CEUR-WS.org/Vol-3433/short2.pdf

Creating Dynamically Evolving Ontologies: A Use Case from the Labour Market Domain

Maaike H.T. de Boer[1], Roos M. Bakker[1,2] and Maaike Burghoorn[1]

[1]*TNO, Anna van Buerenplein 1, 2595DA, The Hague, The Netherlands*
[2]*Universiteit Leiden, Reuvensplaats 3, Leiden, 2311BE, The Netherlands*

Abstract
The world is changing, which means that formal representations of (part of) the world should change with it. In this paper, we explore to what extent automation of updating ontologies or taxonomies could be possible using Hybrid AI. We use Natural Language Processing (NLP) methods to automatically recognize and integrate new concepts and alternative labels in an ontology. The labour market domain is used as a use case, as new jobs and skills should be added on a regular basis. In our experiments we show that with our dataset 1) language-based methods seem to outperform a string-based method, but no clear difference between language-based methods can be observed; 2) it is easier to map skills within one ontology (to alternative labels / synonyms) compared to between different sources; 3) no clear difference in performance between mapping with synonyms / more relevant text compared to without is visible yet. This means that we can certainly take steps towards automation in the field of ontology evolution, but we are not there yet. In the future we plan to further experiment with at least the integration of skills (3), as well as the creation of a human-in-the-loop system to validate our work and to combine the strengths of humans and machines.

Keywords
Ontology Mapping, Natural Language Processing, Ontology Evolution, Transformer

1. Introduction

Ontologies are ‘a set of concepts and categories in a subject area or domain that shows their properties and the relations between them.’[1] They can be seen as a representation of (part of) the world. The world is, however, ever changing; new concepts are formulated, older concepts get forgotten, new relations and properties are created. For example in the fall of 2022 words as ‘shrinkflation’, ‘bachelorx party’ and ‘pawternaty leave’ are added to the dictionary.[2] One of the domains in which ontologies or taxonomies often have to handle new concepts and relations is the labour market, as new jobs and skills are often created.

These new versions of ontologies or taxonomies are currently often manually created, which is

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2023 Spring Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering (AAAI-MAKE 2023), Hyatt Regency, San Francisco Airport, California, USA, March 27-29, 2023.

maaike.deboer@tno.nl (M. H.T. d. Boer); roos.bakker@tno.nl (R. M. Bakker); maaike.burghoorn@tno.nl (M. Burghoorn)
0000-0002-2775-8351 (M. H.T. d. Boer); 0000-0002-1760-2740 (R. M. Bakker); ? (M. Burghoorn)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

[1] https://www.oxfordlearnersdictionaries.com/
[2] https://www.dictionary.com/e/new-dictionary-words-fall-2022/

very labour intensive. In this paper we explore to what extent automation of updating ontologies or taxonomies could be possible using Hybrid AI - a combination of learned knowledge and engineered knowledge. We specifically do not aim for full automation, but we foresee that a part can be automated using ontology mapping techniques and a human-in-the-loop for verification will still be necessary.

In the next section, we discuss related work on ontology mapping and ontology evolution. Section 3 discusses our approach and outlines our experimental setup and the results. The last section concludes this paper and provides an outlook to future work.

2. Related Work

2.1. Ontology Mapping

The comparison between ontologies is often called ontology mapping or ontology matching. Related work in this field can be divided into element-level and structure-level mapping [1, 2]. Element-level mapping is a mapping in which each element in an ontology is considered independently from the other elements in the ontology. Structure-level mapping is a mapping that includes the whole ontologies or groups of concepts with groups of concepts. As our work will not include matching whole ontologies or groups of concepts with groups of concepts, we will focus on element-based mapping. Within the element-based mapping, we focus on two approaches as they use Natural Language Processing: string-based and language-based approaches [2]. The string-based approaches only use the letters of the words to create a mapping, whereas language-based approaches also use other information of the words, such as the lemmas or the morphology.

In recent years, there has been a development within the language-based approaches to use more semantic approaches [3]. One of these approaches is using word embeddings to compute similarity between concepts [4, 5]. These word embeddings have also been combined in a hybrid way with human knowledge (hand-crafted features), such as in OntoEmma [6]. Other approaches include deep learning approaches such as DeepAlignment [7] and Transformer models [8, 9].

2.2. Ontology Evolution

Most research presented in recent papers is focused on the creation of ontologies. However, when ontologies are used in real world applications or adopted in companies, it is important to keep the ontology up to date. This field is called ontology evolution [10, 11, 12]. According to Zablith et al. [11] ontology evolution can be split into five phases: a detection phase, a change suggestion phase, a change validation phase, an evolution impact phase, and a change management phase. The detection phase is focused on the detection of a need for change, also named change capturing [13] or information discovery [14]. The change suggestion phase suggests possible changes to the ontology. The change validation phase validates the changes proposed in the suggestion phase. The evolution impact phase assesses the impact of the changes, often on an application level. The change management phase is a continuous task that records and versions the changes in the ontology.

3. Experiments

In our experiments we explore to what extent automation towards a new version of an ontology could be possible. We conduct three experiments of which experiment 1 and 2 are focused on change suggestion - given new information, does the ontology need adaption? - and experiment 3 combines change suggestion and validation - where should the new information go and what does this mean in terms of performance? - [11]. We explain the data, methods, evaluation and results per experiment. All experiments are focused on the domain of the labour market, in which we expect that skills are added or adjusted in an ontology most often.

3.1. Experiment 1: Recognize similar skills (Dutch)

The goal of this experiment is to recognize similar skills. If we are able to recognize similar skills from source documents (such as vacancies or other skill ontologies) to our target ontology or taxonomy, these could automatically be suggested as a synonym or other relevant link to an existing skill.

Dataset The Dutch CompetentNL[3] is used as a source ontology and the translated version of the skills in the ESCO ontology[4] is used as a target. Both have 20,000+ skills (such as *use foreign language*), and in the experiments we compare a random sample of 10% of the data as a 20,000 times 20,000 skills match would take too much time. Our ground truth is an externally created mapping between CompetentNL and ESCO, which is created using existing crosswalks as well as manual validation.

Method We use the following state-of-the-art methods mentioned in section 2.1:

- LVS [15]: Levenshtein distance, this is the baseline (string-based);
- SpaCy (NL) [16]: the Dutch word vector model trained on a large news corpus (nl_core_news_lg) (language-based);
- BERTje (NL) [17]: an Dutch Transformer model trained on several large text corpora, including books and wikipedia (language-based);
- XLnet (EN) [18]: an English transformer model that is claimed to outperform BERT on 20 tasks (language-based).

In each method, we calculate for each skill the similarity to each other skill. The skills are compared and ranked according to the (cosine) similarity score.
The hypothesis is that the language-based methods ourperform the string-based method.

Evaluation Four different metrics are used to evaluate the performance of the various methods:

- Accuracy: if the top (all) skill(s) is (are) the same as the ground truth skill(s), assign value 1.0; otherwise 0.0;

[3] https://www.werk.nl/arbeidsmarktinformatie/skills/competentnl-standaard-voor-skills-in-nederland
[4] https://ec.europa.eu/esco

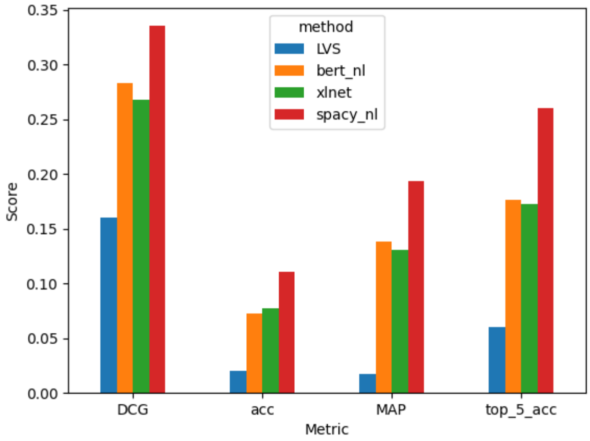


Figure 1: Performance Results Experiment 1: Recognize similar skills in Dutch using different methods

- Top 5 accuracy: the number of ground truth skills that appear in the top 5 divided by the number of ground truth skills;
- MAP: Mean Average Precision; the mean area under the precision-recall curve. This takes into account the place of the ground truth skills in the ranking;
- DCG: Discounted Cumulative Gain; the graded relevance scale of skills that evaluates the gain. Similar to MAP the place of the ground truth skills in the ranking is used.

Results Figure 1 shows the results of recognizing similar skills in Dutch. Overall, the result is not as good as expected. On all metrics, Spacy is the top performer and LVS is the worst. This confirms our hypothesis, as LVS only uses the information of the string, whereas Spacy and the other models also use linguistic information. We chose one examples to show the top 1 result of the various methods (translated to English):
Skill: *check availability of military resources*;
Ground Truth Skill: *identify and monitor physical assets*
LVS: *estimate resources needed*; BERT_nl: *control financial and economic resources and activities*; XLnet: *use computer aided design and drawing tools*; SpaCy_nl: *control operational activities.*

These results show that the ground truth skill is often not very close to the original skill. The reason for this is that the Dutch CompetentNL is often more specific compared to the (translated to Dutch) ESCO skill set, and different choices in skills are made. This makes it very hard to create an automatic mapping. Also, the current metrics only look at one ground truth skill. This motivated us to create experiment 2, within one ontology and multiple ground truth skills.

3.2. Experiment 2: Recognize similar skills (English)

Experiment 2 has the same goal as experiment 1, but is conducted on English data and considers data from only one source (the taxonomy / ontology). The mapping is created from an original skill to an alternative label.

Dataset We zoom in on a part of the English version of the ESCO ontology. Based on the existing occupancies and their skills, two datasets are created:

- Data Scientist (DS): total of 47 skills, with each zero or more alternative labels;
- Systems Analysts (SA): total of 19 occupancies (incl. DS) and 231 skills, with each zero or more alternative labels.

The alternative labels are used as the ground truth for the skills.
The hypothesis is that performance in experiment 2 is higher compared to experiment 1, as it might be easier to map within one source of information (and English models are potentially also better than Dutch models as English is a bigger language).

Method The English version of the language based methods of experiment 1 are used, as well as a specific Transformer model for this domain:

- SpaCy (EN) [16]: the English version of Spacy (en_core_web_lg);
- BERT (EN) [19]: the English version of BERTje;
- JobBERT (EN) [20]: English Transformer model trained on vacancy text;
- XLnet (EN) [18]: same as experiment 1.

Similar to experiment 1, all (alternative) labels for all skills are compared and ranked according to the (cosine) similarity score.

Evaluation One evaluation metric is added to the four of experiment 1:

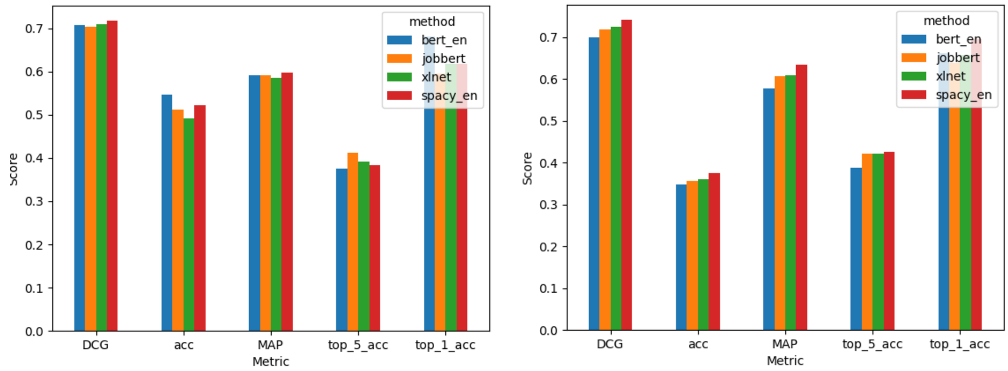
- Top 1 accuracy: if (one of) the ground truth skill(s) is the top skill, assign value 1.0; otherwise 0.0;

Results Figure 2 shows the results for the experiment on the DS dataset and the SA dataset, respectively. For each dataset, we include 1 example again, this time with the top 3 results.
(DS) Skill: *manage data*;
Ground Truth Skills: *data resource management, operate data quality tools, manage data lifecycle, data administration, administer data*

- BERT_en: 1. *administer data*, 2. *use model data*, 3. *use data bases.*
- JobBERT: 1. *administer data*, 2. *prepare data*, 3. *verify data.*
- XLnet: 1. *manage data models*, 2. *data*, 3. *administer data.*
- SpaCy_en: 1. *manage data lifecycle*, 2. *manage data models*, 3. *data.*

(SA) Skill: *signal processing*;
Ground Truth Skill : *digital signal processing, analogic transmission digital transmission, DSP*

- BERT_en: 1. *digital signal processing*, 2. *data processing*, 3. *image acquisition.*
- JobBERT: 1. *digital signal processing*, 2. *data processing*, 3. *analogic transmission digital transmission.*



(a) Data Scientist

(b) Systems Analyst

Figure 2: Performance Results Experiment 2: Recognize similar skills in English using different methods

- XLnet and SpaCy_en: 1. *digital signal processing*, 2. *data processing*, 3. *processing of data*.

The first observation is that the performance is much higher compared to experiment 1. In the examples we can see that the alternative labels, which are the ground truth skills, are much closer in meaning compared to the ground truth skills in experiment 1. A second observation is that the performance of the different methods is quite close to each other (the baseline LVS is not included). This is also visible in the examples. A third observation is that performance - especially accuracy - for the Systems Analyst is slightly lower compared to Data Scientist. This could be explained by the larger number of skills.

3.3. Experiment 3: Integrate / Map skills (English)

The goal of this experiment is the integration and validation of the new synonyms in the ontology. The difference in performance is calculated when new synonyms are added to the comparison. We compare four settings which could be used to compare a skill or skill set (incl. alternative labels) to other skills in the same occupancy group. Four settings are used: 1) nosyn: just the new skill used; 2) syn: skill + all alternative labels; 3) random_alt: skills + random 1 alternative label from that skill; 4) wrong_alt: skills + random 2 alternative labels from another skill.

The hypothesis is that the wrong_alt has the worst performance as wrong information is added and this should thus be least close to other skills from the same group, followed by random, nosyn and syn. More and complete information will probably improve mapping performance.

Dataset The same dataset as experiment 2 is used, but the skill mappings are manually created.

Method The same 4 methods as experiment 2 are used.

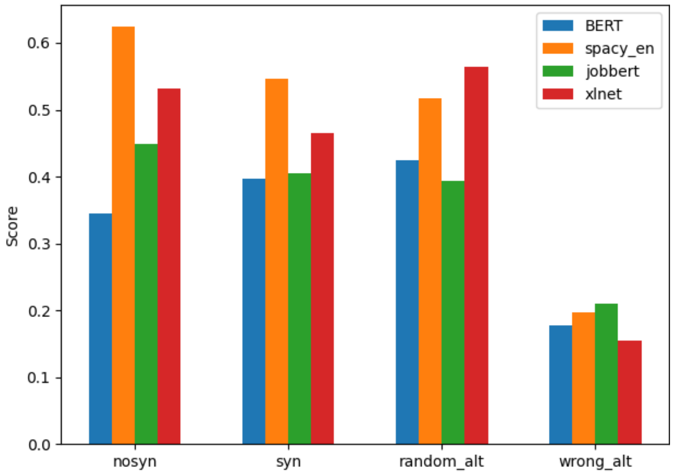


Figure 3: Performance Results Experiment 3: Integrate / Map skills (English) in different settings

Evaluation The same evaluation as experiment 2 is used, but only results for MAP are shown in the result.

Results Figure 3 shows the results of the integration of new skills, on the metric MAP and the DS dataset only due to space limitations. Other metrics show a similar trend.

We will use the example of DS from the previous section, where the skill for nosyn is the same as mentioned before (*manage data*), syn is the skill and all ground truth skills mentioned. Random_alt is *manage data manage data lifecycle* and wrong_alt is *manage data develop own practices continuously Language Integrated Query*. The mappings are, however, not on the synonyms any more, as those are directly used in the experiment, but on other skills in the same dataset. We show the result of the mapping of the random_alt below:
Random_alt Skill: *manage data manage data lifecycle*;
Ground Truth Skills: *manage findable accessible interoperable and reusable data ensure corrected data storing*

- BERT_en: 1. *create data models manage data models*, 2. *manage ICT data architecture define enterprise data architecture*, 3. *design database in the cloud design cloud data architecture.*
- JobBERT: 1. *manage ICT data architecture define enterprise data architecture*, 2. *design database in the cloud design cloud data architecture*, 3. *implement data quality processes verify data.*
- XLnet: 1. *create data models manage data models*, 2. *unstructured data data analytics*, 3. *implement data quality processes verify data*
- SpaCy_en: 1. *implement data quality processes verify data*, 2. *establish data processes develop data processes*, 3. *create data models manage data models*

The results show that if no synonyms (nosyn) or correct alternative labels (syn) are used the SpaCy model performs best. In case of one wrong alternative label, XLnet outperforms SpaCy. This means that XLnet is less affected by this wrong alternative label. If more than one wrong alternative labels are added performance drops with all methods, as would be expected. JobBERT - trained on relevant data - does outperform the general BERT model, but not XLnet and Spacy in this experiment.

4. Conclusion & Future Work

In this paper we perform experiments to find out to what extent automation towards a new version of an ontology could be possible. We focus ourselves on one Dutch and two English datasets within the labour market domain. Our experiments show that exp 1) language-based methods outperform a string-based method, but no clear difference between language-based methods can be observed; exp 2) it is easier to map skills within one ontology (to alternative labels / synonyms) compared to between different sources; exp 3) no clear difference in performance between mapping with synonyms / more relevant text compared to without is visible yet.

Our results motivate the deployment of automated concept mapping to support the evolution of ontologies. We now used Hybrid AI - the combination of learned knowledge and human knowledge - and we foresee that hybrid intelligence - the combination of human interaction with a machine - is necessary in the first step towards automation. In future work, we want to verify the strengths and weaknesses of both, and create an interface to pose suggestions for new skills and to work on a new version of an ontology in an user-friendly way.

Acknowledgments

We would like to thank the internal TNO program on AI (APPL.AI) for their financial support, as well as the partners of the Skills Matching project. Furthermore, we would like to thank Quirine Smit for helping with the data analysis, and Jok Tang and Stephan Raaijmakers for internally reviewing this paper.

References

[1] E. Rahm, P. A. Bernstein, A survey of approaches to automatic schema matching, the VLDB Journal 10 (2001) 334–350.

[2] J. Euzenat, P. Shvaiko, et al., Ontology matching, volume 18, Springer, 2007.

[3] I. Harrow, R. Balakrishnan, E. Jimenez-Ruiz, S. Jupp, J. Lomax, J. Reed, M. Romacker, C. Senger, A. Splendiani, J. Wilson, et al., Ontology mapping for semantically enabled applications, Drug discovery today 24 (2019) 2068–2075.

[4] Y. Zhang, X. Wang, S. Lai, S. He, K. Liu, J. Zhao, X. Lv, Ontology matching with word embeddings, in: Chinese computational linguistics and natural language processing based on naturally annotated big data, Springer, 2014, pp. 34–45.

[5] M. Tounsi Dhouib, C. Faron Zucker, A. G. Tettamanzi, An ontology alignment approach combining word embedding and the radius measure, in: International Conference on Semantic Systems, Springer, Cham, 2019, pp. 191–197.

[6] L. L. Wang, C. Bhagavatula, M. Neumann, K. Lo, C. Wilhelm, W. Ammar, Ontology alignment in the biomedical domain using entity definitions and context, arXiv preprint arXiv:1806.07976 (2018).

[7] P. Kolyvakis, A. Kalousis, D. Kiritsis, Deepalignment: Unsupervised ontology matching with refined word vectors, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 787–798.

[8] S. Neutel, M. H. T. de Boer, Towards automatic ontology alignment using BERT, in: AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering, 2021.

[9] Y. He, J. Chen, D. Antonyrajah, I. Horrocks, BERTMap: A BERT-based Ontology Alignment System, arXiv preprint arXiv:2112.02682 (2021).

[10] L. Stojanovic, B. Motik, Ontology Evolution within Ontology Editors, in: EON, 2002, pp. 53–62.

[11] F. Zablith, G. Antoniou, M. d'Aquin, G. Flouris, H. Kondylakis, E. Motta, D. Plexousakis, M. Sabou, Ontology evolution: a process-centric survey, The knowledge engineering review 30 (2015) 45–75.

[12] F. Osborne, E. Motta, Pragmatic ontology evolution: reconciling user requirements and application performance, in: International Semantic Web Conference, Springer, 2018, pp. 495–512.

[13] L. Stojanovic, Methods and tools for ontology evolution (2004).

[14] F. Zablith, Evolva: A comprehensive approach to ontology evolution, in: European Semantic Web Conference, Springer, 2009, pp. 944–948.

[15] C. Room, Levenshtein distance, algorithms 12 (2019) 32.

[16] M. Honnibal, I. Montani, spacy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, To appear 7 (2017) 411–420.

[17] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, M. Nissim, Bertje: A dutch bert model, arXiv preprint arXiv:1912.09582 (2019).

[18] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Advances in neural information processing systems 32 (2019).

[19] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[20] J.-J. Decorte, J. Van Hautte, T. Demeester, C. Develder, JobBERT: Understanding job titles through skills, arXiv preprint arXiv:2109.09605 (2021).