Research Proposal for Distributed Deep Web Search

Kien-Tsoi T.E. Tjin-Kam-Jet
University of Twente
The Netherlands
tjinkamj@cs.utwente.nl

ABSTRACT

This proposal identifies two main problems related to deep web search, and proposes a step by step solution for each of them. The first problem is about searching deep web content by means of a simple free-text interface (with just one input field, instead of a complex interface with many input fields). To this end, we propose a real-time query conversion layer to translate a free-text query into a structured query. The second problem concerns the scalability of the system, and we propose to use a distributed approach.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation*; H.5 [**Information Interfaces and Presentation**]: User Interfaces—*Natural language*

General Terms

Languages, Experimentation, Measurement

Keywords

Deep web, query translation, query reformulation, natural language interfaces

1. INTRODUCTION

Centralized search engines like Bing and Google use crawlers to download web content and build an inverted index so that users can quickly search within the content. Crawlers are given a set of seed pages and recursively download content by following the links on the downloaded pages. However, many pages on the web are hidden behind web forms and are inaccessible to crawlers. These pages are commonly referred to as the deep web [7, 19], in contrast, those pages accessible by following links are referred to as the surface web. The number of deep web pages is estimated to be up to two orders of magnitude larger than the surface web [7, 10]. This content is often of high quality and highly relevant to the user's information need. In other words, it is of crucial importance to provide adequate deep web search functionality. For the remainder of this proposal, the inaccessibility of deep web pages to web crawlers will be referred to as the *deep* problem.

Another problem related to web search is its immense size and continuous growth, which poses many challenges and hard requirements on the scalability of any web search solution [6]. In 1999, it was estimated that no web search engine indexes more than 16% of the surface web, and that the web consisted of 800 million pages [18]. In 2005, a new estimate put this number at 11.5 billion pages [14], and in 2008, Google announced[1] the discovery of one trillion unique URLs on the web at once. The extremely large number of web pages and the continuous web growth will be referred to as the *big* problem.

The following scenario illustrates some of the hurdles when searching for deep web content. Imagine that you are planning a short trip. You are gathering information about possible routes and trying to determine the preferred means of public transport: whether to go by bus, metro, train, or a taxi. In addition, you are comparing them to the costs and benefits of traveling by car. To gather such information, you must submit a structured query to a complex web form (i.e. a form with multiple input fields) like the form in Figure 1. Chances are that you will have to re-type your complete query several times, once for each site about a particular means of transportation. This repeated process is tiresome. Furthermore, you must first find the right sites to query, otherwise you might not even find the (best) solution.
It would be much easier if one could submit a single free-text query to a simple search interface like the one in Figure 2, and search many complex forms at the same time (especially if one does not know about their existence).

The two problems together with the given scenario lead to the following research questions:

Question 1 How to automatically convert free-text queries into structured queries for complex web forms, in order to solve the *deep* problem?

Question 2 How to adapt the solution to Question 1, such that it can cope with the *big* problem, thereby enabling deep web search?

Outline of proposal: The two following sections each discuss one research question in detail, stating: the scientific chal-

[1] `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`

PIKM'10, October 30, 2010, Toronto, Ontario, Canada.

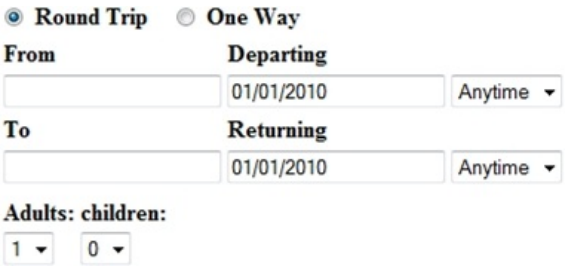

Figure 1: A complex web form

Figure 2: A simple web form

lenges, related research, and the research method. Finally, a global planning is given in the final section.

2. EXPOSING THE "DEEP" PROBLEM

He et al. [16] report that major search engines each were able to index part of the deep web. However, almost two thirds of the deep web was not indexed by any engine, indicating certain inherent barriers for crawling and indexing the deep web.

Current approaches related to searching deep content include universal search[2] of enterprise verticals[3][5, 12]; domain specific mediators like `www.cheapflights.com`; and surfacing [19, 2], i.e. automatically filling in and submitting web forms, and indexing the resulting web pages.

The first two approaches are not general solutions to distributed deep web search. The verticals (and the accompanying query brokering system) are maintained by the same company, so they have complete knowledge of each vertical's index and querying interface, therefore they can build a custom system that works in their environment. Mediator frameworks are often set up by collaborating companies that allow access to their databases. These frameworks do not crawl and index, instead, they broadcast every query to all databases. The mediator often has a complex web form to ease the conversion of the query to the specific query format of each database.

The third approach, surfacing, is a more general solution towards deep web search (since there is no collaboration between companies). However, there are deep web sites for which surfacing is not suitable, for example, sites that offer traveling schedules. Such indexed web pages would get outdated quickly.

A better solution would be to transform the query on-the-fly, submit the transformed query to the deep web site, and show those results to the user. Therefore, we propose, and will examine, the idea of a query-converter layer on top of each deep web resource, which translates free-text queries to queries in the deep resource's query format.

[2]`http://www.google.com/intl/en/press/pressrel/universalsearch_20070516.html`

[3]A search system dedicated to a certain medium or topic, such as news, or videos

2.1 Scientific challenges

In regard to the *deep* problem, the challenges are given below and are divided into two different groups: query conversion (the first three) and user interfacing (the last two).

Query description A formal syntax in which web administrators can specify the accepted language of the particular resource. How can we keep this intuitive and simple, while allowing enough freedom to specify almost any kind of query, and strict enough to allow easy parsing?

Query translation Due to possible spelling errors, ambiguity, or unknown words to the system, extracting the intended meaning of free-text queries is challenging. A query could be interpreted in different ways. How to devise a feasible approach that achieves reasonable performance (e.g. correctly interprets and translates over, say, 75% of the queries)?

Interpretation ranking As stated in the previous point, a query could be interpreted in many ways. How to rank these interpretations in order to minimize the user's effort to scan through all interpretations, thus quickly finding the right one?

User ignorance How to bridge the gap between the expectations of the user and the capabilities of the system? Is it feasible to automatically suggest available search facets while typing (i.e. the aspects in which the search query can be narrowed further to obtain more specific results)? How to automatically choose suggestions such that the user: 1) is guided while formulating more distinctive queries, and 2) can finish formulating the query faster?

System ignorance How to automatically expand the system's knowledge about valid queries? For example, given a query that contains unknown words, the system presents several annotated interpretations. Then, if the same query is given many times and a particular interpretation is often selected, the system could learn a new rule which includes the unknown word.

Of these challenges, the main focus of this research will be on the query description, query translation, and ranking challenges.

2.2 Related research

He et al. [17] worked on translating structured source queries into differently structured target queries, using a two-step process: semantic mapping and syntax construction. However, our notion of query translation is more related to the field of natural language interfaces to databases [3, 11]. Indeed, the goal is to convert free-text queries — which might be given in a natural language like English — into structured queries suitable to be sent to a complex web form.

The input fields of web forms typically restrict the accepted text to a specific kind of information. Therefore, it is vital to recognize and extract all valid pieces of text, and label them with the corresponding input field. Patterns in the form of grammar rules (e.g. in the form of regular expressions) can be used to perform this information extraction [4]. Agarwal et al. [1] recently mined a big search log and found

many patterns. This also supports our focus on patterns, as users apparently do exhibit patterned search behavior.

2.3 Research method

A prototype will be built that converts free-text queries into structured queries that are suited for some deep web site, i.e. which has a form with multiple input fields. The interface will look like that of a simple search engine (e.g. a text box and a search button), so that the user can freely enter any text to search for.

Comparative user studies will then be performed to assess how users finish a pre-defined set of search tasks with a standard system and the newly built prototype. Among the measurements will be: task completion time, user satisfaction, the use of query suggestions, result ranking, and the query translation effectiveness (i.e., the percentage of correctly translated queries).

3. EXPOSING THE "BIG" PROBLEM

DIR (Distributed Information Retrieval) [8] can potentially solve the *big* problem. In a DIR scenario, a user queries a central broker which then distributes the query to several remote resources (e.g. search engines). The broker then receives results from each queried resource and merges these into one final result list which is then presented to the user. Intuitively, as new websites are created, a small additional search engine would be installed for indexing those sites. In the extreme, all web hosting servers could index their local content (there would be no crawling) and participate in the DIR system. All indices would be up to date, and everything would be searchable. The big obstacles are which resources to select for actual querying, and how to merge their results.

Before any selection can be made by the broker, it must have some description about the contents of each resource. Typically, a sample of the resource's index is used as a description of the resource [9]. Resource selection often treats all samples as (very large) documents and applies standard IR techniques for selecting the top(k) resources [22, 24, 5, 12]. Finally, results merging can be based on several features, such as, the resource's rank (obtained from the previous step), or the result's rank given by the remote resource [23, 20, 21].

DIR, due to its distributed nature, can take natural advantage of deep web resources if we could incorporate our query-conversion solution from Question 1. A query could then be sent to any deep web site, thereby enabling distributed deep web search. Therefore, we will examine the idea of a query converter layer between the broker and a deep resource, and experiment with different strategies concerning: query translation, resource selection, and results merging.

3.1 Scientific challenges

With regard to the *big* problem, and the proposed distributed search solution, many challenges largely stem from the field of distributed information retrieval.

Resource description An index-sample describes the contents of a resource. However, resources could also be described by their accepted queries. A resource description should facilitate the process of resource selection. How to possibly adapt and use the query description from Section 2.1.1, not just for query translation, but also for resource selection?

Resource selection Traditionally, standard IR techniques are applied to rank and select the top(k) resources. Blindly applying these techniques to our resource descriptions will not work, since we describe the accepted queries instead of the resource's contents. How to rank these resources in order to select the top(k), given our resource descriptions? How to determine k?

Results merging A problem of re-ranking a set of results by their relevancy in order to maximize retrieval precision. One resource may return many relevant results, while another may return very few. How to determine the number of results to retrieve from each resource? How to measure their relevancy, in order to rank by relevancy?

Suggestion ranking The first few suggestions shown to the user, as the user starts typing, could be ranked by their popularity. But as the query gets longer and more discriminating, it might make sense to generate more deep-resource-specific suggestions. More generally, for any given query, multiple deep resources could generate appropriate and relevant suggestions. Similar problems to resource selection and result merging apply here: which suggestions of which resources to show (resource selection), and how to rank (merge) these suggestions?

Of these challenges, the main focus of this research will be on the resource description, resource selection, and results merging.

3.2 Related research

Distributed deep web search also brings us to the concept of *Dataspaces* [13, 15], a visionary data management abstraction where all data sources are interconnected and every data source, regardless of its storage structure, would support at least some form of free-text search. As such, our work naturally contributes to some of the scientific challenges addressed in [15], which are cited here for convenience:

- *Sub-challenge 1.3. Develop algorithms that given a keyword query and a large collection of data sources, will rank the data sources according to how likely they are to contain the answer.*
- *Sub-challenge 1.4. Develop methods for ranking answers that are obtained from multiple heterogeneous sources (even when semantic mappings are not available).*
- *Sub-challenge 2.2. Develop a formal model for approximate semantic mappings and for measuring the accuracy of answers obtained with them.*

As can be seen, Sub-challenge 1.3 closely corresponds to the challenge of resource selection, Sub-challenge 1.4 corresponds to the challenge of results merging, and finally, Subchallenge 2.2 corresponds to our challenge of query translation.

3.3 Research method

The prototype system will be expanded with a broker. At first, it will simply broadcast the query to all resources. Resource selection strategies will be incrementally developed, added to the system, and evaluated. In particular, we will

start with strategies based on whether or not, or to what extent, the query can be converted to the format of the deep resource.

User studies will then be performed to assess how users finish a pre-defined set of search tasks with the newly built prototype. Among the measurements will be: task completion time, user satisfaction, the use of query suggestions, result ranking.

4. GLOBAL PLANNING

First, we will build custom query-conversion prototypes for product-sales web sites (for example `www.gaspedaal.nl`, where you can enter for instance the car make, car model, mileage, car age, and price range) and for web sites about traveling (such as `www.ns.nl`, the web site of the Dutch railway company, where you can enter for instance the time, date, arrival and departure locations). The aim is to have at least 3 such converters for different sites, preferably of different domains.

Second, we will conduct an extensive user survey and evaluate the prototype.

Third, the prototype broker will be built, and functionality will be added incrementally. For instance, at the start there will be no resource selection, the query will simply be broadcasted to all resources. This would serve both as a sanity check that the system really works, and as a baseline for comparing amongst others: retrieval performance, total query time, and network traffic.

Fourth, selection mechanisms will be developed and evaluated "offline", at first, to see if selection works reasonably as expected. Afterwards, several (simple) results merging algorithms will be implemented and then an extensive user study will be performed, for evaluating the whole system.

5. ACKNOWLEDGEMENTS

Many thanks to my supervisor Djoerd Hiemstra and colleague Dolf Trieschnigg for their valuable input and guidance in writing this proposal. This research is funded by the Netherlands Organization for Scientific Research, NWO, grant 639.022.809.

6. REFERENCES

[1] G. Agarwal, G. Kabra, and K. C.-C. Chang. Towards rich query interpretation: walking back and forth for mining query templates. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 1–10, New York, NY, USA, 2010. ACM.

[2] M. Álvarez, J. Raposo, A. Pan, F. Cacheda, F. Bellas, and V. Carneiro. Deepbot: a focused crawler for accessing hidden web content. In *DEECS '07: Proceedings of the 3rd international workshop on Data enginering issues in E-commerce and services*, pages 18–25, New York, NY, USA, 2007. ACM.

[3] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. Natural language interfaces to databases – an introduction. *Natural Language Engineering*, 1(01):29–81, 1995.

[4] D. E. Appelt and B. Onyshkevych. The common pattern specification language. In *Proceedings of a workshop on held at Baltimore, Maryland*, pages 23–30, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

[5] J. Arguello, F. Diaz, J. Callan, and J.-F. Crespo. Sources of evidence for vertical selection. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322, New York, NY, USA, 2009. ACM.

[6] R. Baeza-Yates, C. Castillo, F. Junqueira, V. Plachouras, and F. Silvestri. Challenges on distributed web retrieval. 2007.

[7] M. K. Bergman. The deep web: Surfacing hidden value. *Journal of Electronic Publishing*, 7(1), August 2001.

[8] J. Callan. Distributed information retrieval. In *Advances in Information Retrieval*, pages 127–150. Kluwer Academic Publishers, 2000. Croft, W. B. (Ed.).

[9] J. Callan and M. Connell. Query-based sampling of text databases. *ACM Trans. Inf. Syst.*, 19(2):97–130, 2001.

[10] K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang. Structured databases on the web: observations and implications. *SIGMOD Rec.*, 33(3):61–70, 2004.

[11] A. Copestake and K. S. Jones. Natural language interfaces to databases. *The Knowledge Engineering Review*, 5(04):225–249, 1990.

[12] F. Diaz and J. Arguello. Adaptation of offline vertical selection predictions in the presence of user feedback. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 323–330, New York, NY, USA, 2009. ACM.

[13] M. Franklin, A. Halevy, and D. Maier. From databases to dataspaces: a new abstraction for information management. *SIGMOD Rec.*, 34(4):27–33, 2005.

[14] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 902–903, New York, NY, USA, 2005. ACM.

[15] A. Halevy, M. Franklin, and D. Maier. Principles of dataspace systems. In *PODS '06: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–9, New York, NY, USA, 2006. ACM.

[16] B. He, M. Patel, Z. Zhang, and K. C.-C. Chang. Accessing the deep web. *Commun. ACM*, 50(5):94–101, 2007.

[17] B. He, Z. Zhang, and K. C.-C. Chang. Metaquerier: querying structured web sources on-the-fly. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 927–929, New York, NY, USA, 2005. ACM.

[18] S. Lawrence and C. L. Giles. Accessibility of information on the web. *Intelligence*, 11(1):32–39, 2000.

[19] J. Madhavan, D. Ko, L. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy. Google's deep web crawl. *Proc. VLDB Endow.*, 1(2):1241–1252, 2008.

[20] G. Paltoglou, M. Salampasis, and M. Satratzemi. Hybrid results merging. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on*

information and knowledge management, pages 321–330, New York, NY, USA, 2007. ACM.

[21] G. Paltoglou, M. Salampasis, and M. Satratzemi. Results merging algorithm using multiple regression models. In *ECIR'07: Proceedings of the 29th European conference on IR research*, pages 173–184, Berlin, Heidelberg, 2007. Springer-Verlag.

[22] Y. Rasolofo, F. Abbaci, and J. Savoy. Approaches to collection selection and results merging for distributed information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 191–198, New York, NY, USA, 2001. ACM.

[23] M. Shokouhi and J. Zobel. Robust result merging using sample-based score estimates. *ACM Trans. Inf. Syst.*, 27(3):1–29, 2009.

[24] L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 298–305, New York, NY, USA, 2003. ACM.