OntoJob: Automated Ontology Learning from Labor Market Data

Vrolijk, J.; Mol, S.T.; Weber, C.; Tavakoli, M.; Kismihók, G.; Pelucchi, M.

DOI
10.1109/ICSC52841.2022.00040

Publication date
2022

Document Version
Final published version

Published in
16th IEEE International Conference on Semantic Computing

OntoJob: Automated Ontology Learning from Labor Market Data

Jarno Vrolijk*, Stefan T. Mol*, Christian Weber‡, Mohammadreza Tavakoli†, Gábor Kismihók†, and Mauro Pelucchi§
*Amsterdam Business School
University of Amsterdam, Amsterdam, Netherlands
Email: {j.vrolijk, s.t.mol}@uva.nl
†Leibniz Information Centre for Science and Technology (TIB), Hannover, Germany
Email: {gabor.kismihok, reza.tavakoli}@tib.eu
‡Institute of Knowledge Based Systems & Knowledge Management
University of Siegen
Siegen, Germany
Email: christian.weber@uni-siegen.de
§Emsi Burning Glass, 66 Long Wharf 2nd Floor, Boston, Massachusetts

2022 IEEE 16th International Conference on Semantic Computing (ICSC) | 978-1-6654-3418-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICSC52841.2022.00040

***Abstract*—Due to the rapidly changing labor market and the consequently widening information gap between the labor market and education, there is a need for methods that can tackle, or at least ease, the construction of labor market ontologies. The current study set out to examine the viability of Ontology Learning (OL) methods for the (semi-)automated construction of labor market ontologies and / or taxonomies. The purpose of this paper is to propose an unsupervised framework, OntoJob, that can identify and extract from raw vacancy text instances, attributes, and relations, such as job titles, worker qualities, and the non-taxonomic "is-a" relations between those concepts, and convert those to an expressive descriptive logic. Evaluation of the extracted worker qualities from OntoJob, using a small sample of 5621 job postings representing 1048 occupations, showed an overall lexical precision of 0.36 and recall of 0.22.**

***Index terms*—ontology engineering, ontology learning, labor market intelligence**

1. Introduction

The unpredictable dynamism caused by contemporary and complex phenomena, such as climate change, globalization, and technological development, means educators and job seekers alike are having to actively adapt their offerings in an effort to meet the demand of an ever changing labor market [1]. This implies a strong need on the part of these parties (and indeed others, such as governments and organizations) for valid, updated, and reliable job information.

The reputable and publicly available sources of job information (such as O*NET [2] and ESCO [3]) that these stakeholders have come to rely on, however, are ill equipped to keep up with the fast paced nature of these changes, and therewith imply a reactionary as opposed to a strategic approach to addressing labor market demand. The main problem that these knowledge-based applications have in common, is that they rely heavily on ontologies constructed by domain experts. As such, keeping their information valid and up to date remains a cumbersome and time-consuming process [4], [5].

Although developments in data science may alleviate the need for a one-by-one updating of occupations in job information sources, to the best of our knowledge, most of the research on labor market ontology engineering relies on semi-supervised or fully supervised learning techniques, meaning that, to varying degrees, they all require external data sources, even if these are only leveraged for enrichment purposes. This being the case, these approaches too ultimately force serious manual labor, for instance in e.g. annotation of sentences.

What is needed then are unsupervised (semi)automatic methods that can process vast amounts of labor market data in real time. Although humans will clearly always be needed for quality control purposes, bypassing the need for manual labor in the information processing stage opens the door to more responsive, and thus timely labor market information and with time perhaps even prediction of how the labor market will evolve. However, to the best of our knowledge, no studies to date have focused on developing a completely unsupervised approach to deriving actionable job information from data, that is not constrained by the need for manual annotation or external data sources, and that takes into consideration both taxonomic and non-taxonomic relations extracted from job vacancies.

The aim of this paper is to: 1) propose an architecture to (semi-) automatically extract and represent the knowledge from online job vacancies in OWL 2, and 2) use Smoothed Pointwise Mutual Information (SPMI) to make predictions for sparse taxonomic relations based on the similarity of the worker qualities in the online job vacancies. To guide our argumentation, this paper is structured as follows. The next section will cover the different elements of the OntoJob architecture. Adjoining (sub)sections will present the experimental results and our evaluation. Subsequently we

978-1-6654-3418-8/22/$31.00 ©2022 IEEE
DOI 10.1109/ICSC52841.2022.00040

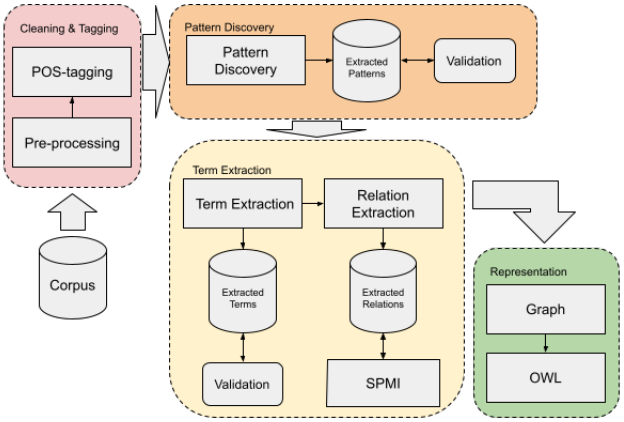

Figure 1. The OntoJob architecture.

provide an overview of the current state of research done in Ontology Learning (OL), exploring the strengths and weaknesses of the different techniques employed. Lastly, we discuss practical and theoretical implications and provide suggestions for future research.

2. OntoJob

Our approach focusses on the construction of labor market ontologies from raw vacancy texts[1]. This section provides a step-by-step overview of the different phases of OntoJob.

2.1. Preprocessing and POS-tagging

We initiated the cleaning and preprocessing of the job vacancies by first ensuring that all characters in the document were in lowercase. Next, we removed non-alphanumerics and trailing whitespaces. In contrast to the work of [6], we chose to remove the stopwords after the extraction phase, to ensure the word order in the job vacancy text remained as close to the original job postings as possible. Removal of the stopwords is important, in that it reduces the extraction of strings that are unlikely to be terms for our domain [6]. Finally, we labeled all the words in the corpus with their corresponding Part-of-Speech-tag (POS-tag).

2.2. Pattern Discovery

The initial discovery of patterns starts with the introduction of seed terms. OntoJob then scans the corpus to discover contextual patterns in which the given seed instances[2] are commonly found. Next, we used regular expressions (Regex) to generate a "general pattern" to identify other worker qualities that exhibit the given patterns. All in all, we are only interested in those patterns that are able to efficiently identify two or more worker qualities. Since we had no prior knowledge of the exact worker qualities present in the vacancies, we chose to approximate the generalizability of the pattern by checking how capable the pattern was in identifying different seed terms [7]. Formally, this approximation, the so-called estimated recall, can be described as $\frac{c(p)}{S}$, where S is the total number of seeds used and $c(p)$ is the number of distinct seeds found with the pattern p. If we keep those patterns that have an estimated recall of $1/S$, then 96% of the potential rules are eliminated, and average efficiency increases fivefold [7].

2.3. Term Extraction

After discovering and validating the patterns, we can start extracting the worker qualities from the job vacancies. However, the discovered patterns focus solely on the extraction of single-word worker qualities, whereas most of the worker qualities are multi-word. To extract multi-word worker qualities and evaluate their appropriateness for the domain we wish to extract, we chose the application of the C-value method as proposed by [6].

The extraction phase thus consists of a linguistic part a statistical part. The preparation for the linguistic part is largely completed during the preprocessing and POS-tagging phases. Specifically, the exclusion of words occurring in the stop word list and the POS-tagging of the text are necessary steps before applying the linguistic filter to search for the multi-words. Similar to the work by [6], we chose to add three generic patterns[3] to our set of validated patterns.

The first evaluation was done using the C-value method. This method considers the termhood of a worker quality string, ranking it in the output list of worker quality terms. The measure uses statistical characteristics of the worker quality string, namely: i) the total frequency of occurrence of the worker quality string in the corpus; ii) the frequency of the worker quality string as part of other longer worker quality strings; iii) the number of distinct longer worker quality strings; and iv) the length of the worker quality string [6]. The measure of termhood, called C-value is then given as

$$C(a) = \begin{cases} log_2|a| \cdot f(a) \\ log_2|a|(f(a) - \frac{1}{P(T_a)}) \sum_{b \in T_a} f(b)) \end{cases} \quad (1)$$

where a is the worker quality string, $f(.)$ is the frequency of occurrence in the corpus, T_a is the set of extracted terms that contain a and $P(T_a)$ is the number of these worker quality terms [6].

2.4. Relation extraction

Apart from the non-taxonomic relations found during the previous discovery and extraction phases, we are also

1. Our code is publicly available at https://github.com/JarnoV2/OntoJob
2. Note that all the seed terms are single-word terms.

3. $NOUN^{+}NOUN$,
$(ADJ|NOUN)^{+}NOUN$,
$((ADJ|NOUN)^{+}|((ADJ|NOUN)^{*}(NOUNPREP)^{?})(ADJ|NOUN)^{*})NOUN$

interested in the hierarchical relations between worker qualities found in the job vacancies. To find these taxonomic relations in the text, we will perform low-rank embedding using singular value decomposition (SVD) to combat the sparsity constraints in the extracted relations [8]. In short, the hypernym extraction phase consists of three parts: i) identifying and extracting the taxonomic relations from text using the Hearst Patterns; ii) predicting the hypernymy relations based on the Positive Pointwise Mutual Information (PPMI) of the extracted Hearst Patterns; and iii) low-rank embedding the PPMI matrix using SVD to combat the sparsity constraints caused by the pattern-based model and improve the precision and recall for detection of Hearst patterns [8].

Furthermore, we will use hierarchical relations found to include and/or exclude certain extracted terms from the validated worker quality terms. For example, we can look at the hyponyms of "requirements" to include additional import requirements for certain jobs. In contrast, hyponyms of "benefits" should be excluded as a worker quality.

Apart from the extraction of the hierarchical relations, the identification also introduces additional worker qualities not discovered by the earlier discovered and extracted patterns. Furthermore, it gives us additional information about the relations between different worker qualities (e.g. Java, Python, and C++ are all Object-oriented programming languages). In other words, each time we find the knowledge "Java" to be significant, we can infer that "Object-oriented programming", as a "requirement" is also important. Please note that the converse is not necessarily true e.g. if "Object-oriented programming" is important to a certain occupation, then it is not necessarily true that knowledge of "Python" is also important (put differently, the relation is anti-symmetric).

To ensure the significance of the extracted terms, we chose to only keep extracted relations if there were at least two different patterns to discover the relation. To facilitate consistency of our extracted relations, we removed all relations $p(y,x) < p(x,y)$, to account for the anti-symmetric nature of the taxonomic relations.

2.4.1. PPMI. An issue with the simple extraction probabilities used is that they tend to be skewed by the occurrence probabilities of their surrounding words. This is very common in natural languages, yet undesirable for our purposes. We therefore corrected for different word occurrence probabilities by means of PPMI, which is defined as:

$$ppmi(x,y) = max(0, \log \frac{p(x,y)}{p^-(x)p^+(y)}) \tag{2}$$

2.4.2. SPMI. While PPMI does correct for different word occurrence probabilities, it cannot handle sparsity [8]. Since the extraction rules suffer from sparsity, we chose to follow the recommendations from [8], and use low-rank embedding of the PPMI matrix to make predictions for unseen pairs. In particular, let $m = |\{x : (x,y) \in \mathcal{P} \vee (y,x) \in \mathcal{P}\}|$ denote the number of unique terms in $\mathcal{P}$. Furthermore, let $X \in \mathbf{R}^{mxm}$ be the PPMI matrix with entries $M_{xy} = ppmi(x,y)$ and let $M = U\Sigma V^\top$ be its SVD. We can then predict hypernymy relations based on the truncated SVD of M via

$$spmi(x,y) = \mathbf{u}_x^\top \Sigma_r \mathbf{v}_y \tag{3}$$

where $\mathbf{u}_x, \mathbf{v}_y$ denote the x-th and y-th row of U and V, respectively, and where Σ_r is the diagonal matrix of truncated singular values.

The SPMI equation can be interpreted as a smoothed version of the observed PPMI matrix [8]. Where, the truncation of the singular values, it computes a low-rank embeddings such that similar words have similar representations [8]. Since SPMI is calculated for all pairs x, y, it allows us to make predictions based on the similarity of words [8].

2.5. Graph Representation

We then use the validated terms and relations to build a knowledge graph. Here, the terms extracted from the corpus will be used as the nodes in the graph. Whereas, worker qualities are connected if, and only if, they co-occur in certain job postings.

2.5.1. Transversality score extraction. We then use the adjacency matrix and the power iteration algorithm to derive the relativity score for each vertex v in the worker quality network. Note that, although the graph with the worker qualities is undirected, the graph with the taxonomic relations is directed in that the pair (x,y) is different from the pair (y,x). Given a graph G, and adjacency matrix $A = (a_{v,t})$ the relative centrality score of v can be defined as:

$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{v,t} x_t \lambda \tag{4}$$

However, since we are dealing with an undirected graph (we only calculate the transversality scores on the adjacency matrix of the worker qualities), the edge e_{ij} is identical to e_{ji}. Therefore, if a quality v_i has k_i neighbours, then the local clustering coefficient can be defined as:

$$C_i = \frac{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E}{k_i(k_i - 1)} \tag{5}$$

To increase the interpretability of our transversality measure, we chose to feature scale both the eigenvector centrality and the local clustering coefficient.

2.6. Ontology Representation with OWL

Now that all the worker quality terms are extracted and validated, it is time to construct the ontology. As mentioned earlier, our ontology consists of two different relation types; i) the taxonomic "is-a" relations, and ii) the non-taxonomic relations. We will extend the "JobPosting" concept as defined by schema.org[4]. We will convert our ontology representation to OWL 2. OWL documents, known

4. https://schema.org/JobPosting

as ontologies, can be published on the World Wide Web and may refer to, or be referred to by, other OWL ontologies [9].

The upper level of our ontology consists of the two main concepts, namely: *JobPosting* and *WorkerQuality*. Following the schema.org definitions for both the *JobPosting* and the *WorkerQuality*, we define JobPosting as "a listing that describes a job opening in a certain organization" and WorkerQuality as "a statement of knowledge, skill, ability, task or any other assertion expressing a competency that is desired or required to fulfill this role or to work in this occupation." Furthermore, for each worker quality we measure the degree to which the quality is transversal. We define a worker quality to be transversal when that worker quality is reflected and used/required in a variety of roles or occupations. More specifically, transversal worker qualities are employee characteristics that are not limited to a particular job, task, academic discipline, or area of knowledge, and that are called for in a wide variety of situations and work settings. In contrast, a specific worker quality is unique to, or highly prototypical for, a particular profession.

2.7. Evaluation of OntoJob

Since we are primarily interested in the evaluation of the OL algorithm itself, requirements regarding the evaluation tend to be different from traditional ontology evaluation [10]. The reason for this is the different aspects of ontology learning and their impact on the resulting ontology. Therefore, we would like to clarify that our main interest is in the quality of the learning algorithm itself.

Following the practical recommendations of [10], we chose to evaluate the ontology learning algorithms by gold standard based evaluation. Since, to the best of our knowledge, there does not yet exist a labor market ontology learned from our corpus, we chose to use the data set provided by Emsi Burning Glass (EBG) to guide our evaluation [11]. In short, we selected all the EBG extracted job postings posted in the US in January 2019, and used these selected job postings as the gold standard on which to evaluate our performance. Evaluation of the concepts, the so-called lexical layer, is accomplished via the LP and LR measures. Whereas, for the taxonomic layer, we chose to evaluate by enriching the EBG identified concepts with their semantic cotopy from WordNet and use TP_{csc} and TR_{csc} to evaluate the performance of OntoJob.

The annotation will focus on three different elements of ontology. First, the annotators will identify and tag the worker qualities present in each of the randomly selected vacancies. Next, the overall set of worker qualities found in the randomly sampled vacancies will be analyzed for taxonomic relations. Note that there is a discrepancy between whether the relations are present in the text and if the annotators deem the relations to exists. Lastly, the annotators will evaluate the non-taxonomic "requiredFor" relation between the worker qualities and the job titles accompanying each vacancy.

We use the lexical precision and recall to reflect how good the learned terms cover the target domain [10]. Given that there exist various evaluation measures for doing a gold standard-based evaluation of concept hierarchies, we follow the advice of [10] and use the Taxonomic Precision and Recall measures. To overcome the disadvantages of evaluating both the lexical and taxonomic layers of the ontology, we chose to minimize the effort of the lexical layers by using the common semantic cotopy and all the super- and subconcepts of the term, in the calculation of the taxonomic precision and recall [10]. The described measures for evaluation of the taxonomic relations will be calculated with and without generalization by SPMI. Similarly, both the extracted concepts and the extracted relations (taxonomic and non-taxonomic) will be evaluated with differing seed term initializations of 10, 50, or 100 terms.

2.8. Setup

2.8.1. Data. For our analysis, we used 13006 online job vacancies supplied to us by EBG [11]. The online job vacancies are extracted from US job boards in the year 2019. To focus our analysis and better check the outcomes we chose to analyze only a small subset[5] of the entire EBG data set, hich has more than 200 million US vacancies. For each of the 13006 job postings, we thus have i) a unique job posting identifier, ii) the SOC-code[6] of the occupation for which the job posting is posted, iii) the job title, iv) the job description, and lastly v) the worker qualities extracted by EBG for each job posting.

In total, the EBG data set has 5860 worker qualities extracted from the 13006 job postings. However, to increase the robustness of the results, we chose to remove those job titles that have less than 30 job postings. Furthermore, we also excluded worker qualities with an occurrence less than 5. The resulting data set leaves us with 5621 job postings and 1.048 worker qualities. The analyzed job postings[7] on average contained 3137 characters ($M = 3136.56, SD = 2189.54$) and on average 443 words ($M = 443.21, SD = 314.65$).

A closer look into the distribution of worker qualities over the occupational SOC-codes, in the sample, shows us that on average there are 134 worker qualities ($M = 134.27, SD = 70.67$). All in all, the occupational SOC-codes, on average, represent 1 job title (these are the job titles that were posted with the job posting) ($M = 1.37, SD = 0.91$). If we look more granular than the occupational SOC-codes, in this case on the job title level, we see that on average there are 141 ($M = 141.30, SD = 70.67$) worker qualities.

On average there are approximately 61 job postings for each job title ($M = 61.79, SD = 50.38$). In a simi-

5. We selected the first 13006 rows of the dataset.

6. Standard Occupational Classification (SOC), which is a federal statistical standard way used by federal agencies in the U.S. to classify workers into occupational categories.

7. One can find examples of the analyzed job postings on the earlier provided GitHub page.

lar fashion, there are 66 job postings for each SOC-code ($M = 66.15, SD = 67.63$). Whereas each job posting has 8 worker qualities on average ($M = 7.89, SD = 5.91$).

3. Results

In this section we will first discuss our findings and the impact of the differing seed term initializations, then we discuss the quality of the extracted worker qualities resulting from the application of OntoJob. Next, we summarize the overall statistics for the extracted "requiredFor" relation. Lastly, we will check the impact of the generalizing of the extracted taxonomic relations through SPMI.

Surprisingly enough, the three different setups, each with a different collection of seed terms, respectively; 5, 10, and 20 terms, extracted almost an identical number of worker qualities, namely; 2194, 2196, and 2198. Upon further examination, apart from the minor differences, each outcome found the same 2194 worker qualities. As, such we can conclude that the most constrained version, with only one pattern, reached the same results as the setup with larger seed term collections.

Please note that the influence of the number of seed terms is only relevant in the extraction of the single-word worker qualities. As such, the multi-word worker qualities largely remain the same. Given that the impact of the three different seed term collections leads to negligible differences in the extraction of worker qualities, we propose choosing highly transversal worker qualities, since this will likely lead to the most discovered patterns across different job postings. For example, communication skills alone occurred 1717 times.

3.1. Extracted Worker Qualities

For the lexical layer, our initial calculations of the intersection between the collection of concepts from the EBG dataset and the collection of our extracted worker qualities showed only 61 exact matches (for all three different setups of the seed terms). However, we suspect that the EBG dataset was considerably altered (logically so) to harmonize the worker qualities that are essentially covering the same semantics. For example "analytical ability math skills" and "employees math ability" are both referring to a worker quality named "math ability" or to be more precise the EBG equivalent "Basic Mathematics".

To account for this difference in the exact terms, we decided to randomly select 3 occupations by their job title and calculate the lexical precision and recall over their sets of worker qualities. While not as robust and thorough as the analysis of the lexical precision and recall on the entire collection, this should give a good indication of how OntoJob compares with the EBG data, while also remaining feasible for manual mapping.

Overall the three occupations, respectively; "business development specialist", "sales manager" and "caregiver care aid", have 412 unique worker qualities. After the mapping 341 unique worker qualities remain. On the other hand, OntoJob extracted 211 unique worker qualities for the three occupations. Overall, results show $LP = 0.359$ and $LR = 0.223$. Results at the occupation level show that the occupation "sales manager" scores best on lexical precision $LP = 0.553, LR = 0.161$. However, when looking at the lexical recall, the "caregiver care aide" occupation outperforms $LP = 0.226, LR = 0.368$. The results from the "business development specialist" occupation $LP = 0.403, LR = 0.238$ seem to be most aligned with the overall scores on the lexical measures.

3.1.1. Impact of *n* Seed terms. Selection of the seed terms was done with the rationale that transversal worker qualities tend to occur more broadly among different occupations. As such, we used "communication", "sales", "teamwork", "planning" and "writing". However, the approximated precision that is used for validation also becomes more strict when there are fewer seed terms since $1/S = 1/5$ requires the approximation of the precision to be above 0.2. Subsequently, the constraint becomes more flexible as the number of seed terms increases to $1/10$ and $1/50$. Five seed terms resulted in the acceptance of just one discovered pattern; "NOUN? CCONJand NOUNexecution", where "?" serves as a placeholder for the identified worker quality. Ten seed terms result in four discovered patterns namely; 1) "ADPof DETthe NOUN?"; 2) "NOUN? CCONJand NOUNexecution"; 3) "NOUNyears ADPof NOUN?" and 4) "NOUN? NOUNexperience ADPwith". Lastly, twenty seed terms leave us with 56 discovered patterns after validation.

3.2. Non-Taxonomic relations

On average OntoJob extracts 36 worker qualities per occupation ($M = 35.86, SD = 22.82$). The occupations "caregiver care aide" and "operations manager" both had 93 worker qualities extracted from their job postings. The EBG data set on average has 109 additional worker qualities for the occupations used in this analysis ($M = 109.70, SD = 77.02$). However, as mentioned earlier, the EBG data has only 1.048 total worker qualities, which is more than a thousand less than the OntoJob extracted worker qualities.

3.3. SPMI

All in all, 40 taxonomic relations were found among the job description texts. The hypernyms are generally not targeted specifically on worker qualities. However, they do seem to capture relevant information about the job posting. There are 7 hypernyms namely; "benefits", "duties", "influencers", "job", "motion", "position" and "responsibilities".

Unfortunately, it was not possible to evaluate the taxonomic layer via the earlier defined enrichment of the EBG data through WordNet. Furthermore, we suspect that the generalization was incapable of making sound predictions due to the lack of data.

4. Related work

There have been different initiatives on closing the information gap (supra) between worker quality demand and supply. Most effort focuses on providing online tools for job seekers, educators, and HR professionals. For example, tools provided by [2] rely on databases containing hundreds of standardized occupation-specific predictors on close to 1000 occupations for the U.S. labor market. Another notable example, the European Skills, Competences, Qualifications, and Occupations (ESCO), which is a dictionary that describes, identifies and classifies the occupations and skills relevant for the European labor market and education and training [3]. These knowledge-based applications, while useful, draw heavily upon ontologies constructed by domain experts and as such suffer from impaired practical scalability issues and are error-prone due to their high demand on time and resources [4], [5].

Research towards methods for the (semi-)automated construction of ontologies constitutes the field of OL, which is a subfield of the Ontology Engineering domain that works on the integration of numerous disciplines to construct ontologies [5]. While OL is actively researched, to the best of our knowledge, there have only been several attempts towards (semi-)automated ontology learning of the labor market. Notable research in the extraction of labor market ontologies is the work by [1], which presented an ontology-based information extraction method that identified data science skills from job vacancies. Results proved the feasibility of their automated extraction with an F-measure of 79%-81%. Furthermore, [12] designed a system (SKILL) to meet the increasing business need of workforce analytics achieving 91% accuracy and 76% recall on their taxonomy building and 82% accuracy on actual skill tagging with 70% recall. As of yet, there has been no unsupervised ontology learning method capable of constructing a labor market ontology without the usage of manually annotated labor market resources.

5. Conclusion and Future Work

In this paper, we presented a low-cost approach capable of easing the construction of labor market ontologies from online job vacancies. Furthermore, instead of relying solely on the Hearst patterns for the extraction of the taxonomic relations, we chose to increase coverage using SPMI to generalize hypernymy relations that do not occur in the text. Based on the taxonomic relationships found, we deem our method for extraction to be viable. However, the usefulness of the extracted relations does not exceed serving as an exclusion criterion for noise in the identification of worker qualities. We suspect that the added benefit of generalization through SPMI requires much more job postings to fully demonstrate its potential.

In contrast to other works using similar architectures, we proposed to evaluate the performance using data provided to us by EBG to generate a ”golden truth” data set to validate the capabilities of OntoJob’s learning algorithm for the lexical and taxonomic layers. Unfortunately, our findings demonstrate that it is not easy to validate OL algorithms, because data, in our case the EBG data set, contained additional post-processing in the form of harmonization and / or categorization. Furthermore, future work should focus on benchmarking the performance of OntoJob to other, non labor market specific, OL algorithms.

For example, the EBG worker quality ”patient care” was much broader than the OntoJob extracted equivalents ”individual support”, ”aide” and ”health support”. As such, the lexical precision and recall measures do not fully do justice to the extracted worker qualities from OntoJob. Also, there were situations in which it was the other way around, meaning that OntoJob presented a broader term for multiple EBG extracted terms (e.g. ”sales” from OntoJob corresponded with ”complex sales”, ”sales presentation”, ”sales”, ”direct sales”, etc.). A natural progression of the current work would be a follow-up study in which we provide better harmonization of the worker qualities to enable better validation through ”golden truth” validation. We also believe that the harmonization of the worker qualities will further increase the usability of the extracted knowledge.

References

[1] E. M. Sibarani, S. Scerri, C. Morales, S. Auer, and D. Collarana, “Ontology-guided job market demand analysis: A cross-sectional study for the data science field,” in *Proceedings of the 13th International Conference on Semantic Systems*, ser. Semantics2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 25–32. [Online]. Available: https://doi.org/10.1145/3132218.3132228

[2] National Center for O*NET Development., “O*net online.” 2021, online; accessed 11-February-2021. [Online]. Available: https://www.onetonline.org/

[3] European Commission and European Centre for the Development of Vocational Training, “European skills/competences, qualifications and occupations.” 2021, online; accessed 11-February-2021. [Online]. Available: https://ec.europa.eu/esco/portal/home

[4] H. Mousavi, D. Kerr, M. Iseli, and C. Zaniolo, “Harvesting domain specific ontologies from text,” in *2014 IEEE International Conference on Semantic Computing*. IEEE, 2014, pp. 211–218.

[5] A. Maedche and S. Staab, “Ontology learning for the semantic web,” *IEEE Intelligent systems*, vol. 16, no. 2, pp. 72–79, 2001.

[6] K. Frantzi, S. Ananiadou, and H. Mima, “Automatic recognition of multi-word terms:. the c-value/nc-value method,” *International journal on digital libraries*, vol. 3, no. 2, pp. 115–130, 2000.

[7] D. Downey, O. Etzioni, S. Soderland, and D. S. Weld, “Learning text patterns for web information extraction and assessment,” in *AAAI-04 workshop on adaptive text extraction and mining*, 2004, pp. 50–55.

[8] S. Roller, D. Kiela, and M. Nickel, “Hearst patterns revisited: Automatic hypernym detection from large text corpora,” *arXiv preprint arXiv:1806.03191*, 2018.

[9] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, S. Rudolph *et al.*, “Owl 2 web ontology language primer,” *W3C recommendation*, vol. 27, no. 1, p. 123, 2009.

[10] K. Dellschaft and S. Staab, “Strategies for the evaluation of ontology learning.” *Ontology Learning and Population*, vol. 167, pp. 253–272, 2008.

[11] Emsi Burning Glass, “Online job vacancies,” 2021. [Online]. Available: https://www.burning-glass.com

[12] M. Zhao, F. Javed, F. Jacob, and M. McNair, “Skill: A system for skill identification and normalization,” in *Twenty-Seventh IAAI Conference*, 2015.