Question Answering over Pattern-Based User Models

Georgios Meditskos
Information Technologies Institute, CERTH, Greece
gmeditsk@iti.gr

Stamatia Dasiopoulou
Information and Communication Technologies Dept., UPF, Spain
stamatia.dasiopoulou@upf.edu

Stefanos Vrochidis
Information Technologies Institute, CERTH, Greece
stefanos@iti.gr

Leo Wanner
ICREA and Information and Communication Technologies Dept., UPF, Spain
leo.wanner@upf.edu

Ioannis Kompatsiaris
Information Technologies Institute, CERTH, Greece
ikom@iti.gr

ABSTRACT

In this paper we present an ontology-driven framework for natural language question analysis and answering over user models (e.g. preferences, habits and health problems of individuals) that are formally captured using ontology design patterns. Pattern-based modelling is extremely useful for capturing *n-ary relations* in a well-defined and axiomatised manner, but it introduces additional challenges in building NL interfaces for accessing the underlying content. This is mainly due to the encapsulation of domain semantics inside conceptual layers of abstraction (e.g. using reification or container classes) that demand flexible, context-aware approaches for query analysis and interpretation. We describe the coupling of a frame-based formalisation of natural language user utterances with a context-aware query interpretation towards question answering over pattern-based RDF knowledge bases. The proposed framework is part of a human-like socially communicative agent that acts as an intermediate between elderly migrants and care personnel, assisting the latter to solicit personal information about care recipients (e.g. medical history, care needs, preferences, routines, habits, etc.).

CCS Concepts

•Information systems → Ontologies; Query representation; Question answering; •Computing methodologies → Natural language processing;

Keywords

language analysis, question answering, ontology design patterns, user models

1. INTRODUCTION

As the amount of structured knowledge made available in the Linked Data cloud and in proprietary knowledge bases keeps growing, so does the pursuit for effective accessing and querying paradigms. Within this endeavour, recent years have witnessed important advances in natural language interfaces (NLIs) and Question Answering (QA) systems for structured data that allow users to express their information needs in an intuitive manner, while hiding the complexity of formal knowledge representation and query languages [19].

The key challenge in these efforts is to bridge the gap between the way users communicate with the system and the way domain knowledge is captured, and more specifically to translate the questions expressed in natural language into structured queries, such as SPARQL, so that pertinent answers can be retrieved from the underlying knowledge bases. This usually involves the translation of the natural language questions into semantically enriched structures that capture the meaning of requests, and the formulation of pertinent queries in accordance with the conceptualisation of the underlying structured data sources.

Most of the existing approaches provide support only for factoid queries, including predicative (e.g. *Who is the daughter of Robert Kennedy married to?*), list (e.g. *Give me all cities in Germany.*) and yes/no (e.g. *Is Woody Allen an actor?*) ones, translating the natural language questions into triple-based representations; corresponding SPARQL queries are subsequently constructed, relying on some notion of similarity. As such, the answers correspond to plain query variable bindings, and the focus is primarily directed to the two key pertinent challenges [28], namely how to overcome the conceptual mismatch between the triple-based question representations and the underlying knowledge model (e.g. matching the *have inhabitants* in ⟨Barcelona, have inhabitants, value⟩ with the *dbo:populationTotal*) and how to cope with lexical ambiguities.

Confronting these two challenges is clearly fundamental for affording intuitive access to the growing amount of structured knowledge made available (e.g. DBpedia, YAGO); yet, it leaves open question answering over more conceptually demanding domains, such as habits and daily routines profiling, that inherently involve complex relational contexts that go beyond (chains of) binary associations and abide instead to ontology patterns design principles [11]. Although different ODPs endorse different levels of generality [8], they usually describe abstract roles and relationships so that each pattern can be applied in a wide variety of situations. This level of generalization fosters reusability and extensibility, but imposes certain challenges both in the formalisation of the natural language questions and in the subsequent content matching and retrieval. For example, the annotation or encapsulation of domain knowledge within rich *n-ary* relations requires context-driven knowledge extraction solutions, beyond simple queries that are formulated based on one-to-one entity and relation mappings.

Aiming towards NL query interfaces over conceptually rich knowledge bases, the presented framework lies in the intersection of three research fields, namely knowledge distillation from text, question answering and pattern-based user modelling. More specifically, leveraging ontology design principles and linguistic frames, we present a reified representation paradigm for capturing natural language questions that express complex relations (i.e. events and situations involving *n-ary* dependencies). The resulting ontological representations serve then as input to a knowledge-driven interpretation and question answering framework for context matching and retrieval over RDF data sources containing pattern-based conceptualisations. The current investigation emphasis is on accessing individuals' knowledge, such as activity norms and behavioural patterns, diet preferences and health problems, captured by expressive ODPs that extend the DOLCE-DnS Ultralight design patterns[1]. To the best of our knowledge, this is the first attempt to explicitly cope with NL interfaces for QA over conceptual rich KBs, i.e. pattern-based KBs that encapsulate rich axiomatizations.

The rest of the paper is structured as follows. Section 2 discusses related efforts and current limitations in addressing question answering over conceptually rich knowledge bases, motivating and contrasting our work within the existing literature. Sections 3 and 4 present the proposed question analysis and context extraction approaches, which Section 5 explicates through an example use case. Last, Section 6 concludes the paper and outlines next steps.

2. RELATED WORK AND MOTIVATION

2.1 Ontology-based Question Answering

Several approaches have been proposed in the literature that address QA over Semantic Web knowledge bases [19]. Most of them focus on the generation of one or more SPARQL queries through the interpretation of the semantic structure of the user questions, while others opt for graph-based approaches to mitigate the rigidness often entailed in formulating appropriate SPARQL queries.

PowerAqua [17] allows users to choose an ontology and pose queries relevant to this ontology vocabulary. The results of language analysis are serialised into triples, which are further annotated with ontology resources. Finally, the triples are translated into logical queries that retrieve answers from the underlying knowledge sources. NLP-Reduce [16] processes queries as bags of words, employing stemming and synonym expansion. It attempts to match the parsed question words to the synonym-enhanced triples stored in the lexicon generated from a KB and expanded with WordNet synonyms, generating SPARQL statements for those matches. FREyA [7] is an interactive Natural Language Interface for querying ontologies, which combines syntactic parsing with the ontology-based lookup in an attempt to precisely answer questions. If the system fails to automatically generate the answer, suggestions are shown to the user found through ontology reasoning. The system then learns from user selections, and improves its performance over time. Other relevant approaches include [26, 30, 29] for SPARQL generation based on templates or query patterns, [2] for retrieving individual and generic knowledge using the structured query language OASSIS-QL and [25] for keyword-driven SPARQL generation. A domain-restricted QA framework is presented in [9] that is based on fixed QA topics associated with predefined SPARQL queries. Learning and scoring heuristics for filtering out redundant queries are common practices to cope with mismatches between the structure of questions and background knowledge.

As far as graph-driven approaches that reduce QA to subgraph matching problem are concerned, a recent example is the graph-traversal based approach presented in [31] which is based on topological patterns and similarity metrics between predicate labels and entities. In a similar manner, Zou et al. [32] computes the semantic similarity of matching vertices and edges between the subgraph and the query graph. This approach is further supported by an offline process, where a graph mining algorithm maps natural language phrases to top-k possible predicates in a RDF dataset, forming a paraphrase dictionary that is used for question understanding. In [10], after parsing the NL query, the algorithm outputs a list of ranked triple paths following from a pivot entity to the final resource representing the answer, ranked by the average of the relatedness scores in the path. A similar approach is followed in [1].

Summing up, the focus has been on simple, factoid questions, where the NL inputs comprise primarily light linguistic constructions and the answers target respective bindings on (chains of) binary properties. Much the same applies to current evaluation methods, such as the Question Answering over Linked Data (QALD) benchmark initiatives [18], where comparatively few, complex NL questions are included and evaluation is performed on linked data sets with simple conceptual models on highly interlinked resources, assuming that answers are explicitly represented in the KB, possibly following a different terminology.

2.2 Mapping NL to Semantic Representations

As previously outlined, most QA systems adhere to shallow linguistic analysis and triple-based serialisations, falling short to cope with the translation of complex NL questions into faithful semantic representations and respective queries.

A notable exception is the Pythia question answering system [27], where deep linguistic analysis is used to compositionally construct general meaning representations from NL questions involving quantification, aggregation functions and superlatives. Although certain portability and scalability concerns apply, due to the need for explicating admissible linguistic realisations of the considered domain ontology classes and properties, the main concern is about the difficulty of assessing its performance over conceptually demanding domains, as the reported evaluation ran over the

[1]http://ontologydesignpatterns.org/

Mooney's dataset[2] that adheres to a simple ontology for geographical information.

Parallel to QA-motivated efforts for capturing the semantic structure of NL questions, there has been recently a growing interest for paradigms for the principled translation of NL texts into RDF/OWL representations. Seminal examples include among others, LODifier, PIKES and FRED. In LODifier [3], Discourse Representation Structures (DRSs) [15], extracted by means of deep semantic parsing, are converted to RDF triples using transformation rules that map the unary and binary DRS conditions to respective class and property assertions, while RDF reification is used for logical and modal descriptions, such as disjunction and possibility. Focusing on publishing text as linked data, certain design choices, such as the use of blank nodes, become problematic, at least without some further post-processing and refactoring, for application contexts, such as NL QA question, that require cleaner representations that are closer to Semantic Web best practices.

Adopting a more knowledge-oriented paradigm, PIKES [6] extracts entities and complex relations between them, using deep semantic parsing and linguistic frames, and subsequently converts them into respective OWL graphs. The translation follows a neo-Davidsonian representation style, where frames are represented as reified objects, connected to each of their participants by means of properties that reflect the semantic roles of the participants, using, among others, the VerbNet[3] and FrameNet[4] semantic role repositories. To this end, SPARQL-like rules are used to refactor the linguistically grounded representations ("mention layer") to respective knowledge assertions ("instance layer"), while post-processing is applied to materialise implicit knowledge and compact redundant structures. The uniform treatment of the various frame categories can result however in counter-intuitive representations (e.g. introducing instances of two distinct classes for the same real-world entity); moreover, the alignment with foundational ontologies is not considered.

FRED [22] combines Discourse Representation Theory [15], linguistic frames, and ontology design patterns, to produce RDF/OWL ontologies and linked data from text. Deep semantic parsing is used to capture entities and the relations between them as DRS structures. Semantic role labelling is performed using VerbNet and FrameNet roles. What distinguishes FRED from other approaches and renders it as the work that is most relevant to our pursuits, is that it maximises modelling choices in accordance to Semantic Web principles and grounds the transformation and re-engineering of DRS structures to RDF/OWL graphs on the event and situation semantics as defined in DOLCE+DnS Ultra Lite, modelling semantic roles as object properties. Certain features, including the mostly verbal coverage of events and the introduction of periphrastic properties, impact the completeness and transparent semantics of the resulting graphs.

Summing up, in the lack of principled paradigms for formalising NL expressions and given the non-trivial choices involved, the relevant works afford varying degrees of expressivity in line with the considered application contexts.

2.3 Motivation & Approach

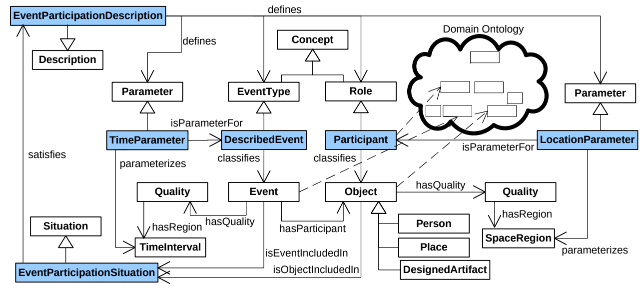

Figure 1: Participation pattern in Event-Model-F

We argue that question answering over conceptually rich KBs (e.g. proprietary models for maintaining care recipients knowledge in clinical institutions and organisations) poses additional challenges, as it requires support for both complex NL questions that involve rich relational contexts, and for flexible, context-aware question interpretation and answering paradigms. Consider as an example, the Event-Model-F [24] patterns. They extend DOLCE-DnS Ultralite and provide conceptual models for representing contextual knowledge about events, such as the participating entities (see Figure 1), as well as causal, correlative and mereological relations between events. Specialising the reified *n-ary* relational context semantics of DnS, the resulting event descriptions comprise highly axiomatised and rich structures, whose effective querying relies on coping with NL questions that allow capturing complex relations between entities and their respective roles.

Moreover, the additional annotation layer embodies, and consequently hides, direct contextual links among resources (e.g. between participants and events) hindering the automated generation of effective query patterns and bindings, unless the structure and axiomatization of the patterns is taken into account following domain-specific solutions, as in [9, 5]. The above shortcoming is evident even in conceptually simpler KBs. For instance, the Web of Know-How dataset[5] [21] contains activities and instructions collected from WikiHow and Snapguide. Although the vocabulary used to represent this knowledge is relatively simple (PROHOW[6]), the generated instantiations encapsulate rich axiomatizations. Figure 2 presents an example instantiation for capturing information about a recipe, which can be easily reused in our domain to model the way individuals perform certain activities. In this example, there is a conceptual gap between the semantics and structure of the questions (i.e. how to make pancakes) and the way information is captured in the dataset. In such cases, domain knowledge is needed to further drive the generation of (possibly) multiple queries to extract the required information from the patterns.

In contrast to the aforementioned approaches, our work builds upon and extends relevant paradigms in frame-based knowledge extraction from text and graph-driven query matching, explicitly addressing QA over pattern-based KBs. For capturing the NL questions semantics, we adhere to ontology design patterns principles, but advance related works by opting for reified event and situation representations that extend the DnS pattern and take into account the ontolog-

[2]http://www.cs.utexas.edu/users/ml/nldata.html
[3]http://verbs.colorado.edu/
[4]https://framenet.icsi.berkeley.edu/

[5]https://datahub.io/dataset/human-activities-and-instructions
[6]http://paolopareti.uk/prohow/vocab.php

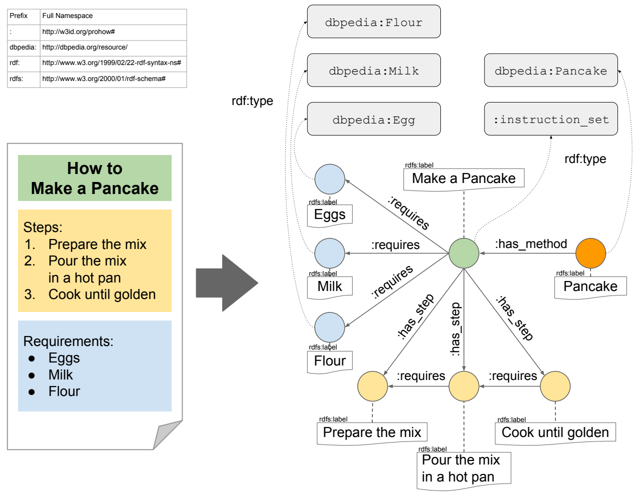

Figure 2: Example instantiation of PROHOW

ical types of the frames. For query interpretation, we have been inspired by the graph traversal paradigm, which we endowed with context-awareness, so that, given a set of query concepts and entities, we can assign context connections, i.e. links among groups of KB triples that satisfy the question. Our aim is to decouple graph expansion from predicate ranking , since in pattern-based modelling, additional layers of axiomatisation are introduced that encapsulate conceptual dependencies and links among resources. These dependencies are usually not relevant to the structure and semantics of questions and thus, cannot be uncovered by graph expansion approaches that are based on predicate ranking.

3. QUESTION ANALYSIS

Capturing the semantics of the natural language user inputs consists of the identification of the pertinent entities and their interrelations, and their subsequent formulation into corresponding semantic representations. In the following, we first present the NLP tools used for frame-based knowledge extraction and then detail the approach for translating the extracted linguistic structures into OWL graphs.

3.1 Linguistic Analysis

To extract linguistic frame-based representations from the NL user inputs we use the TALN frame semantics parser[7]. User inputs are first encoded as semantic predicate-argument structures that abstract away from syntactic variations and language-specific grammatical idiosyncrasies by graph transducers [4] that allow us to incrementally abstract from *surface-syntactic* dependencies to *deep-syntactic* ones, and eventually to semantic ones. Next, availing of SemLink[8] mappings between frame resources, the previously extracted predicate-argument structures are enriched with frame and frame elements annotations. In addition, Babelfy [20] is used for entity linking and word sense disambiguation against BabelNet[9], a multilingual semantic network that integrates several knowledge resources including WordNet and Wikipedia.

[7]https://github.com/talnsoftware/FrameSemantics_parser
[8]https://verbs.colorado.edu/semlink/
[9]http://babelnet.org/

3.2 Translation rules

The frame-based representations extracted during the linguistic analysis step abstract the NL user inputs with respect to conceptual structures (*frames*) that describe particular types of situations, objects, or events along with their participants (frame element fillers) and their roles (*frame elements*). For example, the *Apply_heat* frame describes a cooking situation involving, among others, a *Cook*, some *Food* and a *Heating_Instrument*; the roles of the involved participants, i.e. cook, food and heating instrument, comprise the frame elements (FEs) of the frame, while words that evoke it, such as fry, bake, boil, and broil, its lexical units (LUs) [23].

Inspired by [12] that explicates frame semantics in view of the Descriptions and Situations ontology pattern, we opt for a reified representation of the *n-ary* conceptual structures denoted by frames, interpreting frames as `dul:Descriptions`, frame elements as `dul:Concepts`, and the extracted frame occurrences as `dul:Situations`. This view is in line with FrameNet's intended semantics according to which "Frames describe classes of situations, the semantics of LUs are subclasses of the Frames, and (...) FEs are classes that are arguments of the Frame classes", where the term "Frame Element" has two meanings, namely "the relation itself, and the filler of the relation." [23]. However, the conceptual disparities between the linguistic considerations underpinning Frame-Net's intended semantics and knowledge engineering practices require a certain extent of re-engineering in order to obtain well-defined ontological representations.

Towards this end, we adopted a refined interpretation that takes into account the ontological type of the considered frames. Currently, we distinguish between frames that denote event-centric situations (e.g. *Ingestion*, *Grooming*), attributive ones (e.g. *Age*, *Usefulness*, *Measure_volume*), and frames that relate to objects (e.g. *Artifact*, *Food*).

Event frame situations are captured as specialisations of the class `EventFrameSituation`, which is defined as follows:

```
EventFrameSituation rdfs:subClassOf (
  dul:Situation and
  dul:satisfies some EventFrameDescription )
EventFrameDescription rdfs:subClassOf (
  dul:Description and
  dul:defines some InvolvedEvent )
```

For each extracted event frame occurrence, an instance of the respective frame situation class is introduced along with corresponding instance assertions for each of the participating entities, including the lexical unit that evoked the frame. *dul:isSettingFor* assertions are used to link the frame situation individual with the rest, while respective *dul:classifiedBy* assertions are used to described the semantic roles of the participating entities; the lexical unit class is further typed as a subclass of `dul:Event`. Thus, for example, the sentence "Ann drinks coffee" would result, among others, in the following assertions:

```
:IngestionFrame rdfs:subClassOf dul:Situation .
:ingestion1 rdf:type :IngestionFrame;
  dul:isSettingFor :drink1 , :coffee1 , :Ann.
:Drink rdfs:subClassOf dul:Event .
:coffee1 dul:classifiedBy :ingestibles1 .
```

To capture attributive frames, we have introduced the classes `AttributeFrameSituation` $\sqsubseteq$ `FrameSituation` and

`AttributeFrameDescription` $\sqsubseteq$ `FrameDescription`, while respective specialisations allow distinguishing between relative and absolute attribute descriptions. For example, absolute attribute descriptions specialise the following definition:

```
AbsoluteAttributeDescription rdfs:subClassOf (
  dul:Description and
  dul:defines some Attribute and
  dul:defines some dul:Region and
  dul:defines some dul:UnitType )
```

Lacking the descriptive contexts pertinent to event and attribute frames, frames related to objects are treated as specialisations of the class `dul:Entity`, also augmented with BabelNet and WordNet sense information.

Last, as the application context of the proposed NL interface for pattern-based KBs is part of a socially competent communicative agent, the generated semantic representations of the NL user input capture also information on speech act types. To this end, we have introduced the `SpeechAct` class, which specialising `dul:Situation` serves as a container for the `FrameSituation` objects included in a user utterance. Currently, we distinguish between requesting and informing speech acts using the classes `InformSpeechAct` and `RequestSpeechAct` respectively.

4. CONTEXT EXTRACTION

Context extraction involves the semantic interpretation of the analysed user question (Section 3) and the subsequent extraction, from the KB, of knowledge that satisfies the query context. In the rest of this section, we describe the steps involved in identifying key query concepts, their mapping on KB entities and the extraction of meaningful context from the KB that contextually answers the initial question.

4.1 Extraction of Key Entities

The first step of the algorithm is to extract the key entities of question analysis. As key entities, we define the entities that participate in DnS classification relations, since such axiomatizations encapsulate information about the context of questions. The key entities can be straightforwardly extracted by traversing the frame situation model, collecting the resources classified through `dul:classifies` property assertions. Assuming that k is a key entity, x is a resource and F is the language analysis model, the set K with all the key entities is defined as:

$$K = \{k | \langle x \texttt{ dul:classifies } k\rangle, \forall x \in F\}$$

4.2 Resource Identification

Having extracted the key entities K, the next step is to assign URIs to each $k \in K$. As described in Section 3, using Babelfy each classified entity is assigned to a WordNet synset. These annotations are used to detect entities (synonyms) in the KB that will drive the resource unfolding process described in Section 4.3. Assuming that $label(r)$ is the label of resource $r \in KB$, $syn(k)$ is the synset of key entity $k \in K$ and σ is a similarity function, the set $S(k)$ of all the relevant resources to k is defined as:

$$S(k) = argmax_{k \in K}\ \sigma(k, label(r))$$

The current implementation uses the UMBC Semantic Similarity Service [13] for simplicity, a ready-to-use service that calculates the semantic similarity σ between k and $label(r)$ combining Latent Semantic Analysis (LSA) word similarity and WordNet knowledge. The output of this step is the multiset $\mathcal{S}$ that contains the sets of all relevant resources of key entities in K:

$$\mathcal{S} = \{S(k) \mid k \in K\}$$

4.3 Resource Unfolding and Local Context

The next step is to define the *local context* for each entity $k' \in S(k)$ that captures information relevant to the neighbouring resources (triples) of k'. Therefore, the local context is built by taking into account all the connected triples with k', without examining the similarity of the predicate labels to entities and resources extracted through language analysis. This approach ensures that the local contexts contain information that is part of the conceptual model of the pattern, which is important since it encapsulates implicit contextual relations among key entities and their mappings that should not be ignored. For example, the question "How to make a pancake" does not directly entail that the predicates `requires` or `has_method` (Figure 2) should be part of the graph expansion algorithm, unless domain knowledge is taken into account.

Based on the mappings generated in the previous step, the local context generation task iteratively unfolds a resource k', traversing the KB vocabulary and collecting triples $\langle s, p, o\rangle$ whose subject, predicate or object is linked to k'. A threshold h is used to filter out triples that are more than h property assertions away from the element. More specifically, the local context $X_{k'}$ of resource k' is defined as:

$$X_{k'} = \{\langle s, p, o\rangle \mid k' \xrightarrow{h} \langle s, p, o\rangle, \forall k' \in S(k)\},$$

where $k' \xrightarrow{h} \langle s, p, o\rangle$ denotes all the triples directly or indirectly connected with k', up to h property assertions away. Intuitively, the aim is to enrich local contexts with additional contextual triples from the neighbourhood of key resource $k' \in S(k)$ in the KB. By computing the local context of each k', we create the set $\mathcal{X}$ of all the local contexts relevant to the question, i.e. $\mathcal{X} = \{X_{k'}, \forall k' \in S(k)\}$.

4.4 Context Links

Based on the local contexts $\mathcal{X}$ obtained in the previous section, the next step is to define *context links*. Intuitively, a context link captures a contextual dependency between two local contexts, with respect to the contained triples. For example, if two local contexts contain triples that share at least one common subject, predicate or object, then a contextual dependency is detected and the two local contexts are linked. OWL schema predicates (e.g. `rdfs:domain`) or classes (e.g. `owl:Thing`) are ignored during triple resource matching, in order not to generate generic, contextless dependencies among local contexts. More specifically, two local contexts X_k and X_m are linked, denoted as $X_k \mapsto X_m$, if $\exists\langle s_a, p_a, o_a\rangle \in X_k, \exists\langle s_b, p_b, o_b\rangle \in X_m$, such that $s_a = s_b \vee s_a = o_b \vee p_a = p_b \vee o_a = s_b \vee o_a = o_b$.

4.5 Context Ranking and Responses

The final step of the algorithm is to traverse the paths defined by context links $X_k \mapsto X_l ... \mapsto X_n$, collecting the triples $\langle s, p, o\rangle$ of local contexts in order to generate possible contextual responses. Intuitively, this step merges the local contexts of different key entities, capitalizing on the contextual dependencies identified in the previous step. More specifically, a response multiset $\mathcal{R}$ is defined as:

$$\mathcal{R} = \{X_k \cup X_l ... \cup X_n \mid X_k \mapsto X_l ... \mapsto X_n, \forall X_i \in \mathcal{X}\}$$

Each response set $R \in \mathcal{R}$ is semantically and structurally compared to language analysis results in order to rank them and select the most plausible context as final response to the input question. The ranking is based on two criteria:

- semantic similarity of triple resources in R with the key concept multiset $\mathcal{S}$.
- structural similarity of resource relations in R with the relations generated through language analysis.

More specifically, semantic similarity (φ) is computed taking into account the type of the resources that participate in ABox assertions (1). Intuitively, the multiset $\mathcal{S}$ of all key concepts (might be ontology classes, properties or instances) that have been identified in Section 4.2 are semantically compared to resources in each R.

$$\varphi(\mathcal{S}, R) = \frac{\sum\limits_{\forall S \in \mathcal{S}} \max\limits_{\forall r \in R, \forall k' \in S} \left[\delta(r, k')\right]}{|\mathcal{S}|} \tag{1}$$

We use the δ function (2) to compute the similarity of a key concept k' against a resource r of a triple in R as:

$$\delta(k', r) = \begin{cases} 1, & \text{if } r \sqsubseteq k' \text{ (includes } r \equiv k') \\ \frac{|U(r) \cap U(k')|}{|U(r)|}, & \text{if } k' \sqsubseteq r \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

If k' and r are classes, then their similarity derives based on their hierarchical relationship. A class r exactly matches a class k', if it is equivalent to k' or if it a subclass of k'. On the other hand, if k' is subsumed by r, then r is a more general concept than k' and the similarity is computed based on the rate of the superclasses of r that are also superclasses of k'. $U(C)$ is defined as the set of the superclasses of C, excluding `owl:Thing`, such that $U(C) = \{A \mid C \sqsubseteq A, A \neq \top\}$. If k' and r are instances (or properties), then the similarity derives based on resource equality ($\equiv$) (property hierarchies are not taken into account).

Semantic similarity takes into account only the type of resources involved in a response, without examining their connectivity. Structural similarity is used in order to favour responses whose structural relations of resources better reflect the key concept relations derived through language analysis. For example, if the key concepts `water` and `temperature` are connected in the language analysis results, then responses will be preferred where the corresponding resources are also connected (the distance between the resources is not taken into account). More specifically, assuming that L_C is the set with language analysis resource connections $[r_a, r_b]$ and R_C is the set with response resource connections $[r_1, r_2]$, their similarity is given by (3) and (4).

$$\gamma(R_C, L_C) = \frac{\sum\limits_{\forall [r_1, r_2] \in R_C} \delta'([r_1, r_2], L_C)}{|R_C|} \tag{3}$$

$$\delta'([r_1, r_2], L_C) = \begin{cases} 1, & \text{if } \exists [r_1, r_2] \in L_C \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The overall score of the response context $R \in \mathcal{R}$ with respect to the multiset $\mathcal{S}$ with language analysis key concepts and the set L_C with language analysis resource connections is defined as the weighted mean $\overline{sim}$ of φ and γ as:

$$\overline{sim}(R, \mathcal{S}, L_C) = \frac{a \cdot \varphi(\mathcal{S}, R) + b \cdot \gamma(R_C, L_C)}{a + b} \tag{5}$$

where a and b are normalized weights in [0..1], enabling the empirical adjustment of context ranking criteria. For example, a b weight close to 0 indicates a relaxed policy regarding structural similarity, enabling the return of contextual triples that are not necessarily part of the question. In contrast, a b weight close to 1 reflects a more strict policy to structural similarity, where additional contextual triples negatively affect the overall similarity.

5. EXAMPLE

To illustrate the question analysis and context extraction capabilities of the proposed framework, we use the preference pattern of Figure 3 and the user question *"How often does Ann like to drink coffee?"*.

5.1 User Preference Pattern

An important modelling aspect of user's behaviour is the availability of rich information about various activities of daily living (ADL). Figure 3 depicts the instantiation of the DnS pattern to capture the coffee drinking preferences of Ann. More precisely, the instantiation of DnS in DUL involves the definition of *situation* and *description* instances. The latter *defines* one ore more *concepts* that may further *classify* entities, describing in that way the context of a given situation of interest. That said, the preference pattern of the example defines the `Preference` situation (`Preference` $\sqsubseteq$ `dul:Situation`) and two domain concepts (`Drinkable` and `Ingredient`) for the classification of DUL entities that are involved in this pattern, i.e. `coffee` and `milk`. The `dul:EventType` is reused to classify the `Drink` event/class[10] and the `Frequency` concept to designate the frequency.

In addition, following the conceptual example of Event-Model-F, the situation instance is further associated through `dul:isSettingFor` property assertions with the entities that are classified by concepts. Instead of manually defining such relations, the preference pattern uses the property chain axiom: $describes \circ defines \circ classifies \sqsubseteq isSettingFor$.

5.2 Question Analysis

Applying the afore-described question analysis methodology, the resulting user input knowledge graph comprises information about the speech act type (i.e. request) and the encompassed frame situation occurrences, as shown in the following Turtle extract:

```
:speechAct1 rdf:type RequestSpeechAct ;
  dul:isSettingFor :ingestion1 ,
  dul:isSettingFor :frequency1 ,
  dul:satisfies :requestDesc1 .
:ingestionSit1 rdf:type IngestionSituation ;
  dul:isSettingFor :coffee1 ,
  dul:isSettingFor :Ann ,
```

[10] In DUL, the `dul:EventType` concept classifies `dul:Event` instances. In this example though, we use a class (`Drink`), which conforms to the OWL 2 DL semantics (*punning* [14]).

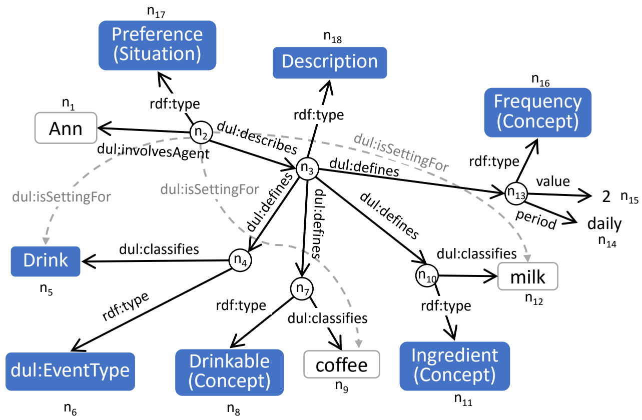

Figure 3: Coffee drinking pattern in DnS

```
  dul:includesEvent :drink1 ,
  dul:satisfies :ingestionDesc1 .
:ingestionDesc1 rdf:type IngestionDescription ;
  dul:defines :ingestor1,
  dul:defines :ingestisble1 .
:frequencySit1 rdf:type FrequencySituation ;
  dul:isSettingFor :ingestionSit1 ,
  dul:isSettingFor :frequency1 ,
  dul:satisfies :frequencyDesc1 .
:drink1 rdf:type Drink ;
  dul:classifiedBy :eventType1 .
Drink owl:equivalentClass wn30-synset%drink-verb-1 .
:coffee1 rdf:type Coffee ;
  dul:classifiedBy :ingestible1  .
Coffee owl:equivalentClass wn30-synset%coffee-noun-1 .
  :Ann rdf:type dul:Person ;
dul:classifiedBy :ingestor1 .
  :frequency1 dul:classifiedBy :request1 .
```

5.3 Context Extraction

The first step of the procedure is to extract the key entities recognized through question analysis. Based on the results described in Section 5.2, the following key entities are extracted

$$K = \{Ann_F, drink1_F, coffee1_F, frequency1_F\},$$

which are mapped on the following resources in the KB:

$$\mathcal{S} = \{Ann_{KB}, Drink_{KB}, coffee_{KB}, Frequency_{KB}\}$$

The next step is to unfold the resources and create the local contexts. We use $h = 2$ and we omit the triples of local contexts, presenting only the connected node ids (illustrated in Figure 3) for presentation purposes. As such, we have the following local contexts for each mapped resource $x' \in \mathcal{S}$:

$$X_{Ann_{KB}} = \{n_2, n_{17}, n_5, n_9, n_{12}, n_3\}$$

$$X_{Drink_{KB}} = \{n_2, n_{17}, n_5, n_9, n_{12}, n_3, n_1, n_4, n_6\}$$

$$X_{coffee_{KB}} = \{n_7, n_2, n_{17}, n_5, n_9, n_{12}, n_3, n_1, n_8\}$$

$$X_{Frequency_{KB}} = \{n_3, n_{15}, n_{14}, n_3\}$$

All local contexts share at least one common resource. Therefore, all local contexts are pairwise linked: $X_{Ann_{KB}} \mapsto X_{Drink_{KB}}, X_{Ann_{KB}} \mapsto X_{coffee_{KB}}, X_{Ann_{KB}} \mapsto X_{Frequency_{KB}}$ (and so on). As such, we have a response multiset $\mathcal{S}$ with a single response $R \in \mathcal{S}$:

$$R = \{n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9, n_{12}, n_{13}, n_{14}, n_{15}, n_{16}, n_{17}\}$$

The semantic similarity equals to 1, since all key entities are exactly matched to the resources of the response. Regarding structural similarity, we can observe that the `milk` resource (n_{12}) is also returned, which is not part of the entities detected through question analysis. This is an example of additional contextual information that may be returned by our framework and can be controlled through the b weight: a high b weight (e.g. "1") would reduce the final similarity of the response, penalizing the additional context that is not part of question analysis results.

As such, the framework provides a contextual response to the question, returning not just a plain value (e.g. "2" in this example), but also the semantics of the answer, e.g. "2 times daily", in a formal pattern-based manner. Such pattern-based responses foster their further processing in different application scenarios, e.g. in dialogue-based systems where agents need to interpret responses and act accordingly, or for generating verbal responses.

6. CONCLUSIONS

Question answering over conceptually complex, pattern-based KBs aggravates further the challenges involved in coping with NL queries over Semantic Web data, as the underlying rich and encapsulated semantics accentuate the need for accurately capturing the semantic structure of complex user questions, while urging for flexible, context-aware query interpretation. In this work, we presented a framework towards QA over pattern-based user models that combines the frame-based reified representation of NL questions with a context-aware, graph-based paradigm for interpreting them against KBs and identifying pertinent answers.

We are currently building rich KBs capturing user models of participants in KRISTINA[11] pilots. The collected data will allow us to evaluate our framework with realistic data, identifying possible limitations that have not been foreseen so far. In parallel, we are working towards further enrichment of the analysis and interpretation of complex relational context so as to support for additional constructions, such as negation, superlatives and aggregation, that will allow for more expressive QA over the profiled users routines.

7. ACKNOWLEDGEMENTS

This work has been partially supported by the H2020-645012 project "KRISTINA: A Knowledge-Based Information Agent with Social Competence and Human Interaction Capabilities".

8. REFERENCES

[1] N. Aggarwal and P. Buitelaar. A System Description of Natural Language Query over DBpedia. *Proc. of Interacting with Linked Data*, pages 96–99, 2012.

[2] Y. Amsterdamer, A. Kukliansky, and T. Milo. A Natural Language Interface for Querying General and Individual Knowledge. *Proc. of the VLDB Endowment*, 8(12):1430–1441, 2015.

[11] http://kristina-project.eu/en/

[3] I. Augenstein, S. Padó, and S. Rudolph. LODifier: Generating Linked Data from Unstructured Text. *Proc. of Extended Semantic Web Conference*, pages 210–224, 2012.

[4] M. Ballesteros, B. Bohnet, S. Mille, and L. Wanner. Data-driven deep-syntactic dependency parsing. *Natural Language Engineering*, pages 1–36, 2015.

[5] P. Cimiano, P. Haase, and J. Heizmann. Porting Natural Language Interfaces between Domains – An Experimental User Study with the ORAKEL System. *Proc. of Intelligent User Interfaces*, pages 180–189, 2007.

[6] F. Corcoglioniti, M. Rospocher, and A. P. Aprosio. A 2-phase Frame-based Knowledge Extraction Framework. *Proc. of ACM Symposium on Applied Computing*, pages 354–361, 2016.

[7] D. Damljanovic, M. Agatonovic, and H. Cunningham. FREyA: An Interactive Way of Querying Linked Data Using Natural Language. *Proc. of Extended Semantic Web Conference Workshops*, pages 125–138, 2011.

[8] R. de Almeida Falbo, M. P. Barcellos, J. C. Nardi, and G. Guizzardi. Organizing Ontology Design Patterns as Ontology Pattern Languages. *Proc. of Extended Semantic Web Conference*, pages 61–75, 2013.

[9] A. Frank, H.-U. Krieger, F. Xu, H. Uszkoreit, B. Crysmann, B. Jörg, and U. Schäfer. Question Answering from Structured Knowledge Sources. *Journal of Applied Logic*, 5(1):20 – 48, 2007.

[10] A. Freitas, J. G. Oliveira, S. O'Riain, J. C. da Silva, and E. Curry. Querying linked data graphs using semantic relatedness: A vocabulary independent approach. *Data & Knowledge Engineering*, 88:126 – 141, 2013.

[11] A. Gangemi. Ontology Design Patterns for Semantic Web Content. *Proc. of International Semantic Web Conference*, pages 262–276, 2005.

[12] A. Gangemi. *What's in a Schema?* C. Huang, N. Calzolari, A. Gangemi, A. Lenci, A. Oltramari, and L. Prevot, editors, Ontology and the Lexicon. Cambridge University Press, 2010.

[13] L. Han, A. Kashyap, T. Finin, J. Mayfield, and J. Weese. UMBC EBIQUITY-CORE: Semantic Textual Similarity Systems. *Proc. of Joint Conference on Lexical & Computational Semantics*, pages 44–52, 2013.

[14] N. Jekjantuk, G. Gröner, and J. Z. Pan. Modelling and Reasoning in Metamodelling Enabled Ontologies. *Proc. of Knowledge Science, Engineering and Management*, pages 51–62, 2010.

[15] H. Kamp and U. Reyle. From Discourse to Logic. *Dordrecht: Kluwer Academic Publishers*, 1993.

[16] E. Kaufmann, A. Bernstein, and L. Fischer. NLP-Reduce: A "naive" but Domain-independent Natural Language Interface for Querying Ontologies. *Proc. of Extended Semantic Web Conference*, 2007.

[17] V. Lopez, M. Fernández, E. Motta, and N. Stieler. PowerAqua: Supporting users in querying and exploring the Semantic Web. *Semantic Web*, 3(3):249–265, Aug. 2012.

[18] V. Lopez, C. Unger, P. Cimiano, and E. Motta. Evaluating question answering over linked data. *Web Semantics Science Services And Agents On The World Wide Web*, 21:3–13, 2013.

[19] V. Lopez, V. Uren, M. Sabou, and E. Motta. Is Question Answering fit for the Semantic Web?: A Survey. *Semantic Web*, 2(2):125–155, Apr. 2011.

[20] A. Moro, A. Raganato, and R. Navigli. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *TACL*, 2:231–244, 2014.

[21] P. Pareti, B. Testu, R. Ichise, E. Klein, and A. Barker. Integrating Know-How into the Linked Data Cloud. *Proc. of Knowledge Engineering and Knowledge Management*, pages 385–396. 2014.

[22] V. Presutti, F. Draicchio, and A. Gangemi. Knowledge Extraction Based on Discourse Representation Theory and Linguistic Frames. *Proc. of Knowledge Engineering and Knowledge Management*, pages 114–129, 2012.

[23] J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, C. R. Johnson, and J. Scheffczyk. *FrameNet II: Extended Theory and Practice*, 2010, https://framenet2.icsi.berkeley.edu/docs/r1.5/book.pdf.

[24] A. Scherp, T. Franz, C. Saathoff, and S. Staab. F–A Model of Events based on the Foundational Ontology DOLCE+DnS Ultralite. *Proc. of Knowledge Capture (K-CAP)*, pages 137–144, 2009.

[25] S. Shekarpour, S. Auer, A.-C. N. Ngomo, D. Gerber, S. Hellmann, and C. Stadler. Keyword-driven SPARQL Query Generation Leveraging Background Knowledge. *Proc. of Web Intelligence and Intelligent Agent Technology*, pages 203–210, 2011.

[26] C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano. Template-based Question Answering over RDF Data. *Proc. of International Conference on World Wide Web*, pages 639–648, 2012.

[27] C. Unger and P. Cimiano. Pythia: Compositional meaning construction for ontology-based question answering on the Semantic Web. *Proc. of Applications of Natural Language to Information Systems*, pages 153–160, 2011.

[28] C. Unger, A. Freitas, and P. Cimiano. An introduction to Question Answering over Linked Data. *Reasoning Web Summer School*, pages 100–140, 2014.

[29] R. Usbeck, A.-C. N. Ngomo, L. Bühmann, and C. Unger. HAWK – Hybrid Question Answering Using Linked Data. Proc. of European Semantic Web Conference, pages 353–368. 2015.

[30] W. Zheng, L. Zou, X. Lian, J. X. Yu, S. Song, and D. Zhao. How to Build Templates for RDF Question/Answering: An Uncertain Graph Similarity Join Approach. *Proc. of International Conference on Management of Data*, pages 1809–1824, 2015.

[31] C. Zhu, K. Ren, X. Liu, H. Wang, Y. Tian, and Y. Yu. A Graph Traversal-based Approach to Answer Non-Aggregation Questions over DBpedia. *Proc. of Joint International Conference on Semantic Technology*, pages 219–234, 2015.

[32] L. Zou, R. Huang, H. Wang, J. X. Yu, W. He, and D. Zhao. Natural Language Question Answering over RDF: A Graph Data Driven Approach. *Proc. of ACM SIGMOD International Conference on Management of Data*, pages 313–324, 2014.