

Characteristics and Desiderata for Competency Question Benchmarks^{*}

Reham Alharbi, Jacopo de Berardinis, Floriana Grasso, Terry R. Payne and
Valentina Tamma^{*}

Department of Computer Science, University of Liverpool, UK

Abstract

Competency Questions (CQs) are essential in ontology engineering; they express an ontology's functional requirements through natural language questions, offer crucial insights into an ontology's scope, and are pivotal for various tasks, such as ontology reuse, testing, requirement specification, and pattern definition. Various approaches have emerged that make use of LLMs for the generation of CQs from different knowledge sources. However, comparative evaluations are hindered by differences in tasks, datasets and evaluation measures used. In this paper, we provide a set of desiderata for a benchmark of CQs, where we position state of the art approaches with respect to a categorisation of tasks, and highlight the main challenges hindering the definition of a community-based benchmark to support comparative studies.

Keywords

Competency Questions, Large Language Models, Benchmark, Evaluation, Dataset

1. Introduction

One of the main bottlenecks in the ontology construction process is the elicitation and articulation of the requirements that are used during the initial phases of the ontology construction process. Competency Questions (CQs) [1] are central to many ontology engineering processes. CQs are questions expressed in natural language that characterise the scope of the knowledge represented by an ontology, and model the functional requirements that an ontology-based information system should satisfy to achieve its intended purpose. They bridge the gap between the domain expert's understanding of the subject matter, and the ontology engineer's formal representation of that knowledge.

CQs are used at various stages of the ontology development process:

- In the requirement definition stage of the ontology development process, CQs are used to suggest possible concepts and relationships to include in the ontology [2, 3, 4, 5];
- They are used to verify and validate the knowledge encapsulated in the ontology [6, 7, 8],
- They are used to support the consumption of ontology content, e.g. through the generation of APIs [9] and the reuse of ontological fragments [10, 11, 12].

Large Language Models (LLMs) and Generative AI have recently demonstrated remarkable capabilities in processing natural language within human-level tasks such as question generation and answering. Consequently, a number of approaches have been proposed to automate knowledge engineering activities (partially or in full), including the formulation of CQs [13, 14, 15, 16] and that differ with respect to the nature of the knowledge resources used. CQ generation approaches can be divided into:

ISWC 2024 Special Session on Harmonising Generative AI and Semantic Web Technologies, November 13, 2024, Baltimore, Maryland

^{*}Corresponding author.

✉ R.Alharbi@liverpool.ac.uk (R. Alharbi); jacodb@liverpool.ac.uk (J. d. Berardinis); floriana@liverpool.ac.uk (F. Grasso); T.R.Payne@liverpool.ac.uk (T. R. Payne); V.Tamma@liverpool.ac.uk (V. Tamma)

ORCID 0000-0002-8332-3803 (R. Alharbi); 0000-0001-8419-6554 (F. Grasso); 0000-0002-0106-8731 (T. R. Payne); 0000-0002-1320-610X (V. Tamma)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. **Reverse engineering of CQs from KGs:** here CQs are reversed engineered from sources of common sense open data, e.g. Wikidata [17] or DBPedia [18]. In this case, CQs are built in a bottom-up fashion, rather than being formulated through interviews with domain experts.
2. **Retrofitting CQs from ontologies:** this applies to cases where an ontology exists, but no associated CQs are publicly available. Therefore, the aim is to identify possible CQs that were used in the development of the ontology, thus facilitating its future reuse [13].
3. **Generating CQs from knowledge sources:** these are approaches that generate CQs from either a set of class and property names [16], or from a corpus of text describing a domain [14].

As more automatic, generative methods emerge, there is a growing need to develop resources that can be used to validate these approaches. This is notwithstanding the challenges arising from the relative scarcity of CQs (and requirements in general), that are: 1) recognised to be of good quality; and 2) are published together with the ontology whose design they have supported.

To date, few common resources have been used by different studies (one exception being Dem@Care [19], which has been used by several studies [13, 16]), and there is little consistency on the use of evaluation measures to assess the CQs. Furthermore, different studies have addressed different phases in the ontology development lifecycle, and thus cannot be directly compared. For example, some approaches target the *Requirement specification* phase [14, 15], whereas others address the context where ontologies have missing or non-existent CQs [13, 16].

In this paper, we address the problem of *identifying the requirements of a multi-purpose benchmark for competency question generation* together with its task specifications and evaluation criteria, that is the blueprint for a benchmark generation activity at the special session.

2. Towards a benchmark for CQs Formulation Approaches

The (semi-)automatic formulation of CQs should ensure that the questions generated are “good” competency questions. While there is no accepted definition of what “good” means in this context, we can leverage the literature on automatic question generation [20, 21] to identify desirable characteristics of “good” CQs, considering also that the generation of questions is a form of information-seeking activity that reveals the implicit connection between reasoning ability and language generation [22]. In the reminder of this section we identify the types of tasks that a CQ generation approach should support together with the resources needed to manage the tasks, namely: the data, the pre-processing steps and the evaluation measures.

2.1. Tasks definition

There is no consensus on how to assess the quality of competency question [23, 24], especially with respect to their aim of identifying the purpose and the explicit concepts and relations in an ontology. However, we can identify a set of criteria, that define the tasks that the benchmark should support [25]. We can broadly categorise CQs into the following categories: (i) **Syntactically or semantically incorrect CQs:** This category addresses common issues in question formulation that can hinder effective ontology modelling, with consequences for query processing and data retrieval [13, 26]; (ii) **Scoping CQs:** Such questions may help to define the domain, but do necessarily translate into a query that can be automatically be processed (e.g. a SPARQL query). These require specialised handling to aid the definition of a domain; (iii) **Verified CQs:** These CQs can be directly queried and can serve as benchmarks for system capabilities.

For each of these categories we identify those tasks that approaches for generating CQs should be able to manage:

Syntactically or semantically incorrect CQs:

1. Linguistic Perspectives:

- a) *Identify Ambiguous Questions*: Create a repository of CQ examples that exhibit ambiguity in wording or context. **Example**: “Which devices can I see?”¹ which is inherently ambiguous given its subjectivity;
 - b) *Develop Clarity Guidelines*: Formulate standards / templates to help rephrase ambiguous questions for improved clarity and specificity. **Example**: the CQ “What are the materials used for a barbecue?”² is inherently ambiguous, since materials here could be interpreted either as the tools (e.g. spatula, tongs) or as the specific material used to make the barbecue and its components (e.g. cast iron), and hence would benefit from being clarified.
2. Question Type Identification:
- a) *Classify Question Types*: Systematically categorise CQs into types such as narrative, factual, or descriptive [26, 27], and assess their suitability in different contexts. Narrative and descriptive questions are typically questions that require a subjective view on a topic, but could contribute to identify relevant knowledge. **Example**: “What is your favourite pizza topping”, which might be useful to some extent in defining a domain (e.g. the concept of popular pizza topping).
 - b) *Evaluate Contextual Appropriateness*: Develop criteria to measure the effectiveness of question types within their intended contexts. In some cases, this is needed to ensure that a CQ is consistent with the original ontology requirements; especially when *generating CQs from knowledge sources* (Section 1) given the potential availability of user stories, interviews, etc.
3. Domain Knowledge Relevance:
- a) *Align Questions with Domain Relevance*: Establish a review process to ensure questions are pertinent with respect to the relevant domain knowledge.
 - b) *Refine Focus Through Filtering*: Implement a mechanism to exclude questions that, while correct, are irrelevant to the task at hand. This is particularly useful in cases of CQs generated by some LLMs (e.g. Llama) that tend to formulate illustrative questions [13].
4. Incorrect or inappropriate CQs detection:
- a) *Correct Erroneous Inputs*: Introduce a correction mechanism for factually incorrect CQs, e.g. “Which vegetarian pizza contains ham?”. This can be used as a CQ only to confirm that there is no entity in the ontology that satisfies this question.
 - b) *Bias*: Set up a robust protocol for verifying and eliminating those CQs generated through generative AI that propagate or reinforce bias due to the pre-training of Large Language Models, which is particularly critical in domains such as healthcare [28], etc.

Scoping CQs:

1. *Catalogue Scoping CQs*: Document all CQs that contribute to defining the scope of the information domain [25, 24]. **Example**: “Which are the types of CheeseTopping?” [29]
2. *Analysis of Domain Contribution*: The analysis of how these CQs help in shaping the understanding of the domain. These can include definition or disambiguation questions or questions to state modelling choices. **Example**: “Is dialect a language” [30].
3. *Integration into Information Architecture*: Strategies that utilise scoping CQs for enhancing the structure of information repositories should be defined.

Verified CQs:

1. *Maintaining databases of Verified CQs*: An up-to-date list of CQs that can be directly transformed into SPARQL queries needs to be maintained together with their SPARQL formulation. Nonetheless, as long as this is well-documented, even CQs expressing requirement that are not (yet)

¹This is req223 for the Vicinity Core ontology listed in the CORAL repository [8].

²<https://keet.wordpress.com/2022/06/08/only-answering-competency-questions-is-not-enough-to-evaluate-your-ontology/>

supported by the ontology can still be of interest for evaluation. For example Zhang et al. define those as *adversarial CQs*, and use them for ontology testing.

2. *Testing and Validation*: Rigorous testing is necessary to ensure that the SPARQL queries (corresponding to the CQs) retrieve accurate and relevant data. Some of these tests can be automated through dedicated tools, for example OWLunit³, a tool that runs unit tests for ontologies, or OOPS!, the Ontology Pitfall Scanner, that detects common errors in ontologies [32];
3. *Documentation and Examples*: Create detailed documentation and examples of successful CQ transformations for training and reference, possibly through tool support (e.g. Widoco[33]).

2.2. Datasets available

We propose a competency question benchmark, CQ-BEN⁴, comprising a corpus of competency questions that have either been curated to support the validation of ontology engineering processes or have been used to construct ontologies supporting some downstream task, e.g. the Polifonia ontology network [34].

Collecting a suitable dataset to support the tasks defined above is not trivial: often ontologies are published without the CQs and the requirements used to design them. As a result, open-source repository data often lack essential components, especially to support the design and testing. We identify two main implementation steps to organise the process; (i) **Gathering all Published Requirements**: Collecting and documenting all existing requirements related to tasks, in a similar effort to repositories such as CORAL [8] and the CQs dataset [23], along with individual ontologies that have published their CQs; (ii) **Categorisation According to Tasks**: Organising the requirements based on the respective tasks they support in order to streamline the benchmark design process. As part of the contribution to this challenge, we have collected a preliminary repository of resources consisting of ontologies, related competency questions and the relevant publications describing these resources. This resource is contributed to the community and is open to extension and improvement.

2.3. Ontology pre-processing

Depending on the static context and other information that is given as input, some ontology pre-processing might be necessary prior to feeding all data to a computational model for CQ extraction. Approaches that extract triples [27] need to handle the possible presence of blank nodes, e.g. by *projecting* an ontology into a simplified graph representation [35]. Ontology verbalisation translates formal ontology structures into natural language [36], and is often used as a preprocessing step. Ontology verbalisation is the process of translating formal ontology structures into natural language expressions. As such, the verbalisation strategy impacts all pipelines that process textual/narrative ontology descriptions. Different strategies have been used, such as triple-based verbalisation [15, 37] or descriptive ontology verbalisation [31], while some approaches skip verbalisation and feed triples directly [27]. Measuring the contribution of ontology verbalisation remains an open direction.

In the context of the benchmark, given that ontology pre-processing impacts CQ extraction, we would expect this to introduce an additional dimension in an experimental setup. For example, if a method uses verbalisation, accounting for this dimension would allow to address the following research questions: “How sensitive is a CQ formulation pipeline to verbalisation?”, “Which verbalisation technique/methodology yields the best performance for CQ?”, “How does a CQ formulation pipeline perform without verbalisation?”.

2.4. Evaluation approaches

Different evaluation approaches have been proposed for the tasks identified in Section 2.1, which complicates further the effort of understanding the performance of CQ generation algorithms. The evaluation measures include both CQ assessment and performance measures:

³<https://github.com/luigi-asprino/owl-unit>

⁴<https://github.com/KE-UniLiv/CQ-benchmark/>

Evaluation approach	Task
Expert evaluation	Linguistic perspective tasks Question type identification tasks Domain Knowledge Relevance tasks Incorrect or inappropriate CQ detection tasks Analysis of domain contribution task Integration into information architecture Database of verified CQs task, and Documentation and example creation task
Similarity assessment	Database of verified CQs task Cataloguing and scoping CQ task,
Testing for verified CQs	Testing and validation task

Table 1

Evaluation approaches with the assessed tasks

- **Expert Evaluation:** These measures typically relate to tasks that identify poor or incorrect CQs, and assess their relevance and accuracy. This type of evaluation is generally performed with the support from domain experts and knowledge engineers, and as such, is particularly time consuming and prone to subjectivity and potential bias. Nonetheless, these approaches typically provide useful insights into the behaviour of the computational models generating CQs, and often result in data collection activities that support the use of automatic metrics[15];
- **Similarity Assessment:** Techniques based on text embeddings, such as Sentence BERT (SBERT) [38], are often employed for assessing the similarity between generated and ground-truth CQs, and only pairs of generated and ground-truth CQs whose similarity is above a threshold (often 70% or above) are considered similar. In turn, this enables the computation of performance metrics such as *accuracy*, *precision*, *recall*, and *F1-scores* [27, 16] for tasks that involve identifying scoping CQs. Computing the cosine similarity between sentence-level (text) embeddings is often used as a proxy to detect paraphrasing [34, 27], i.e. when two CQs have the same meaning but are formulated differently, e.g. “*What is the number of the moons of Jupiter?*” “*How many moons does Jupiter have?*”. However, as this may be prone to false positives and false negatives (high cosine similarity, different meaning; low cosine similarity, same meaning), other approaches determine CQ equivalence through transfer learning [15]. In this case, given two questions, pairs of sentence-level embeddings are fed to a feed-forward neural network and trained for paraphrase detection using related corpora [39].
- **Testing for Verified CQs:** This involves computing the similarity between CQs and developing unit/acceptance testing for the corresponding SPARQL queries [40]. Performance measures vary from the ones used to assess similarity between CQs to those used in Natural Language Processing (NLP), e.g. the BiLingual Evaluation Understudy (BLEU) score [41] used for assessing automatic text translation [15].
- **Emergent approaches:** Other approaches are emerging from different fields (typically question generation for education) that aim to assess the complexity of generated questions using a combination of predefined templates and complexity similarity [42, 43].

Table 1 relates the evaluation approach to specific tasks. The list of approaches and tasks is not exhaustive, and further approaches tailored to the reuse or adaptation of ontologies will be developed.

2.5. Conclusion

In this paper, we identified the requirements for a multi-purpose benchmark for competency question generation approaches together with its task specifications and evaluation criteria, that lays the foundations for a comprehensive benchmark.

References

- [1] M. Grüninger, M. S. Fox, *The Role of Competency Questions in Enterprise Engineering*, Springer US, 1995, pp. 22–31.
- [2] N. F. Noy, D. L. McGuinness, *Ontology development 101: A guide to creating your first ontology*, Technical Report, Stanford knowledge systems laboratory technical report KSL-01-05, 2001.
- [3] V. Presutti, E. Daga, A. Gangemi, E. Blomqvist, *Extreme design with content ontology design patterns*, in: *Proc. of the 2009 International Conf. on Ontology Patterns*, volume 516, 2009, p. 83–97.
- [4] J. F. Sequeda, W. J. Briggs, D. P. Miranker, W. P. Heideman, *A pay-as-you-go methodology to design and build enterprise knowledge graphs from relational databases*, in: *Proc. of the 18th International Semantic Web Conf., ISWC 2019*, 2019, pp. 526–545.
- [5] M. C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, *The neon methodology framework: A scenario-based methodology for ontology development*, *Applied ontology* 10 (2015) 107–145.
- [6] C. Bezerra, F. Freitas, *Verifying description logic ontologies based on competency questions and unit testing*, in: *Proc. of the IX Seminar on Ontology Research and I Doctoral and Masters Consortium on Ontologies*, volume 1908, 2017, pp. 159–164.
- [7] C. M. Keet, A. Ławrynowicz, *Test-driven development of ontologies*, in: *Proc. of the 13th International Conf. on The Semantic Web, ESWC 2016*, 2016, pp. 642–657.
- [8] A. Fernández-Izquierdo, M. Poveda-Villalón, R. García-Castro, *CORAL: A corpus of ontological requirements annotated with lexico-syntactic patterns*, in: *Proc. of the 16th International Conf. on The Semantic Web, ESWC 2019*, Springer International Publishing, 2019, pp. 443–458.
- [9] P. Espinoza-Arias, D. Garijo, O. Corcho, *Extending ontology engineering practices to facilitate application development*, in: *Knowledge Engineering and Knowledge Management*, Springer International Publishing, 2022, pp. 19–35.
- [10] R. Alharbi, *Assessing candidate ontologies for reuse*, in: *Proc. of the Doctoral Consortium at ISWC 2021 (ISWC-DC)*, 2021, pp. 65–72. URL: <https://api.semanticscholar.org/CorpusID:244895203>.
- [11] R. Alharbi, V. Tamma, F. Grasso, *Requirement-based methodological steps to identify ontologies for reuse*, in: *Intelligent Information Systems*, Springer Nature Switzerland, 2024, pp. 64–72.
- [12] S. Azzi, A. Assi, S. Gagnon, *Scoring ontologies for reuse: An approach for fitting semantic requirements*, in: *Proc. of the Research Conf. on Metadata and Semantic Research, MTSR 2022*, Springer Nature, 2023, pp. 203–208.
- [13] R. Alharbi, V. Tamma, F. Grasso, T. Payne, *An experiment in retrofitting competency questions for existing ontologies*, in: *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC '24*, Association for Computing Machinery, 2024, p. 1650–1658.
- [14] M. Antia, C. M. Keet, *Automating the generation of competency questions for ontologies with agocqs*, in: *Knowledge Graphs and Semantic Web*, Springer Nature Switzerland, 2023, pp. 213–227.
- [15] F. Ciroku, J. de Berardinis, J. Kim, A. Meroño-Peñuela, V. Presutti, E. Simperl, *Revont: Reverse engineering of competency questions from knowledge graphs via language models*, *Journal of Web Semantics* 82 (2024) 100822.
- [16] Y. Rebboud, L. Tailhardat, P. Lisena, R. Troncy, *Can LLMs generate competency questions?*, in: *Extended Semantic Web Conference, ESWC2024*, Hersionissos, Greece, 2024.
- [17] D. Vrandečić, M. Krötzsch, *Wikidata: a free collaborative knowledgebase*, *Communications of the ACM* 57 (2014) 78–85.
- [18] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, *Dbpedia: A nucleus for a web of open data*, in: *international semantic web conference*, Springer, 2007, pp. 722–735.
- [19] S. Dasiopoulou, G. Meditskos, V. Efstathiou, *Semantic Knowledge Structures and Representation*, Technical Report D5.1, FP7-288199 Dem@Care: Dementia Ambient Care: Multi-Sensing Monitoring for Intelligence Remote Management and Decision Support, 2012. URL: http://www.demcare.eu/downloads/D5.1SemanticKnowledgeStructures_andRepresentation.pdf.
- [20] G. K. Q. Monfardini, J. S. Salamon, M. P. Barcellos, *Use of competency questions in ontology*

- engineering: A survey, in: *Conceptual Modeling*, Springer Nature Switzerland, 2023, pp. 45–64.
- [21] N. Mulla, P. Gharpure, Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications, *Prog. Artif. Intell.* 12 (2023) 1–32.
 - [22] L. Bertolazzi, D. Mazzaccara, F. Merlo, R. Bernardi, ChatGPT’s information seeking strategy: Insights from the 20-questions game, in: C. M. Keet, H.-Y. Lee, S. Zarri   (Eds.), *Proceedings of the 16th International Natural Language Generation Conference*, Association for Computational Linguistics, Prague, Czechia, 2023, pp. 153–162. URL: <https://aclanthology.org/2023.inlg-main.11>. doi:10.18653/v1/2023.inlg-main.11.
 - [23] J. Potoniec, D. Wi  niewski, A.   wrynowicz, C. M. Keet, Dataset of ontology competency questions to sparql-owl queries translations, *Data in Brief* 29 (2020) 105098. URL: <https://www.sciencedirect.com/science/article/pii/S2352340919314544>. doi:<https://doi.org/10.1016/j.dib.2019.105098>.
 - [24] C. M. Keet, Z. C. Khan, On the roles of competency questions in ontology engineering, in: *24th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, Springer, 2024 – to appear.
 - [25] R. Alharbi, V. Tamma, T. Payne, F. Grasso, A review and comparison of competency question engineering approaches, in: *24th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, Springer, 2024.
 - [26] M. Antia, C. M. Keet, Assessing and enhancing bottom-up CNL design for competency questions for ontologies, in: *Proc. of the Seventh International Workshop on Controlled Natural Language (CNL 2020/21)*, Association for Computational Linguistics (ACL), 2021, pp. 1–11.
 - [27] R. Alharbi, V. Tamma, F. Grasso, T. Payne, The role of Generative AI in competency question retrofitting, in: *Extended Semantic Web Conference, ESWC2024*, Hersonissos, Greece, 2024.
 - [28] S. Chen, J. Gallifant, M. Gao, P. Moreira, N. Munch, A. Muthukkumar, A. Rajan, J. Kolluri, A. Fiske, J. Hastings, H. Aerts, B. Anthony, L. A. Celi, W. G. L. Cava, D. S. Bitterman, Cross-care: Assessing the healthcare implications of pre-training data on language model bias, 2024. URL: <https://arxiv.org/abs/2405.05506>. arXiv:2405.05506.
 - [29] C. Bezerra, F. Santana, F. Freitas, CQChecker: A tool to check ontologies in OWL-DL using competency questions written in controlled natural language, *Learning & Nonlinear Models* 12 (2014) 115–129.
 - [30] F. Gillis-Webber, S. Tittel, C. M. Keet, A model for language annotations on the web, in: B. Villaz  n-Terrazas, Y. Hidalgo-Delgado (Eds.), *Knowledge Graphs and Semantic Web - First Iberoamerican Conference, KGSWC 2019*, Villa Clara, Cuba, June 23–30, 2019, *Proceedings*, volume 1029 of *Communications in Computer and Information Science*, Springer, 2019, pp. 1–16. URL: https://doi.org/10.1007/978-3-030-21395-4_1. doi:10.1007/978-3-030-21395-4_1.
 - [31] B. Zhang, V. A. Carriero, K. Schreiberhuber, S. Tsaneva, L. S. Gonz  lez, J. Kim, J. de Berardinis, Ontochat: a framework for conversational ontology engineering using language models, arXiv preprint arXiv:2403.05921 (2024).
 - [32] M. Poveda-Villal  n, A. G  mez-P  rez, M. C. Su  rez-Figueroa, OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation, *International Journal on Semantic Web and Information Systems (IJSWIS)* 10 (2014) 7–34.
 - [33] D. Garijo, Widoco: A wizard for documenting ontologies, in: C. d’Amato, M. Fernandez, V. Tamma, F. Lecue, P. Cudr  -Mauroux, J. Sequeda, C. Lange, J. Heflin (Eds.), *The Semantic Web – ISWC 2017*, Springer International Publishing, Cham, 2017, pp. 94–102.
 - [34] J. de Berardinis, V. A. Carriero, N. Jain, N. Lazzari, A. Mero  o-Pe  uela, A. Poltronieri, V. Presutti, The polifonia ontology network: Building a semantic backbone for musical heritage, in: *The Semantic Web – ISWC 2023*, Springer Nature Switzerland, 2023, pp. 302–322.
 - [35] Y. He, J. Chen, H. Dong, I. Horrocks, C. Allocca, T. Kim, B. Sapkota, Deeponto: A python package for ontology engineering with deep learning, *Semantic Web: Interoperability, Usability, Applicability* (2024).
 - [36] Y. He, J. Chen, E. Jimenez-Ruiz, H. Dong, I. Horrocks, Language model analysis for ontology subsumption inference, in: *Findings of the Association for Computational Linguistics: ACL 2023*,

2023, pp. 3439–3453.

- [37] G. Amaral, O. Rodrigues, E. Simperl, Wdv: A broad data verbalisation dataset built from wikidata, in: *The Semantic Web – ISWC 2022*, Springer International Publishing, Cham, 2022, pp. 556–574.
- [38] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: *Proc. of the 2019 Conf. on Empirical Methods in Natural Language Proc. and the 9th International Joint Conf. on Natural Language Proc. (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019, pp. 3982–3992.
- [39] Z. Wang, W. Hamza, R. Florian, Bilateral multi-perspective matching for natural language sentences, *arXiv preprint arXiv:1702.03814* (2017).
- [40] M. Poveda-Villalón, A. Fernández-Izquierdo, M. Fernández-López, R. García-Castro, LOT: An industrial oriented ontology engineering framework, *Engineering Applications of Artificial Intelligence* 111 (2022) 104755.
- [41] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: P. Isabelle, E. Charniak, D. Lin (Eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318. URL: <https://aclanthology.org/P02-1040>. doi:10.3115/1073083.1073135.
- [42] S. AlKhuzaey, F. Grasso, T. Payne, V. Tamma, A framework for assessing the complexity of auto generated questions from ontologies, in: *European Conference on e-Learning*, volume 22, 2023, pp. 17–24.
- [43] S. Bi, X. Cheng, Y.-F. Li, L. Qu, S. Shen, G. Qi, L. Pan, Y. Jiang, Simple or complex? complexity-controllable question generation with soft templates and deep mixture of experts model, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 4645–4654. URL: <https://aclanthology.org/2021.findings-emnlp.397>. doi:10.18653/v1/2021.findings-emnlp.397.