Towards an Ontology-Driven Adaptive Dialogue Framework

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

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

Louisa Pragst
Institute of Communications Engineering, Ulm University, Germany
louisa.pragst@uni-ulm.de

Stefan Ultes
Department of Engineering University of Cambridge, UK
su259@cam.ac.uk

Stefanos Vrochidis, Ioannis Kompatsiaris
Information Technologies Institute, CERTH, Greece
{stefanos, ikom}@iti.gr

Leo Wanner
Department of Information and Communication Technologies, UPF, Spain
leo.wanner@upf.edu

ABSTRACT

In this paper, we describe the principles and technologies that underpin the development of an adaptive dialogue manager framework, tailored to carrying out human-agent conversations in a natural, robust and flexible manner. Our research focus is twofold. First, the investigation of dialogue strategies that can handle dynamically created user and system actions, while still enabling the agent to adapt its actions to various and possibly changing contexts. Second, the utilisation of rich semantic annotations for capturing background knowledge, as well as conversation topics and semantics of user utterances extracted through language analysis. The resulting annotations comprise the situational descriptions upon which reasoning takes place to recognise the conversation context and compile appropriate responses.

Keywords

Dialogue systems, language analysis, question answering, ontologies

1. INTRODUCTION

A key prerequisite in dialogue systems is to afford effective strategies for tailoring system behaviour to user actions and ensuring meaningful and coherent interactions. Current dialogue managers (e.g. [19]), however, often restrict their scope to predefined sets of possible user and system actions, severely undermining thereby flexibility and robustness. The latter depend largely on bridging the gap between the users' perception of the domain of discourse and the way domain knowledge is captured. Towards this direction, there have been important advances recently in semantic search and Question Answering (QA) over structured data [13], exploiting ontological relationships to understand and disambiguate a query. However, most of the existing approaches consider syntactic relations (e.g. subject and object dependencies) rather than semantic ones, thus capturing only partially the underlying language semantics.

In this work, extending the proposal presented in [17], we combine advanced techniques in the fields of language analysis (LA), dialogue management, knowledge representation and reasoning to support an adaptive dialogue flow and react flexibly to user input without restricting the wording or the way questions are formulated. More specifically:

- We delineate a semantic language analysis framework that allows to abstract away from syntactic variations and formalise user utterances in OWL.
- We present a context-aware query answering algorithm that combines OWL 2 ontologies and rules to feed the dialogue agent with ongoing situational information.
- We introduce a dialogue management approach that handles dynamically created user and system actions, utilising general dialogue acts combined with ontology semantics to determine the system's behaviour.

The rest of the paper is structured as follows: Section 2 presents related work in the domains of QA, LA, and DM. Section 3 describes the architecture of the framework, elaborating on the provided functionality and component interactions. Section 4 presents an example use case, while Section 5 concludes our work.

2. RELATED WORK

2.1 Capturing Language Semantics

Several works have investigated knowledge extraction from natural language text and its capturing into Semantic Web compliant representations using deep parsing to abstract away from syntactic variations, predicate-argument resources

(e.g. FrameNet[1]) for semantic role labelling, and respective mapping rules for their formalisation [16, 2, 18]. The idiosyncrasies of deep parsing approaches in combination with the non-trivial decisions involved in re-engineering the linguistic rather than ontological considerations underlying the extracted data, impact accordingly the resulting ontological representations; for instance, in [18], only verbal events are considered, while the use of blank nodes in [2] hinders subsequent reasoning tasks.

In the context of DM and QA though, user utterance semantics tend to be considered in a less comprehensive manner. DM systems often rely on restricted, domain-tailored lexical to semantics bindings in order to contextualise and interpret user utterances (e.g. [19, 14]); in contrast, ontology-based question answering approaches tend to allow overall for a greater flexibility, by employing dependency parsing in order to transform user queries into some intermediate structured representation [13]; however, as the deployed dependency parsers address primarily syntactic rather than semantic dependencies, the resulting representations do not effectively capture the wealth of user query semantics (e.g. n-ary relations, events and participant roles).

2.2 Ontology-based Question Answering

Several approaches have been proposed in the literature [13] to enrich QA with ontologies. PowerAqua [12] allows users to choose an ontology and then ask queries relevant to the vocabulary. The results are transformed into triples, which are further annotated with ontology resources. The triples are translated into logical queries that retrieve answers from the knowledge sources. NLP-Reduce [11] 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 knowledge base and expanded with WordNet synonyms, generating SPARQL statements for the matches. FREyA [9] is an interactive Natural Language Interface for querying ontologies. It combines syntactic parsing with ontology-based lookup in an attempt to answer questions. In [10] RDF ontologies are used to populate a KB with product descriptions, while SPARQL queries are generated taking into account domain and range restrictions.

Other systems, such as [1, 20], follow a different approach. Instead of generating SPARQL queries, QA is reduced to a subgraph matching problem. Either way, the focus is mainly put on retrieving answers, without supporting further interaction with the users. Interactivity is often limited to solving disambiguation problems (e.g. in FREyA) and requesting clarifications from users. The possibility to adapt the dialogue strategy to the user is usually not given.

2.3 Dialogue Management

There have been efforts to separate the domain model of the DM, and thereby the available system and user actions, from the dialogue flow control. The RavenClaw DM [5] consists of a Dialog Task Specification layer, which models domain and domain-specific dialogue logic, and a domain-independent Dialog Engine. The Dialog Task Specification Layer is realised as hierarchical structure of tasks to be accomplished in the dialogue. The Dialog Engine incorporates various domain independent conversational strategies, such as turn-taking behaviour or help and repeat actions.

[1]https://framenet.icsi.berkeley.edu/fndrupal/

Nothdurft et al. [15] proposed an architecture, in which a DM and a reasoner worked together to provide explanations for the system's proposed course of action and thereby sustain the user's trust. By comparing a Finite State Machine with a decision tree resulting from a POMDP, potential points of distrust are identified and the DM inserts an explanation for the system's proposal. In [14], the authors use OWL ontologies to model information regarding the digital TV subscription domain. However, no further details are provided regarding the way ontology reasoning is used to support the dialogue management tasks.

3. ADAPTIVE DIALOGUE FRAMEWORK

The proposed framework, which aims to address the described challenges, is structured around three pillars:

Semantic language analysis: Semantic analysis of the natural language user input to extract and formalise the underlying semantics (in terms of the pertinent entities and their interrelations) as ontological representations so that the user input can be interpreted against the KB.

Question answering and feedback: Semantic processing of the LA results to identify conversation topics and to provide a context-aware representation of the situation that drives the query answering and feedback tasks.

Dialogue management: Selection of an appropriate system reaction considering the dialogue state as well as the feedback of QA. This decision process can be further adapted, e.g. to the user's emotion or culture.

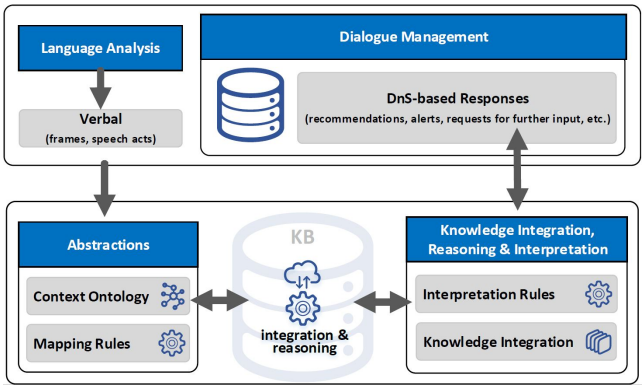

Figure 1: Adaptive DM framework architecture.

Figure 1 illustrates the logical architecture, while the tasks involved in each pillar are described in the following.

3.1 Semantic Language Analysis

In order to capture and formalise the natural language user utterances in OWL, the proposed semantic language analysis framework uses shallow semantic parsing, FrameNet-based role labelling and the design principles underpinning DUL[2]'s Description and Situation (DnS) pattern.

3.1.1 FrameNet-based structure extraction

Advancing beyond the current DM and QA practices of syntactic dependency-based analysis, the user input is analysed in terms of semantic dependencies, through the use of

[2]www.loa.istc.cnr.it/ontologies/DUL.owl

graph transducers [3] that allow its encoding in predicate-argument structures that abstract away from language-specific idiosyncrasies. The extracted predicate-argument structures are subsequently further enhanced with frame and corresponding frame elements annotations using a frame semantics parser[3]. Frames are linguistically-grounded conceptual structures that describe particular types of relational contexts along with the semantic roles of the pertinent entities. For example the Ingestion frame describes a situation involving a Ingestor and some Ingestibles, and is evoked by words such as "drink", "nibble", "eat", "devour", etc.; the roles are called frame elements and the frame-evoking words lexical units (LUs). For example, in the sentence "Elif drinks coffee", "drinks" is the lexical unit that evokes the Ingestion frame, while "Elif" and "coffee" would be annotated as the frame elements Ingestor and Ingestibles respectively.

3.1.2 DnS-based ontological translation

To formalise the extracted FrameNet-based structures and map them into corresponding ontological representations, frame patterns are considered as specialisations of the DnS pattern: frames are interpreted as `dul:Descriptions`, frame elements as `dul:Concepts` that determine how the involved entities should be interpreted, while the extracted frame occurrences correspond to `dul:Situations`. In addition, each frame-based situation is annotated with its corresponding dialogue acts, e.g. request and statement.

To account for the disparate semantics that the various frame categories admit to, we distinguish between frames that denote event situations (e.g. Ingestion, Grooming), frames that are related to attributes (e.g. Age, Usefulness), and frames that relate to objects (e.g. Containers, Food).

More specifically, event frame situations specify the class `EventFrameSituation` defined as follows:

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

For each extracted event frame occurrence, an individual of the respective frame situation class is introduced and linked with the individuals corresponding to the participating entities and the lexical unit that evoked the frame; the latter is further typed as a subclass of `dul:Event`. Thus, for example, the sentence "Elif drinks coffee" would result among others in the assertions:

```
:IngestionFrame rdfs:subClassOf dul:Situation .
:ingestion1 rdf:type :IngestionFrame;
  dul:satisfies :description1.
:desciption1 rdf:type dul:Description;
  dul:defines [dul:classifies :drink1]
:Drink rdfs:subClassOf dul:Event .
:drink1 rdf:type :Drink.
```

Likewise the classes `AttributeFrameSituation` $\sqsubseteq$ `FrameSituation` and `AttributeFrameDescription` $\sqsubseteq$ `FrameDescription` are introduced to capture attributive frames, while respective specialisations allow distinguishing between relative and absolute attribute descriptions. For example, absolute attribute descriptions specialise the following definition:

```
AbsoluteAttributeDescription SubClassOf (
  dul:Description and
  dul:defined some InvolvedAttribute 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 interpreted as specialisations of the class `dul:Entity`.

3.2 Domain Modelling and Reasoning

A number of ontologies have been developed to support context abstraction, reasoning and feedback. These include a) Ontologies that capture the various types of background knowledge involved (user profile, medical information, etc.) in compiling appropriate system responses, and b) Mapping and interpretation models that abstract incoming information into topics, enabling the derivation of situations of interest, answers and feedback.

Regarding the modelling of domain information, the framework does not impose any restriction on the vocabularies used to capture background knowledge. As such, existing foundational ontologies (e.g. DUL), design patterns and vocabularies (e.g. MeSH) can be used to capture knowledge.

The remainder of this section describes the mapping and interpretation of models and rules that enable the recognition of conversation contexts and their further coupling with background knowledge. The objective is to support the derivation of abstract conceptualisations about conversation topics that designate the reasoning task to be triggered to generate meaningful responses to user input.

3.2.1 Context ontology

The context ontology defines the semantics of conversation topics. It is used to abstract detected frames into higher level situations, capitalising on the OWL semantics for defining multi-level concept hierarchies. The ontology defines a hierarchy of topics extending the `dul:Situation` concept.

The current implementation supports two contexts: *Biographical context*, which captures information regarding biographical attributes, such as age, birthdate, etc., and *Behaviour context*, which captures user routines and preferences (e.g. daily water intake). As an example, we present the semantics of the complex class description that recognises routine conversation contexts based on the recognition of routine-related events by LA[4].

```
RoutineContext EquivalentTo dul:Situation and
  (dul:satisfies some (dul:defines some
    (dul:classifies some ent:RoutineEvent)))
```

This abstract context can be further specialised to capture concrete routine contexts. For example, the `DrinkingContext`, which captures conversation context relevant to drinking routines, is defined with the following two axioms:

```
DrinkingContext EquivalentTo core:IngestionFrame
  and (dul:includesEvent some ent:Drink) .
DrinkingContext SubClassOf
  fluid some owl:Thing, period some :Period .
```

[3] https://github.com/talnsoftware/FrameSemantics_parser

[4] The example uses the Manchester OWL Syntax.

Each frame relevant to ingestion is classified as drinking, provided that the frame also includes a routine-related event (in this example, drinking). In addition, the drinking context is associated with property assertions that characterize the drinking element (`fluid`) and period (e.g. daily, weekly).

3.2.2 Rule-based mapping

Despite the fact that OWL 2 ontologies support a rich set of semantics, there are still certain expressive limitations. For example, the native semantics of OWL 2 does not allow the dynamic generation of new individuals. In order to overcome OWL 2 reasoning shortcomings, we follow a hybrid context interpretation approach, complementing the ontology-based mapping with rules. More precisely, we use SPARQL construct graph patterns, enabling property value propagation and instance generation. The core idea is to associate each context concept with one or more SPARQL rules that address specific mapping tasks, e.g. the mapping of property values between frames and the context ontology.

As an example, we present the mapping rule we have defined to populate the `DrinkingContext`s presented in Section 3.2.1 with information about the ingestible (e.g. the fluid).

```
CONSTRUCT {?this :fluid ?fluid}
WHERE {
 ?this a :DrinkingContext;
   dul:satisfies [dul:defines ?c] .
 ?c a core:Ingestibles.ingestion .
 ?c dul:classifies ?fluid .
}
```

Having mapped the frames into the contextual hierarchy, the last step is to query the KB and retrieve the results. This is achieved by a set of domain-dependent SPARQL queries that are attached to context concepts. We present examples of such SPARQL queries in Section 4.2.

3.3 Adaptive Dialogue Management

The task of the DM is to determine an appropriate system reaction to the user input. This can be realised by a policy function mapping all combinations of user actions and dialogue states to system actions. Such a mapping can be either rule-based or learned from training data. However, in both approaches a usual assumption is that the system and user actions are known when defining the policy.

Questions arise if actions are created dynamically after the policy is defined: unknown system actions do not have a corresponding mapping to a user action and unknown user actions would not be chosen as all mappings are already occupied by existing user actions.

In this work we aim to address the following questions:

- How can user utterances be mapped to user actions without the use of grammars?

- How can unknown user actions be correlated with existing ones to estimate a good policy?

- How are unknown system actions generated?

- How can unknown system actions be evaluated in regard to their appropriateness in a given situation?

In order to handle these tasks, we use generalisation. Instead of concretely defining all possible actions, we utilise general dialogue acts [6, 8, 7] and the hierarchical structure of the ontology to describe them in a more general manner.

More specifically, the LA provides the DM with a classification of the user utterance in terms of the dialogue acts request and statement; if applicable, the DM can be further refined to more specialised dialogue acts (e.g. confirm, affirm) by taking into account the dialogue history. In addition, the semantic content of the user utterance, which is extracted by the LA, provides its topic. The hierarchy of the ontology further helps to rank the user action: if a user action with the topic "medicine for headache" has not been anticipated the super topic "medicine" can be used instead.

System actions are dynamically generated from the feedback of the QA module. Information that is missing to complete a query leads to a *request* action, while the result of a query is presented as *inform* action. Similar to the user actions, system actions have a topic depending on their semantic content. The hierarchy of the ontology can then be used to rank unknown system actions.

The appropriateness of a system action is derived considering its general dialogue act as well as its topic (or a superclass of it). This way, the policy for known actions can be extrapolated to unknown actions by shared characteristics.

4. EXAMPLE

Studies show [4] that in migration circles, especially of Turkish origin, there is a low take-up of care services and the provision system is insufficiently aligned with the needs, e.g. comprehension problems relevant to the culture and habits of the care recipient. In such cases, dialogue systems can act as mediators between migrants and caregivers.

In the considered scenario, Elif is an elderly Turkish migrant in a retirement home in Germany who receives support by a German caregiver. Information about the habits and preferences of the care recipient are needed, however, there may be problems of comprehension concerning the culture and habits. The required information can be provided to the dialogue system by Elif, and acquired by the professional caregiver through interaction with the system. We demonstrate the functionality of our framework on the basis of the following example dialogue:

USER/CAREGIVER: How much does Elif drink?
SYSTEM: Elif drinks two glasses of water per day.

4.1 Extracting User Input Semantics

Applying the semantic language analysis described in Section 3.1, the user input is mapped to the knowledge graph depicted in Figure 2. The knowledge graph (DnS pattern) contains structured information that describes (i) the dialogue act (request), (ii) the frame type (ingestion), (iii) the ingestor (person), and (iv) the event (drink). Note that the user has not provided information about the ingestible (e.g. water, beverage) and the period (e.g. daily, weekly).

4.2 Context Recognition and Feedback

4.2.1 User behaviour modelling

Profile information about the drinking routines is captured through instantiations of the DnS pattern. We define the `Frequency` and `View` concepts for modelling the situation and description, respectively. The view instance defines two concepts for modelling the occurrence context and the event type (`Drink`). The former encapsulates information about the period (e.g. `daily`), while the latter points to the entity type (e.g. `Water`). In addition, the occurrence context is associated with the frequency value, which is defined through

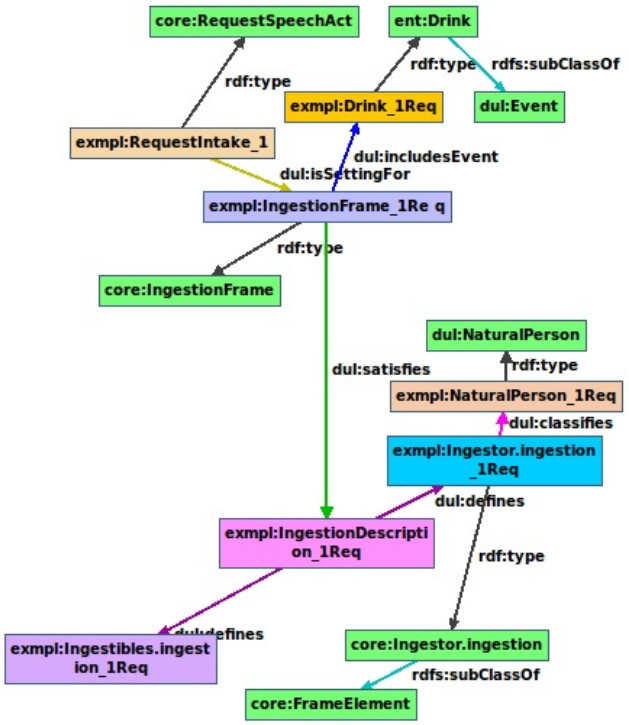

Figure 2: An RDF graph that captures user input.

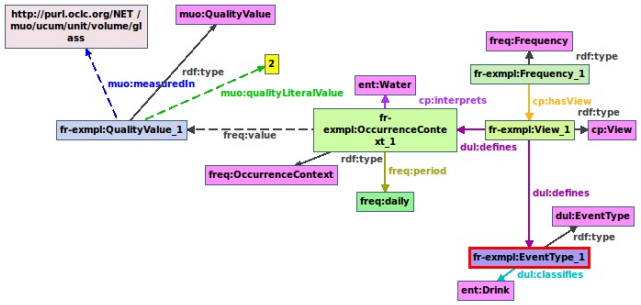

Figure 3: Modelling the user's drinking routine.

an instance of `muo:QualityValue`. This way, we are able to capture information about the unit of measurement (e.g. glass). Figure 3 presents an example instantiation (DnS) that defines the glasses of water the user drinks per day.

4.2.2 *Context recognition*

Based on the semantics of the drinking context described in Section 3.2.1, the OWL 2 reasoning process classifies the `IngestionFrame_1Req` instance of Figure 2 in the `Drinking-Context` class, since all the class restrictions are satisfied. However, the SPARQL rule described in Section 3.2.2 is not able to fill in the `fluid` property of the drinking context, since no relevant information is provided by the user.

4.2.3 *Results and feedback*

In order to answer questions about recognised contexts, we associate each context with SPARQL queries. As such, the `DrinkingContext` class is associated with the following SPARQL query that retrieves the amount, type, unit and period of the drinking routine (e.g. 2 glasses of water daily).

```
SELECT ?val ?unit ?period ?stuff
WHERE {
 ?dc a context:DrinkingContext;
  uomvocab:qualityValue [uomvocab:measuredIn ?unit];
  context:fluid ?stuff; context:period ?period;
  dul:includesEvent ?eventType .
  {
   SELECT ?unit ?eventType ?val ?stuff ?period
   WHERE {
    ?p a freq:Frequency ; cp:hasView ?v .
     ?v dul:defines [dul:classifies ?eventType] .
     ?v dul:defines [cp:interprets ?stuff ;
      freq:period ?period; freq:value [
        uomvocab:measuredIn ?unit ;
         uomvocab:qualityLiteralValue ?val]]
}}}
```

However, in our example the context does not include any fluid and period bindings, and therefore the `WHERE` graph pattern is not matched. In such cases, we collect the unbound variables and try to find resources in the KB that satisfy the semantic restrictions imposed by the ontology. For example, the `context:fluid` property has as range the class `Fluid`. The system collects all the instances that belong to the class hierarchy of fluids and periods and selects the most probable ones, based on information collected from past conversation. Assuming that the bindings `?stuff = water` and `?period = daily` are selected, the SPARQL query successfully retrieves the glasses of water the user drinks each day (`?val=2`, `?unit=glass`, `?period=daily`, `?stuff=water`).

In addition, DM has to be informed about the feedback needed regarding the drinking type and period, so as to provide an alternative response, if needed. This is achieved by returning to the DM a partial DnS instantiation that captures missing information. The final response to the DM contains both the variable bindings of the SPARQL query and the missing information required.

```
:ResultContext rdfs:subClassOf dul:Situation .
:result1 rdf:type :ResultContext;
  dul:satisfies :description.
:desciption rdf:type dul:Description;
  dul:defines [dul:classifies [dul:hasDataValue "2"]]
  dul:defines [dul:classifies [muo:measuredIn :glass]]
  dul:defines [dul:classifies :daily]
  dul:defines [dul:classifies :water]

:MissingContext rdfs:subClassOf dul:Situation .
:missing1 rdf:type :MissingContext
  dul:isSettingFor Fluid, Period .
```

4.3 DM Inference

The DM generates three possible system actions from the feedback of the QA module: a request action for each missing information and an inform action for the result of the query. Which of these actions is selected to be the system response depends on the dialogue strategy and the dialogue state, as elaborated in the following examples.

The decision of the DM is influenced by the dialogue history. If the caregiver and the system have been talking about the significance of drinking enough water during the day just

before the example dialogue, this information supports the conclusion of the QA module regarding the type of fluid and period of time. Therefore, the inform action is selected. On the other hand, if the caregiver has been talking about the dangers of drinking too much soft drinks per day, this renders the concluded type of fluid `water` unlikely. In this case, the DM decides to request this missing information from the caregiver. A further improvement of the cooperation between DM and QA module can be achieved by integrating the dialogue history in the reasoning process of the QA.

The previous examples show DM decisions based on the general dialogue acts of the available system actions. However, the topic can be of importance as well. Assuming that the DM decided to schedule a request action, the topic of the request actions can be utilised in order to decide which one is better suited in the current situation. The dialogue history contains as last user action a request with the topic amount of fluid. The request with the topic kind of fluid is closely related to that last user action, therefore the dialogue strategy chooses it as next system action.

5. CONCLUSIONS

Designing a flexible dialogue system involves many non-trivial decisions. In this paper, we described an approach to utilise an ontology-based QA module as domain model of the DM and integrate it into the process of dialogue flow control. Furthermore, a DnS-based pattern approach is used to capture and leverage the semantics of the user utterances with the underlying domain models. As a result, user and system actions are dynamically generated and the dialogue strategy does not depend on prior predefined actions.

As a future work, emphasis will be placed on extending the disambiguation capabilities of LA, incorporating, in addition to the FN-based annotations, mappings against BabelNet synsets, so as to cater for the development of a semi-automated rule definition approach that will enable the dynamic translation of meta-patterns into queries. In addition, the incorporation of non-verbal aspects, such as user emotion, in the DM selection strategy will be investigated.

6. ACKNOWLEDGEMENTS

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

7. 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] I. Augenstein, S. Padó, and S. Rudolph. Lodifier: Generating linked data from unstructured text. In *The Semantic Web: Research and Applications - Extended Semantic Web Conference*, pages 210–224, 2012.

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

[4] I. Bermejo, L. Hölzel, L. Kriston, and M. Härter. [barriers in the attendance of health care interventions by immigrants]. *Bundesgesundheitsblatt, Gesundheitsforschung, Gesundheitsschutz*, 55(8):944–953, 2012.

[5] D. Bohus and A. I. Rudnicky. Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda. 2003.

[6] H. Bunt. The DIT++ taxonomy for functional dialogue markup. In *AAMAS 2009 Workshop, Towards a Standard Markup Language for Embodied Dialogue Acts*, pages 13–24, 2009.

[7] H. Bunt, J. Alexandersson, J.-W. Choe, A. C. Fang, K. Hasida, V. Petukhova, A. Popescu-Belis, and D. R. Traum. Iso 24617-2: A semantically-based standard for dialogue annotation. In *LREC*, pages 430–437, 2012.

[8] M. G. Core and J. Allen. Coding dialogs with the damsl annotation scheme. In *AAAI fall symposium on communicative action in humans and machines*, pages 28–35. Boston, MA, 1997.

[9] D. Damljanovic, M. Agatonovic, and H. Cunningham. FREyA: An interactive way of querying Linked Data using natural language. In *The Semantic Web: ESWC 2011 Workshops*, pages 125–138. Springer, 2011.

[10] A. Hallili. Toward an Ontology-Based Chatbot Endowed with Natural Language Processing and Generation. 26th European Summer School in Logic, Language & Information, Aug. 2014. Poster.

[11] E. Kaufmann, A. Bernstein, and L. Fischer. NLP-Reduce: A "naive" but Domain-independent Natural Language Interface for Querying Ontologies. ESWC Zurich, 2007.

[12] V. Lopez, M. Fernández, E. Motta, and N. Stieler. Poweraqua: Supporting users in querying and exploring the semantic web. *Semant. web*, 3(3):249–265, Aug. 2012.

[13] V. Lopez, V. Uren, M. Sabou, and E. Motta. Is question answering fit for the semantic web?: A survey. *Semant. web*, 2(2):125–155, Apr. 2011.

[14] D. Mouromtsev, L. Kovriguina, Y. Emelyanov, D. Pavlov, and A. Shipilo. From spoken language to ontology-driven dialogue management. In *TSD*, pages 542–550, 2015.

[15] F. Nothdurft, F. Richter, and W. Minker. Probabilistic human-computer trust handling. In *Proc. of the Annual Meeting of the SIGDIAL*, pages 51–59, 2014.

[16] A. G. Nuzzolese, A. Gangemi, and V. Presutti. Gathering lexical linked data and knowledge patterns from framenet. In *6th International Conference on Knowledge Capture (K-CAP 2011)*, pages 41–48, 2011.

[17] L. Pragst, S. Ultes, M. Kraus, and W. Minker. Adaptive dialogue management in the KRISTINA project for multicultural health care applications. *SEMDIAL 2015 goDIAL*, page 202, 2015.

[18] V. Presutti, F. Draicchio, and A. Gangemi. Knowledge extraction based on discourse representation theory and linguistic frames. In *Knowledge Engineering and Knowledge Management, Ireland*, pages 114–129, 2012.

[19] S. Ultes and W. Minker. Managing adaptive spoken dialogue for intelligent environments. *Journal of Ambient Intelligence and Smart Environments*, 6(5):523–539, 2014.

[20] 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. In *International conference on Management of data*, pages 313–324, 2014.