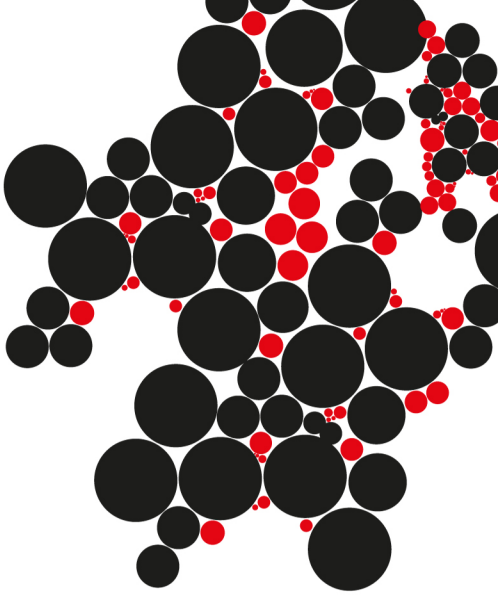

APEX: ZERO-SHOT CROSS-KNOWLEDGE GRAPH NAMED ENTITY EXTRACTION LEVERAGING WIKIFICATION

C.A. Wiefferink (Cas)

MSc Computer Science,
Data Science & Technology

EXAMINATION COMMITTEE:
dr. S. Wang
dr. ir. M. van Keulen
J. Scholten

UNIVERSITY OF TWENTE.

JUNE 1, 2022

Preface

This thesis is the end product of my Computer Science Master, with a specialization in Data Science & Technology, at the University of Twente. It has been a period full of learning, memorable experiences, and exploring new interesting topics. Completing my studies would not have been possible without the support of my supervisors, colleagues, family, and friends.

First of all, I would like to sincerely thank my supervisors Shenghui Wang and Maurice van Keulen for their guidance and valuable feedback throughout the project. Secondly, I would like to thank Little Rocket that gave me the opportunity to work on this project. Thank you to my colleagues, in particular Jelle (2x), for the many discussions leading to new ideas and inspiration. Furthermore, I would like to thank Reiner for helping out with the qualitative evaluation, and Jelle (2x), Jeroen, and Thijs who helped proofreading this thesis. Lastly, a special thanks to Thijs for allowing me to make use of his server which provided the necessary computing power to perform the experiments.

I hope you will enjoy reading this thesis. In case you have any remarks or questions, please feel free to contact me.

June 1, 2022

Cas Wiefferink

Abstract

A key step to bridge the gap between natural language (NL) text and knowledge graphs (KG) such as Wikipedia, is named entity extraction (NEE). In a KG, the nodes represent concepts or named entities, and the edges between the nodes represent semantic relations. NEE is the automated extraction of mentions of named entities appearing in the text and linking them to their corresponding entities in a KG. NEE with a target KG based on Wikipedia is also called wikification. The links contribute to the vision of the semantic web and can help readers to better understand the resource. Furthermore, they aid in creating a semantic representation of the document, and can play a key role in a number of natural language processing (NLP) and information retrieval (IR) tasks.

While wikification models are powerful due to the vast amounts of training data available in Wikipedia that can be used to train an NEE model, they lack domain-specific entities and are computationally expensive. On the other hand, domain-specific KGs lack the more general concepts, and often have no training data available. Leveraging wikification to extract entities from a specific domain that consists of more than one target KG (e.g. a subset of Wikidata combined with a domain-specific KG) is a problem that has not been attempted to solve before. As a case study, we have used the job market as application domain.

To extract entities with two, originally unaligned, target KGs we propose a method that consists out of three main steps: KG Alignment, KG Pruning, and named entity Extraction (APEx). First, we have aligned our domain-specific KG with Wikidata to deal with the overlapping entities. Second, we have employed seed enrichment and a (strongly) local graph clustering (LGC) method, using the set of overlapping entities as seed, to prune entities from Wikidata that are not relevant to the job market domain. Then, we constructed multiple target KGs consisting of (a pruned version of) Wikidata, ESCO, and combinations thereof. Finally, a form of zero-shot learning (ZSL) has been used to leverage a wikification model named Bootleg [1] to perform NEE with the newly constructed target KGs.

We have evaluated the NEE performance in two-fold: quantitatively and qualitatively. We show that APEx outperforms the exact string matching (ESM) with popularity voting baseline, and achieves competitive results compared with Bootleg's original model, in nearly all combinations of target KGs and evaluation strictness levels. The main loss in performance can be attributed to not recognizing entities being mentioned in the text, as opposed to disambiguating the entities. However, the NEE performance is better for IT than non-IT related documents. Finally, we show that APEx can significantly reduce the computation cost in terms of initialization time and memory usage. To summarize, we show the potential of using a wikification model for other applications than merely extracting entities to Wikidata, without the drawbacks of the computation cost of wikification, and without the need for additional training.

Contents

List of Abbreviations

APEx Align Prune Extract

DISCO European Dictionary of Skills and Competencies

EC European Commission

ESCO European Skills, Competences, qualifications and Occupations

ESM Exact String Matching

GDPR General Data Protection Regulation

IAA Inter-Annotator Agreement

IR Information Retrieval

KG Knowledge Graph

KNN K-Nearest Neighbors

LGC Local Graph Clustering

MLP Multi Layer Perceptron

MQI Max-flow Quotient-cut Improvement

MRR Mean Reciprocal Rank

NED Named Entity Disambiguation

NEE Named Entity Extraction

NEL Named Entity Linking

NER Named Entity Recognition

NL Natural Language

NLP Natural Language Processing

PII Personally Identifiable Information

RQ Research Question

TREC Text Retrieval Conference

URI Uniform Resource Identifier

ZSL Zero-Shot Learning

1 Introduction

One of the characteristics of Wikipedia[1] articles is that key terms appearing in the text body have been annotated with links to their respective pages. The links contribute to the vision of the semantic web [2], and help readers to better understand the resource by having direct access to additional relevant information. Furthermore, text documents in general that have been annotated with links add to a computer's ability to create a semantic representation of the document, and play a key role in a number of natural language processing (NLP) and information retrieval (IR) tasks. These include, among others, text categorization, summarization, entailment, and document indexing [3, 4, 5].

A key step to bridge the gap between natural language (NL) text and knowledge graphs (KG) such as Wikipedia, is named entity extraction (NEE). In a KG, the nodes represent concepts or named entities, and the edges between the nodes represent semantic relations [6]. NEE is the automated extraction of mentions of named entities appearing in the text and linking them to their corresponding entities in a KG [7]. NEE with a target KG based on Wikipedia is also called wikification.

When performing NEE we can roughly distinguish between two types of target KGs. First, there are the large-scale, all-encompassing, and crowd-sourced target KGs (e.g. based on Wikipedia). Next to having inferior data quality, these lack highly domain-specific concepts [8], and out of 15 million entity mentions found in web documents it is estimated that 33% cannot be linked to Wikipedia [9]. Wikification has been thoroughly researched [10, P. 14], and has the benefit of having abundant training data available through Wikipedia. Although, as the performance of state-of-the-art models increase, so does the computation cost due to the ever-increasing model complexity, and the millions of possible entities to consider for each mention. Second, there are the small-scale, in-depth target KGs focusing on a single domain, which tend to be curated by a small group of experts. These lack the more general concepts and are updated infrequently. Using these as target KG is challenging, as you are often limited to employing syntactic entity extraction methods, because there usually is no annotated data that can be used for training and testing purposes [7].

Thus, while wikification models are powerful, using them to extract entities related to a specific domain would result in a large percentage of false positives (extracted entities that are not relevant to your domain), as well as a large percentage of false negatives (relevant entities that are not extracted). Yet, using a domain-specific target KG for NEE is challenging, and the KG may not cover the domain of interest entirely, also resulting in a large percentage of false negatives. For this reason, performing NEE with a single target KG can be sub-optimal in some real-world use cases where you are only interested in entities related to a single application domain. In this thesis we introduce a methodology attempting to combine the best of both worlds by leveraging a wikification model

[1] https://wikipedia.org/

for domain-specific NEE that supports multiple target KGs. Performing NEE with multiple target KGs has been done before, but during the literature research no prior work was found that leverages a wikification model for this.

We show that our method achieves competitive NEE performance for a combination of different target KGs: a domain-specific KG, Wikidata (that is pruned to a specific domain), and combinations thereof. This is achieved with a low computation cost, and without the need for data annotation and model training through the use of zero-shot learning (ZSL). Furthermore, we show its potential generalizability to other application domains.

1.1 Case study

As a case study, we have used the job market as application domain, which, especially in the IT-sector, is extremely flexible and constantly evolving. Jobs become obsolete, new jobs and technologies arise, and existing jobs' activities are changing. A relatively new job title and tool such as machine learning engineer and Amazon SageMaker, are present in Wikidata. However they are not present in a KG specifically focused on the job market, namely ESCO [11], which arguably is the most comprehensive one out there. The opposite holds for a job title such as database designer, indicating that both KGs could complement each other.

Performing NEE can form the basis for a variety of applications, of which a few were mentioned already. Performing NEE for entities related to the job market specifically can enable, among others, the following applications [12]:

- Standardization: It can assist in writing job offers, resumes, and curricula by suggesting appropriate concepts based on the input text.
- Job market analysis: It can help analyzing the supply and demand for specific skills, studies, qualifications, and occupations.
- Semantic search: As opposed to keyword-based search, semantic search can also take into account the semantic representation of concepts, and the complex relationships between them.
- Semantic matching: Documents can be matched with one another based on the semantic proximity of concepts they contain. A ranking can be made to list the best candidates for a position or the best suited positions for a candidate. This principle can also be applied to companies internally to find the right person for a project, as well as identifying employees' skill gaps to suggest additional training or different positions.

The usage of AI in recruitment should however be implemented with extreme care to prevent any bias or discrimination. In 2017 there was a case where this was not prevented from happening. Namely, Amazon decided to stop using its AI-based recruitment tool, because it was found to discriminate against female candidates [13]. The model's bias was found to be the result of an under representation of female applicants in the training dataset. The ethics regarding this research will be further discussed in section 7.3.

1.2 Problem statement

Wikification models are powerful, but are not sufficient for each use case, nor is NEE with a domain-specific target KG. The former often lacks specific concepts, while the latter often lacks the more general concepts. Domain-specific KGs also often lack annotated data that can be used to train an NEE model. Leveraging wikification to extract entities related to a specific domain that consists of more than one target KG (e.g. a subset of Wikidata combined with a domain-specific KG) is a problem that has not been attempted to solve before.

NEE in general introduces two main challenges. Next to these, leveraging wikification to extract entities of a single domain with more than one target KG introduces three additional challenges:

(i) Synonymy

(ii) Polysemy

(iii) Dealing with multiple target knowledge graphs

(iv) Preventing the extraction of irrelevant Wikidata entities

(v) Modifying a wikification method to support non-Wikidata entities

To tackle the synonymy problem (i), an NEE system needs to deal with mentions that can appear in NL text in a multitude of forms, such as its full name, partial name, abbreviation, alternate spelling or alias [14]. For example the named entity of "Microsoft Power BI" has its partial name "Power BI", and the named entity "Natural Language Processing" has its abbreviation "NLP".

The polysemy problem (ii) is caused by named entities that have a similar or the same name. NEE thus has to predict to which named entity a mention refers. For instance, the mention "Python" can refer to the programming language or the genus of snakes (or any of the over 20 other entities that have a similar name[2]).

[2] https://en.wikipedia.org/wiki/Python

To deal with multiple target KGs (iii) an alignment between the two KGs is needed such that we can combine them into a single target KG. The alignment can be used to deal with the overlapping entities, but it also allows the wikification method to utilize the information available in Wikidata to extract entities from the domain-specific KG.

To prevent the extraction of irrelevant Wikidata entities (iv) we need to prune our target KG. By pruning irrelevant entities from our target KG before performing NEE we can reduce the polysemy problem as there are less candidate entities for each mention. Furthermore, it will decrease the NEE's model size, which in turn decreases the computation cost.

Lastly, we need to modify a wikification method such that it supports non-Wikidata entities (v). By employing a form of ZSL while using the alignment information we can use a wikification method to extract non-Wikidata entities without the need for additional training.

1.3 Research goals and questions

The goal of this research is to automatically extract all entities that are related to a specific domain from raw text documents. Furthermore, to keep the costs of running to a minimum, the method should be computationally efficient without wasting more resources than necessary. For the real-life use case, the goal is to analyze the method's performance when it has been optimized for the job domain.

To achieve these research goals, the following research question has been defined:

> **How can a wikification model be leveraged for named entity extraction with multiple target knowledge graphs?**

With the following sub questions:

1. How can a domain-specific knowledge graph be aligned with Wikidata?
2. How can Wikidata be pruned using another knowledge graph as seed?
3. How is the performance of named entity extraction of a wikification model affected when an aligned and pruned target knowledge graph is used?

1.4 Contributions

The main contribution of this thesis is providing a method on how wikification can be used for other purposes, including KG alignment, KG pruning, and NEE with multiple target KGs. Other contributions include:

- Adding an alignment with ESCO to Wikidata
- Constructing multiple datasets that can be used to evaluate KG alignment, KG pruning, and NEE methods
- Giving recommendations on how GERBIL [10], which is a benchmarking platform for NEE tools, can be extended to support more detailed evaluation metrics

1.5 Outline

This thesis is structured as follows. First, the theoretical background that is relevant to this research and the related work are discussed in chapter 2. Then, chapter 3 will give an overview of the global methodology, and elaborates on the selected target KGs and wikification method. In the chapters 4, 5, and 6 each of the research questions (RQ) will be discussed (i.e. the KG alignment, the pruning of Wikidata, and the named entity extraction). Chapter 7 will give a more abstract discussion of this thesis, including the generalizability, costs, ethics, and challenges. Finally, chapter 8 concludes the thesis by summarizing our findings, listing a number of limitations, and giving suggestions for future work.

2 Related work

This section explains the most central concepts used in this thesis, including KG alignment, KG pruning, NEE, and NEE with multiple target KGs. Furthermore, we discuss different methods and state-of-the-art relevant for this thesis.

2.1 Knowledge graph alignment

As was mentioned in section 1, we can roughly distinguish between two types of KGs. On the one hand there are the large-scale, all-encompassing ones (e.g. based on Wikipedia), on the other hand there are the small-scale, in-depth ones focusing on a single domain. Large-scale KGs that are based on an encyclopedia such as Wikipedia have a coverage of a wide variety of domains. However, none of these domains is covered with great specificity [8]. For the use-cases where detailed terminology is to be extracted from NL text there often exists a more in-depth KG that can be used for that purpose. These KGs are available for many different domains, including for example the legal, biomedical, and job market domain [11, 14, 15].

To combine both KGs into a single one is where the need for KG alignment arises. This is the process of finding correspondences between entities and/or the structure of two KGs.

2.1.1 Knowledge graph alignment approaches

Multiple approaches can be distinguished for the task of KG alignment. A widely used classification by [16], which is summarized by [17], makes a distinction between schema-level matching and instance-level matching, and a distinction between element-level matching and structure-level matching [18, 19, 20].

Schema-level matching only utilizes the schema information that is defined in the underlying ontology of a KG. The ontology defines the types of entities that can exist in our KG, the different relationships that can exist between two entity types, and the attributes that can be used to describe an entity [21]. Alternatively, instance-based matching uses information on the instance-level, meaning the actual data contents of an entity, which includes for example the label(s) or description of an entity. Both can be subdivided into element-level matching approaches and structure-level matching approaches. Element-level matching considers each entity in a KG independently by matching it to a single entity in the target ontology. Structure-level matching on the other hand, attempts to match combinations of entities to other combinations of entities in the target KG.

These approaches can be subdivided into more precise types of approaches, namely syntactic-based and semantic-based approaches [17, 19]. Syntactic-based approaches use the raw text of labels and descriptions of entities as sequences of characters. This means entities are matched by string similarity, where more similar sequences of characters are more likely to express the same concept. Semantic-based approaches on the other hand apply NLP techniques where semantics (i.e. word understanding) is used to obtain a similarity score. When using semantic-based approaches the similarity score between for example software engineer and developer would be high, in case of syntactic similarity the score would however be low.

2.1.2 State-of-the-art knowledge graph alignment methods

Syntactic-based approaches are simple and efficient. The use of edit-distance as a similarity score was widely adopted by many systems where scores are calculated using for example Jaccard, Levenshtein, longest common substring, exact match, and combinations thereof [22]. Their performance however is limited as they struggle to deal with the polysemy problem, where words can have a different meaning depending on their context.

The first step in the direction of semantic-based approaches was the introduction of WordNet [23], which can be used to determine the semantic similarity between words. WordNet makes use of semantic relationships such as synonyms, antonyms, hyponyms, and more, but its vocabulary coverage was low.

Another major step was the first introduction of word embeddings into the field of KG alignment [24] where the researchers made use of Google's Word2vec [25], which are embeddings that are trained on Wikipedia. They used the cosine similarity between entity labels and descriptions to perform the matching, and managed to outperform all WordNet-based matching approaches.

Next to Word2vec there are other word embedding repositories, including for example Polyglot [26] and Facebook's alternative FastText [27]. The performance of these three word embedding repositories have been compared for the task of KG alignment [28]. They achieved state-of-the-art performance with FastText scoring slightly better than the second best Word2vec.

These state-of-the-art methods however still struggle with the polysemy problem, as they do not take context into account. As a consequence they are not able to differentiate between entities with more than a single meaning, similar to the problem with NEE as was exemplified in section 1.

2.2 Knowledge graph pruning

Large-scale KGs that are based on an encyclopedia such as Wikipedia have a coverage of a wide variety of domains. While in most use use-cases this is beneficial, when trying to extract domain-specific entities using it as a target KG many irrelevant entities would be extracted as well. Either these can be filtered in post-processing with some algorithm, but ideally these entities should be filtered before extraction starts, such that they would not get extracted in the first place.

By pruning the KG beforehand a large portion of the irrelevant entities can be removed. One of the advantages is that this will decrease the NEE's model size, which in turn decreases the computation cost. Since no pruning method is fully accurate some entities that are identified as relevant will also be removed from the KG. In terms of performance the pruned irrelevant entities can increase the NEE model's precision, as there are less ambiguous candidate entities for each mention. On the other hand the recall can decrease due to the pruned relevant entities. This is a trade-off that will require tuning.

The task of pruning a KG is in literature also known as local graph clustering (LGC) [29]. This is different from global graph clustering, which has the goal of assigning clusters to all nodes in the graph. Global methods however can be computationally expensive considering the ever increasing sizes of graphs, even when using relatively fast clustering methods. The scalability problems induced research into LGC. Instead of returning a global clustering of the entire graph, local clustering methods only return a single cluster based on a seed node or a set of seed nodes. A method is considered to be strongly local if its run time depends only on the size of the seed or output set instead of the size of the entire input graph.

The goal of LGC is to return the "best" cluster "nearby" the seed [30]. Here, "best" and "nearby" are intentionally vague, as they can be defined using a variety of methods. One of the early papers to describe a method for LGC, whilst making use of a set of seed nodes, involves a lot of manual labor [31]. They chose to include all nodes on the path from the seed nodes to the root, but still depend on manual inspection to select which subtrees should be added based on the human understanding of the domain. Some later implementations, that are fully automated but still fairly dated, include Max-flow Quotient-cut Improvement (MQI) [32] and FlowImprove [33]. More recent approaches include implementations of k-nearest neighbors (KNN) [34] and SimpleLocal [35].

Lastly, there are some variations on Google's PageRank algorithm, which include the Approximate Personalized PageRank [36], and the more recent L1-regularized PageRank [37]. Google's original PageRank from 1999 [38] is a way to measure the importance of web pages. It can rank web pages (i.e. entities or nodes) of a graph by importance. By counting the number and quality of the relations of a node it can roughly estimate how important the node is. The underlying assumption here is that important nodes have many relations. The algorithm employs random walks, where it will choose one

seed node uniformly at random. It then chooses a neighbor uniformly at random and continues until convergence. To prevent a random walker getting stuck at a node that has no out-going relations an teleportation parameter is introduced. This determines the probability that a random walker "teleports" back to the original seed node.

Each of the mentioned approaches have their own set of strengths and weaknesses in terms of time complexity, resource usage, scalability, and performance. However, evaluating cluster methods in terms of performance is hard. Clustering tends to be application specific, therefore comparing any two methods does not always make sense, since the motivation and intended application areas differ [29]. As a result, there is no single clear method that is considered best, as the performance strongly depends on factors such as the use case and internal structure of the graph. The evaluation metrics used in this thesis will be further elaborated on in section 5.2.

2.3 Named entity extraction

A large portion of the information available on the internet is in the form of NL, which is hard to process for computers. In order to make this interpretable for computers NLP and other methods of information extraction techniques are needed. The automated extraction of mentions of named entities appearing in the text and linking them to their corresponding entities in a KG is called named entity extraction (NEE) [7]. The main challenges of NEE are caused by synonymy and polysemy, as was detailed in section 1.2.

Extracting named entities from NL text can be split into three sub tasks: The first is preprocessing, the next is to *recognize* named entities being mentioned in the text (NER), and the last is to *disambiguate* and *link* to which concept it references (NED / NEL), as depicted in figure 1.

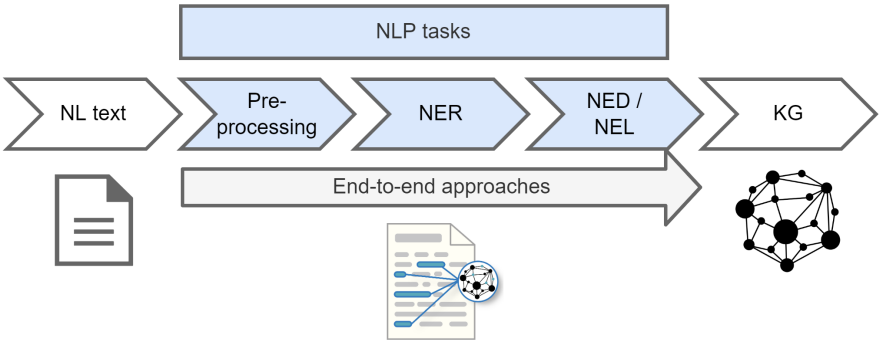

Figure 1: From natural language text to a knowledge graph representation with the use of natural language processing (adapted from [7, Fig. 2]).

2.3.1 Named entity recognition

Different approaches can be taken to deal with the synonymy problem (i.e. recognizing mentions in NL text). These can roughly be subdivided into three categories: rules and pattern, dictionary and KG, and machine learning based approaches [39].

Rules and pattern based approaches

Rule and pattern based approaches identify entities in unstructured text based on predefined rules or known patterns. For example, to extract the amount of experience from a job offer, it is possible to analyze whether phrases like: "# years of experience", "at least # years", or "# years or more" occur in the text.

Rule based techniques have been employed in multiple domains, such as in the medical field [40], where the authors used it to extract medical data from mammography and hospital records of patients. Others tried to extract job titles from Dutch job offers [41] using this method. Results were disappointing, as only 15 percent of the job titles were successfully extracted with a confidence of higher than 80 percent. The main drawback of this method is that it highly depends on the predefined rules and patterns, and is not able to identify new ones without having manual labor involved. Therefore, this method is not suitable for our case.

Dictionary and KG based approaches

Another commonly used method of extracting entities is to match entities with existing dictionaries, taxonomies or KGs. Researchers attempted to recognize mentions of diseases and medications in electronic health records with a dictionary-based approach [42]. To make the final predictions they merged the annotations of three string similarity methods, including exact matching, fuzzy matching and matching based on stemmed words. They achieved a recall of 57 percent with an F1-score of 60 percent. In [15] the researchers used a similar approach, and achieved a recall of 61 to 67 percent.

The first job domain taxonomies were composed in the 1990s, such as Occupational Information Network (O*NET)[3] from the US, or the multilingual European alternative European Dictionary of Skills and Competencies (DISCO)[4]. Next to these there are multiple proprietary ones out there that have been developed by companies, such as LinkedIn's skills taxonomy [43] or Janzz' ontology[5].

[3] https://www.onetcenter.org
[4] http://disco-tools.eu
[5] https://janzz.technology/janzz-on

Arguably, one KG curated by the European Commission (EC) is the most comprehensive that is publicly available as of writing. In 2010 they started developing a classification system named European Skills, Competences, Qualifications and Occupations (ESCO) [11]. It is based on DISCO, and consists out of four main entity types: skills, knowledge, occupations, and qualifications. The first three entity types are structured hierarchically and are interconnected with each other. For example, the skill build recommender systems is considered an essential skill for the occupation data scientist.

ESCO however, depends on being updated by a small group of domain experts, lack the more general concepts, and is updated infrequently. An alternative is to utilize a crowd-sourced KG such as Wikidata[6] or DBpedia[7], which both offer structured data based on Wikipedia to ease its exploitation as linked data [44]. These can also support the NED process, which will be further discussed in section 2.3.2.

Two advantages of using a dictionary or KG approach is that the annotation of training data may not be needed as it could rely on solely lexical resources. Furthermore, it can aid in the candidate generation step during NED, which is discussed in section 2.3.2. The main disadvantage however, is that constructing and maintaining these resources is costly [7]. The importance of maintenance should not be overlooked, since the job market is fast-moving where constantly new technologies are developed and thus new skills arise. Utilizing a crowd-sourced KG can take away these disadvantages. One disadvantage, especially when depending on crowd-sourced KGs however, would be the introduction of a percentage of incorrect labels and relations that are present in the KGs.

Neural network based approaches

The last, and most recent, are feature-inferring neural network based approaches. One of these language representation models is called BERT, which stands for Bidirectional Encoder Representations from Transformers [45]. BERT was introduced and open-sourced in late 2018 and has since been beating multiple records of various NLP tasks, including sentiment analysis, classification, sentence similarity, as well as NER.

Typical NER models are limited to recognizing named entities that fall in the categories of people, organizations, locations and miscellaneous names. A common dataset that is used to evaluate the performance of the NER task for these categories is CoNLL [46]. During testing BERT manages to achieve an F1-score of 92.8%

In [47] researchers have added a classification layer on top of a pre-trained Finnish BERT model to recognize skills in Finnish job offers. The evaluation was done in two steps, once for the recognition of skills consisting of phrases and another time for skills con-

[6] https://www.wikidata.org
[7] https://wiki.dbpedia.org

sisting of individual words. The recognition of skill phrases is considered harder, since finding the edges of skill phrases is cumbersome and can often be ambiguous during data annotation. For the recognition of skill phrases they managed to achieve an F1-score of about 72 percent. In the case of skills consisting of individual words the F1-score increased to 83 percent. The researchers used a dataset of over 200 thousand job offers for training and testing purposes.

In contrast to the previous mentioned NER approaches, for this approach no manually crafted rules or KGs are needed. However, a large dataset is required to train a robust model.

2.3.2 Named entity disambiguation

The next step in the NEE process is disambiguating the recognized mentions in the text. NED attempts to tackle the polysemy problem by determining which entity a mention is referring to. In literature this is also referred to as named entity linking (NEL), meaning that each mention is linked to an uniform resource identifier (URI). Many studies do not differentiate between disambiguating and linking as it often one and the same task.

The process of linking a mention to an entry in a KG includes three main steps: candidate entity generation, candidate entity ranking, and unlinkable mention prediction [7]. First, in candidate entity generation, all entities a particular mention can refer to should be retrieved from a KG. In case the NER was dictionary or KG based, this step has already been done. Second, the candidate entities should be ranked according to how likely it is referred to by the mention that appears in the text. Only the most likely entity is returned. Finally, it can occur that the entity corresponding to the mention in text is not present in the KG. These kind of mentions should then be denoted as NIL, which means the mention is "unlinkable" [14].

2.3.3 End-to-end approaches

In the previous sections the NEE process has been discussed as multiple distinct tasks consisting out of three separate tasks: pre-processing, NER and NED. There are however also solutions that take an end-to-end approach, as depicted in figure 1, meaning all tasks are integrated into a single pipeline. In [48] they argue that an end-to-end approach can enable a system to capture the dependency between the NER and NED tasks, and reduce the propagation of errors from one to the other task.

Researchers in [49] have compared three commercially available NEE solutions from

Google[8], Microsoft[9] and Amazon[10] with three state-of-the-art systems: (i) Bootleg [1], a self-supervised system for NEE, (ii) REL [50], a system combining existing state-of-the-art methods and (iii) WAT [51], an extension to an older entity linker method called TAGME [52]. They have compared the systems on their ability to correctly link mentions to their respective Wikipedia entry. The goal was not to only link mentions from the typical NER categories (i.e. persons, organizations, locations and miscellaneous names), but to link any mention that can be associated with a Wikipedia page. The evaluation was done for the English language across different entity popularities ranging from head, torso, tail to toe. In this context head examples involve the extraction of popular entities that occur frequently in Wikipedia, while toe entities are almost never mentioned.

They found that Bootleg performs best overall, while Microsoft is the best-performing commercial system. On average Bootleg achieves a recall of 78.7 percent, outscoring the next best system (Microsoft) by 12 percentage points, while Microsoft in turn scores more than 16 percentage points better than the other commercial systems. When solely taking head entities into account the differences are smaller, as Bootleg achieves a recall of 85.1 percent, outscoring the next best system (REL) by only 2 percentage points. Taking only toe entities into account however, the differences in performance are large. Bootleg again outscoring the other systems with a recall of 66.4 percent, almost double or even more than other systems. Only Microsoft keeps up somewhat, scoring 15 percentage points lower, while systems from Amazon, Google and REL drop below 25 percent.

Thus next to being open-source, Bootleg also manages to outscore alternative solutions. For this reason, we have opted to make use of Bootleg in this thesis.

2.4 Named entity extraction with multiple target knowledge graphs

We have now seen various NER, NED, and end-to-end NEE approaches, however none of them support entity extraction for a specific domain that consists out of multiple target KGs. Three closely related papers were found that are arguably the closest related to the problem we aim to solve.

In [53], which is one of the early papers on NED with multiple target KGs, propose an NED system that can disambiguate mentions to any KG that is provided in the Boyce-Codd normal form [54]. They extract domain-independent features by performing supervised training with one domain-specific target KG, and use domain adaptation enabling NED with another domain-specific target KG along with Wikipedia. When performing training

[8] https://cloud.google.com/natural-language
[9] https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics
[10] https://aws.amazon.com/comprehend

using a target KG on sports and testing on a target KG on movies, and vice versa, they achieve an accuracy of 0.73 and 0.66, respectively. No values were given for the recall. It should be noted that they do not take the NER phase into account, as they assume the mentions have been recognized already. This can give a biased comparison when comparing to an end-to-end NEE system that does include NER. Furthermore, the method relies on supervised learning, albeit using annotations for a different target KG.

In [55] researchers attempted to perform NEE with multiple target KGs simultaneously. They used two KGs in the biomedical domain that first were aligned with indirect supervision by making use of entity attributes that contain references to Wikipedia pages. Two entities referencing the same page were presumed to be the same concept. They found that combining entity information from both KGs resulted in an improved performance over using an individual target KG. A limitation of this method however is for it to function the researchers depend on two target KGs that have been manually aligned on beforehand. They do not provide a method to automatically find an alignment between two KGs, nor do they show generalizability to other domains.

In [56] a different strategy is employed. Before performing NEE on a document they first classify which KG from Fandom[11] (a platform with community-developed KGs for multimedia domains such as movies, TV, and games) is most closely related, and is subsequently chosen as main target KG. The Fandom KG is aligned with Wikipedia on the fly with indirect supervision by making use of the Fandom's article texts. These are structured similarly to Wikipedia's articles, as the mentions of other entities also contain hyperlinks to both the corresponding Wikipedia and Fandom entities. For the NEE a recall of 0.53 and 0.67 was achieved for links to Wikipedia and Fandom, respectively. Although this method does not require the target KGs to be pre-aligned, it does have a similar limitation as the one described above. It namely depends on the supervised nature for it to produce the alignment between Wikipedia and the chosen Fandom KG.

To summarize, we have now seen different approaches for NED/NEE with multiple target KGs. However, all have limitations in one way or another. Either they only focus on NED or they depend on the supervised nature to align two KGs. Especially the latter is challenging, as an annotated dataset for a domain specific KG is usually unavailable or expensive to come by. This is where wikification can play an important role, which will be discussed in the next chapters.

[11] https://www.fandom.com

3 Global methodology

Figure 2 gives a high level overview of the method that we propose to extract relevant entities to two, originally unaligned knowledge graphs. The method consists out of three main steps: KG Alignment, KG Pruning, and named entity Extraction (APEx). The first two steps are used to construct the final target KG, which is then used for NEE in the third and last step. Since we are dealing with multiple target KGs we first need to find which of the entities in either KGs refer to the same concept. By first aligning the KGs we can deal with overlapping entities in the final target KG. This will be elaborated on in section 4. A large portion of the entities in Wikidata are not relevant to our use case. To prevent textual mentions of irrelevant entities to be extracted during NEE these entities first need to be pruned from the target KG. This is explained in section 5. Finally, the constructed target KG has been used for NEE, which will be discussed in section 6.

Before getting into the three main steps of this research, first the two knowledge graphs, Wikidata and ESCO, will be described in more detail in section 3.1. Then section 3.2 will elaborate on Bootleg that has been used throughout this thesis.

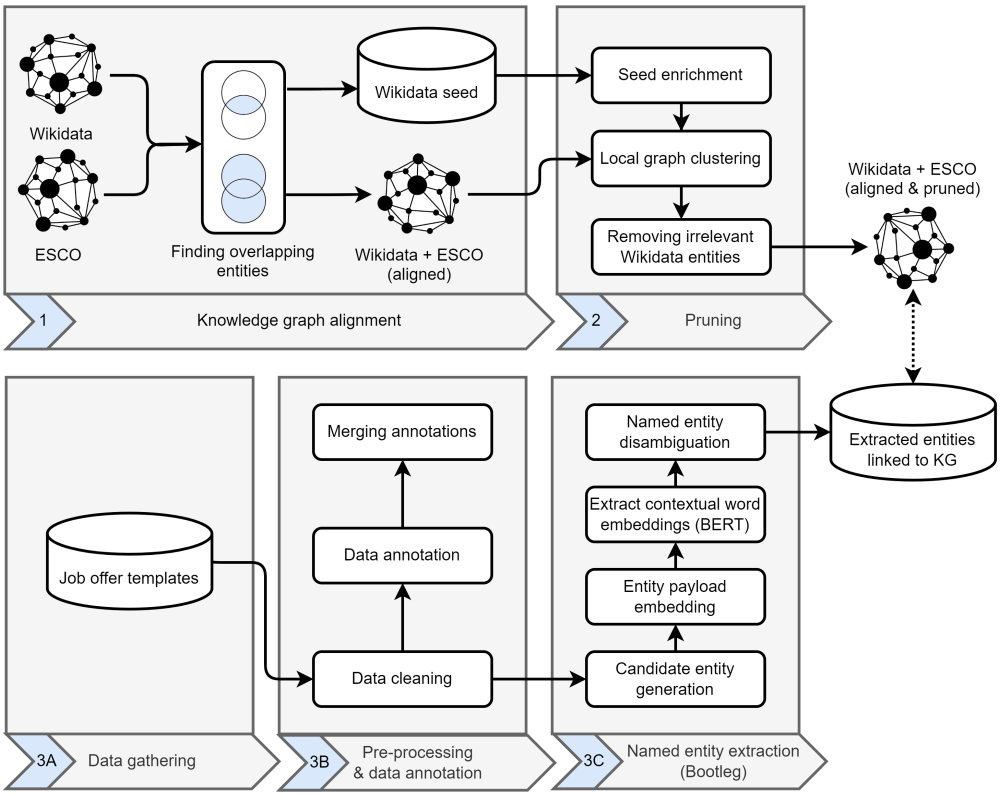

Figure 2: A high level overview of the proposed solution method (APEx)

3.1 Knowledge graph selection

To construct our target KG that encompasses the job domain it has been chosen to make use of both Wikidata and ESCO. When either of KGs is used independently, neither of them fully satisfy the requirements for our use case. Using both combines the best from both worlds. On the one hand large, general and crowd-sourced (Wikidata), and on the other hand smaller, fine-grained, and curated by experts (ESCO). Wikidata and ESCO differ vastly on various aspects, such as in terms of size, how they are structured, and their entity's data contents, including the labels and descriptions. This is summarized in table 1. In this research a Wikidata dump from October 2020 has been used, and ESCO version 1.0.8 which was released in August 2020.

Table 1: Differences and similarities between how Wikidata and ESCO are structured, their sizes, and how entities are labeled and described (adapted from [17])

	Wikidata	ESCO
Relations	Parent-child, sibling, properties, associations	Parent-child, sibling, associations
Organization	Hierarchy	Hierarchy
Language	Multilingual	Multilingual
Labels	Preferred and alternative	Preferred and alternative
Ontology language	OWL/RDF [57]	SKOS [58]
Writing style	Upper- / lowercase, singular, nouns only	Lowercase, singular, nouns/sentences
Granularity	Large, less specific	Small, detailed
Entity types	23.4k	1,002
Total entities	5.83M	17.4k
Total relationships	39.3M	259k

ESCO is structured in four main entity types: occupations, skills, knowledge, and qualifications (actually knowledge is a sub type of skills, but we distinguish it as a main type. This is further explained in section 4.2). The latter entity type is not useful for our use case as it contains definitions and descriptions of various educational titles from institutes across Europe. The qualifications are far from comprehensive and the entities do not have any relations with entities from one of the three main entity types. It will therefore be ignored in the rest of this research. The three main entity types are structured hierarchically, meaning the KG has a tree-like structure where the entities are subdivided into categories up to a depth of 6 layers. These layers consist of 1,002 distinct entity types (i.e. non-leaf/parent entities: entities that have at least 1 or more children entities) of which 619 are occupations, and 383 are skills/knowledge. The number of (leaf/child) entities per main entity type is not equally distributed, as it consists of 2,942 occupations, 10,583 skills, and 2,902 entities of the type knowledge. Furthermore, there are relations between entities from different entity types. For example, the skill build

recommender systems is considered an essential skill for the occupation data scientist, and the knowledge data extraction, transformation and loading tools is considered essential knowledge for the occupation database developer.

Wikidata on the other hand comprises of almost 6 million entities. These are structured hierarchically as well, and distinguishes over 23 thousand entity types up to a depth of 12 layers. When comparing the average number of relationships per entity for either KGs it can be seen that ESCO is over twice as densely connected. Entities in Wikidata and ESCO have approximately 7 and 15 relations on average, respectively.

The writing style of the entity's data contents differs between Wikidata and ESCO. An example of two entities from either KGs that refer to the same concept can be seen in the tables 2 and 3. Labels of entities in Wikidata are written using one ore more nouns, whereas in ESCO they are written as a noun or short sentence depending on the entity type. The entities of type knowledge and occupations are written as one or multiple nouns, but the entities of the type skills are often written as a verb combined with a noun. Here the noun is often some concept or activity, as in the example above. This is expected to have a significant impact on the alignment performance when a simple syntactic measure is used. Furthermore, there are differences in the usage of (non-) capitalized letters and singular/plural forms. Lastly, the entity's data contents such as the alternative labels in ESCO are present for over 96 percent of the entities. For Wikidata this is only true for about 71 percent of the entities.

Table 2: Example of an entity in Wikidata

Preferred label	data analyst
Alternative labels	data analist
Entity type(s)	profession, analyst, data scientist
Description	profession that finds trends and patterns in data used to make decisions

Table 3: Example of an entity in ESCO

Preferred label	data analyst
Alternative labels	data warehousing analyst, data analysts, data warehouse analyst, data storage analyst
Entity type(s)	Systems analysts
Description	Data analysts import, inspect, clean, transform, validate, model, or interpret collections of data with regard to the business goals of the company. They ensure that the data sources and repositories provide consistent and reliable data. Data analysts use different algorithms and IT tools as demanded by the situation and the current data. They might prepare reports in the form of visualisations such as graphs, charts, and dashboards.

3.2 Named entity extraction with Bootleg

Bootleg [1] is a self-supervised NEE model with Wikidata as target KG, and was designed to perform well on both popular and tail entities by leveraging textual and structural information. The textual information is sourced from Wikipedia articles. These also provide the ground truth used for self-supervision, since the article texts contain hyperlinks to other articles, where the anchor texts can be used as mentions for the entity that is being linked to. The number of times a mention (i.e. an anchor text) in the Wikipedia articles is used to refer to some entity, is stored as a weight. This is used to indicate the importance of this mention for a particular entity. The structural information is sourced from Wikidata which includes knowledge graph relations and type information.

Figure 3 (right) shows the data flow of Bootleg. An entity database is queried to find all mentions in the sentence. For each mention it generates entity candidates using exact string matching (ESM), where larger n-gram mentions are favored. Next to this, it retrieves each candidate's entity profile, which consists out of it's type, relationships, possible mentions, and other features (this is further discussed in section 6.1). After embedding this as an entity payload, and encoding the sentence into contextual word embeddings using BERT, both are input to Bootleg's model.

Entity payload embedding

A three-layered hierarchy of signals, see figure 3 (left), is used to embed the entity payload. At the base of the hierarchy are entity patterns, which are the most discriminative and least general. These patterns were learned by BERT using textual co-occurrences and the relative and absolute positions of words (e.g. the entity "Harry Styles" is associated with the phrase "Dunkirk").

Knowledge graph relations form the middle layer of the hierarchy, where the relationships between entity candidates in a sentence serve as cues for disambiguation. Using the example from figure 3, *"What roles does Harry play in Dunkirk and Dolittle?"*, there are two actors named "Harry" that act in Dunkirk, but Harry Collet is the only one with a KG relationship "acts in" with the film Dolittle.

The top layer consists of type patterns, which are the most general, and uses linguistic or semantic cues that may indicate a mention must be of a certain type. For example, in *"What role does Harry play in Dunkirk?"*, the word "play" suggests that "Dunkirk" refers to the movie, rather than the WWII evacuation.

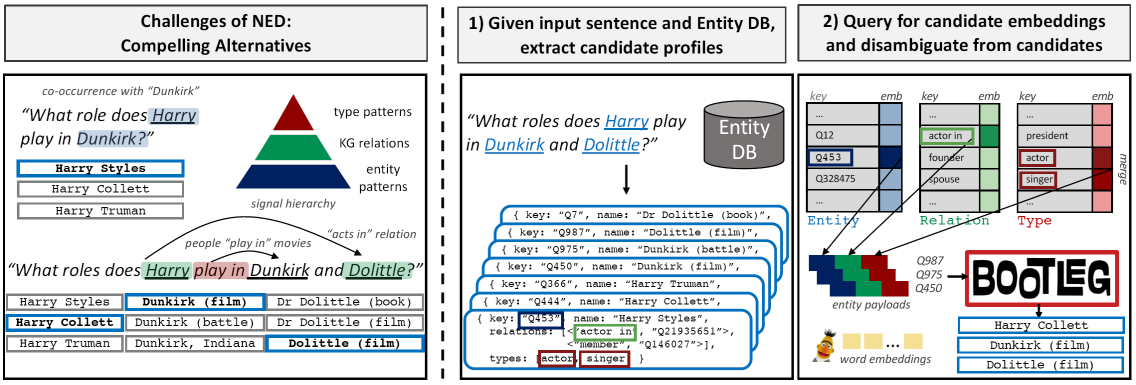

Figure 3: (left) shows the three reasoning patterns used for disambiguation and (right) shows Bootleg's data flow from an input sentence to disambiguated entities as output [1].

Architecture

Subsequently, the embedded signals and contextual word embeddings are input to Bootleg's model, which comprises of two stacked, standard transformer modules [59]. One is for phrase memorization, and the other is for co-occurrence memorization, as depicted figure 4. This way dependencies can be learned between related phrases as well as between embeddings, for each level in the hierarchy. The most likely candidate for each mention is then selected by adding the outputs of both modules, and scoring the result using a multi-layer perceptron (MLP) softmax layer for each candidate entity.

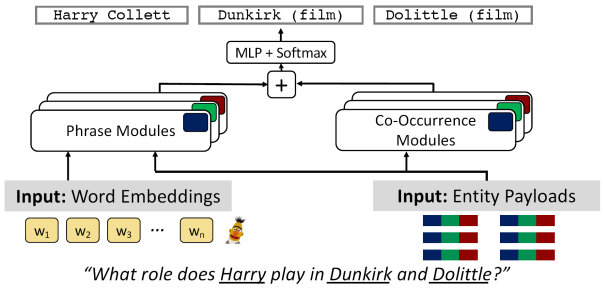

Figure 4: Bootleg's architecture [1].

Pre-trained model

A pre-trained Bootleg model is available that was trained on the full English Wikipedia dump from October 2020 consisting of over 50 million sentences. Some heuristic filter-

ing has been applied to filter out noisy mentions and entities such that terms like "of", "from", and "also" are excluded. Furthermore, if mentions of entities in Wikipedia sentences are labeled less than 3 percent of the time, then they are filtered out as well. It reduces the number of false positives significantly, although it does mean that a percentage of relevant entities are filtered out as well. This is a limitation of using a pre-trained model, and is further discussed in section 8.1.

3.3 Summary

In this section we have given a high level overview of the solution method consisting out of three main steps. Furthermore, we have seen the similarities and differences between Wikidata and ESCO, and elaborated on Bootleg that has been used throughout this thesis. In the following sections each of the three steps, KG alignment, KG pruning, and NEE, will be discussed in detail.

4 RQ 1: Knowledge graph alignment

ESCO first needs to be aligned with Wikidata before both can be used as target KG. The alignment between the two KGs can then be used to deal with the overlapping entities. This section aims at providing a method to answer the first research question: *How can a domain-specific knowledge graph be aligned with Wikidata?* This is the first step in the processing pipeline (see step 1 in figure 2). The output of this step will be used downstream in the next research questions. Firstly, the alignment between the KGs is used to prune Wikidata such that mostly entities related to the job domain remain. Secondly, the alignment is used to interconnect both KGs such that this information can be used during NEE.

4.1 Method

In section 2.3.3 six different NEE solutions have been compared. It was found that Bootleg, next to being open-source, outperformed the alternative solutions. Even though Bootleg is originally intended to be used for NEE by linking the mentions in raw text to the respective entity in Wikidata (wikification), we have employed it to align ESCO with Wikidata.

Since Bootleg cannot deal with relationship types and structure information, an instance-level matching has been used rather than a schema-level approach. This means the entity's data contents has been utilized while ignoring any relationship types and structure information. A mostly element-level matching approach has been taken, where only the entity itself and up to three ancestors have been used to find a matching equivalent in Wikidata. It is likely that using more structure-level information (i.e. taking into account other connected entities such as children and siblings) would further increase the performance. This is however out of scope of this research.

As was touched upon in section 3.2, Bootleg largely depends on a syntactic method to generate entity candidates for a mention. However, to decide which of the candidates to predict, and calculating the confidence score (i.e. the NED phase) is a semantically based method. One could therefore conclude that Bootleg is a hybrid form, making both use of syntactic and semantic properties in its approach.

Bootleg's architecture uses a hierarchy of different signals to embed an entity payload. Among these are the textual co-occurrences of, and KG relations between, entity candidates. Because of this, it is key that the entity's data contents of a single entity are inputted in a single batch. The parts of the entity's data contents that have been used are, in order of decreasing importance:

1. *Preferred label*
2. *Alternative label(s)*
3. *Preferred labels of up to three ancestor entities*
4. *Description*

All these parts have been combined into a single piece of running text (similar to the entity's data contents shown in table 3). In case each part of the entity's data contents are inputted separately, the performance decreases as Bootleg can not leverage the sentence context. To illustrate, in the case the words "Java" and "software" appear in a single sentence, this information can be used to disambiguate "Java" to Java (programming language) as opposed to Java (island of Indonesia).

For each entity up to four alignment predictions have been made. The order of the predictions fully depends on the order of the entity's data content parts (i.e. a prediction for the preferred label weighs heavier than one for an ancestor).

We can also extract the confidence level for a single NEE prediction. This could be used to weigh a prediction for an alternative label heavier than a preferred label. For example, when the confidence levels for the NEE predictions for the preferred label and an alternative label are 0.67 and 0.93, respectively. Then it may make sense to weigh the NEE prediction for the alternative label heavier than the one for the preferred label due to its increased confidence level. Different thresholds and weights have been tested, but none gave an improvement over the method described above, and thus has been left out of the results. As a result, we in fact can find an alignment without needing to do additional training.

Baseline

As a baseline Jaccard similarity with word-based tokenization and popularity voting has been used, which is a relatively common baseline method in the field of KG alignment [22, 28]. Calculating the Jaccard similarity between two sets of tokens is done by taking the number of common tokens and dividing it by the total number of unique tokens. It is mathematically expressed by:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where, the numerator is the intersection and denominator is the union. Before calculating the Jaccard similarity between two strings, similar pre-processing steps are taken

as in Bootleg's method, which was described in section 3.2. This includes normalizing the strings to UTF-8 characters, eliminating diacritics, lower-casing, and removing ASCII characters that are not alpha-numeric. Finally, the strings are tokenized by splitting them using space as the delimiter.

In section 2.1.1 various matching approaches have been discussed and categorized. Jaccard similarity depends on string matching, and thus is considered a syntactic matching approach. Furthermore, we only use the data contents of entities, as opposed to using schema-level information, therefore this method falls in the category of instance-level matching. In terms of textual input to this method, there are two differences compared to the input of Bootleg that are worth mentioning.

First, rather than combining all parts of the entity's data contents into a single piece of text, here each part has been inputted separately. Reason being that the Jaccard similarity score strongly depends on tokenized inputs that are of comparable lengths. For each part of the ESCO entity's data contents the Jaccard similarity has been calculated with the labels of all the entities in Wikidata. The one with the highest similarity is returned for the preferred label, and for each of the alternative labels and up to three ancestors. Thus, again mostly element level matching has been used, as opposed to structure level matching. In case multiple candidate entities are found with equal Jaccard similarity scores for a given ESCO entity, then the most popular entity is selected (i.e. the one with the largest weight, as discussed in section 3.2). Secondly, the description was not used here since the Jaccard method would find too many false positives when splitting the description into n-grams.

4.2 Evaluation

The alignment of two KGs can be approached as an IR problem. In other words, for each entity from ESCO we can return a ranked list of entities from Wikidata ordered by confidence level of the predictions. Conferences such as the Text Retrieval Conference (TREC)[12] give us recommendations on the relevant rank-based measures to evaluate this task. Three different measures have been chosen to evaluate our predicted alignment:

- Mean reciprocal rank (MRR)
- Precision at k ($P@k$)
- Coverage

[12] https://trec.nist.gov/

The MRR is commonly used to evaluate methods that return a ranked list of predictions to a query. The MRR is the average of the reciprocal ranks of predictions for multiple queries Q:

$$\text{Mean reciprocal rank } (MRR) = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

where $rank_i$ refers to the rank position of the first relevant entity for the i^{th} query. If none of the predictions to a query are relevant, then the reciprocal rank is 0. Only the rank of the first relevant prediction is considered, any further relevant predictions are not taken into account.

The second measure we use is the precision at k ($P@k$). Here, k is the number of predictions that are returned, and $P@k$ is the percentage of those predictions that are relevant, independent of their rank. This has been done for a k in the range of one to four.

The final measure is the coverage, which is similar in meaning to recall. Essentially it is the percentage of ESCO entities for which at least one relevant entity in Wikidata was found.

Determining whether an alignment is relevant is not straightforward, as the two entities may be closely related, but not have the exact same meaning. Only distinguishing between exact or no match would therefore give a limited understanding on the alignment performance. Therefore it is chosen to consider four matching types in which an Wikidata and ESCO entity are a match (the same definitions as in [17] are used):

1. *Exact match:* The entities have exactly the same definition
2. *Close match:* The entities are similar in definition
3. *More general match*: The Wikidata entity is a supercategory of the ESCO entity
4. *More specific match*: The Wikidata entity is a subcategory of the ESCO entity

Making this distinction is needed as the number of entities that have an exact match between the two KGs is relatively low, which is further discussed in the section below. This is partly due to ESCO often describing activities that involve an object or concept, Wikidata however only describes the object/concept itself. For example, ESCO has an entity construct dams, which was linked to the entity dam in Wikidata, and develop business plans being linked to business plan in Wikidata. It has been chosen to label these examples as close matches, as they are similar to each other but do not match exactly.

Other examples that were labeled as close matches include meteorology technician from ESCO being linked to meteorologist in Wikidata, and feed concrete mixer being linked to concrete mixer.

Dataset

To evaluate the performance of the KG alignment a dataset of gold annotations is needed (i.e. a set of entities in ESCO that is mapped with the entity in Wikidata with which it exactly or closely matches). To construct the evaluation dataset at random in total 200 ESCO entities were sampled for which alignment predictions were made. For each entity the top 4 predictions of both Bootleg and Jaccard were annotated without knowing which method made the prediction. The entities consist out of 100, 78, and 22 of the ESCO types occupations, skills, and knowledge, respectively. As was mentioned in section 3.1, ESCO truly only distinguishes between two main entity types: occupations and skills. Knowledge, along with language skills, transversal skills, and 'normal' skills, are sub types of skills. During evaluation it was found that the difference in performance between knowledge and the other sub types of skills was significant. For this reason knowledge has been treated as its own main entity type, which explains the odd sample size ratio. Annotating additional data was considered out of scope, as annotating is a labor intensive task, while the current evaluation dataset should still give a satisfactory portrayal of the alignment performance.

Table 4 summarizes the complete set of annotations that was constructed. 7 out of the 78 sampled skills in ESCO were found to have an exact match in Wikidata. Accordingly, based on this sample we can estimate that about 9 percent of the skills in ESCO have an exact match in Wikidata. This 9 percent would then also be the theoretical maximum coverage (or recall). Looking at exact or close matches, this figure increases to 67.9 percent.

Table 4: The complete set of annotations used for testing purposes of our alignment method

ESCO Entity type	Sample size		Exact match		Exact or close match	
Occupations	100	50.0%	24	24.0%	92	92.0%
Skills	78	39.0%	7	9.0%	53	67.9%
Knowledge	22	11.0%	8	36.4%	18	81.8%
Total	**200**	**100%**	**39**	**19.5%**	**163**	**81.5%**

4.3 Results and discussion

Figure 5 compares the performance between Bootleg and Jaccard for the KG alignment of Wikidata and ESCO. On the x-axis are the matching types that are being counted towards a correct prediction. In this case only the close matches are included. These are counted cumulatively, thus in the case of close matches also the exact matches are counted as correct. This is compared for a range of one to four predictions that are returned per entity to be aligned. Furthermore, three different metrics are shown: the coverage (blue), mean reciprocal rank (orange), and precision at k (red), where k is the number of predictions per entity. The performance of the baseline (Jaccard) is shown as a black line on top of this.

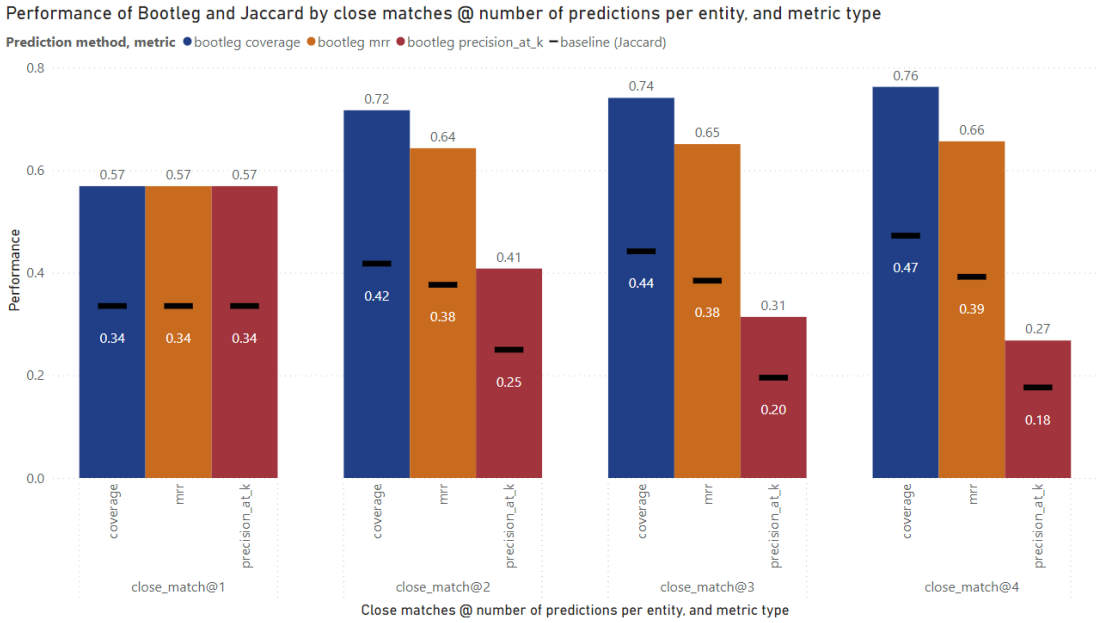

Figure 5: Comparing the KG alignment performance between Bootleg and Jaccard in the range of one to four prediction(s) per entity

The performance when only a single prediction is made per entity is equal for each of the metrics, which is to be expected as these metrics are formally the same with a single prediction. As the number of predictions (k) increases, the coverage improves. This means as we progress down the ranked list of predictions there are still predictions that are relevant. However, as k is further increased there is a diminishing return, where it levels at around 0.76. Whereas the coverage improves as k is increased, the precision at k deteriorates. This means that as we progress down the ranked list of predictions, the chance that the predicted entity is relevant declines. This can be explained by the entity's data contents being inputted as a single batch in order of decreasing importance, as was described in section 4.1. A similar pattern is present for the Jaccard performance, but it

is consistently outperformed by Bootleg.

Figure 6 is similar to the one above. Instead, the results are shown for two predictions per entity, and compares how the performance differs as more matching types are counted towards correct. As more matching types are counted towards a correct prediction, each of the metrics increases. The increase is most significant when the close matching type is added. The increase when adding the matching type more general is overall larger than when specific types are added. Considering that Wikidata mostly contains more general entities than the ones in ESCO, which was discussed in section 3.1, this makes sense. Furthermore, Jaccard is outperformed by Bootleg in all scenarios.

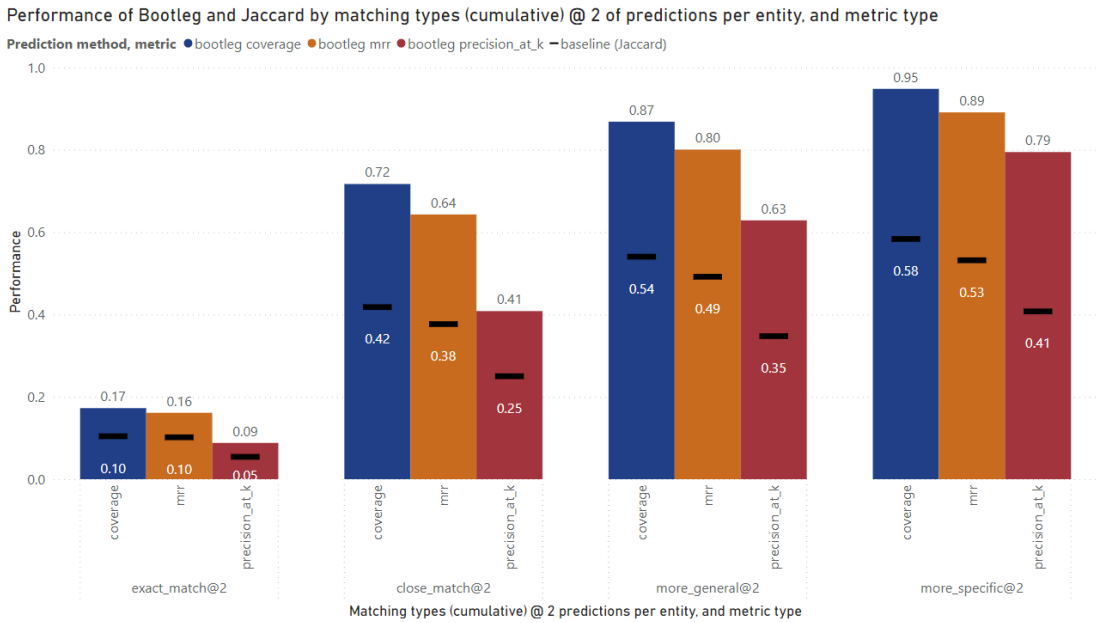

Figure 6: Comparing the KG alignment performance between Bootleg and Jaccard as more matching types are counted towards a correct prediction, ranging from exact matches to more specific matches

Figure 7 is similar to the previous two. The results are shown for two predictions per entity, and only exact and close matches are counted as correct. Instead, the distinction is made between each of the three main ESCO types: occupations, skills, and knowledge. Bootleg's performance for the occupations and knowledge seem to be roughly similar, where both are scoring better than skills. This is true when looking at the percentage of ESCO entities for which at least one relevant alignment is found (coverage), but it does not take into account the theoretical maximum for each ESCO type as listed in table 4. Considering this, the coverage of the entities of which they are presumed to have a correct alignment is 84.8%, 88.3%, and 94.1% for occupations, skills, and knowledge, respectively. The weighted average is 88.7%. In either case, the performance for the skills is being outscored by that of knowledge. This may be due to the difference in

writing style (see discussion in section 3.1) where the labels of skills tend to be written as a combination of a verb and noun(s), while knowledge is written as one or multiple noun(s). However, it does not explain the worse performance for occupations. One possible explanation contributing to this may be some of the highly specific occupations that are present in ESCO, and as a consequence have lengthy labels, such as sporting and outdoor accessories shop manager or numerical tool and process control programmer. Wikidata does not contain many entities with this level of specificity, thus Bootleg will link them to more general ones (e.g. shop manager in the case of the former).

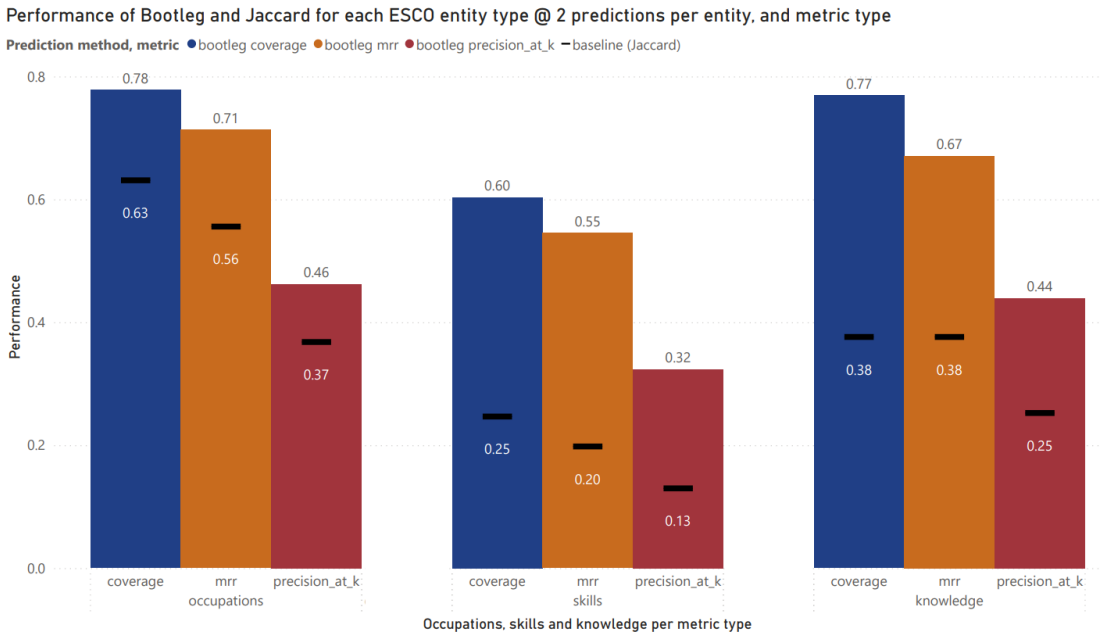

Figure 7: Comparing the KG alignment performance between Bootleg and Jaccard for each of the three ESCO types: occupations, skills and knowledge

When taking a look at the individual incorrect alignments made by both methods, there exists a clear distinction between the type of errors they make. The main difference is that when using Jaccard, ESCO entities are frequently linked to Wikidata entities that are unrelated to the job domain. The most occurring Wikidata entity types are names of things, such as albums, films, literary works, and businesses. Examples of errors include lead teambuilding efforts being linked to lead (element), operate pasteurisation processes being linked to Operate (song performed by Peaches), and bolt engine parts being linked to Naming of Parts (poem by Henry Reed). Compared with Bootleg's incorrect alignments, the Wikidata entity is often related to the ESCO entity in one way or another. The most occurring Wikidata entity types are professions, academic disciplines, activities, and academic majors. Examples of errors include assessment of risks and threats being linked to skill assessment and data quality assessment being linked to quality assurance. Thus, in the cases where the alignment predictions are found to

be incorrect, Bootleg's predictions seem more relevant. It shows that Bootleg can better deal with the polysemy problem than the Jaccard baseline.

4.4 Summary

To summarize, we have shown how wikification can be leveraged to find overlapping entities between a domain-specific KG and Wikidata. This has been done by taking an alignment approach consisting of a combination of element-level, instance-level, and a hybrid of syntactic/semantic matching. We have evaluated Bootleg's alignment performance, and found that it outperforms the Jaccard baseline in each of the used evaluation metrics.

5 RQ 2: Knowledge graph pruning

The output of the previous step is an alignment between Wikidata and ESCO. Or in other words, a set of entities in Wikidata which is predicted to have some relation to one or multiple entities in ESCO. At the moment however, Wikidata still mostly contains entities that are not relevant to our application domain. This section aims at providing a method to answer the second research question: *How can Wikidata be pruned using another knowledge graph as seed?* This is the second step in the processing pipeline (see step 2 in figure 2), which includes removing entities from Wikidata that are irrelevant to the job market domain.

The definition of irrelevant to the job market domain in this context would be the entities that do not appear in the same "local cluster" as the entities that have some relation with ESCO. Here local cluster essentially means a smaller sub graph within the larger Wikidata graph. Thus the entities in Wikidata that are closely related to the job market but do not appear in the alignment with ESCO are retained as well to further increase the coverage of the job market domain. The constructed KG then consists out of the entities in ESCO combined with the cluster of relevant entities that is found in Wikidata. The output of this step will be used in the next research question, where it will be used as target KG for named entity extraction.

5.1 Method

In order to find the so-called local cluster of relevant entities, a distinction is made between three main steps. These are depicted in figure 8. The first step (left) is the KG alignment where a set of Wikidata entities has been found that were predicted to be overlapping with ESCO. This has already been discussed in section 4, and will not be discussed any further here. In the second step (middle) this set of Wikidata entities, the seed, has been used to find a local cluster by adding the most occurring entity types, and by using the appropriately named method local graph clustering. In the final step (right) the entities have been pruned that did not appear in the local cluster from the graph, and thus leaving us with the target KG that can be used for NEE downstream.

Seed enrichment

Entities that are in the close vicinity of the seed are presumed to be relevant to our application domain as well. However, determining how close an entity should be to be included in the local cluster is where the trade-off will be made between precision and recall for the named entity extraction.

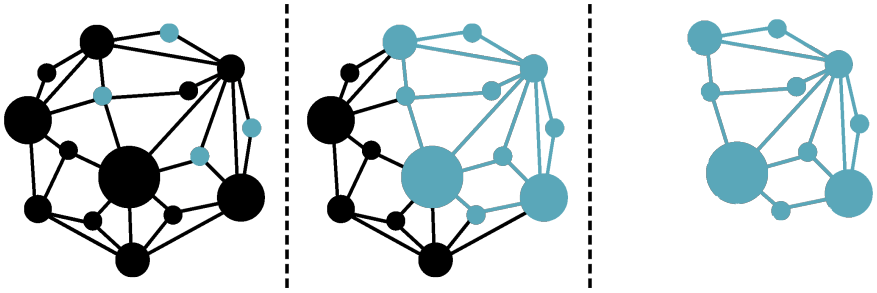

Figure 8: Wikidata KG (not to scale) where (left) the overlapping entities with ESCO are highlighted in blue, (middle) shows the results after seed enrichment and applying LGC, and (right) after irrelevant entities are pruned.

One of the limitations of LGC is that it cannot take into account the entity types in the seed. To overcome this, the entity types of the seed entities can be used to enrich the seed entities before applying LGC.

Seed enrichment is based on the presumption that if the seed contains a large percentage of entities of a certain type, then the other entities of that type are likely to be relevant as well. If for example the seed contains over half of the entities that are of the type academic discipline, then the other Wikidata entities that are not mapped with ESCO (and thus are not present in the seed) can be presumed to be equally as relevant. By this presumption we can enrich the local cluster in an automated way by adding the direct children of the most common entity types.

Five parameters have been identified that affect the seed enrichment. These have been summarized in table 5. The maximum number of KG alignment predictions has been determined using the results of the KG alignment (figure 5), and has been set to 2. This gave the best trade-off between coverage, MRR, and precision at k. To find the optimal set of parameter values a grid search has been done utilizing the evaluation dataset as described in section 5.2.

Local graph clustering

In section 2.2 a number of state-of-the-art LGC methods have been discussed. What they all have in common is that they take two main things as input, namely the graph in which to find a local cluster and a seed around which to find the cluster. In our scenario the graph is Wikidata, and the seed is the output that was found and enriched in the previous steps.

A limitation of LGC is that it cannot deal with input graphs that contain relation types, entity types or data contents of an entity. The Wikidata graph was therefore cast to

Table 5: Seed enrichment parameters

Parameter name	Description	Value
max_alignment_predictions	Maximum number of alignment predictions per entity to generate the seed	2
top_n	Number of entity types to add ordered by the percentage of entities of that type in the seed	25
min_children	Minimum number of children entities of the entity type to be added	100
max_children	Maximum number of children entities of the entity type to be added	10000
add_before_clustering	Adding the children entities before or after applying LGC	Before

the 1st normal form [54]. To explain this in terms of rows and columns: The graph is stored as a table with two columns that both contain unique identifiers (IDs) of an entity. The table contains as many rows as there are relations between entities after filtering duplicate relations (i.e. a pair of entities that has more than one relation with each other is only present once). Thus each row represents that there is a relation between one entity and some other entity.

Different LGC methods have been experimented with, including spectral clustering, flow clustering, MQI, SimpleLocal, Approximate PageRank, and L1 regularized PageRank . Again, to find the optimal set of parameter values a grid search (combined with the grid search of the seed enrichment) has been done utilizing the evaluation dataset as described in the next section. Eventually we settled on using the strongly local graph clustering method L1 regularized PageRank with the teleportation probability and regularization parameter set to 0.5 and $1.0\mathrm{e}{-4}$, respectively.

5.2 Evaluation

Six different metrics have been used to measure the quality of the resulting KG [37, 60]:

- Recall
- Average population
- Cohesion
- Conductance
- NEE performance

- NEE computation cost

The recall is the percentage of the entities in the train/test set (described below) that is present in the predicted cluster.

Secondly, the average population is a measure that gives an indication of the number of entities compared to the number of entity types. Here, entity types are considered to be entities that are a parent of one or more entities. The number of entities that are a child of an entity is only defined by the direct children, thus not including grandchildren. For example, Python is an entity that is a child of the entity programming language. The entity computer science term has in turn programming language as a direct child, but not Python. Formally, the average population (AP) of entity types in a KG is defined as the number of entities of the KG (E) divided by the number of entity types defined in the KG (T):

$$\text{Average population } (AP) = \frac{|E|}{|T|}$$

A low average population would indicate that the entities in the KG may be insufficient to represent all the knowledge in the schema.

Thirdly, the cohesion represents the number of separate connected clusters among the entities in a KG. It can help identify the "islands", and indicate what areas need more entities in order for it to be more closely connected. Formally, the cohesion of a KG is defined as the number of separate connected clusters (SCC) of the KG:

$$\text{Cohesion} = |SCC|$$

The outcome is an integer representing the number of separate clusters. Ideally a cohesion of 1 is achieved, indicating a fully connected KG.

Fourthly, the conductance of a cluster within a graph is defined as the ratio of the number of relations that go outside of the cluster over the relations within the cluster [37]:

$$\text{Conductance} = \frac{\text{between-cluster relations}}{\text{within-cluster relations}}$$

A lower conductance indicates a better cluster as it indicates that the internal connectivity is better than its external connectivity. A conductance below 1 is therefore desirable.

Second to last, and possibly the most important for the resulting KG's intended application, is comparing how the NEE performance is affected after pruning. This greatly determines whether pruning should be done more or less aggressively. It is presumed that the pruning will lead to an increase in NEE precision, and a decrease in recall.

Lastly, the computation cost of NEE, which is expected to decrease as a consequence of less entities that are present in the KG. The smaller model size should lead to a lower memory usage, initialization time, and inference time.

Dataset

To evaluate the pruning performance a set of Wikidata entities is needed of which are known to be relevant for the job domain. A small subset of the entities in the current online version of Wikidata already contain a property with the URI of its equivalent in ESCO. However, it is unknown how these mappings were produced. It is possible that (a subset of) these have been generated automatically using ESM, which seems plausible after some manual sampling. Therefore, this set of mappings was not used to evaluate the alignment method. The reason being that it would likely give a biased result, as a mapping of two entities of which both have the exact same name are generally the easiest to align with an automated approach.

However, it is relatively safe to say that this set of Wikidata entities is in fact relevant for the job domain, since they were found to be equivalent to some entity in ESCO. Therefore, it was decided for them to be used to evaluate to what extent the cluster that was found covers these entities.

Table 6 summarizes the complete evaluation dataset. 236 Wikidata entities have a mapping with an entity in ESCO with the entity type occupation. For the entity types skills and knowledge, this is 3 and 162, respectively, resulting in a total of 401 Wikidata entities that are known to be relevant. However, it was found some were qualitatively poor. 14 entities did not have a single relation to other entities (i.e. it was unconnected from the rest of the KG), such as medical laboratory assistant. Only the preferred label from ESCO was copied over. Another 3 entities did not have a preferred or alternative label (i.e. there was no label defined). These have been excluded from the dataset.

In total 384 entities were used with a stratified split of 70/30 for the training and test set. The training set has been used to do our grid search to find good parameters for the seed enrichment and LGC. The test set has been used to evaluate the pruning performance, of which the results are presented in the next section. It should be noted that the ratio of occupations, skills, and knowledge is not representative of the true ratio in ESCO. The Wikidata entities, that were found to be overlapping with ESCO entities of the type skills, are underrepresented.

Table 6: The complete set of labels used for training and test purposes of our pruning method

ESCO Entity type	Exact match	Unconnected	No label	Total	
Occupations	236	13	3	220	57.3%
Skills	3	1	0	2	0.5%
Knowledge	162	0	0	162	42.2%
Total	**401**	**14**	**3**	**384**	**95.8%**

5.3 Results and discussion

Figure 9 shows the top 25 entity types with the highest percentage of child entities in the seed. All other child entities of these entity types have been added to the cluster during the seed enrichment. Two entity types stick out above the rest, namely professions and academic disciplines. Together they make up 7.6 percent of the seed entities. This is caused by a large portion of the ESCO occupations being aligned with entities of these Wikidata types. Skills and knowledge on the other hand are aligned with entities of a more diverse set of Wikidata types. When looking at the other entity types, they all seem to be related or relevant to the job domain.

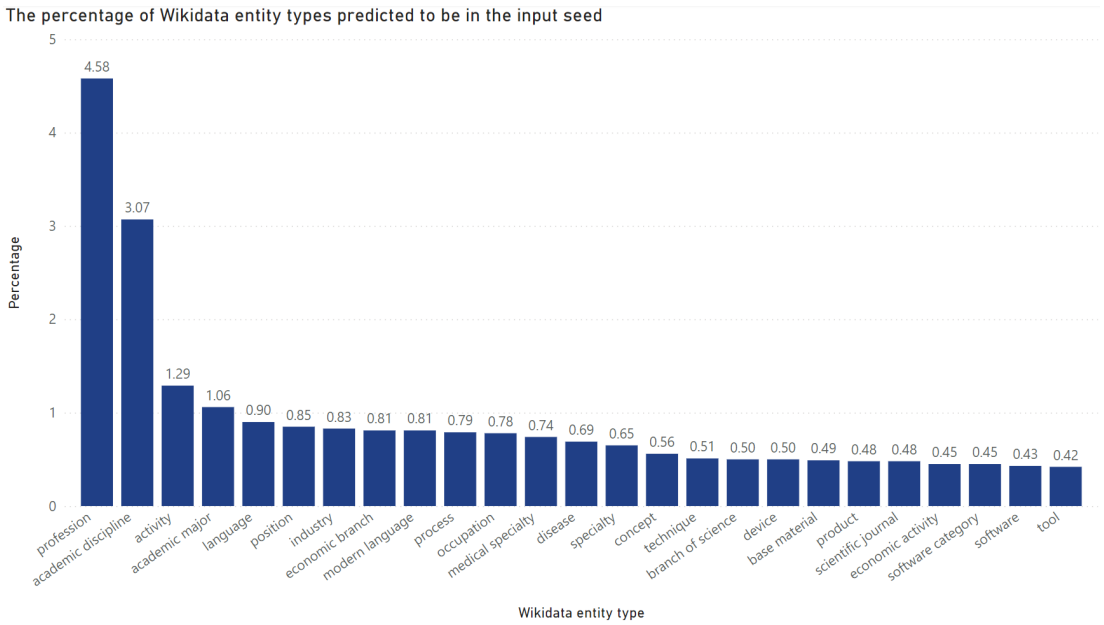

Figure 9: The entity types (25) that have been added during seed enrichment (using ESCO to generate the seed)

Table 7 summarizes the results for each step in the pruning pipeline[13]. The pipeline steps are listed from left to right where each set of predicted relevant entities is the seed for the next step. The input is a seed of 9,904 entities with a conductance of almost 1, and the output is a cluster of 616k entities where the conductance dropped to 0.700. This indicates that the internal connectivity has improved relatively to the external connectivity, resulting in a more densely connected cluster. The overall run time is under 30 seconds (single-threaded).

Table 7: Pruning results for each step in the pruning pipeline

	Seed	**Seed enrichment**	**Local graph clustering**	
Total seed entities	9,904	9,904	53,007	
Total predicted relevant entities	NA	53,007	616,442	
Run time (s)	NA	0.89	26.15	
Conductance	0.995	0.966	0.700	
Train / test set	Train	Train	Train	Test
Recall before	0.6618	0.6618	0.9564	0.9492
Recall after	NA	0.9564	0.9927	0.9915

Interesting to point out is that the set of seed entities already achieves a recall of 0.6618. This is significantly higher than the recall achieved in the similar scenario using the KG alignment evaluation dataset, namely 0.21 (corrected with the theoretical maximum). This confirms our assumption that the mappings that were already present in Wikidata may in fact be automatically generated in some way.

After seed enrichment the recall increases to 0.9564. After applying LGC the recall increases further to 0.9927 and 0.9915 for the training and test set, respectively. For the training set there were two false negatives (entities that were not "found"), these were the entities golf and tennis. For the test set there was a single false negative, namely the entity boxing. All three entities are a type of sport, which apparently was not identified as relevant in our pruning pipeline.

Furthermore, table 8 summarizes the results for each of the KGs along with the constructed target KGs. The pruned KG is almost a factor 10 smaller than the aligned Wikidata and ESCO KG, while the number of entity types decreased with a factor of about 4. The average population of the pruned KG is roughly in between Wikidata's and ESCO's. At the same time the cohesion (i.e. the number of SCCs) decreases from around 11k to 260. This decrease can be attributed to a portion of the SCCs in Wikidata being pruned from the KG, but also by some SCCs now being connected to the cluster through the alignment with one or multiple entities in ESCO.

[13] The computer hardware that was used for performance benchmarks include a dual Intel Xeon X5675 (2012) CPU, of which 10 cores (20 threads) were available, with 88 GB of DDR3 memory.

Table 8: Pruning results summarizing each of the knowledge graphs

	Wikidata	ESCO	Wikidata + ESCO	Wikidata (pruned)	Wikidata + ESCO (pruned)
Total entities	5.83M	17.4k	5.85M	599k	616k
Total entity types	23,413	1,002	24,415	5,080	6,082
Average population	249.12	17.40	239.62	117.91	101.36
Cohesion	11,526	1	11,362	297	260

Finally, figure 10 shows the 25 entity types that have been added during seed enrichment when the raw text of the General Data Protection Regulation (GDPR) was used to generate the seed. This will be further discussed in section 7.1.

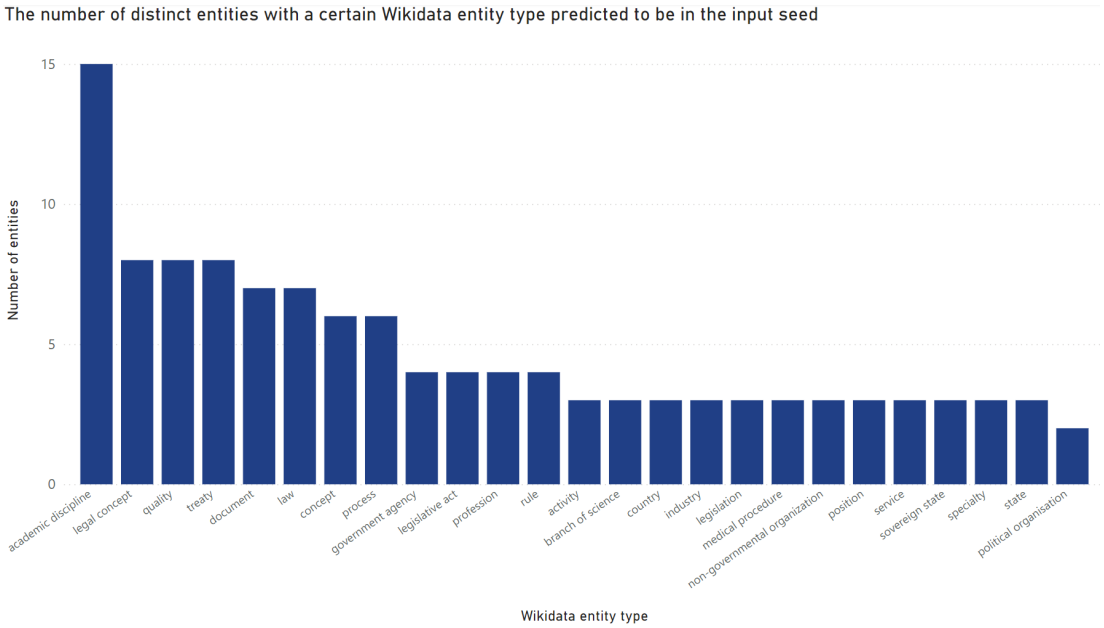

Figure 10: The entity types (25) that have been added during seed enrichment (using the text of the General Data Protection Regulation (GDPR) to generate the seed)

5.4 Summary

To summarize, we have shown how a set of seed entities, generated using wikification, can be used to prune Wikidata by removing entities that are irrelevant to our application domain. This has been done using a strongly local method, meaning the run time only depends on the size of the seed. During evaluation it was found that over 99% of the entities that were marked as relevant were present in the final cluster.

6 RQ 3: Named entity extraction

The output of the first research question was an alignment between Wikidata and ESCO. In the second research question this alignment has been used to prune irrelevant entities from Wikidata. This section explains how these results have been used to construct the final target KG by interconnecting (the pruned version of) Wikidata and ESCO. Furthermore, this section aims at providing a method to answer the third research question: *How is the performance of named entity extraction of a wikification model affected when an aligned and pruned target knowledge graph is used?* This is the third and last step in the processing pipeline (see steps 3A, B and C in figure 2), which includes data gathering, pre-processing & data annotation, and NEE.

GERBIL [10], which is an extension to the BAT framework [4], is a benchmarking platform for NEE/D tools. It distinguishes between multiple experiment types, including scored annotation to knowledge graph (S/A2KG), as depicted in figure 11, which has been attempted here. For a given NL document, the goal of the experiment is to find all named entities, including their position in the document (NER), and link them to a KG with a certain confidence level (NED / NEL). In case the entity is present in both KGs, a link to either Wikidata or ESCO is considered correct, as long as one of them is found.

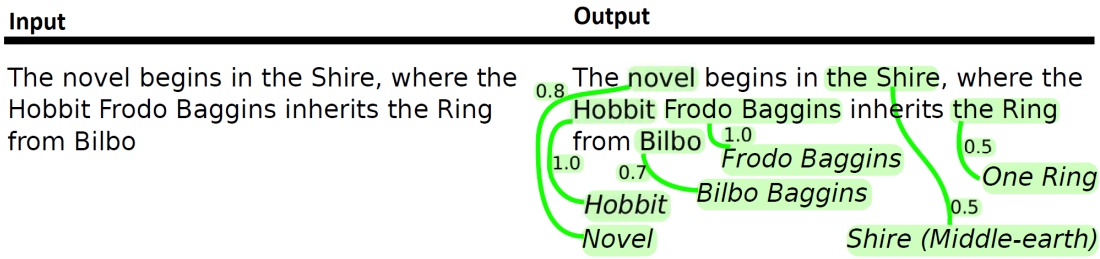

Figure 11: Scored annotation to knowledge graph (S/A2KG) (adapted from [4])

6.1 Method

For evaluation purposes, a set of NL documents has been gathered, pre-processed, and annotated with links to both KGs. This will be further discussed in section 6.2.

In order for a wikification tool like Bootleg, which was discussed in section 3.2, to work with (entities from) another target KG, a form of zero-shot learning (ZSL) [61] has been used. ZSL is an ML problem where at testing, mentions are linked to certain entities while having never seen those entities during training. It is known for having minimal human intervention, as it relies on combining previously known concepts with existing auxiliary information.

To make optimal use of Bootleg's three-layered hierarchy of signals, several adaptations have been made. To embed the entity payload, ESCO first has been parsed to Bootleg's internal format. The entity patterns, which is the first signal, have been initialized by extracting the embeddings of the entity titles using BERT's base, cased model [45]. The other two signals, the KG relations and type patterns, are contained in each entity's so-called entity profile, as was briefly touched upon in section 3.2. This consists of the entity's ID, title, types, mentions & weights, and relations with other entities including its type of relation. For example, the entity profile that was constructed for the ESCO entity data scientist contains the following information (truncated):

```
{
  "entity_id": "258e46f9-0075-4a2e-adae-1ff0477e0f30",
  "title": "data scientist",
  "types": ["occupation"],
  "mentions": [
    {"mention": "data scientist", "weight": 2.0},
    {"mention": "data research scientist", "weight": 1.0},
  ],
  "relations": [
    {"relation": "same as", "object": "Q29169143"}
    {"relation": "has field", "object": "7ee4c2ea-b349..."},
  ]
}
```

For Wikidata entities the mentions and weights were sourced by using the anchor texts in Wikipedia articles and their number of occurrences, respectively. For the ESCO entities this information is not available. Instead, the preferred and alternative label(s) are used as mentions, where the former has an increased weight over the latter indicating a greater importance.

In order to interconnect Wikidata and ESCO, we again use the results of the KG alignment (shown in figure 5). While we made use of 2 alignment predictions per entity during the pruning process, we only use a single prediction to interconnect the two KGs. This gives us the best precision, which is deemed more important than the coverage, because it limits the amount of noise that is introduced into the target KG. The alignment that was found in Wikidata, data scientist, is stored in the relations field of the entity profile.

An alternative to constructing new entity profiles for ESCO entities is to actually merge them with their aligned entity in Wikidata. However, we opted to keep them separated such that, again, the least possible amount of noise is introduced into Wikidata's KG (only the alignments with ESCO entities are added). In the case more KGs need to be aligned with Wikidata, this is the more scalable approach, as it remains possible to untangle the KGs easily.

Four models with different target KGs have been constructed: one with only ESCO as target KG, one with a pruned version of Wikidata, one with both Wikidata *and* ESCO, and lastly one with both a pruned version of Wikidata *and* ESCO. In reality the model with ESCO as target KG utilizes the pruned model with Wikidata and ESCO as target KG internally. If a mention is linked to a Wikidata entity which in turn is aligned with an entity in ESCO, then the latter entity is predicted. On a similar note, if a mention is linked to a Wikidata entity that has no alignment with ESCO, then it is discarded (because we only want predictions for the ESCO target KG).

Baseline

Initially the same baseline method described in section 4.1 was used, namely Jaccard similarity. This resulted however in a large number of false positives, even when using character-based as opposed to word-based similarity. Instead, ESM with popularity voting has been used with the same pre-processing steps from section 3.2. This includes normalizing the strings to UTF-8 characters, eliminating diacritics, lower-casing, and removing ASCII characters that are not alpha-numeric. In case multiple candidate entities are found for a given mention, then the most popular entity is selected (i.e. the one with the largest weight, as discussed in section 3.2).

6.2 Evaluation

The NEE performance has been evaluated in two-fold: quantitatively and qualitatively. The latter is mainly used to assess the quality of the former.

Quantitative evaluation

In a way, named entity extraction can be considered a type of classification task: one with an extreme number of classes. A common approach to quantitatively evaluate the performance of such tasks is by using confusion matrices with the number of true positives (TP), false positives (FP), and false negatives (FN). From these matrices other metrics such as the precision, recall and F1-score have been calculated:

$$\text{Precision } (P) = \frac{TP}{TP + FP}$$

$$\text{Recall } (R) = \frac{TP}{TP + FN}$$

$$\text{F1-score} = \frac{2 \cdot P \cdot R}{P + R}$$

These metrics are calculated across all entities that appear in the set of IT-related documents used during testing. A distinction has been made between five strictness levels:

(i) Weak matching

(ii) Strong matching

(iii) Ignoring entity ID

(iv) With (in)directly connected entities

(v) Non-IT related documents

A match is weak (i) if the (predicted and annotated) entity IDs are equal, and the mentions overlap with at least one word. A match is strong (ii) if the entity IDs are equal, and the mentions are exactly the same. This distinction is made because strong matching can be misleading due to the ambiguity of the task [10, P. 7]. A document can contain an entity such as "Microsoft Excel", while the prediction of an annotator only marks "Excel" as entity. Strong matching would count this prediction as incorrect, while a human might judge it as correct. The distinction between ignoring the entity ID or not (iii) helps evaluating the ability to recognize entities being mentioned in text (i.e. the NER performance). Second to last is the distinction between counting only the annotated entity ID correct, and counting (in)directly connected entities correct as well (iv). This tells something about how far off of the truth the predictions are. Finally, we distinguish the weak matching performance on non-IT related documents (v) to see how well the method generalizes to other domains.

Different combinations of strictness levels, approaches, and target KGs have been compared with one another, of which an overview is given in table 9. Here the original Bootleg model means the pre-trained model described in section 3.2. Furthermore, for each of the APEx models and Bootleg a comparison has been made between the target KG size, initialization time, run time, and memory usage.

Table 9: Combinations of approaches and target KGs that have been evaluated

Target KG	Baseline	Bootleg (original)	APEx
Wikidata	✓	✓	×
Wikidata (pruned)	✓	×	✓
ESCO	✓	×	✓
Wikidata + ESCO (without alignment)	×	×	✓
Wikidata + ESCO	✓	×	✓
Wikidata + ESCO (pruned)	✓	×	✓

As of writing, GERBIL is limited to 6 evaluation metrics, namely the micro- and macro-variants of precision, recall, and F1-score. Unfortunately, it does not provide insight into the individual cases of TPs, FPs, and FNs making the analysis of errors cumbersome. Although it does support differentiating between weak and strong matching (strictness levels i and ii), there is no support for strictness levels iii and iv. Finally, it does not seem possible to take the confidence levels of the gold annotations into account, which is further discussed below. For these reasons it has been opted to not make use of GERBIL, but instead implement our own evaluation.

Qualitative evaluation

A qualitative evaluation has been performed by an independent expert with experience in the field of recruitment. For each of the three categories (TPs, FPs, and FNs) 50 predictions (by APEx with the pruned Wikidata and ESCO as target KG) have been randomly sampled with distinct predicted entity IDs. Each of the predictions have been scored by the expert to which extent he agrees with the correctness of the prediction (or annotation in case of the FNs). The agreement level has been scored on a scale from 1 to 5: strongly disagree, mostly disagree, neutral, mostly agree, strongly agree. By mapping this to a percentage, where 1 = 0%, and 5 = 100% with steps of 25%, we have assessed the quality of the quantitative evaluation.

Dataset

There are datasets available to evaluate the NEE/wikification performance [10, p. 15], but none of them cover the job domain nor do they have annotations for ESCO. Most of the datasets consist of news articles or Tweets. Instead we decided to manually annotate a dataset. A set of NL documents have been gathered where each mention has been annotated with its entity in Wikidata *and* ESCO (see step 3A and B in figure 2).

20 publicly available job offer templates have been randomly sampled from Talentlyft[14] of which 16 are from the IT domain, and 4 from other domains. Since these are templates, and not actual job offers that have been used by companies, they do not contain personally identifiable information (PII), limiting the ethical concerns. They give an approximation of true job offers in terms of structure and contents, without the negatives that come with documents that contain PII. The ethics regarding this research will be further discussed in section 7.3.

To ensure that the documents remain representative of the original version a limited amount of data cleaning has been performed. Since the job offer templates were provided as HTML pages, the only data cleaning step that was performed was the extraction of the raw text (i.e. the headings and paragraphs). As annotation tool Tagtog[15] was used. The annotations have been done with the help of the services of Zuru[16] who have provided three people with a background in computer science. Each person annotated each of the 20 documents two times, once with Wikidata and once with ESCO as target KG. To annotate a single entity, the annotator has to select the mention in the text that refers to the entity and assign it the correct entity ID. A set of annotation instructions and guidelines have been written, these can be found in appendix A.1.

The annotations for each KG independently have been merged using majority voting (i.e. a mention has to be annotated by at least 50% of the annotators for it to be retained). However, merging the annotations for two distinct KGs with overlapping entities is more complex. Illustrating how the TPs, FPs, and FNs have been calculated is best explained through an example. For this, we use the following sentence: *A Master of Science in Engineering is required*, along with the gold annotations in table 10, and three alternative predictions in table 11.

Table 10: The gold annotations for the example sentence: *A Master of Science in Engineering is required*

	Mention	Confidence level	ID	KG
Gold annotations	Master of Science	1.0	P	W
	Engineering	0.5	Q	E
	Engineering	1.0	R	W
	Master of Science in engineering	1.0	S	W

The gold annotations show for each mention in the sentence the confidence level (which is defined by the percentage of the annotators that agree on this annotation), the correct ID, and in which KG it is present. The predictions also contain a confidence level (as

[14] https://www.talentlyft.com/en/resources/templates
[15] https://www.tagtog.net/
[16] https://zuru.ai/

Table 11: Different predictions for the example sentence: *A Master of Science in Engineering is required*

	Mention	Confidence level	ID	KG	Strong			Weak		
					TP	FP	FN	TP	FP	FN
Prediction A	Engineering	1.0	Q	E	0.5	0	1.0	0.5	0	1.0
Prediction B	Master	0.8	P	W	0	0.8	1.0	0.8	0	1.0
Prediction C	Master of Science in Engineering	1.0	S	W	1.0	0	0	1.0	0	0

predicted by the NEE model), the ID, in which KG it is present, and the confusion matrix for both strong and weak matches.

To calculate the TPs, FPs and FNs the confidence levels of both the annotations and predictions are taken into account. This ensures that annotations and predictions with a high confidence level weigh heavier than ones with a low confidence level. Looking at *prediction A*, the confusion matrix for strong and weak are equal as it is a strong match. The TP is calculated as the confidence level of the correct prediction times the confidence level of the annotation ($1.0 * 0.5 = 0.5$). The FN is explained by the *Master of Science* mention. The mentions *Engineering* with Wikidata as KG, and *Master of Science in Engineering* are ignored because they overlap with the correct prediction for *Engineering*.

Prediction B is a weak match, but not a strong match. The FN is 1.0, which is explained by the *engineering* mention with Wikidata as KG (because this has the highest confidence level). *Prediction C* is again a strong match. The other mentions are ignored, as the TP overlaps with all other mentions.

Table 12 summarizes the complete evaluation dataset. Since there is no need for training, the entire dataset has been used for evaluation purposes. In total 1191 entities have been annotated across 20 documents resulting in on average 32.6 and 27 entities per document, for Wikidata and ESCO, respectively. In total 413 distinct entities have been annotated, which indicates that a fair number of the annotated entities appear more than once in the dataset. Relatively more Wikidata than ESCO entities have been annotated, which is to be expected due to its KG being larger.

To measure the quality of the dataset, the Inter-Annotator Agreement (IAA) is used, which is the degree of consensus among the annotators. The higher the percentage, the likelier that the annotations are correct. When taking into account that each mention must have been assigned the correct entity ID, then the strong IAAs for Wikidata and ESCO are 59.7% and 57.7% , respectively. When ignoring that each mention must have been assigned the correct entity ID, the IAAs are 68.6% and 63.7%, respectively. The difference in IAA for strong and weak matching are marginal.

Table 12: The complete evaluation dataset used to evaluate the NEE performance

KG	Entity type	Entities		Distinct entities	
Wikidata	**Subtotal**	**651**	**54.7%**	**246**	**59.6%**
ESCO	Occupations	148	27.4%	34	20.4%
	Skills	187	34.6%	63	37.7%
	Knowledge	205	38.0%	70	41.9%
	Subtotal	**540**	**45.3%**	**167**	**40.4%**
Total		**1191**	**100.0%**	**413**	**34.7%**

6.3 Results and discussion

An example of a document annotated by APEx using the pruned version of Wikidata and ESCO as target KG can be found in appendix A.2.

Table 13 compares the NEE results for the target KGs Wikidata and ESCO separately using weak matching. When extracting entities with Wikidata as the single target KG, then ESM is outperformed by Bootleg. APEx in turn seems to outperform Bootleg in terms of precision by a small margin, while the recall remains equal. The marginal increase in precision may be explained by the pruning process having removed irrelevant entities (i.e. entities not related to the job domain), and thus increasing the probability that a mention is linked to an entity that is known to be relevant. The same, but to a larger degree, can be seen when comparing the performance of ESM with the (un)pruned version of Wikidata. With ESCO as target KG, APEx scores similarly in terms of precision compared with ESM. At the same time the recall shows an increase of about 10 percentage points. This increase can be explained by the fact that APEx may not link a given mention to an ESCO entity directly, but also through the internal alignment between Wikidata and ESCO, as was elaborated on in section 6.1.

Table 13: Comparing the NEE results for the target KGs Wikidata (pruned) and ESCO separately

Target KG	Method	Precision	Recall	F1-score
Wikidata	Bootleg	**0.582**	**0.589**	**0.585**
	ESM	0.459	0.490	0.474
Wikidata (pruned)	APEx	**0.607**	**0.588**	**0.598**
	ESM	0.501	0.507	0.504
ESCO	APEx	0.674	**0.626**	**0.649**
	ESM	**0.679**	0.528	0.594

Table 14 compares the NEE results for APEx for the cases when making use of the (un)aligned / (un)pruned target KG. In case the alignment between Wikidata and ESCO is used, both the precision and recall show an increase of approximately 4 percentage points. Even though the alignment is not perfect, it still seems to positively affect the

NEE performance. Moreover, when using the pruned KG the recall does not seem to be affected significantly, while the precision seems to show a slight increase. The fact that the recall does not seem to be affected when using the pruned KG is in line with the pruning results that were discussed above, as over 99 percent of the relevant entities were maintained in the pruned KG. The slight increase in precision may again be attributed to the pruning of entities not relevant to the job domain.

Table 14: Comparing the NEE results of APEx for the cases when (not) making use of the aligned / pruned target KG

Target KG	Precision	Recall	F1-score
Wikidata + ESCO (without alignment)	0.625	0.655	0.640
Wikidata + ESCO	0.665	**0.691**	0.678
Wikidata + ESCO (pruned)	**0.688**	0.689	**0.688**

Table 15 lists the results for each of the five strictness levels as defined in the previous section. For each strictness level the performance is compared between the baseline (i.e. ESM), and APEx, as well as between the pruned target KG, and the non-pruned target KG. The strong matching (ii) performance shows a decrease of roughly 2 percent across the board compared with weak matching (i), which is unsurprising given that strong matching is more strict. In either strictness levels APEx shows a better performance compared with ESM.

When looking at the NER performance (i.e. ignoring whether the predicted entity IDs are correct (iii)) both methods and target KGs seem evenly matched in terms of recall. Given that both APEx and ESM largely depend on a syntactic method to generate entity candidates for a mention, this is not surprising. Furthermore, it indicates that only 77.4% of the mentions in text are recognized. Thus, even if the disambiguation were to be correct for 100% of the recognized mentions, still 22.6 percentage points of the loss in recall can be attributed to the NER phase. Taking this 77.4% into account as the theoretical maximum, then the recall of only the disambiguation phase would be estimated at 0.890 ($\frac{0.689}{0.774}$), for APEx using the pruned KG.

In the strictness level where (in)directly connected entities to the gold entity are counted as correct as well (iv), it can be seen that APEx overall is "closer" to the ground truth than ESM. With closer is meant that APEx' predicted entities are in more cases either correct or (in)directly connected to the ground truth than ESM's predicted entities, with 73.9 and 67.8 percent respectively. This may be explained by the fact that APEx can utilize the context in which a mention appears, while ESM cannot.

In the last strictness level, measuring the weak matching performance for non-IT documents (v), the recall takes a hit compared with the weak matching performance for IT documents, dropping from 0.689 to 0.549 for APEx using the pruned KG. One of the

possible explanations for this is that non-IT documents contain relatively more (soft) skills than knowledge. Looking at the gold set of annotations for ESCO, the knowledge to skills ratio for IT and non-IT documents is on average 62:38 and 40:60, respectively. While (soft) skills tend to be harder to automatically extract, they also seem more ambiguous in definition, as opposed to knowledge. This also seems to be confirmed by the IAA, which is about 5 percentage points lower for non-IT documents.

What all of the strictness levels have in common is that APEx outscores ESM in terms of F1-score. The difference in APEx' performance when using the pruned version of the KG or not however, is marginal. Thus, our presumption that pruning Wikidata leads to an increase in NEE precision, and a decrease in recall, cannot be substantiated with confidence in this evaluation.

Table 15: Comparing the NEE results for each of five strictness levels. Each strictness level makes use of IT-related documents and weak matching, unless specified otherwise.

Strictness level / Target KG	**Method**	**Precision**	**Recall**	**F1-score**
(i) Weak matching				
Wikidata + ESCO	APEx	0.665	**0.691**	0.678
	ESM	0.498	0.549	0.522
Wikidata + ESCO (pruned)	APEx	**0.688**	0.689	**0.688**
	ESM	0.534	0.565	0.549
(ii) Strong matching				
Wikidata + ESCO	APEx	0.644	**0.670**	0.657
	ESM	0.476	0.528	0.501
Wikidata + ESCO (pruned)	APEx	**0.665**	0.668	**0.667**
	ESM	0.513	0.545	0.528
(iii) Ignoring entity ID				
Wikidata + ESCO	APEx	0.779	0.785	0.782
	ESM	0.760	**0.788**	0.774
Wikidata + ESCO (pruned)	APEx	**0.793**	0.774	**0.783**
	ESM	0.774	0.774	0.774
(iv) With (in)directly connected entities				
Wikidata + ESCO	APEx	0.715	**0.735**	0.725
	ESM	0.634	0.677	0.655
Wikidata + ESCO (pruned)	APEx	**0.739**	**0.735**	**0.737**
	ESM	0.678	0.693	0.685
(v) Non-IT documents				
Wikidata + ESCO	APEx	0.620	**0.567**	0.592
	ESM	0.512	0.487	0.500
Wikidata + ESCO (pruned)	APEx	**0.662**	0.549	**0.600**
	ESM	0.536	0.468	0.499

Table 16 compares the total number of entities, initialization time, run time (multi-threaded), and memory usage (each using a 10 times average) for Bootleg and each of the APEx models. After pruning the target KG the model initialization time decreases from roughly 9 to 1 minute(s). On a similar note the memory usage is decreased by a factor of 6.5, down to 3.43 GB. The run time to pass a single document through the model on the other hand is not significantly affected. This indicates that Bootleg is already reasonably optimized for run time performance. The metrics for the ESCO and the pruned model are identical. This is to be expected, because the ESCO model in fact uses the pruned model to make its predictions, as was explained in section 6.1.

Table 16: Comparing the total number of entities, initialization time, run time, and memory usage for Bootleg and each of the APEx models

Target KG	Method	Total entities	Initialization (mm:ss)	Run time 1 doc (mm:ss)	Memory usage (GB)
Wikidata	Bootleg	5.833M	08:27	00:09	22.16
Wikidata (pruned)	APEx	599k	01:09	00:09	3.23
ESCO	APEx	17.4k	01:11	00:09	3.43
Wikidata + ESCO	APEx	5.850M	09:06	00:10	22.36
Wikidata + ESCO (pruned)	APEx	616k	01:11	00:09	3.43

In fact, one more model was constructed with a target KG that was more aggressively pruned than the one described in table 16, containing 353k as opposed to 616k entities. Although the initialization time and memory usage decrease further, also the precision, recall, and F1-score seem to start decreasing. In the case of weak matching the precision dropped from 0.688 to 0.677, and the recall dropped from 0.689 to 0.683, in comparison with the pruned model described above.

Table 17 lists a sample of annotations (by APEx) that were marked as true positive, false positive, and false negative in the quantitative evaluation. For some annotations it is highly ambiguous whether it is correctly annotated or not. For example, the mention *Digital Media* in sentence 3 has arguably been wrongly annotated as the entity describing machine-readable data. Instead, the entity that is an academic discipline, which is also labeled digital media, would suit the context better. However, it seems that there is an error in the Wikidata KG, as the entity describing machine-readable data is also marked as a child of academic discipline, which it should not be. Another interesting one is sentence 4, where the mention *Computer Science, Engineering* is annotated by APEx as Computer Science and Engineering. Although this entity is not explicitly mentioned in the sentence, it is arguably not entirely incorrect either. Furthermore, in sentence 9 the mention *Web Designer* has been annotated as web developer as the ground truth, which could be argued differently as well. Lastly, in sentence 5 the mention *health (Security* is annotated as health security. This can be explained by our NER methodology, which

treats punctuation such as parentheses, (semi-)colons, and slashes as if they were not there. This could be fixed by treating these punctuation marks as full stops, but this would in turn introduce other errors (e.g. in a sentence like: *The usage of data (science) can give businesses a competitive advantage*).

Table 17: A sample of annotations (by APEx) that were marked as true positive, false positive, and false negative in the quantitative evaluation

	ID	Sentence
TPs	1	X years of experience with JavaScript, CSS and jQuery
	2	Stay motivated to actively engage with customers
	3	Bachelor's Degree in Digital Media & Arts program
FPs	4	BS, MS degree in Computer Science, Engineering, MIS or similar relevant field
	5	Take responsibility for the overall operational health (Security, Availability, Performance, Interoperability and Reliability) of our data communications systems
	6	Work in a multidisciplinary team with other professionals such as back-end developers and web designers
	7	Provide guidance and support to Application Developers
FNs	8	X years of experience with Transparent Data Encryption
	9	Web Designer duties and responsibilities
	10	Interpersonal and communication skills

Finally, the agreement scores that were found during the qualitative evaluation for the predictions, or in the case of FNs the annotations, that were marked as TPs, FPs, and FNs were 0.73, 0.31, and 0.65, respectively. In other words, the 0.73 means that the quality assessor on average in 73% of the cases agreed upon a prediction being labeled as a TP. We find that the average agreement score is relatively low. This is corroborated by the relatively low IAA that was found for the evaluation dataset. Especially the low agreement score for the FPs is striking, where 69% of the FPs were found to be "false" false positives. Next to the ambiguity introduced by the synonymy and polysemy of mentions, this may also be partly due to how the annotations were merged, as only mentions annotated by at least 50% of the annotators were retained. In the case we were to correct the quantitative evaluation scores with the agreement scores (i.e. the 27% of the TPs are counted as FPs and vice versa), then the precision, recall, and F1-score for the pruned APEx model (strictness level i) would be 0.840, 0.717, and 0.773, respectively.

6.4 Summary

To summarize, we have shown how an existing wikification method can be adapted to support non-Wikidata entities. This has been used to extract entities with target KGs consisting of (a pruned version of) Wikidata, ESCO, and combinations thereof. The quantitative and qualitative evaluation show that APEx outperforms the ESM baseline, and achieves competitive performance compared with Bootleg's original model, in nearly all combinations of target KGs and evaluation strictness levels.

7 Discussion

Throughout this thesis various results are shown. The results of the KG alignment, KG pruning and NEE have already been discussed in section 4.3, 5.3, and 6.3, respectively. In this section we will limit to a more abstract discussion, including the generalizability, costs, ethics, and challenges.

Coming back to the contributions mentioned in section 1.4, we first of all, have shown how wikification can be leveraged for other purposes. This includes how to align ESCO with Wikidata in section 4, how to prune Wikidata to a specific domain in section 5, and how to perform NEE with multiple target KGs in section 6. Second, we contribute to Wikidata by adding an alignment with ESCO (section 4.3). Third, we have constructed multiple datasets that can be used to evaluate KG alignment (section 4.2), pruning (section 5.2), and NEE (section 6.2) methods. Finally, we give recommendations on how GERBIL can be extended to support more detailed evaluation metrics (section 6.2).

7.1 Generalizability

It has been shown that APEx can be employed with four different types of target KGs. Either a domain-specific KG, such as ESCO, a (pruned) version of Wikidata, or a combination thereof. In this thesis a domain-specific KG, ESCO, has been used to generate a seed to prune Wikidata. However, it should be possible to generalize this to any domain-specific KG, as long as the entity's data contents contain enough information to create a sufficient alignment with Wikidata. The entities should therefore at the bare minimum have a label, and preferably alternative labels and relations to other entities.

Generating a seed to prune Wikidata does not require a (domain-specific) KG per se. Instead, a document that is somewhat representative of the domain of interest should work as well by simply applying the Bootleg model to the document and using the set of extracted entities as seed. Although, this would only work in the case a sub graph of the Wikidata KG already covers the domain of interest adequately. A highly specific domain that has barely any overlap with Wikidata would thus not work. Figure 10 shows the 25 entity types that have been added during seed enrichment when the raw text of the General Data Protection Regulation (GDPR) was used to generate the seed. Similarly to the seed enrichment of Wikidata using ESCO, the majority of the entity types seem related or relevant to the legal domain. This implies that the seed enrichment (and LGC presumably as well) can generalize to other domains. Going a step further, it should be possible to scale this to multiple domains simultaneously as well. In other words, having a single model that covers two or more specific domains without having to train individual models for each, nor using the entire Wikidata KG.

7.2 Costs

To estimate the cost of deploying an APEx model to production, the VM pricing in Azure has been used. With Wikidata and ESCO as target KG the full model uses 22.36 GB of memory. The cheapest option would be a dual-core with 32 GB of memory, which costs €166.83/month (or €0.23/hour) at the time of writing. The pruned model on the other hand uses 3.43 GB of memory. The cheapest option would then be a dual-core with 4 GB of memory, which costs €57.14/month (or €0.08/hour). In both cases this is including data storage, but excluding network usage. Hence, in this scenario the pruned model would be about three times cheaper to run. The construction of a pruned model would be negligible in terms of cost.

7.3 Ethics

In this thesis it has been deliberately opted to make use of publicly available job offer templates to limit the ethical concerns. They give an approximation of true job offers in terms of structure and contents, but without the negatives that come with documents that contain PII.

Naturally, in future work APEx could be applied to documents that do contain PII, such as job offers or people's resumes. In theory, the model's output could be used to enable automated decision making (e.g. through semantic matching between these documents) to select the "best" matching candidate for a position. The usage of AI in recruitment should be implemented with extreme care to prevent any (un)intentional misuse of (the output of) the model or discrimination/bias such as in the Amazon case mentioned in section 1.1. Therefore, APEx will first of all not be made publicly available. Secondly, if it were to be made available to a specific person, or a group of people, then they should first sign a document stating the terms and conditions of using the model. This should at the bare minimum include, but is not limited to:

- If the model were to be applied to documents that contain PII, then first consent must be given by the person who is the owner of that piece of data.
- The output of APEx is not allowed to be used for automated decision making if PII is involved.
- The output can be used to enable data-informed decisions (i.e. with a human involved), but the given "advice" should be accompanied with the fact that it was generated using an AI model. The advice should have an explanation what it was based on, and should only be interpreted by someone who is knowledgeable in this application domain.

The ethics assessment and the review thereof by the UT's ethics committee can be found in appendix A.3.

7.4 Challenges

One of the major challenges in this thesis was the data management, which is introduced by the sheer size of the datasets involved. While the cost of storage is relatively low nowadays, every processing step needs to be implemented efficiently to limit the computation cost. Moving data from disk to memory can already be a lengthy task, which is partly caused by the relatively dated hardware of the server we used. Another challenge in this thesis was to get acquainted with the internal workings of Bootleg, which was fundamental to implement our method.

8 Conclusion

The aim of this research was to develop a methodology that can automatically extract entities related to a specific domain. To achieve this, we set out to answer our main research question, namely how a wikification model can be leveraged for named entity extraction with multiple target KGs. Our main RQ has been subdivided into three sub questions: 1) *How can a domain-specific knowledge graph be aligned with Wikidata?* 2) *How can Wikidata be pruned using another knowledge graph as seed?* and 3) *How is the performance of named entity extraction of a wikification model affected when an aligned and pruned target knowledge graph is used?* Our method then also consists out of three main steps: KG Alignment, KG Pruning, and named entity Extraction (APEx). To find answers to our research questions, we did a case study with the job market as application domain with the goal to analyze the method's performance when it has been optimized for the this domain.

RQ 1 - How can a domain-specific knowledge graph be aligned with Wikidata?

It was found that more than a single target KG is needed to extract all named entities related to the job market domain. Therefore, the first RQ focused on finding an alignment between two KGs. The goal was to identify a set of overlapping entities such that these can be used downstream during the pruning of Wikidata and NEE. We used one KG that is large, general and crowd-sourced (Wikidata), and another specifically covering the job domain, which is smaller, fine-grained, and curated by experts (ESCO). Using both was found to combine the best from both worlds.

We have shown how Bootleg, which is a wikification model with Wikidata as target KG, can be used to find overlapping entities between a domain-specific KG and Wikidata. This has been done by taking an alignment approach consisting of a combination of element-level, instance-level, and a hybrid of syntactic/semantic matching. We used Bootleg as-is, and thus without the need for annotated data for training purposes. It is estimated that for roughly 89% of the ESCO entities, that are known to have a ground truth alignment, we can find a close or exact match in Wikidata. Both in terms of coverage and precision at k, we found that Bootleg outperforms the Jaccard baseline by a significant margin. Also the cases where the alignment predictions are found to be incorrect, Bootleg's predictions seemed more relevant (i.e. closer to the ground truth).

RQ 2 - How can Wikidata be pruned using another knowledge graph as seed?

The second research question focused on pruning Wikidata such that mostly entities related to the job domain are retained. A strongly local method has been used, which

consists out of three steps. First, a set of seed entities has been created out of the Wikidata entities that were predicted to be overlapping with ESCO in RQ 1. Second, the seed has been enriched by looking at the top n entity types that occur in the seed, and adhere to a set of five criteria. The children of these entity types were added to the seed. All entity types that were found seemed relevant or related to the job domain. Finally, LGC has been used to find a local cluster using this seed. Out of 5 different LGC methods L1 regularized PageRank performed best. All entities not appearing in the local cluster have been pruned from the Wikidata KG reducing the total entities with almost a factor of 10 (from roughly 5.9M down to 600k entities), while maintaining 99.2% of the relevant entities.

RQ 3 - How is the performance of named entity extraction of a wikification model affected when an aligned and pruned target knowledge graph is used?

The third RQ focused on how the results of the previous two RQs can be used to create our target KG by interconnecting (the pruned version of) Wikidata and ESCO. Furthermore, its goal was to find a method to leverage Bootleg for NEE with this target KG, and to evaluate its performance.

We have seen how ZSL has been used by Bootleg to support NEE with non-Wikidata entities (i.e. entities that were never seen during training). To enable ZSL, among other things, we have shown how ESCO's KG has been parsed to Bootleg's internal format by extracting entity title embeddings using BERT, and how entity profiles have been initialized utilizing the alignment found in RQ 1 to interconnect Wikidata and ESCO. The entity profiles have been initialized with a scalable approach such that untangling the KGs remains possible. Finally, four models with different target KGs have been constructed, including (a pruned version of) Wikidata, ESCO, and combinations thereof.

The NEE performance has been evaluated in two-fold: quantitatively and qualitatively. It is estimated that the NEE of APEx, using a pruned version of Wikidata and ESCO as target KG, achieves a precision and recall of 0.840 and 0.717, respectively. It has been found that APEx outperforms the ESM baseline, and achieves competitive performance compared with Bootleg's original model, in nearly all combinations of target KGs and evaluation strictness levels. The main loss in performance can be attributed to not recognizing entities being mentioned in the text, as opposed to the disambiguation phase. However, the NEE takes a hit in performance when it is applied to non-IT related documents. Finally, it has been shown that APEx can reduce the computation cost in terms of initialization time and memory usage by a factor of 9 and 6.5, respectively. To summarize, we have shown the potential of using a wikification model for other applications than merely extracting entities to Wikidata, without the drawbacks of the computation cost of wikification, and without the need for additional training.

8.1 Limitations

While this thesis provides answers to the main research question and its sub questions, there are a number of limitations that should be considered when interpreting the results.

First and foremost is the data quality of our evaluation dataset that has been constructed to evaluate the NEE performance. Even though we followed best practices to construct a set of ground truth annotations, among others by attempting to write clear annotation guidelines, having multiple annotators, and merging annotations using majority voting, the IAA of the resulting dataset is relatively low. This is also confirmed by the qualitative evaluation, where a significant percentage of "false" false positives were found. Data quality is one of the major challenges in the field of NEE, in particular when manually-created evaluation datasets are used [62]. Annotation errors can be introduced by a large variety of causes, as there often is no "single ground truth" due to the complexity and ambiguity of the problem.

Second, although we did discuss some state-of-the-art NEE results in the related work section, we did not make a direct comparison with the NEE results found in this thesis. This is a consequence of this thesis mostly limiting to a single application domain for which no suitable evaluation dataset was available. Comparing the NEE performance across different application domains *and* different evaluation datasets is not straightforward, and most importantly, can be misleading.

Finally, our research depends on a pre-trained wikification model. This model was trained on an older version of a Wikipedia dump. Furthermore, it already had some heuristic filtering applied, meaning a percentage of relevant entities have been filtered on beforehand. Ideally we would have trained a model ourselves. However, this was not feasible due to the computing power required, and has therefore been left out of the scope of this thesis.

8.2 Future work

For future research multiple directions can be taken. Possible future research questions include:

(i) How does re-training a Bootleg model with less aggressive filtering affect the named entity extraction performance?

(ii) What other knowledge graph alignment methods can improve the alignment, and how would this affect the named entity extraction performance?

(iii) How would a semantically based named entity recognition method affect the named entity extraction performance?

(iv) How does APEx generalize to other application domains?

(v) How does APEx scale with more than two target knowledge graphs?

One option, which was also highlighted in the previous section, would be to manually train our wikification model with less aggressive filtering (i). It should be possible to increase the recall of NEE this way, but it would also be interesting to see how it would affect the precision.

Another direction for future research would be to try out different KG alignment methods (ii). In this thesis we have experimented with employing Bootleg to create an alignment between the two KGs, however it is reasonable to expect that other methods are better suited for the task, and thus give a better performance. An improved KG alignment could in turn affect the NEE performance.

Furthermore, we saw that a large portion of the errors made during NEE can be attributed to the mentions not being recognized in the text. Employing a semantically based NER method rather than a syntactically based method could improve the performance (iii).

Finally, it would be interesting to see how APEx generalizes to other application domains / KGs (iv), as well as how it scales to more than two target KGs simultaneously (v).

References

[1] Laurel Orr, Megan Leszczynski, Simran Arora, Sen Wu, Neel Guha, Xiao Ling, and Christopher Re. Bootleg: Chasing the tail with self-supervised named entity disambiguation. *arXiv preprint arXiv:2010.10363*, 2020.

[2] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.

[3] Rada Mihalcea and Andras Csomai. Wikify! linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, 2007.

[4] Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 249–260, 2013.

[5] Mathieu Nassif and Martin P Robillard. Wikifying software artifacts. *Empirical Software Engineering*, 26(2):1–31, 2021.

[6] Dean Allemang and James Hendler. *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier, 2011.

[7] Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L Opdahl, and Csaba Veres. Named entity extraction for knowledge graphs: A literature overview. *IEEE Access*, 8:32862–32881, 2020.

[8] Stefano Faralli and Roberto Navigli. Growing multi-domain glossaries from a few seeds using probabilistic topic models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 170–181, 2013.

[9] Thomas Lin, Oren Etzioni, et al. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 84–88, 2012.

[10] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. Gerbil–benchmarking named entity recognition and linking consistently. *Semantic Web*, 9(5):605–625, 2018.

[11] European Commission. European classification of skills/competences, qualifications and occupations. 2020.

[12] Robert Rentzsch and Mila Staneva. Skills-matching and skills intelligence through curated and data-driven ontologies. *Proceedings of the DELFI Workshops 2020*, 2020.

[13] J. Dastin. Amazon scraps secret ai recruiting tool that showed bias against women, 2018. Available at https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G.

[14] Gongqing Wu, Ying He, and Xuegang Hu. Entity linking: an issue to extract corresponding entity with knowledge base. *IEEE Access*, 6:6220–6231, 2018.

[15] Daniel Sánchez Cisneros and Fernando Aparicio Gali. Uem-uc3m: an ontology-based named entity recognition system for biomedical texts. Association for Computational Linguistics, 2015.

[16] Erhard Rahm and Philip A Bernstein. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350, 2001.

[17] Sophie Neutel and Maaike HT de Boer. Towards automatic ontology alignment using bert. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*, 2021.

[18] Elodie Thiéblin, Ollivier Haemmerlé, Nathalie Hernandez, and Cassia Trojahn. Survey on complex ontology matching. *Semantic Web*, 11(4):689–727, 2020.

[19] Jérôme Euzenat, Pavel Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007.

[20] Jaewoo Kang and Jeffrey F Naughton. On schema matching with opaque column names and data values. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 205–216, 2003.

[21] Bess Schrader. What is the difference between an ontology and a knowledge graph. Technical report, Enterprise Knowledge (EK), 2020.

[22] Michelle Cheatham and Pascal Hitzler. String similarity metrics for ontology alignment. In *International semantic web conference*, pages 294–309. Springer, 2013.

[23] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[24] Yuanzhe Zhang, Xuepeng Wang, Siwei Lai, Shizhu He, Kang Liu, Jun Zhao, and Xueqiang Lv. Ontology matching with word embeddings. In *Chinese computational linguistics and natural language processing based on naturally annotated big data*, pages 34–45. Springer, 2014.

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[26] Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662*, 2013.

[27] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[28] Dagmar Gromann and Thierry Declerck. Comparing pretrained multilingual word embeddings on an ontology alignment task. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[29] Satu Elisa Schaeffer. Graph clustering. *Computer science review*, 1(1):27–64, 2007.

[30] Kimon Fountoulakis, David F Gleich, and Michael W Mahoney. A short introduction to local graph clustering methods and software. *arXiv preprint arXiv:1810.07324*, 2018.

[31] Bill Swartout, Ramesh Patil, Kevin Knight, and Tom Russ. Toward distributed use of large-scale ontologies. In *Proc. of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems*, pages 138–148, 1996.

[32] Kevin Lang and Satish Rao. A flow-based method for improving the expansion or conductance of graph cuts. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 325–337. Springer, 2004.

[33] Reid Andersen and Kevin J Lang. An algorithm for improving graph partitions. In *SODA*, volume 8, pages 651–660, 2008.

[34] M Parimala Boobalan, Daphne Lopez, and Xiao Zhi Gao. Graph clustering using k-neighbourhood attribute structural similarity. *Applied soft computing*, 47:216–223, 2016.

[35] Nate Veldt, David Gleich, and Michael Mahoney. A simple and strongly-local flow-based method for cut improvement. In *International Conference on Machine Learning*, pages 1938–1947. PMLR, 2016.

[36] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 475–486. IEEE, 2006.

[37] Kimon Fountoulakis, Farbod Roosta-Khorasani, Julian Shun, Xiang Cheng, and Michael W Mahoney. Variational perspective on local graph clustering. *Mathematical Programming*, 174(1):553–573, 2019.

[38] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[39] Malik Nabeel Ahmed Awan, Sharifullah Khan, Khalid Latif, and Asad Masood Khattak. A new approach to information extraction in user-centric e-recruitment systems. *Applied Sciences*, 9(14):2852, 2019.

[40] Agnieszka Mykowiecka, Małgorzata Marciniak, and Anna Kupść. Rule-based information extraction from patients' clinical data. *Journal of biomedical informatics*, 42(5):923–936, 2009.

[41] K Tijdens and C Kaandorp. Classifying job titles from job vacancies into isco-08 and related job features-the netherlands. Technical report, SERISS, 2019.

[42] Alexandra Pomares Quimbaya, Alejandro Sierra Múnera, Rafael Andrés González Rivera, Julián Camilo Daza Rodríguez, Oscar Mauricio Muñoz Velandia, Angel Alberto Garcia Peña, and Cyril Labbé. Named entity recognition over electronic health records through a combined dictionary-based approach. *Procedia Computer Science*, 100:55–61, 2016.

[43] Baoxu Shi, Jaewon Yang, Feng Guo, and Qi He. Salience and market-aware skill extraction for job targeting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2871–2879, 2020.

[44] David Abián, F Guerra, J Martínez-Romanos, and Raquel Trillo-Lado. Wikidata and dbpedia: a comparative study. In *Semanitic Keyword-based Search on Structured Data Sources*, pages 142–154. Springer, 2017.

[45] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[46] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[47] Mariia Chernova. Occupational skills extraction with finbert. 2020.

[48] Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. *arXiv preprint arXiv:1808.07699*, 2018.

[49] Karan Goel, Nazneen Rajani, Jesse Vig, Samson Tan, Jason Wu, Stephan Zheng, Caiming Xiong, Mohit Bansal, and Christopher Ré. Robustness gym: Unifying the nlp evaluation landscape. *arXiv preprint arXiv:2101.04840*, 2021.

[50] Johannes M van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P de Vries. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2197–2200, 2020.

[51] Francesco Piccinno and Paolo Ferragina. From tagme to wat: A new entity annotator. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation*, ERD '14, page 55–62, New York, NY, USA, 2014. Association for Computing Machinery.

[52] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). volume abs/1006.3498, pages 1625–1628, 01 2010.

[53] Avirup Sil, Ernest Cronin, Penghai Nie, Yinfei Yang, Ana-Maria Popescu, and Alexander Yates. Linking named entities to any database. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 116–127, 2012.

[54] Abraham Silberschatz, Henry F Korth, Shashank Sudarshan, et al. *Database system concepts*, volume 5. McGraw-Hill New York, 2002.

[55] Chen-Tse Tsai and Dan Roth. Concept grounding to multiple knowledge bases via indirect supervision. *Transactions of the Association for Computational Linguistics*, 4:141–154, 2016.

[56] Ning Gao and Silviu Cucerzan. Entity linking to one thousand knowledge bases. In *European Conference on Information Retrieval*, pages 1–14. Springer, 2017.

[57] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[58] Antoine Isaac and E Summers. Skos simple knowledge organization system. *Primer, World Wide Web Consortium (W3C)*, 7, 2009.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[60] Samir Tartir, I Budak Arpinar, Michael Moore, Amit P Sheth, and Boanerges Aleman-Meza. Ontoqa: Metric-based ontology quality analysis. 2005.

[61] Mahdi Rezaei and Mahsa Shahidi. Zero-shot learning and its applications from autonomous vehicles to covid-19 diagnosis: A review. *Intelligence-based medicine*, 3:100005, 2020.

[62] Kunal Jha, Michael Röder, and Axel-Cyrille Ngonga Ngomo. All that glitters is not gold–rule-based curation of reference datasets for named entity recognition and entity linking. In *European Semantic Web Conference*, pages 305–320. Springer, 2017.

A Appendix

A.1 Annotation guidelines

The annotation guidelines for Wikidata can be found here, and for ESCO can be found here.

A.2 Example of an annotated document

An example of an annotated document by APEx using the pruned version of Wikidata and ESCO as target KG can be found here.

A.3 Ethics assessment

The filled in ethics assessment can be found here, and the review thereof by the UT's ethics committee can be found here.