Discovering and Querying Hybrid Linked Data

Zareen Syed[1], Tim Finin[1], Muhammad Rahman[1], James Kukla[2], Jeehye Yun[2]

[1]University of Maryland Baltimore County
1000 Hilltop Circle, MD, USA 21250
zsyed@umbc.edu, mrahman1@umbc.edu, finin@cs.umbc.edu

[2]RedShred
5520 Research Park Drive, Suite 100
Baltimore, MD 21228
jkukla@redshred.net, jyun@redshred.net

Abstract. In this paper, we present a unified framework for discovering and querying hybrid linked data. We describe our approach to developing a natural language query interface for a hybrid knowledge base Wikitology, and present that as a case study for accessing hybrid information sources with structured and unstructured data through natural language queries. We evaluate our system on a publicly available dataset and demonstrate improvements over a baseline system. We describe limitations of our approach and also discuss cases where our system can complement other structured data querying systems by retrieving additional answers not available in structured sources.

Keywords: knowledge discovery, semantic web, text mining, information retrieval, question answering

1 Introduction

There are numerous benefits of extracting structured data from raw text in the form of attribute value pairs aka slots and fillers, it gives the ability to go beyond keyword queries and perform structured queries, such as, get a list of "equipment", "software" or "devices" mentioned in the document or in the corpus as a whole. Furthermore, linking extracted slots and fillers to the knowledge base can greatly increase the recall of such queries by supporting transitivity and other types of inference. For example, a "Digital Camera" is a type of "Camera" which is a type of "Device" in DBpedia Ontology [2]. In addition, a clear taxonomy and aligned attributes enable faceted browsing, which is a powerful and popular way to select articles of interest and also explore corpus statistics. The extracted slots and fillers can serve to provide interesting and informative structured summaries over the raw content of text documents thus helping the reader to quickly decide if the document is of interest. Structured data extracted from text can provide useful semantic features for a variety of tasks such as indexing, clustering, retrieval, and summarization to name a few.

adfa, p. 1, 2011.

One of the biggest challenges faced by the Semantic Web vision is the availability of structured data that can be published as RDF. One approach is to develop techniques to translate information in spreadsheets, databases, XML documents and other traditional data formats into RDF [20]. Another is to refine the technology needed to extract structured information from unstructured free text [8, 9]. Once linked data becomes available, a second challenge arises in being able to easily query large linked data collections such as DBpedia. Using the SPARQL query language requires not only mastering its syntax but also understanding the RDF data model, large ontology vocabularies and URIs for denoting entities. Over the past few years natural language interfaces are becoming popular as they permit users to express queries in natural language without needing to know about the underlying schema or query syntax. Recently, numerous approaches have been developed to address this challenge [7, 14, 25, 26], showing significant advances towards answering natural language questions with respect to large and heterogeneous structured data sources. However, a lot of information is still available only in textual form, both on the web and in the form of labels and abstracts in linked data sources. Therefore approaches are needed that can not only deal with structured data but also with finding information in several sources, processing both structured and unstructured information, and combining such gathered information into one answer.

In this paper, we present exploratory work on a unified framework for discovering as well as querying linked hybrid data collections. The proposed unified framework builds on our previous work on discovering ontology elements from text documents [18] and our new work on developing a natural language interface for a hybrid knowledge base Wikitology [20], which we present as a case study for accessing hybrid information sources through natural language queries.

One motivation for our work is an enhancement to a system we are developing with RedShred, LLC that will help people identify and analyze business documents that include Request for Proposals (RFPs), Request for Quotes (RFQs), calls for proposals, Broad Agency Announcements (BAAs), solicitations and similar business documents. Our prototype uses document analysis, information retrieval, NLP information extraction and question answering techniques and is largely domain independent. It understands general RFP-related concepts (e.g., proposal deadlines, duration, deliverables, security requirements, points of contacts, etc.) and can extract and organize information to help someone quickly evaluate opportunities. However, it does not have built-in knowledge of any particular domain, such as software development or material science, and is thus unable to address potentially critical characteristics involving them. For software, for example, we may need to know if the work requires a particular programming language (e.g., Java), is targeted for a given system or architecture (e.g., iOS), or has special requirements (e.g., 3DES encryption). Given the breadth and variety of domains of interest, manually developing and maintaining custom ontologies, language models and systems for each is not viable. We plan to build on the results of this work to be able to automatically extend a base ontology by identifying and incorporating important domain-specific concepts, relations and axioms.

We see several contributions that this work has to offer:

1. We present a natural language interface over hybrid linked data and present Wikitology Hybrid Knowledge Base as a case study.
2. We discuss examples where the hybrid system retrieves correct results that are not available in the structured source.
3. We describe a unified architecture for discovery and querying of linked data.

In the remainder of the paper, we introduce the Wikitology knowledge base and present a novel natural language query system over the hybrid knowledge base. We perform the evaluation of our system over a publicly available dataset and discuss the results and limitations of our approach and mention related work. Finally, we present a unified framework for discovering and querying linked hybrid data and provide some conclusions and future work directions.

2 Wikitology

Wikitology [20] is a hybrid knowledge base of structured and unstructured information extracted from Wikipedia augmented by RDF data from DBpedia [2], YAGO Ontology [16], WordNet [11] and Freebase [3]. Wikitology is not unique in using Wikipedia to form the backbone of a knowledge base, see [17] and [23] for examples, however, it is unique in incorporating and integrating structured, semi-structured and unstructured information accessible through a single query interface. The query interface supports a variety of queries ranging from simple keyword queries to queries with structural constraints and returns ranked results based on relevance. Wikitology has been tested for a variety of use-cases [20] and has proven to be effective in generating useful features for a variety of tasks.

At the core of Wikitology is an information retrieval (IR) index which is enhanced with fields containing instance data taken from other data structures such as graphs, tables or triples. It also stores references to related instances in native data structures for applications that might need to run data-structure specific algorithms. The specialized IR index enables applications to query the knowledge base using either simple free text queries or complex queries over multiple fields in the index with structural constraints. The current version of Wikitology has 13 fields, the details on the contents of the fields are available in [19].

3 Natural Language Queries over Wikitology

Our natural language question answering system consists of a number of modules, namely, Answer Type Extraction, Property versus Type identification, Named Entity and Concept Linking, and Wikitology Query Formulation. We describe these modules below.

3.1 Answer Type Extraction

For answer type extraction we extract noun chunks from question text using OpenNLP [1]. We generate inflected forms of extracted nouns and map them types to DBpedia classes based on exact match. For example, we match "songs" in question text with "song" DBpedia class. In case we don't find a matching class in DBpedia, we match with WordNet nouns. If we don't find a matching WordNet noun and the noun chunk is composed of multiple words, we remove the first word and try to match with WordNet to match a more generic type, for example, "music albums" is reduced to "albums". We repeat the process until we are left with just one word. If we still don't find a match, we do not detect an answer type and leave the answer type empty when querying Wikitology. In case we detect more than one types, we use the first type. This might not always work for example for queries with conjunctions such as "Which commercial companies and academic universities have collaborated before?" there are more than one types mentioned which are equally important. We currently limit our system to handle simple cases and plan to address complex cases in our future work. For the selected type we further test if it is a property using a heuristic defined in the next section.

3.2 Property vs. Type Identification

In DBpedia, nouns can denote properties or classes. For example, there is a property for "album" and a class for "song" in DBpedia. We use a simple heuristic to differentiate properties from classes i.e. if the noun is followed by the preposition "of" we consider it a property, otherwise a class. We observed that this simple heuristic worked well for several cases in QALD training dataset [22]. We plan to add more heuristics to cover other cases in the future.

3.3 Named Entity and Concept Linking

We use entity linking approach [20] based on Wikitology to link any named entities to concepts in Wikitology. We further enhanced Wikitology's entity linking system with gazetteers of named entities. For linking other concepts we used Wikipedia Miner service [10]. Wikipedia Miner also links named entities, however when we tested with few examples we found Wikitology's named entity linking relatively more accurate and therefore we used Wikitology for named entity linking and Wikipedia Miner for linking other types of concepts. For Wikipedia Miner we used a probability threshold of 0.4. We tested with a lower threshold to improve recall but observed decrease in accuracy. For example, for the question "Which river does the Brooklyn Bridge cross?", the service predicted a link for "cross" to "http://en.wikipedia.org/wiki/Cross" which was not relevant. A threshold of 0.4 worked much better.

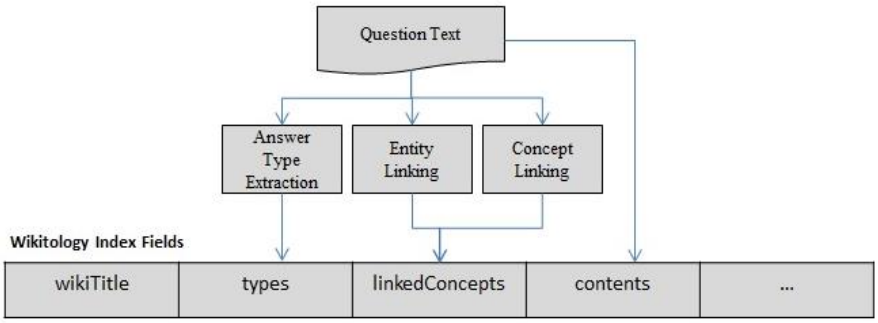

Fig. 1. Question analysis and mapping to Wikitology Index fields

3.4 Property Linking using Semantic Similarity

For questions that ask about a property of a named entity such as "Who is the husband of Amanda Palmer?", we extract the property using the Property Identification heuristic mentioned earlier. For the linked entity we fetch all properties related to the entity from DBpedia. We rank the fetched properties based on semantic similarity with the extracted property using the Semantic Similarity measure in [6]. For example, for the question "Who is the creator of Wikipedia?", we were able to match "creator" with "author". For these types of questions we do not send a query to Wikitology.

3.5 Wikitology Query Formulation

We briefly describe the Wikitology fields that we used for question answering and the query formation below:

a) **wikiTitle**: The 'wikiTitle' field contains the Wikipedia title for a given Wikipedia page.
b) **contents**: The 'contents' field contains the full text of the Wikipedia article including categories, infobox properties, links as well as any redirects to the Wikipedia article.
c) **types**: The 'types' field contains structured data in RDF from the YAGO ontology and the DBPedia Infobox Ontology. The structured data was encoded in an RDFa-like format in the types field for the Wikipedia page. This enables one to query the Wikitology knowledge base using both text (e.g., an entity document) and structured constraints (e.g., rdfs:type =YAGO:President). Freebase resource contained a more comprehensive list of Named Entities (Persons, Locations and Organizations) as compared to YAGO and DBpedia ontology, we therefore generated a list of Wikipedia articles on Persons, Locations and Organizations by extracting all Wikipedia articles defined under the corresponding types in the Freebase resource. We also added the DBpedia

Table 1. Wikitology Query Formulation

```
Input:     question text, answer type, links 1.. N,
topN
Output:    Top N concepts
```

```
Query =
        types:           (answer type) OR
        linkedConcepts: (link1, link2 .. linkN) OR
        contents:        (link1, link2 .. linkN) OR
        contents:        (question text)

List topNConcepts = Wikitology.searchQuery(Query,
topN)

Return topNConcepts
```

WordNet Mappings 5 that are manually created for about 400,000 Wikipedia articles. As Wikipedia has more than 2 million articles we used the Wikipedia Categories to WordNet mappings [13] to heuristically assign a WordNet type to any remaining Wikipedia articles [19].

d) **linkedConcepts**: This field lists the out-links of Wikipedia pages. This field can be used to retrieve linked concepts and also to impose structural constraints while querying (e.g., linkedConcepts = Michelle_Obama, linkedConcepts = Chicago).

Based on the analysis of the question text, we map different query components to different fields in Wikitology index. We create a specialized query to Wikitology by mapping answer type to types field, extracted links to linkedConcepts field as well as contents field, and question text to contents field as seen Table 1.

3.6 Evaluation

For evaluating our system, we used the English questions from the QALD-4 dataset [22]. We restricted to only those questions which had an answer type of "resource" i.e. a URI is provided, and had "aggregation" as false which deals with counting, filtering or ordering, as our system does not currently support these types of queries. We also removed any questions with comparatives and superlatives and which returned Boolean answers i.e. True or False. The total number of questions we considered was 112. We created a baseline system for comparison. The baseline system queried the question text against only the "contents" field in Wikitology. Some questions in QALD dataset have a list of answers. We consider our answer to be correct if any of the top N retrieved

Table 2. Evaluation Results on QALD-4 dataset

Total Questions	Simple Search (top 1)	Simple Search (top 10)	Wikitology Query (top 1)	Wikitology Query (top 10)
112	5	30	29	46

concepts are present in the given answers list. We tested both the baseline and the hybrid system using N = 1 and N = 10. The results are shown in Table 2. The simple search system retrieved one of the correct answers as a top answer in only 5 cases whereas, the Wikitology Query was able to retrieve one of the correct answers as a top answer for 29 queries. Considering top 10 retrieved results, the simple search system retrieved one of the correct answers for 30 questions versus 46 questions by the Wikitology Query.

3.7 Discussion

We experimented with a Wikitology version that was built from Wikipedia dump of March 2010. The QALD-4 dataset uses a more recent version of DBpedia. Using Wikitology constructed from a more recent dump may help in improving recall. We manually looked into answers returned by our system for few queries and found a few cases where the returned concept was correct but was not present in the results of the translated DBpedia SPARQL query in QALD-4 dataset. For example, for the question "Which professional surfers were born in Australia?", our system retrieved the top concept "Layne_Beachley" which is a correct answer, however it is not available as an answer in QALD dataset and hence was not marked as correct in evaluation. Another example is for the question "Which ships were called after Benjamin Franklin?", the system retrieved "French_ship_Franklin_(1797)", which is a correct answer but was not present in QALD answers since those answers are based on DBpedia dataset only. These examples show that a hybrid question answering system that uses linked data as well as text can help in improving recall and complement other natural language query systems that retrieve answers from structured sources only. We also observed that some error was introduced due to linking with a wrong entity, for example, for the query "List all games by GMT.", "GMT" was linked to "Greenwich_Mean_Time" instead of "GMT_Games". In addition to that we came across a number of cases in QALD dataset which required multi-hop path queries. Since our system does not currently support path queries it did not perform well on these types of questions. Another source of error was questions with conjunctions, for example, "Give me all people that were born in Vienna and died in Berlin". Our system does not handle conjunctions yet and hence missed this query. We have employed a basic analysis of the input question, we can improve the approach by exploiting a dependency parse and extracting grammatical relations.

3.8 Related Work

Question Answering systems can be categorized into three different types. 1) Text-based QA systems [15] which first retrieve a relevant set of documents and then extract the answers from these documents. 2) Collaboration-based QA systems [24] exploit answers from the similar questions which have been answered by users on collaborative QA platforms, such as Quora and Yahoo! Answer. 3) Structured data-based QA systems find answers by searching the database instead of the corpus, where the natural language questions are usually translated into some structural queries, such as SQL or SPARQL [4, 5, 14, 21]. Recently the QALD-4 [22] task introduced a hybrid question answering track, in which given a natural language question or keywords, the system is required to retrieve the correct answer(s) from a given repository containing both RDF data and free text. This track was introduced last year, however there was just one submission which was later withdrawn. We find our system in line with the new hybrid question answering track.

4 Unified Framework for Discovering and Querying

We have already discussed our system for natural language querying over hybrid linked data. In this section we describe our earlier work on discovering slots and fillers and how both systems can be integrated to provide a unified framework for discovering and querying semantic data from a given corpus. The unified framework will take as input a corpus of text documents and discover slots and fillers by linking keywords to concepts in the knowledge base using a slot filler discovery approach described below. The discovered slots and fillers will be added to the knowledge base along with the article text. The natural language query interface discussed earlier will provide support for querying over discovered slots and fillers along with associated document text using a hybrid Wikitology query.

4.1 Discovering Slots and Fillers

The approach for discovering slots and fillers is based on the observation that linked concepts can serve as candidate fillers and the "types" associated with linked concepts can serve as candidate slot labels. For example, the Wikipedia article on "Microsoft" links to "Windows", "Office", "Skype" etc. All three of these are a type of "Software" in DBpedia Ontology. By exploiting the types associated with fillers (linked concepts) we can discover a slot for "Software" and provide answers to a structured query such as retrieve list of softwares by Microsoft. The same slot can also serve as a useful facet and enable users to select all articles that are related to "software". The slots and fillers can serve as informative structured summaries like info-boxes in Wikipedia. This approach can be extended to non-Wikipedia articles by first linking keywords and entities to concepts in Wikitology and using the type information in Wikitology to predict a slot label. Not all candidate slots and fillers discovered using the links might be meaningful and will need further selection. Based on the observation that slots are related to entity type and entities of the same type share slots, the documents can be clustered and the

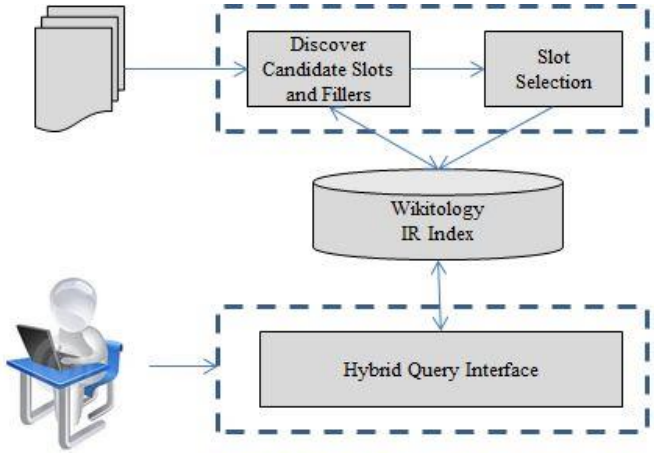

Fig. 2. Unified Framework for Discovery and Query of Linked Hybrid Data

top 'n' most frequent slots can be selected for each cluster whereas, rare slots can be discarded. For more information please see our detailed paper on this approach and its performance [18].

5 Conclusion and Future Work

In this paper, we presented exploratory work on a unified framework for discovering as well as querying linked hybrid data collections. We described our approach to developing a natural language interface for a hybrid knowledge base Wikitology, which we presented as a case study for accessing hybrid information sources through natural language queries. We evaluated our system on a publicly available dataset and demonstrated improvements over a baseline system. We described limitations of our system and also presented examples where our system was able to retrieve additional answers that were not available in structured sources and may complement existing natural language querying systems that retrieve answers from structured sources only. Our current system performs a basic analysis of the input question and therefore can handle limited types of queries, we plan to improve the approach by exploiting a dependency parse and extracting grammatical relations. In addition to that we plan to support path queries by translating natural language queries to SPARQL queries.

6 References

1. Baldridge, J.: The OpenNLP Project. URL: http://Opennlp. Apache.Org/Index.html,(Accessed March 2015)
2. Bizer, C.: The Emerging Web of Linked Data. IEEE Intelligent Systems 24(5) (2009) 87–92

3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T. and Taylor, J.: Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, ACM (2008) 1247-1250
4. Dima, C. Answering Natural Language Questions With Intui3. Conference and Labs of the Evaluation Forum (CLEF) (2014)
5. Han, L., Finin, T. and Joshi, A.: Schema-Free Structured Querying of Dbpedia Data. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, ACM (2012) 2090-2093
6. Han, L., Finin, T., McNamee, P., Joshi, A. and Yesha, Y.: Improving Word Similarity by Augmenting PMI with Estimates of Word Polysemy. IEEE Transactions on Knowledge and Data Engineering, IEEE Computer Society, (2013) 1307-1322
7. He, S., Zhang, Y., Liu, K. and Zhao, J.: CASIA@ V2: A MLN-Based Question Answering System Over Linked Data. In: Proc. of QALD-4 (2014)
8. Ji, H., Dang, H. T., Nothman, J. and Hachey, B.: Overview of TAC-KBP 2014 Entity Discovery and Linking Tasks. In Proc. of TAC-2014 (2014)
9. McNamee, P. and Dang, H.: Overview of the TAC 2009 Knowledge Base Population Track. In: Proc. of TAC-2009 (2009)
10. Milne, D. and Witten, I. H.: Learning to Link with Wikipedia. In: Proc. of the 17th ACM Conference on Information and Knowledge Management (2008) 509-518
11. Miller, G. A.: Wordnet: A Lexical Database For English. Communications of the ACM 38, No. 11 (1995) 39-41
12. Popescu, A. M., Etzioni, O. and Kautz, H.: Towards a Theory of Natural Language Interfaces to Databases. In: Proc. of the 8th International Conference on Intelligent User Interfaces, ACM (2003) 149–157
13. Ponzetto, S. and Strube, M.: WikiTaxonomy: A Large Scale Knowledge Resource. In: Proc. of ECAI 2008, Amsterdam, The Netherlands, IOS Press. (2008) 751–752
14. Park, S., Shim, H. and Lee, G. G.: ISOFT at QALD-4: Semantic Similarity-Based Question Answering System Over Linked Data. In Proc. of CLEF-2014 (2014)
15. Ravichandran D. and Hovy, E.: Learning Surface Text Patterns for a Question Answering System. In: Proc. of The 40th Annual Meeting On Association For Computational Linguistics, ACL'02 (2002) 41–47
16. Suchanek, F. M., Kasneci, G. and Weikum, G.: Yago: A Core of Semantic Knowledge. In: Proc. of the 16th International Conference on World Wide Web, ACM, (2007) 697-706
17. Suchanek, F. M., Kasneci, G. and Weikum, G. : Yago: A Large Ontology from Wikipedia and Wordnet. Web Semant. 6(3) (2008) 203–217
18. Syed, Z. and Finin, T.: Unsupervised Techniques for Discovering Ontology Elements from Wikipedia Article Links. In: Proc. of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading. Association for Computational Linguistics (2010)
19. Syed, Z.: Wikitology: A Novel Hybrid Knowledge Base Derived From Wikipedia. Dissertation, University Of Maryland (2010)
20. Syed, Z. and Finin, T.: Creating and Exploiting a Hybrid Knowledge Base for Linked Data. In Agents and Artificial Intelligence, Revised Selected Papers Series: Communications in Computer and Information Science, V129, Springer, April (2011)
21. Unger, C., Bühmann, L., Lehmann, J., Ngomo, N. A., Gerber, D., and Cimiano, P.: Template-Based Question Answering Over RDF Data. In WWW, (2012) 639–648
22. Unger, C., Forascu, C., Lopez, V., Ngomo, N. A., Cabrio, E., Cimiano, P. and Walter, S.: Question Answering Over Linked Data (QALD-4). In Working Notes for CLEF 2014 Conference (2014)

23. Wu, F. and Weld, D. S.: Automatically Refining the Wikipedia Infobox Ontology. In Proc. of the 17th International Conference on World Wide Web, ACM, (2008) 635-644
24. Wu, Y., Hori, C., Kawai, H., and Kashioka, H.: Answering Complex Questions via Exploiting Social Q&A Collection. In: IJCNLP (2011) 956–964
25. Xu, K., Feng, Y., and Zhao, D.: Xser@ QALD-4: Answering Natural Language Questions via Phrasal Semantic Parsing. QALD-4 (2014)
26. Song, D., Schilder, F., Smiley, C. and Brew, C.: Natural Language Question Answering and Analytics for Diverse and Interlinked Datasets. The 2015 Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies. NAACL HLT 2015 (2015)