FrODO: Generating Ontologies from Competency Questions in One Click

Aldo Gangemi[1,2], Anna Sofia Lippolis[1,*], Giorgia Lodi[1] and Andrea Giovanni Nuzzolese[1,†]

[1]*Institute of Cognitive Sciences and Technologies, Italian National Research Council (ISTC-CNR), Via San Martino della Battaglia 44, 00184, Rome, Italy*

[2]*Department of Classical and Italian Philology, University of Bologna, Via Zamboni 32, 40126, Bologna, Italy*

Abstract

In this paper we demonstrate the potentiality of the Frame-based Ontology Design Outlet (FrODO), a tool that relies on FRED for generating RDF from the natural language. FrODO exploits frame semantics for tailoring sound OWL ontologies addressing the given CQs. We envision that FrODO can be used to make agile ontology design methodologies (e.g. XD, SAMOD, etc.) smoother.

Keywords

Ontology Engineering, Frame semantics, Machine Reading, Knowledge graph, Knowledge representation

1. Introduction

Competency questions [1] (CQs) represent requirements in a number of agile ontology engineering methodologies, such as the eXtreme Design [2] (XD) or SAMOD [3]. In such methodologies most effort lies in the design of ontology modules able to address the CQs that have been previously identified. In this paper we demonstrate the Frame-based Ontology Design Outlet (FrODO). FrODO is a novel method and Web tool for automatically drafting OWL ontologies from CQs. FrODO builds on and benefits from FRED [4] a *machine reading* [5] tool aimed at gathering RDF from text written in natural language. FRED in fact produces RDF graphs from text that are (i) domain- and task-independent, and (ii) designed according to the frame semantics [6] and ontology design patterns [7]. In essence, FrODO extends FRED specifically on the case of CQs by tailoring the RDF produced by FRED into domain ontologies. This is done by leveraging its formal representation based on the frame semantics. The domain ontologies produced by FrODO are drafts that can be used to feed agile ontology design methodologies. In this paper we demonstrate FrODO as a tool for generating domain ontologies formalised as OWL from competency questions automatically. This demo paper is associated with [8], which

SEMANTICS 2022 EU: 18th International Conference on Semantic Systems, September 13-15, 2022, Vienna, Austria

*Corresponding author.

†The authors are sorted alphabetically as they equally contributed to this paper.

aldo.gangemi@cnr.it (A. Gangemi); annasofia.lippolis@istc.cnr.it (A. S. Lippolis); giorgia.lodi@cnr.it (G. Lodi); andreagiovanni.nuzzolese@cnr.it (A. G. Nuzzolese)

0000-0001-5568-2684 (A. Gangemi); 0000-0002-0266-3452 (A. S. Lippolis); 0000-0001-6020-5874 (G. Lodi); 0000-0003-2928-9496 (A. G. Nuzzolese)

was accepted for publication to the Research & Innovation track of SEMANTiCS 2022.

2. Related work

Many solutions for generating ontologies from text address an Ontology Learning and Population (OL&P) task [9, 10]. Examples of such methods include Text2Onto [11], OwlExporter [12], and the approach proposed by Tanev and Magnini [13]. Most of these solutions are implemented on top of machine learning methods. Hence, they are typically data hungry, i.e. they require large corpora, sometimes manually annotated, in order to learn rules for ontology automatic construction. Such rules are defined through a training phase that can take a long time. On the contrary, our solution does not depend on any training and is unsupervised. Other approaches to OL&P use either lexico-syntactic patterns [14], or hybrid lexical-logical techniques [15]. Finally, there are solutions for generating SPARQL queries from CQs, such as [16]. The latter can be seen as a similar and complementary task to the generation of OWL ontologies from text. FrODO differs from all these solutions at the state of the art as it relies on machine reading as the paradigm for achieving the automatic construction of domain ontologies from competency questions.

3. FrODO

FrODO is a web-based application[1] that generates ontologies from CQs. This is done by extending FRED that uses frame semantics [6] for organising domain independent RDF graphs generated from text. Hence, FrODO builds on top of FRED for refactoring its RDF graph by means of graph traversal strategies that exploit the frame semantics. In this we assume that frames and frame arguments are the key tools to leverage on for drawing domain-relevant boundaries around the RDF graph produced by FRED, thus enabling the generation of domain ontology drafts. Two refactoring strategies are implemented by FrODO in order to cope with the possible frame representation patterns available in FRED, i.e. (i) n-ary relations and (ii) periphrastic relations. We refer the interested readers to [4] for more details about FRED and [8] for more details about FrODO. In this paper we aim at demonstrating how FrODO can support ontology engineers and practitioners to generate an ontology from CQs automatically. Figure 1 shows the user interface provided by FrODO. It consists of a main text area meant for entering a CQ to be processed by FrODO and an input text for specifying the IRI to be used as the identifier of the resulting ontology. The former input is mandatory, while the latter is optional, i.e. if no ontology identifier is specified then a default one is used[2].

To demonstrate FrODO we use the CQ "*Who commissioned a component of a system?*" as a running example. The CQ is about the design on an ontology related to the commissioning of systems and their components. Such a CQ comes from a real world set of CQs we defined in the context of the WHOW project[3] for representing the requirements of an ontology network about systems for environmental monitoring. When a user provides the previous CQ and clicks

[1] FrODO is available online at https://w3id.org/stlab/frodo.
[2] The ontology identifier used as default by FrODO is https://w3id.org/stlab/ontology/.
[3] https://whowproject.eu/

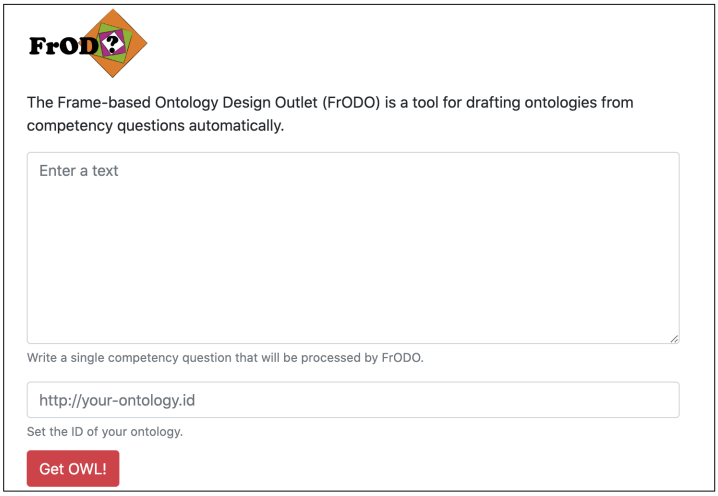

Figure 1: The user interface of FrODO.

on the button labelled "Get OWL!", FrODO starts its processing, which is summarised by the UML activity diagram depicted in Figure 2.

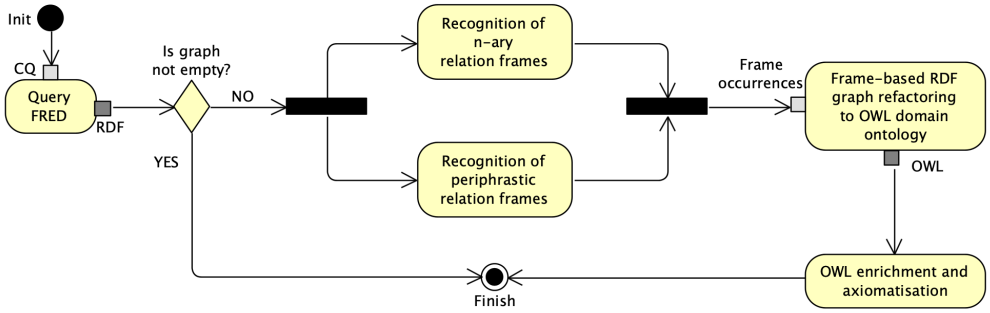

Figure 2: Process implemented by FrODO. In this UML activity diagrams the yellow boxes represent activities, while the black circle and encircled black circle represent the initial and final nodes, respectively. The arrows among activities represent the direction of the workflow execution. Finally, the light-gray and dark-gray boxes pinned on the activities identifies the objects required by the activities as input and output, respectively.

The first step of the worflow requires FrODO to query FRED in order to get an RDF graph from the text of the CQ. Then the frames are recognised in the graph either in the form of n-ary or periphrastic relations. Once the frame occurrences are recognised the frame arguments with their associated roles are used for generating an ontology draft with domain terminology. This draft is finally enriched with annotations (i.e. RDFS labels), inverse properties, and restriction axioms. Listing 1 reports the ontology returned by FrODO to a user who provided the CQ expressed as our running example. The ontology is serialised according to the Manchester syntax.

Listing 1: OWL ontology produced for the CQ "*Who commissioned a component of a system?*".

```
ObjectProperty: involvesComponent
     Annotations: rdfs:label "involves component"
     Range: Component
     InverseOf: isComponentInvolvedIn
ObjectProperty: involvesPerson
    Annotations: rdfs:label "involves person"
    Range: Person
    InverseOf: isPersonInvolvedIn
ObjectProperty: isComponentInvolvedIn
    Annotations rdfs:label "is component involved in"
    Domain: Component
ObjectProperty: isPersonInvolvedIn
    Annotations: rdfs:label "is person involved in"
    Domain: Person
ObjectProperty: componentOfSystem
    Annotations: rdfs:label "component of system"
    Range: System
    InverseOf: isComponentOfSystemOf
ObjectProperty: isComponentOfSystemOf
    Annotations: rdfs:label "is component of system of"
    Domain: System
Class: System
    Annotations:  rdfs:label "System"@en
Class: Commissioning
    Annotations: rdfs:label "Commissioning"@en
    SubClassOf: componentOfSystem some System
Class: Component
    Annotations: rdfs:label "Component"
Class: ComponentCommissioning
    Annotations: rdfs:label "Component commissioning"@en
    SubClassOf: Commissioning, involvesComponent some Component, involvesPerson some Person
Class: Person
    Annotations: rdfs:label "Person"
```

4. Conclusions

In this work we demonstrate the Frame-based Ontology Design Outlet (FrODO), which is a tool able to draft ontologies from competency questions (CQs) automatically. FrODO extends FRED by leveraging frame semantics to gather domain knowledge from the RDF graph produced by FRED from textual CQs. FrODO can be used by ontology engineers to make agile ontology design methodologies (e.g. XD, SAMOD, etc.) smoother.

Acknowledgments

This work was partially supported by the European Commission (EC) through the Connecting Europe Facility (CEF) programme under the WHOW project WHOW (grant agreement no. INEA/CEF/ICT/A2019/2063229).

References

[1] M. Grüninger, M. S. Fox, The role of competency questions in enterprise engineering, in: Benchmarking—Theory and practice, Springer, 1995, pp. 22–31.

[2] V. Presutti, E. Daga, A. Gangemi, E. Blomqvist, extreme design with content ontology design patterns, in: E. Blomqvist, K. Sandkuhl, F. Scharffe, V. Svátek (Eds.), Proc. of WOP 2009, volume 516 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2009.

[3] S. Peroni, A simplified agile methodology for ontology development., in: M. Dragoni, M. Poveda-Villalón, E. Jiménez-Ruiz (Eds.), Proc of OWLED 2016, volume 10161 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 55–69. DOI: 10.6084/M9.FIGSHARE.3189769.V2.

[4] A. Gangemi, V. Presutti, D. R. Recupero, A. G. Nuzzolese, F. Draicchio, M. Mongiovì, Semantic web machine reading with fred., Semantic Web 8 (2017) 873–893. DOI: 10.3233/SW-160240.

[5] O. Etzioni, M. Banko, M. J. Cafarella, Machine reading, in: AAAI Spring Symposium: Machine Reading, AAAI, 2007, pp. 1–55. DOI: 10.5555/1597348.1597430.

[6] C. J. Fillmore, et al., Frame semantics, Cognitive linguistics: Basic readings 34 (2006) 373–400.

[7] A. Gangemi, V. Presutti, Ontology design patterns, in: Handbook on ontologies, Springer, 2009, pp. 221–243.

[8] A. Gangemi, A. S. Lippolis, G. Lodi, A. G. Nuzzolese, automatically drafting ontologies from competency questions with frodo, in: SEMANTiCS 2022, ????, pp. 1–15.

[9] P. Cimiano, Ontology learning and population from text: algorithms, evaluation and applications, volume 27, Springer Science & Business Media, 2006.

[10] F. N. Al-Aswadi, H. Y. Chan, K. H. Gan, Automatic ontology construction from text: a review from shallow to deep learning trend, Artificial Intelligence Review 53 (2020) 3901–3928. DOI: 10.1007/s10462-019-09782-9.

[11] P. Cimiano, J. Völker, A framework for ontology learning and data-driven change discovery, in: Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB), Springer, 2005, pp. 227–238. DOI: 10.1007/11428817_21.

[12] R. Witte, N. Khamis, J. Rilling, Flexible ontology population from text: The owlexporter, in: Proceedings of LREC 2010, 2010.

[13] H. Tanev, B. Magnini, Weakly supervised approaches for ontology population, in: Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics, 2006, pp. 17–24.

[14] M. Hearst, Automatic acquisition of hyponyms from large text corpora in proc, in: 14th International Conference Computational Linguistics, Nantes France, 1992.

[15] J. Völker, S. Rudolph, Lexico-logical acquisition of owl dl axioms - an integrated approach to ontology refinement, in: Proceedings of ICFCA 2008, volume 4933 of *Lecture Notes in Artificial Intelligence*, 2008.

[16] D. Wisniewski, Automatic translation of competency questions into sparql-owl queries, in: Companion Proceedings of the The Web Conference 2018, 2018, pp. 855–859. DOI: 10.1145/3184558.3186575.