RevOnt: Reverse Engineering of Competency Questions from Knowledge Graphs via Language Models

Fiorela Ciroku[a,*], Jacopo de Berardinis[b], Jongmo Kim[b], Albert Meroño-Peñuela[b], Valentina Presutti[a,*] and Elena Simperl[b]

[b]King's College London, London, United Kingdom
[a]Alma Mater Studiorum - University of Bologna, Bologna, Italy

ARTICLE INFO

Keywords:
knowledge engineering
knowledge graph
ontology development
competency question extraction

ABSTRACT

The process of developing ontologies – a formal, explicit specification of a shared conceptualisation – is addressed by well-known methodologies. As for any engineering development, its fundamental basis is the collection of requirements, which includes the elicitation of competency questions. Competency questions are defined through interacting with domain and application experts or by investigating existing datasets that may be used to populate the ontology i.e. its knowledge graph. The rise in popularity and accessibility of knowledge graphs provides an opportunity to support this phase with automatic tools. In this work, we explore the possibility of extracting competency questions from a knowledge graph. This reverses the traditional workflow in which knowledge graphs are built from ontologies, which in turn are engineered from competency questions. We describe in detail RevOnt, an approach that extracts and abstracts triples from a knowledge graph, generates questions based on triple verbalisations, and filters the resulting questions to yield a meaningful set of competency questions; the WDV dataset. This approach is implemented utilising the Wikidata knowledge graph as a use case, and contributes a set of core competency questions from 20 domains present in the WDV dataset. To evaluate RevOnt, we contribute a new dataset of manually-annotated high-quality competency questions, and compare the extracted competency questions by calculating their BLEU score against the human references. The results for the abstraction and question generation components of the approach show good to high quality. Meanwhile, the accuracy of the filtering component is above 86%, which is comparable to the state-of-the-art classifications.

1. Introduction

Knowledge engineering methodologies, especially for supporting ontology development, have been the subject of significant research in the Semantic Web field. Most methodologies (Peroni, 2017; Presutti, Daga, Gangemi and Blomqvist, 2009; Abdelghany, Darwish and Hefni, 2019; John, Shah and Stewart, 2018; Schekotihin, Rodler, Schmid, Horridge and Tudorache, 2018; Paschke and Schäfermeier, 2018) include a requirement elicitation phase and an ontology evaluation/testing phase, as fundamental in their processes. Requirement elicitation is the process of extracting and collecting requirements from domain and application experts, stakeholders, datasets, etc., whereas the evaluation phase deals with assessing the quality of the ontology. The basis of both these phases is the requirements, in the form of Competency Questions (CQs). Competency questions play an important role in the ontology development lifecycle, as they represent the ontology requirements and are criteria for the evaluation of the ontology (Bezerra, Freitas and Santana, 2013). Despite the key role of the CQs in the process, there is no systematic mechanism to elicit them. Moreover, even though there are several methodologies that place competency questions at the centre of the ontology evaluation (Presutti et al., 2009; Bezerra et al., 2013), there is a notable need for tool support for this phase of the process (Fernández-Izquierdo and García-Castro, 2022).

Generally, CQs are defined in a top-down approach, but the potential coming from existing knowledge graphs (e.g. Wikidata (Vrandečić and Krötzsch, 2014), DBPedia (Auer, Bizer, Kobilarov, Lehmann, Cyganiak and Ives, 2007)) provides an opportunity to design a bottom-up approach to extract competency questions. Such an approach, which we propose in this paper, consists of reverse engineering an ontology, i.e. defining a method for extracting competency questions from existing knowledge graphs, rather than doing so from experts or other sources. This proposition is founded on the observation that recently an increasing number of knowledge engineering projects (e.g. Polifonia[1], SPICE[2]) have involved the construction of knowledge graphs (KGs) from already existing ones; therefore, in many cases KGs are the starting point of knowledge engineering projects, rather than the end of the process. A direct application of the method is being able to query the knowledge graph of origin in order to understand its content. Modern knowledge graphs are built using a combination of human and automated actions (e.g. information extraction from text, data input from

*Corresponding author
fiorela.ciroku2@unibo.it (F. Ciroku); Via San Giuseppe 8, 50122, Florence, Italy (F. Ciroku); valentina.presutti@unibo.it (V. Presutti)
https://www.kcl.ac.uk/people/jacopo-de-berardinis (J.d. Berardinis); https://www.kcl.ac.uk/people/albert-merono-penuela-1 (A. Meroño-Peñuela); https://www.unibo.it/sitoweb/valentina.presutti/en (V. Presutti); https://www.kcl.ac.uk/people/elena-simperl (E. Simperl)
ORCID(s): 0000-0002-3885-6280 (F. Ciroku); 0000-0001-6770-1969 (J.d. Berardinis); 0000-0002-4984-1674 (J. Kim); 0000-0003-4646-5842 (A. Meroño-Peñuela); 0000-0002-9380-5160 (V. Presutti); 0000-0003-1722-947X (E. Simperl)

[1] https://polifonia-project.eu/
[2] https://spice-h2020.eu/

databases). Usually, these actions are not well-documented and derive an ontology that is not adequately structured (Piscopo and Simperl, 2018). Therefore, understanding what competency questions a knowledge graph answers can support the development of its ontology, the knowledge graph's evaluation, documentation, and reuse.

Another application of the method can be to support the knowledge engineering process. We hypothesise that the requirement elicitation and the evaluation phases can be enhanced, both in terms of procedures and rules, and in terms of the quality of the outcomes, with such a method; here we propose a method to address the first step of such an hypothesis, namely, to reliably extract the CQs to which an existing KG provides answers. Regarding the requirement elicitation, the data from already existing knowledge graphs, which have greatly proliferated in the last years Hogan, Blomqvist, Cochez, d'Amato, Melo, Gutierrez, Kirrane, Gayo, Navigli, Neumaier et al. (2021), can provide a different view of the domain of the ontology that is being constructed, complementary to the view provided by domain experts that are involved in the process or other sources. This additional information can improve the coverage of the domain, and in turn, of the ontology model. At the same time, the resulting competency questions can later be used as criteria for the evaluation of the ontology.

Motivated by the need for tool support for the knowledge engineering process, the opportunities that knowledge graphs offer in terms of application, and by leveraging recent advances in natural language processing, we raise the following research questions:

- **RQ1:** To what extent can competency questions be extracted directly from existing knowledge graphs?
- **RQ2:** Which quality features can be inferred from human-made CQs and to which extent are these preserved in those extracted from a knowledge graph?

1.1. Background

The process for engineering an ontology starts with the requirement elicitation phase, as established in many methodologies such as Presutti et al. (2009); Abdelghany et al. (2019); John et al. (2018); Schekotihin et al. (2018); Paschke and Schäfermeier (2018). An initial set of requirements is formalized and prioritized by ontology engineers. Based on the priority given to the competency questions, ontology engineers develop an ontology iteratively. The competency questions are simultaneously used for the testing of the ontology to ensure its quality. While the ontology is built and tested, the engineers construct the respective knowledge graph. A synthesis of the main steps in an ontology development process is displayed in Figure 1.

Based on our experience with eXtreme Design (XD) Presutti et al. (2009), requirement elicitation is an engaging task that includes continuous interactions with experts and data investigation. In-person or online meetings with domain experts generally need substantial planning and time for the ontology engineer to thoroughly grasp the domain.

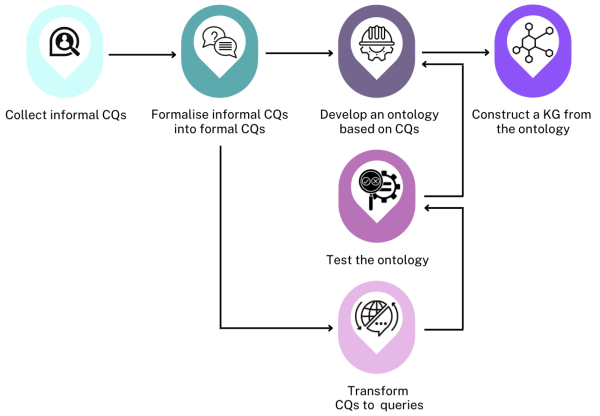

Figure 1: A synthesis of the main steps in an ontology development process

Meanwhile, the current tool support, like Google forms and GitHub templates, is inadequate to provide a consistent framework for gathering information. The process still needs time and effort from the ontology developer to analyse the information, structure it, and comprehend the input. The same techniques and constraints apply to extracting competency questions from data sets. It is critical to emphasise that competency questions must be of a specific quality to be utilised for the intended purpose. A competency question's fundamental quality check is that it must be translatable to a query that can validate the ontology. The transformation process from informal competency questions to formal competency questions lacks standardisation, is time-consuming, and requires a significant amount of effort.

An example of the requirement collection procedure based on eXtreme Design comes from the Polifonia project[3]. First, we asked domain experts to create stories. In the context of the project, a story is a template for collecting requirements which might include information about the persona, the goal, the scenario, competency questions, and resources, described in detail by de Berardinis, Carriero, Jain, Lazzari, Meroño-Peñuela, Poltronieri and Presutti (2023). Given that domain experts often are not knowledge engineers, the competency questions that we received required a considerable amount of manual work to be transformed into formal competency questions. For instance, an informal competency question was *"In which historical documents is there evidence of a musical composition?"*[4]. This competency question was followed up by additional interaction with the domain experts to understand what the concept of evidence means to them, which in turn was defined *"as any direct linguistic sign that refers to concepts (name, or part*

[3]The Polifonia project aims to support providers in revealing the content of value hidden in their digital objects, to support scholars in conducting research based on large, heterogeneous data sources and to enable citizens to access and understand Musical Heritage with all its complexity.

[4]The complete story is found in https://github.com/polifonia-project/stories

of it)". This information was used to reformulate the competency question and have a formal one as follows *"Which historical documents mention a musical composition?"*.

This example demonstrates that the process that an ontology engineer has to complete for manually eliciting a competency question, formalising it, and transforming it to a query for evaluating the ontology is inefficient and demands substantial manual effort and considerable interactions.

1.2. Our contribution

To support the aforementioned scenario, we leverage language models and investigate their use in the process of extracting competency questions from a knowledge graph. The main contributions of this article are threefold:

1. RevOnt, a framework for automatically extracting competency questions from a knowledge graph[5];
2. The WDV-CQ dataset, providing 1786 manually annotated competency questions and 1904 verbalisations from a subset of Wikidata triples, together with those automatically extracted by RevOnt;
3. An experimental evaluation of RevOnt, shedding light on the quality of the extracted CQs in relation to their corresponding human annotations.

RevOnt aims at reversing the ontology engineering process starting from the knowledge graph. The data from the KG is used to formulate questions by leveraging Natural Language Processing (NLP) models. The questions are evaluated and filtered with techniques described in Section 3. The filtering results in a set of core competency questions that describe the areas of the knowledge graph from which the triples were retrieved. These competency questions could be further used to e.g. generate respective SPARQL query templates, supporting the testing of the ontology. An overview of the approach is shown in Figure 2.

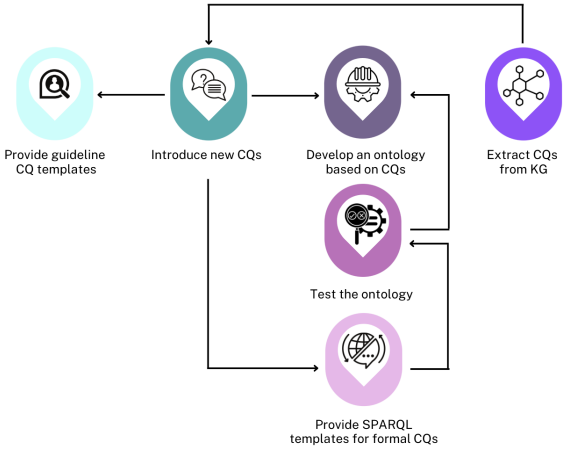

Figure 2: An overview of the RevOnt approach with an example use case for ontology testing. This article covers the stages of competency question extraction (and leaves "Provide SPARQL templates for formal CQs" as future work).

[5]Implementation of RevOnt and data to reproduce the experiments available at https://github.com/King-s-Knowledge-Graph-Lab/revont.

The knowledge graph we use to experiment and evaluate the method is Wikidata, which is a collaboratively-edited multilingual knowledge graph. Wikidata focuses on objects that represent any topic, concept, or object. Here, we chose Wikidata because: it is a popular knowledge graph; it reflects common knowledge on several domains; and it provides resources that help to evaluate the approach such as ID, label, description, and statements. The limitations of this use case remain in the fact that the knowledge graph is not domain-specific. An example data model in Wikidata is illustrated in Figure 3 for the triples related to Douglas Adams[6].

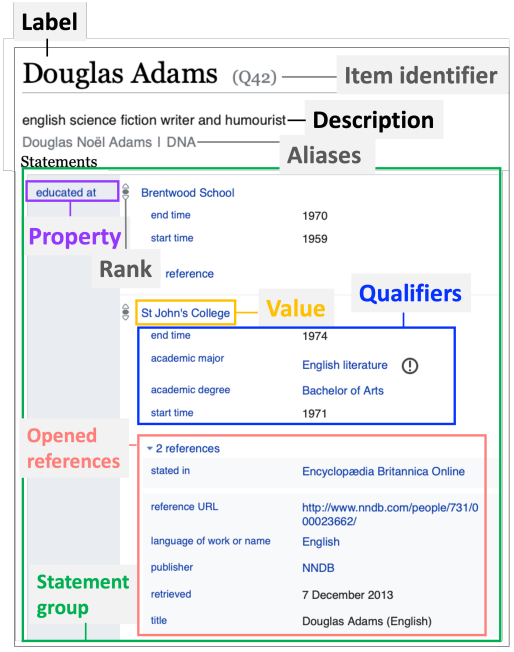

Figure 3: Datamodel of a Wikidata item. The datamodel provides key information for a Wikidata item such as identifier, label, description, aliases, statements, and references.

In the remainder of the article, Section 2 summarises related works regarding ontology development methodologies with a focus on the concept of competency questions and related work on relevant fields of research such as ontology learning and schema induction. Section 3 describes the approach that RevOnt uses to extract competency questions from a knowledge graph. Section 4 presents the experimental setup and the results of the evaluation. Finally, Sections 5 and 6 present the reflections of the work, provide indications for future work, and conclude the article.

2. Related work

The early works in **ontology engineering** surfaced at the beginning of the '90s with projects such as TOVE by Fox, Barbuceanu and Gruninger (1996), Ontolingua by Gruber (1992), among others, described in (Jones, Bench-Capon and Visser, 1998). Building upon these foundational projects, later efforts, including the Enterprise Model Approach by Dietz (2006) and METHONTOLOGY by Corcho,

[6]Picture derived from https://www.wikidata.org/wiki/Q16222597

Fernández-López, Gómez-Pérez and López-Cima (2005), emerged, drawing inspiration from and extending the pioneering works of the early '90s to the 2000s. These works, intending to define methodologies that are closer to engineering practices, are followed in the early 2000s by Sure, Erdmann, Angele, Staab, Studer and Wenke (2002) and Öhgren and Sandkuhl (2005). After a decade, ontology development methodologies that include cyclic processes, continuous integration and testing as (Peroni, 2017; Presutti et al., 2009; Abdelghany et al., 2019; John et al., 2018; Schekotihin et al., 2018; Paschke and Schäfermeier, 2018) emerge and become a shared practice among engineers. One of the common denominators between these works is the concept of competency questions, early defined in Grüninger and Fox (1995). Based on several works such as (Lenat and Guha, 1993; Fadel, Fox and Gruninger, 1994; Uschold and King, 1995; Uschold and Gruninger, 1996; Presutti et al., 2009; Gangemi and Presutti, 2009; Vrandečić, 2009; Ren, Parvizi, Mellish, Pan, van Deemter and Stevens, 2014), the definition of competency question can be reasonably summarised as "a typical query that an expert might want to submit to a knowledge base of its target domain, for a certain task"[7]. The role of a competency question is twofold: (1) to drive the modelling of the ontology by serving as a requirement, (2) to assist the evaluation of the ontology by being expressed as a query. In our research, we use these criteria to evaluate the quality of the generated CQs.

In consideration of the method and techniques that are used in our work, relevant related fields of research are ontology learning, and schema induction and discovery. **Ontology learning** is a multidisciplinary field that extracts terms, concepts, properties, and relationships from unstructured text using approaches from several disciplines such as knowledge representation, natural language processing, machine learning, etc. Surveys in ontology learning such as (Asim, Wasim, Khan, Mahmood and Abbasi, 2018), classify ontology learning techniques into three classes namely linguistic, statistical and logical. *Linguistic techniques* are based on language features and are commonly used for data preparation (speech tagging, parsing and lemmatisation) as well as various other ontology learning tasks such as knowledge extraction. Prime examples of such techniques are Text2Onto[8], CRCTOL[9]. *Statistical techniques* (C/NC value, contrastive analysis, clustering, co-occurrence analysis, term subsumption and ARM) rely entirely on statistics from the underlying corpus and overlook the underlying semantics. The majority of statistical approaches make substantial use of probabilities and are commonly utilised in early stages of ontology learning following linguistics preprocessing. Relevant tools that use statistical techniques are OntoGain (Frantzi, Ananiadou and Mima, 2000), OntoLearn(Kouagou, Heindorf, Demir and Ngonga Ngomo, 2022), ASIUM(Faure and Poibeau, 2000). Lastly, *Inductive logic programming* is a machine learning discipline that employs logic programming to generate hypotheses based on prior knowledge and a set of examples. Significant tools are Syndikate (Hahn and Romacker, 2001) and TextStorm[10].

With the rapid growth of RDF and KG data resources, numerous studies on ontology learning using RDF have been proposed. These studies can be categorised based on the approach adopted for utilising RDF in ontology learning. Some studies have leveraged RDF datasets to enhance the quality of ontology learning, given that ontologies may have limited instances and data values. Consequently, data-driven ontology learning techniques could easily encounter information scarcity issues. RDF datasets serve as a valuable resource for ontology learning due to their semi-structured nature and similarity to ontologies. Andrea and et al. proposed a possibilistic approach for testing OWL (Web Ontology Language) axioms against RDF facts, specifically for testing the 'SubClassOf' relationship (Tettamanzi, Faron-Zucker and Gandon, 2014). The ADOL was designed for automatic domain ontology learning from textual data with RDFs in general-purpose Knowledge Graphs. It utilises RDF data to create semantic similarity with the ontology by comparing extracted words and relations to the RDF set (Chen and Gu, 2021). Conversely, Mid-Ontology has been introduced to automatically construct a simplified ontology from the RDF set of Linked Open Cloud (LOD) for the integration of different ontology schema (Zhao and Ichise, 2012).

Lastly, surveys such as (Ji, Qi, Gao and Wu, 2019) and (Kellou-Menouer, Kardoulakis, Troullinou, Kedad, Plexousakis and Kondylakis, 2022) present an overview of the state-of-the-art in **schema induction** and discovery, a research field dealing with the extraction or discovery of semantic schemas from unstructured or semi-structured data (Gick and Holyoak, 1983). The research is motivated by the fact that the data in the semantic web, either expressed in RDF or JSON, are not based on a predefined schema. The most widely used techniques in the schema discovery approaches are machine learning (classification, clustering and frequent pattern mining) and formal techniques (Formal Concept Analysis, bisimulation). The surveys discuss valuable works regarding implicit and explicit schema discovery approaches by taking into consideration the target problem, techniques, features, input, output, and quality aspects.

Relevant surveys such as (Pouriyeh, Allahyari, Kochut and Arabnia, 2018) and (Čebirić, Goasdoué, Kondylakis, Kotzinos, Manolescu, Troullinou and Zneika, 2019) describe **ontology summarisation** approaches that use centrality metrics (e.g. PageRank) to identify the most informative concepts/nodes or extract important subgraphs to facilitate query-testing for verifying requirements against accessible data. In contrast, a recent research related to the extraction of Common Conceptual Components[11] from multiple Ontologies using Ontology Design Patterns[12] is presented in (Asprino, Carriero and Presutti, 2021). The

[7] Formulated by Gangemi and Presutti (2009)

[8] http://neon-toolkit.org/wiki/1.x/Text2Onto.html

[9] http://nlp.cs.berkeley.edu/

[10] https://dwijottam-dutta.github.io/TextStorm/about/about.html

[11] A conceptual component (CC) is a complex (cognitive) relational structure that a designer implements in an ontology by using classes, properties, axioms, etc.

[12] http://ontologydesignpatterns.org/wiki/Main_Page

authors present a method that employs a non-extractive method to assist in the comprehension and comparison of different ontologies. Starting with a corpus of ontologies, it uses community detection, word sense disambiguation, frame recognition, and clustering to automatically produce a catalogue of conceptual components and observable ontology design patterns. Further, in (Nuzzolese, Gangemi, Presutti and Ciancarini, 2011), the authors present a study for discovering Encyclopedic Knowledge Patterns (EKP)[13] from Wikipedia[14] page links. The patterns, according to the authors, may be used as lenses for exploring DBpedia[15] or for developing new ontologies that inherit the data and textual grounding offered by DBpedia and Wikipedia. Data linking can also benefit from EKPs by modulating the datasets to be linked.

Our research contributes to the field of knowledge engineering by enhancing the requirement elicitation, knowledge graph reuse, and ontology testing processes. The competency questions extracted with our approach can be used to develop and test an ontology. Furthermore, extracting competency questions from a KG can indirectly aid ontology reuse and ontology learning tasks, as this approach retrieves terms and relations through the use of natural language processing models and machine learning. For instance, a wealth of competency questions could assist in identifying the optimal and most suitable ontology by testing and validating the generated ontology throughout the ontology learning process. Moreover, RevOnt supports the extraction of a schema of a KG through its abstraction stage, where the natural language verbalisation of a triple is abstracted from the instance level to the class level.

3. The RevOnt approach

This section presents RevOnt, a framework for extracting competency questions from knowledge graphs. As depicted in Figure 4, this is divided into three stages: 1) verbalisation abstraction, 2) question generation, and 3) question filtering.

Verbalising a Knowledge Graph consists in generating grammatically correct natural language starting from interconnected triple-based claims – formed of subject, predicate, and object, Amaral (2022). Here, the purpose of the *Verbalisation Abstraction* stage is to transform the verbalisation from the instance level to the class level. For example, given the triple verbalisation *"Michael Jackson is a member of the Michael Jackson discography."*, without the abstraction the generated questions are: *"Who is a member of the Michael Jackson discography?"* and *"What is Michael Jackson a member of?"*. These questions are not competency questions because they ask for specific instances, and not classes or properties. Thus, there is a need to abstract the verbalisation into a more general form. To complete the task, the *Verbalisation Abstraction* stage begins with a dataset, which contains the verbalisations of triples from a knowledge graph, as an input. The dataset should include data about the subject, predicate, and object of the triple, descriptions of the instances, IDs that align them to a knowledge graph; and most importantly, the verbalisation of the triple. This information is necessary for the selection of the class in which the subject and the object of the triple is an instance of or a subclass of.

The second stage of the framework is *Question generation*. The aim of this stage is to generate three questions for each triple verbalisation. This choice of design is made to understand the types of competency questions that are generated when the method asks the model about different parts of the verbalisation sentence. Essentially, this stage generates questions based on the abstraction of the triple verbalisation (e.g. from "A music artist is a member of their own discography" to "Who is a member of a discography?"). Part of this stage is a grammar check task that corrects errors that are found in the questions to improve their quality.

Finally, the *Question filtering* stage deals with the quality check and reduction of CQs generated in the previous step. The main task of this stage is *question reduction*, to filter semantically equivalent competency questions (those entailing the same ontology requirements), and identifying meaningful groups based on their similarity. This stage is needed to identify a comprehensive yet minimal set of core competency questions from all the generated CQs. In addition, we conceptualise a further step that aims to map competency questions to their corresponding SPARQL queries; hence providing an additional validation step (a competency question that can been matched to a SPARQL query has higher chances to be correctly formulated). Nonetheless, here we focus on the extraction of CQs and leave the implementation of this last feature as future work.

Table 1 shows a list of NLP models, modules, datasets and services used for the implementation of the framework, their category and the stage where they are used. Our choices are motivated in the corresponding sections below.

3.1. The WDV dataset

There are several datasets that provide verbalisations of data such as the T-REx[16] and WDV dataset[17] for Wikidata entries, WebNLG[18] for DBPedia[19] entries, NYT-FB by Mintz, Bills, Snow and Jurafsky (2009) and FB15K-237 by Toutanova and Chen (2015) for Freebase[20], and so on.

A first implementation of RevOnt was achieved for Wikidata, by leveraging the WDV dataset (Amaral, 2022). This dataset provides verbalisations of Wikidata claims and it contains 7.6K unique triples. According to Amaral, Rodrigues and Simperl (2022), WDV has considerably more entity types and predicates than comparable datasets, and it is intended to serve as a benchmark dataset for data verbalisation models used on Wikidata. WDV enables a tight coupling between single claims and text by directly connecting a triple-based claim to a natural language phrase.

[13] EKPs are Knowledge Patterns that are grounded in encyclopedic knowledge expressed as linked data and as natural language text.

[14] https://en.wikipedia.org/wiki/Main_Page

[15] https://www.dbpedia.org/

[16] https://hadyelsahar.github.io/t-rex/

[17] https://github.com/gabrielmaia7/WDV

[18] https://gitlab.com/shimorina/webnlg-dataset

[19] https://www.dbpedia.org/

[20] https://developers.google.com/freebase

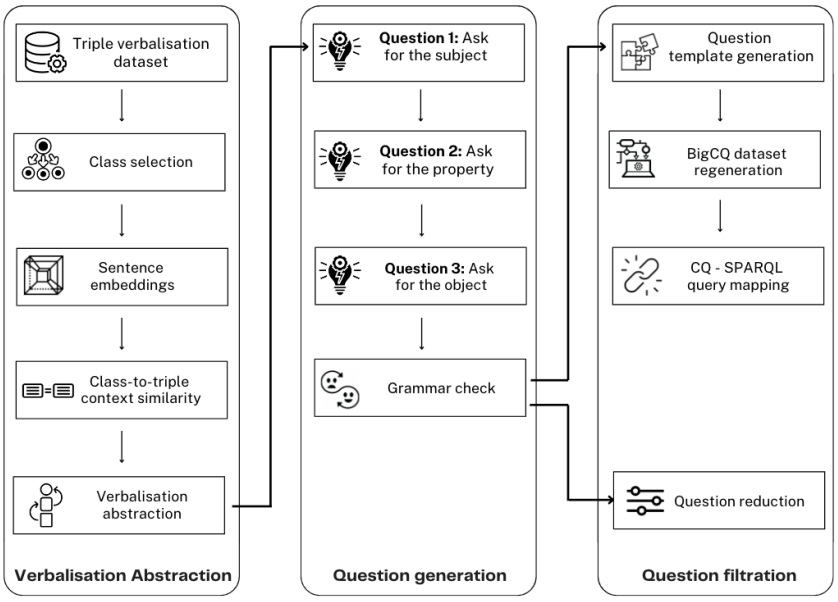

Figure 4: An overview of the RevOnt framework. The first stage, Verbalisation Abstraction, generates the abstraction of a triple verbalisation. The abstraction is used as input in the second stage, Question Generation, to generate three questions per triple and perform a grammar check.

Table 1
A list of the language models, modules, datasets and services used in the RevOnt framework

Model	Category	Stage
WDV	Dataset	Input
MiniLM	LM	Verbalisation abstraction
Wikidata query service	Service	Verbalisation abstraction
T5	LM	Question generation
T5 Grammar Correction	LM	Question generation
SBERT	LM	Question filtering
UMAP	Dim. reduction	Question filtering
HDBSCAN	Clustering	Question filtering

The WDV dataset contains verbalisation of claims from 20 different themes (or domains) from the most common like artist, sport teams, university to celestial body, chemical compounds, taxon, etc. The diversity of triple verbalisations in the dataset contributes to interesting results and imposes challenges as well. An example of a claim verbalisation from the *Airpot* theme is shown in Figure 5. The described claim verbalisation is structured in the form of key-value pairs. It provides RDF triple-related information including *"subject_label"* and *"subject_desc"*, as well as claim-related information including *"theme_label"* and *"verbalisation"*.

A claim verbalisation from the WDV dataset


```
{"claim_id": "Q20451714$49CB1F7A-FC51-41CF-81EA-3EC1EC81AB5D",
"rank": "normal",
"subject_id": "Q20451714",
"property_id": "P366",
"subject_label": "Livramento Do Brumado Airport",
"property_label": "use",
"object_label": "general aviation",
"subject_dec": "airport in Brazil",
"property_desc": "main use of the subject (includes current and former usage)",
"object_desc": "civil use of aircraft excluding commercial transportation",
"subject_alias": ["SNLB"],
"property_alias": ["function", "role", "mission", "purpose", "utility", "used for",
                   "used in", "usage"],
"object_alias": ["GA", "G/A", "genav", "civil aviation"],
"object_datatype": "wikibase-item",
"object": {"value": {"entity-type": "item", "numeric-id": 1571929, "id": "Q1571929"},
                     "type": "wikibase-entityid"},
"theme_root_class_id": "Q12480784",
"theme_label": "Airport",
"verbalisation": "Livramento Do Brumado Airport is used by general aviation.",
"verbalisation_unk_replaced": "Livramento Do Brumado Airport is used by general aviation.",
"sampling_weight": 2.642857143,
"annotations": {
"fluency_scores": [5, 5, 2, 4, 3],
"fluency_mean": 3.8,
"fluency_median": 4.0,
"adequacy_scores": [1, 0, 0, 0, 0],
"adequacy_majority_voted": 0.0,
"adequacy_percentage": 0.8}
```


A depiction of triple-related key & value from a claim

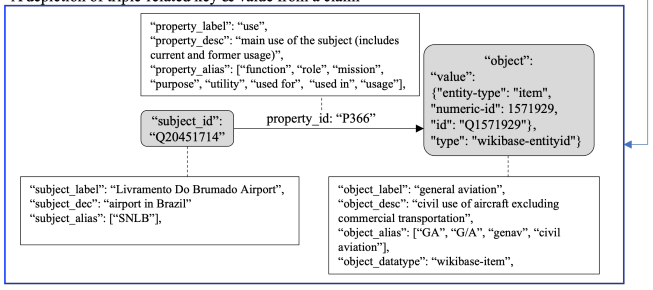

Figure 5: A claim verbalisation from the WDV dataset. For dataset provides the ID, rank, theme, and verbalisation of the claim. There are present also the label, description, aliases and ID of the subject, property and object of the triple.

3.2. Verbalisation abstraction

The first stage of the RevOnt framework is the *Verbalization Abstraction*. Its role is to generate an abstraction of a triple verbalisation. The triple verbalisation of Knowledge Graphs refers to converting RDF triples into natural language text by utilising their components and relationships. To perform the abstraction, it is necessary to recognise and

categorise the entities present in the verbalisation. The initial intuition for this task was to use a Named Entity Recognition (NER) model. We experimented with language models such as Camembert-ner[21], Camembert-base-multilingual-cased-ner-hrl[22], Ner-english-large[23], Bio-Ner[24] and the SpaCy library. The performance of each of these models was not satisfactory for 50% of the themes of the dataset.

Therefore, we designed a novel method to abstract the subject and object of a triple to the most similar-to-context Wikidata class. The procedure is shown in Algorithm 1.

Algorithm 1 Abstraction of a Wikidata triple

```
procedure TRIPLE VERBALISATION ABSTRACTION
    Create the sentence embedding of the subject description
    Create the sentence embedding of the object description
    Retrieve the Wikidata classes of the subject and object of the triple
    for each class do
        Get the synsets
        for each synset do
            Get the synset definition
            Create the sentence embedding of the synset definition
            Calculate the cosine similarity between the synset definition embedding and the subject/object descriptions' embeddings
    return Most similar synset
```

For each triple, the method extracts the sentence embeddings of the descriptions of the subject and the object using the MiniLM language model[25]. MiniLM is a sentence-level transformer model (Reimers and Gurevych, 2019) that maps sentences to a 384-dimensional dense vector space. By leveraging a pre-trained language model[26], MiniLM is fine-tuned on a number of datasets[27] using a contrastive objective. Intuitively, a contrastive loss is used to minimise the cosine similarity between similar sentence pairs, while maximising that of other sentences in the same batch.

In the next step, RevOnt selects the English label of Wikidata classes where an entity is an instance of or a subclass of. The query that we use for selecting this information from the knowledge graph is shown below. For this task, we use the Wikidata Query Service[28]. Wikidata Query Service is an implementation of a SPARQL server, based on Blazegraph[29] engine. This is used to service queries for Wikidata and other datasets.

[21] https://huggingface.co/Jean-Baptiste/camembert-ner
[22] https://huggingface.co/Davlan/bert-base-multilingual-cased-ner-hrl
[23] https://huggingface.co/flair/ner-english-large
[24] https://github.com/librairy/bio-ner
[25] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[26] https://huggingface.co/nreimers/MiniLM-L6-H384-uncased
[27] https://www.sbert.net/docs/pretrained_models.html
[28] https://query.wikidata.org/
[29] https://github.com/blazegraph/database/wiki/Main_Page

```
SELECT DISTINCT ?cLabel
WHERE {{
    wd:{id} wdt:P31/wdt:P279? ?c .
    ?c rdfs:label ?cLabel .
FILTER(LANG(?cLabel) = "en") }}
```

For instance, for the triple verbalisation *"Michael Jackson is a member of the Michael Jackson discography."*, according to the dataset shown in Figure 5, the subject is *Michael Jackson* and the object is *Michael Jackson discography*. The respective descriptions are *American recording artist; singer and songwriter (1958-2009)* and *Wikimedia artist discography*. RevOnt extracts the sentence embeddings of the subject and object description and queries the Wikidata KG to select the classes of the subject and object of the triple. The result for *subject* is shown in Example 3.1.

Example 3.1. *The Wikidata classes for the subject "Michael Jackson"*

```
['human', 'natural person', 'omnivore', 'person',
'mammal', 'Homo sapiens']
```

Once classes are retrieved, RevOnt gets the corresponding Wordnet synsets for each class. Wordnet is a large electronic lexical database for English (Fellbaum, 2010). Cognitive synonyms (synsets) are groups of nouns, verbs, adjectives, and adverbs that each communicate a separate notion. Synsets are linked together via conceptual-semantic and linguistic relationships. In Example 3.2 we present the respective synsets of the classes where *"Michael Jackson* is an *instance of*/*subclass of*. For each synset, it retrieves the definition and computes its sentence embeddings by using the MiniLM model.

Example 3.2. *The synsets of the Wikidata classes*

```
{[Synset('homo.n.02')],
[],
[Synset('omnivore.n.01'), Synset('omnivore.n.02')],
[Synset('person.n.01'), Synset('person.n.02'),
Synset('person.n.03')],  [Synset('mammal.n.01')],
[]}
```

In the last step, RevOnt calculates the cosine similarity between the embeddings of the definition of the synsets of each class with the embeddings of the description of the subject or object. The cosine similarity is calculated with the help of the Natural Language Toolkit (NLTK). NLTK is a suite of Python modules providing many NLP data types, processing tasks, corpus samples, and readers, together with animated algorithms, tutorials, and problem sets Loper and Bird (2002). For the example shown above, the Wikidata class that is the most similar to the description of the subject *Michael Jackson* is *human*. As for the object of the triple, the Wikidata class is *discography*.

The values that the algorithm returns populate a Python dictionary that is used for the abstraction task. More specifically, the subjects/objects and their corresponding most similar-to-description class are added to a dictionary, as shown in Example 3.3.

Example 3.3. *The pattern dictionary*

```
{'Michael Jackson': 'human',
'Michael Jackson Discography': 'discography'}
```

This dictionary is used to replace the subject and the object in a triple with the respective class. To continue on the same example above, the result of the verbalisation abstraction is presented in Example 3.4. This concludes the first stage of the framework, and the abstracted verbalisations are passed on to the second stage, Question generation.

Example 3.4. *The abstraction of the verbalisation*

```
Verbalisation: Michael Jackson is a member of the
               Michael Jackson discography.
Abstraction:   Human is a member of the discography.
```

3.3. Question generation

After the abstraction of the verbalisation, the next stage of the approach is to generate questions. Given a triple verbalisation where entities have been generalised, the goal is to automatically generate a set of questions addressing the content of the verbalisation. For this task, we have used the Text-to-Text Transfer Transformer (T5) (Roberts and Raffel, 2020) fine-tuned on the Stanford Question Answering Dataset (SQuAD) (Rajpurkar, Zhang, Lopyrev and Liang, 2016) for question generation. This was achieved by prepending the answer to the context. The specific T5 instance we used is labelled as `t5-base-finetuned-question-generation-ap` [30]. Notably, Romero (2021) demonstrated that this model can be handle generative tasks such as abstractive summarisation, classification tasks such as natural language inference, and even regression tasks.

To generate questions, the model requires as input a context (sentence) and an answer. When an answer is not provided, the model will generate a question that is answered by the object of the sentence. For each triple, we have provided the model with the abstraction of the verbalisation as the context, and three answers: the class of subject, the property, and the class of the object. In most observed cases, when the class of subject and the class of the object are the answers, the questions that are generated are inverse in relation to each other; hence they may provide complementary requirements for the triple. As for the cases when the property is the answer, the questions that are generated are often the same as the ones when the answer is the class of the object, but there are also many cases when the questions are quite interesting and not so straightforward. This behaviour of the model can be explained with the distance between the triple and the verbalisation. As noticed in the example below, while the property of the triple is `discography`, this property is not present in the verbalisation. This is the case for the majority of the verbalisations present in the dataset. In Example 3.5, we show the results of the Question Generation stage given the abstraction from Example 3.4.

Example 3.5. *Question generation*

```
Context: Human is a member of the discography.
Answer 1: human
Question 1: What is a member of the discography?
Answer 2: discography
Question 2: What is the relationship between
            a human and a discography?
Answer 3: discography
Question 3: What is a human a member of?
```

Once the questions are generated, RevOnt performs a grammar check to detect and fix errors. This task is performed by the T5 Grammar Correction model[31]. Trivially, the model generates a revised version of the given text with the goal of addressing grammatical errors. It is trained with Happy Transformer[32] using a the JFLEG dataset (Napoles, Sakaguchi and Tetreault, 2017).

3.4. Question Filtering

In this section, we cover the third and last stage of RevOnt, which consists in filtering the questions that are generated from the second stage. This is achieved through *question reduction*, which is divided into two parts: paraphrase detection (to remove equivalent questions) and question clustering (to identify groups of similar requirements).

As some of the questions generated in the previous steps may be redundant, or show negligible semantic variations that are of little interest to ontology engineers, we filter out questions based on their similarity. This has two potential benefits: (i) it mitigates the noise and the artifacts introduced in the previous steps (e.g. questions with unclear or inconsistent semantics); (ii) it reduces the number of competency questions that will be presented to the ontology engineers. Depending on the desired level of filtering, here we provide two methods to identify groups of related questions.

- *Similarity grouping*, which aims at detecting groups (or clusters) of semantically similar – yet not necessarily identical, questions.
- *Paraphrase detection*, a specialisation of the former task focused on detecting questions with the same exact meaning (e.g. "Is Batman a friend of Robin?" and "Is Batman Robin's friend?").

If the goal is to retain the largest number of questions, for example, because the ontology should model the domain at a granular level, the latter filtering method is more indicated. Instead, similarity grouping operates a more drastic reduction based on the relatedness of questions; which is also useful to get a high-level overview of the domain of interest.

Both our methods use sentence-level embeddings, rather than word-level representations, meaning that a sentence (in our case, a question) is mapped to a feature vector of a fixed size. The latter can be considered as a point in a highly dimensional space providing a numerical summary of the sentence's meaning. In particular, here we leverage

[30] https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap

[31] https://huggingface.co/vennify/t5-base-grammar-correction

[32] https://github.com/EricFillion/happy-transformer

Sentence-BERT (Reimers and Gurevych, 2019), an adaptation of BERT using Siamese and triplet network structures to derive semantically meaningful sentence embeddings.

Similarity grouping. The identification of questions with similar meaning is achieved via clustering of their embeddings – an application of unsupervised learning.

Following the projection of questions into the embedding space, dimensionality reduction is first applied using Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) (McInnes, Healy and Melville, 2018). This step is meant to preserve the consistency of distance measures over the embedding space due to the curse of dimensionality (Aggarwal, Hinneburg and Keim, 2001).

In general NLP tasks, sentences are encoded as numerical vectors to accurately measure the similarity among them in a distributional space. The size of these vectors, known as the dimension, determines a sort of resolution of the similarity measure. However, an excessively high dimension of embedding vectors can negate the similarity measure due to the curse of dimensionality. Thus, many Machine Leaning (ML) processes typically perform a dimensionality reduction step after vector creation to mitigate the curse of dimensionality, which could adversely affect the ability of ML algorithms to distinguish among samples.

In addition, compared to other dimensionality reduction approaches, UMAP is effective for both visualisation and is more effective at preserving the global structure of the data (e.g. t-SNE is often suitable for visualisation only).

Similar questions are then detected using Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) (McInnes, Healy and Astels, 2017) on the dimensionally-reduced representations. The choice of this algorithm is motivated by the fewer hyper-parameters, its robustness to noise, and the ability to deal with variable-density clusters (e.g. competency questions forming groups of heterogeneous size). Most importantly, as a density-based method, it does not require specifying the number of clusters a priori, and does not make any assumption on the shape of clusters. This is particularly suitable for finding groups of competency questions that are automatically extracted in the previous steps, as no assumptions can be made on the number and shape of their clusters. Notably, the same approach was adopted in IDEA (de Berardinis et al., 2023) to cluster manually curated competency questions, and support participatory workflows for their refinement.

An example of competency question clustering is described in Figure 7. This figure depicts the clustering results of competency questions under the 'WrittenWork' theme following UMAP reduction. The grey cluster encompasses simpler competency questions related to object or subject identification in triples, while the yellow cluster is related to querying numerical values of birthdays for specific characters. The analysis of grey clusters reveals structural redundancies in competency questions. For example, the questions "What is the hairstyle of a fictional character?" and "What is the pseudonym of a manga character?" within the grey cluster exhibit a similar structure. Through similarity grouping based on clustering, redundancy in competency questions with similar structures can be reduced.

Paraphrase detection. Alternatively, if the goal is to filter out only those questions that have the same meaning – thereby avoiding duplicated entries, we look at paraphrase detection. One approach is to define a threshold on the cosine similarity of two sentence embeddings to deem them as semantically equivalent (e.g. considering two sentences as equivalent if they exceed a threshold t_{sim}). However, as demonstrated in Figure 7 for $t_{sim} = 0.8$ (a typical similarity threshold in NLP), this is not reliable for paraphrases, as it introduces a large number of false positive and negatives.

We address the detection of equivalent questions via transfer learning. In this case, given two questions, their sentence-level embeddings are fed to a feed-forward neural network for *paraphrase detection* – a binary classification task. The artificial neural network (ANN) is thus trained to predict whether two sentences are semantically the same (positive pairs, $y = 1$) or not (negative pairs, $y = 0$).

Despite the availability of datasets for paraphrase detection, which have been extensively used for representation learning, we did not find a pre-trained model for question-based paraphrase detection. Therefore, we implemented a deep neural network that leverages sentence-level embeddings of questions to classify them as equivalent or different. The architecture is illustrated in Figure 8. First, the sentence-level (SBERT) embeddings of two questions are passed to linear projection layers with shared parameters. This is expected to fine-tune the embeddings on the question domain, without altering their dimension. An early fusion mechanism is then used to combine the 2 projected embeddings vectors through Hadamard product (element-wise product). The resulting vector, which still preserves the same SBERT embedding dimension, is then passed to 3 blocks of: fully-connected layers, followed by batch normalisation (Ioffe and Szegedy, 2015), ReLU activations, and dropout (Srivastava, Hinton, Krizhevsky, Sutskever and Salakhutdinov, 2014). In the final layer, a sigmoid layer is used for binary classification. By sampling from the learned Binomial distribution, the model can then be used to classify whether two questions (given their embeddings) are equivalent or not.

4. Evaluation

This section describes the end-to-end evaluation of RevOnt, which encompasses its core components: verbalisation abstraction, question generalisation, and question filtering. To systematically evaluate the output of RevOnt at different stages, we assembled a dataset of manually curated *verbalisations* and *competency questions* that were annotated from a subset of WDV (Section 4.1). The resulting annotations were then used as ground truth and compared to the RevOnt generations using BLEU – a metric for automatically evaluating machine-translated text.

The BLEU score is a number between zero and one that measures the similarity of the machine-translated text to a set of high-quality reference translations Papineni, Roukos,

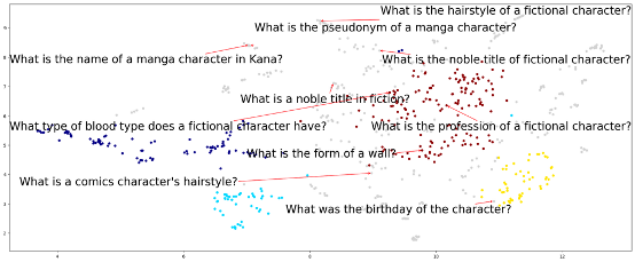

Figure 6: Example of competency question clustering with annotations linking questions to their respective clusters for the 'WrittenWork' theme.

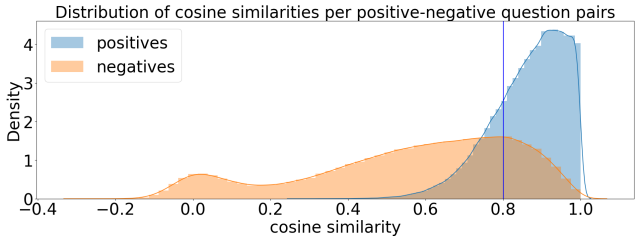

Figure 7: The Distribution of cosine similarities per positive (blue) and negative (orange) question pairs. The vertical blue line denotes a similarity threshold (0.8) commonly used to consider 2 sentences/embeddings as highly related.

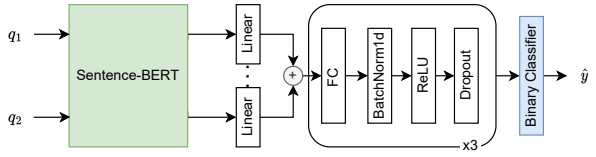

Figure 8: Architecture of the ANN for paraphrase detection. Dotted lines denote parameter-sharing, the plus symbol stands for additive feature fusion, and the rounded box repeats the same stack of layers for 3 times beforebinary classification.

Ward and Zhu (2002). The annotations serve as grams, each with a weight of 0.5. The number of reference human translations influences the results. Usually, more references result in better and more accurate scores. According to (Lavie, 2010) scores over 0.3 generally reflect understandable translations and scores over 0.5 reflect high quality translations (see Table 2 for an interpretation of BLEU scores). Intuitively, this allows to evaluate RevOnt's verbalisations and competency questions as alternative formulations (or valid translations) of their corresponding human annotations.

To complement the BLEU-based evaluation, we also compared syntactic properties of RevOnt-generated and human-annotated competency questions, and performed statistical tests to evaluate their significance (Section 4.3).

4.1. Dataset collection

To manually annotate verbalisation and competency questions from the WDV dataset (c.f. Section 3.1), we have collected data from $N = 15$ participants with an engineering background and familiar with ontology development. The participants are occupied mostly as ontology engineers,

Table 2
Interpretation of BLEU scores

BLEU Score	Interpretation
< 0.10	Almost useless
0.10 - 0.19	Hard to get the gist
0.20 - 0.29	The gist is clear, but has significant grammatical errors
0.30 - 0.40	Understandable to good translations
0.40 - 0.50	High quality translations
0.50 - 0.60	Very high quality, adequate, and fluent translations
> 0.60	Quality often better than human

machine learning engineers, and data scientists. Their selection was primarily motivated by their familiarity with ontology development, especially the concept of competency questions and the participants' ability to formulate them as engineering requirements. Their task consisted in manually reproducing the stages of verbalisation abstraction and question generation. In total, we created 20 forms[33] (one per theme), with a varying number of properties. The properties, which were manually reproduced, were generated using 7.6K unique triples from WDV. The number of properties depends on the variety of triple verbalisations pertaining to a specific theme. Each theme falls within a domain encompassing the 20 most common types on Wikidata, ranging from written works, chemical compounds, and artists, to monuments, sport, and transportation (c.f. Table 3). Although all the WDV themes stem from Wikidata, these categories cover different domains and provide a reasonable level of diversity in terms of ontological requirements.

Given the nature of the tasks, no personal information was collected from the participants and minimal risk clearance was granted from the Research Ethics of King's College London with registration number MRSP-22/23-34232.

First, we introduced the participants to the theme and the data that they need to complete the tasks. For each task, we provide the triple verbalisation, the subject, the subject description, the object, and the object description. The illustration used to describe the data is shown in Figure 9.

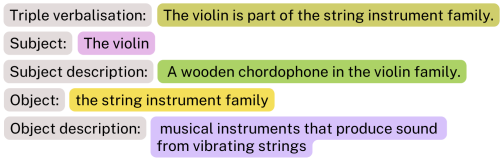

Figure 9: An illustration of example data including the triple verbalisation, the labels and descriptions of the subject and object of the triple. The example serves to explain the data needed to perform the tasks.

Next, we used examples to describe the two tasks that they were required to carry out. The first task is to abstract

[33]The forms and the full responses can be found here https://drive.google.com/drive/folders/1M7LCmqw4dc33U73GTauec02h3JNnNxv4

Table 3
An overview of the number of properties based on themes. The number of properties corresponds to the number of distinct properties of the triples of theme. The Property-Triple Ratio represents the proportion of properties relative to the triples within a specific theme.

Theme	Properties	Property-Triple Ratio
Airport	27	7.06%
Artist	65	16.92%
Astronaut	57	16.23%
Athlete	53	13.76%
Building	67	17.4%
Celestial body	25	6.49%
Chemical compound	33	8.61%
City	72	18.79%
Comics character	79	21.01%
Food	64	17.39%
Mean of transportation	58	15.42%
Monument	62	16.31%
Mountain	23	5.98%
Painting	29	7.53%
Politician	56	14.54%
Sports team	49	12.79%
Street	21	5.46%
Taxon	27	7.01%
University	62	16.4%
Written work	21	5.45%
Total	950	

a triple verbalisation given the information provided, shown in Figure 10. While the second task is to formulate questions based on the abstraction that they have created. Participants were asked to formulate three questions, where the answers to the questions would be 1) the subject of the abstraction, 2) the property, and 3) the object of the abstraction. The tasks are illustrated in Figure 11.

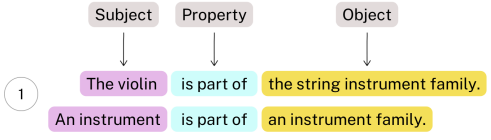

Figure 10: Example illustration for Task 1, describing the abstraction of a verbalisation using the given data. The abstraction is completed by generalising the subject and the object of the triple, and not the property.

We arranged for each theme to be covered by two participants. In total, we have gathered 40 responses from the forms, containing approximately 1.9k annotations. Meanwhile, the WDV dataset has on average 380 triple verbalisations per theme. The coverage of each theme with annotations can be found in Table 3. We release our dataset as **WDV-CQ** and provide both the human-annotated and the RevOnt-generated verbalisations and competency questions as two distinct subsets: WDV-CQ-HA and WDV-CQ-RO, respectively. WDV-CQ provides 1786 manually annotated

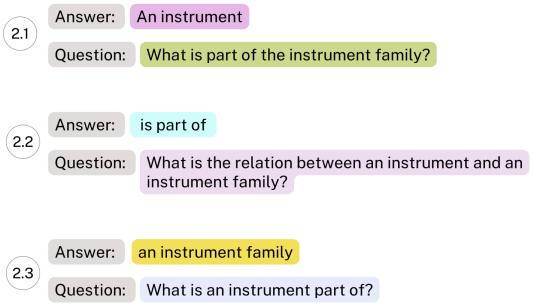

Figure 11: Example illustration for Task 2, describing the generation of three questions when the answer is provided (either the subject, the property, or the object of a triple).

competency questions and 1904 verbalisation abstractions; The dataset can be accessed from Zenodo[34] and is available under the Attribution 4.0 International (`CC-BY 4.0`) license.

4.2. Experimental Results

In the following subsections, we report the results for each of the stages of RevOnt and summarise the results for the whole system. To interpret the distribution of the BLEU scores for the Verbalisation Abstraction and the Question Generation stage, we provide box and whisker plots.

4.2.1. Verbalisation Abstraction evaluation

The distribution of the BLEU scores for the Verbalisation Abstraction stage is presented in Figure 12. As seen from the plot, the median of the scores is 0.41, which is interpreted as a high quality translation. The 75th percentile is 0.55 and the 25th percentile is 0,3. Meanwhile, the highest and the lowest data points are respectively 1 and 0. These results mean that 75% of the abstractions generated from RevOnt have a good to high quality.

During a manual thorough examination of the abstractions generated by RevOnt, that are included in the experiment, and those created by the participants we noticed that the main difference between them is that the abstraction generated by RevOnt is more general. In most cases, the system abstracts the subject and the object of the verbalisation into more general classes than a user. As shown in Example 4.1, the system abstracts Michael Jackson as a human, while a user abstracts it as a singer.

Example 4.1. *The difference between the abstraction generated by RevOnt and by the users*

```
Verbalisation:     Michael Jackson is a member of the
                   Michael Jackson discography.
RevOnt:            Human is a member of the discography.
User annotation: A singer is a member of the discography.
```

4.2.2. Question Generation evaluation

Figure 13 report the distribution of BLEU scores of RevOnt-generated CQs. The median is 0.3, which is close to being interpreted as a "good translation". The 75th percentile

[34] https://zenodo.org/records/10370725

is 0.47 and the 25th percentile is 0.21. The highest and the lowest data points are 0.81 and 0.1 respectively.

The non-satisfactory evaluation of this stage is explained in Figure 14. In this plot, we have evaluated individually the generation of each type of question. The red plot represents the generation of the questions when the answer is the subject of the abstraction, the yellow plot when the answer is the property of the triple, and the green plot when the answer is the object of the abstraction.

The generation of the question when the answer is the property of the triple performs poorly. This result backs the argument made in Section 3 where we hypothesise that the questions that T5 generates in this case are generally identical to the ones that it generates when the answer is the object of the abstraction. In Example 4.2, we show the difference between the question that is generated by RevOnt and by the user when the answer is the property of the triple.

Example 4.2. *The difference between the questions generated by RevOnt and the users when the answer is the property of the triple.*

```
Verbalisation:    Michael Jackson is a member of the
                  Michael Jackson discography.
Answer:           discography/member of
RevOnt:           What is a human a member of?
User annotation:  Which is the relation between a singer
                  and the discography?
```

Comparing the scenarios when the subject and the object of the abstraction are the answer, we observe that the latter is more accurate with a median of 0.42, which is considered a high quality translation. The 75th percentile is 0.61, the 25th percentile 0.28, and the highest point is 1.0 and the lowest is 0.08. 75% of the scores for this category of questions is of good to high quality.

4.2.3. RevOnt 2-stage evaluation

The results of this evaluation provide further insights into the Question Generation stage. We expect a granular evaluation of the quality of each type of question to drive the refinement of the RevOnt framework. As mentioned earlier, the design choice to include all three types of questions is purely experimentation and to support future development.

Figure 15 reports a Box and Whisker plot of BLEU scores (generated against manually annotated questions) when RevOnt considers the *object* of the abstraction as the answer for Question Generation (which is the category of questions that performs the best). Overall, the BLEU scores for the system have a median of 0.4, which is considered as a good quality for language translation.

4.2.4. Question filtering

The proposed model for question-based paraphrase detection was trained and evaluated on the Quora Question Pair (QPP) dataset (Wang, Hamza and Florian, 2017), which contains 400k+ annotated records of the form (q_1, q_2, y) with $y \in \{0, 1\}$ (0 meaning different questions; 1 meaning

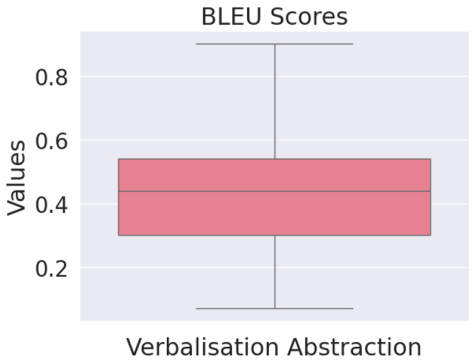

Figure 12: Distribution of BLEU scores for the evaluation of the Verbalisation Abstraction module; considering all the domains/themes of the WDV dataset.

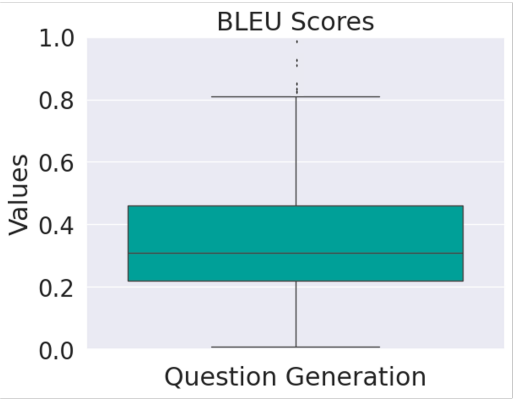

Figure 13: BLEU scores for the Question Generation stage. In the plot is shown the distribution of BLEU scores for the Question Generation stage for all the domains of the dataset including the three types of questions.

equivalent questions). Given the size of the dataset, we use a static split to create the training, validation, and test sets.

Although the QPP dataset covers *questions* in the wider sense (e.g. "What is the difference between a neutral and border state?"), we expect a model trained on this dataset to transfer to *competency questions*. This is motivated by the more general and easier formulation of a competency question, which is supposed to lead to less ambiguity when attempting to classify its equivalence to an alternative formulation. Therefore, the QQP dataset can be considered as a challenging benchmark to train and evaluate our model.

Our model was trained to minimise the binary cross entropy loss between the predicted and the ground-truth labels. The model is implemented in `PyTorch` 1.12 and trained on a NVIDIA T4. An early stopping strategy is used to prevent overfitting with 15 epochs of patience. All dropout layers are set with a drop probability of 0.5 and a fixed learning rate of 0.001 is used by the Adam optimiser with default hyper-parameters ($\beta_1 = 0.9$, and $\beta_2 = 0.999$). A threshold of 0.5 is used to sample from the learned Bernoulli distribution at the sigmoid layer (e.g. the model outputs $\hat{y} \leq$

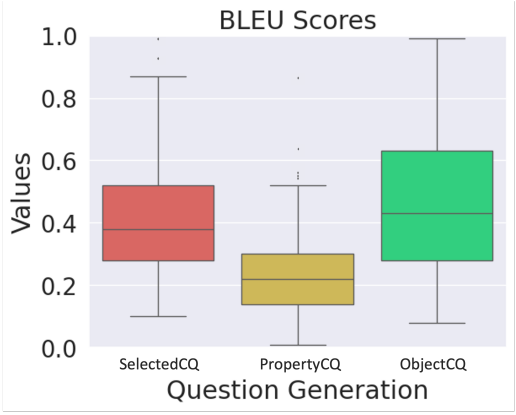

Figure 14: BLEU score for the Question Generation stage individualised. In the plot is shown the distribution of BLEU scores for each type of question that is produced by the Question Generation stage for all the domains of the dataset.

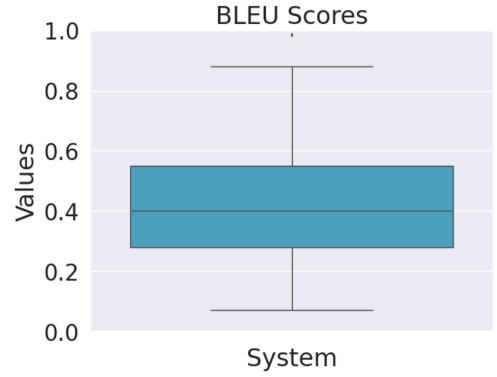

Figure 15: Distribution of BLEU scores for the RevOnt framework containing the Verbalisation Abstraction and the Question Generation stage with only one type of question (the question that is answered by the object of the triple).

0.5 as 0, meaning that the given questions are not equivalent; and $\hat{y} > 0.5$ as 1, in case of equivalence). This is only needed for measuring the model's accuracy on the test set. Notably, we did not find class imbalance in QPP; hence, we can rely on the model's accuracy as the main metric for evaluation.

After ten epochs, training stops with a test loss of 0.32 and accuracy above 86%, which is comparable to the classification results of Wang et al. (2017). The trained model can effectively identify redundant sentences, encompassing both identical sentences and semantically similar ones. For instance, multiple redundant competency questions may convey the same meaning with varied structures, such as "What is the genre of Comics?" and "Comics is in what genre?". In sum, our model can streamline the set of competency questions by eliminating semantically identical sentences.

4.3. Syntactic analysis of verbalisations and CQs

To evaluate and compare qualitative properties of competency questions and verbalisation abstractions, we extracted a set of metrics from the WDV-CQ dataset, and compared their distributions between the human-annotated (WDV-CQ-HA) and the RevOnt-generated (WDV-CQ-RO) subsets. This was done by computing sentence-level features on each subset, individually, and performing statistical tests between both groups to detect significant differences. The evaluation is based on two groups of features: *frequency-based* and *statistic-based*. The first group includes features related to the count of words, verbs, nouns, adjectives, pronouns, etc. The second group include readability features, such as the Flesch-Kincaid grade, the Coleman-Liau test, and the Automated Readability index. We decided to include 6 different readability features as their formulation is designed for different applications (labelling technical manuals, supporting high-tech education, etc.), although they measure readability by approximating the US grade level needed to comprehend a given text. Here, their complementary formulation is needed as there is no universal definition of readability. Together with the frequency-based features, we expect readability scores and syntactic properties to describe structural properties of competency questions and verbalisation abstractions; whose span always corresponds to a single sentence. All features are outlined in Table 4.

The distributions of each measure per subset are illustrated in Figures 16 and 17 for both frequency-based and statistics-based features, respectively. To detect statistical differences between subsets (WDV-CQ-HA and WDV-CQ-RO), we performed pairwise Kolmogorov-Smirnov (KS) tests on each feature. KS tests were chosen due to their non-parametric nature, making them suitable when the normality assumption of the input distributions under comparison is violated (as visible from the skewed distributions plotted in Figures 16 and 17). The results of the statistical tests are provided in the Appendix (Tables 6 and 7), along with the mean and the standard deviation of each feature per subset.

Overall, the boxplots in Figure 16 reveal similarities of features across the human-annotated (HA) and the RevOnt-generated (RO) subsets for Verbalisation, SubjectCQ, and ObjectCQ. In contrast, the boxplots for PropertyCQ differ considerably, displaying high variance and skewness for RO features. The same can be observed for the statistics-based features (readability scores) in Figure 17.

On avergage, when looking at the verbalisations, RO produces slightly longer sentences compared to those in HA (9.10 ± 3.78 and 9.10 ± 3.78 words, respectively); whereas their number of verbs is quite similar (1.23 ± 0.67 and 1.36 ± 0.67). We also observe strong similarities for the number of adverbs, adjectives, pronouns, conjunctions, and modal verbs – which are all close to 0; as well as for adjectives and prepositions (close to 1 for both subsets). This implies that RevOnt manages to approximate the syntactic structure of a well-defined (human-annotated) verbalisation. Instead, the slight increase in length of RO's verbalisations could be associated to the higher number of nouns (HA : 3.28 ± 1.83; RO: 4.93 ± 2.85) and named entities (HA : 0.46 ± 0.89; RO: 2.05 ± 1.16) generated by RevOnt. Concerning the statistics-based features, Figure 17 confirms that while readability scores (US grade levels) returned by the various tests have

Table 4
Overview of the features used to describe and compare qualitative properties of competency questions and verbalisation abstraction. Features are grouped into *frequency-based* and *statistics-based*, with the latter entailing redability properties.

Group	Feature	Description
Frequency-based	word count	Number of words
	verb count	Number of verbs
	noun count	Number of nouns
	adj count	Number of adjectives
	adv count	Number of adverbs
	pron count	Number of pronouns
	conj count	Number of conjunctions
	prep count	Number of prepositions
	modal verb count	Number of modal verbs (could, should, would, etc.)
	unique word count	Number of distinct words
	stop word count	Number of commonly used stop words, e.g. "the", "is", "in", etc.
	NE count	Number of named entities (e.g. persons, organizations, locations, etc.)
Statistic-based	Flesch Kincaid grade	A readability score that approximates the U.S. grade level thought necessary to comprehend the text. It ranges from 0-1 (Pre-kindergarten - 1st grade) to 11-18 (11th grade - 18th grade).
	Coleman Liau index	A test designed to gauge the understandability of a text, which also approximates the U.S. grade level but relies on characters instead of syllables per word. It is computed by accounting for the proportion of letters and sentences per 100 words.
	Automated Readability Index	Another approximate of readability in relation to the required US grade level, which also relies on character statistics. The index was designed for real-time monitoring of readability on electric typewriters.
	Dale Chall Readability	A readability formula based on a list of 3000 familiar words that groups of fourth-grade American students could reliably understand. The formula considers any word not on that list to be difficult.
	Linsear Write	Another approximation of US grade level, based on sentence length and the number of words with 3+ syllables. It was originally developed to compute the readability of US Air Force technical manuals.
	Gunning Fog	A readability test calculating a weighted average of the number of words per sentence, and the number of long words per sentence.

different magnitudes; they all grow consistently with respect to the subsets (RO and HA scores are directly proportional). Therefore, rather than attempting to map our verbalisations to the required US grade levels, we can still observe that RevOnt's verbalisations have a tendency to be more complex to read. For example, the average Flesch-Kincaid scores for HA and RO are 5.16 ± 4.30 and 6.82 ± 5.81, respectively.

In line with the experimental setup of Section 4.2.2, we performed the syntactic analysis of competency questions individually on those defined from subjects (SubjectCQ), predicates (PropertyCQ), and objects (ObjectCQ). The results revealed similar insights to the verbalisations. Frequency-based features related to the count of words, verbs, and nouns have comparable distributions across HA and RO; with adjectives, adverbs, pronouns occurring rarely in both subsets. Overall, CQs originating from predicates (PropertyCQ) are the longest for human-annotators (HA), with a higher number of nouns, conjunctions, and unique words. Instead, RevOnt shows the opposite trend, as the same features have the lowest figures in comparison to SubjectCQ and ObjectCQ. This is also confirmed by the readability scores in Figure 17, which are considerably high for HA (Flesch-Kincaid grade of 6.04 ± 4.17) while being lower yet with higher variability for RO (Flesch Kincaid grade 1.15 ± 7.17). We hypothesise that the inverted trend is due to the nature of the task, as human participants found the formulation of CQs from the predicate to be less intuitive; which in turn, could explain the lengthier and more complex HA competency questions. Instead, RevOnt tends to produce shorter and simpler questions from the triple's predicate. However, the latter yielded the lowest BLEU scores in relation to the human annotations (c.f. Section 4.2.2), meaning that their output is not accurate.

For SubjectCQs, HA competency questions are the easiest to read (e.g. Flesh-Kincaid score: 2.22 ± 4.21), whereas RevOnt produces more complex CQs (e.g. Flesh-Kincaid score: 3.99 ± 3.42). The most alignment among both subsets is observed for ObjectCQ, which produces the smallest differences for both frequency-based and statistics-based features. This finding reinforces the results reported Section 4.2.2, where RevOnt attained the best BLEU scores for the competency question generation module. This implies that not only the CQs extracted from (triple) objects are the most accurate semantically (i.e., they entail similar ontological requirements when compared to human-generated CQs), but they also have similar syntactic properties.

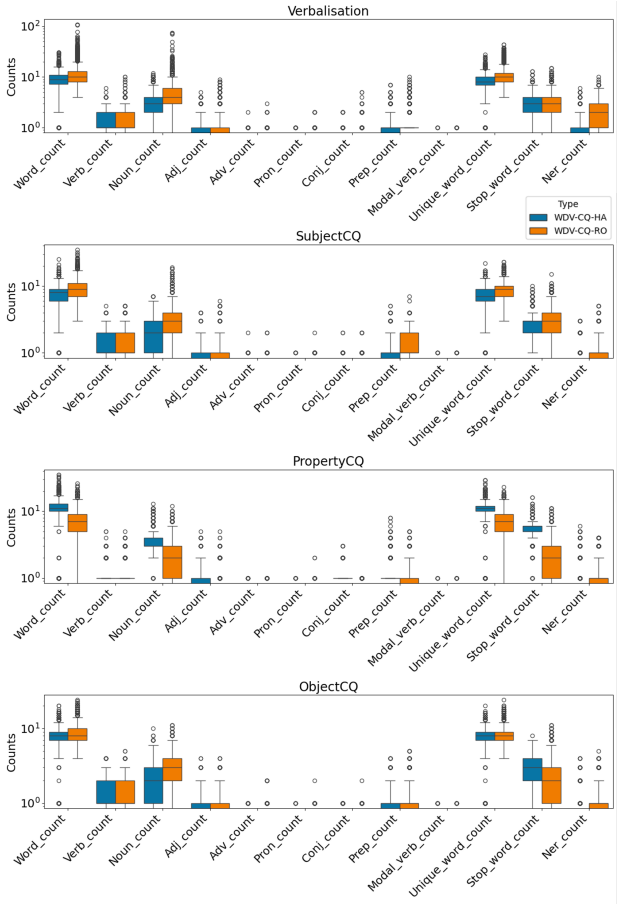

Figure 16: Comparison of frequency-based features between human-annotated (WDV-CQ-HA) and RevOnt-generated (WDV-CQ-RO) verbalisation sand CQs, on a log scale.

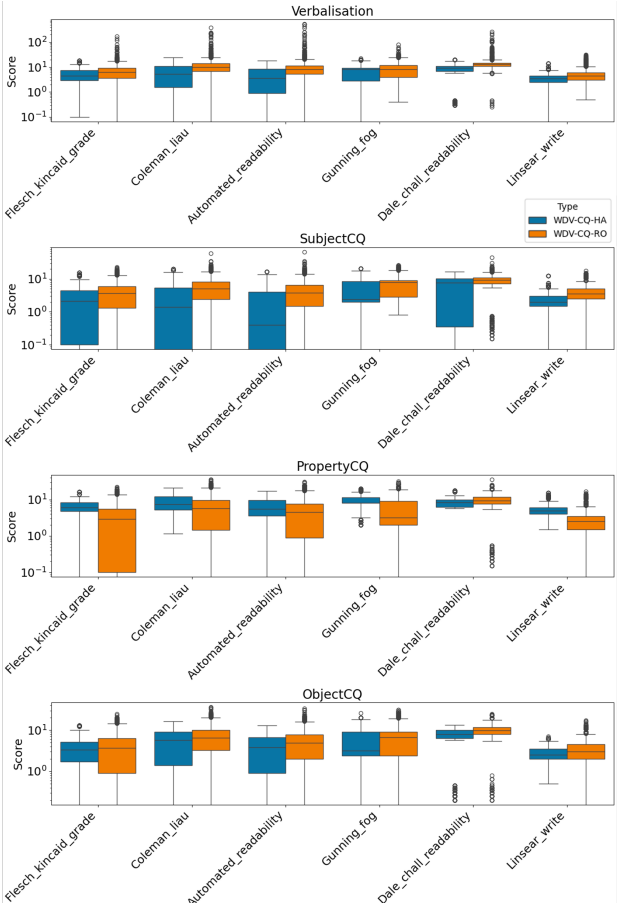

Figure 17: Comparison of statistics-based features (readability measures) between human-annotated (WDV-CQ-HA) and RevOnt-generated (WDV-CQ-RO) verbalisations and CQs.

To shed more light into the latter finding, Table 5 provides comparative examples between HA and RO. It reveals a distinct difference: HA typically employs more abstract and general words, whereas RO still makes extensive use of named entities, such as “Leo” or “Michael J Adams”. For ’SubjectCQ’, the structure of the competency questions is similar; however, RO often incorporates specific named entities into the generation of CQs, forming simpler structured sentences than those found in HA. This is confirmed by the statistical tests, have indicated potential significant differences between WDV-CQ and RevOnt-CQ, which may be attributed to the use of named-entity words. Nevertheless, CQs generated by HA and RO show similarities. Therefore, reducing the reliance on named-entity words could be a key factor in enhancing the quality of RO-generated CQs.

5. Discussion and future work

This work demonstrated how the use of natural language processing methods enables the extraction of competency questions from knowledge graphs. The current implementation of RevOnt is publicly available on GitHub[35] and includes setup scripts and instructions to replicate all the steps of extraction process (c.f. Section 4). All code is released under the MIT license, whereas the WDV-CQ dataset follows the Attribution 4.0 International (CC-BY 4.0).

To use RevOnt, users are required to produce a verbalisation of their knowledge graph using the same format of the WDV dataset (as shown in the example in Figure 5). This ensures that the question generation module produces reliable results as reported in our experiments. Nonetheless, given the modular architecture of RevOnt, users can also redefine or implement their own question generation model (e.g. using an alternative verbalisation format) and plug its outputs to the question filtering module. By doing so, we expect the hyper-parameters of the methods and tools outlined in Table 1 to remain consistent with our experimental setup.

Limitations. Overall, workflows based on language models are notoriously sensitive to the input data and the configuration of the hyper-parameters (Lavie, 2010). This issue is generally observed in the generation of the questions by RevOnt, where the distance between the triple and its verbalisation plays a significant role (c.f. Example 4.2). In addition, language models are still prone to issues and limitations which inevitably propagate into RevOnt’s pipeline.

[35] https://github.com/King-s-Knowledge-Graph-Lab/revont

Table 5
Examples of human-annotated (HA) and RevOnt-generated (RO) competency questions from the WDV-CQ dataset.

Subset	Original statement	subjectCQ	propertyCQ	objectCQ
Good examples				
RO	Klaus Francke's catalog code is 11000569.	Whose catalog code is 11000569?	What is Klaus Francke's?	What is Klaus Francke's catalog code?
HA	Klaus Francke's catalog code is 11000569.	Who has the catalog code 11000569?	What does Klaus Francke has?	What is the catalog code of Klaus Francke
Modest examples				
RO	Zach Trotman has 13 career points.	Who has 13 career points?	How many career points does Zach Trotman have?	How many career points does Zach Trotman have?
HA	Zach Trotman has 13 career points.	Who has 13 career points?	What is the connection between American ice hockey defenceman and the 13 career ponits?	How many career points does American ice hockey defenceman have?
Bad examples				
RO	Kirk Muller has 357 goals in his career.	Who has 357 goals in his career?	How many goals has Kirk Muller scored?	How many goals has Kirk Muller scored in his career?
HA	Kirk Muller has 357 goals in his career.	Who has 357 goals in his career?	What does Canadian ice hockey player have to do with 357 goals?	How many goals does Canadian ice hockey player have?

Below we highlight issues that are well-known at the intersection of natural language processing, ontology learning, and schema induction and discovery.

1. In curating the WDV dataset, Amaral (2022) already found that existing NER models performed poorly on specific themes (50% of the WDV dataset). Their performance is explained by the heterogeneous data used to train these models, which is comprehensively different compared to the dataset. These models are often trained with data from news articles, and emails/chat data, which explains the good performance with themes such as Artist, Athlete, City, etc. By training a NER model with data specific to the domain of interest, it is possible to make the approach independent from the Wikidata query service.
2. In relation to the previous point, the verbalisation abstraction process in RevOnt (c.f. Section 3.2) has still a tendency to keep named entities. In fact, the distributions of NER counts for abstracted verbalisation and competency questions (Tables 6 and 7) have means greater than those observed for the human annotations. This can also be observed in the examples reported in Table 5. Identifying the correct abstraction for a given entity – depending on the context in which it occurs, is indeed a difficult task; especially when the context is poor (e.g. a single triple, rather than a paragraph of which the sentence is part of) or ambiguous ('Michael Jackson' as either a singer, musician, or person). By leveraging additional information extracted from the knowledge graph (e.g. a cluster of similar entities and relations) we expect a larger context to be more informative for abstracting named entities.
3. Generally, the default language of the training data of a language model is English, which limits the usability of the model for multilingual data. Big knowledge graphs as Wikidata and DBpedia contain structured knowledge of entities in several distinct languages, and they are useful resources for cross-lingual artificial intelligence and natural language processing applications (Wang, Lv, Lan and Zhang, 2018). Therefore, language models trained with multilingual data could enhance the performance of the approach. An example of a multilingual language model is a release of BERT[36] that is pre-trained on the 104 languages with the largest Wikipedia pages. Another example is T5-base[37], which covers English, French, Romanian, German (Raffel, Shazeer, Roberts, Lee, Narang, Matena, Zhou, Li, Liu et al., 2020).
4. The dependency between the language models and methods used in NLP pipelines may also pose further issues. As the overall performance of the system is impacted by each component, the effect of errors is multiplicative. This is due to the dependency of each component to the preceding, propagating errors along the processing. In the worst cases, this may result in a final output that may be unsatisfactory despite the good performance of each component, individually Resnik and Lin (2010). In RevOnt, this has been mitigated by finding a configuration of the models (Table 1) that is optimal to the tasks and the data at hand. Hence, extending or replacing one or more components would necessitate a trial and error approach to adjust the parametrisation of our methods.

Overall, a way to mitigate the aforementioned issue is the addition of a manual validation phase across the various steps; where end-users can collaboratively refine the

[36] https://huggingface.co/bert-base-multilingual-cased
[37] https://huggingface.co/t5-base

outputs of the system. This may be facilitated by interactive interfaces as well as crowd-sourcing and social media approaches to validate the intermediate results Asim et al. (2018). Notably, most of the issues arises from known limitations of NLP methods, especially when applied to novel or unconventional uses cases such as the one proposed in this article. Given the latest developments in NLP, we believe that most of these concerns can effectively be addressed by Large Language Models and Conversational AI systems such as ChatGPT[38], and Bard [39]; which we expect to further improve the results reported in this article.

Future work. Based on the discussion above, and the opportunities opened up by RevOnt, we have also identified three directions that we plan to pursue as future work.

1. The method that we have defined generates questions based on the abstraction of triple verbalisations. In particular, the question generation model (T5) performs better when is provided with the sentence and the answer. This limits the usability of the method with datasets where the answer is not explicit. It requires additional steps such as Part-Of-Speech tagging, parsing, and lemmatisation to be able to provide an entity as an answer. We plan to refine this aspect of the method by 1) detecting entities that might serve as answers or not providing an answer, and 2) providing paragraphs instead of sentences. Concerning the second point, the T-Rex dataset is a good fit, since it provides textual descriptions of Wikidata entries. Related to this, we also plan to experiment with other language models and use KGs other than Wikidata to generalise our results.
2. As conceptualised in Figure 1, developing automatic or assisted workflows for ontology testing is also a planned research direction. One way to address this is to leverage large datasets of annotated CQs, e.g. the BigCQ dataset Wiśniewski, Potoniec and Ławrynowicz (2021), to learn a mapping from competency questions to SPARQL templates. This would ensures that the generated CQs can be easily converted into SPARQL queries (for ontology testing and information retrieval) while providing methods for evaluating the potential reuse of a KG resource Alharbi, Tamma, Grasso and Payne (2023).
3. Finally, we also plan to extend the question generation module to handle the automatic creation, or induction, of competency questions corresponding to more than a single triple pattern. This would allow users to extract more complex ontological requirements, beyond those emerging from a local triple-level approach.

6. Conclusions

This work investigated the extent to which it is possible to extract competency questions from knowledge graphs (RQ1); and to understand how quality features of human-annotated and generated questions differ syntactically (RQ2). To address RQ1, we have introduced **RevOnt**, an approach that leverages various language models to extract competency questions from verbalisations of knowledge graphs. Intuitively, the approach is based on reversing the ontology engineering process, hence accommodating projects and use cases where already existent knowledge graphs drive the definition and extension of ontology requirements. RevOnt abstracts the verbalisation of a triple, generates three questions for each of them (using either *subject*, *predicate*, or *object*), and filters out semantically similar questions to reduce potential redundancies. The end result of the approach is a set of core competency questions that represent/describe the triples as ontological requirements. To address RQ2, we provided a literature review on the notion of competency questions, their function in the ontology engineering process, and the quality qualities that they demonstrate. As most of these properties can be inferred from the qualitative properties of competency questions, we designed an experimental setup to automatically extract syntactic features from manually annotated and RevOnt-generated CQs. Statistical comparison were then carried out to detect significant differences of features across groups.

We implemented and tested RevOnt using the WDV dataset, a dataset of verbalisations from the Wikidata knowledge graph. Through manual collection, we extended the former and contributed **WDV-CQ**, a new dataset of manually annotated and RevOnt-generated *verbalisation abstraction* (1786) and *competency questions* (1904). By comparing RevOnt's output with its corresponding human annotations using BLEU, we found that 75% of the verbalisation abstractions generated by the former (through the verbalisation Abstraction module, c.f. Section 3.2), have a good to high quality. Meanwhile, generated questions (Question Generation component, c.f. Section 4.2.2), have a wider range of quality starting from poor to high. Nonetheless, the type of questions that received a higher score quality-wise (median BLEU score of 0.42) is the question that RevOnt generates when the answer is the object of the triple. The last component, Question Filtering (Section 3.4), is evaluated with an accuracy of 86%, comparable to state-of-the-art classification models for paraphrase detection; which is needed to filter out semantically equivalent CQs.

The comparison of syntactic features extracted from verbalisation abstractions and competency questions revealed similar (yet statistically different) distributions across human-annotated and RevOnt-generated outputs. Both verbalisation and CQs have similar word, verb, adverb, pronoun, conjunction and stop word counts. Nevertheless, RevOnt tends to generate CQs and verbalisation abstractions with more named entities and a slight increase in difficulty for readability. Also, CQs generated from the object of a triple (ObjectCQ) show the most similar distributions of features compared to those of human annotations. This also confirms the experimental findings of the BLEU-based evaluation, hence concluding that ObjectCQs are the most

[38] https://openai.com/blog/chatgpt
[39] https://bard.google.com

accurate semantically and share syntactic and readability properties with manually generated competency questions.

RevOnt can be extended by fine-tuning text-to-text transformation models to accept paragraphs instead of sentences, to provide more context. In addition, the model can be adjusted to use a topic detection algorithm to leverage this information when generating more accurate questions on the triple. Meanwhile, performance can also be reusing large language models and fine-tuning (or simply prompting) them on the specific domain data. This could result in better abstraction of named entities, which are still affecting the quality of RevOnt generations.

CRediT authorship contribution statement

Fiorela Ciroku: Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Visualisation. **Jacopo de Berardinis:** Methodology, Software, Validation, Formal analysis, Investigation, Resources, Dataset Curation, CQ Analysis, Writing- Original and Second Draft, Supervision. **Jongmo Kim:** Software, Formal analysis, Resources, Dataset Curation, CQ Analysis, Writing- Second Draft. **Albert Meroño-Peñuela:** Conceptualisation, Methodology, Writing - Review & Editing, Supervision, Project administration. **Valentina Presutti:** Conceptualisation, Methodology, Writing - Review & Editing, Supervision, Project administration. **Elena Simperl:** Conceptualisation, Methodology, Writing - Review & Editing, Supervision, Project administration.

References

Abdelghany, A.S., Darwish, N.R., Hefni, H.A., 2019. An agile methodology for ontology development. International Journal of Intelligent Engineering and Systems 12, 170–181. doi:http://dx.doi.org/10.22266/ijies2019.0430.17.

Aggarwal, C.C., Hinneburg, A., Keim, D.A., 2001. On the surprising behavior of distance metrics in high dimensional space, in: Database Theory — ICDT 2001, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 420–434. doi:http://dx.doi.org/10.1007/3-540-44503-X_27.

Alharbi, R., Tamma, V., Grasso, F., Payne, T., 2023. An experiment in retrofitting competency questions for existing ontologies. arXiv preprint arXiv:2311.05662 .

Amaral, G., 2022. WDV URL: https://figshare.com/articles/dataset/WDV/17159045, doi:10.6084/m9.figshare.17159045.v1.

Amaral, G., Rodrigues, O., Simperl, E., 2022. Wdv: A broad data verbalisation dataset built from wikidata, in: The Semantic Web – ISWC 2022, Springer International Publishing, Cham. pp. 556–574. doi:http://dx.doi.org/10.1007/978-3-031-19433-7_32.

Asim, M.N., Wasim, M., Khan, M.U.G., Mahmood, W., Abbasi, H.M., 2018. A survey of ontology learning techniques and applications. Database 2018. URL: https://doi.org/10.1093/database/bay101, doi:10.1093/database/bay101.

Asprino, L., Carriero, V.A., Presutti, V., 2021. Extraction of common conceptual components from multiple ontologies, in: Proceedings of the 11th on Knowledge Capture Conference, Association for Computing Machinery, New York, NY, USA. p. 185–192. URL: https://doi.org/10.1145/3460210.3493542, doi:10.1145/3460210.3493542.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z., 2007. Dbpedia: A nucleus for a web of open data, in: The semantic web. Springer, pp. 722–735. doi:https://doi.org/10.1007/978-3-540-76298-0_52.

de Berardinis, J., Carriero, V.A., Jain, N., Lazzari, N., Meroño-Peñuela, A., Poltronieri, A., Presutti, V., 2023. The polifonia ontology network: Building a semantic backbone for musical heritage, in: Proceedings of the 22nd International Semantic Web Conference.

Bezerra, C., Freitas, F., Santana, F., 2013. Evaluating ontologies with competency questions, in: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), pp. 284–285. doi:10.1109/WI-IAT.2013.199.

Čebirić, Š., Goasdoué, F., Kondylakis, H., Kotzinos, D., Manolescu, I., Troullinou, G., Zneika, M., 2019. Summarizing semantic graphs: a survey. The VLDB journal 28, 295–327. doi:https://doi.org/10.1007/s00778-018-0528-3.

Chen, J., Gu, J., 2021. Adol: a novel framework for automatic domain ontology learning. Journal of Supercomputing 77, 152–169. doi:10.1007/s11227-020-03261-7. published: 28 March 2020.

Corcho, O., Fernández-López, M., Gómez-Pérez, A., López-Cima, A., 2005. Building Legal Ontologies with METHONTOLOGY and WebODE. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 142–157. URL: https://doi.org/10.1007/978-3-540-32253-5_9, doi:10.1007/978-3-540-32253-5_9.

Dietz, J.L., 2006. What is Enterprise Ontology?. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 7–13. URL: https://doi.org/10.1007/3-540-33149-2_2, doi:10.1007/3-540-33149-2_2.

Fadel, F., Fox, M., Gruninger, M., 1994. A generic enterprise resource ontology, in: Proceedings of 3rd IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 117–128. doi:10.1109/ENABL.1994.330496.

Faure, D., Poibeau, T., 2000. First experiments of using semantic knowledge learned by asium for information extraction task using intex, in: Ontology learning ECAI-2000 workshop, Citeseer. pp. 7–12.

Fellbaum, C., 2010. WordNet. Springer Netherlands, Dordrecht. pp. 231–243. URL: https://doi.org/10.1007/978-90-481-8847-5_10, doi:10.1007/978-90-481-8847-5_10.

Fernández-Izquierdo, A., García-Castro, R., 2022. Ontology verification testing using lexico-syntactic patterns. Information Sciences 582, 89–113. doi:https://doi.org/10.1016/j.ins.2021.09.011.

Fox, M.S., Barbuceanu, M., Gruninger, M., 1996. An organisation ontology for enterprise modeling: Preliminary concepts for linking structure and behaviour. Computers in Industry 29, 123–134. URL: https://www.sciencedirect.com/science/article/pii/0166361595000798, doi:https://doi.org/10.1016/0166-3615(95)00079-8. wET ICE '95.

Frantzi, K., Ananiadou, S., Mima, H., 2000. Automatic recognition of multi-word terms:. the c-value/nc-value method. International journal on digital libraries 3, 115–130.

Gangemi, A., Presutti, V., 2009. Ontology Design Patterns. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 221–243. URL: https://doi.org/10.1007/978-3-540-92673-3_10, doi:10.1007/978-3-540-92673-3_10.

Gick, M.L., Holyoak, K.J., 1983. Schema induction and analogical transfer. Cognitive psychology 15, 1–38. doi:https://doi.org/10.1016/0010-0285(83)90002-6.

Gruber, T.R., 1992. Ontolingua: A mechanism to support portable ontologies.

Grüninger, M., Fox, M.S., 1995. The Role of Competency Questions in Enterprise Engineering. Springer US, Boston, MA. pp. 22–31. URL: https://doi.org/10.1007/978-0-387-34847-6_3, doi:10.1007/978-0-387-34847-6_3.

Hahn, U., Romacker, M., 2001. The syndikate text knowledge base generator, in: Proceedings of the first international conference on Human language technology research.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., et al., 2021. Knowledge graphs. ACM Computing Surveys (Csur) 54, 1–37.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, PMLR. pp. 448–456.

Ji, Q., Qi, G., Gao, H., Wu, T., 2019. Survey on schema induction from knowledge graphs, in: Knowledge Graph and Semantic Computing. Knowledge Computing and Language Understanding, Springer

Singapore, Singapore. pp. 136–142. doi:http://dx.doi.org/10.1007/978-981-13-3146-6_12.

John, S., Shah, N., Stewart, C., 2018. Towards a software centric approach for ontology development: Novel methodology and its application, in: 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE), pp. 139–146. doi:10.1109/ICEBE.2018.00030.

Jones, D., Bench-Capon, T., Visser, P., 1998. Methodologies for ontology development .

Kellou-Menouer, K., Kardoulakis, N., Troullinou, G., Kedad, Z., Plexousakis, D., Kondylakis, H., 2022. A survey on semantic schema discovery. The VLDB Journal 31, 675–710. doi:10.1007/s00778-021-00717-x.

Kouagou, N.J., Heindorf, S., Demir, C., Ngonga Ngomo, A.C., 2022. Learning concept lengths accelerates concept learning in alc, in: European Semantic Web Conference, Springer Nature Switzerland. pp. 236–252.

Lavie, A., 2010. Evaluating the output of machine translation systems, in: Proceedings of the 9th Conference of the Association for Machine Translation in the Americas: Tutorials.

Lenat, D., Guha, R., 1993. Building large knowledge-based systems: Representation and inference in the cyc project. Artificial Intelligence 61, 4152.

Loper, E., Bird, S., 2002. Nltk: The natural language toolkit, in: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1, Association for Computational Linguistics, USA. p. 63–70. URL: https://doi.org/10.3115/1118108.1118117, doi:10.3115/1118108.1118117.

McInnes, L., Healy, J., Astels, S., 2017. hdbscan: Hierarchical density based clustering. Journal of Open Source Software 2, 205. URL: https://doi.org/10.21105/joss.00205, doi:10.21105/joss.00205.

McInnes, L., Healy, J., Melville, J., 2018. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 .

Mintz, M., Bills, S., Snow, R., Jurafsky, D., 2009. Distant supervision for relation extraction without labeled data, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2, Association for Computational Linguistics, USA. p. 1003–1011.

Napoles, C., Sakaguchi, K., Tetreault, J., 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, Valencia, Spain. pp. 229–234. URL: https://aclanthology.org/E17-2037, doi:http://dx.doi.org/10.18653/v1/E17-2037.

Nuzzolese, A.G., Gangemi, A., Presutti, V., Ciancarini, P., 2011. Encyclopedic knowledge patterns from wikipedia links, in: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (Eds.), The Semantic Web – ISWC 2011, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 520–536. doi:http://dx.doi.org/10.1007/978-3-642-25073-6_33.

Öhgren, A., Sandkuhl, K., 2005. Towards a methodology for ontology development in small and medium-sized enterprises, in: IADIS International Conference Applied Computing 2005, pp. 369–376.

Papineni, K., Roukos, S., Ward, T., Zhu, W.J., 2002. Bleu: A method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, USA. p. 311–318. URL: https://doi.org/10.3115/1073083.1073135, doi:10.3115/1073083.1073135.

Paschke, A., Schäfermeier, R., 2018. OntoMaven - Maven-Based Ontology Development and Management of Distributed Ontology Repositories. Springer International Publishing, Cham. pp. 251–273. URL: https://doi.org/10.1007/978-3-319-64161-4_12, doi:10.1007/978-3-319-64161-4_12.

Peroni, S., 2017. A simplified agile methodology for ontology development, in: Dragoni, M., Poveda-Villalón, M., Jimenez-Ruiz, E. (Eds.), OWL: Experiences and Directions – Reasoner Evaluation, Springer International Publishing, Cham. pp. 55–69. doi:http://dx.doi.org/10.1007/978-3-319-54627-8_5.

Piscopo, A., Simperl, E., 2018. Who models the world? collaborative ontology creation and user roles in wikidata. Proceedings of the ACM on Human-Computer Interaction 2, 1–18.

Pouriyeh, S., Allahyari, M., Kochut, K., Arabnia, H.R., 2018. A comprehensive survey of ontology summarization: measures and methods. arXiv preprint arXiv:1801.01937 .

Presutti, V., Daga, E., Gangemi, A., Blomqvist, E., 2009. extreme design with content ontology design patterns, in: Proc. Workshop on Ontology Patterns, pp. 83–97.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., et al., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. 21, 1–67.

Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P., 2016. SQuAD: 100,000+ questions for machine comprehension of text, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas. pp. 2383–2392. URL: https://aclanthology.org/D16-1264, doi:10.18653/v1/D16-1264.

Reimers, N., Gurevych, I., 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China. pp. 3982–3992. URL: https://aclanthology.org/D19-1410, doi:10.18653/v1/D19-1410.

Ren, Y., Parvizi, A., Mellish, C., Pan, J.Z., van Deemter, K., Stevens, R., 2014. Towards competency question-driven ontology authoring, in: The Semantic Web: Trends and Challenges, Springer International Publishing, Cham. pp. 752–767. doi:http://dx.doi.org/10.1007/978-3-319-07443-6_50.

Resnik, P., Lin, J., 2010. 11 evaluation of nlp systems. The handbook of computational linguistics and natural language processing 57.

Roberts, A., Raffel, C., 2020. Exploring transfer learning with t5: the text-to-text transfer transformer. Accessed on , 23–07.

Romero, M., 2021. T5 (base) fine-tuned on squad for qg via ap. https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap.

Schekotihin, K., Rodler, P., Schmid, W., Horridge, M., Tudorache, T., 2018. Test-driven ontology development in protégé., in: ICBO.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15, 1929–1958.

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D., 2002. Ontoedit: Collaborative ontology development for the semantic web, in: The Semantic Web — ISWC 2002, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 221–235. doi:http://dx.doi.org/10.1007/3-540-48005-6_18.

Tettamanzi, A.G.B., Faron-Zucker, C., Gandon, F., 2014. Testing owl axioms against rdf facts: A possibilistic approach, in: Janowicz, K., Schlobach, S., Lambrix, P., Hyvönen, E. (Eds.), Knowledge Engineering and Knowledge Management, Springer International Publishing, Cham. pp. 519–530.

Toutanova, K., Chen, D., 2015. Observed versus latent features for knowledge base and text inference, in: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, Association for Computational Linguistics, Beijing, China. pp. 57–66. URL: https://aclanthology.org/W15-4007, doi:10.18653/v1/W15-4007.

Uschold, M., Gruninger, M., 1996. Ontologies: Principles, methods and applications. The knowledge engineering review 11, 93–136. doi:http://dx.doi.org/10.1017/S0269888900007797.

Uschold, M., King, M., 1995. Towards a methodology for building ontologies. Citeseer.

Vrandečić, D., 2009. Ontology Evaluation. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 293–313. URL: https://doi.org/10.1007/978-3-540-92673-3_13, doi:10.1007/978-3-540-92673-3_13.

Vrandečić, D., Krötzsch, M., 2014. Wikidata: A free collaborative knowledgebase. Commun. ACM 57, 78–85. URL: https://doi.org/10.1145/2629489, doi:10.1145/2629489.

Wang, Z., Hamza, W., Florian, R., 2017. Bilateral multi-perspective matching for natural language sentences, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 4144–4150. URL: https://doi.org/10.24963/ijcai.2017/579, doi:10.24963/ijcai.2017/579.

Wang, Z., Lv, Q., Lan, X., Zhang, Y., 2018. Cross-lingual knowledge graph alignment via graph convolutional networks, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium. pp. 349–357. URL: https://aclanthology.org/D18-1032, doi:10.18653/v1/D18-1032.

Wiśniewski, D., Potoniec, J., Ławrynowicz, A., 2021. Bigcq: A large-scale synthetic dataset of competency question patterns formalized into sparql-owl query templates. arXiv preprint arXiv:2105.09574 .

Zhao, L., Ichise, R., 2012. Mid-ontology learning from linked data, in: The Semantic Web: Joint International Semantic Technology Conference, JIST 2011, Hangzhou, China, December 4-7, 2011. Proceedings 1, Springer. pp. 112–127.

A. Appendix

Table 6
Overview of aggregated *frequency-based* features extracted from human-annotated (WDV-CQ-HA subset) and RevOnt-generated (WDV-CQ-RO subset) verbalisations and competency questions. Mean (μ) and standard deviation (σ) are reported for each feature and subset, alongside the results of the statistical comparison across the distributions of the former groups.

		WDV-CQ-HA		WDV-CQ-RO		Statistical test	
Type	**Feature**	μ	σ	μ	σ	KS Statistic	p-value
Verbalisation	Word_count	9.10982	3.787903	11.174313	5.303054	0.1564	< 0.001
	Verb_count	1.239176	0.676624	1.366373	0.675432	0.0709	< 0.001
	Noun_count	3.284055	1.835723	4.932694	2.858977	0.3595	< 0.001
	Adj_count	0.659451	0.836106	0.619955	0.920612	0.0390	0.1352
	Adv_count	0.024815	0.158961	0.024583	0.161509	-	-
	Pron_count	0.007392	0.08568	0.017221	0.135061	-	-
	Conj_count	0.053326	0.238429	0.08926	0.327621	-	-
	Prep_count	0.998416	0.917919	1.151308	0.911466	0.1106	< 0.001
	Modal_verb_count	0.00264	0.051326	0.002235	0.047224	-	-
	Unique_word_count	8.520063	3.342623	10.491652	3.861173	0.2159	< 0.001
	Stop_word_count	2.92397	1.618542	2.893651	1.530433	0.0685	< 0.001
	Ner_count	0.464097	0.894622	2.055475	1.165087	0.6488	< 0.001
SubjectCQ	Word_count	7.461457	2.809941	9.397923	3.047233	0.2213	< 0.001
	Verb_count	1.250792	0.624332	1.386881	0.627028	0.0713	< 0.001
	Noun_count	2.133052	1.301077	2.853556	1.573297	0.1839	< 0.001
	Adj_count	0.388596	0.619286	0.382674	0.619288	0.0069	1
	Adv_count	0.022175	0.150836	0.011174	0.108809	-	-
	Pron_count	0.003696	0.060697	0.012488	0.112237	-	-
	Conj_count	0.029567	0.175559	0.022479	0.151752	-	-
	Prep_count	0.790919	0.745192	1.120547	0.71831	0.2038	< 0.001
	Modal_verb_count	0.003168	0.05621	0.001972	0.044365	-	-
	Unique_word_count	7.217529	2.585299	8.825292	2.354825	0.2441	< 0.001
	Stop_word_count	2.684794	1.399725	3.491784	1.841941	0.1689	< 0.001
	Ner_count	0.230201	0.534905	0.644275	0.690399	0.3595	< 0.001
PropertyCQ	Word_count	11.3717	4.290453	6.929539	3.610875	0.6247	< 0.001
	Verb_count	1.051742	0.506132	1.108321	0.634981	0.1078	< 0.001
	Noun_count	3.542239	1.902833	2.462732	1.633899	0.3193	< 0.001
	Adj_count	0.458289	0.707743	0.208492	0.471001	0.1708	< 0.001
	Adv_count	0.009504	0.097048	0.009728	0.098156	-	-
	Pron_count	0.00528	0.07249	0.005653	0.078405	-	-
	Conj_count	0.80887	0.46585	0.018667	0.135355	-	-
	Prep_count	1.222809	0.873382	0.509531	0.688841	0.4616	< 0.001
	Modal_verb_count	0.00264	0.051326	0.001446	0.038002	-	-
	Unique_word_count	10.57075	3.799798	6.741422	3.378023	0.6150	< 0.001
	Stop_word_count	5.181098	2.047106	1.995662	1.552773	0.6886	< 0.001
	Ner_count	0.402323	0.862872	0.847246	0.681109	0.4734	< 0.001
ObjectCQ	Word_count	7.896515	2.769706	7.855133	3.344069	0.0882	< 0.001
	Verb_count	1.359029	0.666083	1.225976	0.634546	0.1073	< 0.001
	Noun_count	2.219113	1.327125	2.754042	1.537968	0.2102	< 0.001
	Adj_count	0.477825	0.653557	0.340607	0.581339	0.1080	< 0.001
	Adv_count	0.015839	0.124887	0.016958	0.135094	-	-
	Pron_count	0.003696	0.060697	0.008282	0.092072	-	-
	Conj_count	0.030623	0.17234	0.012883	0.113936	-	-
	Prep_count	0.684266	0.703621	0.712633	0.701583	0.0240	< 0.001
	Modal_verb_count	0.004752	0.068788	0.001709	0.041307	-	-
	Unique_word_count	7.68849	2.609079	7.641777	3.090298	0.0882	< 0.001
	Stop_word_count	2.892819	1.345149	2.413698	1.623725	0.2009	< 0.001
	Ner_count	0.247624	0.568993	0.855922	0.635948	0.5386	< 0.001

Table 7
Overview of aggregated *statistics-based* features extracted from human-annotated (WDV-CQ-HA subset) and RevOnt-generated (WDV-CQ-RO subset) verbalisations and competency questions. Notation is consistent with Table 6.

		WDV-CQ-HA		WDV-CQ-RO		Statistical test	
Type	**Feature**	μ	σ	μ	σ	KS Statistic	p-value
Verbalisation	Flesch_kincaid_grade	5.1645	4.3038	6.8234	5.4145	0.1696	< 0.001
	Coleman_liau	6.7585	6.9995	11.2218	10.4399	0.3544	< 0.001
	Automated_readability	4.9664	5.3694	9.3776	14.7070	0.3638	< 0.001
	Gunning_fog	6.6066	4.3905	8.9610	5.4593	0.2105	< 0.001
	Dale_chall_readability	9.0684	4.3007	13.3295	6.5789	0.4760	< 0.001
	Linsear_write	3.6367	1.8109	5.1159	2.9878	0.2371	< 0.001
SubjectCQ	Flesch_kincaid_grade	2.2204	4.2172	3.9923	3.4255	0.2393	< 0.001
	Coleman_liau	1.5207	6.0379	5.6575	4.6460	0.3771	< 0.001
	Automated_readability	1.1825	4.4910	4.2432	4.0083	0.3626	< 0.001
	Gunning_fog	4.9301	4.3180	6.8362	4.3812	0.3436	< 0.001
	Dale_chall_readability	6.7107	4.5744	9.1959	3.1029	0.2597	< 0.001
	Linsear_write	2.4029	1.4958	3.9192	1.9726	0.3359	< 0.001
PropertyCQ	Flesch_kincaid_grade	6.0499	4.1777	1.1599	7.4969	0.5381	< 0.001
	Coleman_liau	8.4990	5.6713	3.9016	9.1768	0.3069	< 0.001
	Automated_readability	6.6017	3.9172	4.8573	4.5096	0.2536	< 0.001
	Gunning_fog	8.6725	3.4422	5.6077	5.0341	0.4506	< 0.001
	Dale_chall_readability	8.2708	2.5979	8.8422	4.3962	0.2439	< 0.001
	Linsear_write	5.2887	2.0900	2.5093	2.1629	0.6004	< 0.001
ObjectCQ	Flesch_kincaid_grade	3.5927	3.1698	2.6648	6.6696	0.1449	< 0.001
	Coleman_liau	5.5235	5.3804	5.4988	8.0585	0.1294	< 0.001
	Automated_readability	4.0484	4.1828	5.2317	4.2196	0.1492	< 0.001
	Gunning_fog	5.5520	4.1198	6.5423	5.0143	0.1262	< 0.001
	Dale_chall_readability	7.3251	3.8126	9.3388	3.9111	0.2759	< 0.001
	Linsear_write	2.7394	1.1107	3.1702	2.1217	0.1863	< 0.001