An Open-Ended Web Knowledge Retrieval Framework for the Household Domain with Explanation and Learning through Argumentation

Alexandros Vassiliades

Aristotle University of Thessaloniki, School of Informatics, Thessaloniki, 54124, Central Macedonia, Greece

Foundation for Research and Technology, Institute of Computer Science, Heraklion, 70013, Crete, Greece

Nick Bassiliades

Aristotle University of Thessaloniki, School of Informatics, Thessaloniki, 54124, Central Macedonia, Greece

Theodore Patkos

Foundation for Research and Technology, Institute of Computer Science, Heraklion, 70013, Crete, Greece

Dimitris Vrakas

Aristotle University of Thessaloniki, School of Informatics, Thessaloniki, 54124, Central Macedonia, Greece

Abstract

Semantic Web knowledge bases can provide explainability and knowledge availability to the knowledge representation of any cognitive robotic system. For this reason, this paper presents a knowledge retrieval framework for the household domain enhanced with external knowledge sources that can argue over the information that it returns, and even learn new knowledge through an argumentation dialogue. The framework provides access to commonsense knowledge about sequences of actions on how to perform human-scaled tasks in a household environment, answers queries about household objects, and performs semantic matching between entities from the web knowledge graph ConceptNet, using semantic knowledge from DBpedia and WordNet, with the ones existing in the knowledge graph of the framework. The paper offers a set of predefined SPARQL templates that directly address the ontology on which the knowledge retrieval framework is built and querying capabilities through SPARQL. The framework also features an argumentation component, where the user can argue against the answers of the knowledge retrieval component of the framework under two different scenarios; the missing knowledge scenario, where an entity should be in the answers of the framework, according to the user, and the wrong knowledge scenario where an entity should not be in the answers of the framework. This argumentation dialogue can end up in learning a new piece of knowledge when the user wins the dialogue. The framework was evaluated via two different user evaluations and compared to baseline methods that use the native retrieval APIs of the external knowledge sources.

Keywords: Knowledge Retrieval, Argumentation, Learning, Household Domain, Explainability

INTRODUCTION

The use of Semantic Web Knowledge Bases (KBs) can provide both explainability and scalability to the knowledge representation component of cognitive robotic systems. In some cases, Semantic Web KBs, such as ConceptNet (Liu & Singh, 2004) and WordNet (Fellbaum, 2010), can provide commonsense reasoning skills (see Section 2). Moreover, it is important for a knowledge retrieval framework to be able to argue over and explain the answers it returns.

Ideally, a knowledge retrieval framework should answer as many queries as possible, regardless of the complexity of the involved entities of the query; in practice, high levels of completeness cannot easily be achieved. This is due to the fact that the quality of knowledge stored in any KB that is selected for knowledge retrieval purposes is limited for various reasons, thus, affecting the answers that it can return. A possible solution would be to use external knowledge sources and, specifically, Semantic Web KBs, which can provide commonsense knowledge (Zamazal, 2020).

In addition, it is desirable for the knowledge retrieval framework to justify the answers that it returns in response to a query. One common method to justify an opinion is through argumentation (Vassiliades et al., 2021a). The use of argumentation to support the validity of the answers returned by a knowledge retrieval framework can help the framework provide a human-friendly way of explanation. Moreover, argumentation in many cases can act as a method of learning new knowledge, if one's opinion is proved wrong through an argumentation dialogue. Thus, argumentation can also assist a knowledge retrieval framework to learn new knowledge that can be used in the future.

Considering the aforementioned observations, the problem that this paper aims to address is, given the contextual information relevant to a household environment, to construct a knowledge retrieval framework for the household domain, enhanced with external knowledge sources, that can argue over the answers that it returns and learn new knowledge, by means of an argumentation process[1]. The research questions that emerge are the following: a) *"How can a domain-specific knowledge retrieval framework be extended, so as to retrieve knowledge that exists in several knowledge bases?"*, b) *"How can a knowledge retrieval framework support its answers in a way closer to the human way of reasoning?"*, c) *"How can a knowledge retrieval framework learn new knowledge with methods closer to human learning?"*, and d) *"How can a knowledge retrieval framework be evaluated when there is no clear pipeline for the evaluation?"*. Question (a) is what this study tries to tackle with the knowledge retrieval component, questions (b), (c) are addressed with the learning-through-argumentation component, and question (d) is addressed in the evaluation section, where flexible methods on how to evaluate a framework with user evaluations are presented.

The framework provides knowledge about sequences of actions on how to perform human tasks in a household environment, answers queries about household objects, and performs semantic matching between entities originating from the web knowledge graph ConceptNet with the ones that exist in the internal knowledge graph, using knowledge from DBpedia (Bizer et al., 2009) and WordNet (see Figure 1). The framework offers a set of predefined SPARQL templates that directly address the ontology of the internal KB, as well as an API for general-purpose querying through SPARQL. The knowledge of the internal KB was extracted from the VirtualHome dataset (Liao et al., 2019; Puig et al., 2018). The framework also features an argumentation component, where the user can argue against the answers of the knowledge retrieval component of the framework under two different scenarios; the *missing* knowledge scenario, where an entity can be found in the answers of the framework, according to the user, and the *wrong* knowledge scenario, where an entity does not exist in the answers of the framework. Finally, the framework can learn new knowledge through argumentation, if the argumentation dialogue ends in favor of the human user (see Figure 4).

The framework can be used by cognitive robotic systems that act in a household environment, in order to provide information and instructions about the environment and the objects in it[2]. The most common queries that a user can issue to a robotic system were selected through an extensive literature overview on the topic of household cognitive robotics (Gouidis et al., 2019). For instance, a human user (target users can be elderly people in the early stages of dementia) can issue queries, such as *"What can I do with object X?"*, *"Which other objects are related to object X?"*, or *"Can I perform the activity Y with the objects X and Z?"*, among others, in order to assist their everyday tasks in a household environment. For this group of questions, a framework was developed for the user to handpick a query template and just give a keyword, to complete the template of a full query and get an answer. But the framework is not restricted only to this group of queries, as the user can issue their own SPARQL query to the system.

The framework was evaluated following two different user evaluation methods. The first one concerned the knowledge retrieval component of the framework, where 42 subjects were asked to express how satisfied they were with the answers returned by the framework over different query types. The results seem promising, with an 82% satisfaction score. Next, a gold standard dataset for a set of queries that the framework can answer was gathered from a group of 5 persons, not part of the first group. Subsequently, a group of 34 people was asked to answer the same queries selecting only from the answers found in each dataset and comparing them to the answers of the knowledge retrieval component of the framework. The second evaluation concerned the learning-through-argumentation component of the framework. Again, a gold standard dataset was created from 5 individuals for the *missing* knowledge and the *wrong* knowledge scenarios. Then, another group of individuals was asked to evaluate the quality of the argumentation process for the gold standard datasets. The results for the second evaluation also seem promising, as the missing knowledge scenario got a 79% satisfaction score, and the wrong knowledge scenario got an 80% satisfaction score. Moreover, the Semantic Matching Algorithm was compared to baseline methods that use the native knowledge retrieval APIs, of the external knowledge sources.

Next, two indicative scenarios of use of the framework are described.

Scenario A: Consider a household environment inhabited by an elder person at an early stage of dementia, who interacts with a robotic platform acting as an assistant that uses the framework presented in this paper for knowledge retrieval. In the case that the person has difficulty remembering how exactly an activity can be performed (i.e., does not remember the objects that are involved), the framework can recommend how to perform the activity and what objects might be needed. For instance, if the person asks for help in how to prepare a sandwich, the framework can return a sequence of steps to achieve this task and the objects involved (e.g., knife, bread, among others). The argumentation component can help the person when the outcome of the activity is not the desired one, by learning new methods on how to perform the activity and proposing similar utensils, if those proposed in the first place were not good enough to perform the activity or if the person cannot locate them.

Scenario B: Consider a computer vision mechanism that infers missing objects after it has perceived a set of objects. The framework can offer recommendations as to what the missing object(s) might be, by utilizing its ability to retrieve semantically similar objects, as well as by examining how objects are related to each other through an activity or an action. The argumentation component, in this case, can help to restrict or extend the recommendations returned by the framework, with the help of a human supervisor that will indicate, if some of the recommendations should not be part of the returned answers or indicate whether some other objects are missing from the returned answer.

The main contributions of the paper can be classified into theoretical and empirical ones. Theoretical contributions include a) a multi-KB knowledge retrieval methodology for the domain of household objects, b) a Semantic-Matching Algorithm that finds semantic similarity between entities of an internal KB with entities from external KBs, namely the knowledge graph of ConceptNet, using semantic knowledge from DBpedia and WordNet, and c) a supervised method of learning-through-argumentation based on commonsense reasoning, rather than a data-driven model.

Empirical contributions include the development of one of the largest knowledge bases about object affordances, namely actions that an object allows to be performed on/with it (Fischer et al., 2018; Pinacho et al., 2018; Tenorth & Beetz, 2017) (e.g. cut can be performed by knife). Furthermore, the evaluation of the knowledge retrieval component of the framework highlights three points: a) the large satisfaction of the users with the answers of the knowledge retrieval component (82%) signifies that the framework can be used by any cognitive robotic system acting in a household environment as a primary (or secondary) source of knowledge, b) the second method of evaluation implies that the knowledge retrieval component could be used as a baseline for evaluating other cognitive robotic systems acting in a household environment, and c) the scores that the Semantic Matching Algorithm achieved through the evaluation of the component, comparing it with baseline methods that use the native retrieval APIs of the external knowledge sources, shows that it can be used as an individual service for matching entities semantically. Finally, the evaluation of the learning-through-argumentation component of the framework indicates that the component achieves its purpose to a large extent, as both argumentation scenarios (missing and wrong answer) got a satisfaction score close to 80%. Therefore, users find the argumentation dialogue rational and convincing.

The novelty of this paper work, lies in the combination of these two cognitive processes (i.e., information retrieval and reasoning with argumentation for the information retrieved), which, to the best of our knowledge, has not been given much attention in the literature. Even though it is important to have a knowledge retrieval framework that can argue over the information that it returns, because it can convince the user about its answers with methods closer to the human way of thinking.

This paper is an extended version of (Vassiliades et al., 2020). The study was extended with: a) a new component that the framework can use in order to learn new knowledge, through an argumentation dialogue, b) a user evaluation for the new component, c) a comparison of the Semantic Matching Algorithm to baseline methods that use the native knowledge retrieval APIs of the external knowledge sources, and d) a web User Interface (UI).

The remainder of the paper is as follows. The next Section discusses the Related Work. In the Knowledge Retrieval Section, the knowledge retrieval component of the framework is described, as well as the Semantic Matching Algorithm which extends the knowledge of the framework using external knowledge sources. In the Argumentation and Learning Section, the learning-through-argumentation component is present, through which the framework can: (a) create an argumentation dialogue with a human user over the validity of its answers, and (b) learn new knowledge through argumentation. Next, the Evaluation is presented, with the user evaluation results of the framework components, the comparison with the baselines knowledge retrieval methods, and a discussion for the results. The final Section concludes the paper and discusses future work.

RELATED WORK

The study has two main components, the knowledge retrieval component, which uses external knowledge from ConceptNet (Liu & Singh, 2004), WordNet (Fellbaum, 2010), and DBpedia (Bizer et al., 2009), and the learning-through-argumentation component. To the best of our knowledge, a similar study that combines these aspects, in the context of household cognitive robotics, does not exist. For this reason, related work is presented separately for each component.

Knowledge Retrieval: The knowledge retrieval component was constructed to be fused into any cognitive robotic system acting in a household environment, but it can be also used by any intelligent system that might need the aspects that the component offers. A cognitive robotic system using the framework can enhance its knowledge about object properties, which objects are related to a certain object, affordances understanding, and semantically connect entities of its internal KB with entities from ConceptNet, using knowledge from DBpedia and WordNet. Moreover, the KB of the framework can be compared to other Linked Open Data KBs about products, and household objects.

Object identification methods have been implemented in many robotic platforms (Fischer et al., 2018; Pinacho et al., 2018; Wiedemeyer et al., 2015). Usually, object identification is based on the shape and the dimensions perceived by the vision module, or in some cases (Beetz et al., 2018; Tenorth & Beetz, 2017) reasoning frameworks such as grasping area segmentation, or a physics-based module contribute to understanding an object's label. In (Ruiz-Sarmiento et al., 2015), spatial-contextual knowledge is used to infer the label of an object; for example, the object *x* is usually found near objects $y_1, \ldots, y_n$, or *x* is found on *y*. Even though these are state-of-the-art frameworks, the robotic platform still must match knowledge from two or more different ontologies, to understand the label of an object.

Understanding affordances based on a (mainly OWL) ontology, is widely studied. Affordances are the set of real-life actions that a real-life object allows you to perform on/with it. For instance, the object *knife* allows you to *clean* it, and you can *cut* with it. In (Lemaignan et al., 2017; Ramirez-Amaro et al., 2017), the authors try to understand affordances by observing human motion. They capture the semantics of a human movement and correlate it with an action label. On the other hand, Jäger et. Al. (Jäger 2018) has connected objects with physical and functional properties, but the functional properties which can be considered affordances, capture very abstract properties, such as *containment, support, movability*, and *blockage*. Similarly, Beßler et. Al. (Beßler et al., 2018) define 18 actions that can be performed on objects if some preconditions hold, such as the reachability of the object, or the material of the object, among others. The affordances existing in the framework's KB are more than 70, combined with other features. Thus, the framework presented in this paper can offer greater plurality from the aforementioned works.

This paper attempts to fill the gap found in the previous studies. The knowledge retrieval component of the framework compared to the previous ones can offer the following: a) a rich KB of household objects related to actions, b) a rich KB with sequences of actions to achieve human scaled tasks, and c) a semantic match-making framework between an entity of the internal KB and entities found at external knowledge sources.

The Semantic Matching Algorithm was mostly inspired by the works of Young et. Al. (Young et al., 2016), and Icarte et. Al. (Icarte et al., 2017), where the authors use commonsense knowledge from the web ontologies DBpedia, ConceptNet, and WordNet to find the label of unknown objects. Furthermore, inspiration was given from the studies (Chernova et al., 2020; Young et al., 2017), where the label of the room can be understood through the objects that the cognitive robotic system perceived from its vision module. One drawback that can be noticed in these works, is that all of them depend on a single ontology. Young et. Al. compares only the DBpedia comment boxes between the entities, Icarte et. Al. acquires only the property values of the entities from ConceptNet, and (Chernova et al., 2020; Young et al., 2017) work on the synonyms, hypernyms, and hyponyms of WordNet entities.

The ontology the framework uses can be compared with existing product ontologies, such as the ones found in (Radinger et al., 2013; Wagner & Rüppel, 2019), the more recent (Sanfilippo, 2018), and the general-purpose ontology GoodRelations (Hepp, 2008). The difference is that these ontologies offer information about objects, geometrical, physical, and material properties, and create object taxonomies with hierarchical relations. Instead, this paper is representing knowledge about objects and their affordances. Other similar ontologies are (i) O-Pro (Bhattacharyya et al., 2016) which is an ontology for object-affordance relations but is considerably smaller with respect to the number of objects and affordances, and (ii) ATOMIC (Sap et al., 2019) which can be considered in the same category, containing generic causal relations. More specifically, ATOMIC is a knowledge graph with *if-then* relations (i.e., if the event X happens then there is some result Y). Even though ATOMIC has some object-action relations, its variety is much lesser than the framework presented in this paper. Also, these relations are embedded in small textual descriptions, for instance, *"Person cut the cucumber with a knife (X) → the cucumber is in pieces (Y)"*, thus some text processing is required to extract the actual object-action relation, e.g., *knife-cut*. On the other hand, the KB presented in this paper represents object-action relations as RDF triples, i.e., using semantic links between entities in a knowledge graph (aka semantic network). Thus, to the best of our knowledge, this paper offers the richest KB about object affordances, in a household environment.

Learning through Argumentation: The idea of learning through an argumentation dialogue is an innovative idea which has not been given much attention in the field of machine learning. It is a very interesting method to allow an individual (i.e., machine or human) to learn new knowledge, because it resembles a commonsense reasoning procedure which is closer to the human way of thinking. This method is applied in schools (Berland & McNeill, 2010; Von Aufschnaiter et al., 2008; Pratiwi et al., 2019; Akhdinirwanto et al., 2020; Hu et al., 2022; Fiorini, 2020), to allow students to argue and learn through argumentation by accepting other students' arguments, if they cannot defend their position. In most cases the need of a supervisor, a teacher in most cases, which will check the rationality of arguments is mandatory. This method of learning is also found in collaborative learning (Veerman, 2000), between a human and a machine at a theoretical level. Nevertheless, some data driven models that can be trained over datasets of argumentation dialogues for legal cases have been presented in (Mŏzina et al., 2005, 2007). The difference is that our framework does not need training in order to perform argumentation dialogues, and it can learn through the dialogue.

Collaborative learning through argumentation in multi-agent systems has been proposed in (Ontañón & Plaza, 2010; Ontañón & Plaza, 2007). The authors of these papers present some protocols upon which a group of agents can start learning from each other to improve the individual and joint performance for decision making. Moreover, the authors state that *"argumentation is a useful framework for joint deliberation and can improve over other typical methods such as voting"*. The difference is that these are theoretical studies which propose protocols that agents should follow, to enhance learning-through-argumentation. Also, they solve conflicts through preference rules which needs tuning based on the context of the conversation. Instead, the framework presented in this paper uses commonsense knowledge from human individuals to learn new knowledge. Learning-through-argumentation is also presented in (Chen et al., 2019; Drapeau et al., 2016), and (Clark et al., 2007; Lin et al., 2020; Slonim et al., 2021). In (Chen et al., 2019; Drapeau et al., 2016), the authors present a framework that human individuals can learn from each other, and in (Clark et al., 2007; Lin et al., 2020; Slonim et al., 2021) external knowledge from the Web is used for humans to learn. Instead, this paper focuses on agent learning.

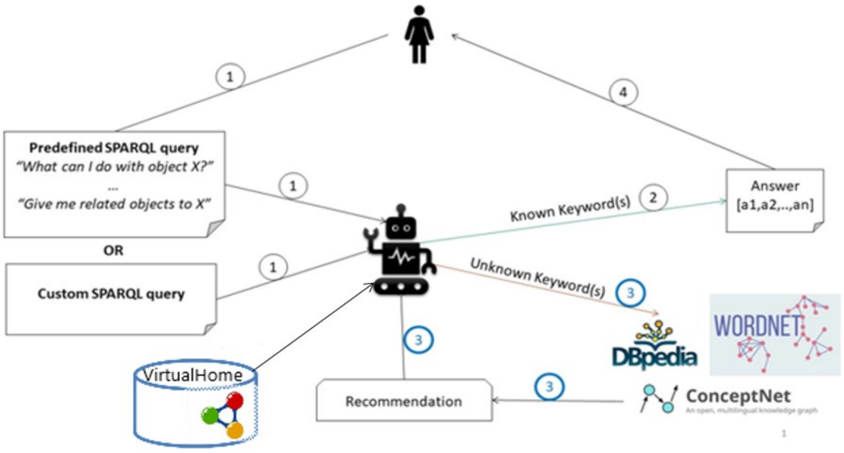
Another way to view this part of the framework, is that a method of knowledge refinement through argumentation is introduced, that supervises the knowledge that enters in the KB using the commonsense of the user. One can notice that belief revision is a method of learning (Kelly, 1998); an agent removes or adds knowledge to its KB to learn. Therefore, argumentation can be considered as a method of belief revision (Vassiliades et al., 2021a), because argumentation can convince the opposing participant(s) to refine the knowledge in their KB. Belief revision with argumentation has been used for various tasks such as to enhance the knowledge in a KB (Rahwan et al., 2004; Falappa et al., 2009; Cayrol et al., 2008), to explain why an argument is true or false (Coste-Marquis et al., 2014b; Fan & Toni, 2015; Coste-Marquis et al., 2014a), and for negotiation (Pilotti et al., 2015; Okuno & Takahashi, 2009; Pilotti et al., 2014). But to the best of our knowledge, studies about knowledge refinement through an argumentation procedure, remain at a theoretical level.

Learning through case-based argumentation (Cyras et al., 2016; Heras et al., 2013; Aleven & Ashley, 1997; Ashley et al., 2002; Fan et al., 2013), is a method for learning how to classify an argument. In case-based argumentation the argument is classified as acceptable, or non-acceptable based on similar examples of argument classification. Therefore, arguments are classified using similarity metrics extracted from a knowledge graph, or based on feature importance, or employing embeddings, among others. Instead, in this work the user's commonsense is employed to classify an argument as acceptable, or not.

KNOWLEDGE RETRIEVAL

In this section, the architecture and the different aspects of the knowledge retrieval component are described in detail. In the first subsection, the dataset, from which knowledge was extracted and fused in the schema, is presented. Next, the ontology that this component is built on is introduced. In the last subsection, the algorithm that semantically matches entities from ConceptNet, using semantic similarity from DBpedia and WordNet, with entities in the KB of the framework is analyzed. The workflow of the knowledge retrieval component can be seen in Figure 1. Each step in the workflow is annotated with a number in a circle that indicates the order in the workflow path. Blue colored circles indicate optional steps. Notice that all parts of this component were developed by the authors of this paper, except for the VirtualHome dataset, which was adapted from the VirtualHome project (Liao et al., 2019; Puig et al., 2018) and was transformed into an ontology and KB, used by the framework internally.

Figure 1. Architecture of the Knowledge Retrieval Component



Dataset

The VirtualHome dataset (Liao et al., 2019; Puig et al., 2018) contains activities that people perform at home. For each activity, different descriptions are given on how to perform them. The descriptions are present in the form of sequence of actions, i.e., steps that contain an action related with an object(s), illustrated in Example 1. Moreover, the dataset offers a virtual environment representation for each sequence of actions with Unity[3]. The dataset contains ∼2800 sequences of actions, for human scaled activities. Moreover, the dataset holds more than 500 objects, usually found in a household environment, which are semantically connected with each other, and with specific human scaled actions.

Example 1. *Browse Internet*

Comment: walk to living room. look at computer. switch on computer. sit in chair. watch computer. switch off computer.

[Walk] ⟨living_room⟩ (1)
[Walk] ⟨computer⟩ (1)
[Find] ⟨computer⟩ (1)
[TurnTo] ⟨computer⟩ (1)
[LookAt] ⟨computer⟩ (1)
[SwitchOn] ⟨computer⟩ (1)
[Find] ⟨chair⟩ (2)
[Sit] ⟨chair⟩ (2)
[Watch] ⟨computer⟩ (1)
[SwitchOff] ⟨computer⟩ (1)

Each sequence of actions has a template: (a) Activity Label, (b) Comment, i.e., small description, and (c) the sequence of actions. Each step has the general form shown below:

$$[Action]\ \langle Object_1\rangle(ID_1)\ \ldots\ \langle Object_n\rangle(ID_n)$$

where *Action* is the human scaled action, $Object_1, \ldots, Object_n$ are the objects on which the action is performed ($n \in N^*$), and $ID_1, \ldots, ID_n$ are the identity numbers of each object. The identity numbers are different for objects that refer to the *same* physical entity, for example ⟨*chair*⟩ (1) and ⟨*chair*⟩ (2); otherwise, the identity numbers can be identical, for instance ⟨*living_room*⟩ (1) and ⟨*computer*⟩ (1). In the experiments there are approximately 500 objects, but since the ontology can be freely extended with objects, *n* is considered as a natural number.

Ontology

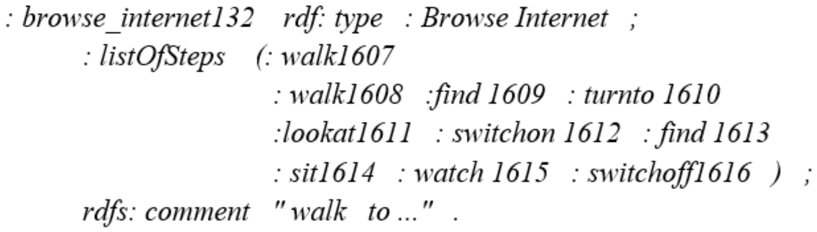
The main component of the knowledge retrieval framework is the ontology that was inspired by the VirtualHome dataset. Figure 2 presents part of the ontology concepts, while Figure 3 shows the relationships between the major concepts. The construction of the ontology was seamless as the VirtualHome dataset provided hierarchical relations, and only the relations between the classes were devised. The relations are described in detail in this subsection. Nevertheless, some ideas from ontology construction methods (Subhashini & Akilandeswari, 2011) have been used.

The class *Activity* contains some subclasses which follow the hierarchy provided by the dataset; these were hand coded. Moreover, the instances of these classes are the sequence of actions existing in the VirtualHome dataset. The class *Activity* is connected through the property *listOfSteps* with the class *Step*. Additionally, the class *Step* is connected through the properties *object* and *step_type* with the classes *ObjectType* and *StepType*, respectively. Next, the class *ObjectType* contains the labels of all the objects found in the sequences. On the other hand, the class *StepType* is similar to *ObjectType*, as it gives natural language labels to the steps.

Every sequence of actions was represented as a list because this gave stronger coherency and interaction on the knowledge provided by the activity. Thus, the framework can answer queries like

“What is the third step in the sequence of activity X?”, or *“Return all the sequences where firstly I walk to the living room, then I open the TV, and after that I sit on the sofa”*. This kind of information can prove crucial for a system with planning capabilities. Also, an instance generator algorithm that transforms the sequences of actions from the form shown in Example 1 into instances in the ontology was developed. The class that the sequence belongs to is provided by the Activity label. Such an instance is given in Example 2.

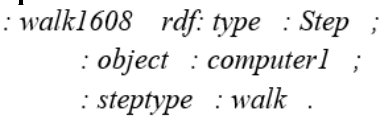
Example 2.



```
: browse_internet132   rdf: type   : Browse Internet   ;
          : listOfSteps   (: walk1607
                               : walk1608   :find 1609   : turnto 1610
                               :lookat1611   : switchon 1612   : find 1613
                               : sit1614   : watch 1615   : switchoff1616   )   ;
          rdfs: comment   " walk   to ..."   .
```

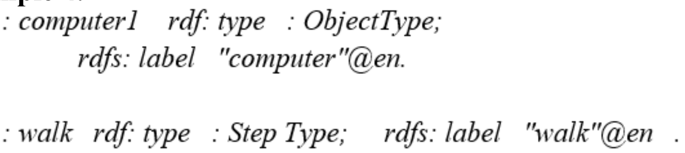
Each step shown in the property *listOfSteps* is an instance of the class *Step*. Each step has a unique ID that distinguishes it from all the other steps. Example 3 shows an instance step from the *listOfSteps*, and Example 4 shows the object and action from the *ObjectType* and *StepType* classes with which the instance, from Example 3, is connected.

Example 3.



```
: walk1608   rdf: type   : Step   ;
          : object   : computer1   ;
          : steptype   : walk   .
```

Example 4.



```
: computer1   rdf: type   : ObjectType;
          rdfs: label   "computer"@en.

: walk   rdf: type   : Step Type;     rdfs: label   "walk"@en   .
```

Figure 2. Part of the Ontology Scheme

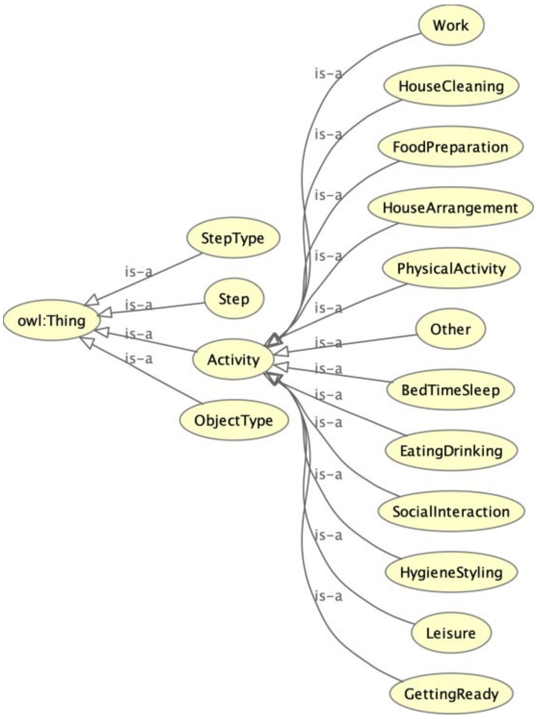
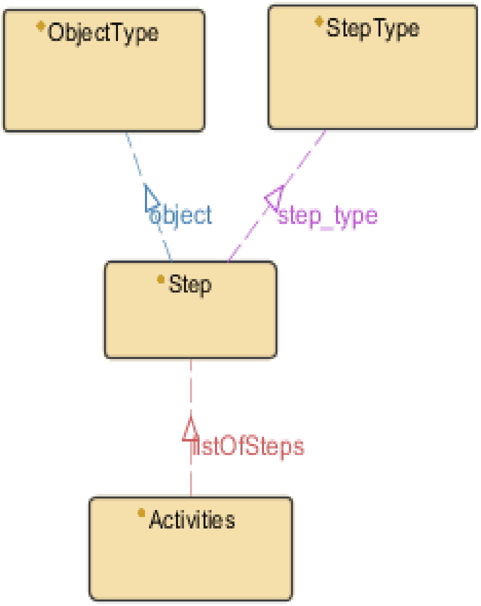


Figure 3. Ontology Properties



After constructing and populating the ontology, a library in Python that constructs SPARQL queries addressed to the ontology and fetches answers was developed. The library consists of 9 predefined query templates that represent the most probable question types to the household ontology. These templates were selected as the most important ones, following an extensive literature review of studies about cognitive robotic systems that act in a household environment (Gouidis et al., 2019). Among many other studies, primarily KnowRob (Beetz et al., 2018; Tenorth & Beetz, 2017), RoboSherlock (Beetz et al., 2015), RoboBrain (Saxena et al., 2014), the paper of Bai et al. (Bai et al., 2021), and RoboCSE (Daruna et al., 2019), were considered. These questions apply to cognitive robotic systems that have a physical body to perform tasks, and to cognitive robotic systems that work only as recommenders or problem solvers. The findings of the above review led to the construction of these query templates, as the most common and crucial queries addressed to a cognitive robotic system, acting in as household environment. Listing 1 shows the SPARQL template that returns the objects which are related to two other objects through an activity, *Object1* and *Object2*.

Listing 1: SPARQL query that returns all the objects that are related to *Object1* and *Object2* through an activity.

```
SELECT DISTINCT ?object WHERE {
          ?instance :listOfSteps ?list.
          ?list rdf:rest*/rdf:first ?element.
          ?element :object ?object
          SELECT DISTINCT ?instance WHERE {
                 ?intl rdfs:subClassOf* :Activity.
                 ?instance rdf:type ?intl;
                     :listOfSteps ?listl.
                 ?listl rdf:rest*/rdf:first ?stepl.
                 ?stepl  :object <Object1>.
                 ?listl rdf:rest*/rdf:first ?step2.
                 ?step2  :object <Object2>.}}
```

For instance, if the user provides the objects *coffee* and *coffee_cup* then the SPARQL query will return the answer:

```
      {milk, coffee pot, coffee filter, coffee
     table, kettle, spoon, stove, coffee maker,
  kitchen counter, sink, faucet, kitchen cabinet,
    ground coffer, table, button, cupboard, pot,
     living room, couch, water, chair, laptop,
          dining room, newspaper, sugar}
```

Alternatively, ad-hoc SPARQL queries can be asked to the ontology, such as Listing 2, where a user wants to see the objects involved in the activity *Activity1*.

Listing 2: SPARQL query that returns all the objects involved in *Activity1* .

```
SELECT DISTINCT ?object WHERE
     {<Activity1> :listOfSteps  ?list.
     ?list rdf:rest*/rdf:first ?step.
     ?step :object ?object}
```

Therefore, users can hand pick one of the predefined query templates and then give the keywords that are needed to fill the SPARQL template (Listing 1), to formulate a proper query and access the information they desire, or they can write their own SPARQL query to access the information they desire (Listing 2).

Semantic Similarity Algorithm

Since the dataset, upon which the knowledge retrieval framework was constructed, has a specific number of objects, in order to retrieve knowledge about objects on a larger scale, a mechanism that can take advantage of the web knowledge graphs DBpedia, ConceptNet, and WordNet to answer queries about objects that do not exist in the KB of framework was developed.

This would broaden the range of queries that the framework can answer, and would overcome the downside of the framework being dataset oriented. The user with the Semantic Matching Algorithm (SMA) can address queries with labels that do not exist in the internal KB of the framework. An aspect which could not be achieved without the SMA.

Algorithm 1 was implemented using Python. The libraries *Request* and *NLTK* [4] offer web APIs for all three aforementioned ontologies. Similar methods can be found in (Icarte et al., 2017; Young et al., 2016), where they also exploit the commonsense knowledge existing in web ontologies. Algorithm 1 starts by getting as input any word that is part of the English language; this is checked by obtaining the WordNet entity, line 1. Note that any entity that is part of WordNet is also part of ConceptNet (Liu & Singh, 2004). The input is given by the user indirectly, when (s)he gives a keyword in a query that does not exist in the KB of the framework.

Subsequently, the algorithm turns to ConceptNet, and collects the properties and values for the input word, line 2. In the framework, only the values of the properties *RelatedTo, UsedFor, AtLocation, CapableOf, Causes, ReceivesAction*, and *IsA* are collected. These properties were chosen because they are the most related to the target application of providing information for household objects[5]. Also, the weights that ConceptNet offers for each triplet are acquired. These weights represent how strong the connection is between two different entities with respect to a property in the ConceptNet graph, and are defined by the ConceptNet community. Therefore, a hash map of the following form is constructed:

$$\{\,Property_1\!:\left[\left(entity_1{}^1, weight_1{}^1\right), \dots\right.$$
$$\left.(entity_m{}^1, weight_m{}^1)\right], \dots,$$
$$Property_l\!:\left[\left(entity_1{}^l, weight_1{}^l\right), \dots,\right.$$
$$\left.(entity_n{}^l, weight_n{}^l)\right]\}$$

for $m, l, k \in N^*$.

Then, the semantic similarity between the given entity and the returned property values is extracted using WordNet and DBpedia, lines 3-6. Firstly, the algorithm finds the least common path that the given entity has with each returned value from ConceptNet, in WordNet, line 7. The knowledge in WordNet is in the form of a directed acyclic graph with hyponyms and hypernyms. Thus, in each case the number of steps that are needed to traverse the path from one entity to another is obtained. Subsequently, the algorithm turns to DBpedia to extract comment boxes of each entity using SPARQL, lines 9-11. If DBpedia does not return any results, the entity is searched in Wikipedia, which has a better search engine, and with the returned URL DBpedia is asked again for the comment box, based on the mapping scheme between Wikipedia URLs and DBpedia URIs, lines 12-18. Notice that when a redirection list is encountered the first URL of the list is acquired which in most cases is the desired entity and its comment box is retrieved. If the entity linking through Wikipedia does not work, then it is considered that the entity was not found, and an empty list is returned.

Algorithm 1 Semantic Matching Algorithm

Require: entity
Ensure: hash_semantic_similarity

```
1:  if entity in WordNet then
2:      hash_property_values=get.ConceptNet_property_values(entity)
3:      Comment_Box_Input = get.DBpediaCommentBox(entity)
4:      hash_semantic_similarity = {}
5:      for property in hash_property_values do
6:          for value in hash_property_values [property] do
7:              WordNet_Path = get.WordNet_Path(entity, value)
8:              Comment_Box = ∅
9:              if value in DBpedia then
10:                 Comment_Box = get.DBpediaCommentBox(value)
11:             end if
12:             if Comment_Box = ∅ then
13:                 wiki_entity = get.WikipediaEntityURL(value)
14:                 Comment_Box = get.DBpediaCommentBox(wiki_entity)
15:             end if
16:             if Comment_Box = ∅ then
17:                 continue
18:             end if
19:             TFIDF = TF-IDF(Comment_Box_Input, Comment_Box)
20:             hash_semantic_similarity[property] = (value, Sim(entity, value))
21:         end for
22:     end for

23:     hash_semantic_similarity = sorted(hash_semantic_similarity)
24: end if
```

The comment box of the input entity is compared with each comment box of the returned entities from ConceptNet, using the TF-IDF algorithm to extract semantic similarity, line 19. The rationale here is that the descriptions of two entities which are semantically related will contain common words. Therefore, the cosine similarity of the vectors that represent the two descriptions (after the TF-IDF algorithm) will be larger than the cosine similarity of two vectors that represent the description of two entities which are not related. TF-IDF was preferred in order not to raise the complexity of the framework using pre-trained embedding vectors like Glove (Pennington et al., 2014), Word2Vec (Rong, 2014), or FastText (Joulin et al., 2016). In this case, TF-IDF was used with stemming over the texts; stemming was applied when the texts were pre-processed, and thus words can be related which are not exactly the same.

TF-IDF can reduce the time complexity of Algorithm 1, as the complexity of TF-IDF is $O\ (n * log(n))$, where n is the number of words in both texts. On the other hand, word embeddings in Algorithm 1 could rise the time complexity at levels of class $O\ (n * m)$, where n is the number of words in the text, and m the number of vectors in the dataset of word embeddings. Looking at the complexities of TF-IDF and word embeddings one can see that TF-IDF is much quicker. More specifically, for the system to produce an answer using word embeddings, it needed approximately more than 3 minutes for its computation (Vassiliades et al., 2021b), which could not support an online question-answering system.

In order to define the semantic similarity between the entities, a new metric that is based on the combination of WordNet paths, TF-IDF scores, and ConceptNet weights was devised (Equation (1)). This metric takes into consideration the smallest WordNet path, the ConceptNet weights, and the TF-IDF scores. TF-IDF and ConceptNet scores have a positive contribution to the semantic similarity of two words. On the other hand, the bigger the path is between two words in WordNet the smaller the semantic similarity is.

$$Sim(i, v_j) = \frac{\frac{1}{WNP(i, v_j)} + TFIDF(i, v_j) + +CNW(i, p, v_j)}{3} \qquad (1)$$

In (1), i is the entity given as input by the user, and v_j, for $j \in \mathrm{N}^*$, is each one of the different values returned from ConceptNet properties. $CNW(i, p, v_j)$ is the weight that ConceptNet gives for the triplet (i, p, v_j), p stands for the property that connects i and v_j, and $0 \leq CNW\ (\cdot, \cdot, \cdot) \leq 1$. $TFIDF\ (i, v_j)$ is the score returned by the TF-IDF algorithm when comparing the DBpedia comment boxes of i and v_j, and $0 \leq TFIDF\ (\cdot, \cdot) \leq 1$. $WNP\ (i, v_j)$ is a 2-parameter function that returns the least common path between i and v_j, in the WordNet directed acyclic graph.

In case i and v_j have at least one common hypernym (ch), then the smallest path is acquired for the two words, whereas in case i and v_j, do not have a common hypernym (nch), their depths are added. Let, $depth(\cdot)$ be the function that returns the number of steps needed to reach from the root of WordNet to a given entity, then:

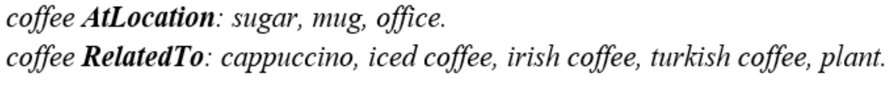
$$WNP(i, v_j) = \begin{cases} min_{c \in C}\{depth(i) + depth(v_j) - 2 * depth(c)\}\ ch \\ \quad depth(i)\ +\ depth(v_j)\ nch \end{cases} \qquad (2)$$

where C is the set of common hypernyms for i and v_j, c is a common hypernym of i and v_j, and $min_{c \in C}\{. . .\}$ returns the minimal value of the equation $depth(i) + depth(v_j) - 2 * depth(c)$ for $c \in C$. Also, $0 < WNP\ (\cdot, \cdot) \leq 1$.

Equation 1, can take scores in the range of (0,1], and the weights for $CNW\ (\cdot, \cdot, \cdot)$, $TFIDF\ (\cdot, \cdot)$, and $WNP\ (\cdot, \cdot)$ are equal to 1. Other variations of weights are possible, but they do not capture the synergy between the various metrics of the semantic relatedness of many pairs of entities that were examined. Moreover, the information from ConceptNet, WordNet, and DBpedia is used without any bias among them. Table 1 shows the values that Equation 1 returns for the pairs of Example 5. The results are rounded to three decimals. The last step of the algorithm sorts the semantic similarity results of the returned entities with respect to the ConceptNet property, and stores the new information into a hash map, line 23.

An example of the returned information is given in Example 5 where the Top-5 entities for each property are displayed if there exist as many.

Example 5.



coffee ***IsA****: stimulant, beverage, acquired taste, liquid.*
coffee ***AtLocation****: sugar, mug, office.*
coffee ***RelatedTo****: cappuccino, iced coffee, irish coffee, turkish coffee, plant.*

The evaluation of the Algorithm 1 was performed through a user evaluation since the question *Q4* from the *Knowledge Retrieval Evaluation* sub-section had exactly this purpose. The users were asked to evaluate the answers that the framework returned from Q4 which exclusively used the Semantic Matching Algorithm, therefore indicating if the quality of the semantic matching that the algorithm achieves is satisfactory or not. Moreover, the accuracy of the Semantic Matching Algorithm is compared with the accuracy of other baseline methods, in the *Comparison with Baselines Methods* sub-Section.

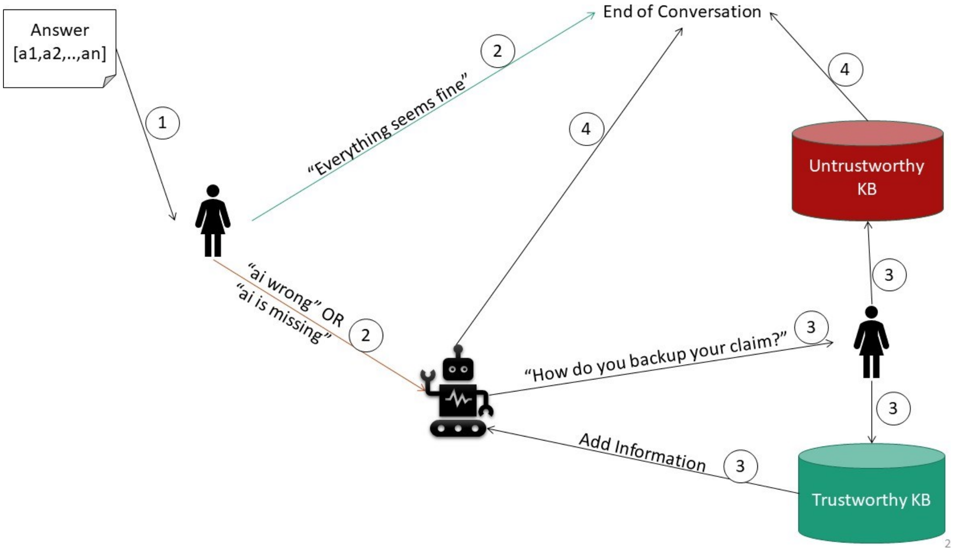
Table 1. Results for Equation 1

Pair	**Relation**	***CNW*** **(·, ·, ·)**	***TFIDF*** **(·, ·)**	***WNP (·,·)***	***Sim*****(·, ·)**
coffee-stimulant	IsA	0.529	0.324	0.256	0.369
coffee-beverage	IsA	0.2	0.291	0.221	0.237
coffee-acquired taste	IsA	0.2	0.216	0.1	0.172
coffee-liquid	IsA	0.1	0.198	0.127	0.141
coffee-sugar	AtLocation	0.283	0.371	0.321	0.325
coffee-mug	AtLocation	0.1	0.256	0.293	0.216
coffee-office	AtLocation	0.2	0.189	0.1	0.163
coffee-cappuccino	RelatedTo	0.1	0.398	0.43	0.309
coffee-iced coffee	RelatedTo	0.1	0.337	0.387	0.274
coffee-irish coffee	RelatedTo	0.1	0.284	0.231	0.205
coffee-turkish coffee	RelatedTo	0.1	0.283	0.172	0.185

ARGUMENTATION AND LEARNING

Learning-through-argumentation is the second component of the framework and is built upon the knowledge retrieval component. The framework allows the user, after (s)he has received an answer to her/his question, to argue against the validity of the answers, if (s)he considers that there is an entity which is missing in the answers returned, or something is wrong and should not be part of the answers returned. The user must back up her/his opinion by indicating a trustworthy KB in which (s)he has found the information that (s)he supports. This component uses external knowledge from ConceptNet (Liu & Singh, 2004), and WordNet (Fellbaum, 2010). If the user wins the argumentation dialogue, the argumentation component of the framework accepts that the knowledge retrieval component has missed some entities, or it has returned something wrong in its answers, and it refines its KB to add the new information or delete existing knowledge. Notice that in both scenarios the user cannot change the trust score of any KB by hand. This policy allows the knowledge retrieval component to build its own trust for each KB. This idea is mostly because a human individual may trust a KB, for her/his own personal reason, but this does not mean that this KB can be trustworthy in general. The workflow of the argumentation and learning component can be seen in Figure 4. Notice that Figure 4 is basically the extension of Figure 1, and the answers the user receives at the first step of Figure 4 are the answers returned by the framework in step 4 of Figure 1. Also notice, that many details of Figure 1 are not shown in Figure 4, to focus mainly on the argumentation component. Each step in the workflow is annotated with a number in a circle that indicates the order in the argumentation dialogue. Finally notice, that there are alternative paths in the dialogue. But first some preliminaries need to be given which will be needed in the *Missing Knowledge Scenario* and *Wrong Knowledge Scenario* sub-sections.

Figure 4. Architecture of the Learning through Argumentation Component



Preliminaries

The argumentation component presented in this paper is based on an Abstract Argumentation Framework F = (A, R) (Dung 1995), where A is the set of arguments, and $R \subseteq A^2$ is a binary attack relation. The set of arguments A is composed by: (a) The answers that the framework gives to the user (B) (during the whole dialogue), and (b) The disputing statements that the user can make either in the missing scenario (A_1), or in the wrong scenario (A_2). Thus, if $C = A_1 \cup A_2$ then $A = B \cup C$. Moreover, the attack relation is defined as shown below:

$$R = \{(b, c), (c, b) \mid b \in B, c \in C\}$$

As for the attack relations no self-attacks are allowed, because otherwise the framework could attack its answers, and the user could attack her/his questions.

The facts in the KB that the framework searches for are represented using RDF triples. Moreover, each triple is associated with a trust score which is called *trust score of fact*. The trust score of fact helps computing the trust score of the entire KB. Equation 3, shows how the trust score of a KB is computed.

$$trustKB = \frac{trustFact_1 + \cdots + trustFact_n}{n} \qquad (3)$$

where *trustKB* is the trust score of the KB, and $trustFact_i$ for $i = 1,...,n$ are the trust scores of the facts. New facts in the KB can be inserted either through the missing knowledge scenario where the user indicates that an entity is missing from the answers returned by the framework, or through the wrong scenario but only when the framework needs to use the SMA to answer the question that the user addressed initially. Therefore, trust scores of new facts are given a default score of 50% each, and according to the argumentation dialogues which are described in this Section can increase or decrease.

Missing Knowledge Scenario

In this scenario the user indicates a *missing* entity that should exist in an answer of the knowledge retrieval component (see Knowledge Retrieval Section). The argumentation component then starts an argumentation dialogue with the user, in order to infer if the entity which the user claims to be missing can be found in a trustworthy KB, and if yes (i.e., if it finds the information in the KB by searching) it will accept her/his argument and refine the internal KB of the framework. In the dialogue protocol and the example below *S-* is used for a dialogue move performed by the framework and *U-* for a move performed by the user.

Dialogue protocol for the missing scenario.

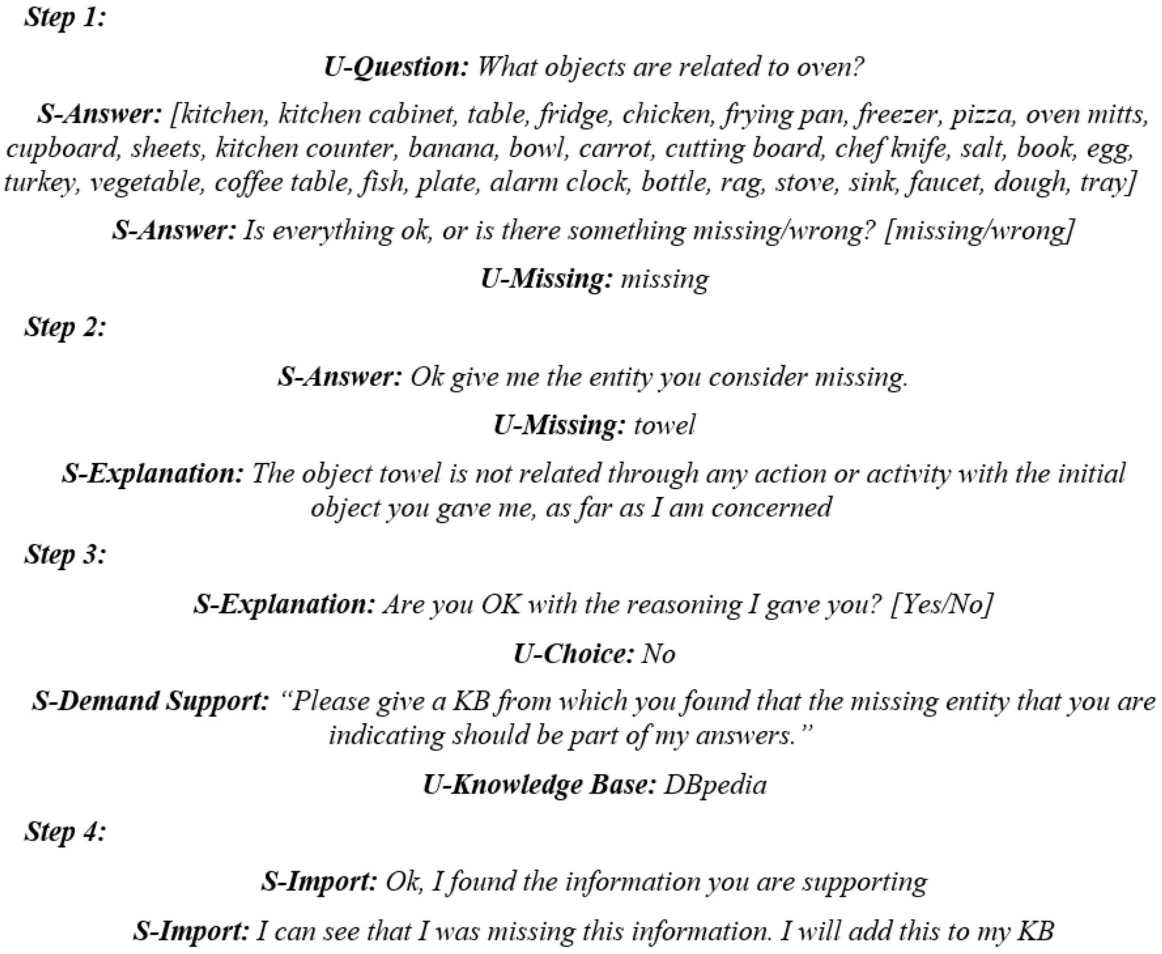
- **Step 1:** Given a question to the framework by the user, the framework will then ask if everything is fine or there is something missing or wrong in its answer. The user should then indicate which entity is missing.
- **Step 2:** The component will then give an explanation to the user why the entity does not belong to its answers, according to where the entity belongs. The text that the component returns at this point may vary according to the type of question the user initially performs.
  (1) If the missing entity that the user is indicating already exists in the list of answers, the component will give the explanation: *S - "I have this in the returned list of answers."*, and the argumentation dialogue will stop
  (2) If the missing entity that the user is indicating exists in the KB of the framework, the component will give the explanation: *S - "The object **X** is not related through any action or activity with the initial object you gave me, as far as I am concerned"*, and the argumentation dialogue will proceed to **Step 3**.
  (3) If the missing entity that the user is indicating does not exist in the KB of the framework, the component will try to correlate the entity with an entity from the KB using external knowledge. Firstly, the component will use the text distance metric *Ratcliff-Obershelp*[6] from NLTK with a very high threshold of 90% to allow correlation of entities that have a difference of one or two letters. If this method does not work, the component will use the SMA (see Section 3.3). If any correlation is found, regardless the method, the explanation will be the following: *S - "The object **X** does not exist in my KB, but I have related it with **Y**. Unfortunately, **Y** is not related through any action or activity with the initial object you gave me, as far as I am concerned"*, and the argumentation dialogue will proceed to **Step 3**.
  (4) If the missing entity that the user is indicating does not exist in the KB of the framework, and no recommendation is found with the help of external knowledge (i.e., (3)), the component will give the explanation: *S - "Sorry, I could not relate the object **X** with any entity in my KB"*, and the argumentation dialogue will proceed to **Step 3**.
- **Step 3:** The user can then challenge the explanation that the component returned. At this point the user must give a trustworthy KB in which (s)he found the information. Initially, the external KBs that the user can indicate have a trust score of 50%, except of those used in the SMA (see *Semantic Similarity Algorithm* sub-section) which have a 60% trust score (i.e., ConceptNet, WordNet, and DBpedia). Currently, only *WikiHow*[7] and *BabelNet*[8] were considered. To be consider an external KB trustworthy, it must have a trust score greater than or equal to 50%. Only then the framework will proceed to **Step 4**; otherwise, the framework will return the message *S - "Sorry, but I do not trust that KB"*. Since the component cannot extract knowledge graphs from natural text, the information that exists in the KB must be in RDF format.

  Notice that if the trust score of the fact is 0 then the framework considers that the *KB* which the user indicated does not contain the fact, and it stops the argumentation dialogue. The argumentation dialogue in this case ends with the text *"Sorry I could not find the information that you point."*
- **Step 4:** In this final step the component has found the information that the user is indicating in the KB, and it informs her/him with a message *S - "I can see that I was missing this information. I will add this to my KB"*, and it will create a triplet in its KB. Thus, if the user asks again the question from **Step 1** the missing entity will be part of the component's answers. Finally, the component will increase the trust score of the fact in the KB that the user indicated by 10%, and by extension the trust in the

external KB.

Example 6 demonstrates how the missing protocol works.

Example 6.



Step 1:

U-Question: *What objects are related to oven?*

S-Answer: *[kitchen, kitchen cabinet, table, fridge, chicken, frying pan, freezer, pizza, oven mitts, cupboard, sheets, kitchen counter, banana, bowl, carrot, cutting board, chef knife, salt, book, egg, turkey, vegetable, coffee table, fish, plate, alarm clock, bottle, rag, stove, sink, faucet, dough, tray]*

S-Answer: *Is everything ok, or is there something missing/wrong? [missing/wrong]*

U-Missing: *missing*

Step 2:

S-Answer: *Ok give me the entity you consider missing.*

U-Missing: *towel*

S-Explanation: *The object towel is not related through any action or activity with the initial object you gave me, as far as I am concerned*

Step 3:

S-Explanation: *Are you OK with the reasoning I gave you? [Yes/No]*

U-Choice: *No*

S-Demand Support: *"Please give a KB from which you found that the missing entity that you are indicating should be part of my answers."*

U-Knowledge Base: *DBpedia*

Step 4:

S-Import: *Ok, I found the information you are supporting*

S-Import: *I can see that I was missing this information. I will add this to my KB*

Wrong Knowledge Scenario

In this scenario the user indicates an entity that should not exist in an answer of the knowledge retrieval component (Section 3). The difference with the missing knowledge scenario is that this scenario has two smaller sub-cases, based on where the keywords that the user used in the initial question belong to. If, all the keywords belong to the internal KB of the framework, the component considers its internal knowledge beyond any possible doubt and does not proceed with an argumentation dialogue. Otherwise, if the framework needs to use the Semantic Matching Algorithm (Section 3.3) then it will start an argumentation dialogue with the user in order to infer, if the user's claim that an entity was wrongly included in an answer can be substantiated via an external trustworthy KB, and if yes it will accept her/his argument and delete the relation from the internal KB of the framework. Both sub-scenarios are analyzed in this sub-section. The steps for the first case are annotated with the suffix **(a)** whereas for the second case **(b)**. Moreover, in the dialogue protocol and the example *S-* is used for dialogue performed by the framework and *U-* for dialogue performed by the user.

Dialogue protocol for the wrong scenario.

- **Step 1 (a):** Given a question to the framework by the user, the framework will then ask if everything is fine or there is something missing or wrong in its answer. The user should then indicate which entity is wrong. If the keyword(s) in the question is/are part of the internal KB of the framework, which the framework totally trusts, the framework does not allow any dispute. The reason why the framework is so strict when a user is trying to find something wrong in an answer that came purely from information of the internal KB, is because the internal KB is constructed on studies that have already been evaluated for the quality of knowledge that they contain (Liao et al., 2019; Puig et al., 2018; Vassiliades et al., 2020). Therefore, the framework should prevent the user from questioning this form of knowledge.

- **Step 1 (b):** Given a question to the framework by the user, the framework will then ask if everything is fine or there is something missing or wrong in its answer. The user should then indicate which entity is wrong. If the keyword(s) of the question is/are *not* part of the internal KB of the framework, the framework will then proceed to the next step.
- **Step 2 (b):** At this point the user must provide a trustworthy KB in which she found the information which indicates that some of the answers are wrong. If the *KB* is trustworthy, i.e., it has trust score greater than 50%, and the framework has found the information that the user supports, the framework will proceed to *Step 3 (b)*.
- **Step 3 (b):** The user has indicated a trustworthy *KB*, and the framework will then ask a human *arbitrator* if (s)he considers that the entity should or should not be part of its answers.

  – **Arbitrator answers yes**; in this case the framework has won the argumentation dialogue, it informs the user and reduces the trust score of the fact in the KB that the user provided by 10%, and by extension the trust score of the KB.

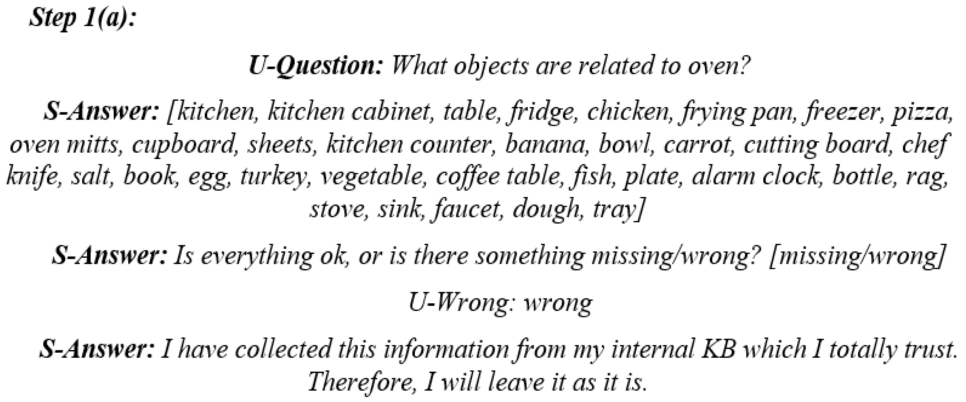
  – **Arbitrator answers no;** in this case the user has won the argumentation dialogue and the framework proceeds to Step 4 (b), increasing the trust score of the fact in the KB that the user provided by 10%, and by extension the trust score of the KB.

  Notice that in both cases, if the trust score of the fact is 0 then the framework considers that the *KB* which the user indicated does not contain the fact, and it stops the argumentation dialogue. The argumentation dialogue in these cases ends with the text *"Sorry I could not find the information that you point."*

- **Step 4 (b):** In this final step, the component has found the information that the user is indicating in the *KB*. The component informs her/him with a message *S* - *"I can see that I was wrong. I will delete this information from my KB"*, and it deletes from its internal knowledge base the triplet that relates the keyword, based on which the query from **Step 1 (b)** was performed, with the indicated (wrong) entity. Thus, if the user asks again the question from **Step 1 (b)** the wrong entity will not be part of the component's answers.

Example 7 demonstrates the argumentation dialogue that follows **Step 1 (a)**, and Example 8 the argumentation dialogue that follows **Step 1 (b)**. For the Example 8, consider that the user has provided a trustworthy KB.

Example 7.



Step 1(a):

U-Question: *What objects are related to oven?*

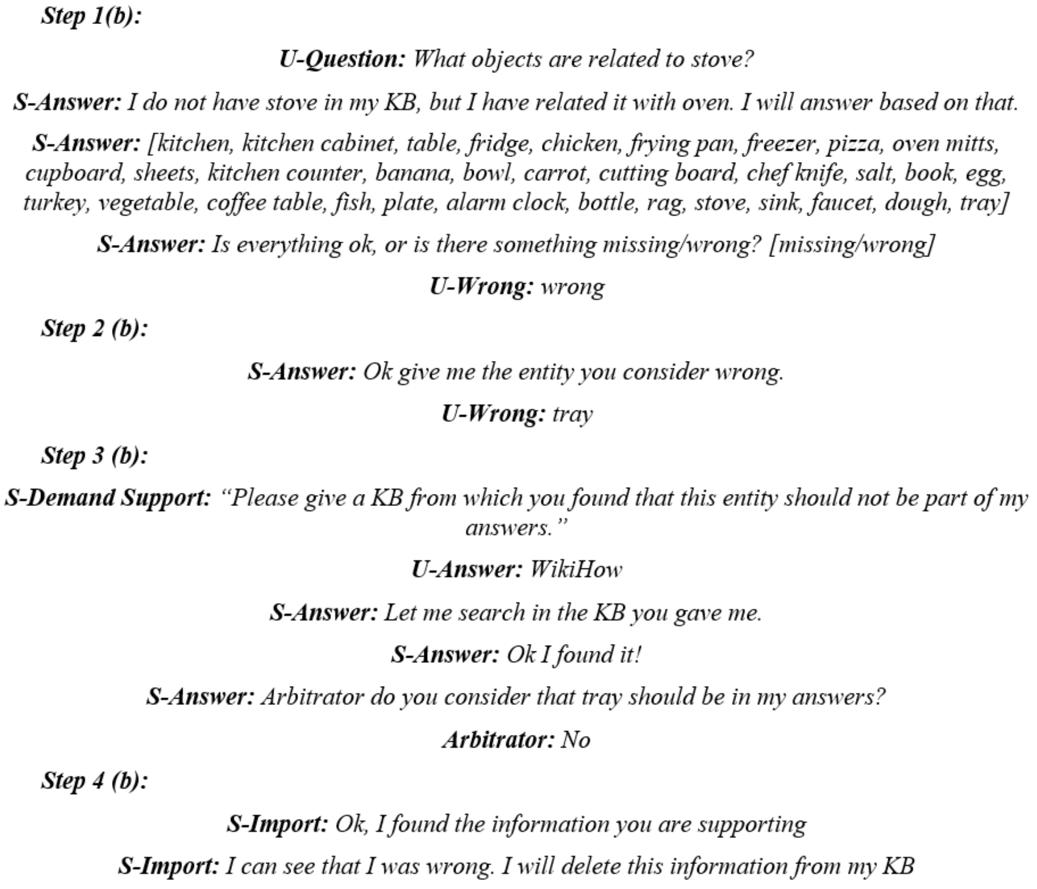
S-Answer: *[kitchen, kitchen cabinet, table, fridge, chicken, frying pan, freezer, pizza, oven mitts, cupboard, sheets, kitchen counter, banana, bowl, carrot, cutting board, chef knife, salt, book, egg, turkey, vegetable, coffee table, fish, plate, alarm clock, bottle, rag, stove, sink, faucet, dough, tray]*

S-Answer: *Is everything ok, or is there something missing/wrong? [missing/wrong]*

U-Wrong: wrong

S-Answer: *I have collected this information from my internal KB which I totally trust. Therefore, I will leave it as it is.*

Example 8.



Step 1(b):

U-Question: *What objects are related to stove?*

S-Answer: *I do not have stove in my KB, but I have related it with oven. I will answer based on that.*

S-Answer: *[kitchen, kitchen cabinet, table, fridge, chicken, frying pan, freezer, pizza, oven mitts, cupboard, sheets, kitchen counter, banana, bowl, carrot, cutting board, chef knife, salt, book, egg, turkey, vegetable, coffee table, fish, plate, alarm clock, bottle, rag, stove, sink, faucet, dough, tray]*

S-Answer: *Is everything ok, or is there something missing/wrong? [missing/wrong]*

U-Wrong: *wrong*

Step 2 (b):

S-Answer: *Ok give me the entity you consider wrong.*

U-Wrong: *tray*

Step 3 (b):

S-Demand Support: *"Please give a KB from which you found that this entity should not be part of my answers."*

U-Answer: *WikiHow*

S-Answer: *Let me search in the KB you gave me.*

S-Answer: *Ok I found it!*

S-Answer: *Arbitrator do you consider that tray should be in my answers?*

Arbitrator: *No*

Step 4 (b):

S-Import: *Ok, I found the information you are supporting*

S-Import: *I can see that I was wrong. I will delete this information from my KB*

EVALUATION

The evaluation of the framework comprises two basic pipelines. The first one is for the knowledge retrieval component of the framework, and the second one for the learning-through-argumentation component of the framework. This separation is reasonable because if the users were asked to evaluate their experience with the framework in general, it would not be possible to reveal which were the component(s) and the aspect(s) that the users did not like. Therefore, evaluating separately the two basic components of the framework, indicates which one needs (or not) improvement. The UI that was used during the evaluation is slightly different than the current UI, because in the previous one there was a mechanism where the user could evaluate each component.

Knowledge Retrieval Evaluation

The knowledge retrieval component was evaluated via two different user evaluations. Firstly, by asking people how much they are satisfied with the results returned. Basically, the evaluation tried to explore whether the answers returned by the framework satisfied the users in terms of commonsense. Since one cannot define strict rules on what can be considered as commonsense, each subject gives their opinion to evaluate how satisfied they are with each answer. Thus, users were asked for a score from 1 to 5 to eight categories of queries. Each person had to evaluate 40 answers (5 queries of each of the eight categories). People both related to Computer Sciences (CSc) and not related to Computer Science (N-CSc) were employed, resulting in 19 and 23 subjects, respectively. Another clustering with the same people based on their education level, Workers 13 (W) that did not go to University, Bachelor/Master Students 23 (B/M), PhD Students 6 (P), was also made.

Table 2. Table with the query categories

	Query	**Input**	**Output**
Q1	*On what objects can I perform the actions X1,..,Xn if I am in room Y?*	*actions X1,..,Xn & condition Y*	*objects O1,...,Ot*
Q2	*On what objects can I perform the actions X1,..,Xn?*	*actions X1,..,Xn*	*objects O1,...,Ot*
Q3	*What can I do with objects O1,...,Om?*	*objects O1,...,Om*	*actions X1,..,Xl*
Q4	*What objects are related to objects O1,...,Om?*	*objects O1,...,Om*	*objects O1,...,Ot*
Q5	*Give me the category of activities for A*	*activity A*	*activities A1,...,An*
Q6	*Give me related objects to O1,...,Om*	*objects O1,...,Om*	*objects O1,...,Ot*
Q7	*Give me similar action(s) to X*	*action X*	*actions X1,..,Xl*
Q8	*Recommend an Activity based on the description* A	*description* A	*activity A*

The categories of queries that were evaluated are displayed in Table 2. Notice that Q4 involves objects that do not exist in the internal KB of the framework. Q4 was created in order to see how satisfied people are with the recommendations from Algorithm 1. Table 3 and Table 4 present the Mean and Variance scores, respectively. The results are rounded to two decimals in all the tables.

Table 3. Table with Mean scores for Q1...Q8

	General	**W**	**B/M**	**P**	**CSc**	**N-CSc**
Q1	4.20	4.18	4.17	4.22	4.21	4.29
Q2	4.35	4.36	4.39	4.32	4.39	4.35
Q3	4.08	4.08	4.16	4.06	4.10	4.08
Q4	3.79	3.78	3.76	3.79	3.78	3.79
Q5	4.19	4.16	4.24	4.16	4.16	4.18
Q6	4.11	4.09	4.12	4.10	4.11	4.09
Q7	3.99	4.09	3.97	3.95	3.91	4.06
Q8	4.12	4.10	4.16	4.25	4.02	4.09
Mean	4.12	4.12	4.13	4.12	4.1	4.12

Table 4. Table with Variance scores for Q1...Q8

	General	W	B/M	P	CSc	N-CSc
Q1	0.96	1.52	0.95	0.78	1.20	0.74
Q2	0.80	0.83	0.76	0.92	0.71	0.87
Q3	1.12	0.5	1.44	0.91	1.25	1.06
Q4	1.52	1.06	1.52	1.69	1.51	1.51
Q5	1.61	1.54	1.59	1.65	1.73	1.49
Q6	0.98	0.86	0.95	1.09	0.97	0.98
Q7	1.75	1.75	1.82	1.38	1.88	1.66
Q8	1.56	1.46	1.54	1.74	1.64	1.52
Variance	1.20	1.13	1.21	1.21	1.26	1.13

As one can see, an overall score of 4.10/5 was obtained, which translates to an 82% score. Moreover, the low score of Q4 in comparison to other queries can be attributed to the fact that there was a very high threshold value to the *Ratcliff-Obershelp* string similarity metric, which compared the returned results from Algorithm 1 with the ones in the internal KB of the framework. On top of that, the entity from the internal KB with which the result of Algorithm 1 was close enough was displayed and not the recommendation from Algorithm 1 (i.e., SMA). The threshold was 0.8 and was reduced to 0.6; for smaller values the recommendations of Algorithm 1 in most cases were not related to the target application. Therefore, the value of the threshold was reduced, and the external recommendation was displayed. These changes affected only Q4. The new results are displayed in Table 5. Observe that the Mean score for Q4 increased by 13.5%, and the Variance shows that the scoring values came closer to the Mean value.

In the second evaluation for the knowledge retrieval component, 5 subjects which were not part of the first group were asked to give their own answers to queries Q1...Q7. Q8 was excluded because the 5 subjects were reluctant to answer it, considering it very time consuming (it required to provide 25 full sentences; not just words as in the case of the other queries), so a quantitatively appropriate dataset could not be gathered. Therefore, the 5 subjects had to give us 5 answers based only on their opinion/experience for 5 queries from each one of Q1...Q7. The result was a baseline dataset of 125 answers for each query. Next, 34 subjects from the first evaluation agreed to proceed with the second round of evaluation. Each user had to give 5 answers to 5 queries from each one of the categories Q1...Q7 (5*5*7= 175 answers in total) using options from the aforementioned datasets. The datasets were collected with spreadsheets, as it was less time consuming for the subjects. On the other hand, in the second round of evaluation the 34 subjects were given the options with which they could answer each question, and they had to use the UI.

Table 5. Table with Mean and Variance scores for Q4, with the new changes

	General	W	B/M	P	CSc	N-CSc
Mean	4.41	4.35	4.36	4.6	4.45	4.36
Variance	0.61	0.74	0.65	0.39	0.42	0.73

Table 6. Table of accurately predicted answers to questions

General	W	B/M	P	CSc	N-CSc
84.1%	82.7%	89.6%	83.3%	83.8%	84.3%

Subsequently, the answers of the second group of subjects were compared with the answers of the knowledge retrieval component, over the same questions. More specifically, given a question *Q* and the subject's answer *a*, if *a* existed in the answers of the component for the same *Q* this would be considered as accurately predicted, otherwise it would be considered as wrongly predicted. In Table 6 one can see the percentage of accurately predicted answers over the total number of questions. See that the component achieved an 84.1% total score over all queries, which is high considering that this is not a data driven framework which could learn the connections between the queries and answers, nor uses embeddings between queries and answers that could point to the correct answer.

Information Retrieval Evaluation Discussion: Considering potential biases, notice that between the first and second evaluation there was a time lapse of over 40 days, so the subjects could not have recalled their answers from the first evaluation. Furthermore, although 9 predefined SPARQL templates exist only 8 were used in the first evaluation; the one omitted involves the activities that are part of the VirtualHome dataset, so this was already evaluated by previous related work.

Finally, looking at the results of the evaluation, the following conclusions can be driven. Firstly, the large percentage (82%) on how much satisfied with the answers of the knowledge retrieval component the subjects are, signifies that the framework can be used by any cognitive robotic system acting in a household environment as a primary (or secondary) source of knowledge. Secondly, the second method of evaluation implies that the knowledge retrieval component could be used as a baseline for evaluating other cognitive robotic systems acting in a household environment. Thirdly, the scores that Algorithm 1 achieved, show that it can be used as an individual service for semantically matching entities of a knowledge graph with entities from ConceptNet, using semantic similarity from DBpedia and WordNet, as it can be easily extended with more properties

Comparison with Baselines Methods

In this sub-section, the performance of the SMA is compared with baseline methods which access the single sources ConceptNet, WordNet, and DBpedia, individually, for semantic matching of two labels. The gold standard dataset of *Q4*, which addresses queries with labels that do not exist in the KB of the framework, was used because it contains labels that are part of all external KBs (ConceptNet, WordNet, DBpedia). Moreover, the semantic similarity between two labels was annotated by users which tackles any possible bias in favor of the framework.

The gold standard dataset for Q4 contains 125 object-object pairs, where one is the label that was given to the framework in question Q4, and the other one is the answer of one of the users that took part in the user evaluation of the Knowledge Retrieval Evaluation sub-section. For instance, consider the question *"What objects are related to object **beef**?"*, where a user gave three answers *chicken-table-knife*. Then, three object-object pairs will be created *beef-chicken*, *beef-table*, and *beef-knife*. These pairs are considered semantically related, and therefore the SMA of the framework should return the relations between them.

The algorithm is evaluated using the usual precision, recall and F1-score used for information retrieval systems (Equations 4, 5 and 6).

$$precision = \frac{|\{Relevant\ Pairs\} \cap \{Retrieved\ Pairs\}|}{|\{Retrieved\ Pairs\}|} \quad (4)$$

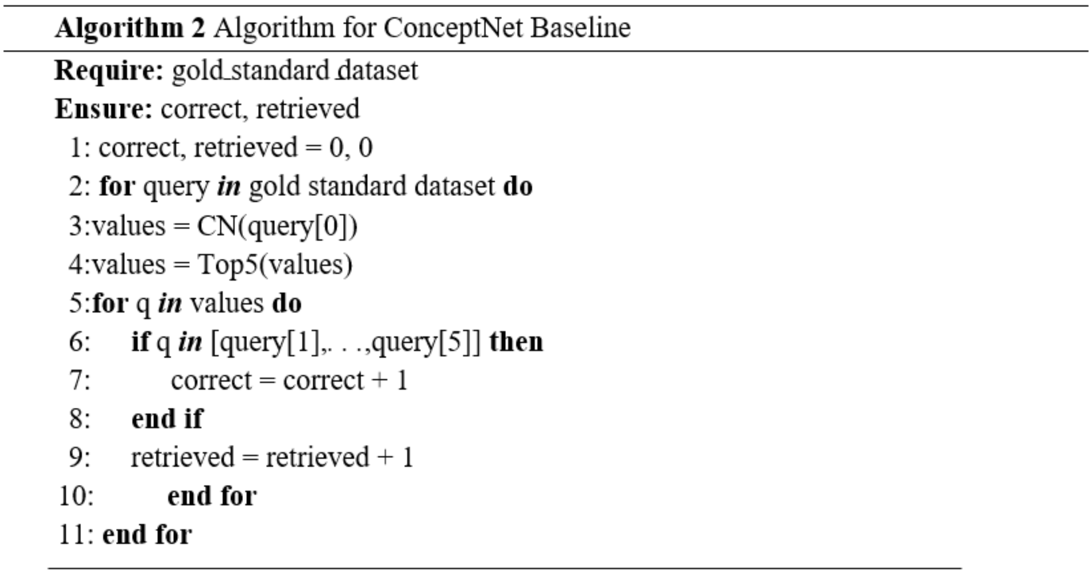
$$recall = \frac{|\{Relevant\ Pairs\} \cap \{Retrieved\ Pairs\}|}{|\{Relevant\ Pairs\}|} \quad (5)$$

$$F1 = 2 * \frac{recall * precision}{recall + precision} \quad (6)$$

Relevant Pairs are those that belong in the gold standard dataset and *Retrieved Pairs* are the pairs that the SMA retrieved (i.e., the 125 pairs of the gold standard dataset). Its precision, recall and F1-score are shown in Table 7 (it retrieved correctly 114 out of 125). Notice that for the evaluation of the method (and for most of the baseline methods) the number of retrieved pairs is equal to the number of relevant pairs, since for each question the Top 5 answers were used in the retrieval methods; also, the gold standard dataset contains 5 answers for each question. This leads to the same (or almost the same) results for precision, recall, and F1-s core metrics.

ConceptNet: For ConceptNet the Web API of NLTK[4] was used, to get the values of the *RelatedTo, UsedFor, AtLocation, CapableOf, Causes, ReceivesAction*, and *IsA*, which are the same properties used in the SMA, and their weights of similarity.

For each label that belongs in a query of the gold standard dataset the Top 5 values are retrieved for each property, and if the values belong to the gold standard dataset they are considered as Relevant (it retrieved correctly 10 out of 125). The values of the properties are sorted according to their value of similarity. Algorithm 2 shows the retrieve procedure followed for ConceptNet.



Algorithm 2 Algorithm for ConceptNet Baseline

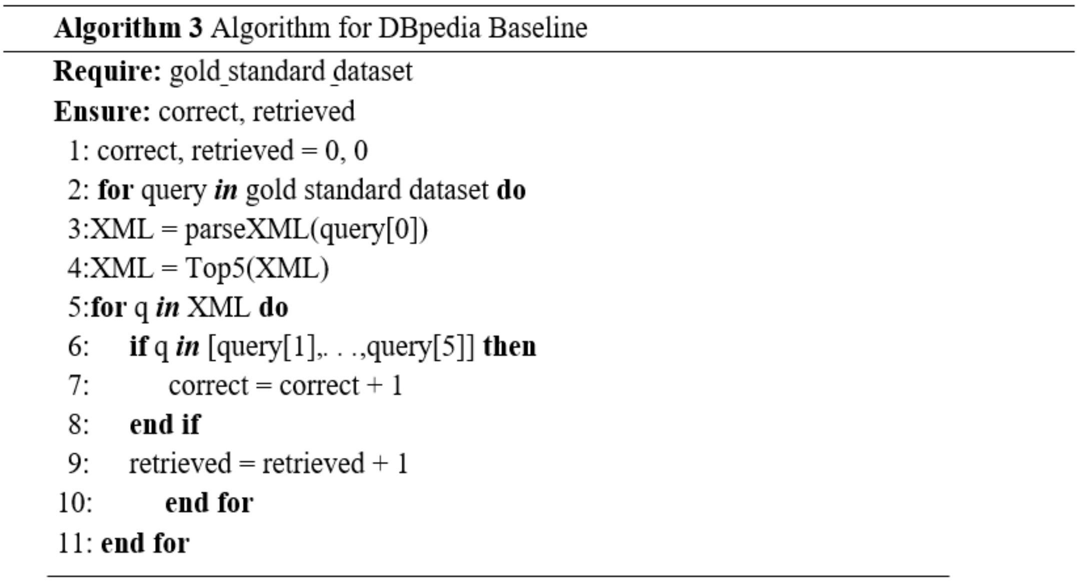
```
Require: gold_standard_dataset
Ensure: correct, retrieved
 1: correct, retrieved = 0, 0
 2: for query in gold standard dataset do
 3:values = CN(query[0])
 4:values = Top5(values)
 5:for q in values do
 6:    if q in [query[1],. . .,query[5]] then
 7:        correct = correct + 1
 8:    end if
 9:    retrieved = retrieved + 1
10:        end for
11: end for
```

Algorithm 2, needs as input the gold standard dataset, and it will return the number of Retrieved Pairs that are Relevant. Let, the *gold standard dataset* contain lists of 6 labels, where the first label represents the entity that was given in the query, and the other five be the answers given by the users.

Next, the values of the properties *RelatedTo, UsedFor, AtLocation, CapableOf, Causes, ReceivesAction*, and *IsA* were gathered with the function *CN()* (line 3), and only the Top 5 were kept for each property based on the weight that ConceptNet gives, with the function *Top5()* (line 4). Finally, if any of the properties' values belong in the gold standard dataset then those are considered as correctly retrieved (lines 5-10). Notice that each question is unique in the gold standard dataset. Table 7 shows the precision, recall and F1-score results for the ConceptNet baseline method.

DBpedia: For DBpedia, *DBpedia Lookup*[9] was used, where for each label in a question of the gold standard dataset, a query to DBpedia Lookup was casted. Next, in the XML file that DBpedia Lookup returns, each retrieved label is searched in the gold standard dataset. If found, then it is considered as Relevant (51 out of 125 were retrieved correctly). Moreover, when the query to DBpedia Lookup was casted only the Top 5 most related entities, according to DBpedia similarity framework, were retrieved. Algorithm 3 shows the retrieval procedure followed for DBpedia.

In Algorithm 3 the XML file that DBpedia Lookup returns is parsed with the function *parseXML()* (line 3), and only the Top 5 most related entities are kept, with the function *Top5()* (line 4). Finally, if any of the values returned (from parseXML) belong in the gold standard dataset then those are considered as correctly retrieved (line 5-10). Table 7 displays the precision, recall, and F1-score of DBpedia.



```
Algorithm 3 Algorithm for DBpedia Baseline
Require: gold_standard_dataset
Ensure: correct, retrieved
correct, retrieved = 0, 0
for query in gold standard dataset do
    XML = parseXML(query[0])
    XML = Top5(XML)
    for q in XML do
        if q in [query[1],. . .,query[5]] then
            correct = correct + 1
        end if
        retrieved = retrieved + 1
    end for
end for
```

WordNet: For WordNet the Top 5 most similar pairs, based on the *Wu Palmer Similarity* (WUP) similarity, for each one of the labels that was casted as a query of the gold standard dataset, were retrieved. Thus, if a pair belongs in the gold standard dataset, it would be considered as Relevant (it retrieved correctly 94 out of 125).

The WUP uses the acyclic graph of WordNet to calculate relatedness by considering the depth of two nodes in the WordNet taxonomies, along with the depth of their Least Common Subsumer (LCS). Given two nodes from the WordNet acyclic graph, the LCS of these nodes is their most specific common ancestor. The score can never be zero because the depth of the LCS is never zero (the depth of the root of the taxonomy is one). This metric calculates the similarity based on how close the nodes are to each other in the WordNet acyclic graph. The WUP similarity between two nodes (n_1, n_2) is defined as

$$WUP(n_1, n_2) = 2 * \frac{depth\big(LCS(n_1, n_2)\big)}{depth(n_1) + depth(n_2)} \quad (7)$$

where *depth*($\cdot$) is the depth of an entity in the WordNet graph.

In Algorithm 4 the WUP similarity was gathered for each entity that is at hops 1 or 2 distance in the directed acyclic graph of WordNet from the entity *query[0]*, with the function *returnWUP()* (line 3), and only the Top 5 of them were kept, with the function *Top5()* (line 4). By experimentation, it was noticed that entities at distance greater than 2 (i.e., hop 3 and more) have very small WUP similarity and they are never part of the Top 5 returned entities; this led to the decision to retrieve entities only up to distance 2. Finally, if any of the values belongs in the gold standard dataset then this is considered as correctly retrieved (line 5-10). Table 7 displays the precision, recall, and F1-score of the baselines.

Table 7. Precision-Recall-F1 of Baselines

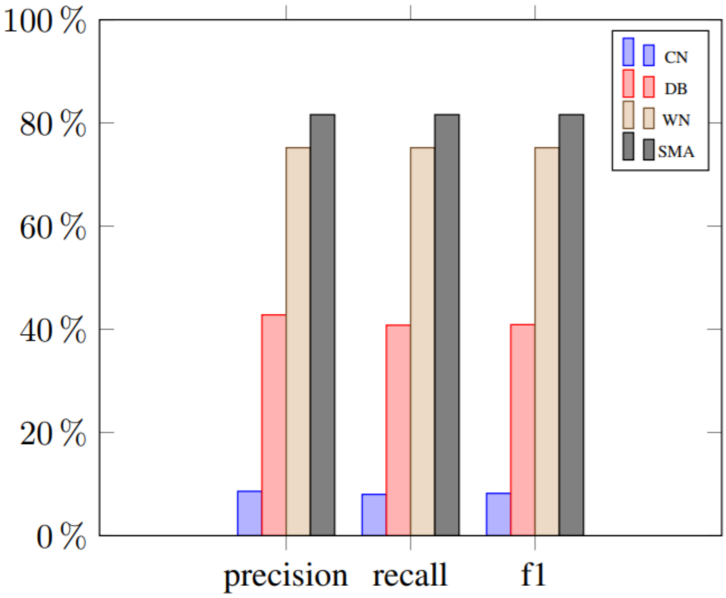
Baseline	**Precision**	**Recall**	**F1**
ConceptNet	8.6%	8%	8.2%
DBpedia	42.8%	40.8%	40.9%
WordNet	75.2%	75.2%	75.2%
SMA	81.6%	81.6%	81.6%

Algorithm 4 Algorithm for WordNet Baseline

```
Require: gold standard dataset
Ensure: correct, retrieved
correct, retrieved = 0, 0
for query in gold standard dataset do
WUP = returnWUP(query[0])
WUP = Top5(WUP)
for q in WUP do
    if q in [query[1],. . .,query[5]] then
        correct = correct + 1
    end if
    retrieved = retrieved + 1
end for
end for
```

Figure 5 shows a bar chart with the precision, recall, and F1-score of each baseline method (ConceptNet (CN), DBpedia (DB), WordNet (WN)), compared with the SMA.

Figure 5. Precision-Recall-F1 of each baseline method compared with the SMA



The SMA has better precision, recall, and F1-score than any of the other method which were consider as baseline. This is because the SMA combines knowledge from ConceptNet, DBpedia, and WordNet, to understand sophisticated similarity relations between two entities. On the other hand, WordNet and DBpedia use only the topology of their knowledge graph, and ConceptNet uses only the weights that the community defines for the similarity of two entities.

Learning-through-Argumentation Evaluation

The learning-through-argumentation component of the framework was evaluated with a user evaluation, through a Web UI. To tackle potential biases, 5 spreadsheets from the previous evaluation (see Knowledge Retrieval Evaluation sub-section) were constructed, which contained for each category of questions Q1...Q8, 5 unique questions along with the input that the user gave and the output that the framework produced, resulting to 200 questions (5 spreadsheets*8 categories*5 questions in each category). Then, one spreadsheet was given to each of the 5 individuals, and they were asked to provide 3 missing entities and 3 wrong entities to the questions contained in their spreadsheet. The result was a gold-standard dataset of 600 missing argumentation scenarios, and another 600 wrong argumentation scenarios, that would be given to another group of individuals for evaluation. Some of the 5 individuals in this step were part of the evaluation for the knowledge retrieval component. This part of the evaluation for the learning-through-argumentation component was performed with spreadsheets because it was less time consuming for the subjects. The next steps of the evaluation were performed with the Web UI.

Next, a group of 34 subjects had to give an opinion on how satisfied they are with the argumentation scenarios gathered, based on a Likert scale, where 1 is the worst score and 5 the best score. Some individuals were part of the evaluation for the knowledge retrieval component. Each individual, using the Web UI, had to perform 3 argumentation dialogues for batches of 5 questions for each one of Q1...Q8; thus, each individual had to evaluate 120 missing scenarios and 120 wrong scenarios. More specifically, the second group of individuals was given a set of questions, as well as 3 missing entities and 3 wrong entities for each question, on which they had to perform argumentation dialogues, and subsequently they had to evaluate each one, based on how satisfied they are with the interaction with the framework. The evaluation wanted to explore whether the argumentation dialogues that the framework generates are rational, based on the commonsense that each human has, and if the users would be convinced by the framework after they had argued with it. Individuals were clustered as in Knowledge Retrieval Evaluation sub-section; Table 8 shows the mean score for the missing knowledge scenario, and Table 9 shows the mean for the wrong knowledge scenario. The results seem promising, as the missing scenario got an 79% overall score, and the wrong scenario got an 80% overall score.

Table 8. Table of mean scores for the missing knowledge scenario of Q1...Q8

	General	**W**	**B/M**	**P**	**CSc**	**N-CSc**
Q1	3.92	3.91	3.92	3.93	3.88	3.94
Q2	3.95	3.89	4	3.97	3.87	3.98
Q3	3.95	3.96	3.95	3.96	3.96	3.96
Q4	4.01	4.08	4.02	3.69	4.02	4
Q5	4.01	3.97	4.08	3.93	3.98	4.03
Q6	3.96	4.03	3.93	3.8	4.02	3.93
Q7	4.03	3.96	4.07	4.18	4.06	4.02
Q8	4	3.99	4.04	3.91	4.03	3.99
Mean	3.98	3.97	4	3.92	3.98	3.98

Table 9. Table of mean scores for the wrong knowledge scenario of Q1...Q8

	General	**W**	**B/M**	**P**	**CSc**	**N-CSc**
Q1	4.03	4.06	4.04	3.92	4.03	4.04
Q2	3.95	4.02	3.9	3.9	3.97	3.94
Q3	3.95	4.02	3.92	3.76	3.92	3.96
Q4	4	3.97	4.05	3.92	3.97	4.01
Q5	4.03	4.03	4	4.13	4.02	4.03
Q6	4.06	4.09	4.03	4.03	4.05	4.02
Q7	4.05	4.15	3.94	4.03	4.01	4.06
Q8	4.25	4.24	4.33	3.99	4.25	4.25
Mean	4.04	4.07	4.03	3.96	4.03	4.05

Learning-through-Argumentation Discussion: This method of evaluation where the user must use their commonsense, to evaluate the argumentation component, indicates whether a user would have found rational the argumentation dialogue that the component produce, and if the argumentation component would manage to convince the user. The argumentation scenarios (the gold standard) were constructed to tackle potential biases, since the evaluators used the system in querying cases with established disputes. If this step was omitted, it could have been considered that argumentation scenarios that favor the argumentation scheme had been selected.

Notice that during the evaluation when the user needed to suggest an external KB to back up her/his argument, the *WikiHow* KB was always suggested to the users as an external source. The reason for this was because not all users are familiar with Semantic Web knowledge resources and what kind of information they contain.

Finally, in the argumentation for the wrong knowledge scenario in **Step 3 (b)**, in the ideal scenario, a second user (the arbitrator) should be present to make the final decision, if the user or the system is the winner. This was simulated by flipping a coin.

CONCLUSION

In this paper, an open-ended web knowledge retrieval framework for the household domain with external knowledge that can argue over the information that it returns and learn new knowledge or refine existing knowledge through an argumentation dialogue with the user, was presented. To the best of our knowledge the combination of these two cognitive processes has not been given much attention in the literature, even though it is important to have a knowledge retrieval framework that can argue over the information that it returns, because it can convince the user about its answers with methods closer to the human way of thinking.

The framework is oriented to the context of a household environment, because this is the most common environment for a cognitive robotic system (Vassiliades et al., 2020; Gouidis et al., 2019). Information was extracted from the VirtualHome dataset (Liao et al., 2019; Puig et al., 2018) and it was fused into the framework. Furthermore, with an instance generator algorithm the activities, from the VirtualHome dataset, were translated as instances of the ontology classes. Therefore, the framework can obtain knowledge, about how actions and objects are related, what objects are related to each other, what objects and actions exist in an activity, and suggestions on how to perform an activity in a household environment, through a set of predefined SPARQL query templates. The knowledge retrieval component of the framework can also address ad hoc SPARQL queries to its own KB. Additionally, the range of queries the framework can answer was broadened by developing a Semantic-Matching Algorithm that finds semantic similarity, between entities existing in the internal KB of the framework, and entities from the knowledge graph of ConceptNet, using semantic knowledge from DBpedia and WordNet.

In the learning-through-argumentation component of the framework two common scenarios were modelled that the user can argue over the framework's returned information. The first one is the missing knowledge scenario where the user indicates an entity that should be part of the answers that the framework returns. The second one is the wrong knowledge scenario where the user indicates that an entity should not be part of the answers that the framework returns. In both cases, the user must back up her/his argument with a trustworthy knowledge source. If the outcome of the argumentation dialogue is in favor of the user, then the framework refines its knowledge base and learns new knowledge.

The evaluation of the knowledge retrieval component of the framework highlights three points. Firstly, the large percentage (82%) on how much satisfied with the answers of the knowledge retrieval component the subjects are, signifies that the framework can be used by any cognitive robotic system acting in a household environment as a primary (or secondary) source of knowledge. Secondly, the second method of evaluation implies that the knowledge retrieval component could be used as a baseline for evaluating other cognitive robotic systems acting in a household environment. Thirdly, the scores that the Semantic Matching Algorithm achieved through the evaluation of the component as well as by comparing it with baseline methods that use the native retrieval APIs of the external knowledge sources, show that it can be used as an individual service for semantically matching entities.

On the other hand, the evaluation of the learning-through-argumentation component of the framework indicates whether a user finds rational and convincing the argumentation dialogue that the component produces. The component achieves its purpose to a large extent as both argumentation scenarios (missing and wrong answer) got a satisfaction score close to 80%.

The limitations of this study, in both components, are as follows. First, the knowledge retrieval component is currently restricted to the household domain for finding object-object, object-action, and object-activity relations, among others. To extend the scope of the framework to other objects and activities (outside the household domain), the ontology should be extended with new classes and relations to represent the new knowledge. Another, limitation lies in the Semantic Similarity Algorithm as the information that it returns still contains a lot of noise, even in the "controlled way" that is being used (i.e., in a specific domain with a specific set of queries). Therefore, more sophisticated methods (apart from the metric that was proposed) should be developed to prune the returned answers. Furthermore, the learning-through-argumentation component currently supports only the wrong and missing knowledge scenarios, whereas humans may argue in many more ways. For this reason, more scenarios should be developed in the future, or rather a flexible dialogue protocol that can capture multiple argumentation scenarios. Finally, the learning-through-argumentation component is currently based on the Abstract Argumentation Framework (Dung 1995), the most well-established argumentation framework, whose abstract nature, however, does not allow much of explainability. To overcome this, other argumentation frameworks could be considered to compute arguments, such as the Structured Argumentation Framework (Modgil & Prakken 2014) or the Argumentation Framework with Domain Assignments (Vassiliades et al. 2021c).

As for future work, the scheme of the ontology is planned to be extended with spatial information about objects, for example *soap is usually found near sink, sponge, bathtub, shower, shampoo*. To add spatial information to the framework more external knowledge bases, such as BabelNet (Navigli & Ponzetto 2010) and VisualGenome (Krishna et al. 2017), need to be added. Also, the Semantic Matching Algorithm will be extended by obtaining information from other ontologies. Moreover, a user ID will be introduced so that the system can keep a trust score for each user, and when the user is considered trustworthy (s)he will not need to provide an external KB to back up her opinion in the argumentation dialogue. This could also tackle the need of a human arbitrator in the wrong knowledge scenario. The trust score of the user could be computed by the framework based e.g., on the quantity of argumentation dialogues the user won. Finally, the framework could be extended to accept questions in natural language using methods from the research area of Semantic Question Answering (Antoniou & Bassiliades, 2022). These methods allow for mapping natural language questions to SPARQL query templates, which subsequently can be posed to the underlying knowledge graph(s).

Conflict of Interest

The authors of this publication declare there is no conflict of interest.

Funding Agency

This research was supported by the European Regional Development Fund of the European Union, and by Greek national funds through the operational program, Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH–CREATE–INNOVATE (project code: TΔEK 00343 - Energy Controlling Voice Enabled Intelligent Smart Home Ecosystem).

REFERENCES

Akhdinirwanto, R., Agustini, R., & Jatmiko, B. (2020). Problem-based learning with argumentation as a hypothetical model to increase the critical thinking skills for junior high school students. *Jurnal Pendidikan IPA Indonesia*, *9* , 340–350.

Aleven, V., & Ashley, K. D. (1997). Evaluating a learning environment for case-based argumentation skills. In *Proceedings of the 6th international conference on Artificial intelligence and law* (pp. 170–179).

Antoniou, C., & Bassiliades, N., "A Survey on Semantic Question Answering Systems", The Knowledge Engineering Review, vol. 37, p. e2, 2022.

Ashley, K. D., Desai, R., & Levine, J. M. (2002). Teaching case-based argumentation concepts using dialectic arguments vs. didactic explanations. In *International Conference on Intelligent Tutoring Systems* (pp. 585–595). Springer.

Bai, S., & Khoja, S. A. (2021). Hybrid Query Execution on Linked Data With Complete Results. *International Journal on Semantic Web and Information Systems (IJSWIS)*, *17*(1), 25-49.

Beetz, M., Bálint-Benczédi, F., Blodow, N., Nyga, D., Wiedemeyer, T., & Marton, Z.-C. (2015). Robosherlock: Unstructured information processing for robot perception. In 2015 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1549–1556). IEEE.

Beetz, M., Beßler, D., Haidu, A., Pomarlan, M., Bozcuoğlu, A. K., & Bartels, G. (2018). Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 512–519). IEEE.

Berland, L. K., & McNeill, K. L. (2010). A learning progression for scientific argumentation: Understanding student work and designing supportive instructional contexts. Science Education, 94 , 765–793.

Beßler, D., Koralewski, S., & Beetz, M. (2018). Knowledge representation for cognition- and learning-enabled robot manipulation. In CogRob@ KR (pp. 11–19).

Bhattacharyya, R., Bhuyan, Z., & Hazarika, S. M. (2016). O-pro: An ontology for object affordance reasoning. In International Conference on Intelligent Human Computer Interaction (pp. 39–50). Springer.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., & Hellmann, S. (2009). Dbpedia-a crystallization point for the web of data. Web Semantics: science, services, and agents on the world wide web, 7 , 154–165.

Cayrol, C., de Saint-Cyr, F. D., & Lagasquie-Schiex, M.-C. (2008). Revision of an argumentation system. In KR (pp. 124–134).

Chen, Q., Bragg, J., Chilton, L. B., & Weld, D. S. (2019). Cicero: Multi-turn, contextual argumentation for accurate crowdsourcing. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (pp. 1–14).

Chernova, S., Chu, V., Daruna, A., Garrison, H., Hahn, M., Khante, P., Liu, W., & Thomaz, A. (2020). Situated bayesian reasoning framework for robots operating in diverse everyday environments. Springer , (pp. 353–369).

Clark, D. B., Sampson, V., Weinberger, A., & Erkens, G. (2007). Analytic frameworks for assessing dialogic argumentation in online learning environments. Educational Psychology Review, 19, 343–374.

Coste-Marquis, S., Konieczny, S., Mailly, J.-G., & Marquis, P. (2014a). On the revision of argumentation systems: Minimal change of arguments statuses. KR, 14, 52–61.

Coste-Marquis, S., Konieczny, S., Mailly, J.-G., & Marquis, P. (2014b). A translation-based approach for revision of argumentation frameworks. In European Workshop on Logics in Artificial Intelligence (pp. 397–411). Springer.

Cyras, K., Satoh, K., & Toni, F. (2016). Abstract argumentation for case-based reasoning. KR.

Daruna, A., Liu, W., Kira, Z., & Chetnova, S. (2019). Robocse: Robot common sense embedding. In 2019 International Conference on Robotics and Automation (ICRA) (pp. 9777–9783). IEEE.

Dung, P. M. (1995). An argumentation-theoretic foundation for logic programming. The Journal of logic programming. Elsevier, 22(3), (pp. 151-177).

Drapeau, R., Chilton, L., Bragg, J., & Weld, D. (2016). Microtalk: Using argumentation to improve crowdsourcing accuracy. In Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (pp. 210– 218). Volume 4.

Falappa, M. A., Kern-Isberner, G., & Simari, G. R. (2009). Belief revision and argumentation theory. Argumentation in artificial intelligence, (pp. 341–360).

Fan, X., Craven, R., Singer, R., Toni, F., & Williams, M. (2013). Assumption based argumentation for decision-making with preferences: A medical case study. In International Workshop on Computational Logic in Multi-Agent Systems (pp. 374–390). Springer.

Fan, X., & Toni, F. (2015). On explanations for non-acceptable arguments. In International Workshop on Theory and Applications of Formal Argumentation (pp. 112–127). Springer.

Fellbaum, C. (2010). Wordnet. Theory and applications of ontology: computer applications, (pp. 231–243).

Fiorini, R. A. (2020). Computational intelligence from autonomous system to super-smart society and beyond. *International Journal of Software Science and Computational Intelligence (IJSSCI)*, *12*(3), 1-13.

Fischer, L., Hasler, S., Deigmöller, J., Schnürer, T., Redert, M., Pluntke, U., Nagel, K., Senzel, C., Ploennigs, J., Richter, A. et al. (2018). Which tool to use? grounded reasoning in everyday environments with assistant robots. In CogRob@ KR (pp. 3–10).

Gouidis, F., Vassiliades, A., Patkos, T., Argyros, A., Bassiliades, N., & Plexousakis, D. (2019). A review on intelligent object perception methods combining knowledge-based reasoning and machine learning. arXiv preprint arXiv:1912.11861.

Hepp, M. (2008). Goodrelations: An ontology for describing products and services offers on the web. In International conference on knowledge engineering and knowledge management (pp. 329–346). Springer.

Heras, S., Jordán, J., Botti, V., & Julián, V. (2013). Argue to agree: a case-based argumentation approach. International Journal of Approximate Reasoning, 54, 82–108.

Hu B., Gaurav A., Choi C., & Almomani A. (2022). Evaluation and Comparative Analysis of Semantic-Web Based Strategies for Enhancing Educational System Development. International Journal on Semantic Web and Information System (IJSWIS), 18(1).

Icarte, R. T., Baier, J. A., Ruz, C., & Soto, A. (2017). How a general-purpose commonsense ontology can improve performance of learning-based image retrieval. arXiv preprint arXiv:1705.08844 .

Jäger, G., Mueller, C. A., Thosar, M., Zug, S., & Birk, A. (2018). Towards robot centric conceptual knowledge acquisition. arXiv preprint arXiv:1810.03583.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). Fasttext. zip: Compressing text classification models. arXiv preprint arXiv:1612.03651.

Kelly, K. T. (1998). The learning power of belief revision. In TARK (pp. 111–124). Citeseer volume 98.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L., & David A. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision. 123(1), 32-73. Springer.

Lemaignan, S., Warnier, M., Sisbot, E. A., Clodic, A., & Alami, R. (2017). Artificial cognition for social human–robot interaction: An implementation. Artificial Intelligence, 247, 45–69.

Liao, Y.-H., Puig, X., Boben, M., Torralba, A., & Fidler, S. (2019). Synthesizing environment-aware activities via activity sketches. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6291– 6299).

Lin, Y.-R., Fan, B., & Xie, K. (2020). The influence of a web-based learning environment on low achievers' science argumentation. Computers & Education, 151.

Liu, H., & Singh, P. (2004). Conceptnet—a practical commonsense reasoning toolkit. BT technology journal, 22 , 211–226.

Mŏzina, M., Zabkar, J., Bench-Capon, T., & Bratko, I. (2005). Argument based machine learning applied to law. Artificial Intelligence and Law, 13, 53–73.

Mŏzina, M., Zabkar, J., & Bratko, I. (2007). Argument based machine learning. Artificial Intelligence, 171 , 922–937.

Modgil, S., & Prakken, H. (2014). The ASPIC+ framework for structured argumentation: a tutorial. Argument & Computation, 5(1) (pp. 31-62. IOS Press.

Navigli, R., & Ponzetto, S. P. (2010). BabelNet: Building a very large multilingual semantic network. Proceedings of the 48th annual meeting of the association for computational linguistics,(pp 216-225).

Okuno, K., & Takahashi, K. (2009). Argumentation system with changes of an agent's knowledge base. In Twenty-First International Joint Conference on Artificial Intelligence (pp. 312–320).

Ontañón, S., & Plaza, E. (2007). Learning and joint deliberation through argumentation in multiagent systems. In Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems (pp. 1–8).

Ontañón, S., & Plaza, E. (2010). Multiagent inductive learning: an argumentation-based approach. In ICML (pp. 210–217).

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532–1543).

Pilotti, P., Casali, A., & Chesnevar, C. (2014). Incorporating object features in collaborative argumentation-based negotiation agents. In Brazilian Conference on Intelligent Systems (BRACIS)/Encontro Nacional de Inteligencia Artificial e Computacional (ENIAC), Sao Carlos, SP, Brazil (pp. 31–37).

Pilotti, P., Casali, A., & Chesnevar, C. (2015). A belief revision approach for argumentation-based negotiation agents. International Journal of Applied Mathematics and Computer Science, 25 , 455–470.

Pinacho, L. S., Wich, A., Yazdani, F., & Beetz, M. (2018). Acquiring knowledge of object arrangements from human examples for household robots. In Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz) (pp. 131–138). Springer.

Pratiwi, S., Cari, C., Aminah, N., & Affandy, H. (2019). Problem-based learning with argumentation skills to improve students' concept understanding. In Journal of Physics: Conference Series (p. 012065). IOP Publishing volume 1155.

Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., & Torralba, A. (2018). Virtualhome: Simulating household activities via programs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8494–8502).

Radinger, A., Rodriguez-Castro, B., Stolz, A., & Hepp, M. (2013). Baudataweb: the austrian building and construction materials market as linked data. In Proceedings of the 9th International Conference on Semantic Systems (pp. 25–32). ACM.

Rahwan, I., Moraitis, P., & Reed, C. (2004). Argumentation in multi-agent systems. In First International Workshop, ArgMAS (pp. 167–172). Springer.

Ramirez-Amaro, K., Beetz, M., & Cheng, G. (2017). Transferring skills to humanoid robots by extracting semantic representations from observations of human activities. Artificial Intelligence, 247, 95–118.

Rong, X. (2014). word2vec parameter learning explained. arXiv preprint arXiv:1411.2738.

Ruiz-Sarmiento, J.-R., Galindo, C., & Gonzalez-Jimenez, J. (2015). Exploiting semantic knowledge for robot object recognition. Knowledge-Based Systems, 86, 131–142.

Sanfilippo, E. M. (2018). Feature-based product modelling: an ontological approach. International Journal of Computer Integrated Manufacturing, 31 , 1097–1110.

Sap, M., Le Bras, R., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N. A., & Choi, Y. (2019). Atomic: An atlas of machine commonsense for if-then reasoning. In Proceedings of the AAAI Conference on Artificial Intelligence (pp. 3027–3035). volume 33.

Saxena, A., Jain, A., Sener, O., Jami, A., Misra, D. K., & Koppula, H. S. (2014). Robobrain: Large-scale knowledge engine for robots. arXiv preprint arXiv:1412.0691.

Slonim, N., Bilu, Y., Alzate, C., Bar-Haim, R., Bogin, B., Bonin, F., Choshen, L., Cohen-Karlik, E., Dankin, L., Edelstein, L. et al. (2021). An autonomous debating system. Nature, 591, 379–384.

Subhashini, R., & Akilandeswari, J. (2011). A survey on ontology construction methodologies. International Journal of Enterprise Computing and Business Systems, 1, 60–72.

Tenorth, M., & Beetz, M. (2017). Representations for robot knowledge in the knowrob framework. Artificial Intelligence, 247, 151–169.

Vassiliades, A., Bassiliades, N., Gouidis, F., & Patkos, T. (2020). A knowledge retrieval framework for household objects and actions with external knowledge. In International Conference on Semantic Systems (pp. 36–52). Springer, Cham.

Vassiliades, A., Bassiliades, N., & Patkos, T. (2021a). Argumentation and explainable artificial intelligence: a survey. The Knowledge Engineering Review, 36.

Vassiliades, A., Patkos, T., Efthymiou, V., Bikakis, A., Bassiliades, N., & Plexousakis, D. (2021b). Object-action association extraction from knowledge graphs. International Conference on Semantic Systems Amsterdam.

Vassiliades A., Patkos T., Flouris G., Bikakis A., Bassiliades N. & Plexousakis, D. (2021c). Abstract Argumentation Frameworks with Domain Assignments. 30th International Joint Conference on Artificial Intelligence (IJCAI-21)

Veerman, A. L. (2000). Computer-supported collaborative learning through argumentation. Ph.D. thesis Proefschrift Universiteit Utrecht.

Von Aufschnaiter, C., Erduran, S., Osborne, J., & Simon, S. (2008). Arguing to learn and learning to argue: Case studies of how students' argumentation relates to their scientific knowledge. Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching, 45, 101–131.

Wagner, A., & Rüppel, U. (2019). Bpo: The building product ontology for assembled products. Proceedings of the 7th Linked Data in Architecture and Construction workshop (LDAC 2019), Lisbon, Portugal.

Wiedemeyer, T., Bálint-Benczédi, F., & Beetz, M. (2015). Pervasive calm perception for autonomous robotic agents. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (pp. 871–879). International Foundation for Autonomous Agents and Multiagent Systems.

Young, J., Basile, V., Kunze, L., Cabrio, E., & Hawes, N. (2016). Towards lifelong object learning by integrating situated robot perception and semantic web mining. In Proceedings of the Twenty-second European Conference on Artificial Intelligence (pp. 1458–1466). IOS Press.

Young, J., Basile, V., Suchi, M., Kunze, L., Hawes, N., Vincze, M., & Caputo, B. (2017). Making sense of indoor spaces using semantic web mining and situated robot perception. In European Semantic Web Conference (pp. 299–313). Springer.

Zamazal, O. (2020). A survey of ontology benchmarks for semantic web ontology tools. *International Journal on Semantic Web and Information Systems (IJSWIS)*, *16*(1), 47-68.

ENDNOTE

[1] https://github.com/valexande/homeontology-and-argumentation
[2] https://socola.ics.forth.com
[3] https://unity.com
[4] https://www.nltk.org
[5] https://github-wiki-see.page/m/commonsense/conceptnet5/wiki/Relations

[6] https://pypi.org/project/textdistance/
[7] https://www.wikihow.com/Main-Page
[8] https://babelnet.org
[9] https://lookup.dbpedia.org