OpenKnowledge

FP6-027253

D8.5 - Evaluation of Semantic Web Applications

Vanessa Lopez, Davide Guidi, Enrico Motta, Silvio Peroni,
Mathieu d'Aquin, and Laurian Gridinoc
{v.lopez, d.guidi, e.motta, s.peroni, m.daquin, l.gridinoc}@open.ac.uk

Knowledge Media Institute,
The Open University, UK

Report Version: Final

Report Preparation Date: 24/10/2008

Classification: Deliverable D8.5

Contract Start Date: 1.1.2006

Duration: 36 months

Project Co-ordinator: University of Edinburgh (David Robertson)

Partners: IIIA(CSIC) Barcelona
Vrije Universiteit Amsterdam
University of Edinburgh
KMi, The Open University
University of Southampton
University of Trento

As discussed in an earlier project deliverable (D8.4), the integration between the Semantic Web applications developed in OpenKnowledge, PowerAqua and PowerMagpie, and the OpenKnowledge peer to peer architecture has been realized at the level of the Watson Semantic Web Gateway. In other words, a common infrastructure, produced by decentralizing Watson over a peer to peer network, has been developed in the project and can be accessed transparently by any Watson-centric application. The aim of this deliverable is then to assess this infrastructure. In particular we carry out a task centric comparison between this infrastructure and the 'standard', centralized version of Watson. We perform this comparison in the context of PowerAqua, although some insights about the issues associated with the integration of PowerMagpie with the OpenKnowledge architecture is provided as well. The evaluation shows that the integration of Watson with the OpenKnowledge architecture can be used effectively to realize "Cautious Knowledge Sharing" scenarios. Specifically, the use of automatic key concept extraction has negligible impact on recall, while spreading ontologies over a network of Watson peers improves the overall efficiency of the system.

1. Introduction

In earlier project deliverables, D8.2 [1] and D8.4 [2], we described an initial integration of the Semantic Web (SW) applications, PowerAqua [3] and PowerMagpie [4], with the OpenKnowledge (OK) peer to peer infrastructure. This 'vanilla integration' was achieved by decentralising the Watson Semantic Web Gateway [5] over the OpenKnowledge peer to peer architecture and by creating appropriate interaction models to support the broadcasting of queries from *application peers* to nodes in the network – i.e., to the *mini-Watsons*.

In this deliverable we describe a second integration scenario, which realises a much tighter integration between the OK infrastructure and the SW applications, which we call "Cautious Knowledge Sharing". In this scenario, each node in the network only advertises a subset of the knowledge it contains and queries to the network are routed through a *DescriptionRepository*, which only sends them to those peers, which can in principle answer them. Having realised this second integration scenario, we have evaluated it, by comparing its performance with that achieved using the standard, centralised Watson repository, which is publicly available at http://watson.kmi.open.ac.uk/. In our evaluation we have focused on a PowerAqua-centric scenario, however some information about the issues associated with the integration of PowerMagpie with the OpenKnowledge architecture is provided as well.

The deliverable is structured as follows. In Section 2 we describe the technical realization of the "Cautious Knowledge Sharing" integration scenario. In Section 3 we briefly highlight the key features of PowerAqua and we detail the specific system used for the evaluation. Section 4 provides a detailed description of the evaluation, as well as the main findings that have come out of the experiment. Finally we draw the main conclusions from this work in Section 5 and we highlight the next steps we intend to take to explore this integration scenario further.

2. Cautious Knowledge Sharing

2.1 The scenario

In a "Cautious Knowledge Sharing" scenario we have a number of knowledge providers as well as client applications, which are looking for information. This may be possessed by one or more of the knowledge providers. Each provider retains full control over what information is made available to an application and indeed different applications may have different access rights to the various sources or even within a source. For efficiency reasons, we also want to minimize the requests for information by avoiding broadcasting a query to peers that are unlikely to be able to answer it. In our experiments the application requiring knowledge is PowerAqua, while the Watson Peers are the knowledge providers.

2.2 Architectural details

We implemented the "Cautious Knowledge Sharing" scenario by adding new functionalities to the 'vanilla' integration scenario described in D8.2 and D8.4. In the 'vanilla' scenario an application, e.g., PowerAqua, interacts with a dedicated *WatsonApplication Peer* to access the knowledge provided by one or more *WatsonPeers*. Requests for knowledge by the WatsonApplication Peer are simply broadcast to every WatsonPeer. In the "Cautious Knowledge Sharing" scenario we improve this architecture in several ways, as reported in what follows.

2.2.1 Key concept extraction and advertising

The first functionality we have introduced is the *key concepts detection facility*. Key concepts are representative concepts of an ontology, which can be used to summarize it. In other words, they provide the *signature* of the ontology. In order to exploit them, we applied the key concept detection algorithm, described in [6], to the collection of ontologies used for the PowerAqua evaluation. This collection contains 25 sets of potentially overlapping ontologies, whose domains include organizational ontologies, medical domains, publication references, and several others. The size of the collection is 3 GB. In order to be used by the Watson Peers, these ontologies were further analysed, indexed and finally converted to a series of 25 indexes (9.2GB in total). Applying the key concept detection algorithm to these indexes we produced 25 files (whose total size is less then 100KB), which summarize the content of each ontology in the whole collection. For example, the following 20 key concepts were extracted automatically from the TAP ontology, which contains 5622 classes:

> { **sport**, **animal**, **person**, **material**, **computer**, **television**, **place**, **product**, **software**, **travel**, **furniture**, **magazine**, **account**, **food**, **electronics**, **telephone**, **number**, **event**, **brand**, **sedan**}

As a result, each WatsonPeer contains its own index together with the extracted key concepts. These key concepts are then advertised, enabling the WatsonApplication to select which peers to contact for any given query. This mechanism makes it possible to advertise ontologies 'cautiously', without giving away too much information about the contents of the ontologies.

A number of architectural solutions were considered for realizing this idea. At the end we opted for a solution which uses an extra component, called *DescriptionRepository*, which has been added to the architecture in addition to the WatsonPeers and WatsonApplication. The role of the DescriptionRepository is to store the descriptions (the key concepts) of any ontology hosted in any WatsonPeer.

To advertise the key concepts to the DescriptionRepository, the WatsonPeer starts an interaction, which is described by the following LCC code:

```
// WatsonPeer Registration

r( watsonPeer, initial, 1)
r( descriptionRepository, necessary, 1 )


a( descriptionRepository, R ) ::
register(D) <= a(watsonPeer, W)
then
null <- registerWP(D, W) // call registerWP for the endpoint W

a( watsonPeer, P ) ::
register(D) => a(descriptionRepository, R) <- getDescription(D)
```

Figure 1. LCC code for WatsonPeer registration

As shown in Figure 1, the WatsonPeer is the initiator of the interaction. The interaction itself is very simple: the WatsonPeer sends its description to the DescriptionRepository, which stores it internally. The DescriptionRepository stores also the IDs of the WatsonPeers participating in the interactions.

2.2.2 WatsonPeer selection and querying

Once WatsonPeers have been registered in the DescriptionRepository, the WatsonApplication peer can request information. In this scenario, requesting information involves two phases:

1. Identifying a subset of WatsonPeers that can potentially contain information related to the specific domain addressed by the query;
2. Sending the query to this subset and retrieving the responses.

These two phases are independent, in the sense that a WatsonApplication can send many queries related to the same domain, thus interacting several times with the same subset of WatsonPeers identified in the first phase. Technically, the first phase is executed by running the *startInteraction()* method provided by the WatsonApplication class. The query is sent every time PowerAqua requests a function of the Watson API, such as *searchWatsonWithKeywords*, through the WatsonApplication.

In the case of PowerAqua, applying these phases is straightforward. The identification of the WatsonPeers –i.e., the execution of the startInteraction() function, is done as soon as the user sends a query. Any subsequent call to the Watson API sent by WatsonApplication will then use this selected set of peers, until the user sends a new query.

In addition, as discussed later in more detail, when PowerAqua sends a query, the synonyms of the words used in the query, together with their singular and plural forms, are retrieved from WordNet. This set of keywords is then sent to the DescriptionRepository, which matches them with its copy of the WatsonPeer descriptions, and returns a list of relevant WatsonPeers. The WatsonApplication then exploits this list providing access to the Watson API. The whole LCC Interaction used for selection and query of WatsonPeers is shown in Figure 2.

```
// WatsonPeer Selection and Query

r( application, initial, 1)
r( watsonPeer, necessary, 1,1000)
r( manageQueries, auxiliary, 1)
r( forwarder, auxiliary, 1)
r( receiver, auxiliary, 1)
r( descriptionRepository, necessary, 1 )

a( application, WA ) ::
  null <- getKeywords(K)
  then
  getMatchingPeers(K) => a(descriptionRepository, DR)
  then
  setMatchingPeers(MatchingPeers) <= a(descriptionRepository, DR)
  then
  null <- printPeerList(MatchingPeers)
  then
  a( manageQueries(MatchingPeers), WA)

a( manageQueries(MatchingPeers), WA) ::
  null <- getOperation(X)
  then
  null <- X = [] and printTimings(X)
  or
  (
    a(forwarder(MatchingPeers, X), WA)
    then
    null <- getEmptyList(Results) and getTime(T) and size(
MatchingPeers, RemainingPeers)
    then
    a( receiver(T, Results, FinalResults, RemainingPeers), WA)
    then
    null <- finalResults(FinalResults)
    then
    a( manageQueries(MatchingPeers), WA)
  )

a( forwarder(MatchingPeers, X), WA) ::
  null <- MatchingPeers = []
  or
  (
    ask(X) => a( watsonPeer, WP) <- MatchingPeers = [WP|Tail]
    then
    a( forwarder(Tail, X), WA)
  )
```

```
  a( receiver(T, Results, FinalResults, RemainingPeers), WA) ::
  null <- equalZero(RemainingPeers) and unify(Results, FinalResults)
  or
    (
        (
        tell(Y) <= a( watsonPeer, WP)
        then
        null <- appendValues(Results, Y, PartialResult) and
dec(RemainingPeers, PartialRemainingPeers)
        then
        a( receiver(T, PartialResult, FinalResults,
PartialRemainingPeers), WA)
        )
      or
        (
        null <- timeExpired(T) and unify(Results, FinalResults)
        or
          (
          null <- waitOneSecond(T, NewT)
          then
          a(receiver(NewT, Results, FinalResults, RemainingPeers),WA)
        )
      )
    )

  a( watsonPeer, WP) ::
    ask(X) <= a( forwarder, WA) then
    tell(Y) => a( receiver, WA) <- execute(X, Y)
    then
    a( watsonPeer, WP)

  a( descriptionRepository, DR) ::
    getMatchingPeers(K) <= a(application, WA)
    then
    setMatchingPeers(MatchingPeers) => a(application, WA) <-
getPeers("watsonPeer", L) and getMatchingPeers(L, K, MatchingPeers)
```

Figure 2. Interaction for selection and query of WatsonPeers

The interaction shown in the previous LCC code is based on the interaction originally built for the 'Vanilla' scenario. The main change here is the introduction of a DescriptionRepository component, contacted at the beginning of the interaction by the WatsonApplication to retrieve the list of selected WatsonPeers.

2.2.3 Discussion and results

The integration scenario described in the previous section exploits as much as possible the OpenKnowledge Kernel. The two phases, peer selection and query, for example, have been compacted in just one LCC interaction. This is possible because the DescriptionRepository returns a list of WatsonPeers composed of *EndPointIds*.

EndPointIDs are the objects that represent OpenKnowledge peers and can be interpreted directly at the LCC level.

Being tightly integrated with the OpenKnowledge Kernel, however, has also drawbacks. One of the main issues for applications such as PowerMagpie or PowerAqua, is obviously performance and, necessarily, the OpenKnowledge Kernel introduces another overhead layer in the process, compared to the standard scenario, where applications interact directly with the Watson system. For this reason, during the development of this second integration, we worked closely with the kernel developers to improve its performance. The results of our tests are shown in Table 1. For this experiment we set up 25 WatsonPeers on 3 different computers and we run the query "Which wines are sweet?". Only 2 WatsonPeers were automatically selected by the DescriptionRepository and all of them were matched using the keyword "wine".

Experimental setup:	25 WatsonPeers running on 3 different computers
Query:	"Which wines are sweet?"
WatsonPeers matched:	wine_and_food, www_ontologies, (keyword matched: "wine")
Results:	Total time for accessing Watson API using OK: 38.368 secs Total number of API access: 184 Average time for each API access: 206 ms

Table 1. Experimental results obtained with the second integration scenario

These results represent a major step forward, in comparison with the results published in the earlier deliverable, D8.4. Specifically, the average time for each API access has dropped from 900 ms to almost 200 ms. Having said so, even considering these improvements, PowerAqua still needs 38 seconds to reply to this question.

More complex queries present some problems due to the recursion patterns used in the LCC code. For complex queries PowerAqua may need to invoke the Watson API thousands of times, and the use of recursion can produce an out of memory condition. On the other hand, we cannot avoid recursion because large numbers of consecutive interactions can also cause problems to the OK kernel.

2.2.4 Hybrid integration

Given the aforementioned problems, we decided to implement a hybrid integration approach using plain TCP communication. The key idea here is to use the OK Kernel to select the WatsonPeers, as before, and then to contact the WatsonPeers directly using TCP. The resulting architecture provides much faster performance, because unnecessary overheads in the interaction process are avoided.

This hybrid integration shares the same LCC code shown in Figure 1, which is used for registering the WatsonPeers in the DescriptionRepository. The LCC code used for the selection phase, shown in Figure 3, is instead just a fraction of the one used before.

```
// Watson Selection - Hybrid integration

r( application, initial, 1)
r( descriptionRepository, necessary, 1 )

a( application, WA ) ::
  null <- getKeywords(K)
  then
  getMatchingPeers(K) => a(descriptionRepository, DR)
  then
  setMatchingPeers(MatchingPeers) <= a(descriptionRepository, DR)
  then
  null <- setPeers(MatchingPeers)
  then a( application, WA)

a( descriptionRepository, DR) ::
  getMatchingPeers(K) <= a(application, WA)
  then
  setMatchingPeers(MatchingPeers) => a(application, WA)
      <- getMatchingPeers(K, MatchingPeers)
  then
  a( descriptionRepository, DR)
```

Figure 3. LCC code for WatsonPeer selection – Hybrid version

The key difference here is that the main actors are the WatsonApplication and the DescriptionRepository. Once the WatsonApplication retrieves the list with the selected WatsonPeers, it contacts them directly, using TCP communication.

The results of the same experiment run with the revised architecture are shown in Table 2. The total time for accessing the Watson API has now dropped from 38 seconds to less than 16, while the average time for each API access has decreased from 200 to 85 ms.

Experimental setup:	25 WatsonPeers running on 3 different computers
Query:	“Which wines are sweet?”
WatsonPeers matched:	wine_and_food, www_ontologies (keyword matched: “wine”)
Results:	Total time for accessing Watson API using OK: 15.787 secs Total number of API access: 184 Average time for each API access: 85.7 ms

Table 2. Experimental results obtained with the hybrid integration

3. PowerAqua

3.1 PowerAqua in a nutshell

PowerAqua [5] is a natural language (NL) front-end, which supports query answering in large scale, heterogeneous data sources, such as those that make up the Semantic Web. For instance, through its interface to Watson, PowerAqua can take as input a question expressed in natural language and return all the answers to the question that can be found anywhere on the Semantic Web (or at least in the portion of the Semantic Web, which has been crawled and indexed by Watson).

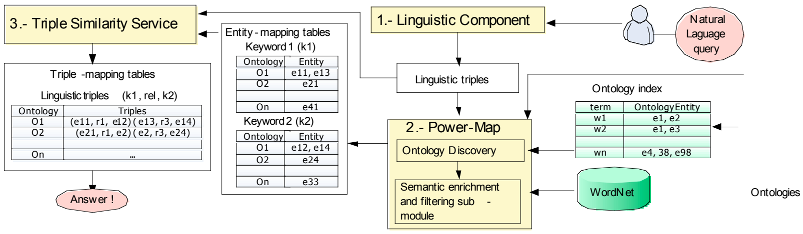

Figure 4: PowerAqua Components.

PowerAqua consists of three main components, as shown in Figure 4. Firstly, its linguistic component (detailed in [5]) uses GATE [7] to translate a NL query into its linguistic triple form <*query term, relation, term*>. This is achieved by identifying triple associations that relate terms together through verbs and prepositions. For instance, the query *"which are the members of the rock group Nirvana?"* is translated to <*what-is*, *members, rock group nirvana*>. Secondly, the *PowerMap* algorithm [8] [9] maps the terms of each linguistic triple to semantically relevant ontology entities by exploring all available ontologies offered by its back-end – e.g., by Watson. Finally, the PowerAqua *triple similarity service* examines the candidate ontologies in detail, to generate the answers (i.e., the triples) that satisfy the user's query.

3.2 PowerAqua as part of the OK platform – details about the demo system

In this section we describe the demo system we used for the experiments described in Section 4. In total we collected 660 ontological documents that we stored and indexed in the centralized Watson. The collection includes high level ontologies, like ATO, TAP, SUMO, DOLCE, as well as very large ontologies like SWETO and SWETO_DBLP, which contain around 800.000 entities and 1.600.000 relations[1]. The total size of this collection is 3GB. As already pointed out, we organized these 660 ontological documents into 25 partitions or sets of ontologies (25 indexes whose size is 9.2 GBs in total). These were then associated with 25 WatsonPeers, covering distinct but in some cases overlapping domains.

The WatsonPeers run on 3 virtual machines with the following specification: Intel processor 3Ghz, 2 GB RAM, 10 GB Hard disk. A second computer with similar characteristics is used to run the *DiscoveryService*, which is the core OK component

[1] These ontologies are also stored in 130 Sesame repositories available online in http://kmi-web07.open.ac.uk:8080/sesame

used to support communication among peers, the DescriptionRepository and PowerAqua itself. Note that due to the use of firewalls these computers have to be on the same physical network. The final implementation used for the evaluation is the hybrid integration described in section 2.2.4.

As mentioned earlier, in order to discover the relevant peers for a given query, we generate a set of relevant keywords for the query taken from the terms and relations on the linguistic triples. We also add into the set of relevant keywords the singular, plurals, the WN lemma (if any) and the WN synonyms (if any) of each linguistic keyword. In case the keyword is a compound we also tokenize it. These keywords are then sent to the DescriptionRepository that contains the WatsonPeer descriptions or signatures. To maximize recall, if a peer contains at least one of these keywords in its signature, then it is considered relevant. For our data, and our evaluation, this strategy has worked effectively, however it can easily be changed. For example, if we want to speed up performance by minimizing the number of relevant peers retrieved, we can adopt a strategy in which we select only the peers that best cover the set of keywords (and therefore the linguistic query). Conversely, if we want to try and increase recall we can add hypernyms and hyponyms to the set of relevant keywords.

4. Evaluation

An evaluation of PowerAqua focuses on its capability to answer queries by relying on information provided by multiple ontologies, which are identified on the fly during the query answering process. Additionally, grounding this task-centric evaluation in the "Cautious Knowledge Sharing" scenario adds two new dimensions to the evaluation task. The first one examines whether the mechanism[2], based on the DescriptionRepository, which automatically locates the peers that may answer a user query has any negative impact on recall, compared to the results obtained on the same data with the standard (i.e., centralized) Watson back-end. The second one compares the overall performance (measured in terms of the criteria described in Section 4.1.1) of the decentralized setup with that exhibited by the centralized one.

The experimental setup is explained in Section 4.1 and the analysis of results in Section 4.2.

4.1 Experimental Setup

4.1.1 Evaluation criteria

Our goal is to build a system that provides a correct answer to a query, in a reasonable amount of time, by making use of at least one ontology (peer) that provides the required information. Therefore, *accuracy, success*, and *speed* define the major evaluation criteria.

Accuracy is calculated as the percentage of correctly answered questions from a given corpus of questions. An answer is characterized as correct with respect to a query over an ontology or set of ontologies. In order for an answer to be correct,

[2] In the rest of the document we will use the term "Enhanced Discovery Service" (EDS), to refer to the key concept-based mechanism for routing queries to peers.

PowerAqua has to align the vocabularies of both the query and the answering ontologies. Therefore, a valid answer is the one considered correct in the ontology's view. PowerAqua fails to give an answer if the knowledge is in the ontology(ies) but it can not find it. Note that a conceptual failure (the knowledge is not in the ontology) is not considered as a PowerAqua failure. It can also happen that a correct mapping, corresponding to a complete ontology-compliant representation, may give no results at all if the ontology is not populated.

It is important to emphasize that recall can not be measured precisely in this open scenario, as we don't know in advance how many ontologies contain the answer to a user's query. Therefore we measure **success** in terms of getting (or not) at least one answer. However, we can compare the difference in recall between the use of PowerAqua in the "Cautious Knowledge Sharing" scenario and that with the centralized Watson gateway.

Speed is measured in terms of seconds to answer a query and most importantly in terms of number of calls to the Watson API. Querying Watson is the major bottleneck here, therefore we measure both the total time that the system spends in answering a query and the time that the system spends querying the WatsonPeers. However, the key goal is to measure the difference between the use of distributed WatsonPeers versus the centralized Watson. To make a fair comparison we need to look at the total number of calls to the Watson API, which directly depends on the number of peers (data) that take part in the answering process, rather than the overall time performance. The reason is that for the OpenKnowledge infrastructure we have used virtual servers, while the centralized Watson is placed in a very fast physical server. Hence a purely time-centric comparison would be misleading. This aspect will be further elaborated during the discussion in Section 4.2.2.

4.1.2 Data sets

The questions used during the evaluation were selected as follows. We asked four members of KMi, familiar with the Semantic Web and ontology engineering, to generate questions for the system that were covered by at least one ontology in our collection. We pointed out to our colleagues that the system is limited in handling temporal information; therefore we asked them not to design questions which required temporal reasoning (e.g., today, last month, between 2004 and 2005, last year, etc). Moreover, we also pointed out that PowerAqua is not a conversational system; each query must be resolved on its own with no references to previous queries. Apart from these points, no other constraints or guidance was given to the people generating the questions and indeed, because no 'quality control' was carried out on the questions, it was admissible for these to contain spelling mistakes and/or grammatical errors. In total, we collected 25 questions.

4.2 Results

Using the OpenKnowledge architecture adds another dimension to our evaluation, in the sense that we have to measure if the Enhanced Discovery Service (EDS) is successful and capable of finding the relevant peers, and its impact on recall. If the EDS fails, the query is then broadcast to all the peers. As said before, speed is measured taking into account not just the total time to answer a query or the total time expended by Watson, but more accurately by the number of calls to the API, which directly depends on the number of selected peers (precision).

In what follows we use the term *S-Watson* to refer to the standalone, centralized Watson, and the term *OK-Watson* to refer to the network of mini-Watson servers integrated with the OpenKnowledge peer to peer architecture. Similarly we talk about *S-Performance* and *OK-Performance* to refer to the performance of PowerAqua with the standalone or decentralized Watson back-ends.

4.2.1 List of queries and results

Q1: Give me all publications authored by Enrico Motta.

S-Watson – Success

OK- Watson – Success

S-Performance: 2680 calls to Watson API (49 secs)

OK-Performance: 2165 calls to Watson API (14 mins total time, 13.85 mins OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies? 1 / 1

Watson Ontologies which answer the query:

/data/home/davide/vane/KSW-2Abril07/populatedAKTOntology.owl

Peers selected by the Enhanced Discovery Service: 8 peers matched

- sweto_26_june/@kmi-dev07.open.ac.uk-5007 - *publication*,
- akt/@kmi-dev08.open.ac.uk-5000 - *publication*,
- kmiSemanticPortal/@kmi-dev08.open.ac.uk-5005 - *publication*,
- sweto_dblp_august_2007/@kmi-dev08.open.ac.uk-5007 - *publication*,
- ka2/@kmi-dev08.open.ac.uk-5004 - *publication*,
- upm-organizational-ontology/@kmi-dev08.open.ac.uk-5008 - *publication*,
- wsd/@kmi-dev09.open.ac.uk-5005 - *publication*,
- www_ontologies/@kmi-dev09.open.ac.uk-5006 - *authors*

Q2: What is the title of the paper authored by Enrico Motta at ISWC'2007

Linguistic Component failure: out of coverage.

Q3: Give me all albums of Metallica

S-Watson – Success

OK – Watson - Success

S-Performance: 1611 calls to Watson API (19 secs)

OK-Performance: 1446 calls to Watson API (75.029 secs total time, 72.436 secs OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies? 2 / 2

Watson Ontologies which answer the query:

/data/home/davide/vane/ontologias TREC/ontologies-updated/music.owl

/data/home/davide/vane/ontologias TREC/494--Nirvana/music.owl

Peers selected by the Enhanced Discovery Service: 1 peer matched

- trec/@kmi-dev09.open.ac.uk-5003 - *album*

Q4: Which are the albums of the rock group Metallica?

S-Watson – Success

OK – Watson - Success

S-Performance: 25338 calls to Watson API (5 mins 29)

OK-Performance: 4706 calls to API (27.89 mins total time, 27.33 mins OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies? 2 / 2

Watson Ontologies which answer the query:

/data/home/davide/vane/ontologias TREC/494--Nirvana/music.owl

/data/home/davide/vane/ontologias TREC/ontologies-updated/music.owl

Peers selected by the Enhanced Discovery Service: 7 peers matched

- trec@kmi-dev09.open.ac.uk-5003 - *album*,
- ato/@kmi-dev07.open.ac.uk-5000 - *rock*,
- wsd@kmi-dev09.open.ac.uk-5005 - *rock*,
- www_ontologies@kmi-dev09.open.ac.uk-5006 - *rock*,
- fao_agrovoc/@kmi-dev07.open.ac.uk-5001 - *group*,
- kimOntology/@kmi-dev07.open.ac.uk-5004 - *group*,
- eswc06/@kmi-dev08.open.ac.uk-5001 - *group*

Q5: Give me all californian dry wines.

PowerMap failure (ontology discovery): it fails to map the term “Californian” to the concept “CalifornianWine” in the relevant ontology.

Q6**: **Which religion does Easter belong to?

S-Watson – Success

OK- Watson – Success

S-Performance: 901 calls to Watson API (18 secs)

OK-Performance: 444 calls to API (75.851 secs total time, 72.781 secs OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies: 2 / 2

Watson Ontologies which answer the query:

/data/home/davide/vane/ontologias TREC/458--fasting/fasting_458.owl

/data/home/davide/vane/ontologias TREC/ontologies-updated/fasting.owl

Peers selected by the Enhanced Discovery Service: 2 peers matched

- www_ontologies@kmi-dev09.open.ac.uk-5006 - *religion*,
- trec@kmi-dev09.open.ac.uk-5003 - *religions*

Q7: Which are the fasting periods in Islam?

S-Watson – Success

OK- Watson – Success

S-Performance: 1116 calls to Watson API (19 secs)

OK-Performance: 462 calls to API (112.269 secs total time, 99.885 secs OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies? 2 / 2

Watson Ontologies which answer the query:

/data/home/davide/vane/ontologias TREC/458--fasting/fasting_458.owl

/data/home/davide/vane/ontologias TREC/ontologies-updated/fasting.owl

Peers selected by the Enhanced Discovery Service: 3 peers matched

- wsd@kmi-dev09.open.ac.uk-5005 - *period*,
- trec@kmi-dev09.open.ac.uk-5003 - *period*,
- www_ontologies@kmi-dev09.open.ac.uk-5006 - *period*

Q8: Is there a railway connection between Amsterdam and Berlin?

Linguistic Component failure: out of coverage.

Q9: What are the main rivers of Russia

S-Watson – Success

OK- Watson – Success

S-Performance: 8473 calls to Watson API (2 mins 4 secs)

OK-Performance: 5580 calls to API (10.15 mins total time, 10.02 mins OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies? 2 / 2

Watson Ontologies which answer the query:

/data/home/davide/vane/LinkedData/ontoworld/ontoworld.xml

/data/home/davide/vane/ontologies WSD- mappings/background ontologies/russiaB.rdf

Peers selected by the Enhanced Discovery Service: 2 peers matched

- linkedData/@kmi-dev07.open.ac.uk-5005 - *river*,
- wsd@kmi-dev09.open.ac.uk-5005 - *russia*

Q10: Which russian rivers flow into the Black Sea

S-Watson – Success

OK- Watson – Success

S-Performance: 1447 calls to Watson API (3 mins 18 secs)

OK-Performance: 1038 calls to API (4.22 mins total time, 3.85 mins OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies? 3 / 3

Watson Ontologies which answer the query:

/data/home/davide/vane/LinkedData/ontoworld/ontoworld.xml

/data/home/davide/vane/ontologies WSD- mappings/background ontologies/russiaB.rdf

http://www.cyc.com/2002/04/08/FreeToGovCyc

Peers selected by the Enhanced Discovery Service: 4 peers matched

- linkedData/@kmi-dev07.open.ac.uk-5005 - *river*,
- wsd@kmi-dev09.open.ac.uk-5005 - *flow*,
- www_ontologies@kmi-dev09.open.ac.uk-5006 - *black*,
- trec@kmi-dev09.open.ac.uk-5003 - *sea*

Q11: Which prizes have been won by Laura Linney?

S-Watson – Success

OK- Watson – Success.

S-Performance: 914 calls to Watson API (30 secs)

OK-Performance: 309 calls to API (54.958 secs total time, 49.591 secs OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies? 1 / 1

Watson Ontologies which answer the query:

http://nets.ii.uam.es/film_festival.owl

Peers selected by the Enhanced Discovery Service: 3 peers matched

- wsd@kmi-dev09.open.ac.uk-5005 - *award*,
- trec@kmi-dev09.open.ac.uk-5003 - *award*,
- www_ontologies@kmi-dev09.open.ac.uk-5006 - *award*

Q12: What are the symptoms of Parkinson?

S-Watson – Success

OK- Watson – Success

S-Performance: 1188 calls to the API (21 secs)

OK-Performance: 486 calls to API (22.296 secs total time, 21.466 secs OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies?. 3 / 3

Watson Ontologies which answer the query:

/data/home/davide/vane/ontologias TREC/ontologies-updated/diseases.owl
/data/home/davide/vane/ontologias TREC/ontologies-updated/parkinson.owl
/data/home/davide/vane/wwwontologies/finalontologiesTREC/tap/tap-project.owl

Peers selected by the Enhanced Discovery Service: 1 peer matched

- trec@kmi-dev09.open.ac.uk-5003 - *symptom*

Q13: Who stars in Bruce Almighty?

S-Watson – Success

OK- Watson – Success

S-Performance: 6233 calls to the API (4 mins 6 secs)

OK-Performance: 5498 calls to API (70.51 mins total time, 70.09 mins OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies? 2 / 2

Watson Ontologies which answer the query:

/data/home/davide/vane/ontologias TREC/ontologies-updated/Jennifer.owl
/data/home/davide/vane/wwwontologies/finalontologiesTREC/moviedatabase/moviedatabase.rdf

Peers selected by the Enhanced Discovery Service: 16 peers matched

- linkedData/@kmi-dev07.open.ac.uk-5005 - *person*,
- wsd@kmi-dev09.open.ac.uk-5005 - *person*,
- kimOntology/@kmi-dev07.open.ac.uk-5004 - *person*,
- sweto_26_june/@kmi-dev07.open.ac.uk-5007 - *person*,
- akt/@kmi-dev08.open.ac.uk-5000 - *person*,
- eswc06/@kmi-dev08.open.ac.uk-5001 - *person*,
- semrank/@kmi-dev08.open.ac.uk-5006 - *person*,
- kmiSemanticPortal/@kmi-dev08.open.ac.uk-5005 - *person*,
- trec@kmi-dev09.open.ac.uk-5003 - *person*,
- ka2/@kmi-dev08.open.ac.uk-5004 - *person*,
- www_ontologies@kmi-dev09.open.ac.uk-5006 - *person*,
- upm-organizational-ontology/@kmi-dev08.open.ac.uk-5008 - *person*,
- tap/@kmi-dev07.open.ac.uk-5008 - *person*,
- ato/@kmi-dev07.open.ac.uk-5000 - *organization*,
- sweto_dblp_august_2007/@kmi-dev08.open.ac.uk-5007 - *organization*,
- dolce@kmi-dev09.open.ac.uk-5000 - *organization*

Q14: Who presented a poster at ESWC 2006?

Triple Similarity Services failure: the TSS fails to understand the term "ESWC 2006", even though the selected ontology provides information about the 2006 European Semantic Web Conference.

Q15: What is the capital of Turkey?

PowerMap failure (filtering heuristics): PowerMap identifies both an exact mapping, "Capital", and an approximate one, "CountryCapital". It then proceeds to rule out the approximate one, which is actually the one that would have led to the correct answer –the exact mapping "Capital" generates regional capital cities instead.

Q16: Who are the people working in the same place of Paolo Buquet?

Linguistic Component failure: out of coverage.

Q17: Give me all the articles written by people from KMi

PowerMap failure (ontology discovery): the term “people” is not mapped to “person” in the relevant ontology. Moreover the term “KMi’ does not appear as an alternative name for “knowledge media institute” anywhere, and therefore it can not be found.

Q18: Give me the list of restaurants which provide italian food in San Francisco

Triple Similarity Services failure: The integration of PowerAqua with Watson is limited in its handling of literals, hence this query fails in both the centralized and distributed Watson scenarios. However, PowerAqua is able to answer this query correctly when using its own internal Sesame repository.

Q19: Which restaurants are located in San Pablo ?

S-Watson – Success

OK- Watson – Success

S-Performance: 11223 calls to the API (2 mins 47 secs)

OK-Performance: 9861 calls to API (66.77 mins total time, 66.59 mins OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies? 1 / 1

Watson Ontologies which answer the query:

http://www.mooney.net/restaurant

Peers selected by the Enhanced Discovery Service: 9 peers matched

- ato/@kmi-dev07.open.ac.uk-5000 - *restaurant*,
- linkedData/@kmi-dev07.open.ac.uk-5005 - *restaurant*,
- wsd@kmi-dev09.open.ac.uk-5005 - *place*,
- sweto_26_june/@kmi-dev07.open.ac.uk-5007 - *place*,
- eswc06/@kmi-dev08.open.ac.uk-5001 - *place*,
- semrank/@kmi-dev08.open.ac.uk-5006 - *place*,
- trec@kmi-dev09.open.ac.uk-5003 - *place*,
- www_ontologies@kmi-dev09.open.ac.uk-5006 - *place*,
- tap/@kmi-dev07.open.ac.uk-5008 - *place*

Q20: Which cities are located in the region of Sacramento Area?

PowerMap failure –the correct mapping for Sacramento Area is not found.

Q21: What is the apex lab?

S-Watson – Success

OK- Watson –OK Enhanced Discovery Service Failure. However, success is achieved through broadcasting.

S-Performance: 18 calls to the API (6 secs)

OK-Performance: 10 calls to Watson API (49.89 secs total time, 48.21 secs OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies?

Not relevant – EDS actually fails in this case.

Watson Ontologies which answer the query:

/data/home/davide/vane/iswc-aswc07/iswc-aswc-2007-complete.rdf

Peers selected by the Enhanced Discovery Service: No peer matching - BROADCASTING to 25 peers

Q22: Who believe in the apocalypse?

S-Watson – Success

OK- Watson – Success (*OK-key concepts*: it does not find “apocalypse” or “religious_organization”, it selects the peer by matching “organization”).

S-Performance: 5045 calls to the API (2 mins 5 secs)

OK-Performance: 5504 calls to Watson API (65.20 mins total time, 64.89 mins Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies? 1/1.

Watson Ontologies which answer the query:

/data/home/davide/vane/ontologias TREC/ontologies-updated/apocalypse.owl

Peers selected by the Enhanced Discovery Service: 16 peers matched

- linkedData/@kmi-dev07.open.ac.uk-5005 - *person*,
- wsd@kmi-dev09.open.ac.uk-5005 - *person*,
- kimOntology/@kmi-dev07.open.ac.uk-5004 - *person*,
- sweto_26_june/@kmi-dev07.open.ac.uk-5007 - *person*,
- akt/@kmi-dev08.open.ac.uk-5000 - *person*,
- eswc06/@kmi-dev08.open.ac.uk-5001 - *person*,
- semrank/@kmi-dev08.open.ac.uk-5006 - *person*,
- kmiSemanticPortal/@kmi-dev08.open.ac.uk-5005 - *person*,
- trec@kmi-dev09.open.ac.uk-5003 - *person*,
- ka2/@kmi-dev08.open.ac.uk-5004 - *person*,
- www_ontologies@kmi-dev09.open.ac.uk-5006 - *person*,
- upm-organizational-ontology/@kmi-dev08.open.ac.uk-5008 - *person*,
- tap/@kmi-dev07.open.ac.uk-5008 - *person*,
- ato/@kmi-dev07.open.ac.uk-5000 - *organization*,
- sweto_dblp_august_2007/@kmi-dev08.open.ac.uk-5007 - *organization*,
- dolce@kmi-dev09.open.ac.uk-5000 - *organization*

Q23: What is skoda doing?

Linguistic Component failure: query not correctly classified.

Q24: Where are Sauternes produced?

PowerMap failure (filtering heuristics): the system correctly maps the linguistic term "Sauternes" to the wine "Sauterne". However this is not related to a region, leading to a failure at the next stage of the process. PowerMap had indeed identified the mapping which would have led to the answer ("SauterneRegion", which is located in "Bordeaux"), however this mapping was discarded because PowerMap considered it less likely to be correct that the exact mapping to "Sauterne".

Q25: Give me the papers written by Marta Sabou.

S-Watson – Success

OK-Watson – Partial success: one answer found, rather than the eight ones provided by the centralized version.

S-Performance: 1285 calls to Watson API (32 secs)

OK-Performance: 344 calls to Watson API (1.87 mins total time, 1.72 mins OK-Watson time)

Comparative Recall (Enhanced Discovery Service): does the P2P find all the ontologies? Only one out of eight. The reason for this is that WordNet fails to provide key synonyms for 'paper', such as 'publication'. This limitation does not cause these failures in the centralized version, because the individual "marta sabou" is also used as an index (while in the distributed version, no indexing on individuals is provided).

Watson Ontologies which answer the query:

/data/home/davide/vane/iswc-aswc07/iswc-2006-complete.rdf

/data/home/davide/vane/KSW-2Abril07/populatedAKTOntology.owl

/data/home/davide/vane/sweto 26-june-06/Ontology/sweto_big.rdf

/data/home/davide/vane/iswc-aswc07/om-2007-complete.rdf

/data/home/davide/vane/iswc-aswc07/pc
/data/home/davide/vane/iswc-aswc07/iswc-aswc-2007-complete.rdf
/data/home/davide/vane/UPM-LIA organizational ontologies/RDFsInstancesPerson.rdfs
/data/home/davide/vane/iswc06/iswc2006researchPapers.rdf

Peers selected by the Enhanced Discovery Service: 5 peers matched

- iswc06/@kmi-dev08.open.ac.uk-5003 - *paper*,
- eswc06/@kmi-dev08.open.ac.uk-5001 - *paper*,
- www_ontologies@kmi-dev09.open.ac.uk-5006 - *paper*,
- kimOntology/@kmi-dev07.open.ac.uk-5004 - *document*,
- sweto_dblp_august_2007/@kmi-dev08.open.ac.uk-5007 - *document*

4.2.2 Analysis of results

From the 25 questions collected, 14 were answered successfully, thus PowerAqua is able to correctly answer more than half of the queries (56% success). This is actually a rather good result, given i) the open nature of the question answering set up – hardly any constraints were imposed on the choice of the questions, ii) the size and heterogeneity of the dataset, and iii) the use of an innovative approach to advertise ontologies, based on the use of key concepts.

We have analyzed the 11 failures and divided them into four categories depending on the component that caused the failure:

- *Linguistic analysis*. A failure can be due to the query being out of the scope of the linguistic component's coverage, an incorrect annotation on the underlying GATE linguistic platform and grammars (e.g., annotating a verb as a noun), or the linguistic component misunderstanding the query type. Four queries, i.e., 16% of the total number of queries posed to the system, failed because of limitations with the linguistic component. Extending its coverage should solve some (if not all) of these errors.
- *PowerMap*. The PowerMap algorithm tries to maximize recall by broadening the search space and avoiding correct mappings to be dismissed as invalid. Accuracy is not so crucial at this stage, given that, even if an incorrect mapping is suggested by PowerMap, it will normally be discarded at a later stage by using the semantics provided by the ontologies. However, too many irrelevant mappings collected in this phase inevitably affect the overall performance of the system. In our evaluation, five queries, 20% of the test set, failed due to limitations of the PowerMap algorithm. In three queries, this happened because the relevant mappings could not be found by the algorithm. In the remaining two queries, Q15 and Q24, PowerMap had indeed found the correct mappings, however these were discarded because they were considered less likely to lead to the correct solution than other ones.
- *Triple Similarity Service (TSS)*. A TSS-related failure occurs when PowerMap correctly selects an ontology which contains the answer to the current query, but the TSS fails to complete the matching process by locating the correct triple(s) answering the query. In our evaluation, two of the queries, 8% of the total, failed because of this component.
- *Enhanced Discovery Service*. All the 14 queries answered by the standalone Watson version were also answered by the OK distributed version. Having said so, in one case, query Q21, the Enhanced Discovery Service failed to locate the relevant peers for the query and the system had then to resort to

broadcasting. This produced the right answer, but of course is a very inefficient solution, which also goes against the whole idea of modelling "Cautious Knowledge Sharing" scenarios. We also had a problem with query Q25, where only one of the eight ontologies that can answer this query was returned by the decentralized version. Despite these issues with two of the queries, the almost complete success of the EDS in locating the peers and ontologies relevant to a query provides a very positive and to some extent surprising result, given that the set of key concepts extracted from an ontology is usually a very small part of the ontology. We will discuss this aspect in more detail in the next section.

As far as speed is concerned, answering times for the successful queries in the PowerAqua standalone version, ranged from 6 seconds to 5 mins 29 seconds for the query "Which are the albums of the rock group Metallica?". The average time was 98 seconds. The reasons for the slow performance on the query "Which are the albums of the rock group Metallica?" are mainly two. Firstly, the ontology mappings for the compound "rock group metallica" and "rock group" as a whole do not produce valid ontology triples, and therefore a second set of iterations of PowerAqua algorithms with the individual terms "rock", "group" and "Metallica" is needed. Secondly, "group" is a very generic keyword that returns many approximate mappings on several ontologies, most of which are later discarded. In total, processing this query requires 25338 calls to the Watson API. Note that these times can be improved by further optimizing the interaction between PowerAqua and Watson. In particular, in the PowerAqua evaluation using the TREC corpus presented in [10], in which PowerAqua relies on its internal Sesame repositories with the same collection used in this evaluation, the average time for answering a query was 60 secs, a 30% improvement over the figure achieved with the standalone Watson. To this purpose we are currently working on achieving a tighter integration between PowerAqua and Watson to improve the performance of the interaction between the two systems.

In the OK version the overall performance is a bit slow, the best time is 22 secs for the query "What are the symptoms of Parkinson?". Here 21 seconds are used by Watson, while the other second is taken by the PowerAqua algorithms. Moreover, the worst performing queries can take up to a hour to be answered. However these results have to be interpreted cautiously, as they depend on the actual computer specifications and network speed. In particular, as already pointed out, we are using virtual servers for the OK evaluation instead of the physical server we use for the centralized version. Indeed we are confident that if the evaluation had been carried out on physical servers[3] the time performance on the OK architecture would have actually been better than the one on the centralized Watson. The reason for this is that the number of calls to the Watson API (the main bottleneck in the process) is always considerably less in the "Cautious Knowledge Sharing" scenario. For instance, for the aforementioned query, "Which are the albums of the rock group metallica?", only 4706 calls are sent to the API in the OK infrastructure (where 7 peers are selected as relevant), instead of 25338 calls, as it is the case with the centralized version. This dramatic

[3] Note that there are some issues with firewalls on the OK P2P network, therefore the computers hosting the peers need to be in the same physical network.

reduction in the number of calls sent to Watson in the decentralized scenario is a result of the key-concept-based indexing, which drastically limits the number of potentially relevant ontologies returned by PowerMap.

4.2.3 Discussion

Here we elaborate further on some of the issues highlighted during this evaluation in the OK scenario:

- Queries formed with “who” (“who” maps to the very generic linguistic terms “person” and “organization”) return more than half of the peers (16 out of 25 peers). Therefore, these queries are slow, almost like broadcasting, as more peers equals more calls to the Watson API. However, for the rest of the queries that do not involve only instances, the Enhanced Discovery Service works very well returning mostly just the relevant peers. Also, we should consider that we only have 25 peers for around 130 ontological repositories (and more than 600 ontological documents). If we had set up more peers, the Enhanced Discovery Service would have been even more precise, and this would have reduced further the number of calls to the API.
- In this evaluation only one query, "what is the apex lab?", required broadcasting in the OK scenario. The reason for this is that this query is translated to a linguistic query such as <thing, ?, apex lab>, which does not match any key concept. This is actually a very interesting case, which highlights two different ways in which queries in the “Cautious Knowledge Sharing” scenario may fail. To clarify the first case, let’s assume that the query had been formulated as: "what is apex?". In such a case, the only handle we have to make sense of the query is the term “apex”. However only classes are advertised as key concepts, hence there is no way we can find the relevant peer solely on the basis of the name of an individual. Hence, this is not a problem with this particular query, but a general limitation of our achitecture, which expects ‘conceptual handles’ to be provided, in order to dispatch a query to the right peers. This issue is particularly problematic for PowerMagpie, which mostly looks for individuals in Watson, and knows very little about the class they belong to. As a result broadcasting is invariably needed, which negatively affects performance. For instance, PowerAqua can perform 486 calls in 22 secs when 3 peers are selected as relevant (0.04 secs per iteration in "what are the symptoms of Parkinson?"), while in the case of broadcasting, as in "what is the apex lab?", 10 calls to the API take 49 secs (4 secs per iteration). One solution to this problem could be to try and guess the type of individuals, using either the public version of Watson, or other sources, such as the Web, before posting the query to the EDS. However, more work is needed to formulate this approach precisely and to evaluate it.
- It is interesting however to note that the EDS mechanism failed in the previous query despite the fact that the concept “lab” had actually been provided in the query. Here the problem was that the keyword “lab” (or any of its synonyms provided by WordNet) did not match any key concept in any ontology. This is quite surprising and indeed it provides the only example of an actual failure by the key concepts mechanism. The reason for this is that the class “Laboratory” only appears in very big ontologies, such as SUMO, TAP, and Cyc, which have only been incompletely characterized by the

twenty, automatically extracted, key concepts. It is very likely that, if a higher number of key concepts had been extracted for these very large ontologies, such a failure would have not occurred, even though in practice the query itself would still have failed, given that none of the ontologies which cover the class "Laboratory" actually provide an answer to the query[4]. To address this problem, i.e., the incomplete characterization of very large ontologies with only twenty concepts, we would have to modify our algorithm to deduce automatically the right number of concepts needed to characterize fully an ontology, rather than providing this number explicitly as an input to the algorithm.

- Although the Enhanced Discovery Service is able to find answers to all queries except the one about the apex lab, the query "give me the papers written by Marta Sabou", when executed in the OK scenario, only finds one ontology instead of eight. As already mentioned, the reason for this is that WordNet fails to provided key synonyms for 'paper', such as 'publication'. This limitation does not cause these failures in the centralized version, because the individual "marta sabou" is also used as an index (while in the distributed version, no indexing on individuals is provided).

5. Conclusions and Future Work.

For the integration of our Semantic Web testbeds within the OpenKnowledge platform, the P2P infrastructure replaced our centralized gateway to the Semantic Web, Watson, providing the same functionality in a potentially scalable distributed scenario.

At this stage, the main advantage of using the P2P infrastructure has to do with the Enhanced Discovery Service, which efficiently finds the peers that may answer a user's query. Crucially, the use of the EDS improves performance (measured as calls to the Watson API), with a very minor adverse effect on recall. This is a very positive and somewhat surprising result, given that the use of key concepts means that ontologies containing hundreds (or even thousand) or ontologies are advertised using only twenty key concepts. More experiments and a more detailed analysis will be required to clarify whether this is the case in general, however it is worth pointing out that our results are consistent with the findings presented in [11], which reports that, out of about 70K queries which had been posted to an ontology-based repository, all but twelve focused on only four classes: Technology, Organization, Research-Area and Person. An interesting feature that links these four classes to our approach to selecting key concepts is that all four of them appear to be pitched at a level of abstraction akin to what Eleanor Rosch termed *natural categories* [12], one of the criteria used by our algorithm to select key concepts. In other words, an interesting working hypothesis is that people use by and large natural categories to construct queries and, as a result of this psycholinguistic phenomenon, it might be feasible to reduce drastically the indexes needed by Watson peers and simply advertise the key natural categories contained in an ontology. If this hypothesis can be proven to hold in general, then it would be worthwhile to use key concepts as

[4] Indeed, if the EDS had provided a match for "Laboratory" in this particular query, then the system would not have resorted to broadcasting, and therefore the query would have failed.

primary indexes in any scenario, regardless of whether we need to enforce "Cautious Knowledge Sharing", to try and improve performance across the board.

A P2P infrastructure also provides a scalable platform for multiple users and trust policies. Currently we are working on incorporating a trust engine and a semantic ranking mechanism with PowerAqua, which will be part of the final version of the system released at M36 of the project.

In conclusion, the evaluation described in this deliverable shows that the integration of Watson with the OpenKnowledge peer to peer architecture can be used very effectively to realize "Cautious Knowledge Sharing" scenarios. In particular, the use of automatic key concept extraction to support 'cautious advertising' has negligible impact on recall (at least in the case of PowerAqua), while spreading ontologies over a network of Watson peers and using the Enhanced Discovery Service to dispatch queries significantly reduces the number of expensive interactions with the Watson back-end.

Acknowledgements

We would like to thank all the members of the OpenKnowledge Kernel Development Team for their excellent support. In particular a special acknowledgement goes to Paolo Besana for the many intensive interactions aimed at finalising the design of the architecture supporting the "Cautious Knowledge Sharing" scenario. We would also like to thank Marta Sabou for her useful comments on an earlier draft of this report.

References

[1] Gridinoc, L., Guidi, D., Motta, E., d'Aquin, M., Anadiotis, G., Besana, P., and Dupplaw, D. (2008). OpenKnowledge Semantic Browser v2. Deliverable D8.2, OpenKnowledge Project (FP6-027253). August 2008.

[2] Guidi, D., Lopez, V., Peroni, S., Motta, E., d'Aquin, M., Sabou, M., Anadiotis, G., Besana, P., and Dupplaw, D. (2008). OpenKnowledge Question Answering System. *Deliverable, D8.4,* OpenKnowledge Project (FP6-027253). August 2008.

[3] Lopez, V., Motta, E. and Uren, V. (2006). PowerAqua: Fishing the Semantic Web, Proceedings of the *European Semantic Web Conference 2006*, Montenegro.

[4] Gridinoc, L., d'Aquin, M., Guidi, D., Dzbor, M., and Motta, E. (2007). PowerMagpie: A Semantic Web Browser. *Deliverable D8.1*. OpenKnowledge Project (FP6-027253). June 2007.

[5] d'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., Motta, E., (2007). Watson: A Gateway for Next Generation Semantic Web Applications. *Poster, International Semantic Web Conference, ISWC 2007.*

[6] Peroni, S., Motta, E., and d'Aquin, M. (2008). Identifying key concepts in an ontology through the integration of cognitive principles with statistical and topological measures. *Proceedings of the 3rd Asian Semantic Web Conference 2008 (ASWC 2008).* Bangkok, Thailand, 8th-11th December 2008

[7] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V. (2002). 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 (ACL'02).*

[8] Lopez, V., Sabou, M. and Motta, E. (2006). PowerMap: Mapping the Real Semantic Web on the Fly. *Proceedings of the International Semantic Web Conference (ISWC 2006)*, Atlanta, Georgia.

[9] Schorlemmer, M., Atencia, M., Bundy, Giunchiglia, F., Lopez, V., and McNeill, F. (2006). Deliverable D3.2, OpenKnowledge Project (FP6-027253). August 2006.

[10] Fernandez, M., Lopez, V., Motta, E., Sabou, M., Uren, V., Vallet, D., Castells, P. (2008). Semantic Search meets the Web. In Proc of the IEEE Conference on Semantic Computing, 2008.

[11] Alani, H., Harris, S., and O'Neil, B. Winnowing Ontologies based on Application Use, Proceedings of 3rd European Semantic Web Conference (ESWC), Budva, Montenegro, 11-14 June 2006.

[12] Rosch, E. Principles of Categorization, Cognition and Categorization, Lawrence Erlbaum, Hillsdale, New Jersey, 1978.