Argumentation-based Inconsistencies Detection for Question-Answering over DBpedia

Elena Cabrio, Julien Cojan, Serena Villata, and Fabien Gandon

INRIA Sophia Antipolis, France
{firstname.lastname}@inria.fr

Abstract. In the Web of Data, the pieces of information obtained by querying distributed SPARQL endpoints may provide different results for the same query. Moreover, the combination of these query results may lead to an inconsistent set of information about the same topic. In particular, the problem of reconciling information obtained by distributed SPARQL endpoints is encountered in question-answering systems over linked data, where different SPARQL endpoints are queried to retrieve the answer to the user's question. In this paper, we propose to address this problem by adopting argumentation theory to reason over inconsistent information sets, and provide nevertheless a unique and motivated answer to the user. We implement and evaluate our approach on QAKiS (Question Answering WikiFramework-based system), that exploits multilingual chapters of DBpedia as RDF data sets to be queried using a natural language interface.

1 Introduction

In the Web of Data, the combination of the information items concerning a single real-world object coming from different data sources, e.g., the results of a single SPARQL query on different endpoints, may lead to an inconsistent results set. This is an open problem for consuming in the Web of Data since these inconsistencies mine the overall quality of the data itself. In particular, this problem arises while querying the multilingual chapters of DBpedia [12]. Such chapters, well connected through Wikipedia instance interlinking, can in fact contain different information with respect to the English version. Assuming we wish to query a set of multilingual DBpedia SPARQL endpoints with the same query, the answers we collect can be either identical, or one can subsume the other, or they can be contradictory.

In this paper, we answer the following research question: *How to reconcile information provided by the multilingual chapters of DBpedia to obtain a consistent results set?* This issue is particularly relevant to Question Answering (QA) systems over DBpedia [11], where the user expects a unique (and possibly correct) answer to her factual natural language question. In this scenario, another open issue is to motivate the answer the system provides to the user in such a way that the overall question answering system appears as transparent to her, and, as a consequence, more reliable. Following these considerations, our research question breaks down into the following subquestions:

1. How to semantically relate information items provided by different multilingual chapters of DBpedia to detect possible inconsistencies?
2. How to compute the acceptability degree of information items to provide a unique answer?
3. How to motivate and explain the reasons behind the answer provided by the question answering system?

First, we need to detect the semantic relations which relate each piece of information to the others returned by the different sources, i.e., SPARQL endpoints. In particular, two kinds of relations can be highlighted: a *conflict relation* such that the information items represent contradicting answers to a single question (e.g., given the question "*Which is the capital of Italy?*", the answers "*Paris*" and "*Rome*" conflict), and a *support relation* such that the information items are semantically connected in a way that one item may be *derived* from the other by means of an ontology (e.g., given the question"*In which place was William Shakespeare born?*", the answers "*Stratford-upon-Avon*" and "*England*" do not actually conflict because Stratford-upon-Avon is located in England).

Second, we adopt *abstract argumentation theory* [7] to reason over the inconsistencies among a set of information items called *arguments*, and to return a consistent (sub)set of them. Roughly, an abstract argumentation framework is a directed labeled graph whose nodes are the arguments and the edges represent a *conflict* relation. Since there are situations where the sources provide the same answer or a subsumed one, we need to represent also a positive relation among the arguments. For this reason, we rely on bipolar abstract argumentation [4] where also a support relation is considered. We compute the acceptability degree of the arguments depending on the confidence assigned to their sources [5].

Third, the overall argumentation framework together with the acceptability degree of each argument is used to motivate to the user the answer the system returns. We evaluate our approach through its integration in QAKiS [3], that exploits DBpedia multilingual chapters as RDF data sets to be queried using a natural language interface. The argumentation module is embedded to provide a (possibly unique) answer whose acceptability degree is over a given threshold, and the graph structure linking the different answers is provided as motivation. In this paper, we do not address the issue of linked data quality assessment and fusion (Sieve [13]), and we do not improve DBpedias instances alignment. We do not use argumentation theory to find agreements over ontology alignments [14].

The remainder of the paper is as follows: Section 2 provides the basic notions of argumentation theory. Section 3 presents our argumentation-based framework for inconsistencies detection. In Section 4 our approach is evaluated on QAKiS. Section 5 compares the existing research with the proposed approach.

2 Background

A Dung-style abstract argumentation framework [7] (AF) aims at representing conflicts among elements called *arguments* through a binary *attack* (i.e., conflict) relation. The need to introduce also a positive relation among the arguments, i.e., a *support* relation, leads to the proposal of so called *bipolar* argumentation frameworks [4] (BAF). An example of BAF is visualized in Figure 1.b where the dotted edge represents the support relation. This kind of framework allows to reason about these conflicts to detect, starting by a set of arguments and the conflicts among them, which are the *accepted arguments*. Accepted arguments are those arguments which are considered as believable by an external evaluator, who has a full knowledge of the argumentation framework. Roughly, an argument is *accepted* (i.e., labelled *in*) if all the arguments attacking it are rejected, and it is *rejected* (i.e., labelled *out*) if it has at least an argument attacking it which is accepted. Figure 1.a shows an example of abstract argumentation framework. The arguments are visualized as circles, and the attack relation is visualized as edges. Gray arguments are the accepted ones. We have that argument a attacks argument b and b attacks a, and argument c attacks a. Using Dung's acceptability semantics [7], the set of accepted arguments is $\{b, c\}$.

However, associating a *crisp* label, i.e., *in* or *out*, to the arguments is limiting in a number of real life situations where a numerical value expressing the acceptability degree of each argument is required [8, 5, 9]. In particular, da Costa Pereira et al. [5] propose a fuzzy labeling algorithm to account for the fact that arguments may originate from sources that are trusted only to a certain degree. They define α a fuzzy labeling *iff* for all arguments A, $\alpha(A) = \min\{\mathcal{A}(A), 1 - \max_{B:B \to A} \alpha(B)\}$ where $\mathcal{A}(A)$ is given by the trust degree of the most reliable source that offers argument A. Consider the example in Figure 1.a, if we have $\mathcal{A}(a) = \mathcal{A}(b) = \mathcal{A}(c) = 0.8$, then the algorithm returns the following labeling: $\alpha(a) = 0.2$ and $\alpha(c) = \alpha(b) = 0.8$.

3 The framework

This section describes our framework to detect the inconsistencies among the results obtained querying the SPARQL endpoints of multilingual DBpedia chapters: *i)* how we assign an attack or a support relation between two pieces of information depending on the semantic relation linking them (Section 3.1), and *ii)* how compute a fuzzy evaluation of the arguments (Section 3.2).

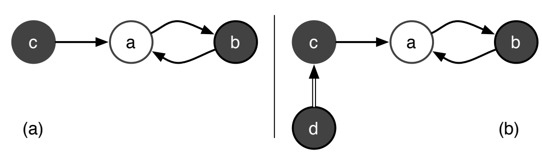

Fig. 1: Example of (a) AF, (b) BAF.

3.1 From ontological relations to argumentation relations

Given a set of answers to a certain factual query, each provided by a different endpoint, we define and apply our algorithm to assign the relations of support and attack between two arguments (where an argument is an answer independently provided by a specific SPARQL endpoint). Multilingual chapters of DBpedia [12], well connected through Wikipedia instance interlinking, can in fact contain different information with respect to the English version. Assuming we wish to query a set of multilingual DBpedia SPARQL endpoints with the same query, the answers we collect can be either identical, or one can subsume the other, or they can be contradictory. Each endpoint is assigned a confidence score, according to the probability that such data set is reliable with respect to the information items it contains. In this case study, we assign an apriori confidence score to the endpoints according to their dimensions and solidity in terms of maintenance, but other methods to assign such scores can be explored as future work (e.g., letting the user to select the endpoints that she considers as more reliable).

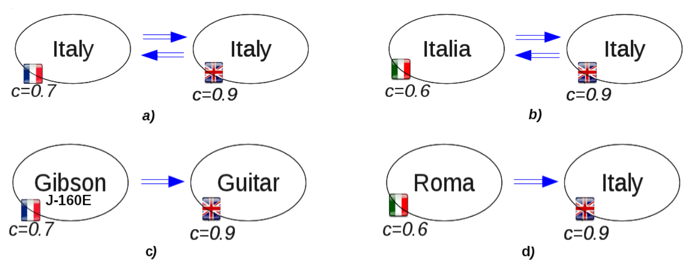

Fig. 2: Examples of support relations: *identity* (a,b), *subsumption* (c,d).

Given a certain query, if two endpoints provide identical answers, we identify a positive relation between them (that we call *identity*) which is then translated into a *support* relation between such arguments. For instance, in Figure 2a both the French and the English DBpedia SPARQL endpoints provide *Italy* as answer to *Where is the Colosseum located?*. The algorithm assigns therefore a support relation between such arguments (double arrows in the figures), and merges them into a unique argument with value *Italy*. We do not consider these arguments as independent for two reasons: first, every other answer conflicting with one of them is also conflicting with all the others, and second, given that this argument is shared among several sources then it is highly acceptable, i.e., reliable. The confidence score of this new argument is calculated as the arctangent of the confidence scores of the endpoints providing such answer (max value = 1). We benefit from the *sameAs* links between the translation of the same word in DBpedia multilingual chapters, and we consider also the case reported in Figure 2b as *identity*, since both answers contain the same value expressed in different languages (i.e., *Italia* in Italian, and *Italy* in English).
Figure 2 (c,d) reports another positive relation between arguments, i.e., what we call *subsumption*. This case arises when one of the obtained answers is more specific than the other, both in terms of *i)* spacial/geographical relation, e.g.,

x is located in y (Figure 2d: *Rome* is located in *Italy*); *ii)* hyperonymy (*is_a* relation) as in Figure 2c (*Gibson_J-160E* is a *Guitar*: this example shows two possible answers to the question *Which instruments did John Lennon play?*). Our algorithm exploits external sources of semantic knowledge to detect the relations between the arguments (e.g., GeoNames[1] for geographical entities, and the DBpediaYAGO class hierarchy[2] [15]).

Once such relations are found and verified, the algorithm sets a *support* relation between the two arguments following the direction from the most specific argument toward the more general one (double arrows in Figure 2).

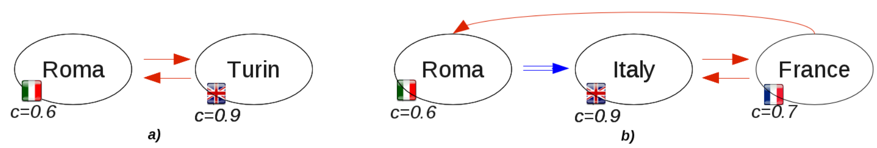

Fig. 3: Examples of attack relation: a) direct attack, b) additional attack.

In case the answers provided by two different endpoints for a certain query are different, and no subsumption relation between them is identified, then the algorithm assigns an *attack* relation between such arguments. For instance, an attack arises between the arguments visualized in Figure 3a, since the answers provided by the Italian and by the English DBpedia endpoints to the query concerning the location of the Colosseum are contradictory. It should be noted that the attack relation is always *bidirectional* in our framework (i.e., the two arguments conflict with each other, as shown by simple arrows in Figure 3).

A subtler case of attack is reported in Figure 3b. Here we have a support relation between the arguments *Roma* and *Italy* provided respectively by the Italian and the English endpoints (*subsumption* relation, as in the example visualized in Figure 2). Then, a third endpoint, i.e., the French one, provides the value *France* as answer to the same query. Our algorithm assigns an attack relation between this argument and the argument *Italy*, and it automatically adds an additional attack from *France* to the argument *Roma* which is supporting the argument it attacks (i.e., *Italy*). This additional attack results from the assumption that if an argument attacks *Italy*, it attacks also *Rome* since *Rome* is in *Italy*.

The last case we consider in our algorithm is when each endpoint provides a list of values (i.e., a *list*) as answer to a certain query (e.g., non-functional properties in DBpedia). For instance, in the example reported in Figure 4 the English DBpedia endpoint provides both *Guitar* and *Voice* as instruments played by J. Lennon, while the French endpoint provides both *Guitare* and *Harmonica*. In this case, our algorithm assigns a support relations between *Guitar* and *Guitare*, and it merges them in a unique value (see *identity* in Figure 2). Moreover, dif-

[1] `http://www.geonames.org/`

[2] `http://dbpedia.org/Downloads38#download-links-to-yago2`

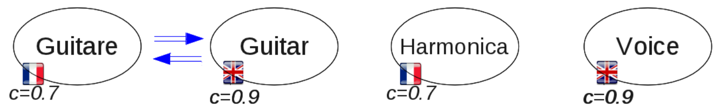

Fig. 4: Example of lists answers.

ferently from the case illustrated in Figure 3, in case of lists we do not consider arguments of the same list as conflictual. For this reason, we limit the assignment of the attack relation to those arguments which are not provided together in the same list. As illustrated in Figure 4, no attack relation is assigned between, e.g., *Harmonica* and *Guitar*, since both answers are provided by the French endpoint.

In this section we have described the algorithm that, given the answers provided by the endpoints to a single query, highlights the support/attack relation among them. Next section explains how such list of arguments and relationships is sent to the argumentation module that calculates the arguments' acceptability degree (i.e., the arguments that will be proposed to the user as more reliable).

3.2 Arguments evaluation and answer's motivation

Since we need to take into account the confidence associated to an information source when computing the set of accepted arguments and their own acceptability degree, we go beyond standard argument semantics introduced by Dung [7], and we rely on the computation of fuzzy confidence-based degrees of acceptability. In particular, the fuzzy labeling algorithm proposed by da Costa Pereira et al. [5] exploits a scenario where the arguments cannot be evaluated in the same way because of the confidence assigned to their source. In order to account for this fact and to consider also a positive, i.e., support, relation among the arguments, in addition to the attack relation used in [5] for the computation of the fuzzy labels of the arguments, in this paper we propose a bipolar fuzzy labeling algorithm. Let $\mathcal{A}$ be a fuzzy set of trustful arguments, and $\mathcal{A}(A)$ be the membership degree of argument A in $\mathcal{A}$, we have that $\mathcal{A}(A)$ is given by the trust degree of the most reliable (i.e., trusted) source that offers argument A, and it is defined as follows: $\mathcal{A}(A) = \max_{s \in \mathrm{src}(A)} \tau_s$ where τ_s is the degree to which source $s \in \mathrm{src}(A)$ is evaluated as reliable [5]. We follow da Costa Pereira et al. [5] in this choice, which implies an "optimistic" assignment of the labels. In case a pessimistic assignment is preferred, the min operator has to be used.

We now extend the definition of fuzzy labeling for standard abstract argumentation into a bipolar fuzzy labeling. Note that in this paper we assume that the following two constraints hold: an argument cannot attack and support at the same time another argument, and an argument cannot support an argument attacking it, and vice versa. These constrains are not on the computation of the acceptability degree of the arguments, but they underlie the construction of the argumentation framework itself.

Definition 1. *Let* $\langle \mathcal{A}, \rightarrow, \Rightarrow \rangle$ *be an abstract bipolar argumentation framework where* $\mathcal{A}$ *is a fuzzy set of (trustful) arguments,* $\rightarrow \subseteq \mathcal{A} \times \mathcal{A}$ *and* $\Rightarrow \subseteq \mathcal{A} \times \mathcal{A}$ *are two binary relations called attack and support, respectively. A bipolar fuzzy labeling is a total function* $\alpha : \mathcal{A} \rightarrow [0, 1]$.

Such an α may also be regarded as (the membership function of) the fuzzy set of acceptable arguments where the label $\alpha(A) = 0$ means that the argument is outright unacceptable, and $\alpha(A) = 1$ means the argument is fully acceptable. All cases inbetween provide the degree of the acceptability of the arguments which may be considered accepted at the end, if they overcome a given threshold. In [5], the acceptability of an argument cannot be greater than the degree to which the arguments attacking it are unacceptable: $\alpha(A) \leq 1 - \max_{B:B \rightarrow A} \alpha(B)$. This constraint is reformulated in bipolar fuzzy labeling as follows: the acceptability of an argument cannot be greater than the degree to which the arguments attacking it are unacceptable *unless* there exists at least one argument supporting it.

The rationale behind this constraint is the following: if the set of supports is empty, then the standard fuzzy labeling is used; otherwise, the support of an argument towards another is intended to augment the acceptability degree of such supported argument. As we will show by means of examples, the acceptability degree of an argument which is both supported and attacked depends on the confidence assigned to the sources proposing the supporter and the attacker of such argument. Using the above constraint, we obtain the following definition

Definition 2. *(Bipolar Fuzzy Labeling) Let* α *be a bipolar fuzzy labeling. We say that* α *is a bipolar fuzzy labeling iff, for all arguments* A, $\alpha(A) = \mathrm{avg}\{\min\{\mathcal{A}(A), 1 - \max_{B:B \rightarrow A} \alpha(B)\}; \max_{C:C \Rightarrow A} \alpha(C)\}$.

da Costa Pereira et al. [5] show that the convergence speed of the labeling algorithm is linear (as their proof of convergence suggests) since in practice a small number of iterations is enough to get so close to the limit that the error is less than the precision with which the membership degrees are represented in the computer. The bipolar fuzzy labeling algorithm presented here is a variant of the one developed in [5] thus on account of this fact, the computational feasibility result can be imported here. Table 1 reports the iterations performed by the algorithm to assign the labels to the arguments of a BAF.

When the argumentation module receives the couples of arguments linked by the appropriate relation and the degree of confidence associated to each source, the bipolar fuzzy labeling algorithm is raised on the argumentation framework to obtain the acceptability degree of each argument. This step returns also the overall bipolar argumentation framework where each argument is linked to its source, and the acceptability degree is associated to the arguments. This overall view is then used to explain to the user how the QA system comes to find this answer, thus its *motivations*. This step is necessary to prevent the user from seeing the QA system as a black box, but to understand the reasons behind correct, and more importantly, erroneous answers.

The fact that an argumentation framework can be used to provide explanations [1] is one of the reasons behind the choice of this formalism to detect inconsistencies. Other possible solutions to rank a set of information items would be (possible weighted) voting mechanisms, where the preferences of some voters, i.e., the most reliable information sources, carry more weight than the preferences of other voters. We choose to rely on argumentation-based inconsistency detection instead of adopting a voting system to rank the answers provided by the QA system because the latter does not consider the presence of semantic relations (of positive or negative nature) among the items within the list. The additional value of an argumentation-based approach is the graph-based visualization of the answers.

Table 1: BAF: $A \rightarrow B, B \rightarrow C, C \rightarrow A, D \Rightarrow C$

t	$\alpha_t(A)$	$\alpha_t(B)$	$\alpha_t(C)$	$\alpha_t(D)$
0	1	0.4	0.2	1
1	0.9	0.2	**0.6**	**1**
2	0.65	0.15	↓	↓
3	0.52	0.25		
4	0.46	0.36		
5	0.43	**0.4**		
6	0.41	↓		
7	**0.4**			
8	↓			

4 Framework integration in a QA system and evaluation

To evaluate the argumentation-based framework to detect inconsistencies that we described in Section 3 in a real setting, we integrate it in an existing Question-Answering system, QAKiS [3] (Section 4.1). QAKiS allows to query multilingual chapters of DBpedia as RDF data sets using a natural language interface, and its architecture can be flexibly modified to account for the proposed extension.

4.1 QA system

QAKiS[3] (Question Answering wiKiFramework-based System) [3] addresses the task of QA over structured knowledge-bases (e.g., DBpedia), where the relevant information is expressed also in unstructured forms (e.g., Wikipedia pages). It implements a relation-based match for question interpretation, to convert the user question into a query language (e.g., SPARQL). More specifically, it makes use of relational patterns (automatically extracted from Wikipedia and collected in the WikiFramework repository [3]), that capture different ways to express a certain relation in a given language. QAKiS is composed of four main modules (Figure 5): *i)* the **query generator** takes the user question as input, generates the typed questions, and then generates the SPARQL queries from the retrieved patterns; *ii)* the **Named Entity (NE) Recognizer**; *iii)* the **pattern matcher** takes as input a typed question, and retrieves the patterns (among those in the repository) matching it with the highest similarity; and *vi)* the **sparql package** handles the queries to send to multiple multilingual DBpedia endpoints [2]. The actual version of QAKiS targets questions containing a Named Entity related

[3] `http://dbpedia.inria.fr/qakis/`

to the answer through one property of the ontology, as *Which river does the Brooklyn Bridge cross?*. Such questions match a single pattern, i.e., one relation.

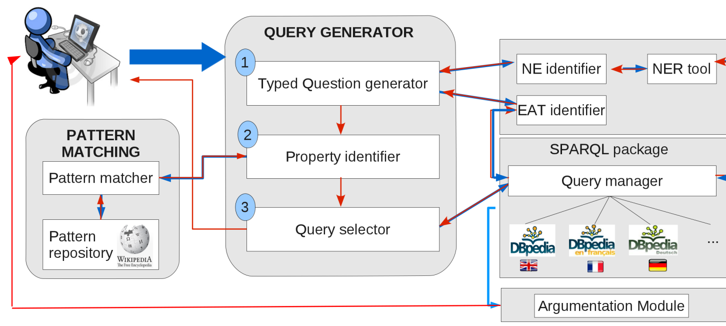

Fig. 5: QAKiS workflow

We embed into the QAKiS architecture the *argumentation module* (Figure 5), that collects the answers obtained from the different endpoints and applies the algorithm described in Section 3.1 to identify the arguments and the relations among them, and then it runs the bipolar fuzzy labelling algorithm (Section 3.2) to detect inconsistencies and return the more reliable arguments.

4.2 Data set

To run our experiments, we extract from the reference data set of QALD-2 (Question Answering over Linked Data challenge)[4] the questions that the current version of QAKiS is built to address (i.e. questions containing a NE related to the answer through one property of the ontology), corresponding to 26 questions in the training and 32 in the test sets. The discarded questions require either some forms of reasoning (e.g., counting or ordering) on data, aggregation (from datasets different from DBpedia), involve n-relations, or are boolean questions. We consider these 58 questions as the reference data set for our experiments.

4.3 Results and error analysis

To evaluate the validity of the proposed approach, we run the questions contained into our reference datasets on the English, German and French chapters of DBpedia (confidence scores: 0.7, 0.6 and 0.5 respectively). We chose these three chapters based on *i)* their dimensions and the robustness of their endpoints, and *ii)* the presence of the *SameAs* relation among the translations of the words in the different languages. Since the questions of QALD-2 dataset were

[4] `http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=challenge&q=2`

created to query the English chapter of DBpedia only, it turned out that only in 25 cases out of 58 at least two endpoints provide an answer (in all the other cases the answer is provided by the English chapter only, not useful for our purposes). For instance, given the question *List the children of Margaret Thatcher* the English DBpedia provides *Mark* and *Carol Thatcher* as answers, while the French one provides only the answer *Mark Thatcher*. Or given the question *How many employees does IBM have?*, the English and the German DBpedia provide 426751 as answer, while the French DBpedia provides 433362 and 2010.

We evaluate our approach with two sets of experiments: in the first case, we start from the answers provided by the different DBpedia endpoints to the 25 questions, and we run our argumentation-based algorithm on it. In the second case, we add QAKiS in the loop, meaning that the data we use as input for the argumentation module are directly produced by the system, as explained in Section 4.1. In this second case, the input are the 25 natural language questions. Table 2 reports the results we obtained for the two experiments over 24 questions, one question timed out. We evaluated both the ability of the argumentation module to correctly identify and (if necessary merge) the answers from the different endpoints (1st row), and its ability to assign the correct relations among such arguments (2nd row), w.r.t. a manually annotated goldstandard of arguments and their relations (in total, over the 25 questions, 90 items should be recognized as arguments-answers, and 219 attack and 3 support relations should be detected). For instance, given the examples before, for the question *How many employees does IBM have?* the algorithm generates 4 arguments (where the answer 433362 provided by both the English and German DBpedia is merged and its confidence score augmented, see Section 3) and 12 relations (attacks among the different values). Applying the fuzzy labeling algorithm, the answer provided with the highest confidence is 433362 (in future work, we could allow some approximation, merging values $\sim$433000 to this value). For *List the children of Margaret Thatcher*, the algorithm identifies two arguments (i.e. *Mark* and *Carol*), and no relations, since the DBpedia property *child* allows for list answers. Both answers are provided to the user, where *Mark* has a highest confidence score.

Most of the errors in identifying the arguments are due to the missing *SameAs* links in DBpedia: the algorithm is therefore not able to merge translations of the same answer, considering them as different. Wrong relation assignments are mainly due to missing attacks among arguments (in particular for numerical values, or for arguments from the same endpoint). Concerning the second column of Table 2, since QAKiS performances are about $\sim$50%, the results are obtained accordingly (the argumentation-based module is biased by QAKiS mistakes in submitting the query to DBpedia). The average computation cost of the argumentation-based algorithm is high ($\sim$124 seconds), mainly due to the n-answers questions. Considering only 1-answer question, the computation cost drops to 5 seconds. The complexity is quadratic, at least one SPARQL query is sent for each couple of answers. This could be improved by importing data for each answers individually and then process it locally. We are currently investigating this solution to optimize the algorithm.

Task	Argumentation module			QAKiS+Argum. module		
	Precis.	*Recall*	*F-meas.*	*Precis.*	*Recall*	*F-meas.*
Argument identification	0.95	1	**0.97**	0.71	0.75	**0.73**
Relation assignment	0.71	0.73	**0.72**	0.54	0.56	**0.55**

Table 2: Performances over 24 questions from QALD-2 test set

5 Related work

State of the art QA systems over Linked Data generally address the issue of question interpretation mapping a natural language question to a triple-based representation (see [11] for an overview). Moreover, they examine the potential of open user friendly interfaces for the SW to support end users in reusing and querying the SW content. None of these systems provides a mechanism to detect the inconsistencies among the set of items composing the answers, and none of them allows the user to understand the reasons behind the retrieved answer.

Several works address alignment agreement based on argumentation theory. More precisely, Laera et al. [10] address alignment agreement relying on argumentation to deal with the arguments which attack or support the candidate correspondences among ontologies. Doran et al. [6] propose a methodology to identify subparts of ontologies which are evaluated as sufficient for reaching an agreement, before the argumentation step takes place, and dos Santos and Euzenat [14] present a model for detecting inconsistencies in the selected sets of correspondences relating ontologies. In particular, the model detects logical and argumentation inconsistency to avoid inconsistencies in the agreed alignment. The framework we propose has common points with this line of works, i.e., the use of argumentation theory to select a consistent set of information items, but the scenario in which the two approaches are exploited is different and this leads to a different addressed issues and proposed solutions.

6 Conclusions

In this paper, we propose an automatic framework to detect the inconsistencies which may arise into a set of answers provided by a QA system over linked data. These inconsistencies are due to i) heterogeneous answers provided by the same information source, e.g., a single SPARQL endpoint, or ii) heterogeneous answers from distributed information sources. Our framework adopts bipolar abstract argumentation to provide an overall view of the information items provided by the single sources. This argumentation framework is built thank to the semantic relations among the information items automatically extracted using ontological knowledge, e.g., DBpedia ontology. The framework is then evaluated to assign the acceptability degree of the arguments using the bipolar fuzzy labeling algorithm. We have shown the feasibility of our approach on the QAKiS question answering system on the multilingual chapters of DBpedia.

There are several points to be addressed as future work. First, we assign to each information source a confidence degree which is derived from the coverage of such DBpedia version. However, another possibility is to leave the user itself to assign the confidence degree to the information sources. We plan to embed this feature in the QAKiS interface. Second, we have to extend the set of ontologies we consider in order to be able to detect further relations (positive and negative) among the information items. Finally, the user evaluation should not be underestimated and we will perform an evaluation campaign to verify which kind of visualization of the motivations is more usable by consumers.

References

1. Bench-Capon, T.J.M., Lowes, D., McEnery, A.M.: Argument-based explanation of logic programs. Knowl.-Based Syst. 4(3), 177–183 (1991)
2. Cabrio, E., Cojan, J., , Gandon, F., Hallili, A.: Querying multilingual DBpedia with QAKiS. In: Procs ESWC 2013 (demo paper) (2013)
3. Cabrio, E., Cojan, J., Aprosio, A.P., Magnini, B., Lavelli, A., Gandon, F.: Qakis: an open domain qa system based on relational patterns. In: Procs of ISWC 2012 (Posters & Demos). vol. 914 (2012)
4. Cayrol, C., Lagasquie-Schiex, M.C.: Bipolarity in argumentation graphs: Towards a better understanding. In: Procs of SUM 2011, LNCS, vol. 6929, pp. 137–148. Springer (2011)
5. da Costa Pereira, C., Tettamanzi, A., Villata, S.: Changing one's mind: Erase or rewind? In: Procs of IJCAI 2011. pp. 164–171. IJCAI/AAAI (2011)
6. Doran, P., Tamma, V., Palmisano, I., Payne, T.: Efficient argumentation over ontology correspondences. In: Procs of AAMAS 2009. pp. 1241–1242 (2009)
7. Dung, P.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321–358 (1995)
8. Dunne, P.E., Hunter, A., McBurney, P., Parsons, S., Wooldridge, M.: Weighted argument systems: Basic definitions, algorithms, and complexity results. Artif. Intell. 175(2), 457–486 (2011)
9. Hunter, A.: A probabilistic approach to modelling uncertain logical arguments. Int. J. Approx. Reasoning 54(1), 47–81 (2013)
10. Laera, L., Blacoe, I., Tamma, V., Payne, T., Euzenat, J., Bench-Capon, T.: Argumentation over ontology correspondences in MAS. In: Procs of AAMAS 2007. pp. 1–8 (2007)
11. Lopez, V., Uren, V.S., Sabou, M., Motta, E.: Is question answering fit for the semantic web?: A survey. Semantic Web 2(2), 125–155 (2011)
12. Mendes, P., Jakob, M., Bizer, C.: DBpedia: A multilingual cross-domain knowledge base. In: Procs of LREC 2012. ELRA (2012)
13. Mendes, P.N., Mühleisen, H., Bizer, C.: Sieve: linked data quality assessment and fusion. In: Procs of the Joint EDBT/ICDT Workshops. pp. 116–123. ACM (2012)
14. dos Santos, C.T., Euzenat, J.: Consistency-driven argumentation for alignment agreement. In: Procs of OM 2010, CEUR Workshop Proceedings 689 (2010)
15. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Procs of WWW 2007. pp. 697–706. ACM (2007)