Advancing Ontology Alignment in the Labor Market: Combining Large Language Models with Domain Knowledge

Lucas L. Snijder[1, 2], **Quirine T. S. Smit**[1], **Maaike H. T. de Boer**[1]
[1] TNO, dep. Data Science
[2] Eindhoven University of Technology
lucasnijder@gmail.com, quirine.smit@tno.nl, maaike.deboer@tno.nl

Abstract

One of the approaches to help the demand and supply problem in the labor market domain is to change from degree-based hiring to skill-based hiring. The link between occupations, degrees and skills is captured in domain ontologies such as ESCO in Europe and O*NET in the US. Several countries are also building or extending these ontologies. The alignment of the ontologies is important, as it should be clear how they all relate. Aligning two ontologies by creating a mapping between them is a tedious task to do manually, and with the rise of generative large language models like GPT-4, we explore how language models and domain knowledge can be combined in the matching of the instances in the ontologies and in finding the specific relation between the instances (mapping refinement). We specifically focus on the process of updating a mapping, but the methods could also be used to create a first-time mapping. We compare the performance of several state-of-the-art methods such as GPT-4 and fine-tuned BERT models on the mapping between ESCO and O*NET and ESCO and CompetentNL (the Dutch variant) for both ontology matching and mapping refinement. Our findings indicate that: 1) Match-BERT-GPT, an integration of BERT and GPT, performs best in ontology matching, while 2) TaSeR outperforms GPT-4, albeit marginally, in the task of mapping refinement. These results show that domain knowledge is still important in ontology alignment, especially in the updating of a mapping in our use cases in the labor domain.

Introduction

One of the challenges in our current (European) society is the labor market (Green and Henseke 2021). Despite many job openings and many unemployed people, they struggle to find matching jobs. A skill-based approach can aid in this struggle between demand and supply (Brunello and Wruuck 2019). For example, during the COVID pandemic, many flight attendants in the Netherlands transitioned to healthcare jobs[1] [2], leveraging their skills in a new domain. These skill-based approaches ask for a common and up-to-date skill language that describes how occupations and skills relate in order to achieve their full potential. This language should be a common language across the world, but of course there are differences between countries. Therefore, it is preferable to have one for each country or region that is aligned with all other standards for interoperability. The term 'aligned' means that the entities used in one ontology should be matched to one or more entities in the other ontology, forming a mapping. In that mapping, the specific relations are included, such as *broader than* or *is-a*. As an example, the occupancy 'Head Chef' in ESCO[3] should be matched to the occupancy 'Chefs and Head Cooks' in O*NET[4], with a *broadMatch* relation (meaning the head chef is more specific than chefs and head cooks). In this example, notice that both occupation classifications refer to the same job but are phrased differently. Additionally, the O*NET occupation is more broad than the ESCO occupation. These two aspects are among the challenges that make the matching of entities between two ontologies challenging.

In literature, ontology alignment is not a new topic, with initial mentions dating back to 1986 (Batini, Lenzerini, and Navathe 1986). Currently, many implementations exist of automatic or semi-automatic methods, but research is still ongoing (Portisch, Hladik, and Paulheim 2022). Recent advancements in generative large language models have opened up new possibilities. This paper investigates how combining these language models with domain-specific knowledge can enhance ontology alignment. Our main contributions in this paper are: 1) an exploration of GPT-4 in the labor market domain using two use cases, with a focus on updating mappings; 2) a new method using BERT and GPT named Match-BERT-GPT for ontology matching; and 3) a comparison of several mapping refinement methods, including STROMA, TaSeR, and GPT-4.

In the next section, we give a short overview of the related work in this field. Afterwards, we explain our experiments in which we outline the use cases, methods, results and discussions. We end the paper with a conclusion and foreseen next steps for future work.

[1] https://www.parool.nl/nederland/klm-helpt-personeel-met-carriereswitch-naar-de-zorg~b0152dc5/

[2] https://www.volkskrant.nl/nieuws-achtergrond/nu-stewardessen-hun-baan-verliezen-staat-de-zorg-klaar-om-ze-een-nieuwe-te-bieden~b40b3bed/

[3] https://esco.ec.europa.eu/en

[4] https://www.onetonline.org/find/all

Related Work

In this related work we focus on ontology alignment and mapping refinement. We first explain some history in ontology alignment, including the shift towards language-based approaches, starting with non-contextual word embeddings, to contextual word embeddings and now to generative large language models. Then we focus on mapping refinement, where we look at methods that refine the relations of existing mappings. We finish the related work with an overview of ontology alignment and mapping refinement in the labor market.

Ontology Alignment

Ontology alignment has been extensively researched, dating back to the 80s of the last century (Batini, Lenzerini, and Navathe 1986). In this paper, we follow the definition of ontology alignment introduced by Euzenat (2013), with one modification: we do not incorporate a confidence measure. We start with the formalization of an ontology, which consists of a set of classes, a set of instances, a set of relations, a set of data types, a set of values and the specialization, exclusion, instantiation and assignment between them. A source ontology O_1 and a target ontology O_2 are aligned via a mapping, which is a set of correspondences between them. These correspondences are defined as a tuple containing a head, tail and relation (h, t, r), where:

- h is an entity in O_1,
- t is an entity in O_2,
- r is the semantic relationship between h and t such as *equivalence* ($=$), *is-a* ($<$),

Many methods have been proposed in ontology alignment (Euzenat and Shvaiko 2013; Ardjani, Bouchiha, and Malki 2015; Harrow et al. 2019). We follow the distinction between element-level and structure-level (Euzenat and Shvaiko 2013). Element-level techniques consider each entity in isolation, subdivided into string-based, language-based, and informal resources-based techniques. String-based techniques focus on the text associated with entities, whereas language-based techniques use Natural Language Processing (NLP) methods. Informal resources-based techniques deduce relations based on associated resources like links or pictures. Structure-level techniques, on the other hand, use connections to surrounding entities for additional information. This includes graph-based and taxonomy-based techniques. Several methods blend element- and structure-level techniques (Euzenat and Shvaiko 2013), such as the COMA++ framework (Aumueller et al. 2005), which combines string- and language-based techniques with structure-level techniques by comparing sub-trees. Other notable frameworks using a mix of these techniques are QOM (Ehrig and Staab 2004), AgreementMaker (Cruz, Antonelli, and Stroe 2009), YAM++ (Ngo and Bellahsene 2012), and LogMap (Jiménez-Ruiz and Cuenca Grau 2011).

Language-Based Techniques: Using External Knowledge Within the language-based techniques, a distinction between intrinsic and extrinsic techniques can be made. Intrinsic techniques are solely using the information within words and sentences, whereas extrinsic techniques incorporate external sources (Euzenat and Shvaiko 2013). Examples of such external sources are lexicons and thesauri such as WordNet (Miller 1995). OLA (Euzenat et al. 2004), SAMBO (Lambrix and Tan 2006), Falcon (Hu and Qu 2008), and AgreementMaker (Cruz, Antonelli, and Stroe 2009) translate entities into WordNet senses for matching, utilizing the information in WordNet for this purpose.

In recent years, other forms of external sources have been used such as word embeddings. A first application of (non-contextual) word embeddings for ontology alignment was introduced by Zhang et al. in 2014 (Zhang et al. 2014), combining a Wikipedia-trained Word2Vec model (Mikolov et al. 2013) with edit distance metrics. This method achieved good results, prompting further research in using word embeddings for ontology alignment (Vieira and Revoredo 2017; Gromann and Declerck 2018; Wang et al. 2018; Kolyvakis, Kalousis, and Kiritsis 2018; Nkisi-Orji et al. 2019; Giabelli et al. 2022).

Building upon these developments, contextual embeddings have emerged as a significant advancement and have made their way into ontology alignment. This is exemplified by methods such as BERTMap (He et al. 2022) and Truveta Mapper (Amir et al. 2023).

Generative Large Language Models The emergence of generative large language models, such as GPT-4 (OpenAI et al. 2023), hold potential for enhancing ontology matching. Generative large language models are trained on a massive amount of data, and trained using multi-task training. Early research indicates their significant potential for ontology matching. For example, He et al. (2023) show that generative large language models have the potential to outperform ontology alignment systems such as BERTMap given proper prompt engineering, as tested on the Ontology Alignment Evaluation Initiative (OAEI) Bio-ML track (He et al. 2021). In the OAEI 2022 challenge, Saki Norouzi, Mahdavinejad, and Hitzler (2023) demonstrated the first application of ChatGPT for ontology alignment. They indicate achieving high recall but low precision, along with the capability to create new triples. However, they also face issues due to limitations in context length, handling inverse functional properties, and unwanted matching with subclasses. Hertling and Paulheim (2023a) integrated generative large language models functionality in their framework to make it feasible to run with their MELT framework as used in OAEI (Hertling, Portisch, and Paulheim 2019). They use Sentence-BERT (Reimers and Gurevych 2019) to generate possible candidates. Generative large language models can then be used in a binary setting, outputting whether a candidate is correct or not, or in a multiple-choice setting, in which the most likely concept is chosen from the different possibilities. The proposed methods show state-of-the-art performance on several OAEI tasks. To the best of our knowledge, generative large language models have mainly be used in ontology alignment on the OAEI tasks. In this paper, we explore the usage of generative large language models in different use cases in the labor market domain.

Mapping Refinement

The definition of ontology alignment extends beyond mere equivalence between entities to encompass a variety of relations. Despite this, the methods discussed thus far predominantly concentrate on establishing equivalence correspondences. There exists a subset of ontology alignment techniques that acknowledge and incorporate relations beyond equivalence (Giunchiglia, Shvaiko, and Yatskevich 2004; Li et al. 2009; Jean-Mary, Shironoshita, and Kabuka 2009; Hamdi et al. 2010; Jain et al. 2010; Spiliopoulos, Vouros, and Karkaletsis 2010; Zong et al. 2015; Chen et al. 2023). However, the majority still focus solely on equivalence. This focus presents an opportunity for mapping refinement: existing equivalence-only alignments can be enhanced to include a broader range of semantic relations. Mapping refinement is defined as refining the relations in existing mappings into more specific relations (Arnold and Rahm 2014).

We will zoom in on a few of the mapping refinement methods. The first method was proposed by Arnold and Rahm (2014). Their method, Semantic enrichment of Ontology Mappings (STROMA), is a technique to enhance the semantic quality of ontology matches. Besides *equivalence* relations they include *is-a*, *disjointness* and *overlap* relations. STROMA uses 5 strategies to determine the relation type: compound (linguistic), background knowledge (such as WordNet), itemization (handles lists), structure (hierarchical organization) and multiple linkage (based on involvement of elements in other relations). Second, Tounsi Dhouib, Faron, and Tettamanzi (2021) propose a mapping refinement approach based on a set of rules exploiting the embedding space and measuring clusters of entity labels to discover the relationship of correspondences. They show that the combination of word embeddings and a measure of dispersion of the clusters of the entities in the embedding space makes it possible to determine the semantic relation between entities. Third, inspired by knowledge graph completion methods such as KG-BERT (Yao, Mao, and Luo 2019), Hertling and Paulheim (2023b) trained a BERT model for mapping refinement, named Transformer-based Semantic Relation Typing (TaSeR). Their method includes various semantic relationships such as *equivalence*, *is-a*, *inverse is-a*, *part of*, *has part*, *co-hyponym*, and *disjointness*. TaSeR is fine-tuned on a general data set for mapping refinement, and can be further fine-tuned for the test case at hand. So far, there have been no publications yet that include generative large language models in mapping refinement. In this study, we will explore the potential of such methods in mapping refinement.

Ontology Alignment and Mapping Refinement in the Labor Market

The labor market knows several ontologies or taxonomies. One of the international standards is ESCO. ESCO stands for the European Skills, Competences, Qualifications and Occupations and is an ontology that includes occupations, skills and competences (Smedt, le Vrang, and Papantoniou 2015). It is built upon ISCO, the International Standard Classification of Occupations [5]. Based on or related to the European standard, several specific country related ontologies or taxonomies exist, such as the Dutch CompetentNL[6] (CNL) and the German Labor Market Ontology (GLMO) (Dörpinghaus et al. 2023).

Another international standard is the North American labor market ontology O*NET. It is built on top of SOC, the Standard Occupational Classification[7]. O*NET, the Occupational Information Network, is a thesaurus / database that contains occupations, workforce characteristics, occupational requirements, worker characteristics, worker requirements and experience requirements (Peterson et al. 1999; Cifuentes et al. 2010). A first version was published over twenty years ago, and a new version is available every few years, based on input from experts, job holders and job postings.

Several mappings between ontologies with roots in ISCO and SOC have been created. To stay concise, we will only focus on the mappings that have been created using automatic or semi-automatic methods. The first method that mapped ESCO to O*NET was the fully automatic BERT-based method by Neutel and de Boer (2021). A similar semi-automatic approach was proposed by Frugoli (2022). Later, Guru Rao et al. (2022) proposed a XLNet-based method. Related to the mapping between ESCO and CompetentNL, de Boer, Bakker, and Burghoorn (2023) have performed experiments with the Dutch mapping and the English mapping, both on skills and occupations.

Last year, a challenge was published focused on the evolving ontologies of ESCO and O*NET (de Boer, Oosterheert, and Bakker 2023) and keeping mappings between them up-to-date. To this date, there appear to be no articles addressing this specific challenge. In this paper, we take the first steps towards a solution for this challenge, investigating the potential of various models, including generative large language models, for updating and refining existing mappings. Furthermore, this study is the first to explore mapping refinement within the labor market domain, a combination of topics that, to our knowledge, has not been previously examined.

Experiments

In this section, we explain the experimental setup in which we compare various state-of-the-art methods and generative large language models for the purpose of updating an existing ontology mapping. Based on the dynamic ontology matching challenge (de Boer, Oosterheert, and Bakker 2023) we chose to focus on a scenario where we want to update an existing mapping after new entities have been added to the source ontology. This updating scenario entails that we have an existing mapping that can be used as input. Note that our methods are structured in such a way that they can also be used in scenarios where there is not yet an existing mapping, known as first-time ontology alignment. The alignment update process has two phases, ontology matching and mapping refinement, both of which we will cover. We focus on the labor market domain and define two use

[5] https://www.ilo.org/public/english/bureau/stat/isco/

[6] https://www.competentnl.nl/

[7] https://www.bls.gov/oes/current/oes_stru.htm

cases, one on mapping the European standard ESCO to the American standard O*NET and one on mapping ESCO to the Dutch standard CompetentNL. We chose to use GPT-4 (OpenAI et al. 2023) as our generative large language models in all experiments because its performance has been best so far on various benchmarks (López Espejel et al. 2023).

In the ontology matching experiment, we use a SpaCy model as our baseline model, as it performs state-of-the-art in ontology matching on similar data (de Boer, Bakker, and Burghoorn 2023). Second, we use a BERT model as the state-of-the-art transformer method, in which we add domain knowledge to the existing foundation model using fine-tuning. We name this method Match-BERT. Third, we use a similar setup as Olala (Hertling and Paulheim 2023a), in the sense that we use SpaCy / Match-BERT to generate candidate matches as a first step and ask the generative large language model to pick the best candidate match. Our GPT methods are different from Olala in both the way the candidate generation model is trained and the generative large language model used.

For the mapping refinement experiment, we use state-of-the-art methods STROMA and TaSeR, as they are mapping refinement methods that can handle both *equivalence* and *is-a* relations. STROMA serves as the baseline model as it is rule based. TaSeR is the state-of-the-art transformer method, and allows us to add the domain knowledge from the ontologies in the use cases. This method combines element-based and structure-based aspects, as we use the hierarchical structure from the ontologies as well as the element information for each occupation. We compare both methods to GPT-4 directly. Because of the limitations of the use cases, we choose to only use the relations *exactMatch*, *broadMatch* and *narrowMatch*, as defined in SKOS (Miles and Pérez-Agüera 2007).

First, we will explain the use cases used in the experiments. Then, for the first part of the experiments, we will look at our proposed methods for ontology matching. Afterwards, we will look at our proposed methods for mapping refinement.

Use Cases

We define two labor market related use cases: ESCO-O*NET and ESCO-CNL. Both use cases consist of two ontologies, with exclusively *is-a* relations, and an existing mapping between them containing *exactMatch*, *broadMatch* and *narrowMatch* relations. The ESCO-O*NET use case consists of the ESCO occupation ontology version 1.1.1 [8], the O*NET occupation ontology version 2019 [9] and the mapping between them. The existing mapping exists of 2778 correspondences. The ESCO-CNL use case combines the English translations of ESCO's occupation ontology version 1.1.1 and CompetentNL's occupation ontology version 0.91, along with their mapping. Both CompetentNL and the mapping to ESCO are currently private, but will be available once the development of CompetentNL is finished. The existing mapping exists of 5475 correspondences. We use the hierarchical information from the separate ontologies to create a train set, which is explained later in more detail. This is the domain knowledge we include in the methods. We use the existing mapping as the test set for both matching and refinement.

[8]https://esco.ec.europa.eu/

[9]https://www.onetonline.org/

Ontology Matching

In the ontology matching experiment, the goal is to find a match between occupations in the test set (from the source ontology) and occupations in the target ontology. We compare each occupation in the ESCO test set against all occupations in O*NET or CNL and predict whether the two occupations are a match (1) or not (0). We evaluate the methods using the precision, recall, and F1 scores for the true class, using confusion matrices for support. We prioritize the true class because finding true correspondences is our priority. True negatives are largely ignored due to their abundance and minimal insight contribution. To get a sense of the difficulty of the task, we include random classifications in our results. In the following sections, we will first look at the test set used, and afterwards we will cover the various methods.

Test Set We aim to simulate an update of an existing mapping to incorporate new occupations. This involves dividing the current mapping into training and test sets. The training set, used during the update, facilitates model training, while the test set represents new occupations to be added, unseen during training. The existing mappings are split up randomly according to a 80/20 train/test, and the relations are removed to give us only the pairs of occupations, as we are only interested in equivalence correspondences during matching. The existing ESCO-O*NET mapping has 2778 pairs. After splitting, the train set exists of 2222 pairs and the test set exists of 556 pairs. The existing ESCO-CNL mapping has 5475 pairs. After splitting, the train set exists of 4380 pairs and the test set exists of 1095 pairs.

SpaCy For our SpaCy method we use the English word vector model released by SpaCy, with 514157 unique vectors, each of 300 dimensions (Honnibal and Montani 2017). This model is trained on a large web corpus (*en_core_web_lg*). For each occupation in the test set, we compare the embedding to the embeddings of all occupations in the target ontology and calculate the cosine similarity scores. For each occupation we rank the possible matches by their cosine similarity and use the top k with the highest cosine similarity as the output. After conducting tests with various values of k in relation to the F1-score, we determined that the optimal value is 5. In figure 1 the SpaCy method is visualised.

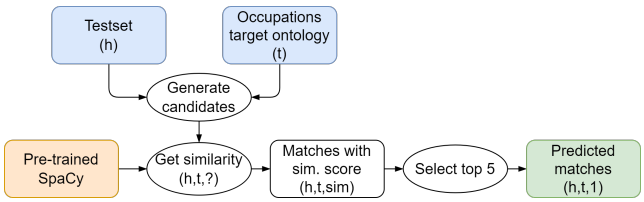

Figure 1: Diagram of the SpaCy Method

Match-BERT For the Match-BERT method we create a train set using the connections between the occupations via the *is-a* relations. We transform the ontology into a list of pairs consisting of a head, tail and a label to indicate that the pair is true $(h, t, 1)$, where each pair has an *is-a* relation between the head and tail occupations in one ontology. In order to give the model the ability to not only find *is-a* pairs but also *equivalence* pairs in the prediction phase, we create synonym pairs by matching each occupation to itself $(h, h, 1)$. After dropping duplicates, the *is-a* and synonym pairs have a total of 10740 pairs for ESCO, 3959 pairs for O*NET and 12922 pairs for CNL. These pairs form the positives of the training set. Negatives are generated by shifting the tails 300 steps down while keeping the heads in place, for each ontology separately, giving us negatives pairs $(h, t, 0)$. The number 300 is chosen because this ensures that the heads are not matched to a related tail, which should be avoided as such cases would be false negatives. In addition, the pairs of the train set of the existing mapping are added to the set as positives. This step can be left out in case of first-time ontology alignment. Given the positive and negative pairs, the pairs $(h, t, 1/0)$ are used as input data.

We use the BERT-base-uncased model (Devlin et al. 2018) with an additional pooling and linear classification layer so it can perform relation prediction. The relation prediction task has the objective of predicting the relation between a given pair of entities, in our case the head and tail $(h, t, ?)$. We want to predict whether there is a relation between two entities yes or no $(1/0)$. We use a learning rate of 1e-5, 4 epochs and a batch size of 64. In figure 2 the Match-BERT method is visualised. For more details on fine-tuning BERT for relation prediction, we refer to the KG-BERT paper by Yao, Mao, and Luo (2019).

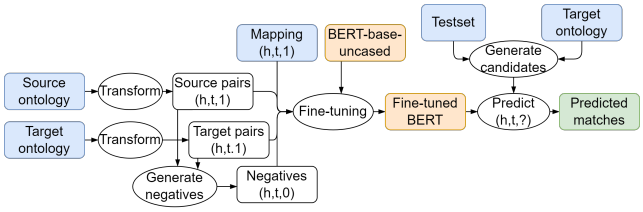

Figure 2: Diagram of the Match-BERT method

SpaCy-GPT The SpaCy-GPT method combines the outputs of the SpaCy method with GPT-4. Firstly, we take the potential pairs that the SpaCy method identified. For each occupation, the 5 potential pairs are given to GPT-4, which evaluates and selects the most appropriate one. We use the *gpt-4-1106-preview* model configured using the default settings. We specify the *system role* as: *'You are a occupational research expert. You know what tasks and responsibilities every occupation has.'*, and the *user's message* is specified as: *'From this list of matched occupations, decide which of the pairs is the most correct. Pick at most one pair. Only output the pair without any extra words, characters or symbols. The candidate pairs are: \n {formatted_string}'*. Where *formatted_string* has all candidate pairs for the given occupation.

Match-BERT-GPT This method is similar to the SpaCy-GPT method, the only difference being that it uses the Match-BERT predicted pairs. In figure 3 the SpaCy-GPT and Match-BERT-GPT methods are visualised.

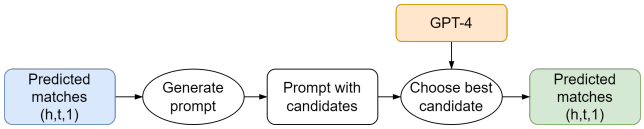

Figure 3: Diagram of the SpaCy/Match-BERT + GPT methods

Results The results for the ontology matching methods SpaCy (S), Match-BERT (MB), SpaCy-GPT (S-GPT) and Match-BERT-GPT (MB-GPT) can be found in table 1. The confusion matrices are presented in tables 2 and 3. Since the results between the two use cases are very similar, we only included the confusion matrices of the ESCO-O*NET use case.

Model	ESCO-O*NET			ESCO-CNL		
	P	R	F1	P	R	F1
Random	.001	.001	.001	.0004	.0004	.0004
S	.07	.30	.11	.05	.18	.08
MB	.01	.99	.02	.00	.94	.01
S-GPT	.28	.26	.27	.17	.12	.14
MB-GPT	.55	.48	**.51**	.61	.44	**.51**

Table 1: Precision, recall, and F1-scores for the ESCO-O*NET and ESCO-CNL use cases using our matching methods

		Predicted	
		False	**True**
Actual	**False**	534587	2321
	True	387	169

		Predicted	
		False	**True**
False		477187	60277
True		8	548

Table 2: Confusion matrices for SpaCy and Match-BERT on the ESCO-O*NET use case, respectively

		Predicted	
		False	**True**
Actual	**False**	536530	378
	True	409	147

		Predicted	
		False	**True**
False		536687	221
True		287	269

Table 3: Confusion matrices for SpaCy-GPT and Match-BERT-GPT on the ESCO-O*NET use case, respectively

Discussion Analyzing the base models, we observed that both Match-BERT and SpaCy demonstrated higher recall than precision, with Match-BERT showing an imbalance of almost zero precision and near-perfect recall. The confusion matrices in table 2 provide more details, showing many samples in the predicted true column. The inclusion of GPT-4, particularly when combined with Match-BERT, significantly improves the performance of our matching methods. The combination leads to significant gains in F1-scores,

most notably with the Match-BERT-GPT model. The combination of SpaCy with GPT-4 somewhat compensates for SpaCy's low recall, yet this approach is less effective compared to the Match-BERT-GPT method.

Compared to previous research, such as Olala's study, the scores are relatively low. However, the use cases are not comparable. The Bio-ML use cases that are used for evaluating Olala have been created specifically for evaluating ontology alignment methods. The labor market use cases used in this study have not been created specifically for evaluation, likely increasing the chance of inconsistencies and ambiguities.

A limitation of the SpaCy-GPT method is that we only use a *k* of 5 in the top *k* highest similarities. Choosing a larger *k* would have increased the recall (but reduced the precision) of the SpaCy method, likely improving the SpaCy-GPT method's performance by increasing the chances of including the correct answer among the candidates.. Another limitation of our GPT-methods is that it only outputs the single best tail, but in some cases one ESCO occupation can be mapped to multiple O*NET or CNL occupations. This could increase the recall, and potentially, the precision in those cases.

Mapping Refinement

In the mapping refinement experiments we simulate the scenario in which you have matched pairs, and want to determine the relation between them. In order to do this we predict, for each given pair of occupations, the relation between the occupations, the options being *broadMatch*, *narrowMatch* and *exactMatch*. The matched pairs for which we try to predict the relation are the pairs from the test set of the existing mapping. We evaluate the performance using the macro F1 score and confusion matrices. The macro F1 score is calculated without considering the class sample sizes. This approach is particularly appropriate here due to the significant class imbalance in the test sets. The ESCO-O*NET mapping contains a relatively large number of *broadMatch* triples, while the ESCO-CNL mapping contains a large number of *narrowMatch* triples. We add the scenario of random classifications to the results to get a sense of the difficulty of the task at hand. In the following sections, we will first look at the test set used, and afterwards we will cover the various methods.

Test Set The test set is created in the same way as for matching. The only difference being that we include the relation, providing triples (h, t, r) instead of pairs (h, t). Similar to the matching experiments, we use a 80/20 split on the existing mapping to create a train and a test set. After splitting, the train set for the ESCO-O*NET use case exists of 1634 *broadMatch*, 180 *narrowMatch* and 408 *exactMatch* triples. The test set exists of 419 *broadMatch*, 47 *narrowMatch* and 90 *exactMatch* triples. For the ESCO-CNL use case, after splitting, the train set has of 1552 *broadMatch*, 2039 *narrowMatch* and 789 *exactMatch* triples. The resulting test set exists of 405 *broadMatch*, 493 *narrowMatch* and 197 *exactMatch* triples.

STROMA The STROMA method is a rule-based mapping refinement method that combines intrinsic and extrinsic language-based techniques (Arnold and Rahm 2014). The following strategies are used: compounds, background knowledge , itemization, structural and multiple linkage. It does not need data to be trained. STROMA predicts six classes: *equivalence*, *is-a*, *inverse is-a*, *part-of* and *inverse part-of* relations and *noMatch*. In figure 4 the STROMA method is visualised.

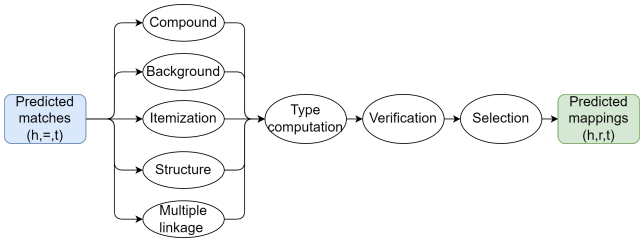

Figure 4: Diagram of the STROMA method

TaSeR In the previous experiments in ontology matching, we transformed the ontologies into a list of pairs (h, t). For mapping refinement with TaSeR, we create a similar list, but with triples (h, t, r), with a head, tail and a relation. Similar to matching, the head and tail represent the occupations. Additionally, the relation represents the relation between them. The possible SKOS relations are: *broadMatch*, *narrowMatch* and *exactMatch*. Using the inverse properties of the *is-a* relation, we can create both the triple $(h, t, broadMatch)$ and the triple $(t, h, narrowMatch)$ from the same *is-a* relation. The *exactMatch* is created by using the same occupation as both the head and the tail $(h, h, exactMatch)$. In addition, for ESCO there are alternative labels available for most of the occupations. These are also used as *exactMatch* triples. We combine the triples from the ontologies with the train set of the existing mapping into one train data set.This step can be left out in case of first-time ontology alignment.

For the ESCO-O*NET use case, the whole ESCO ontology is transformed into 14,682 triples, of which 3,580 are *broadMatch*, 3,580 are *narrowMatch* and 7,522 are *exactMatch* triples (the higher number is caused by the use of the alternative labels). The O*NET ontology is transformed into 3,936 triples, 1,312 of each relation type. The triples from the ontologies are then combined with the train set of the existing mapping, to get the complete train set. Lastly, we filter out duplicates from this train set based solely on their head and tail, excluding the relation, to counter ambiguities. The distribution of the final train set can be seen in table 4.

For the ESCO-CNL use case, the same ESCO triples are used. The CNL ontology is transformed into 1,922 triples, of which 4,322 *broadMatch*, 4,322 *narrowMatch* and 4,278 *exactMatch* triples. The lower number of *exactMatch* triples is caused by dropping duplicates by filtering on the head and tail. Again, the triples from the ontologies are then combined with the train set of the existing mapping to get the complete train set and the duplicates are filtered out. The distribution of the final train set can be seen in table 4.

Relation	ESCO-O*NET	ESCO-CNL
exactMatch	9,232	11,756
broadMatch	5,062	8,931
narrowMatch	6,526	9,418
Total	20,820	30,105

Table 4: An overview of the number of triples use for training per use case

Similar to our Match-BERT method for matching, Hertling and Paulheim's TaSeR method uses a DistilBERT-base-uncased model with an additional pooling and linear classification layer so it can perform relation prediction on $(h, t, ?)$. The relations in this case are *broadMatch*, *narrowMatch* and *exactMatch*. The TaSeR method exists of two fine-tune phases: pre-fine-tuning and fine-tuning. The pre-fine-tune phase fine-tunes a DistilBERT-base-uncased model on a collection of datasets containing triples based on WordNet, DBpedia, Schema.org and Wikidata. In the second phase, the pre-fine-tuned model is further fine-tuned on the triples from the use-case-specific train datasets. The pre-fine-tuned model has been published by the authors on Hugging Face[10]. We use a learning rate of 1e-5, 4 epochs and a batch size of 64. In figure 5 the TaSeR method is visualised. For more details on fine-tuning BERT for relation prediction, we refer to the KG-BERT paper by Yao, Mao, and Luo (2019).

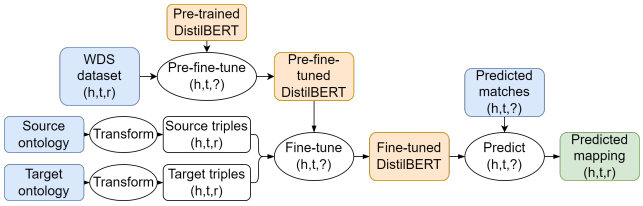

Figure 5: Diagram of the TaSeR method

GPT-4 The GPT-4 method uses prompting to establish the relation between two given entities. We use the *gpt-4-1106-preview* version configured using the default settings. We specify the *system role* as: *'You are a relation prediction expert. You know the relation between two given concepts. The choices of relations are exactMatch, narrowMatch and broadMatch. Only output the predicted relation in without any extra words, characters or symbols. Here is a list of examples:* {*example_string*}*'*, and the *user's message* is specified as: *'Determine the relation between the concepts:* [*concept*1], *and:* [*concept*2]*'*. Where {example_string} has 50 example triples from the existing mapping. By incorporating the examples, we adopt a few-shot learning approach. In figure 6 the GPT-4 method is visualised.

[10]https://huggingface.co/dwsunimannheim/TaSeR

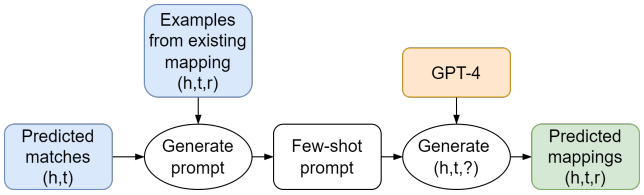

Figure 6: Diagram of the GPT-4 method

Results The macro averaged metrics are shown in table 5. The confusion matrices for TaSeR and GPT-4 are presented in tables 6 and 7. Given that the results for STROMA were worse than random, we omitted the corresponding confusion matrices. The labels used in the confusion matrices are abbreviations for *broadMatch* (BM), *narrowMatch* (NM) and *exactMatch* (EM).

Model	ESCO-O*NET			ESCO-CNL		
	P	R	F1	P	R	F1
Random	.33	.33	.33	.33	.33	.33
STROMA	.67	.27	.38	.71	.12	.20
TaSeR	.80	.58	**.64**	.69	.71	**.68**
GPT-4	.64	.61	.62	.53	.52	.51

Table 5: Comparison of macro average Precision, Recall, and F1-Score for STROMA, TaSeR, and GPT-4 across use cases

		Predicted		
		BM	NM	EM
Actual	BM	414	4	1
	NM	21	21	5
	EM	58	4	28

	Predicted		
	BM	NM	EM
BM	283	35	87
NM	48	336	109
EM	24	26	147

Table 6: Confusion matrices for TaSeR on the ESCO-O*NET and ESCO-CNL use cases

		Predicted		
		BM	NM	EM
Actual	BM	376	16	27
	NM	30	13	4
	EM	26	4	60

	Predicted		
	BM	NM	EM
BM	184	187	34
NM	197	266	30
EM	31	57	109

Table 7: Confusion matrices for GPT-4 on the ESCO-O*NET and ESCO-CNL use cases

Discussion If we compare the methods to the random baseline, we see that all are performing better than random except for STROMA, which is very close to random for ESCO-O*NET, and worse than random for ESCO-CNL. These results show that STROMA is not fit for real-world applications in the labor market domain. The results for TaSeR are the best of all methods, with GPT-4 following closely. When comparing the performance of TaSeR and GPT-4 between the two use cases, we find that the difference is smaller for ESCO-O*NET compared to ESCO-CNL. An explanation for this could be the fact that ESCO and O*NET

were included in the dataset used during the pre-training of GPT-4, and CNL was not since it has not yet been published. The results suggest that including or excluding ontologies in the training dataset affects the performance of large generative language models in downstream tasks, especially in mapping refinement. For a detailed view into the performance of TaSer and GPT-4 we can look at the confusion matrices. These show that TaSeR has a slight tendency towards wrongly predicting relations as *broadMatch* for the ESCO-O*NET use case, and a slight tendency towards wrongly predicting relations as *exactMatch* for the ESCO-CNL use case. This last finding could be explained by the fact that CNL is more fine-grained than O*NET, meaning that the differences between the occupations are also more subtle, making it likely that *narrowMatch* or *broadMatch* relations are classified as *exactMatch*. GPT-4's errors are more uniformly distributed across all relations, and it offers the advantage of not requiring task-specific fine-tuning due to its independence from available data. Despite its slightly lower performance, these factors contribute to GPT-4's robustness in mapping refinement tasks.

Conclusion

This study delved into the use of generative large language models like GPT-4 for ontology alignment in the labor market domain, focusing on ontology matching and mapping refinement. For ontology matching, we found that GPT-4 significantly enhances methods that use non-contextual and contextual word embeddings, with the Match-BERT and GPT-4 combination being particularly effective in reducing false positives included in the prediction by Match-BERT. In mapping refinement, our analysis of STROMA, TaSeR, and a GPT-based method highlighted the importance of efficiently integrating domain knowledge. This was demonstrated by TaSeR's higher performance over GPT-4 even though GPT-4 received domain knowledge via examples. Our study emphasizes the potential of generative large language models in ontology alignment while highlighting the necessity of domain knowledge for their effective application.

Exploring the applicability of our findings beyond the labor market domain would be an interesting extension of our work to determine if the performance is consistent across different domains. Another possible extension of our work would be to put the matching and mapping refinement methods together in one pipeline. Furthermore, the lack of impact observed from adding domain knowledge examples to the GPT-4 mapping refinement method suggests an opportunity to enhance GPT-4 by exploring alternative approaches, such as varied prompting strategies or fine-tuning.

Acknowledgments

We would like to thank the project 'Vaardig met Vaardigheden' for their financial support and invaluable feedback throughout this research. Special thanks to Gino Kalkman for his pioneering efforts in GPT, which laid the groundwork for our study. Additionally, we thank Prof. George Fletcher for his generosity.

References

Amir, M.; Baruah, M.; Eslamialishah, M.; Ehsani, S.; Bahramali, A.; Naddaf-Sh, S.; and Zarandioon, S. 2023. Truveta Mapper: A Zero-shot Ontology Alignment Framework. ArXiv: 2301.09767.

Ardjani, F.; Bouchiha, D.; and Malki, M. 2015. Ontology-Alignment Techniques: Survey and Analysis. *International Journal of Modern Education and Computer Science*, 7(11): 67–78.

Arnold, P.; and Rahm, E. 2014. Enriching ontology mappings with semantic relations. *Data & Knowledge Engineering*, 93: 1–18.

Aumueller, D.; Do, H.-H.; Massmann, S.; and Rahm, E. 2005. Schema and ontology matching with COMA++. In *Proc. of the 2005 ACM SIGMOD*, 906–908.

Batini, C.; Lenzerini, M.; and Navathe, S. B. 1986. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 18(4): 323–364.

Brunello, G.; and Wruuck, P. 2019. *Skill shortages and skill mismatch in Europe – A review of the literature*. European Investment Bank.

Chen, J.; He, Y.; Geng, Y.; Jimenez-Ruiz, E.; Dong, H.; and Horrocks, I. 2023. Contextual Semantic Embeddings for Ontology Subsumption Prediction. ArXiv:2202.09791 [cs].

Cifuentes, M.; Boyer, J.; Lombardi, D. A.; and Punnett, L. 2010. Use of O*NET as a job exposure matrix: a literature review. *American journal of industrial medicine*, 53(9): 898–914.

Cruz, I. F.; Antonelli, F. P.; and Stroe, C. 2009. AgreementMaker: efficient matching for large real-world schemas and ontologies. *Proc. of the VLDB Endowment*, 2(2): 1586–1589.

de Boer, M. H.; Bakker, R. M.; and Burghoorn, M. 2023. Creating Dynamically Evolving Ontologies: A Use Case from the Labour Market Domain. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*.

de Boer, M. H. T.; Oosterheert, L.; and Bakker, R. M. 2023. Dynamic Ontology Matching Challenge. In Martin, A.; Fill, H.; Gerber, A.; Hinkelmann, K.; Lenat, D.; Stolle, R.; and van Harmelen, F., eds., *Proceedings of the AAAI 2023 Spring Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering*, volume 3433 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Dörpinghaus, J.; Binnewitt, J.; Winnige, S.; Hein, K.; and Krüger, K. 2023. Towards a German labor market ontology: Challenges and applications. *Applied Ontology*, 18: 343–365. 4.

Ehrig, M.; and Staab, S. 2004. QOM–quick ontology mapping. In *The Semantic Web–ISWC 2004. Proc. 3*, 683–697. Springer.

Euzenat, J.; Loup, D.; Touzani, M.; and Valtchev, P. 2004. Ontology alignment with OLA. In *Proc. 3rd ISWC2004 workshop on Evaluation of Ontology-based tools (EON)*, 59–68. Hiroshima, Japan.

Euzenat, J.; and Shvaiko, P. 2013. *Ontology Matching*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-38720-3 978-3-642-38721-0.

Frugoli, P. 2022. The crosswalk between ESCO and O*NET. Technical report, European Commission.

Giabelli, A.; Malandri, L.; Mercorio, F.; and Mezzanzanica, M. 2022. WETA: Automatic taxonomy alignment via word embeddings. *Computers in Industry*, 138. Publisher: Elsevier B.V.

Giunchiglia, F.; Shvaiko, P.; and Yatskevich, M. 2004. S-Match: an algorithm and an implementation of semantic matching. In *The Semantic Web: Research and Applications: First European Semantic Web Symposium, ESWS 2004, Proceedings 1*, 61–75. Springer.

Green, F.; and Henseke, G. 2021. Europe's evolving graduate labour markets: supply, demand, underemployment and pay. *Journal for Labour Market Research*, 55: 1–13.

Gromann, D.; and Declerck, T. 2018. Comparing Pretrained Multilingual Word Embeddings on an Ontology Alignment Task. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Guru Rao, S.; Oosterheert, L.; Bakker, R.; Wang, S.; Strisciuglio, N.; and Theune, M. 2022. *Ontology Matching using Background Knowledge and Semantic Similarity*. Master's thesis.

Hamdi, F.; Safar, B.; Reynaud, C.; and Niraula, N. 2010. TaxoMap alignment and refinement modules: Results for OAEI 2010. In *The Fifth International Workshop on Ontology Matching*, 212–219. Shanghai, China.

Harrow, I.; Balakrishnan, R.; Jimenez-Ruiz, E.; Jupp, S.; Lomax, J.; Reed, J.; Romacker, M.; Senger, C.; Splendiani, A.; Wilson, J.; et al. 2019. Ontology mapping for semantically enabled applications. *Drug discovery today*, 24(10): 2068–2075.

He, Y.; Chen, J.; Antonyrajah, D.; and Horrocks, I. 2021. Biomedical Ontology Alignment with BERT.

He, Y.; Chen, J.; Antonyrajah, D.; and Horrocks, I. 2022. BERTMap: A BERT-Based Ontology Alignment System. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5): 5684–5691.

He, Y.; Chen, J.; Dong, H.; and Horrocks, I. 2023. Exploring Large Language Models for Ontology Alignment. *arXiv preprint arXiv:2309.07172*.

Hertling, S.; and Paulheim, H. 2023a. OLaLa: Ontology Matching with Large Language Models. In *Proceedings of the 12th Knowledge Capture Conference 2023*, 131–139. Pensacola FL USA: ACM. ISBN 9798400701412.

Hertling, S.; and Paulheim, H. 2023b. Transformer Based Semantic Relation Typing for Knowledge Graph Integration. In *The Semantic Web*, volume 13870, 105–121. Cham: Springer Nature Switzerland. ISBN 978-3-031-33454-2 978-3-031-33455-9. Series Title: Lecture Notes in Computer Science.

Hertling, S.; Portisch, J.; and Paulheim, H. 2019. MELT - Matching EvaLuation Toolkit. In *International conference on semantic systems*, 231–245.

Honnibal, M.; and Montani, I. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1): 411–420.

Hu, W.; and Qu, Y. 2008. Falcon-AO: A practical ontology matching system. *Journal of Web Semantics*, 6(3): 237–239.

Jain, P.; Hitzler, P.; Sheth, A. P.; Verma, K.; and Yeh, P. Z. 2010. Ontology alignment for linked open data. In *International semantic web conference*, 402–417. Springer.

Jean-Mary, Y. R.; Shironoshita, E. P.; and Kabuka, M. R. 2009. Ontology matching with semantic verification. *Journal of Web Semantics*, 7(3): 235–251.

Jiménez-Ruiz, E.; and Cuenca Grau, B. 2011. LogMap: Logic-Based and Scalable Ontology Matching. In *The Semantic Web – ISWC 2011*, volume 7031, 273–288. Springer Berlin Heidelberg. Series Title: Lecture Notes in Computer Science.

Kolyvakis, P.; Kalousis, A.; and Kiritsis, D. 2018. DeepAlignment: Unsupervised Ontology Matching with Refined Word Vectors. In *Proc. of the 2018 Conference of the NAACL: Human Language Technologies, Volume 1 (Long Papers)*, 787–798. New Orleans, Louisiana: Association for Computational Linguistics.

Lambrix, P.; and Tan, H. 2006. SAMBO—A system for aligning and merging biomedical ontologies. *Journal of Web Semantics*, 4(3): 196–206.

Li, J.; Tang, J.; Li, Y.; and Luo, Q. 2009. RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. *IEEE Transactions on Knowledge and Data Engineering*, 21(8): 1218–1232.

López Espejel, J.; Ettifouri, E. H.; Yahaya Alassan, M. S.; Chouham, E. M.; and Dahhane, W. 2023. GPT-3.5, GPT-4, or BARD? Evaluating LLMs reasoning ability in zero-shot setting and performance boosting through prompts. *Natural Language Processing Journal*, 5: 100032.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. ArXiv:1301.3781 [cs].

Miles, A.; and Pérez-Agüera, J. R. 2007. SKOS: Simple Knowledge Organisation for the Web.

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11): 39–41.

Neutel, S.; and de Boer, M. H. T. 2021. Towards Automatic Ontology Alignment using BERT. In *Proceedings of the AAAI 2021 Spring Symposium: Combining Machine Learning with Knowledge Engineering*.

Ngo, D.; and Bellahsene, Z. 2012. YAM++ : A Multi-strategy Based Approach for Ontology Matching Task. In *Knowledge Engineering and Knowledge Management*, volume 7603, 421–425. Springer Berlin Heidelberg.

Nkisi-Orji, I.; Wiratunga, N.; Massie, S.; Hui, K.-Y.; and Heaven, R. 2019. Ontology Alignment Based on Word Embedding and Random Forest Classification. In *Machine*

Learning and Knowledge Discovery in Databases, volume 11051, 557–572. Springer International Publishing.

OpenAI; :; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; and et al. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Peterson, N. G.; Mumford, M. D.; Borman, W. C.; Jeanneret, P.; and Fleishman, E. A. 1999. *An occupational information system for the 21st century: The development of O*NET*. American Psychological Association.

Portisch, J.; Hladik, M.; and Paulheim, H. 2022. Background knowledge in ontology matching: A survey. *Semantic Web*, (Preprint): 1–55.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992.

Saki Norouzi, S.; Mahdavinejad, M.; and Hitzler, P. 2023. Conversational Ontology Alignment with ChatGPT. In *OM 2023: The 18th International Workshop on Ontology Matching collocated with the 22nd International Semantic Web*.

Smedt, J. D.; le Vrang, M.; and Papantoniou, A. 2015. ESCO: Towards a Semantic Web for the European Labor Market. In Bizer, C.; Auer, S.; Berners-Lee, T.; and Heath, T., eds., *Proceedings of the Workshop on Linked Data on the Web, LDOW 2015, co-located with the 24th International World Wide Web Conference*, volume 1409 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Spiliopoulos, V.; Vouros, G. A.; and Karkaletsis, V. 2010. On the discovery of subsumption relations for the alignment of ontologies. *Journal of Web Semantics*, 8(1): 69–88.

Tounsi Dhouib, M.; Faron, C.; and Tettamanzi, A. G. B. 2021. Measuring Clusters of Labels in an Embedding Space to Refine Relations in Ontology Alignment. *Journal on Data Semantics*, 10(3-4): 399–408.

Vieira, R.; and Revoredo, K. 2017. Using word semantics on entity names for correspondence set generation. In *OM@ ISWC*, 223–224.

Wang, L. L.; Bhagavatula, C.; Neumann, M.; Lo, K.; Wilhelm, C.; and Ammar, W. 2018. Ontology alignment in the biomedical domain using entity definitions and context. In Demner-Fushman, D.; Cohen, K. B.; Ananiadou, S.; and Tsujii, J., eds., *Proceedings of the BioNLP 2018 workshop*, 47–55. Melbourne, Australia: Association for Computational Linguistics.

Yao, L.; Mao, C.; and Luo, Y. 2019. KG-BERT: BERT for Knowledge Graph Completion. ArXiv:1909.03193 [cs].

Zhang, Y.; Wang, X.; Lai, S.; He, S.; Liu, K.; Zhao, J.; and Lv, X. 2014. Ontology Matching with Word Embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, volume 8801, 34–45. Springer International Publishing. Series Title: Lecture Notes in Computer Science.

Zong, N.; Nam, S.; Eom, J.-H.; Ahn, J.; Joe, H.; and Kim, H.-G. 2015. Aligning ontologies with subsumption and equivalence relations in Linked Data. *Knowledge-Based Systems*, 76: 30–41.