., Ajmera, J., Joshi, S.: Unsupervised approach for shallow domain ontology construction from corpus. In: Proceedings of the 23rd International Conference on World Wide Web. WWW 2014 Companion, pp. 349–350. ACM, New York (2014). https://doi.org/10.1145/2567948.2577350
14. Oliveira, H., Lima, R., Gomes, J., Ferreira, R., Freitas, F., Costa, E.: A confidence–weighted metric for unsupervised ontology population from web texts. In: Liddle, S.W., Schewe, K.-D., Tjoa, A.M., Zhou, X. (eds.) DEXA 2012. LNCS, vol. 7446, pp. 176–190. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32600-4_14
15. Posse, C.: Cloud jobs API: machine learning goes to work on job search and discovery (2016). https://cloud.google.com/blog/big-data/2016/11/cloud-jobs-api-machine-learning-goes-to-work-on-job-search-and-discovery
16. Tartir, S., Arpinar, I.B., Sheth, A.P.: Ontological evaluation and validation. In: Poli, R., Healy, M., Kameas, A. (eds.) Theory and Applications of Ontology: Computer Applications, pp. 115–130. Springer, Dordrecht (2010). https://doi.org/10.1007/978-90-481-8847-5_5

17. 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)
18. Wandmacher, T., Ovchinnikova, E., Krumnack, U., Dittmann, H.: Extraction, evaluation and integration of lexical-semantic relations for the automated construction of a lexical ontology. In: Proceedings of the Third Australasian Workshop on Advances in Ontologies, vol. 85, pp. 61–69. Australian Computer Society, Inc. (2007)