Computing Authoring Tests from Competency Questions: Experimental Validation

Matt Dennis
University of Portsmouth

Kees Van Deemter
University of Aberdeen

Daniele Dell'Aglio
University of Zurich

Jeff Z. Pan
University of Aberdeen

Abstract

This paper explores whether Authoring Tests derived from Competency Questions accurately represent the expectations of ontology authors. In earlier work we proposed that an ontology authoring interface can be improved by allowing the interface to test whether a given Competency Question (CQ) is able to be answered by the ontology at a given stage of its construction, an approach known as CQ-driven Ontology Authoring (CQOA). The experiments presented in the present paper suggest that CQOA's understanding of CQs matches users' understanding quite well, especially for inexperienced ontology authors.

Author's post-print copy

This is a post-peer-review, pre-copyedit version of an article published in Lecture Notes in Computer Science, vol 10587. The final authenticated version is available online at: `http://dx.doi.org/10.1007/978-3-319-68288-4_15`

1 Introduction

Ontology authoring. Formal ontologies have become a widely accepted vehicle for representing knowledge in a range of domains, where they offer precise explanations of key terminologies. Many of these ontologies are formulated in terms of Description Logic (DL, [2]), a family of formalisms based on decidable fragments of First-Order Logic (FOL).

Examples include the medical SNOMED CT ontology[1], and the ontologies of the Open Biomedical Ontologies Consortium (GO [1], MGED [32]). The W3C standard Web Ontology Language (OWL 2) uses DLs as its underpinnings.

However, the precision offered by DL comes at a cost. Despite the existence of sophisticated ontology authoring interfaces such as *Protégé* [17], users and

[1] Cf. `http://www.ihtsdo.org/snomed-ct/`

developers frequently fail to comprehend important implications of the information contained in the ontology [26, 8]. In some cases, particular DL constructs are to blame (such as DL's use of the universal quantifier); in other cases, the main difficulty is to combine a large number of individually simple propositions and to establish their combined reasoning consequences.

Competency Questions and CQOA. These challenges have led to the notion of a Competency Question (CQ) [12]: a question that, in the opinion of the developer, the finished ontology should be able to answer. Initially, CQs were mainly used as a "pencil and paper" tool for ontology authors (henceforth: authors): the idea is to encourage authors to formulate a number of CQs at the start of the authoring process. For example, a CQ for a restaurant domain might ask: "What is the price of asparagus soup?". The idea is that listing such CQs can help to make authors aware of what information they need to encode.

Recently a number of authors have proposed that CQs should become part of the authoring interface.

One approach, which comes to terms with a particularly wide range of CQs, was Ren et al.'s [27], where we proposed CQ-driven Ontology Authoring (CQOA), in which the authoring interface checks continually, during authoring, which of the CQs are *handled correctly* by the ontology.

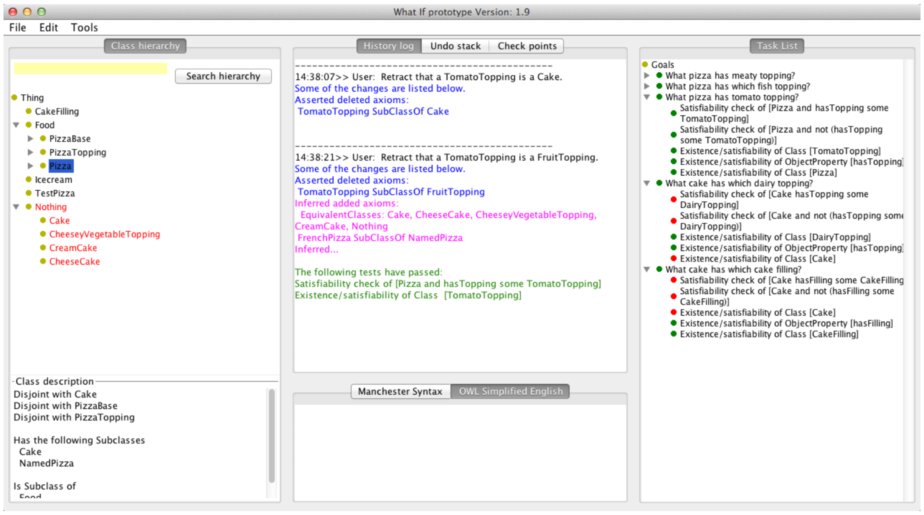
In formalising what it means to handle a CQ correctly, we draw on a key concept in linguistics, called *presupposition* (e.g., [18]). A presupposition of a *declarative* sentence is a proposition whose truth is a precondition to assessing the truth or falsity of the sentence: if the presupposition does not hold, the sentence is neither true nor false. Applied to a *question*, a presupposition is a proposition that needs to be true in order for the question to have an answer. For example, the question "What is the price of the cutlery?", when asked in a restaurant, presupposes that cutlery is on sale in that restaurant: if it is not, then the question cannot be answered. We argued that the idea of a failed presupposition (i.e., a question presupposing a falsehood) captures what happens when an ontology is unable to answer a CQ.

[27] contains an empirical study into the kinds of CQs that ontology authors tend to ask, yielding a set of CQ *archetypes*, see Table 1 on page 6 (and their *subtypes*, see Table 2 on page 7). Next, each type of presuppositions was mapped to some Authoring Tests (ATs), each of which is testable using satisfiability checking or subsumption checking services in the $\mathcal{ALCQ}$ DL. For example, the CQ "Which pizzas contain chocolate?" (called a *Selection Question*) triggers the following ATs:

> **Positive Presupposition**: $Pizza \sqcap \exists contains.Choc$ is satisfiable
> **Complement Presupposition**: $Pizza \sqcap \forall contains.\neg Choc$ is satisfiable

The first formula denotes the set of things that are pizzas and contain chocolate. By using the standard logical notion of satisfiability (e.g., [10]), the *Positive Presupposition* asserts not simply that the above-mentioned set is non-empty, but that the ontology *permits* it to be non-empty (i.e., its emptiness does not follow logically from the ontology). The second formula denotes the set of things that are pizzas and do *not* contain chocolate; the *Complement Presupposition* asserts that the ontology permits this set to be non-empty. If both presuppositions hold, it is possible for the ontology to contain pizzas that contain chocolate and ones that do not. Complement Presuppositions are less often discussed than Positive ones (though see e.g., [34]), suggesting that they might be less firmly associated with the sentences in question; we return to this issue in Section 3. CQOA is potentially powerful because it can help ontology authors to understand, at every stage of the authoring process, whether each of the CQs that they have specified is handled *correctly* by the ontology they are constructing.

Use in an Authoring Tool. We are incorporating CQs and the checking of their presuppositions into an ontology authoring tool. This tool uses a natural language-like dialogue as the main mode of interaction, using the controlled natural language OWL Simplified English (OSE) [23].



Thing
- CakeFilling
- Food
  - PizzaBase
  - PizzaTopping
  - Pizza
- Icecream
- TestPizza
- Nothing
  - Cake
  - CheeseyVegetableTopping
  - CreamCake
  - CheeseCake

Class description
Disjoint with Cake
Disjoint with PizzaBase
Disjoint with PizzaTopping

Has the following Subclasses
Cake
NamedPizza

Is Subclass of
Food

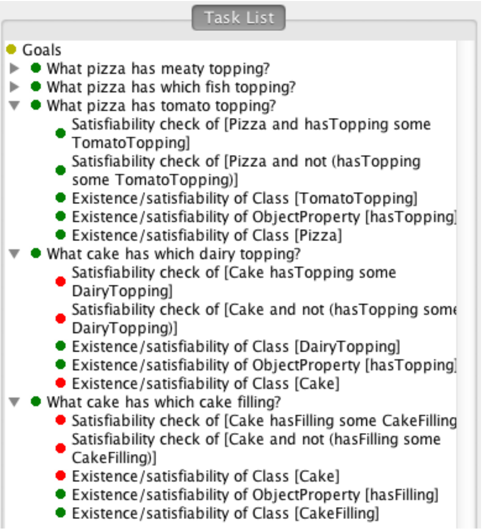
```
--------------------------------------------
14:38:07>> User:  Retract that a TomatoTopping is a Cake.
Some of the changes are listed below.
Asserted deleted axioms:
 TomatoTopping SubClassOf Cake


--------------------------------------------
14:38:21>> User:  Retract that a TomatoTopping is a FruitTopping.
Some of the changes are listed below.
Asserted deleted axioms:
 TomatoTopping SubClassOf FruitTopping
Inferred added axioms:
  EquivalentClasses: Cake, CheeseCake, CheeseyVegetableTopping,
CreamCake, Nothing
 FrenchPizza SubClassOf NamedPizza
Inferred...

The following tests have passed:
Satisfiability check of [Pizza and hasTopping some TomatoTopping]
Existence/satisfiability of Class  [TomatoTopping]
```

● Goals
- ● What pizza has meaty topping?
- ● What pizza has which fish topping?
- ● What pizza has tomato topping?
  - ● Satisfiability check of [Pizza and hasTopping some TomatoTopping]
  - ● Satisfiability check of [Pizza and not (hasTopping some TomatoTopping)]
  - ● Existence/satisfiability of Class [TomatoTopping]
  - ● Existence/satisfiability of ObjectProperty [hasTopping]
  - ● Existence/satisfiability of Class [Pizza]
- ● What cake has which dairy topping?
  - ● Satisfiability check of [Cake hasTopping some DairyTopping]
  - ● Satisfiability check of [Cake and not (hasTopping some DairyTopping)]
  - ● Existence/satisfiability of Class [DairyTopping]
  - ● Existence/satisfiability of ObjectProperty [hasTopping]
  - ● Existence/satisfiability of Class [Cake]
- ● What cake has which cake filling?
  - ● Satisfiability check of [Cake hasFilling some CakeFilling
  - ● Satisfiability check of [Cake and not (hasFilling some CakeFilling)]
  - ● Existence/satisfiability of Class [Cake]
  - ● Existence/satisfiability of ObjectProperty [hasFilling]
  - ● Existence/satisfiability of Class [CakeFilling]

Figure 1: Our Prototype Authoring Tool

Figure 1 shows the main panel of the authoring tool, which consists of three main regions. On the left is a (clickable) presentation of the hierarchy of named classes in the ontology and a small window which can show a simple verbalisation of the axioms about a given class. In the centre is a history log, which shows the whole past dialogue interaction, and an area where the user can compose their next contribution to the dialogue. The user can choose between using OSE and using Manchester Syntax for DLs [14].

18 out of the 28 types of CQs identified by [27] have wordings that can be incorporated into an extension of OSE. Once the CQ has been entered, its presuppositions are extracted for use as authoring tests. The "task list" (shown in Figure 2) is expanded to show the new CQ; the authoring tests coming from the CQ are shown indented underneath. From then on, the status of



- 🟡 Goals
  - ▶ 🟢 What pizza has meaty topping?
  - ▶ 🟢 What pizza has which fish topping?
  - ▼ 🟢 What pizza has tomato topping?
    - 🟢 Satisfiability check of [Pizza and hasTopping some TomatoTopping]
    - 🟢 Satisfiability check of [Pizza and not (hasTopping some TomatoTopping)]
    - 🟢 Existence/satisfiability of Class [TomatoTopping]
    - 🟢 Existence/satisfiability of ObjectProperty [hasTopping]
    - 🟢 Existence/satisfiability of Class [Pizza]
  - ▼ 🟢 What cake has which dairy topping?
    - 🔴 Satisfiability check of [Cake hasTopping some DairyTopping]
    - 🔴 Satisfiability check of [Cake and not (hasTopping som DairyTopping)]
    - 🟢 Existence/satisfiability of Class [DairyTopping]
    - 🟢 Existence/satisfiability of ObjectProperty [hasTopping]
    - 🔴 Existence/satisfiability of Class [Cake]
  - ▼ 🟢 What cake has which cake filling?
    - 🔴 Satisfiability check of [Cake hasFilling some CakeFilling
    - 🔴 Satisfiability check of [Cake and not (hasFilling some CakeFilling)]
    - 🔴 Existence/satisfiability of Class [Cake]
    - 🟢 Existence/satisfiability of ObjectProperty [hasFilling]
    - 🟢 Existence/satisfiability of Class [CakeFilling]

Figure 2: Task list — the CQs are the top level list elements, with their associated ATs shown below

the authoring tests (succeed or fail) from reasoning [22] is indicated by "traffic lights" in front of the tests. When an authoring action (e.g. adding a new axiom) creates a change in the status of one or more authoring tests, this is announced in feedback as part of the main dialogue (the top central panel of Figure 1). We hope that this kind of dynamic feedback can help authors to understand their progress.

However, CQOA hinges on the accuracy of the mapping from presuppositions to ATs: it hinges on whether a CQOA system's understanding of what it means for a CQ to be "handled correctly" matches the user's understanding. It is conceivable, for example, that authors who have entered the CQ "Which pizzas contain chocolate?" are happy with an ontology that defines pizzas as not containing chocolate. If so, the Complement Presupposition fails, yet the author maight consider the CQ to have been handled correctly: Ren et al.'s mapping would be wrong. In this paper, we present a series of experiments to investigate whether the interpretation of CQs embodied in the mapping from presuppositions to ATs is in accordance with users' understanding.

2 Related Work

Empirical studies of ontology authoring emphasise the complexity of the ontology authoring task both for novice and experienced users [25, 9]. These studies suggest that current ontology authoring tools let users control the authoring process while many users prefer to be guided by the system.

A range of solutions has been proposed. Ontology testing is widely used to provide feedback to authors on the quality of the ontology. For example, the *Rabbit* interface [6] and the *Simplified English prototype* [24] test for the presence of incorrect words and syntactically disallowed structures in the ontology. *Protégé* [17] and *OntoTrack* [19] use reasoners to offer basic semantic checking, testing for inconsistency, for example. Systems such as *Roo* [7] intend to advise the user of the consequences of an authoring action. Justification engines [19] explain the feedback given by the system, for example when an inconsistency is detected. Systems such as the *OWL Unit Test Framework* in *Protégé*, *Tawny-OWL* [20] and *OntoStudio*[2] allow users to define unit tests and run these in the authoring environment.

These techniques have difficulty capturing requirements specific to the ontology in question. CQOA, by contrast, has the potential of capturing requirements that are specific to one ontology and one user. Exploiting CQs for ontology authoring is not a new idea [31, 11, 29, 16] — interesting approaches include the formalisation of CQs into SPARQL queries [33] or DL queries [21]. An algorithm for checking natural language CQs has been developed by [4]. However, most of these studies have focused on simple CQs such as "What is . . . ?", "How much . . . ?", and on answering CQs, instead of informing the user which CQs *can be* answered, and explaining why this is. An exception is Hofer et al. [13], which evaluates the coverage of biomedical ontologies by checking whether all terminologies in CQs can be mapped to terminologies in a target biomedical ontology. The CQOA approach goes further by addressing a wider range of CQs.

With a feature-based framework, Ren et al. [27] identified 12 archetypes of CQ patterns in their collection (Table 1), where the 2nd and 3rd columns show the pattern and an example from the corpus. The last four columns show the primary features of a pattern. Some archetype patterns have sub-types; subtypes of archetype 1 are shown in Table 2 on page 7, in which the last three columns are the secondary features of the subtype.

3 Study Design

Our question is whether the interpretation of CQs embodied in the mapping from presuppositions to ATs is in accordance with users' understanding.

We conducted a series of three experiments mainly for the ATs of occurance and relation satisfiability. Study 1 used a lay audience with participants recruited from the general population; the ontology and authoring tests were presented in English. Studies 2 and 3 were run with participants who had some experience with Description Logics, so the ontology and authoring tests in these experiments were expressed using DL syntax.

[2] `http://www.semafora-systems.com/en/products/ontostudio/`

Table 1: CQ Archetypes (from [27]) (PA = Predicate Arity, RT = Relation Type, M = Modifier, DE = Domain-independent Element; obj. = object property relation, data. = datatype property relation, num. = numeric modifier, quan. = quantitative modifier, tem. = temporal element, spa. = spatial element; CE = class expression, OPE = object property expression, DP = datatype property, I = individual, NM = numeric modifier, PE = property expression, QM = quantity modifier)

ID	Pattern	Example	PA	RT	M	DE
1	Which [CE1] [OPE] [CE2]?	Which pizzas contain pork?	2	obj.		
2	How much does [CE] [DP]?	How much does Margherita Pizza weigh?	2	data.		
3	What type of [CE] is [I]?	What type of software (API, Desktop application etc.) is it?	1			
4	Is the [CE1] [CE2]?	Is the software open source development?	2			
5	What [CE] has the [NM] [DP]?	What pizza has the lowest price?	2	data.	num.	
6	What is the [NM] [CE1] to [OPE] [CE2]?	What is the best/fastest/most robust software to read/edit this data?	3	both	num.	
7	Where do I [OPE] [CE]?	Where do I get updates?	2	obj.		spa.
8	Which are [CE]?	Which are gluten free bases?	1			
9	When did/was [CE] [PE]?	When was the 1.0 version released?	2	data.		tem.
10	What [CE1] do I need to [OPE] [CE2]?	What hardware do I need to run this software?	3	obj.		
11	Which [CE1] [OPE] [QM] [CE2]?	Which pizza has the most toppings?	2	obj.	quan.	
12	Do [CE1] have [QM] values of [DP]?	Do pizzas have different values of size?	2	data.	quan.	

3.1 Participants

Study 1 (English, crowdsourcing) Participants were recruited by Mechanical Turk (www.mturk.com), a crowdsourcing tool. Participants had to have an approval rate of 90% (i.e. 90% of their work was judged by other requesters as of good quality) and pass a Cloze test [30] for English fluency. We recruited 54 participants (50% male, 50% female; 32% aged 18–25, 57% 26–40, and 11% 40–65).

Study 2 (DL, Summer School) The first of the two experiments related to the DL version occurred during the 12^{th} Reasoning Web Summer School (Aberdeen 2016). The event targeted beginner and intermediate DL practitioners, such as PhD students and researchers in the Semantic Web area. Of our 15 participants, 86% were male, 14% female. 33% were aged 18–25 and 66% aged 26–40. 46% of participants were self-assessed novices, 40% were beginners and 14% reported as having intermediate skills; no participants identified as experts. The experiment was conducted during a dedicated 60-minute session of the school. The average time to complete the test was 32 (SD 21) minutes.

Study 3 (DL, Conference in China)

The second DL-based experiment was conducted at the CCKS2016 conference, held in China in September 2016. The conference targeted people in-

Table 2: CQ Sub-types of Archetype 1 (from [27]) (QT = Question Type, V = Visibility, QP = Question Polarity, sel. = selection question, bin. = binary question, cout. = counting question, exp. = explicit, imp. = implicit, sub. = subject, pre. = predicate, pos. = positive, neg. = negative)

ID	Pattern	Example	QT	V	QP
1a	Which [CE1] [OPE] [CE2]?	What software can read a .cel file?	sel.	exp.	pos.
1b	Find [CE1] with [CE2].	Find pizzas with peppers and olives.	sel.	imp. pre.	pos.
1c	How many [CE1] [OPE] [CE2]?	How many pizzas in the menu contains meat?	cout.	exp.	pos.
1d	Does [CE1] [OPE] [CE2]?	Does this fotware provide XML editing	bin.	exp.	pos.
1e	Be there [CE1] with [CE2]?	Are there any pizzas with chocolate?	bin.	imp. pre.	pos.
1f	Who [OPE] [CE]?	Who owns the copyright?	sel.	imp. sub.	pos.
1g	Be there [CE1] [OPE]ing [CE2]?	Are there any active forums discussing its use?	bin.	exp.	pos.
1h	Which [CE1] [OPE] no [CE2]?	Which pizza contains no mushroom?	sel.	exp.	neg.

terested in learning about semantic technologies; it contained tutorial sessions about DLs and ontology authoring. The experiment was conducted after the tutorial, to ensure that participants were able to understand the proposed DL formulae. 67 participants were recruited. 55% were male, 42% were female, and 3% undisclosed. 36% were aged 18–25, 54% 26–40, 7% 41–65, and 3% undisclosed. 61% were self-assessed novices, 22% beginners, 12% intermediate, 3% experts and 2% undisclosed. The average time to complete the test was 17 (SD 6) minutes.

3.2 Materials

3.2.1 Ontology

Given that our interest was not in testing peoples comprehension of complex ontologies, but in testing the treatment of presuppositions in CQs, we wanted the ontology to be fairly easy to comprehend, in a domain that many people understand, while still containing all the phenomena we are interested in. We therefore created a simple ontology from scratch. The subject was hot drinks, a topic that many people have a good understanding of. The complete ontology is shown in Table 3.

3.2.2 Competency Questions

Seven CQs were used. All but one were judged (via their authoring tests) to be non-answerable. In other words, the criteria of [27] assert that the ontology in its current form fails to make all the CQs answerable. Table 4 shows the CQs, their archetype according to the classification proposed in [27], and whether they can be answered: if not, a brief explanation is provided.

Most of the CQs are of archetype 1 (see Table 1 on page 6) — a realistic design choice, as this is the most common type of CQ used by human users [27]. CQs 6 and 7 are more complex than 2, 3 and 4, since they exploit logical connectors between the concepts proposed.

Table 3: The ontology, as presented to participants in the DL and in the English studies reported below.

	DL	Non DL
1	$hasContent \circ hasContent \sqsubseteq hasContent$	The robot understands that things can 'contain' other things, and that this is transitive. Transitive means that, for example, if flour contains gluten, and a loaf of bread contains flour, then the loaf of bread therefore contains gluten.
2	$Drink \equiv CoffeeDrink \sqcup TeaDrink$	All drinks are coffee drinks or tea drinks
3	$Coffee \sqsubseteq \exists hasContent.Caffeine$	Coffee beans contain caffeine
4	$CoffeeDrink \equiv \exists hasContent.Coffee$	Coffee drinks contain coffee beans
5	$TeaDrink \equiv \exists hasContent.Tea$	Tea Drinks contain tea leaves
6	$CoffeeDrink \sqcap TeaDrink \sqsubseteq \bot$	Nothing can be both a coffee drink and a tea drink at the same time
7	$Cappuccino \sqsubseteq Drink \sqcap \exists hasContent.SteamedMilk \sqcap \exists hasContent.Coffee$	A cappuccino is a drink that contains steamed milk and coffee beans
8	$Americano \sqsubseteq CoffeeDrink$	An Americano is a coffee drink

Table 4: The competency questions proposed to the participants of the study

	Competency Question	Archetype	Answ.	Reason
1	Which are the coffee drinks?	8	yes	
2	Which coffee drinks contain caffeine?	1	no	All the coffee drinks contain caffeine
3	Which tea drinks contain caffeine?	1	no	The relation between tea drinks and caffeine is undefined
4	Which coffee drinks contain tea leaves?	1	no	No coffee drink contains tea leaves
5	Which coffee drinks contain the most caffeine?	11	no	The answer cannot be computed
6	Which drinks contain coffee beans or tea leaves?	1	no	All drinks contain either beans or tea leaves
7	Which drinks contain coffee beans and tea leaves?	1	no	No drink contains both coffee beans and tea leaves

Table 5: Authoring test types and examples

Type	DL	Non DL
1. Occurrence (conc.)	$CoffeeDrink$ should occur in the ontology	A coffee drink should be defined
2. Occurrence (prop.)	*hasContent* should occur in the ontology	It must be possible for something to contain something
3. Relation Satisfiability	$CoffeeDrink \sqcap \exists hasContent.Caffeine$ should be satisfiable in the ontology	It must be possible for a coffee drink to contain caffeine
4. Relation Satisfiability (complement)	$CoffeeDrink \sqcap \neg\exists hasContent.TeaLeaf$ should be satisfiable in the ontology	It must be possible for a coffee drink to not contain tea leaves

3.2.3 Authoring Tests

Each CQ had a set of ATs associated with it, following the mapping proposed by Ren and colleagues. We focus on four types (from [27]) of AT in this paper, with examples shown in Table 5.

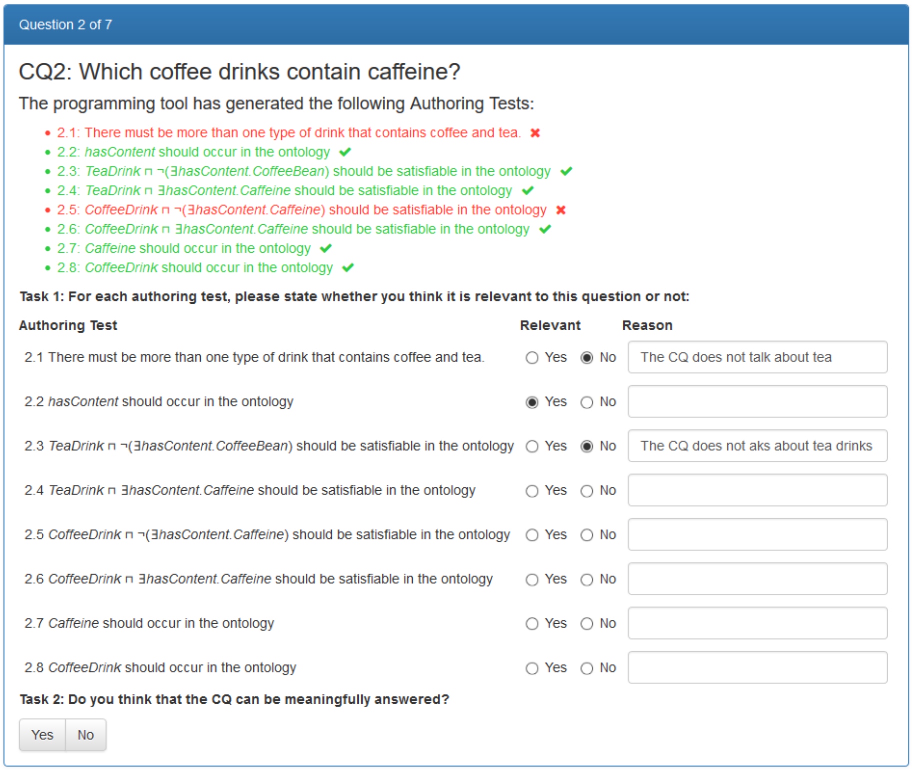
ATs of types 1 and 2 assess the presence in the ontology of concepts and properties, respectively. These tests pass if the concept or property is defined in the ontology. The example AT of type 1 presented in Table 5 is associated with CQs 1, 2, 4 and 5 of Table 4; similarly, the AT example of type 2 is associated with all the CQs except 1.

ATs of types 3 and 4 are Relation Satisfiability tests. They aim at verifying whether relations between classes are possible. For example, the AT proposed (in Table 3) for Relation Satisfiability assesses whether a coffee drink can contain caffeine. If it cannot, the associated CQ is judged to be not answerable.

55 ATs were used in total: 21 fillers (Non-relevant ATs used as an attention check) and 34 non-fillers. Disregarding fillers, there were 16 of type 1, 6 of type 2, 6 of type 3 and 6 of type 4.

3.3 Variables

The independent variable was the type of authoring test. The dependent variable was what we call *relevance*; this records whether a participant judged an authoring test to be relevant to a given CQ (i.e., whether the AT expresses a presupposition of the CQ) or not.



Question 2 of 7

CQ2: Which coffee drinks contain caffeine?

The programming tool has generated the following Authoring Tests:

- 2.1: There must be more than one type of drink that contains coffee and tea. ✗
- 2.2: *hasContent* should occur in the ontology ✓
- 2.3: $\mathit{TeaDrink} \sqcap \neg(\exists \mathit{hasContent}.\mathit{CoffeeBean})$ should be satisfiable in the ontology ✓
- 2.4: $\mathit{TeaDrink} \sqcap \exists \mathit{hasContent}.\mathit{Caffeine}$ should be satisfiable in the ontology ✓
- 2.5: $\mathit{CoffeeDrink} \sqcap \neg(\exists \mathit{hasContent}.\mathit{Caffeine})$ should be satisfiable in the ontology ✗
- 2.6: $\mathit{CoffeeDrink} \sqcap \exists \mathit{hasContent}.\mathit{Caffeine}$ should be satisfiable in the ontology ✓
- 2.7: *Caffeine* should occur in the ontology ✓
- 2.8: *CoffeeDrink* should occur in the ontology ✓

Task 1: For each authoring test, please state whether you think it is relevant to this question or not:

Authoring Test	Relevant	Reason
2.1 There must be more than one type of drink that contains coffee and tea.	○ Yes ● No	The CQ does not talk about tea
2.2 *hasContent* should occur in the ontology	● Yes ○ No	
2.3 $\mathit{TeaDrink} \sqcap \neg(\exists \mathit{hasContent}.\mathit{CoffeeBean})$ should be satisfiable in the ontology	○ Yes ● No	The CQ does not aks about tea drinks
2.4 $\mathit{TeaDrink} \sqcap \exists \mathit{hasContent}.\mathit{Caffeine}$ should be satisfiable in the ontology	○ Yes ○ No	
2.5 $\mathit{CoffeeDrink} \sqcap \neg(\exists \mathit{hasContent}.\mathit{Caffeine})$ should be satisfiable in the ontology	○ Yes ○ No	
2.6 $\mathit{CoffeeDrink} \sqcap \exists \mathit{hasContent}.\mathit{Caffeine}$ should be satisfiable in the ontology	○ Yes ○ No	
2.7 *Caffeine* should occur in the ontology	○ Yes ○ No	
2.8 *CoffeeDrink* should occur in the ontology	○ Yes ○ No	

Task 2: Do you think that the CQ can be meaningfully answered?

Figure 3: Screenshot of study 2 showing one experimental participant's judgement of ATs for one CQ. The participant has so far only addressed the first three ATs. In two cases, she has offered a reason.

3.4 Procedure

In the DL experiment, we collected information about participants' experience in ontology authoring. Next, participants were given a written scenario to read. The scenario was designed to make sense to people not previously acquainted with the notion of an ontology, and in such a way that the role of the CQs would nonetheless be clear. The scenario read as follows:

Costabucks is a hot drinks company. They are creating a robot that can answer questions from customers about the hot drinks that they sell. The robot's programmers have to tell the robot some facts about hot drinks so that it understands enough to answer the questions. To do this, the robot's designers give the robot "rules" about the world.

Once all of the Customer Questions (CQs) can be answered by the robot, its knowledge of the coffee menu is considered complete, and it can be used in the shop.

The programmers are using a special ***programming tool*** *which allows them* *to add possible customer questions to its interface, and the tool can inform them when the questions are able to be answered by the current set of rules.*

To do this, the tool breaks down the questions into several smaller ***authoring tests****, which all must be passed in order for the question to be judged as answerable. The authoring tests are automatically generated by the tool, based on what the customer question is.*

Participants were shown the ontology (set of axioms) shown in Table 3, using one of the two formats (DL or English). Next, a simple example CQ was shown and the types of authoring test that could arise from it. The symbols used to highlight whether the AT 'passed' or not were explained. Following this, participants were shown the 7 CQs, one by one. For each CQ, participants were shown the CQ's associated ATs and asked, in each case, whether they agreed. Participants could give a reason to explain their judgement if they wished; an example is shown in Figure 3. The list of authoring tests also contained certain non-relevant *fillers* which served as an attention check, and allowed us to gauge the ability of participants to understand the task. For example, for the first CQ: *Which are the coffee drinks?*, we inserted fillers such as *Steamed milk should be defined* and *Tea-leaves should be defined*, which are not relevant to coffee drinks.

Participants were told that as long as all ATs had *passed*, the CQ was considered answerable by the programming tool, and if any ATs were *failing*, then the CQ was judged as non-answerable. Participants were asked whether they agreed with this answerability judgement or not.

3.5 Hypotheses

In order to verify the mappings from presuppositions to the 4 types of ATs from [27], we formulated the following hypotheses before conducting our experiments.

> H1: occurrence ATs are agreed with more often than disagreed with.
>
> H2: satisfiability ATs that focus on a concept mentioned in a CQ are agreed with more often than disagreed with.
>
> H3: satisfiability ATs that focus on the complement of a concept mentioned in a CQ are agreed with more often than disagreed with.
>
> H4: satisfiability ATs that focus on a concept mentioned in a CQ are agreed with more often than satisfiability ATs that focus on the complement of a concept mentioned in a CQ.

The first three hypotheses are the core of our investigation, making explicit an expectation inherent in the literature. They assert that these ATs proposed in [27] are agreed with more often than disagreed with (separating out three different types of ATs). If linguistic theory is right about presuppositions, then we would expect to see at least the first two of these hypotheses overwhelmingly supported. The fourth hypothesis reflects a more tentative expectation, namely that *positive* presuppositions are more firmly associated with questions of the form "Which ..." than are *complement* presuppositions (cf., Section 1).

Table 6: Results of study 1 (English, crowdsourcing), showing the percentage of times an AT of a particular type was marked as relevant. * indicates significance of binomial test. Here and in Tables 5 and 6, *Filler thresholds* indicate the percentage of filler ATs that has to be answered correctly to be counted. Thus, the 66% column shows only results for those who understood ATs quite well, whereas the 0% column shows all.

Filler threshold	0%		50%		66%	
AT type	relevant	not relevant	relevant	not relevant	relevant	not relevant
Occurrence (conc)	96% (830)	4% (34) *	97% (482)	3% (14) *	98% (298)	2% (6) *
Occurrence (prop)	91% (296)	9% (28) *	90% (168)	10% (18) *	84% (96)	16% (18) *
Satisfiability (conc)	76% (245)	24% (79) *	82% (152)	18% (34) *	83% (95)	17% (19) *
Satisfiability (comp)	72% (233)	28% (91) *	71% (132)	29% (54) *	72% (82)	28% (32) *

Table 7: Results of study 2 (DL, Summer School, UK) showing the percentage of times an AT of a particular type was marked as relevant. * indicates significance of binomial test. Bold - non significance

Filler threshold	0%		50%		66%	
AT type	relevant	not relevant	relevant	not relevant	relevant	not relevant
Occurrence (conc)	95% (227)	5% (13) *	94% (181)	6% (11) *	96% (108)	4% (4) *
Occurrence (prop)	100% (90)	0% (0) *	100% (72)	0% (0) *	100% (42)	0% (0) *
Satisfiability (conc)	78% (70)	22% (20) *	82% (59)	18% (13) *	93% (39)	7% (3) *
Satisfiability (comp)	64% (58)	36% (32) *	62.5% (45)	37.5% (27) *	**55%** (23)	**45%** (19)

Table 8: Results of study 3 (DL, Conference in China) showing the percentage of times an AT of a particular type was marked as relevant. * indicates significance of binomial test. Bold - non significance

Filler threshold	0%		50%		66%	
AT type	relevant	not relevant	relevant	not relevant	relevant	not relevant
Occurrence (conc)	83% (893)	17% (179) *	82% (499)	18% (109) *	78% (250)	22% (70) *
Occurrence (prop)	86% (347)	14% (55) *	88% (201)	12% (27) *	79% (95)	21% (25) *
Satisfiability (conc)	70% (280)	30% (122) *	74% (168)	26% (60) *	73% (88)	27% (32) *
Satisfiability (comp)	**50%** (202)	**50%** (200)	**43%** (98)	**57%** (130)	36% (43)	64% (77) *

4 Results

Results are given for all participants, followed by participants who successfully identified at least 50% of the filler ATs (i.e. a 50% *filler threshold*), and finally results for those participants who successfully identified at least 66% of filler ATs. H1, H2 and H3 were assessed by binomial test. Hypothesis H4 was analysed by a χ^2 test of *attype* (authoring test type) × *answer* (relevant or not relevant). In all analyses, a significance threshold of $p < .05$ was used.

Study 1 (English, crowdsourcing) As shown in Table 6, H1, H2 and H3 are confirmed with a significant majority of participants agreeing with the generated authoring tests. H4 is not supported.

Study 2 (DL, Summer school, UK) The results from the first description logic experiment are shown in Table 7, for filler thresholds 0 and 50%, H1, H2 and H3 are confirmed with a significant majority agreeing with the generated ATs. However, at a 66% filler threshold, H3 is not supported, with only a small majority of 'satisfiability of complement of concept' ATs being marked as relevant. Once again, H4 is not supported.

Study 3 (DL, Conference in China). The results from the second description logic experiment are shown in Table 8. As before, H1 and H2 are confirmed for all filler thresholds, but this time there is no support for H3: as the filler threshold is increased, more of the 'satisfiability of complement of concept' ATs are marked as non-relevant. For the filler thresholds of 50% and 66% (representing the DL-logically more capable participants), significant majorities marked these ATs as *non-relevant.* For hypothesis H4, a χ^2 test of *ATtype* × *answer* (for both types of satisfiability AT) shows this to be significant for all filler thresholds (0%: $\chi^2 = 31.517$, $p < 0.001$; 50%: $\chi^2 = 44.211$, $p < 0.001$; 66%: $\chi^2 = 34.036$, $p < 0.001$), hence this hypothesis is confirmed.

5 Discussion

We have found that *occurrence* ATs are almost universally agreed with; this was not surprising, since an ontology that does not define a given concept or relation is unable to shed light on any CQ containing it. We also found broad agreement that the key concept involved in a Selection Question must be satisfiable. However, when dealing with the complement of such a concept, participants in study 3 did not agree that this had to satisfiable. Remarkably (cf., the three levels of Filler threshold in Table 6), the better a participant was at identifying relevant ATs, the more this type of ATs was disagreed with. We did not reliably find this declining pattern with the other DL experiment (study 2, the three levels of Filler threshold in Table 5); this could be due to the smaller number of participants or to the type of participants taking part in study 2.

How to explain these findings? Presentation format may have affected CQ interpretation: when an ontology is presented using DL formulas, a less "natural language-like" interpretation of CQs (which were themselves formulated in English in all experiments) may be triggered. It seems possible that participants' exposure to DL formulas in the DL-based experiments activates in their minds a literal interpretation of CQs, which has no presuppositions. For example, according to this literal interpretation, "Which As have a B?" can have none of the As and all of the As as legitimate answers. If this explanation is correct, one would expect that H2 and H3 are less well supported by the DL experiments than by the non-DL experiment, which was not the case. Conversely, the reasoning above gives one no reason to expect the observed difference between positive and complement presuppositions. Alternative explanations need to be explored.

It might seem plausible that subjects with more experience using formal logic are more likely to use a literal interpretation of these formulas, which has no presuppositions (as explained above). If this is correct, then one should expect that both H2 and H3 are less well supported by participants with a high level of expertise in logic than by subjects with a low level of expertise in logic. To investigate this, we performed a post-hoc analysis on the two DL experiments. We partitioned participants into two groups - those who reported their experience as 'novice' vs. those reporting as 'beginner','intermediate' and 'expert' (because only those reporting as novices had no prior DL experience). We found no significant differences between the two groups. For satisfiability of a concept, 69% were marked as relevant by the novice group and 73% were marked as relevant by the others. For satisfiability of the complement of a concept, 52% of ATs were marked as relevant and 54% for the other group.

A second option is to use the filler ATs (rather than self-reported experience) as a guide to participants' expertise. We split the results into three groups - one for those who identified under 50% of fillers, a second for those who identified over 50% but under 66%, and a third for those who identified over 66% (Table 9). For both types of AT, a χ^2 test of $relevant \times group$ was significant (Satisfiability of Concept: $\chi^2 = 8.955$, $p < 0.02$; Satisfiability of complement of concept: $\chi^2 = 15.053$, $p < 0.001$). Strikingly, the trend differs for the two types of AT. For satisfiability of a concept, more ATs are marked as relevant as the filler threshold is increased. However, for satisfiability of the complement of a concept, the trend is in the other direction, with fewer of these ATs being marked relevant as the filler threshold is increased.

Table 9: Results of the post-hoc test for authoring test relevance across the three groups of participants.

	Filler group	relevant	not-relevant
Satisfiability (conc)	<50%	64%	36%
	50-66%	72.5%	27.5%
	over 66%	78%	22%
Satisfiability (comp)	<50%	61%	39%
	50-66%	56%	44%
	over 66%	41%	59%

6 Conclusions

Our experimental findings suggest that the CQ-driven Ontology Authoring (CQOA) approach to testing the answerability of a CQ, as embodied in Ren et al.'s mapping from CQs to ATs, is on the right track: in each of our three experiments, participants agreed with the way in which this mapping decides whether a CQ can be answered by a given ontology.

We consider these findings to be an important milestone towards the goal of improving ontology authoring via CQOA. Our results do not yet prove the *usefulness* of CQs for ontology authoring: it is possible that even an authoring interface that understands perfectly how a user has intended a given set of CQs, and which uses this understanding to tell the user which CQs have yet to be addressed, might still not contribute much to the authoring task, for example because of the manner in which the interface indicates which CQs and ATs have been met (perhaps the "traffic lights" illustrated in Fig. 2 are not understood well enough). We hope to do further experiments, to investigate the effect of CQOA on the speed and accuracy of ontology authoring, and the effect on the user's understanding of, and trust in, the ontology they have authored.

Our studies allowed us to flesh out some additional issues. Intriguing questions arose from the asymmetry between *positive* and *complement* presuppositions, particularly among users with higher DL proficiency. While it is easy to see why these users may have "unlearnt" to assign presuppositions to sentences, it is more difficult to see why this should hold particularly for complement presuppositions. These issues should be investigated further, to find out whether CQOA's usefulness is different for users with different backgrounds and/or aptitudes.

The literature on Linguistic Pragmatics is rich in theories that formalise what the presuppositions of a given sentence type are thought to be [18, 3], but there has only been a limited amount of empirical testing of these theories ([28] for an overview; [5, 15] for empirical studies). Our findings suggest that the support for many presuppositions is far from universal, where some were supported by as few as 36% of participants. In other words, the idea of "determining the presuppositions of a question" turns out to be a subtle affair. Perhaps the question of what information is presupposed by a given sentence is a matter of degrees, best thought of in terms of an expectation that the sentence can raise in the hearer's mind, where the strength of this expectation can differ. To vary on a classic example, suppose someone asks you "Is the king of France bald?". Traditional approaches can only say that this question does, or does not, presuppose that France has a king; perhaps a more graded approach is preferable, which asserts that the question raises the expectation that France has a king, but the strength of this expectation can differ in strength between different hearers.

Acknowledgments

This research has been funded by the EPSRC project: WhatIf: Answering "What if" questions for Ontology Authoring. EPSRC reference EP/J014176/1.

References

[1] Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene ontology: tool for the unification of biology. Nature genetics 25(1), 25–29 (2000)

[2] Baader, F.: The description logic handbook: Theory, implementation and applications. Cambridge university press (2003)

[3] Beaver, D.: Presupposition. In: van Benthem, J., ter Meulen, A. (eds.) Handbook of Logic and Language, pp. 939–1009. North Holland (1997)

[4] Bezerra, C., Freitas, F., Santana, F.: Evaluating ontologies with competency questions. In: Web Intelligence (WI) and Intelligent Agent Technologies (IAT), IEEE/WIC/ACM International Joint Conference. vol. 3, pp. 284–285. IEEE (2013)

[5] Breheny, R., Katsos, N., Williams, J.: Are generalised scalar implicatures generated by default? An on-line investigation into the role of context in generating pragmatic inferences. Cognition 100, 434–463 (2006)

[6] Denaux, R., Dimitrova, V., Cohn, A.G., Dolbear, C., Hart, G.: Rabbit to owl: ontology authoring with a cnl-based tool. In: CNL. Springer (2010)

[7] Denaux, R., Thakker, D., Dimitrova, V., Cohn, A.G.: Interactive semantic feedback for intuitive ontology authoring. In: FOIS. pp. 160–173 (2012)

[8] Dzbor, M., Motta, E., Buil, C., Gomez, J.M., Görlitz, O., Lewen, H.: Developing ontologies in OWL: an observational study. In: OWLED. CEUR Workshop Proceedings, vol. 216. CEUR-WS.org (2006)

[9] Dzbor, M., Motta, E., Gomez, J.M., Buil, C., Dellschaft, K., Görlitz, O., Lewen, H.: D4.1.1 analysis of user needs, behaviours & requirements wrt user interfaces for ontology engineering. Tech. rep. (August 2006)

[10] Enderton, H.B.: A mathematical introduction to logic. Academic press (2001)

[11] Fernandes, P.C.B., Guizzardi, R.S., Guizzardi, G.: Using goal modeling to capture competency questions in ontology-based systems. JIDM 2(3), 527 (2011)

[12] Grueninger, M., Fox, M.: Methodology for the design and evaluation of ontologies. In: IJCAI Workshop on Basic Ontology Issues in Knowledge Sharing (1995)

[13] Hofer, P., Neururer, S., Helga Hauffe, T.I., Zeilner, A., Gbel, G.: Semi-Automated Evaluation of Biomedical Ontologies for the Biobanking Domain Based on Competency Questions. Studies in Health Tech. and Informatics 212, 65 – 72 (2015)

[14] Horridge, M., Drummond, N., Goodwin, J., Rector, A.L., Stevens, R., Wang, H.: The manchester owl syntax. In: OWLed. vol. 216 (2006)

[15] Huang, Y.T., Snedeker, J.: On-line interpretation of scalar quantifiers: insight into the semantic-pragmatics interface. Cognitive Psychology 58, 376–415 (2009)

[16] Keet, C.M., Lawrynowicz, A.: Test-driven development of ontologies. In: ESWC. Lecture Notes in Computer Science, vol. 9678, pp. 642–657. Springer (2016)

[17] Knublauch, H., Fergerson, R.W., Noy, N.F., Musen, M.A.: The protégé OWL plugin: An open development environment for semantic web applications. In: ISWC. LNCS, vol. 3298, pp. 229–243. Springer (2004)

[18] Levinson, S.C.: Pragmatics. Cambridge University Press (1983)

[19] Liebig, T., Noppens, O.: Ontotrack: A semantic approach for ontology authoring. Web Semantics: Science, Services and Agents on the World Wide Web 3(2), 116–131 (2005)

[20] Lord, P.: The semantic web takes wing: Programming ontologies with tawny-owl. In: OWLED 2013 (2013), `http://www.russet.org.uk/blog/2366`

[21] Malheiros, Y., Freitas, F.: A method to develop description logic ontologies iteratively based on competency questions: an implementation. In: ONTOBRAS. pp. 142–153 (2013)

[22] Pan, J.Z., Ren, Y., Zhao, Y.: Tractable approximate deduction for OWL. Artificial Intelligence 235, 95–155 (2016)

[23] Power, R.: Owl simplified english: a finite-state language for ontology editing. In: International Workshop on Controlled Natural Language. pp. 44–60. Springer (2012)

[24] Power, R.: Owl simplified english: a finite-state language for ontology editing. In: Controlled Natural Language, pp. 44–60. Springer (2012)

[25] Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C.: Owl pizzas: Practical experience of teaching owl-dl: Common errors & common patterns. In: Engineering Knowledge in the Age of the Semantic Web, Lecture Notes in Computer Science, vol. 3257, pp. 63–81. Springer (2004)

[26] Rector, A.L., Drummond, N., Horridge, M., et al., J.R.: OWL pizzas: Practical experience of teaching OWL-DL: common errors & common patterns. In: EKAW. LNCS, vol. 3257, pp. 63–81. Springer (2004)

[27] Ren, Y., Parvizi, A., Mellish, C., Pan, J.Z., van Deemter, K., Stevens, R.: Towards competency question-driven ontology authoring. In: ESWC. Lecture Notes in Computer Science, vol. 8465, pp. 752–767. Springer (2014)

[28] Sedivy, J.C.: Implicature during real time conversation: a view from language processing research. Philosophy compass 2/3, 275–496 (2007)

[29] Suárez-Figueroa, M.C., Gómez-Pérez, A.: Ontology requirements specification. In: Ontology Engineering in a Networked World, pp. 93–106. Springer (2012)

[30] Taylor, W.L.: Cloze procedure: A new tool for measuring readability. Journalism Quarterly 30, 415433 (1953)

[31] Uschold, M., Gruninger, M., et al.: Ontologies: Principles, methods and applications. Knowledge engineering review 11(2), 93–136 (1996)

[32] Whetzel, P.L., Parkinson, H.E., Causton, H.C., Fan, L., Fostel, J., Fragoso, G., Game, L., Heiskanen, M., Morrison, N., Rocca-Serra, P., Sansone, S., Taylor, C.F., White, J., Jr., C.J.S.: The MGED ontology: a resource for semantics-based description of microarray experiments. Bioinformatics 22(7), 866–873 (2006)

[33] Zemmouchi-Ghomari, L., Ghomari, A.R.: Translating natural language competency questions into SPARQL queries: A case study. In: WEB 2013, The First International Conference on Building and Exploring Web Based Environments. pp. 81–86 (2013)

[34] Zuber, R., Zuber, R.: Non-declarative sentences. John Benjamins Publishing (1983)