A RADAR for Information Reconciliation in Question Answering Systems over Linked Data [1]

Elena Cabrio [a,*], Serena Villata [b] and Alessio Palmero Aprosio [c]

[a] *University of Nice Sophia Antipolis, 2000 Route des Lucioles BP93 06902, Sophia Antipolis - France*
E-mail: elena.cabrio@unice.fr
[b] *CNRS - I3S laboratory, 2000 Route des Lucioles BP93 06902, Sophia Antipolis - France*
E-mail: villata@i3s.unice.fr
[c] *Fondazione Bruno Kessler, Via Sommarive, 18, 38123 Povo (TN) - Italy*
E-mail: aprosio@fbk.eu

Abstract. In the latest years, more and more structured data is published on the Web and the need to support typical Web users to access this body of information has become of crucial importance. Question Answering systems over Linked Data try to address this need by allowing users to query Linked Data using natural language. These systems may query at the same time different heterogenous interlinked data sets, that may provide different results for the same query. The obtained results can be related by a wide range of heterogenous relations, e.g., one can be the specification of the other, an acronym of the other, etc. In other cases, such results can contain an inconsistent set of information about the same topic. A well known example of such heterogenous interlinked data sets are language-specific DBpedia chapters, where the same information may be reported in different languages. Given the growing importance of multilingualism in the Semantic Web community, and in Question Answering over Linked Data in particular, we choose to apply information reconciliation to this scenario. In this paper, we address the issue of reconciling information obtained by querying the SPARQL endpoints of language-specific DBpedia chapters. Starting from a categorization of the possible relations among the resulting instances, we provide a framework to: *(i)* classify such relations, *(ii)* reconcile information using argumentation theory, *(iii)* rank the alternative results depending on the confidence of the source in case of inconsistencies, and *(iv)* explain the reasons underlying the proposed ranking. We release the resource obtained applying our framework to a set of language-specific DBpedia chapters, and we integrate such framework in the Question Answering system QAKiS, that exploits such chapters as RDF data sets to be queried using a natural language interface.

Keywords: Question Answering over Linked Data, Information reconciliation, Language-specific DBpedia chapters, Answers justification, Argumentation Theory

---

[1]This paper is an extended version of the paper "Argumentation-based Inconsistencies Detection for Question-Answering over DBpedia" published at the Workshop NL&DBpedia-2013 and of the extended abstracts titled "Reconciling Information in DBpedia through a Question Answering System" and "Hunting for Inconsistencies in Multilingual DBpedia with QAKiS" published at the Posters & Demonstrations Tracks of ISWC-2013 and of ISWC-2014, respectively. The improvements of the present paper with respect to the previous versions are detailed in the Related Work section.

*Corresponding author. E-mail: elena.cabrio@unice.fr

1. Introduction

In the Web of Data, it is possible to retrieve heterogeneous 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. It is not always the case that these results are identical, it may happen that they conflict with each other, or they may be linked by some other relation like a specification. The automated detection of the kind of relationship holding between different instances about a single object with the goal of reconciling them is an open problem for consuming information in the Web of Data. In particular, this problem arises while querying the language-specific chapters of DBpedia [24]. 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 language-specific DBpedia SPARQL endpoints with the same query, the answers we collect can be either identical, or in some kind of specification relation, or they can be contradictory. Consider for instance the following example: we query a set of language-specific DBpedia chapters about *How tall is the soccer player Stefano Tacconi?*, receiving the following information: 1.88 from the Italian chapter and the German one, 1.93 from the French chapter, and 1.90 from the English one. How can I know what is the "correct" (or better, the more reliable) information, knowing that the height of a person is unique? Addressing such a kind of issues is the goal of the present paper. More precisely, in this paper, we answer the research question:

– How to reconcile information provided by the language-specific chapters of DBpedia?

This open issue is particularly relevant to Question Answering (QA) systems over DBpedia [23], where the user expects a unique (possibly correct) answer to her factual natural language question. A QA system querying different data sources needs to weight them in an appropriate way to evaluate accordingly the information items they provide. In this scenario, another open problem is how to explain and justify the answer the system provides to the user in such a way that the overall QA system appears transparent, and as a consequence, more reliable. Thus, our research question breaks down into the following subquestions:

1. How to automatically detect the relationships holding between information items returned by different language-specific chapters of DBpedia?

2. How to compute the reliability degree of such information items to provide a unique answer?
3. How to justify and explain the answer the QA system returns to the user?

First, we need to classify the relations connecting each piece of information to the others returned by the different data sources, i.e., the SPARQL endpoints of the language-specific DBpedia chapters. We adopt the categorization of the relations existing between different information items retrieved with a unique SPARQL query proposed by Cabrio et al. [13]. Up to our knowledge, this is the only available categorization that considers linguistic-based fine-grained relations among the information items returned by language-specific DBpedia chapters, given a certain query. This categorization considers ten *positive* relations among heterogenous information items (referring to widely accepted linguistic categories in the literature), and three *negative* relations meaning inconsistency. Starting from this categorization, we propose the RADAR (ReconciliAtion of Dbpedia through ARgumentation) framework, that adopts a classification method to return the relation holding between two information items. This first step results in a graph-based representation of the results set where each information item is a node, and edges represent the identified relations.

Second, we adopt *argumentation theory* [18], a suitable technique for reasoning about conflicting information, to assess the acceptability degree of the information items, depending on the relation holding between them and the trustworthiness of their information source [15]. Roughly, an abstract argumentation framework is a directed labeled graph whose nodes are the arguments and the edges represent a *conflict* relation. Since positive relations among the arguments may hold as well, we rely on bipolar argumentation [14] that considers also a *positive* support relation.

Third, the graph of the results set obtained after the classification step, together with the acceptability degree of each information item obtained after the argumentation step, is used to justify and explain the resulting information ranking (i.e., the order in which the answers are returned to the user).

We evaluate our approach as standalone (i.e., over a set of heterogeneous values extracted from a set of language-specific DBpedia chapters), and through its integration in the QA system QAKiS [9], that exploits language-specific DBpedia chapters as RDF data sets to be queried using a natural language interface. The reconciliation module is embedded to provide a (possi-

bly unique) answer whose acceptability degree is over a given threshold, and the graph structure linking the different answers highlights the underlying justification. Moreover, RADAR is applied over 300 DBpedia properties in 15 languages, and the obtained resource of reconciled DBpedia language-specific chapters values is released.

Even if information reconciliation is a way to enhance linked data quality, this paper does not address the issue of linked data quality assessment and fusion [25,7], nor ontology alignment. Finally, argumentation theory in this paper is not exploited to find agreements over ontology alignments [17]. Note that our approach is intended to reconcile and explain the answer of the system to the user. We do not think that such a kind of explanations would be possible with alignment, but we do not claim that our solution is better. Exploring the alignment-based solution is an open future work. In our paper, we have addressed the open problem of reconciling and explaining a results set from language-specific DBpedia chapters first using well known conflict detection and explanation techniques, i.e., argumentation.

We are not aware of any other available QA system that queries several information sources (in our case language-specific chapters of DBpedia) and then it is able to verify the coherence of the proposed results set, and show *why* a certain answer has been provided. The merit of the present paper is to describe the proposed framework (i.e., RADAR 2.0) with the addition of an extensive evaluation over standard QA data sets.

In the remainder of the paper, Section 2 presents our reconciliation framework for language-specific DBpedia chapters, Section 3 reports on the experiments run over DBpedia to evaluate it, and Section 4 describes its integration in QAKiS. Section 5 reports on the related work. Finally, some conclusions are drawn.

2. RADAR 2.0: a Framework for Information Reconciliation

The RADAR 2.0 (ReconciliAtion of Dbpedia through ARgumentation) framework for information reconciliation is composed by three main modules (see Figure 1). It takes as input a collection of results from the same SPARQL query raised against the SPARQL endpoints of the language-specific DBpedia chapters (Section 3 provides more details about the chapters considered in our experimental setting). Given such results set, RADAR retrieves two kinds of information:

(i) the sources proposing each particular element of the results set, and *(ii)* the elements of the results set themselves. The first module of RADAR (module A, Figure 1) takes each information source, and following two different heuristics, assigns a *confidence degree* to the source. Such confidence degree will affect the reconciliation in particular with respect to the possible inconsistencies: information proposed by the more reliable source will obtain a higher acceptability degree. The second module of RADAR (module B, Figure 1) instead starts from the results set, and it matches every element with all the other returned elements, detecting the kind of relation holding between these two elements. The result of such module is a graph composed by the elements of the results set connected with each other by the relations of our categorization. Both the sources associated with a confidence score and the results set under the form of a graph are then provided to the third module of RADAR, the argumentation one (module C, Figure 1). The aim of such module is to reconcile the results set. The module considers all positive relations as a *support* relation and all negative relations as an *attack* relation, building a bipolar argumentation graph where each element of the results set is seen as an argument. Finally, adopting a bipolar fuzzy labeling algorithm relying on the confidence of the sources to decide the acceptability of the information, the module returns the acceptability degree of each argument, i.e., element of the results set. The output of the RADAR framework is twofold. First, it returns the acceptable elements (a threshold is adopted), and second the graph of the results set is provided, where each element is connected to the others by the identified relations (i.e., the explanation about the choice of the acceptable arguments returned).

In the remainder of this section, we will describe how the confidence score of the sources is computed (Section 2.1), and we will summarize the adopted categorization detailing how such relations are automatically extracted (Section 2.2). Finally, the argumentation module is described in Section 2.3.

2.1. Assigning a confidence score to the source

Language-specific DBpedia chapters can contain different information from one language to another, providing more specificity on certain topics, or filling information gaps. Moreover, the knowledge of certain instances and the conceptualization of certain relations can be culturally biased. For instance, we expect to have more precise (and possibly more reliable) infor-
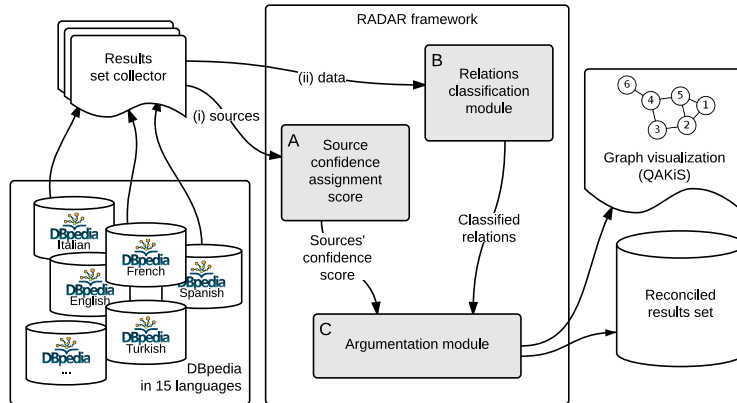
Fig. 1. RADAR 2.0 framework architecture.

mation on the Italian actor A. Albanese on the Italian DBpedia, than on the English or on the French ones.

To trust and reward the data sources, we need to calculate the reliability of the source with respect to the contained information items. In [11], an apriori confidence score is assigned to the endpoints according to their dimensions and solidity in terms of maintenance (the English chapter is assumed to be more reliable than the others on all values, but this is not always the case). RADAR 2.0 assigns, instead, a confidence score to the DBpedia language-specific chapter depending on the queried entity, according to the following two criteria:

– *Wikipedia page length.* The chapter of the longest language-specific Wikipedia page describing the queried entity is rewarded with respect to the others, following [6] that demonstrates that the article length is a very good predictor of its precision. The length is calculated on the Wikipedia dump of the considered language (# of characters in the text, ignoring image tags and tables). The longest page is assigned a score equal to 1, and a proportional score is assigned to the other chapters.
– *Entity geo-localization.* The chapter of the language spoken in the places linked to the page of the entity is rewarded with respect to the others (assuming that if an entity belongs to a certain place or is frequently referred to it, it is more likely that the DBpedia chapter of such country contains updated and reliable information). All Wikipedia page hyperlinks are considered, and their presence in GeoNames[1] is checked. If exist-

ing, the prevalent language in the place (following the GeoNames matching country-language[2]) is extracted, and to the corresponding chapter is assigned a score equal to 1. A proportional score is then assigned to the other chapters (i.e. if an entity has e.g. 10 links to places in Italy and 2 to places in Germany, the score assigned to the Italian DBpedia chapter is 1, while for the German chapter is 0.2).

Such metrics (whose appropriateness for our purposes has been tested on the development set, see Section 3.1) are then summed, and normalized with a score ranging from 0 to 1, where 0 is the less reliable chapter for a certain entity and 1 is the most reliable one. The obtained scores are then considered by the argumentation module (Section 2.3) for information reconciliation.

2.2. Relations classification

Cabrio et al. [13] propose a classification of the semantic relations holding among the different values obtained by querying a set of language-specific DBpedia chapters with a certain query. More precisely, such categories correspond to the linguistic phenomena (mainly discourse and lexical semantics) holding among heterogeneous values obtained querying two DBpedia chapters at a time, given a subject and an ontological property. In the following, we list the positive relations between values resulting from the data-

---

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

[2] Such table connecting a country with its language can be found here http://download.geonames.org/export/dump/countryInfo.txt.

driven study in [13]. Then, in parallel, we describe how RADAR 2.0 addresses the automatic classification of such relations.

identity i.e., same value but in different languages (missing `SameAs` link in DBpedia).
E.g., `Dairy product` vs `Produits laitiers`
acronym i.e., initial components in a phrase or a word. E.g., `PSDB` vs `Partito della Social Democrazia Brasiliana`
disambiguated entity i.e., a value contains in the name the class of the entity. E.g., `Michael Lewis (Author)` vs `Michael Lewis`
coreference i.e., an expression referring to another expression describing the same thing (in particular, non normalized expressions). E.g., `William Burroughs` vs `William S. Burroughs`

Given the high similarity among the relations belonging to these categories, we cluster them into a unique category called *surface variants* of the same entity. Given two entities, RADAR automatically detects the *surface variants* relation among them, if one of the following strategies is applicable: cross-lingual links[3], text identity (i.e. string matching), Wiki redirection and disambiguation pages.

geo-specification i.e., ontological geographical knowledge. E.g., `Queensland` vs `Australia`
renaming i.e., reformulation of the same entity name in time. E.g., `Edo`, old name of `Tokyo`

Given the way in which *renaming* has been defined in [13], it refers only to geographical renaming. For this reason, we merge it to the category *geo-specification*. RADAR classifies a relation among two entities as falling inside this category when in the GeoNames one entity is contained in the other one (*geo-specification* is a directional relation between two entities). We also consider the alternative names gazette included in GeoNames, and geographical information extracted from a set of English Wikipedia

infoboxes, such as `Infobox former country`[4] or `Infobox settlement`.

meronymy i.e., a constituent part of, or a member of something. E.g., `Justicialist Party` is a part of `Front for Victory`
hyponymy i.e., relation between a specific and a general word when the former is included within the latter. E.g., `alluminio` vs `metal`
metonymy i.e., a thing/concept not called by its own name, but by the name of something intimately associated with that thing/concept. E.g., `Joseph Hanna` vs `Hanna-Barbera`
identity:stage name i.e., pen/stage names pointing to the same entity. E.g., `Lemony Snicket` vs `Daniel Handler`

We cluster such semantic relations into a category called *inclusion*.[5] To detect this category of relations, RADAR exploits a set of features extracted from:

MusicBrainz[6] to detect when a musician plays in a band, and when a label is owned by a bigger label.
BNCF (Biblioteca Nazionale Centrale di Firenze) Thesaurus[7] for the broader term relation between common names.
DBpedia, in particular the data sets connecting Wikipedia, GeoNames and MusicBrainz trough the `sameAs` relation.
WikiData for the *part of*, *subclass of* and *instance of* relations. It contains links to GeoNames, BNCF and MusicBrainz, integrating DBpedia `sameAs`.
Wikipedia contains hierarchical information in: infoboxes (e.g. property `parent` for companies, `product` for goods, `alter ego` for biographies), in categories (e.g., `Gibson guitars`), "see also" sections and links in the first sentence (e.g., *Skype was acquired by [United States]-based [Microsoft Corporation]*).

---

[3]Based on WikiData, a free knowledge base that can be read and edited by humans and machines alike `http://www.wikidata.org/`, where data entered in any language is immediately available in all other languages. In WikiData, each entity has the same ID in all languages for which a Wikipedia page exists, allowing us to overcome the problem of missing `SameAs` links in DBpedia (that was an issue in DBpedia versions prior to 3.9). Moreover, WikiData is constantly updated (we use April 2014 release).

[4]For instance, we extract the property "today" connecting historical entity names with the current ones (reconcilable with GeoNames). We used Wikipedia dumps.

[5]Royo [26] defines both relations of *meronymy* and *hyponymy* as relations of *inclusion*, although they differ in the kind of inclusion defined (hyponymy is a relation of the kind "B is a type of A", while meronymy relates a whole with its different parts or members). Slightly extending Royo's definition, we joined to this category also the relation of *metonymy*, a figure of speech scarcely detectable by automatic systems due to its complexity (and *stage name*, that can be considered as a particular case of *metonymy*, i.e., the name of the character for the person herself).

[6]`http://musicbrainz.org/`
[7]`http://thes.bncf.firenze.sbn.it/`

Inclusion is a directional relation between two entities (the rules we apply to detect *meronymy*, *hyponymy* and *stage name* allow us to track the direction of the relation, i.e. if $a \rightarrow b$, or $b \rightarrow a$).

Moreover, in the classification proposed in [13], the following negative relations (i.e., values mismatches) among possibly inconsistent data are identified:

text mismatch i.e. unrelated entity. E.g., `Palermo` vs `Modene`

date mismatch i.e. different date for the same event. E.g., `1215-04-25` vs `1214-04-25`

numerical mismatch i.e. different numerical values. E.g., `1.91` vs `1.8`

RADAR labels a relation between entities/objects as negative, if every attempt to find one of the positive relations described above fails (i.e., negation as a failure). For numerical values, a *numerical mismatch* identifies different values.[8]

The reader may argue that a machine learning approach could have been applied to this task, but a supervised approach would have required an annotated data set to learn the features. Unfortunately, at the moment there is no such a training set available to the research community. Moreover, given the fact that our goal is to produce a resource as precise as possible for future reuse, the implementation of a rule-based approach allows us to tune RADAR to reward precision in our experiments, so that to accomplish our purpose.

2.3. Argumentation-based information reconciliation

This section begins with a brief overview of abstract argumentation theory, and then we detail RADAR 2.0 argumentation module.

An abstract argumentation framework (AF) [18] aims at representing conflicts among elements called *arguments*, whose role is determined only by their relation with the other arguments. An AF encodes, through the conflict (i.e., *attack*) relation, the existing conflicts within a set of arguments. It is then interesting to identify the conflict outcomes, which, roughly speaking, means determining which arguments should be accepted, and which arguments should be rejected, according to some reasonable criterion.

---

[8]At the moment no tolerance is admitted, if e.g. the height of a person differs of few millimeters in two DBpedia chapters, the relation is labeled as *numerical mismatch*. We plan to add such tolerance for information reconciliation as future work.

The set of accepted arguments of an argumentation framework consists of a set of arguments that does not contain an argument conflicting with another argument in the set. Dung [18] presents several acceptability semantics that produce zero, one, or several *consistent* sets of accepted arguments. 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 2.a shows an example of AF. The arguments are visualized as nodes of the argumentation graph, and the attack relation is visualized as edges. Gray arguments are the accepted ones. Using Dung's acceptability admissibility-based semantics [18], the set of accepted arguments is $\{b, c\}$. For more details about acceptability semantics, we refer the reader to Baroni at al. [2].

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 [19,15,20]. In particular, da Costa Pereira et al. [15] have proposed 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 a fuzzy labeling for argument A as $\alpha(A) = \min\{\mathcal{A}(A), 1 - \max_{B:B \rightarrow A} \alpha(B)\}$ where $\mathcal{A}(A)$ is given by the trust degree of the most reliable source that offers argument A, and argument B is an argument attacking A. We say that $\alpha(A)$ is the fuzzy label of argument A. Consider the example in Figure 2.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 $\alpha(c) = \alpha(b) = 0.8$.

Since we want to take into account the confidence associated to the information sources to compute the acceptability degree of arguments, we rely on the computation of fuzzy confidence-based degrees of acceptability. As the fuzzy labeling algorithm [15] exploits a scenario where the arguments are connected by an attack relation only, in Cabrio et al. [11] we have proposed a bipolar version of this algorithm, to consider also a positive, i.e., support, relation among the arguments (bipolar AFs) for the computation of the fuzzy labels of the arguments.

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
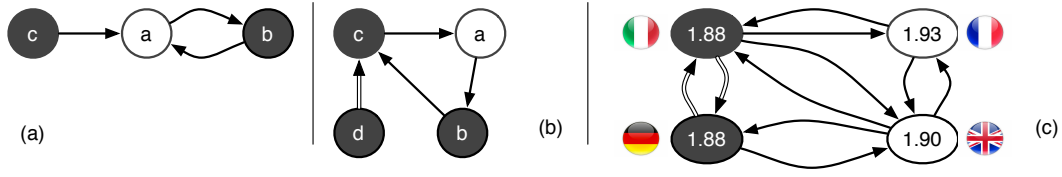
Fig. 2. Example of (a) AF, (b) a bipolar AF, and (c) example provided in the introduction modeled as a bipolar AF.

reliable (i.e., trusted) source that offers argument A[9], 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. The starting confidence degree associated to the sources is provided by RADAR's first module. The bipolar fuzzy labeling algorithm [11] assumes that the following two constraints hold: *(i)* an argument cannot attack and support another argument at the same time, and *(ii)* an argument cannot support an argument attacking it, and vice versa. These constraints underlie the construction of the bipolar AF itself. In the following, the attack relation is represented with $\rightarrow$, and the support relation with $\Rightarrow$.

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.

A bipolar fuzzy labeling is defined as follows[10], where argument B is an argument attacking A and C is an argument supporting A:

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

---

[9]We follow da Costa Pereira et al. [15] choosing the max operator ("optimistic" assignment of the labels), but the min operator may be preferred for a pessimistic assignment.

[10]For more details about the bipolar fuzzy labeling algorithm, see Cabrio et al. [11].

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	↓			

When the argumentation module receives the elements of the results set linked by the appropriate relation and the confidence degree associated to each source, the bipolar fuzzy labeling algorithm is raised on the argumentation framework to obtain the acceptability degree of each argument. Consider the example in Figure 2.b, if we have $\mathcal{A}(a) = \mathcal{A}(d) = 1$, $\mathcal{A}(b) = 0.4$ and $\mathcal{A}(c) = 0.2$, then the fuzzy labeling algorithm returns the following labels: $\alpha(a) = \alpha(b) = 0.4$, $\alpha(c) = 0.6$, and $\alpha(d) = 1$. The step by step computation of the labels is shown in Table 1. Figure 2.c shows how the example provided in the introduction is modeled as a bipolar argumentation framework, where we expect to have the Italian DBpedia chapter as the most reliable one being Stefano Tacconi an Italian soccer player. The result returned by the bipolar argumentation framework is that the "correct" answer is 1.88. A more precise instantiation of this example in the QA system is shown in the next section.

The fact that an argumentation framework can be used to provide an explanation and justify positions is witnessed by a number of applications in different contexts [3], like for instance practical reasoning [27], legal reasoning [4,5], medical diagnosis [21]. This is the reason why we choose this formalism to reconcile information, compute the set of reliable information items, and finally justify this result. Other possible solutions would be (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. However, voting mechanisms do not consider the presence of (positive and negative) relations among the items within the list, and no justification beyond the basic trustworthiness of the sources is provided to motivate the ranking of the information items.

Notice that argumentation is needed in our use case because we have to take into account the trustworthiness of the information sources, and it provides an explanation of the ranking, which is not possible with simple majority voting. Argumentation theory, used as a conflict detection technique, allows us to detect inconsistencies and consider the trustworthiness evaluation of the information sources, as well when proposing a single answer to the users. As far as we know, RADAR integrated in QAKiS is the first example of QA over linked data system coping with this problem and providing a solution. Simpler methods would not allow to cover both aspects mentioned above. We use bipolar argumentation instead of non bipolar because we have not only the negative conflict relation but also the positive support relation among the elements of the results set.

3. RADAR experimental setting and evaluation

In this section, we describe the data set on which we evaluate the RADAR framework (Section 3.1), and we discuss the obtained results (Section 3.2). Moreover, in Section 3.3 we describe the resource of reconciled DBpedia information we create and release.

3.1. Data set

To evaluate the RADAR framework, we rely on the data set presented in Cabrio et al. [13], the only available annotated data set of possibly inconsistent information in DBpedia language-specific chapters up to our knowledge. It is composed of 400 annotated pairs of values (extracted from English, French and Italian DBpedia chapters), a sample that is assumed to be representative of the linguistic phenomena holding among values in DBpedia chapters. Note that the size of the DBpedia chapter does not bias the types of relation identified among the values, nor their distribution, meaning that given a specific relation (property), each DBpedia chapter deals with that property in the same way. We randomly divided such data set into a development (to tune RADAR) and a test set, keeping

the proportion among the categories distribution.[11] Table 2 reports on the data set statistics, and shows how many annotated relations belong to each of the categories (described in Section 2.2).

3.2. Results and discussion

Table 3 shows the results obtained by RADAR on the relation classification task on the test set. As baseline, we apply an algorithm exploiting only cross-lingual links (using WikiData), and exact string matching. Since we want to produce a resource as precise as possible for future reuse, RADAR has been tuned to reward precision (i.e., so that it does not generate false positives for a category), at the expense of recall (errors follow from the generation of false negatives for positive classes). As expected, the highest recall is obtained on the *surface form* category (our baseline performs even better than RADAR on such category). The *geo-specification* category has the lowest recall, either due to missing alignments between DBpedia and GeoNames (e.g. Ixelles and Bruxelles are not connected in GeoNames), or to the values complexity in the *renaming* subcategory (e.g., Paris vs First French Empire, or Harburg (quarter) vs Hambourg). In general, the results obtained are quite satisfying, fostering future work in this direction.

Since we consider *text mismatch* as a negative class (Section 2.2), it includes the cases in which RADAR fails to correctly classify a pair of values into one of the positive classes. For date and numerical mismatches, f1=1 (detecting them is actually a trivial task, and therefore they are not included in Table 3. See footnote 8). *Overall positive* means that RADAR correctly understands the fact that the different answers to a certain query are all correct and not conflicting. RADAR precision in this case is 1, and it is important to underline this aspect in the evaluation, since this confirms the reliability of the released reconciled DBpedia with this respect. The overall positive result is higher than the partial results because in the precision of partial values we include the fact that if e.g., a *surface form* relation is wrongly labeled as *geo-specification*, we consider this mistake both as a false negative for *surface form*, and as a false positive for *geo-specification*. This

---

[11]The data set is available at http://www.airpedia.org/radar-1.0.nt.bz2. The original work is based on DBpedia 3.9, but we updated it to DBpedia 2014. Thus, we deleted one pair, since the DBpedia page of one of the annotated entities does not exist anymore.

Table 2

Statistics on the data set used for RADAR 2.0 evaluation

data set	*# triples*	*# annotated positive relations*			*# annotated negative relations*		
		surface-form	geo-specific.	inclusion	text mismatch	date mismatch	numerical mismatch
dev set	104	28	18	20	13	13	12
test set	295	84	48	55	36	37	35
total	399	112	66	75	49	50	47

Table 3

Results of the system on the relations classification

system	*Relation category*	*precision*	*recall*	*f1*
RADAR 2.0	*surface form*	0.91	0.83	0.87
	geo-specification	0.94	0.60	0.73
	inclusion	0.86	0.69	0.77
	overall positive	1.00	0.74	0.85
	text mismatch	0.45	1	0.62
baseline	*surface form*	1.00	0.44	0.61
	geo-specification	0.00	0.00	0.00
	inclusion	0.00	0.00	0.00
	overall positive	1.00	0.21	0.35
	text mismatch	0.21	1	0.35

means that RADAR is very precise in assigning positive relations, but it could provide a less precise classification into finer-grained categories.

3.3. Reconciled DBpedia resource

We applied RADAR 2.0 on 300 DBpedia properties - the most frequent in terms of chapters mapping such properties, corresponding to the 47.8% of DBpedia instantiated properties. We considered ∼5M Wikipedia entities. The outcoming resource, a sort of *universal DBpedia*, counts ∼50M of reconciled triples, from 15 DBpedia chapters: Bulgarian, Catalan, Czech, German, English, Spanish, French, Hungarian, Indonesian, Italian, Dutch, Polish, Portuguese, Slovenian, Turkish. Notice that we did not consider the endpoint availability as a requirement to choose the DBpedia chapters: data are directly extracted from the resource.

For functional properties, the RADAR framework is applied as described in Section 2. On the contrary, the strategy to reconcile the values of non functional properties is slightly different: when a list of values is admitted (e.g. for properties `child` or `instruments`), RADAR merges the list of the elements provided by the DBpedia chapters, and ranks them with respect to the confidence assigned to their source, after reconcil-

ing positive relations only (there is no way for lists to understand if an element is incorrect or just missing, e.g. in the list of the instruments played by John Lennon). But since the distinction between functional/non functional properties is not precise in DBpedia, we manually annotated the 300 properties with respect to this classification, to allow RADAR to apply the correct reconciliation strategy, and to produce a reliable resource. In total, we reconciled 3.2 million functional properties, with an average accuracy comparable to the one described in Table 3.[12]

Moreover, we carried out a merge and a light-weight reconciliation of DBpedia classes applying the strategy "DBpedia CL" [1] (e.g., *Michael Jackson* is classified as a `Person` in the Italian and German DBpedia chapters, an `Artist` in the English DBpedia and a `MusicalArtist` in the Spanish DBpedia. The most specific class is the last one, so the entity *Michael Jackson* becomes `MusicalArtist` in every language).

4. Integrating RADAR in a QA system

We integrate RADAR into a QA system over language-specific DBpedia chapters, given the impor-

---

[12]Available here: `http://qakis.org/resources.htm`

tance that information reconciliation has in this context. Indeed, a user expects a unique (and possibly correct) answer to her factual natural language question, and would not trust a system providing her with different and possibly inconsistent answers coming out of a black box. A QA system querying different data sources needs therefore to weight in an appropriate way such sources in order to evaluate the information items they provide accordingly.

As QA system we selected QAKiS (Question Answering wiKiFramework-based System) [9], because it allows *i)* to query a set of language-specific DBpedia chapters using a natural language interface, and *ii)* its modular architecture can be flexibly modified to account for the proposed extension. QAKiS addresses the task of QA over structured knowledge-bases (e.g. DBpedia) [10], 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, making use of relational patterns (automatically extracted from Wikipedia), that capture different ways to express a certain relation in a language. The actual version of QAKiS targets questions containing a Named Entity (NE) related to the answer through one property of the ontology, as *Which river does the Brooklyn Bridge cross?*. Such questions match a single pattern.

In QAKiS, the SPARQL query created after the question interpretation phase is sent to a set of language-specific DBpedia chapters SPARQL endpoints (i.e., English, French, German and Italian) for answer retrieval. The set of retrieved answers from each endpoint is then sent to RADAR 2.0 for answer reconciliation. To test RADAR integration into QAKiS[13], the user can select the DBpedia chapter she wants to query besides English (that must be selected as it is needed for NE recognition), i.e., French, German or Italian DBpedia. Then the user can either write a question or select among a list of examples. Clicking on the tab *Reconciliation*, a graph with the answers provided by the different endpoints and the relations among them is shown to the user (as shown in Figures 3 and 4 for the questions *How tall is Stefano Tacconi?*, and *List the children of Margaret Thatcher*, respectively). Each node has an associated confidence score, resulting from the fuzzy labeling algorithm (described in Section 2.3). Moreover, each node is related to the oth-

ers by a relation of support or attack, and a further specification of such relations according to the categories described in Section 2.2 is provided to the user as answer justification of why the information items have been reconciled and ranked in this way.

4.1. QA experimental setting

To provide a quantitative evaluation of RADAR integration into QAKiS on a standard data set of natural language questions, we consider the questions provided by the organizers of the QALD challenge (Question Answering over Linked Data challenge), now at its fifth edition, for the DBpedia track.[14] More specifically, we collect the questions sets of QALD-2 (i.e. 100 questions of the training and 100 questions of the test sets), the test set of QALD-4 (i.e. 50 questions), and the questions sets of QALD-5 (50 additional training questions with respect to the previous years training set, and 59 questions in the test sets). These 359 questions correspond to all the questions released in the five years of the QALD challenge (given the fact that the questions of QALD-1 are included into the question set of QALD-2, and the question set of QALD-3 is the same as QALD-2, but translated into 6 languages, and the training sets of QALD-4 and 5 include all the questions of QALD-2). QALD-3 also provides natural language questions for Spanish DBpedia, but given that the current version of QAKiS cannot query the Spanish DBpedia, we could not use this question set.

We extract from this reference data set of 359 questions, 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 QALD-2 training set, 32 questions in QALD-2 test sets, 12 in QALD-4 test set, 18 in QALD-5 training set, and 11 in QALD-5 test set. The discarded questions require either some forms of reasoning (e.g., counting or ordering) on data, aggregation (from data sets different from DBpedia), involve n-relations, or are boolean questions. We consider these 99 questions as the QALD reference data set for our experiments.

4.2. Results on QALD answers reconciliation

We run the questions contained into our QALD reference data set on the English, German, French and
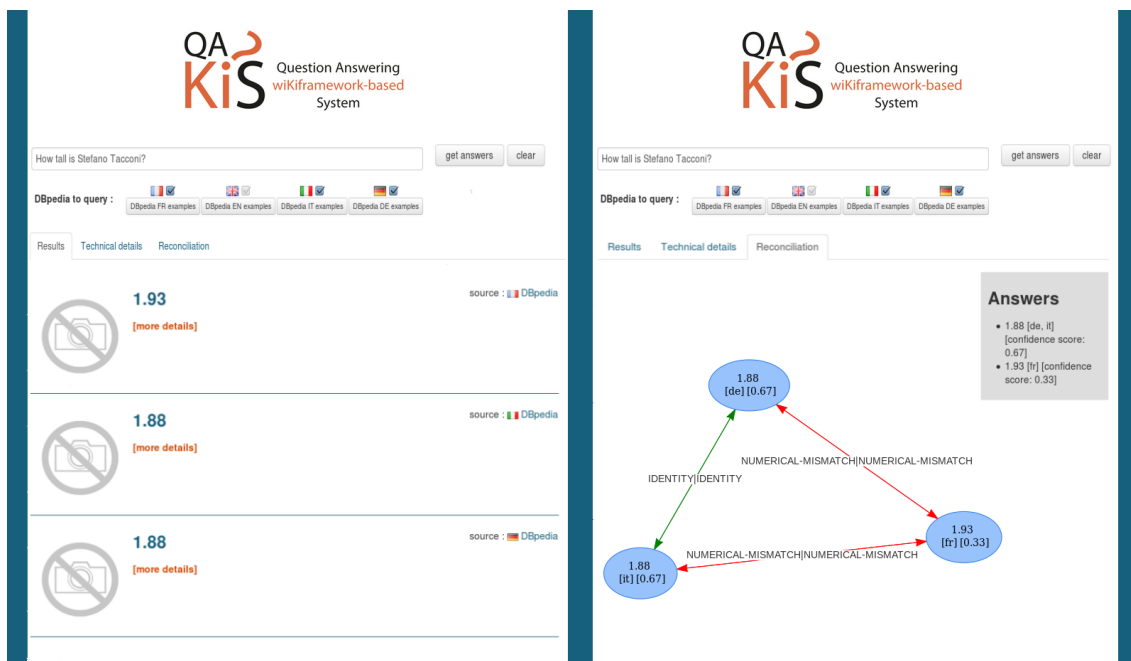

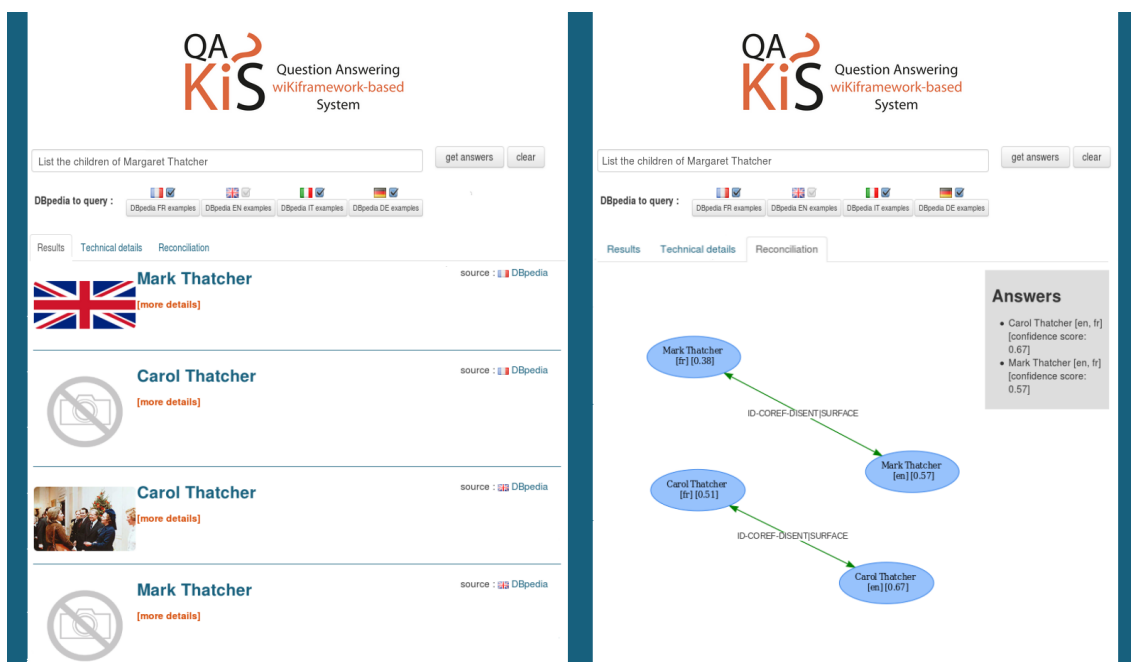Fig. 3. QAKiS + RADAR demo (functional properties)

Fig. 4. QAKiS + RADAR demo (non functional properties)

Italian chapters of DBpedia. Since the questions of QALD were created to query the English chapter of DBpedia only, it turned out that only in 43/99 cases 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 *Who developed Skype?* the English DBpedia provides *Skype Limited* as the answer, while the French one outputs *Microsoft* and *Microsoft Skype Division*. Or given the question *How many employees does IBM have?*, the English and the German DBpedia chapters provide 426751 as answer, while the French DBpedia 433362. Table 5 lists these 43 QALD questions, specifying which DBpedia chapters (among the English, German, French and Italian ones) contain at least one value for the queried relation. This list of question is the reference question set for our evaluation.

We evaluated the ability of RADAR 2.0 to correctly classify the relations among the information items provided by the different language-specific SPARQL endpoints as answer to the same query, w.r.t. a manually annotated goldstandard, built following the methodology in Cabrio et al. [13]. More specifically, we evaluate RADAR with two sets of experiments: in the first case, we start from the answers provided by the different DBpedia endpoints to the 43 QALD questions, and we run RADAR 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. In this second case, the input are the 43 natural language questions.

Table 4 reports on the results we obtained for the two experiments. As already noticed before, QALD data set was created to query the English chapter of DBpedia only, and therefore this small data set does not capture the variability of possibly inconsistent answers that can be found among DBpedia language-specific chapters. Only three categories of relations are present in this data, i.e., *surface forms*, *geo-specification*, and *inclusion*, and for this reason RADAR has outstanding performances on it when applied on the correct mapping between NL questions and the SPARQL queries. When QAKiS is added into the loop, its mistakes in interpreting the NL question and translating it into the correct SPARQL query are propagated in RADAR (that receives in those cases a wrong input), decreasing the total performances.

Notice that in some cases the question interpretation can be tricky, and can somehow bias the evaluation of the answers provided by the system. For instance, for the question *Which pope succeeded John Paul II?*, the English DBpedia provides *Benedict XVI* as the answer, while the Italian DBpedia provides also other names of people that were successors of John Paul II in other roles, as for instance in being the Archbishop of Krakow. But since in the goldstandard this question is interpreted as being the successor of John Paul II in the role of Pope, only the entity *Benedict XVI* is accepted as correct answer.

When integrated into QAKiS, RADAR 2.0 outperforms the results obtained by a preliminary version of the argumentation module, i.e. RADAR 1.0 [11], for the positive relations classification (the results of the argumentation module only cannot be strictly compared with the results obtained by RADAR 2.0, since *i)* in its previous version the relation categories are different and less fine-grained, and *ii)* in [11] only questions from QALD-2 were used in the evaluation), showing an increased precision and robustness of our framework. Note that this evaluation is not meant to show that QAKiS performance is improved by RADAR. Actually, RADAR does not affect the capacity of QAKiS to answer questions: RADAR is used to disambiguate among multiple answers retrieved by QAKiS in order to provide to the user the most reliable (and hopefully correct) one.

One of the reasons why RADAR is implemented as a framework that can be integrated on the top of a QA system existing architecture (and is therefore system-independent), is because we would like it to be tested and exploited by potentially all QA systems querying more than one DBpedia chapter (up to our knowledge QAKiS is the only one at the moment, but given the potential increase in the coverage of a QA system querying multiple DBpedia language-specific chapters [10], we expect other systems to take advantage of these interconnected resources soon).

5. Related work

The present paper is an extended version of our previous work [12,11,8], introducing RADAR 1.0. The following common points are present: the idea of using argumentation theory to detect inconsistencies over the results set of a question answering system exploiting DBpedia, and the bipolar extension of the original fuzzy labeling algorithm [15] to judge arguments' acceptability in presence of both support and attack relations. However, the present paper presents a substantial extension with respect to this preliminary work. More

Table 4
Results on QALD relations classification

system	Relation category	precision	recall	f1
RADAR 2.0 (only)	surface form	1.00	0.98	0.99
	geo-specification	0.88	0.80	0.84
	inclusion	0.80	1.00	0.88
	overall positive	1.00	0.98	0.99
baseline	surface form	1.00	0.97	0.98
	geo-specification	0.00	0.00	0.00
	inclusion	0.00	0.00	0.00
	overall positive	1.00	0.86	0.92
QAKiS + RADAR 2.0	surface form	1.00	0.59	0.74
	geo-specification	0.88	0.80	0.84
	inclusion	0.80	1.00	0.88
	overall positive	1.00	0.63	0.77
QAKiS + baseline	surface form	1.00	0.58	0.74
	geo-specification	0.00	0.00	0.00
	inclusion	0.00	0.00	0.00
	overall positive	1.00	0.52	0.68
QAKiS + RADAR 1.0 [11] (on QALD-2 questions only)	**overall positive**	0.54	0.56	0.55

specifically, the main enhancements are reported in the following:

Relations categorization : RADAR 2.0 exploits the categorization we introduced in [13], as mentioned in Section 2.2. However, the work presented in [13] is purely theoretic and the contribution on this side is to study how to make RADAR 2.0 match these linguistic-based relations with respect to the DBpedia use case. Moreover, the categorization of the possible relations holding between the information items we adopt here is different (more linguistically-motivated) and more fine-grained than the one we used in [11]. This fine-grained categorization allows for a more insightful justification graph;

Relations extraction : the relations holding between the elements of the results set are here automatically extracted with the application of more robust techniques than in [11]. More precisely, the way RADAR 2.0 extracts these relations in an automated way is different from the way RADAR 1.0 extracts them: RADAR 2.0 adopts external resources to improve the extraction of the correct relation, such as MusicBrainz, the BNCF (Biblioteca Nazionale Centrale di Firenze), DBpedia and Wikipedia, GeoNames, and WikiData;

Evaluation : While in [8] only data from QALD-2 have been used, here we use all data available from the QALD challenges (all editions), and the Italian chapter of DBpedia is added as RDF data set to be queried with QAKiS (not present in our previous works on the topic). Moreover, the results presented in this paper show a higher precision with respect to the results obtained with RADAR 1.0 and reported in [11] (f1 increments from 0.55 to 0.77 for the positive relations classification if we consider QALD-2 data only). In addition, the new evaluation considers 15 DBpedia chapters instead of the 3 chapters used in [8], i.e., English, German and French;

Resource : differently from [11] where no resource resulted from the inconsistencies detection process, here we generate a resource applying the proposed framework to 15 reconciled language-specific DBpedia chapters, and we release it.

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 [23] 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 considers language-specific DBpedia chapters, and they do not provide a mechanism to reconcile the different answers returned by heterogenous endpoints. Finally, none of

them provides explanations about the answer returned to the user.

Several works address alignment agreement based on argumentation theory. More precisely, Laera et al. [22] address alignment agreement relying on argumentation to deal with the arguments which attack or support the candidate correspondences among ontologies. Doran et al. [16] 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 [17] 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. We share with these approaches the use of argumentation to detect inconsistencies, but RADAR goes beyond them: we identify in an automated way relations among information items that are more complex than `sameAs` links (as in ontology alignment). Moreover, these approaches do not consider trust-based acceptance degrees of the arguments, lacking to take into account a fundamental component in the arguments' evaluation, namely their sources.

We mentioned these works applying argumentation theory to address ontology alignment agreements as examples of applications of this theory to open problems in the Semantic Web domain. Actually, the two performances cannot be compared to show the superiority of one of the two approaches, as the task is different.

6. Conclusions

In this paper, we have introduced and evaluated the RADAR 2.0 framework for information reconciliation over language-specific DBpedia chapters. The framework is composed of three main modules: a module computing the confidence score of the sources depending either on the length of the related Wikipedia page or on the geographical characterization of the queried entity, a module retrieving the relations holding among the elements of the results set, and finally a module computing the reliability degree of such elements depending on the confidence assigned to the sources and the relations among them. This third module is based on bipolar argumentation theory, and a bipolar fuzzy labeling algorithm [11] is exploited to return the acceptability degrees. The resulting graph of the results set, together with the acceptability degrees assigned to

each information item, justifies to the user the returned answer and it is the result of the reconciliation process. The evaluation of the framework shows the feasibility of the proposed approach. Moreover, the framework has been integrated in the question answering system over Linked Data called QAKiS, allowing to reconcile and justify the answers obtained from four language-specific DBpedia chapters (i.e. English, French, German and Italian). Finally, the resource generated applying RADAR to 300 properties in 15 DBpedia chapters to reconcile their values is released.

There are several points to be addressed as future work. First, the user evaluation should not be underestimated: we will soon perform an evaluation to verify whether our answer justification in QAKiS appropriately suits the needs of the data consumers, and to receive feedback on how to improve such visualization. Second, at the present stage we assign a confidence score to each source following two criteria, however another possibility is to leave the data consumer itself to assign such confidence degree to the sources depending on the kind of information she is looking for. Finally, the proposed framework is not limited to the case of multilingual chapters of DBpedia. The general approach RADAR is based on allows to extend it to various cases like inconsistent information from multiple English data endpoints. The general framework would be the same, the only part to be defined are the rules to extract the relations among the retrieved results. Investigating how a module of this type can be adopted as a fact checking module is part of our future research plan.

Table 5

QALD questions used in the evaluation (in bold the ones correctly answered by QAKiS; *x* means that the corresponding language specific DBpedia chapter (EN, FR, DE, IT) contains at least one value for the queried relation; *dbo* means DBpedia ontology)

ID, question set	question	dbo relation	EN	FR	DE	IT
84, QALD-2 train	Give me all movies with Tom Cruise.	starring	x	x	x	
10, QALD-2 train	**In which country does the Nile start?**	sourceCountry	x	x		
63, QALD-2 train	**Give me all actors starring in Batman Begins.**	starring	x	x	x	x
43, QALD-2 train	Who is the mayor of New York City?	leaderName	x		x	x
54, QALD-2 train	Who was the wife of U.S. president Lincoln?	spouse	x	x		
6, QALD-2 train	**Where did Abraham Lincoln die?**	deathPlace	x	x	x	
31, QALD-2 train	**What is the currency of the Czech Republic?**	currency	x	x	x	x
73, QALD-2 train	**Who owns Aldi?**	keyPerson	x	x		x
20, QALD-2 train	**How many employees does IBM have?**	numberOfEmployees	x	x	x	x
33, QALD-2 train	**What is the area code of Berlin?**	areaCode	x			
2, QALD-2 test	**Who was the successor of John F. Kennedy?**	successor	x	x		
4, QALD-2 test	How many students does the Free University in Amsterdam have?	numberOfStudents	x	x	x	
14, QALD-2 test	Give me all members of Prodigy.	bandMember	x	x		
20, QALD-2 test	**How tall is Michael Jordan?**	height	x		x	x
21, QALD-2 test	**What is the capital of Canada?**	capital	x	x	x	x
35, QALD-2 test	**Who developed Skype?**	product	x	x		
38, QALD-2 test	How many inhabitants does Maribor have?	populationTotal	x			x
41, QALD-2 test	**Who founded Intel?**	foundedBy	x	x		x
65, QALD-2 test	**Which instruments did John Lennon play?**	instrument	x	x		
68, QALD-2 test	**How many employees does Google have?**	numberOfEmployees	x	x		x
74, QALD-2 test	**When did Michael Jackson die?**	deathDate	x	x	x	
76, QALD-2 test	**List the children of Margaret Thatcher.**	child	x	x		
83, QALD-2 test	How high is the Mount Everest?	elevation	x	x		x
86, QALD-2 test	**What is the largest city in Australia?**	largestCity	x	x		
87, QALD-2 test	Who composed the music for Harold and Maude?	musicComposer	x		x	x
34, QALD-4 test	**Who was the first to climb Mount Everest?**	firstAscentPerson	x		x	
21, QALD-4 test	**Where was Bach born?**	birthPlace	x	x	x	x
32, QALD-4 test	**In which countries can you pay using the West African CFA franc?**	currency	x		x	
12, QALD-4 test	How many pages does War and Peace have?	numberOfPages	x	x		
36, QALD-4 test	Which pope succeeded John Paul II?	successor	x			x
30, QALD-4 test	When is Halloween?	date	x	x		
259, QALD-5 train	Who wrote The Hunger Games?	author	x	x		
280, QALD-5 train	**What is the total population of Melbourne, Florida?**	populationTotal	x	x		x
282, QALD-5 train	In which year was Rachel Stevens born?	birthYear	x	x	x	x
283, QALD-5 train	**Where was JFK assassinated?**	deathPlace	x	x	x	x
291, QALD-5 train	**Who was influenced by Socrates?**	influencedBy	x	x		
295, QALD-5 train	Who was married to president Chirac?	spouse	x	x		
298, QALD-5 train	**Where did Hillel Slovak die?**	deathPlace	x	x	x	x
7, QALD-5 test	Which programming languages were influenced by Perl?	influencedBy	x	x	x	x
18, QALD-5 test	**Who is the manager of Real Madrid?**	manager	x	x		
19, QALD-5 test	**Give me the currency of China.**	country	x		x	
32, QALD-5 test	What does the abbreviation FIFA stand for?	name	x		x	x
47, QALD-5 test	**Who were the parents of Queen Victoria?**	parent	x		x	x

References

[1] Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. Automatic expansion of dbpedia exploiting wikipedia cross-language information. In *ESWC*, pages 397–411, 2013.

[2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4):365–410, 2011.

[3] Trevor J. M. Bench-Capon, D. Lowes, and A. M. McEnery. Argument-based explanation of logic programs. *Knowl.-Based Syst.*, 4(3):177–183, 1991.

[4] Trevor J. M. Bench-Capon and Giovanni Sartor. Theory based explanation of case law domains. In *ICAIL*, pages 12–21, 2001.

[5] Floris Bex and Douglas Walton. Burdens and standards of proof for inference to the best explanation. In Radboud Winkels, editor, *Legal Knowledge and Information Systems - JURIX 2010: The Twenty-Third Annual Conference on Legal Knowledge and Information Systems, Liverpool, UK, 16-17 December 2010*, volume 223 of *Frontiers in Artificial Intelligence and Applications*, pages 37–46. IOS Press, 2010.

[6] Joshua E. Blumenstock. Size matters: Word count as a measure of quality on wikipedia. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 1095–1096, New York, NY, USA, 2008. ACM.

[7] Volha Bryl and Christian Bizer. Learning conflict resolution strategies for cross-language wikipedia data fusion. In *WWW (Companion Volume)*, pages 1129–1134, 2014.

[8] Elena Cabrio, Alessio Palmero Aprosio, and Serena Villata. Reconciling information in dbpedia through a question answering system. In Matthew Horridge, Marco Rospocher, and Jacco van Ossenbruggen, editors, *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*, volume 1272 of *CEUR Workshop Proceedings*, pages 49–52. CEUR-WS.org, 2014.

[9] Elena Cabrio, Julien Cojan, Alessio Palmero Aprosio, Bernardo Magnini, Alberto Lavelli, and Fabien Gandon. Qakis: an open domain qa system based on relational patterns. In *Procs of ISWC 2012 (Posters & Demos)*, volume 914, 2012.

[10] Elena Cabrio, Julien Cojan, and Fabien Gandon. Mind the cultural gap: bridging language specific dbpedia chapters for question answering. In Philipp Cimiano and Paul Buitelaar, editors, *to appear in Towards the Multilingual Semantic Web*. Springer Verlag, 2014.

[11] Elena Cabrio, Julien Cojan, Serena Villata, and Fabien Gandon. Argumentation-based inconsistencies detection for question-answering over dbpedia. In *NLP-DBPEDIA@ISWC*, 2013.

[12] Elena Cabrio, Julien Cojan, Serena Villata, and Fabien Gandon. Hunting for inconsistencies in multilingual dbpedia with qakis. In Eva Blomqvist and Tudor Groza, editors, *Proceedings of the ISWC 2013 Posters & Demonstrations Track, Sydney, Australia, October 23, 2013*, volume 1035 of *CEUR Workshop Proceedings*, pages 69–72. CEUR-WS.org, 2013.

[13] Elena Cabrio, Serena Villata, and Fabien Gandon. Classifying inconsistencies in dbpedia language specific chapters. In *Procs of LREC-2014*, 2014.

[14] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Bipolarity in argumentation graphs: Towards a better understanding. In *Procs of SUM 2011*, volume 6929 of *LNCS*, pages 137–148. Springer, 2011.

[15] Célia da Costa Pereira, Andrea Tettamanzi, and Serena Villata. Changing one's mind: Erase or rewind? In *Procs of IJCAI 2011*, pages 164–171. IJCAI/AAAI, 2011.

[16] Paul Doran, Valentina Tamma, Ignazio Palmisano, and Terry Payne. Efficient argumentation over ontology correspondences. In *Procs of AAMAS 2009*, pages 1241–1242, 2009.

[17] Cássia Trojahn dos Santos and Jérôme Euzenat. Consistency-driven argumentation for alignment agreement. In *Procs of OM 2010, CEUR Workshop Proceedings 689*, 2010.

[18] Phan M. Dung. 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.

[19] Paul E. Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artif. Intell.*, 175(2):457–486, 2011.

[20] Anthony Hunter. A probabilistic approach to modelling uncertain logical arguments. *Int. J. Approx. Reasoning*, 54(1):47–81, 2013.

[21] Anthony Hunter and Matthew Williams. Aggregating evidence about the positive and negative effects of treatments. *Artificial Intelligence in Medicine*, 56(3):173–190, 2012.

[22] Loredana Laera, Ian Blacoe, Valentina Tamma, Terry Payne, Jérôme Euzenat, and Trevor Bench-Capon. Argumentation over ontology correspondences in MAS. In *Procs of AAMAS 2007*, pages 1–8, 2007.

[23] Vanessa Lopez, Victoria S. Uren, Marta Sabou, and Enrico Motta. Is question answering fit for the semantic web?: A survey. *Semantic Web*, 2(2):125–155, 2011.

[24] Pablo Mendes, Max Jakob, and Christian Bizer. DBpedia: A multilingual cross-domain knowledge base. In *Procs of LREC 2012*. ELRA, 2012.

[25] Pablo N. Mendes, Hannes Mühleisen, and Christian Bizer. Sieve: linked data quality assessment and fusion. In *Procs of the Joint EDBT/ICDT Workshops*, pages 116–123. ACM, 2012.

[26] Ana Rojo. *Step by Step: A Course in Contrastive Linguistics and Translation*. Peter Lang, 2009.

[27] Douglas Walton. Explanations and arguments based on practical reasoning. In Thomas Roth-Berghofer, Nava Tintarev, and David B. Leake, editors, *Explanation-aware Computing, Papers from the 2009 IJCAI Workshop, Pasadena, California , USA, July 11-12, 2009*, pages 72–83, 2009.