Merging and Ranking answers in the Semantic Web: The Wisdom of Crowds

Vanessa Lopez, Andriy Nikolov, Miriam Fernandez, Marta Sabou, Victoria Uren, Enrico Motta. [1]

[1] Knowledge Media Institute. The Open University, Walton Hall, MK76AA, United Kingdom
{v.lopez, a.nikolov, m.fernandez, r.m.sabou, e.motta}@open.ac.uk, v.uren@dcs.shef.ac.uk

Abstract. In this paper we propose algorithms for combining and ranking answers from distributed heterogeneous data sources in the context of a multi-ontology Question Answering task. Our proposal includes a merging algorithm that aggregates, combines and filters ontology-based search results and three different ranking algorithms that sort the final answers according to different criteria such as popularity, confidence and semantic interpretation of results. An experimental evaluation on a large scale corpus indicates improvements in the quality of the search results with respect to a scenario where the merging and ranking algorithms were not applied. These collective methods for merging and ranking allow to answer questions that are distributed across ontologies, while at the same time, they can filter irrelevant answers, fuse similar answers together, and elicit the most accurate answer(s) to a question.

Keywords: Merging, Ranking, Fusion, Question Answering, Semantic Web.

1 Introduction

Large-scale, open-domain question answering has been addressed with a variety of approaches in the last decades. Firstly, open domain Question Answering (QA) across unstructured Web data has been stimulated since 1999 by the TREC QA track evaluations. Secondly, the intuition that it would be easier to obtain answers from structured data (i.e., an ontology) where ambiguities in the queries can be resolved using reasoning techniques, has lead to much interest in Natural Language Interfaces (NLI) to knowledge bases, and in particular on NLI systems that directly query a given ontology [2, 4, 15]. However, although existing ontology-based NLI systems are generally domain independent or portable across domains, their scope is limited to one (or a set of) a-priori selected domain(s) at a time. A third recent trend in obtaining structured answers in a an open domain scenario has seen several industrial startups such as Powerset, START, Wolfram Alpha, True Knowledge[1], among others. A well-established approach for these systems is to semi-automatically build their own comprehensive factual knowledge bases. For example, similarly to OpenCyc and Freebase[2], the Wolfram Alpha knowledge inference engine builds a broad trusted knowledge base about the world by ingesting massive amounts of information (currently storing approximately 10TBs, still a tiny fraction of the Web). True Knowledge relies on users to add and curate its information, while PowerSet uses Freebase and annotates Wikipedia with its semantic resources.

[1] http://www.powerset.com/, http://start.csail.mit.edu/, http://www.wolframalpha.com/index.html and http://www.trueknowledge.com/ respectively.
[2] www.opencyc.org, http://www.freebase.com

Differently from this last trend, PowerAqua [9] attempts to perform open domain QA by taking advantage of the freely available structured information on the Semantic Web (SW)[3]. This is a key difference as, unlike the previous systems, PowerAqua does not impose an internal structure on its knowledge nor does it claim ownership of its knowledge base, but rather explores the increasing number of multiple, heterogeneous knowledge sources available on the Web. As such, PowerAqua supports users in searching and exploring information on the SW. Users introduce a factual query in natural language and PowerAqua is able to match it into one or many ontological facts, from which an answer can be inferred.

A major challenge faced by PowerAqua is that answers to a query may need to be derived from different ontological facts and even different semantic sources and domains. Often, multiple, redundant information needs to be combined (or *merged*) to obtain a reduced number of answers. Then, because different semantic sources have varying levels of quality and trust, when multiple answers are derived to a query it is important to be able to *rank* them in terms of their relevance to the query at hand.

In this paper we present merging and ranking methods for combining results (answers given as ontological facts) across ontologies. These methods have been integrated in PowerAqua and evaluated in the context of a multi-ontology QA task. The question that we try to answer here is: are aggregated answers from many heterogeneous independent semantic sources better than answers derived from single ontological facts? Or, similarly to the hypothesis of *Wisdom of Crowds*[4], are the many smarter than the few? These initial experiments confirm that the quality of derived answers can be improved by cross-ontology merging and ranking techniques.

The rest of the paper is structured as follows. Section 2 introduces a motivating scenario, Sections 3 and 4 describe the merging and ranking algorithms respectively. Section 5 describes the evaluation setup and the analysis of the results. We present related work in Section 6, and conclude in Section 7.

2 Motivating scenario: Question Answering on the Semantic Web

Because PowerAqua derives answers from multiple online semantic resources, thus operating in a highly heterogeneous search space, it requires mechanisms for merging and ranking answers to generate a commonly agreed set of answers across ontologies. Consider, for example, the query "Which languages are spoken in Islamic countries?". PowerAqua is designed following a cascade model (see Fig. 1). Steps 1, 2 and 3 are have been detailed in [9], so here we only summarize the key aspects of its behavior. At the first stage, the linguistic component using the GATE NL processing tool [5] transforms the NL query into an intermediate format called Query-Triples (QT). These QTs relate words together and mimic the structure of triples in the ontology but using the NL terms in the user query. For instance, our example query is initially translated into the QT <languages, spoken, Islamic countries>.

At the next step, the QTs are passed on to the PowerMap component, which identifies potentially suitable ontologies to answer a query, producing initial element level mappings between the QT terms and the entities in these sources. The output of PowerMap is a set of Entity Mapping Tables (EMTs), where each table associates each QT term with a set of entities found on the SW. To identify all semantic sources that are likely to describe QT terms,

[3] To be precise, PowerAqua accesses multiple ontologies through the Watson semantic gateway (http://watson.kmi.open.ac.uk/WatsonWUI/) or by exploring information stored in online servers.

[4] A book written by Jame Surowiecki in 2004, primarily on the fields of economic and psychology, stating that "a diverse collection of independently-deciding individuals is likely to make certain types of decisions and predictions better than individuals or even experts." It is also connected to social and collective intelligence on the Web (http://en.wikipedia.org/wiki/The_Wisdom_of_Crowds).

PowerMap maximizes recall by searching for approximate (lexical overlap) and exact (lexical equality) mappings. These are jointly referred to as equivalent mappings. PowerMap also uses both WordNet and the SW itself as sources of background knowledge to perform query expansion and to find lexically dissimilar (but semantically similar) matches – including synonyms, hypernyms and hyponyms. A semantic validation component attempts to generate WordNet synsets for all classes and individuals included in the EMTs. PowerMap uses the Watson[5] semantic search engine as a gateway to the SW. In addition, PowerMap can also query its own repositories and offers the capability to index and add new online ontologies[6].

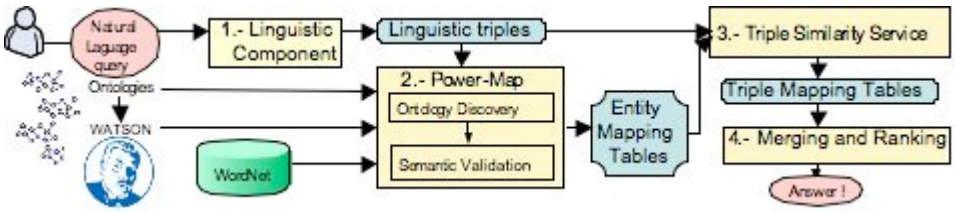

Fig. 1: PowerAqua components

In the third step, the Triple Similarity Service (TSS) matches the QTs to ontological expressions. The TSS takes as input the EMTs returned by PowerMap and the initial QTs, and returns a set of Triple Mapping Tables (TMTs), which define a set of complete mappings between a QT, and the appropriate Onto-Triples (OTs). The TSS chooses whenever possible the ontologies that better cover the user query and domain. In our example, as PowerMap does not find any covering ontology with mappings for both arguments in the QT: "languages" and "Islamic countries", the TSS algorithm reiterates again by splitting the compound term "Islamic countries", and consequently modifying the QT into: <languages, spoken, countries / Islamic> and creating a new QT for the compound <Islamic, ?, countries>. For the QTs obtained in this second iteration, the TSS extracts, by analyzing the ontology relations, a small set of covering ontologies containing the valid OTs that jointly cover the user query and produce an answer. The TMTs generated for each QT by the TSS are presented in Table 1.

Table 1. Triple Mapping Tables returned by PowerAqua for the example query.

QT_1: <languages, spoken, countries / islamic>	
Dbpedia_infoboxes	OT_1 <*language, regionalLanguage, Country*> - 164 answers. E.g.: English (Pakistan), Arabic (Somalia), French (Algeria), Kurdish (Iran), Pashto (Pakistan), Welsh (United Kingdom), Albanian (Serbia), Catalan (Spain), Munji (Afghanistan).
Dbpedia_infoboxes	OT_2 <*language, states, country* > - 713 answers. E.g.: Canadian_French (Canada), Japanese (Japan), Russian (Poland), Bukawa (Papau New Guinea, Malay (Philippines), Filipino (Philippines), Hindi (India), Wakhi (Pakistan)
QT_2: <Islamic, ?, countries>	
Dbpedia_infoboxes	OT_3 <*country, governmenttype, Islamic_republic*> - 3 answers : Afghanistan (Afghanistan), Islamic_State_of_Afghanistan, Iran, Pakistan
Dbpedia_infoboxes	OT_4 <*Country, country, Islamic University*> - 1answer: Bangladesh
SWETO ontology	OT_5 <*Country, occurred_in, Terrorist_Attack*> <*Terrorist_Attack, responsible_for, Armed Islamic Group*> - *2 answers:* Algeria (Aug31,1998,ArmedIslamicGroup_Bombing), France (Jul11,1995,ArmedIslamicGroup_Shooting)

[5] http://watson.kmi.open.ac.uk/WatsonWUI

[6] PowerMap uses Lucene (lucene.apache.org) for the offline creation of the inverted indexes in order to provide efficient keyword searches to an ontology store in platforms such as Sesame.

Finally, because each resultant OT only leads to partial answers, they need to be combined into one complete answer. The goal of the fourth component is to merge and rank the various interpretations that different ontologies may produce. In our example, this is achieved by intersecting the answers from both QTs to obtain as a final set of answers those in "languages spoken in a country" whose "country" is shared with the answers obtained from "countries that are Islamic". Among other things, merging requires to identify similar entities across ontologies, e.g., "France" and "French republic".

In our example, from a total of 885 partial answers retrieved by PowerAqua the final set of answers obtained after merging contains 63 answers (e.g.: Aramaic (Iran), Abduyi_dialect (Iran), Kurdish (Iran), Pashto (Pakistan), Wakhi (Pakistan), among others. However, from those 63 answers 57 are correct. The 7 incorrect answers are derived from the partial answer "France" from the SWETO ontology, namely the languages, regional languages or extinct languages in France: French, Breton, Zarphatic, Balearic, Shuadit, Judeo-Spanish, Basque. Nevertheless, ranking measures can be applied to sort the answers and filter these results. As will be explained in Section 4, a ranking measure based on OTs confidence is capable of providing a lower confidence to answers derived from the SWETO OT than OTs formed from other ontologies by a direct relationship. This example illustrates how merging and ranking algorithms can enhance the quality of the search results. We now describe these algorithms.

3 Merging Algorithm

A side effect of the fact that PowerAqua explores multiple knowledge sources to obtain an answer is that, the TSS frequently associates the query to several OTs from different ontologies, each OT generating an answer. Depending on the complexity of the query, i.e., the number of QTs it has been translated to and the way each QT was matched to OTs, these individual answers may fall in one of the following categories: a) valid but duplicated answers, b) part of a composite answer and c) alternative answers derived from different ontological interpretations of the QTs. Different merging scenarios suit different categories: some cases require the intersection of the partial answers while other cases require their union. In this section we discuss the various merging scenarios (Section 3.1) and the fusion algorithm they rely on (Section 3.2).

3.1 Merging scenarios

Scenario 1 - A query translates into one QT only**.** These are the simplest queries, and therefore the easiest ones to merge, as each OT provides an answer on its own. The final set of answers is the union of all OTs across ontologies. E.g., "find me cities in Virginia".

Scenario 2 A query translates into two QTs that are linked together by the first QT term (the subject). Because each QT only leads to partial answers, they need to be merged to generate a complete response. This is achieved by intersecting the answers from both QTs. E.g., for the question "which Russian rivers flow into the Azov sea?" the final answers are composed by intersecting the results obtained with "*rivers* in Russia" and "*rivers* that flow in the Azov sea".

Scenario 3 A query translates into two QTs that are linked together by the object of the first one and the subject of the second one. In this scenario a complete answer can only be assigned by merging the partial answers. The answers for the first main QT are conditioned by the answers for the second QT. E.g., for the query "which rivers flow in European countries?" the

final set of answers are the set of all countries in which rivers flow, <rivers, flow, *country*>, which are linked to the set of European countries <*countries*, ?, European>.

Scenario 4 Complex queries which are translated into multiple QTs. These queries are solved in a similar way as a combination of scenarios 1, 2 and 3. E.g., for the query "What are the main cities located in US states bordering Georgia?", where Georgia is an ambiguous term that can represent a state in the USA or a country in the Caucasus, the valid answers come from the intersection or condition between first "cities in US states" and second "cities bordering Georgia" (both as a state and as a country) or "US states bordering Georgia". Both alternative paths result in cities in the state of Georgia (USA), rather than those in the country of Georgia.

In sum, the merging procedure deals with these four scenarios by applying three types of operators over the set of retrieved answers: *union*, *intersection* and *condition*. The union operator combines answers related to the same QT but coming from different ontologies (scenario 1). The intersection operator is needed when a query specifies more than one constraint for a single first query term (scenario 2). The intersection operator merges the answers from two corresponding QTs and removes those, which only occur in one answer set. The condition operator is similar to the intersection one; however, the condition operator filters the answers not by the first query term but by the second one (scenario 3). These operators can be applied to any complex cases in which the query is translated into several combinations of QTs (scenario 4) so all answers produced by alternative paths are merged.

3.2 The co-reference fusion algorithm

The merging procedure assigns the individuals returned as answers from different ontologies into subsets of answers that represent identical entities. The union operation processes a set of answers from a single QT and merges the similar answers representing identical entities. For example, the QT: <countries, locatedIn, Asia> returns, among its answers, "Thailand" from the TAP ontology and "Kingdom of Thailand" from the KIM ontology. These answers need to be grouped into a single subset as they refer to the same entity. As described above, depending on the query type, these subsets of answers can afterwards be combined by various operations.

The atomic procedure performed by all of these operations is matching. Two answers are compared and a decision is made about whether or not they are identical. To increase the speed, initial matching is performed only between the labels and local names of the returned entities. The entities are considered identical either if they are WordNet synonyms or if one of the used string similarity functions (Jaro, edit distance) returns a value above a certain threshold. A special case is the processing of ambiguity, which occurs when an entity has two potentially identical matching entities that belong to the same ontology. E.g., in "Give me all cities in the USA," a single entity, "arlington" from the *FAO* ontology, has two potential matches, "arlingtonVa" and "arlingtonTx" from the *UTexas geographic* ontology. Assuming that individuals belonging to the same ontology are distinct, the system tries to choose the best match out of the two using additional context data from the ontologies. The system receives, for each entity, all of their property values from their respective ontologies and compares these sets using the same similarity functions as above on their elements. Thus, in our example, context sets for both of the entities "arlington" and "arlingtonVa" mention "Virginia", while "arlingtonTx" mentions "Texas" instead. The similarity between the context sets of "arlington" and "arlingtonVa" is greater and, therefore, these entities are merged.

Pairwise comparison of entities would make the complexity of the procedure N^2 with respect to the input set size. In order to avoid this, candidate matches are selected using a

search over the indexes and the comparison focuses only on the entities that appear among the search results. This makes the complexity linear with respect to the answer set size.

4. Ranking algorithms

As we can see in Fig. 2, a filtered set of answers for each query is obtained after the merging step. While an unsorted list of answers can be manageable in some cases, the search system may become useless if the retrieval space is too big. In these cases a clear ranking criteria is needed to sort the final list of answers. The aim of the ranking measures presented here is to: a) assign a score to each individual answer and b) cluster the set of answers according to their score. Cluster analysis of ranking data attempts to identify typical groups of rank choices. In our case, according to the chosen ranking criteria, clusters identify results of different quality, popularity or meaning. The cluster ranked at position one (C@1) represents the best subset of results according to the chosen ranking method.

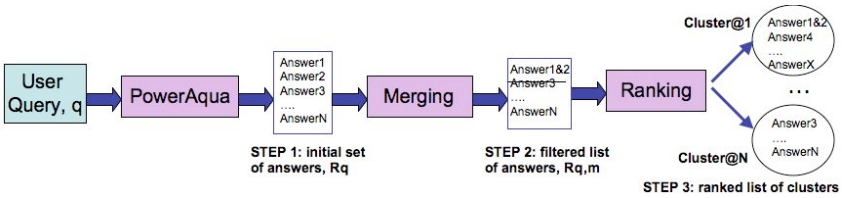

Fig. 2: Flow of the data retrieval, merging and ranking process.

The ranking component defines three different ranking algorithms:

- *Ranking by semantic similarity*: this ranking criterion exploits a semantic standard (WordNet) to compute the distance between OTs (Section 4.1).
- *Ranking by confidence*: this ranking criterion is based on the confidence of the OTs from which the answer is extracted. The quality of the OT depends on the type of the mapping between the OT and the QT (e.g., direct or indirect - Section 4.2).
- *Ranking by popularity:* this ranking criterion is based on the popularity of the answer, defined as the number of ontologies from which this answer can be derived (Section 4.3).

4.1 Ranking by semantic similarity

Answers are ranked according to the popularity of the semantic interpretation of the OT they belong to. The hypothesis behind this is that if an answer is derived from an OT that has similar interpretations to other OTs from different ontologies, it is more likely to be correct than answers coming from unique semantically different interpretations. This criterion takes advantage of the knowledge inherent in the ontology and WordNet high quality descriptions, and combines some well-founded ideas from the Word Sense Disambiguation community to compute semantic similarity distance across ontological entities as detailed in [8].

Let's clarify this idea with the example "Give me cities in Virginia", a query matched by PowerAqua into eight ontologies (and eight OTs). The final set of answers obtained after merging should be the union of all the answers describing cities in Virginia. However, an instance labeled "Copenhagen" appears between the set of merged answers. In order to rank last this inaccurate answer, semantic similarity between OTs is computed by comparing the distance path and common ancestors between the WordNet synsets for each ontological concept representing the subject and object of the triple (predicates are not well covered in WordNet, and in the case of instances we look at its type). The WordNet synset (i.e. the true

meaning) of an ontological term A, is determined by its parents in the hierarchy of the ontology (that is, those synsets of A that are similar to at least one synset of its ancestors in the ontology), and by its intended meaning in the user query (those synsets of A that are similar to at least one synset of the user term it matches to, if their labels differ). Having said that, while "city" has similar meanings in all its eight ontological matches (the synsets are semantically similar, even if they are not exactly the same), the ontological meaning of "Virginia" differs. Indeed, seven of the ontologies are referring to Virginia as an instance of an *state* or *province* (in USA), while the answer "Copenhagen" is derived from an eighth ontology about *film festivals* with the only semantically different OT, namely <city, hasActorWinner, VirginiaMadsen>, where "VirginiaMadsen" is classified as *person* in the ontology and not as a *state*, and therefore the intended meaning of the OT differs from the previous ones.

Ranking among answers is then calculated according to the popularity of the interpretation of the OT they belong to. Therefore, the first complete set of ranked answers comes from the union of the answers from the seven semantically similar OTs referring to cities in the *state* or *province* of Virginia. The answer, labeled "Copenhagen" because its derived from the only semantically different OT, it would be ranked lower than the previous answers.

To conclude with, the score for each answer is the number of ontologies that share the semantic interpretation of the OT they belong to, or -1 if an answer is coming from two OTs with different semantic interpretation. The C@1 groups all the answers ranked with score 2 or highest (at least two ontologies with the same semantic interpretation).

4.2. Ranking by confidence

The quality of the matching between a QT and one (or, in some cases, two) OTs often has an influence on the quality of the derived answers. We identified a set of rules to predict which of these OTs are likely to be more reliable and potentially lead to a better set of answers. These rules are listed in the same order as they are applied, i.e., from the most to the least significant. The rules we use can be seen as nodes in a decision tree. Their order of preference is discriminative in order to avoid conflicts.

1) OTs that are based on only equivalent (i.e., exact and approximate match) or synonym type mappings to the corresponding QT terms are ranked highest. E.g., for QT <capitals, ?, USA> ("Find me capitals in the USA") the OT1 = <capital (exact), isCityOf, State> <State, isStateOf, USA (exact)> with only equivalent mappings is ranked higher than OT2 =<City (hypernym), attribute_country, USA (exact)> which contains an hypernym.
2) OTs that link the two arguments in the QT through an IS-A relation (instead of an ad-hoc relationship) are ranked lower than any other triples. The reason for this is that many online ontologies misuse IS-A relations to model other types of relations (e.g., partonomy). We do not apply this rule when the original question contains an IS-A relation, as this is an indication that such a relation is expected (e.g. "which animals ARE reptiles?"). For instance, the QT <person, plays, Nirvana> ("Who play in Nirvana?") is matched to OT1 = <person, hasMember, MusicianNirvana> and OT2 = <Nirvana Meratnia, IS-A, person>. Note that while rule 1 ranks these two triples equally, this rule ranks OT1 higher (even if, in this particular case, OT2 complies with correct modeling).
3) OTs that cover not only all of the terms in the QT, but also the linguistic relation (mapped as an ontological entity), are ranked first over triples that do not cover the relation. E.g., for the QT <states, bordering, Colorado> ("what are the states bordering Colorado?") the OT <state, borders, Colorado (state)> is ranked higher than <state, runsThrough, Colorado (river)>.
4) The OTs containing more exact mappings are preferred. E.g., for QT <london, capital, country> ("is London the capital of any country?"), the OT <London (exact), hasCapitalCity,

Country(exact)> is preferred over <capital_city (synonym), has_capital_city, country (exact)>. This rule is similar to rule 1, but because it is applied at a later stage, it is more restrictive.

5) For "who queries", OTs formed with "person" are preferred over "organization".
6) OTs based on direct mappings (1:1 mapping between a QT and an OT) are preferred to those relying on indirect mappings (1:2 mapppings). E.g, <person, works, open university> ("who works in the open university?") is translated to both OT1= <person, memberOf, openUniversity> and to OT2 = <person, mentions-person, kmi-planet-news (subclassOf publication)>, <kmi-planet-news, mentions-organization, the-open-university>. OT1 is ranked higher than OT2.

Once the score is assigned to each answer the clusters are created as follow: C@1 is all the answers ranked highest (score 1), C@2 is all the answers ranked in position 2, and so on.

4.3 Ranking by popularity

Finally, answers are ranked according to their popularity, i.e., the number of individual ontologies from which they are derived. For instance, "where is Paris?" produces two answers: France (or French Republic) and United States (as Paris is a city in the state of Texas). In this case, France is the most popular answer across ontologies and therefore is ranked first. The number of ontologies for a given answer is provided by the merging algorithm described in section 3. An answer is C@1 if its popularity is higher than 1 (more than 1 ontology).

4.4 Ranking by combination

Finally, we propose a last strategy to improve ranking, by the combined use of all the ranking methods presented before. We argue that, due to the different nature of these approaches, relevant answers not selected and irrelevant answers not filtered by one specific method, are suitable to be selected or filtered by the others. For the combination strategy we have used the weighted Borda method [1], in which votes are weighted taking into account the quality of the source. The combined weight for the answer i within the context of the query q, Wi,q, is therefore computed as: Wi,q = 2*x (x=1, i ∈ confidence C@1) + 1*x (x=1, i ∈ confidence C@2) + 1*x (x=1, i ∈ semantic similarity C@1) + 1*x (x=1, i ∈ popularity C@1).

We have empirically tested that the most important ranking algorithm is confidence. With the proposed combination we attempt, on the one hand, to provide this measure for a significant number of answers (selecting C@1 and C@2) and, on the other hand, to provide a higher score to those answers with a higher confidence value. Once the scores are computed each answer is then clustered according to: a) its final score value and b) the selected degree of relevance for precision and recall measures in the final answer. To maximize precision, C@1 is generated with all the answers for which Wi,q > 1. To maximize recall, C@1 is generated with all the answers for which Wi,q >2.

5 Evaluation

In this section we describe the evaluation of PowerAqua's merging and ranking capabilities for queries that require to be answered by combining multiple facts from the same or different ontologies. The design of this evaluation is focused around two main questions:

a) How do we measure if the quality of the collective results obtained after merging and ranking are better than the individual answers?
b) Which datasets are more suitable to be used for this evaluation?

Because there are different steps in the merging and ranking process that can influence the final quality of the answers, we have divided the evaluation in three main stages:

- Evaluation of the efficiency and effectiveness of the fusion algorithm.
- Evaluation of the level of filtering performed by the merging algorithm over the initial set of answers retrieved by PowerAqua.
- Evaluation of the three proposed ranking algorithms applied to the final set of answers obtained after the merging process.

This initial evaluation is conducted using our own benchmark which comprises:

- *Ontologies and Knowledge Bases*: We collected around 4GBs of data stored in 130 Sesame repositories. Each repository contains one or more semantic sources. We have collected in total more than 700 documents. The dataset includes high-level ontologies, e.g., ATO, TAP, SUMO, DOLCE and very large ontologies, e.g., SWETO (around 800,000 entities and 1,600,000 relations) or the DBPedia Infoboxes (around 1GB of metadata). This set of ontologies is stored in several online Sesame repositories. Even though PowerAqua can access larger amounts of SW data through Watson, in this experiment we decided to use a substantial static dataset in order to make these experiments reproducible.
- *Queries*: We collected a total of 40 questions selected from the PowerAqua website[7] and from previous PowerAqua evaluations that focused on its mapping capabilities [10]. These are factual questions[8] that PowerAqua maps into several OTs, each of them producing partial answers. Merging and ranking is needed for these queries to generate a complete answer, or to rank between the different interpretations.
- *Judgments*: In order to evaluate the merging and ranking algorithms a set of judgments over the retrieved answers is needed. To perform this evaluation two ontology engineers provided a True/False manual evaluation of answers for each query.

The construction of this benchmark was needed due to the lack of SW standard evaluation benchmarks comprising all the required information to judge the quality of the current semantic search methods [7].

5.1 Evaluating the fusion algorithm

The gold standard for the evaluation of the merging algorithm was created by manually annotating the answer sets produced by the 40 test queries. For each answer set, subsets of identical answers were identified. The generated gold standard was compared to the fusion produced by the merging algorithm and standard precision and recall measures were calculated. Each pair of answers correctly assigned to the same subset was considered a "true positive" result, each pair erroneously put into the same subset constituted a "false positive" result, and each pair of individuals, which were assigned to different subsets, while being in the same subset in the gold standard, represented a "false negative" result. The results are shown in Table 2.

[7] http://technologies.kmi.open.ac.uk/poweraqua/fusion-evaluation.html

[8] Factual queries formed with wh-terms (which, what, who, when, where) or commands (give, list, show, tell,..) vary in length and complexity:from simple queries, with adjunct structures or modifiers, to complex queries with relative sentences and conjuctions/disjunctions

Table 2. Test results of the co-reference resolution stage

Gold standard size	Precision	Recall	F1-measure
1006	0.946	0.931	0.939

When analyzing the results we found that most errors of the merging stage were caused by:

- Syntactically dissimilar labels for which no synonyms could be obtained from WordNet, e.g: #SWEET_17874 (Longview/Gladewater), or grammatical mistakes (like "she_sthe_one" instead of "she_the_one").
- Homonymous or syntactically similar labels for different entities.
- Incorrectly modeled ontologies, which contain duplicate instances under different URIs: e.g., in SWETO the city of Houston, Texas has 5 distinct URIs. Since such errors were not caused by the merging algorithm, they were not counted during the evaluation experiments.

5.2 Evaluating the level of filtering performed by the merging algorithm

The major advantage of merging the multiple answers derived by PowerAqua is that irrelevant answers are filtered out (eliminated). The filtering obtained by the merging algorithm described in this paper allows, on the one hand to eliminate duplicated information by means of fusing redundant answers together and, on the other hand, to compose a complete answer using different subsets of partial responses. The filtering of duplicated and partial information helps to eliminate non relevant responses from the initial set of results. The following measure is used to compute the level of non relevant results filtered by the merging algorithm.

$$f_q = \frac{|R_q| - |R_{qm}|}{|R_q|}$$

Where f_q is the percentage of filtering for the query q, R_q is the set of initial results retrieved by PowerAqua for the query q and $R_{q,m}$ is the set of answers that remain after merging. Note that, for simplicity, we consider that all the eliminated answers are irrelevant. This is not necessarily true when the merging algorithm intersects partial answers. For those cases, the rate of false positives (or number of relevant results lost in the filtering process) has been computed (section 5.1) and discarded as irrelevant. Results are presented in Section 5.4

5.3 Evaluating the three proposed ranking algorithms

Here we present the evaluation of the three ranking algorithms detailed in Section 4 in terms of Precision and Recall. As the golden standard for the evaluation we consider the completed list of answers for query q including all the potential relevant and irrelevant results as the unsorted list of answers obtained after the merge step, $R_{q,m}$ (see Fig. 2 for further details). For each ranking metric we consider as retrieved list of answers for the query q the first ranked cluster (C@1). Taking into account this, we define Precision and Recall as:

$$P_q = \frac{|\{\mathrm{Re}l_q \cap C@1_q\}|}{|C@1_q|}, R_q = \frac{|\{\mathrm{Re}l_q \cap C@1_q\}|}{|\mathrm{Re}l_q|}$$

Where: P_q is precision for query q, R_q is recall for the query q, Rel_q is the set of relevant answers included in $R_{q,m}$ for the query q and $C@1_q$ is the set of retrieved answers, or answers included in the first ranked cluster.

Once these measures have been defined we compare the results obtained by our three different ranking metrics against our baseline, $R_{q,m}$. For the ranking based on confidence the precision is computed not just for the first ranked cluster C@1 but also for the union of the first

two clusters C@1 U C@2. As explained in Section 4.4, the most accurate ranking algorithm is confidence, therefore both confidence clusters are used in the combined ranking.

5.4 Results

The results of our experiments are reported in Tables 3 and 4 for the 40 selected queries. Table 3 contains the queries merged by union while Table 4 contains the results for the queries merged by intersection and condition. The different columns of the table represent:

1) The type of merging done for that query (U=union, I=intersection, C=condition) / the number of ontologies involved in the merging process.
2) The percentage of irrelevant queries filtered by the merging algorithm.
3) The precision obtained for the set of answers returned after the merging process (the baseline ranking).
4) The error type as explained below.
5) Precision/Recall measures for the confidence ranking at the level of the first cluster C@1.
6) Precision/Recall measures for the confidence ranking at the level of the first two clusters C@1 U C@2.
7) Precision/Recall measures for the popularity ranking at the level of the first cluster C@1.
8) Precision/Recall measures for the semantic similarity ranking at the level of the first cluster C@1.
9) Precision/Recall measures for the combined approach at the level of the first cluster C@1 with the target of optimizing recall.
10) Precision/Recall measures for the combined approach at the level of the first cluster C@1 with the target of optimizing precision.

An empty sets {} represent that no answer was retrieved for that cluster while – indicates that the query generates only 1 unique answer after merging, and therefore there is nothing to rank.

As we can see in the tables, the merging component is able to filter an average of 93% of irrelevant answers for intersections/conditions and 32% for unions. For instance, in Q14: *find me university cities in Japan*, 20 final answers are selected out of 991 partial answers, by intersecting 417 from cities with a university from *dbpedia ontology* and 574 from cities in Japan from *fao*, *ato*, *KIM*, *tap*, *SWETO*. The average recall of the fusion algorithm, as shown in Section 5.1, is 0.93, i.e., a 0.07 loss in recall occurs in the case of intersections/conditions when partial answers representing the same individual are not recognized. The average precision of the fusion algorithm is 0.94, which indicates that most of the answers are correctly fused. The high precision and recall values obtained for the fusion algorithm, as well as the high percentage of filtering of irrelevant answers performed by this method, reflect PowerAqua's ability to derive valid semantic interpretations to a query across ontologies.

The causes of the merging algorithm leading to irrelevant results in the final answers are:

- Incorrect modeling of the ontological elements in the OTs that lead to the answer (M). For instance in Q30: *what mountains are in Alaska?,* the instance Germany is given as an answer because it is defined as rdf:type {country, mountain} in one of the ontologies.
- An inaccurate semantic interpretation given by PowerAqua (I). For instance Q36: *who belongs to the Open University?*. Among OTs representing people that work for the Open University, there is an OT : <organization, type, open universities>.
- Retrieval of irrelevant answers (R). E.g., the answer Houston to Q29: *Where is Houston?*

These sets of errors are often filtered out afterwards by the ranking algorithms. As we can see in the tables, for the union queries all ranking methods are able to provide better precision than the baseline, with an increase of 0.22 points of precision for the best ranking algorithm, in this case ranking by confidence at C@1. This increase in precision is usually translated in a recall

loss as in the case of the popularity ranking algorithm where recall drops to 0.31. However, the rest of the ranking metrics are able to keep the recall measure between 0.77 and 0.94. Finally, the best combined approach is able to enhance with 0.12 points the precision of the baseline without causing a drop in recall.

5.4 Discussion on the Results

For the intersection and condition queries all the ranking methods are able to keep or increase the precision, except in the case of the popularity algorithm that decreases precision to 0.31 points. The same effect occurs with recall. All the ranking algorithms are able to provide levels of recall between 0.98 and 1, which means nearly no loss of relevant answers, except for the popularity ranking, which reduces recall to 0.54. The best ranking method for intersection and condition queries is the ranking by confidence at C@1. This ranking slightly increases precision with 0.03 points with respect to the baseline, keeping at 0.98 the level of recall. Finally, for this set of queries, the best combined approach is able to preserve the same precision and recall values as the baseline: 0.96/1. In other words, the effect of ranking measures on intersection queries is neutral, this was expected as for intersection and condition queries the filtering has already eliminated most (if not all) of the inaccurate answers.

In summary, we can say that the best ranking method for both subset of queries is the ranking by confidence at C@1 that is able to produce a 0.22 percentage increase of precision for union queries and a 0.03 for intersection ones. Semantic similarity depends on being able to calculate the semantic interpretation of each OT, but that's not the case if the OT entities are not covered in WordNet, or the taxonomical information is not significant enough to elicit the meaning of the entity in the ontology. The worst ranking method in both cases is ranking by popularity. It drops precision by 0.14 points for union queries and by 0.31 points for intersection and condition queries. This is because popularity at C@1 (answers obtained from at least two ontologies) is empty in the cases in which no answers were fused from different ontologies (empty set {} being equivalent to 0/0 for precision/recall). Interestingly, in the 25 cases where C@1 is not empty, this measure gives precision 1 in 22 cases. Therefore, precision would have been closer to 1 than with any other ranking if we would have chosen to C@1 all the answers with popularity 1, when there are not answers with popularity 2 or higher (empty set {} equivalent to 1/1 for precision/recall as all the answers are rank at the same level). The effect in recall is even worse, dropping to 0.31 for union queries and to 0.54 for the intersection and condition ones. At this early stage of the SW, PowerAqua's results are hampered by the knowledge sparseness and its low quality. We believe that any extension of the online ontologies and semantic data will result in direct improvements for both popularity and semantic similarity ranking measures.

Even with the different behavior of these ranking methods, the combined algorithm is over-performed by the confidence ranking in terms of precision but it is able to improve the precision and recall ratio. Contrary to what was expected, maximizing precision does not improve the precision value on the combined measure. This is because the average measure was affected by queries in which none of the answers ranked high enough (C@1={}).

These results confirm our initial hypothesis that the use of cross-ontological information to rank the retrieved answers helps to enhance the precision of the results, and therefore, to provide, from the wisdom of semantic crowds, better answers to users. An important remark is that, this increase of precision does not imply, in any case except for the popularity algorithm, a significant loss in recall.

Table 3. Test results for union queries

	Type /N° Ont.	Fil-ter	Baseline		P/R confidence		P/R Pop	P/R Sem. Sim.	P/R Combined	
			P	Er	C@1	C@1 U C@2			+ R	+ P
Q_1	U / 3	0.03	0.97	M	0.97/0.88	0.97/1	1/0.03	0.97/1	0.97/0.88	1/0.03
Q_2	U / 3	0.53	1	-	1/0.66	1/1	1/0.73	1/1	1/1	1/0.97
Q_5	U / 4	0	1	-	1/0.4	1/0.6	{}	1/0.97	1/0.97	1/0.97
Q_6	U / 4	0.27	0.77	R,I	0.66/0.4	0.5/0.4	1/0.4	0.77/1	0.77/1	{}
Q_7	U / 6	0.38	1	-	1/0.53	1/1	1/0.29	1/1	1/1	1/0.52
Q_{13}	U / 2	0.91	1	-	1/0.81	1/1	1/0.08	1/1	1/0.81	1/0.09
Q_{15}	U / 5	0.55	1	-	1/0.47	1/1	1/0.35	1/1	1/1	1/**0.47**
Q_{16}	U / 7	0.31	1	-	1/1	1/1	1/0.32	1/1	1/1	1/1
Q_{17}	U / 8	0.25	0.35	I,M	1/1	1/1	1/0.14	0.53/1	1/1	1/1
Q_{18}	U / 3	0.13	1	-	1/1	1/1	1/0.13	1/1	1/1	1/1
Q_{19}	U / 8	0.55	0.94	I	1/1	0.94/1	1/0.61	0.94/1	0.94/1	1/1
Q_{20}	U / 2	0.05	1	-	1/1	1/1	1/0.05	1/1	1/1	1/1
Q_{21}	U / 4	0.53	0.75	R	1/1	0.75/1	1/0.16	1/1	0.75/1	1/1
Q_{22}	U / 6	0.36	0.28	I	1/1	0.66/1	0/0	0.16/0.5	0.5/1	0.5/0.5
Q_{23}	U / 4	0.4	0.5	R	1/1	1/1	0.5/1	0/0	0.5/1	0.5/1
Q_{24}	U / 8	0.57	0.12	R	1/1	0.12/1	1/1	0/0	0.12/1	1/1
Q_{25}	U / 10	0.64	0.88	I,M	1/1	0.96/1	0.96/0.98	0.9/1	0.95/1	0.98/1
Q_{29}	U / 9	0.21	0.45	R	0.45/1	0.45/1	1/0.4	0.4/0.8	0.45/1	0.45/1
Q_{30}	U / 3	0.08	0.95	M	1/1	0.95/1	1/0.09	0.95/1	0.95/1	1/1
Q_{31}	U / 3	0	1	-	1/0.5	1/1	{}	{}	1/0.5	1/0.5
Q_{34}	U / 3	0.29	0.74	I	1/0.07	0.74/1	1/0.07	0.74/1	0.74/1	1/0.07
Q_{36}	U / 5	0.15	0.14	I	1/0.32	0.14/1	1/0.02	1/0.77	1/0.77	1/0.32
Q_{37}	U / 3	0.5	0.66	R	1/1	0.66/1	1/0.5	0.66/1	1/1	1/0.5
Q_{38}	U / 6	0.3	0.57	M,I	1/0.25	1/0.5	1/0.5	0.57/1	0.75/1	1/0.5
Q_{39}	U / 2	0	0.33	I	1/1	0.33/1	{}	0.33/1	1/2	{}
Avg	**4.84**	**0.32**	**0.74**		**0.96/ 0.77**	**0.81/ 0.94**	**0.82/ 0.31**	**0.72/ 0.85**	**0.86/ 1**	**0.86/ 0.66**

Table 4. Test results for intersection and condition queries

	Type /N° Ont.	Fil-ter	Baseline		P/R confidence		P/R Pop	P/R Sem. Sim.	P/R Combined	
			P	Er	@1	C@1 U C@2			+ R	+ P
Q_3	I / 3	0.96	1	-	-	-	-	-	-	-
Q_4	I / 3	0.93	1	-	-	-	-	-	-	-
Q_8	I / 3	0.97	1	-	1/1	1/1	{}	1/1	1/1	1/1
Q_9	C / 4	0.96	0.88	I	1/0.87	0.88/1	{}	0.88/1	0.88/1	1/0.87
Q_{10}	C / 3	0.92	1	-	1/1	1/1	{}	1/1	1/1	1/1
Q_{11}	C / 13	0.88	1	-	1/0.77	1/1	1/0.07	1/1	1/1	1/0.77
Q_{12}	C / 4	0.78	0.75	M	1/1	0.75/1	{}	0.75/1	0.75/1	1/1
Q_{14}	I / 8	0.98	1	-	1/1	1/1	**1/1**	1/1	1/1	1/1
Q_{26}	I / 2	0.72	0.97	I	0.97/1	0.97/1	1/0.03	0.97/1	0.97/1	0.97/1
Q_{27}	I / 6	0.99	0.83	R	0.83/1	0.83/1	0.83/1	0.83/1	0.83/1	0.83/1
Q_{28}	I / 1	0.99	1	-	-	-	-	-	-	-
Q_{32}	I / 1	0.99	1	-	-	-	-	-	-	-
Q_{33}	I / 2	0.84	1	-	1/1	1/1	{}	1/1	1/1	0/0
Q_{35}	I/ 1	0.98	1	-	-	-	-	-	-	-
Q_{40}	C / 7	0.99	1	-	-	-	-	-	-	-
Avg	**4.06**	**0.93**	**0.96**		**0.99 /0.98**	**0.96 /1**	**0.65/ 0.54**	**0.96/ 1**	**0.96/ 1**	**0.92/ 0.91**

6 Related Work

The problem of retrieving information by means of the aggregation of data from different sources on the Web has been tackled in Sig.ma (http://sig.ma/). In this system the user enters a keyword and is able to explore all the aggregated data coming from the search engine Sindice. Their contribution at this early stage of the SW is to show that "the sum is really bigger that the single parts". The system uses large scale indexing, data aggregation heuristics, and ontology

alignments for automatic semi-structured data discovery and consolidation. However, as opposed to our approach, sig.ma does not attempt to automatically disambiguate or rank between different interpretations.

More specifically, the problem of merging, or finding identical individuals, was mostly considered in the context of offline data fusion scenarios. Basic similarity metrics based on string comparison were developed in the database community (e.g., [16, 3]). These metrics are used as a basis for the majority of algorithms, which compare values of attributes of different data instances and aggregate them to make a decision about two instances referring to the same entity (see [6] for a survey). The main distinction of our work is that, in the PowerAqua scenario, the fusion of answers is done in real time.

The problem of ranking applied to semantic search results has been also addressed in the literature. Among these works we can highlight [10, 11, 13]. [10] provides a criterion for query result ranking in the SEAL Portal based on a similarity measure between query results and the original KB without axioms. [11] proposes the expansion of query results through arbitrary ontology relations starting from the initial query answer, where the distance to the initial results is used to compute a similarity measure for ranking. [13] proposes a sentence ranking scheme based on the number of times an instance appears as a term in a relation type, and the derivation tree by which a sentence is inferred. To our knowledge, none of these works is applied to results derived from different knowledge sources, therefore, they do not consider the so-called "wisdom of the crowds" paradigm within their ranking algorithms.

7 Conclusions and Future Work

In this work we present a set of merging and ranking algorithms that aim to integrate information derived from different knowledge sources in order to enhance the results obtained by semantic search systems. These algorithms have been integrated and tested in an open QA system, PowerAqua. The experiments are promising, showing that the ranking algorithms can exploit the increasing amount of collectively authored, highly heterogeneous, online semantic data, in order to obtain, more accurate answers to a questions. On the one hand, the merging algorithm is able to filter out a significant subset of irrelevant results. On the other hand, the ranking algorithms are able to increase the precision of the final set of, thus showing a deeper semantic "understanding" of the intent of the question.

The merging algorithm is able to filter out up to 91% (32% on average) for union-based queries, and up to 99% (93% on average) for intersection based queries.

The best ranking algorithm (ranking by confidence) is able to obtain an average of 96% precision for union queries and 99% for intersection queries. An interesting, observed, side effect of this approach is that, answers to some questions that are distributed across ontologies can only be obtained if the partial results are merged. In this case, the introduction of the merging algorithm provides PowerAqua with the capability to answer queries that cannot be answered when considering a single knowledge source.

The high precision values produced by the merging and ranking algorithms, that are responsible for amalgamating information from different sources, support the comparison with the idea of the *Wisdom of Crowds* that we suggested in the paper. We further observe that it is known that the Wisdom of crowds only works if the crowd is diverse and free to think independently[14], allowing it to converge on good solutions. Similarly PowerAqua works well where ontologies have different emphasis, to allow the assembly of composite answers, but also overlaps between ontologies exist, to allow mapping and identification of ranking criteria, such as popularity. Both too much homogeneity and isolated "silo" ontologies would weaken our approach.

Another interesting side-effect of this approach is that, apart from the obvious advantage to the final user, the filtering of negative results and the ranking capabilities of the retrieval system increase its adaptability for other tasks, e.g., query expansion using SW resources.

An issue remains nonetheless open: the use of our own dataset to perform the experiments. However, to our knowledge, the SW community has not yet proposed standardized benchmarks to evaluate semantic merging and/or ranking evaluation. Despite this fact we have tested our algorithms with a significant amount of queries and large amounts of distributed semantic metadata (around 4GB).

Finally, we are currently working on a trust propagation mechanism where the user can rank the answers as a way of giving feedback to the system. We believe that this mechanism will further improve the ranking so that answers replicated across many ontologies do not bias less frequently occurred facts generated from specialist knowledge from trusted ontologies.

References

1. Bartell, B. T. (1994). Optimizing Ranking Functions: A Connectionist Approach to Adaptive Information Retrieval. PhD thesis, University of California, San Diego
2. Bernstein, A., Kaufmann, E. (2006). GINO- A Guided Input Natural Language Ontology Editor. In Proc of the ISWC
3. Bilenko, M., Mooney, R. J.: Adaptive duplicate detection using learnable string similarity measures. KDD-2003, 39—48 (2003)
4. Cimiano, P., Haase, P., Heizmann, J. (2007). Porting Natural Language Interfaces between Domains- An Experimental User Study with the ORAKEL System. In Proc of the Int Conf on Intelligent User Interfaces
5. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In Proc of the 40th Anniversary Meeting of the Association for Computational Linguistics (2002).
6. Elmagarmid A. K., Ipeirotis, P. G., Verykios, V. S.: Duplicate record detection: a survey. IEEE Transactions on Knowledge and Data Engineering 19(1), 1—16 (2007)
7. Fernandez, M., Lopez, V., Motta, E., Sabou, M., Uren, V., Vallet, D., Castells, P. Using TREC for cross-comparison between classic IR and ontology-based search models at a Web scale. Semantic search workshop at WWW 2009.
8. Gracia, J., Lopez, V., d'Aquin, M., Sabou, M., Motta, E., Mena, E. (2007). Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching. In Proc of the Ontology Matching Workshop at ISWC/ASWC
9. Lopez, V., Sabou, M., Uren, V., Motta, E. (2009). Cross-Ontology Question Answering on the Semantic Web –an initial evaluation. In Proc. of Knowledge Capture Conference.
10. Maedche, A., Staab, S., Stojanovic, N., Studer, R., and Sure, Y.: SEmantic portAL: The SEAL Approach. In: Fensel, D., Hendler, J. A., Lieberman, H., Wahlster, W. (eds.): Spinning the Semantic Web. MIT Press, Cambridge London (2003) 317-359
11. Rocha, C., Schwabe, D., and de Aragão, M. P.: A Hybrid Approach for Searching in the Semantic Web. In Proc. of the 13th International World Wide Web Conference (WWW 2004), NY (2004).
12. Sais, F., Pernelle, N., Rousset, M.: L2R: a logical method for reference reconciliation. AAAI-07. 329—334 (2007)
13. Stojanovic, N., Studer, R., and Stojanovic, L.: An Approach for the Ranking of Query Results in the Semantic Web. In: Fensel, D., Sycara, K. P., Mylopoulos, J. (eds.): The Semantic Web ISWC 2003, 2nd International Semantic Web Conference. Lecture Notes in Computer Science, Vol. 2870. Springer Verlag, Berlin Heidelberg (2003) 500-516.
14. Suroweicki J., The Wisdom of Crowds, Doubleday, New York (2004)
15. Tablan, V, Damljanovic, D., and Bontcheva, K. (2008). A Natural Language Query Interface to Structured Information. In Proc of the ESWC
16. Winkler, W.: The state of record linkage and current research problems. US Bureau of the Census Technical Report RR99/04. (1999)