Querying for Information Integration: How to go from an Imprecise Intent to a Precise Query?

Aditya Telang, Sharma Chakravarthy, Chengkai Li

Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX 76019, USA
{aditya.telang, sharmac, cli}@uta.edu

Abstract

In this paper, we address the problem of query formulation in the context of multi-domain integration of heterogeneous data on the Web. We argue that effectively tackling this problem requires solutions to query specification and refinement, development and organization of domain taxonomies, and designing query templates to incorporate spatial and temporal conditions across multiple domains. We discuss our approaches in designing the *query formulation* component for *InfoMosaic*, our proposed framework for multi-domain information integration.

1 Motivation

Today, the problem of information integration has assumed a completely different, complex connotation than what it used to be. The advent of the Internet, the proliferation of information sources on the surface Web and the deep Web, the presence of structured, semi-structured, and unstructured data – all have added new dimensions to the problem as known earlier. Although existing frameworks have successfully addressed a number of data integration issues [9], an important challenge that has not garnered significant attention is the problem of *query formulation*.

In the context of information integration, user queries tend to span multiple sources and involve a number of conditions (e.g., spatial, temporal, generic) depending upon the heterogeneity of data types. For instance, consider a sample user intent: "*Retrieve castles reachable by train within 2 hours from London*". It is evident that this intent spans multiple sources and involves spatial as well as temporal constraints. All the information for answering this intent is available on the web; however, it is currently not possible to formulate such an intent as a *search/query* and obtain meaningful results. Although a multitude of retrieval mechanisms have been designed and successfully applied for formulating searches on the surface Web and queries over the deep Web, they prove to be inadequate in supporting the formulation of user intents such as the one shown earlier. For instance, mechanisms such as search engines, meta-search engines, faceted-search systems, question-answering frameworks, etc. only perform lookup of the keywords in the user query across Web documents followed by a ranking of the results. Similarly, query mechanisms such as the interfaces exported by deep Web sources (e.g., Amazon, Yahoo Travel), integration architectures (e.g., Ariadne [1], Google Base [2]), etc. are designed for a single or a set of pre-defined sources based on the schema of the underlying sources and thus, are restrictive in terms of the flexibility offered to users in framing queries.

In the paper, we address the problem of *query formulation* in the context of *InfoMosaic*, a generic framework we are currently developing for addressing the problem of heterogeneous information integration across multiple Web domains. As Figure 1 indicates, the idea is to accept minimal user intent, formulate one or more queries using a knowledge-base, select the one the user is interested in, transform the query into a detailed plan and evaluating it on actual information sources and perform integration of information from multiple sources. The results will be ranked and displayed to the user. This paper discusses two approaches we are developing for solving the query formulation problem, and establishes the challenges that need to be tackled for using these approaches.

2 The Query Formulation Problem

Ideally, a *query formulation* component should be able to accept minimal input from the user and refine that into a *complete* query that corresponds to the user intent and contains necessary information for processing it on the web sources. For instance, in response to

International Conference on Management of Data
COMAD 2008, Mumbai, India, December 17–19, 2008

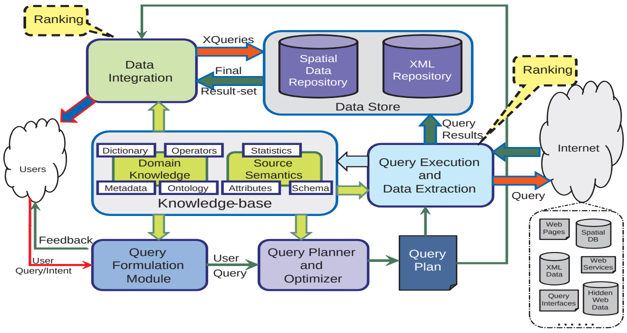

Figure 1: InfoMosaic Architecture

an input specified for the query intent shown earlier, the final *complete* query generated by the formulation component (assuming a variant of SQL representation) could be:

```
SELECT  *
FROM
  SOURCES      www.castles.org, www.national-rail.com
  IN DOMAINS   tourist attractions, transportation
  FOR ENTITIES castle, train
WHERE
    SPATIAL CONDITIONS
        train.source       =     'London'
        train.destination =     castle.location
        castle.location  near  'London'
    TEMPORAL CONDITIONS
        train.start_date  =     09/19/2008
        train.return_date =     09/19/2008
        train.duration    <     2 hours
```

Simple as it appears, automating such a *query formulation* process presents several challenges. As pointed out in [7], Web users prefer to express their intent in a succinct manner (such as keyword queries) and desire minimal interactions with the underlying system. Although these user preferences work well for search engines and database systems where the result of a user query is a relevant list of answers (Web links in the case of search engines and tuples in the case of database systems), it is not easy to satisfy these preferences in the context of query formulation where one needs to generate a complete structured query that matches the user intent (instead of an answer-list). For example, taking the input: "*castle*, *train*, *London*" (for the sample intent shown earlier), it is not possible to generate a complete query (shown above) with *no additional interactions* with the user.

The usage of statistics from past user queries is a possible alternative for formulating a succinct keyword intent into a complete query; however this technique would be inadequate as multiple potential queries can possibly map the user intent. For instance, some of the queries representing the above keyword intent could be: i) *Retrieve Castles near London that are reachable by Train*, ii) *Retrieve Hotels near London that are Castles and can be reached by a Train*, iii) *Retrieve Books whose title contain the words 'Castle' or 'Train' written by an author whose name is 'London'*, etc., and it is difficult to resolve which of these represent the actual user intent. Moreover, for a minimal set of input keywords, the data for values of certain attributes (e.g., *train.start date*, *train.return date*, *castle.location*), needed to be fed to underlying Web sources to obtain relevant data, and in the absence of additional user interactions, it would be difficult to deduce these values.

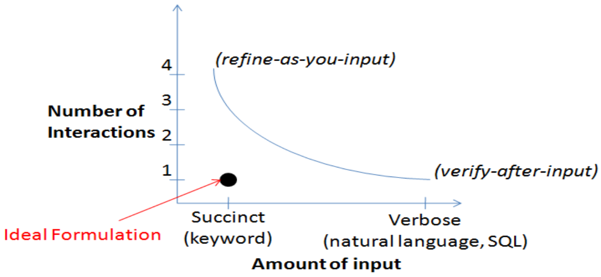

Figure 2: The Formulation Challenge

An additional challenge in this formulation process is elucidated by the final complete query that contains several components such as the Web domains (e.g., tourist attractions), the sources to be used for querying (e.g., www.castles.org), the entities involved (e.g., castle), the attributes of interest (e.g., location), operators (e.g., near), and values (e.g., London). The formulation component needs to possess the knowledge about these different types of information associated with user intent. In addition, information about synonyms and heteronyms also needs to be associated since the heterogeneity of the Web can lead to different keywords having the same meaning (e.g., source-city and start-city indicate the same concept in the domain of *transportation*) or the same keyword subsuming different meanings (e.g., castle can indicate a *tourist attraction* as well as a *hotel*). This gives rise to the challenge of discovery and organization of such extensive knowledge across multiple Web domains.

3 Our Approach

As Figure 2 shows, the ideal method for formulating a query involves minimal input and minimal interactions. However, as evident from the challenges discussed in the previous section, formulating it by satisfying both these parameters seems to be a hard task.

Hence, we propose two approaches – *refine-as-you-input* and *verify-after-input* that relaxes one of the above parameters respectively and helps in transforming the intent to a complete query. We also discuss our approach to the creation of a *knowledge-base*, that would contain the necessary pieces of information (domains, sources, entities, ...) required by the formulation component. In addition, we also describe our approach to some of the other challenges that may arise as a part of the formulation component.

- **Approach 1: Refine-as-you-input:** In this approach the parameter of *minimal interactions* is relaxed for ensuring *minimal input* from the user in building the complete query. At each interaction, the user provides a limited input (from a set of choices based on the information in the knowledge-base) that facilitates the completion of the query. Initially, the user provides the keywords representing the basic **entities** of interest (e.g., {*castle*, *train*, *city*} for the sample query intent). These keywords are checked against the *knowledge base* and in the case of heteronyms, they would be ranked and shown to the user to choose the exact intent. Subsequently, a query template (in stages for various types of conditions) will be generated based on the information associated with these entities in the *knowledge base*. The possible list of simple conditions (e.g., *train.startTime* {*genericOperator*} {*value*}) as well as integration conditions (e.g., *castle.location* {*spatialOperator*} *train.startLocation*) would be displayed.

The user would fill/modify such a template in a manner similar to filling Web query interfaces so that an unambiguous query is formed at the end. Although this approach seems restrictive in terms of user input and involves a minimum of three to four steps, we believe that such an approach will be useful for a novice user in understanding and using the system.

• **Approach 2: Verify-after-input:** This approach will reduce the number of *interactions* by relaxing the *minimal input* parameter. The motivation behind this approach is to accept a verbose input that represents the entire intent. This input will be matched against the *knowledge-base* for attribute coverage and occurrence to refine and generate the complete query. It may be possible to not only verify but resolve some misrepresentations (such as attribute names and conditions) using the meta-data in the *knowledge-base*.

An important issue to be addressed in this approach is to determine the method of specifying the entire intent. An intuitive approach would be to allow the user to express the intent in natural language. This is certainly a preferred alternative; however, to the best of our knowledge, the capability to accept arbitrary natural language queries and convert them to structured queries does not exist. Alternately, a feasible approach would be to display an empty query template such as:

```
SELECT                  <output attributes>
FROM
    SOURCES              <sources>
    IN DOMAINS           <domains>
    FOR ENTITIES         <entities>
WHERE
    GENERIC CONDITIONS  <generic conditions>
    SPATIAL CONDITIONS  <spatial conditions>
    TEMPORAL CONDITIONS <temporal conditions>
```

and ask the user to fill and complete the entire query. However, this would require the user to have an underlying knowledge about the data model, the schema for different entities, the attribute and their meta-data (ranges, values, types, etc.), and the set of feasible conditions possible for the intent. These constraints would require the user to have certain level of experience with the capability of the system and hence, although it ensures minimal interactions, this approach would be better suited as the user gains confidence in using the system and is familiar with the domains handled by the system.

• **Knowledge Base:** Both the approaches require different types and extents of information about the entities, domains, sources, operators, attributes and values at different stages of the formulation process. The usage of taxonomies and ontologies for representing the entities within a domain and their relationships seems to be a natural choice. Similarly, a mapping between the entities (e.g., train) and the underlying Web sources (e.g., national-rail.com) that provide data for these entities has been clearly established. Furthermore, analysis of the sources for understanding the attributes associated with the entities and the supported meta-data (data type, range of values, operators supported, etc.) is necessary for formulating query conditions in the *refine-as-you-input* approach. In addition, understanding the compatibility between domains and entities in forming integration conditions and the possible set of operators that can defined over these operations seem to be necessary. The categorization of attributes for entities that participate in spatial and temporal conditions can further aid the formulation process.

Additionally, we also maintain a workload comprising the collection of statistics and feedback associated with past queries such as – users' preference for keyword meanings in the case of heteronyms, popularity of attributes for entities in a taxonomy in forming query conditions, etc. This workload repository would be a constantly evolving collection of statistics based on the queries generated from user input.

We would like to clarify that in our work, we do not address the discovery (or automatic acquisition) of information used for this approach. To the best of our knowledge, the current process [5] of designing taxonomies and ontologies is done manually with a small amount of automation involved for particular domains such as science, literature, etc.. Hence, in our framework, we intend on using the different pieces of existing information available in the form of Web directories (e.g., Yahoo Directory), domain ontologies (e.g., OntoBuilder [5]), language repositories (e.g., WordNet), domain portals (e.g., Yahoo Travel) and using the basic theories of taxonomy creation to manually construct the knowledge base for a few domains in the prototype system.

• **Ranking Model:** In the *refine-as-you-input* approach, the input keyword, if a heteronym, may give rise to multiple query combinations (e.g., the keyword *castle* may map as a travel attraction to the domain of *recreation* and as a hotel to the domain of *lodging*). We are investigating the applicability of a ranking model that can aid in pruning some of the meanings for a keyword based on other keywords in a query followed by ordering the remaining meanings based on metrics such as semantics or statistics. We are currently working towards understanding the parameters that could influence ranking and formulating such a generic model.

4 Evaluation Plan

The two approaches we discussed would formulate a complete query by relaxing one of the two parameters (input and interactions). Our larger goal is to determine how close these approaches can converge to the ideal formulation in Figure 2 without losing the capability of performing *complete* query formulation. Hence, a benchmark needs to be established that defines the effectiveness of these approaches that will then direct us to focus on the larger goal.

However, evaluating these approaches and the effectiveness of the query formulation component is definitely not a trivial issue. In the *refine-as-you-input* approach, the goal is to reduce the number of interactions, and hence, we plan to test this parameter for evaluating the overall approach. The impact of the workload repository and knowledge-base coupled with an effective ranking model would need to be tested to determine their effect on reduction of interactions.

In the second approach, our goal is to evaluate least *minimal input* that needs to be provided to generate a query with only one user interaction. For instance, the effectiveness of the approach if certain components such as sources, domains, condition-values, etc. are not specified would need to be tested. However, since the query formulation in the second approach also depends on the experience of the user, evaluating this approach seems to be a tricky issue.

The crucial part at this stage is to determine a reasonable benchmark for deciding the effectiveness of these approaches. On a Web scale, it is difficult to evaluate these approaches unless it is actually tested by the users. Our current plan is to test them in a relational database environment with SQL queries. We are currently investigating the formalization of such a setup that could set the necessary benchmarks for the evaluation of the component.

5 Related Work

Query formulation in existing integration frameworks is fairly simple since the focus is on integrating data from multiple sources in a single or a set of fixed domains. For instance, Havasu [6] supports a keyword search on it BibFinder application to extract data for technical publications from multiple sources in a single domain (Literature). Systems such as Ariadne [1], Whirl [3], etc. export mediated query interfaces for fixed set of domains that allow user to formulate queries similar to the ones in deep Web sources. Commercial systems such as Google Base [7] and desktop-level data-space applications [4] advocate the usage of keyword queries. Building queries for integration in a demonstrative manner was investigated in[10] and [8]. However, the focus of these frameworks is to perform a simple text/Web-search to obtain different types of data in response to the keywords (e.g., blogs, web-links, videos, etc.) instead of formulating a query where every keyword corresponds to a distinct entity. To the best of our knowledge, we are the first ones to address multi-domain information integration over arbitrary domains and work towards designing a query formulation component that is not bound to the underlying schema.

6 Conclusion

In this paper, we introduced the problem of formulating queries in the context of information integration. We articulated the challenges associated with designing algorithms and techniques for the query formulation problem and how extant techniques are not adequate for addressing this problem. We discussed alternative approaches for formulating queries in the context of our proposed *InfoMosaic* framework and highlighted the importance of different types of meta-information needed for the same.

References

[1] J. L. Ambite, C. A. Knoblock, M. R. Kolahdouzan, M. Muslea, C. Shahabi, and S. Thakkar. The WorldInfo Assistant: Spatio-Temporal Information Integration on the Web. In *VLDB*, pages 717–718, 2001.

[2] T. Cheng and K. C.-C. Chang. Entity Search Engine: Towards Agile Best-Effort Information Integration over the Web. In *CIDR*, pages 108–113, 2007.

[3] W. W. Cohen. Providing Database-like Access to the Web Using Queries Based on Textual Similarity. In *SIGMOD Conference*, pages 558–560, 1998.

[4] M. Franklin, A. Halevy, and D. Maier. From Databases to Dataspaces: A new abstraction for Information Management. *SIGMOD*, 34(4):27–33, 2005.

[5] A. Gal. Why is schema matching tough and what can we do about it? *SIGMOD Rec.*, 35(4):2–5, 2006.

[6] S. Kambhampati, U. Nambiar, Z. Nie, and S. Vaddi. Havasu: A Multi-Objective, Adaptive Query Processing Framework for Web Data Integration. Technical report, Arizona State University, 2002.

[7] J. Madhavan, S. Cohen, X. L. Dong, A. Y. Halevy, S. R. Jeffery, D. Ko, and C. Yu. Web-Scale Data Integration: You can afford to Pay as You Go. In *CIDR*, pages 342–350, 2007.

[8] A. Nandi and H. V. Jagadish. Assisted querying using instant-response interfaces. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1156–1158, July 2007.

[9] A. Telang and S. Chakravarthy. Information Integration across Heterogeneous Domains: Current Scenario, Challenges and the InfoMosaic Approach. Technical report, University of Texas, Arlington, 2007.

[10] R. Tuchinda, P. Szekely, and C. A. Knoblock. Building data integration queries by demonstration. In *Proceedings of the 12th International Conference on Intelligent User Interfaces*, pages 170–179, 2007.