

Where are the ‘Killer Applications’ of Restricted Domain Question Answering?

Michael Minock

Umeå University

Department of Computing Science
C445 MIT-huset Umeå, Sweden 90187
mjm@cs.umu.se

Abstract

From a language technologist’s point of view, the penetration of natural language interfaces onto today’s web is somewhat disappointing; it seems that information retrieval, forms based, metaphor-based and hyper-link interfaces dominate all points of the design space. While open domain question answering promises to rival or extend information retrieval systems, restricted domain question answering systems likewise represent a rival to forms-based interfaces. The purpose of this position paper is to discuss the properties of potential web-based ‘killer applications’ of restricted domain question answering. The paper entertains a set of candidate domains, proposes a general methodology for building restricted domain interfaces and highlights some near term challenges that must be confronted.

1 Introduction

Although open domain question answering is a very promising area, the position in this paper is that an effort must likewise be undertaken for a set of promising restricted domains. Though one may view restricted domain question answering as a special case of open domain question answering, employing an isolated set of domain documents and perhaps a tailored lexicon and grammar, the view here is more expansive; it is assumed that domain knowledge and data are explicitly represented to one degree or another. This enables the introduction of domain concepts and facts in addition to domain ‘documents’. It also points toward a revival of question answering over databases [1; 3]. In any case a resulting restricted domain question answering system, with access to domain knowledge, is presumably more able to use reasoning in support of question answering. The answer itself is expected to be direct, consisting of some combination of natural language, tabular data, graphs, video clips, documents, etc.

This paper shall refrain from making explicit assumptions about the nature of a representation backing a restricted domain system, however it is assumed that it

will consist of a conceptual model (or ontology) which covers some significant number of entities (instances or assertions). We assume that this domain information is built-up and maintained in some coherent, quality state through some combination of fact extraction from a set of documents and hand crafted knowledge representations or databases. Furthermore, and perhaps optimistically, we assume that a suitable natural language interface may be built over a given domain that: 1.) lets the user pose questions in a full natural language; 2.) offers paraphrases when user questions are unclear or ambiguous; 3.) accurately describes answers to avoid misunderstandings. Given these assumptions, the thought experiment in this paper is to ask, *assuming that problems of representation and natural language processing can be handled, what domains are ripe for restricted domain question answering on the web?* Since most web users are already comfortable with key-word search, forms and hyper-links, the application of natural language question answering to a restricted domain must somehow eclipse these techniques, individually or in combination.

Section 2 of this position paper outlines some desiderata for promising restricted domains. Section 3 proposes a set of restricted domains and discusses these proposals in relation to the desiderata. Section 4 proposes a general development methodology for building restricted domain question answering systems. Section 5 discusses some of the high priority challenges that need to be addressed to open the way for the deployment of such systems. Section 6 concludes this position paper.

2 Desiderata for Restricted Domains

To enjoy success on the web, a restricted domain must be:

D1: Circumscribed

D1.1: topic is focussed

D1.2: level of detail is evident

D1.2: knowledge may be represented and data is factual

D2: Complex

D2.1: spans more than several concepts

D2.2: entities have complex properties and are involved in complex relationships

D2.3: numerous entities

D3: Practical

D3.1: answers to single sentence questions are useful to an identified group of users

D3.2: data and knowledge acquisition and maintenance is feasible

D3.3: query volume is high

2.1 Circumscribed

If D1.1 is adhered to a user may ascribe a bounded understanding to the system and thus revisit the system when they wish to know something about the domain. Normally this is not too difficult to achieve. If for instance we pick the domain of ‘Aquarium Fish: Species, Habits and Care’ a user would have a reasonable idea of what the domain encompasses, though they would perhaps need to experiment a bit to get an idea. If the domain is too general, say ‘Current events’, then the user might have a bit more difficulty ascertaining and remembering what the system really ‘knows’ about.

Though D1.1 bounds the topic in conceptual space, D1.2 bounds the level of detail that is to be captured. For example assume that we have the topic ‘World War II: leaders, footage and battles’. It may be that we could ask questions such as “show some footage of the battle of Britain”, but of questions such as “which was the first suburb of Minsk to fall to the Nazi armies?” are not likely to be represented.

Another consideration is whether the information of the domain is simple enough to be adequately captured with contemporary knowledge and data representation. For example the aquarium domain is probably somewhat simpler ontologically than the World War II domain. The judgement of the knowledge requirements is qualitative, but, unlike our computers we do have common sense and can rate certain applications as requiring significantly more knowledge representation requirements than others.

2.2 Complex

The key issue here is to insure that the domain has sufficient structure to warrant a question answering interface. For example if D2.1 is not adhered to and the domain is spread over only a few concepts (tables), then a forms based interface will probably suffice. For example if we consider flight information of a particular airline (e.g. www.sas.se), we see that a forms based interface suffices. At one point however, when the schema spans more than a few concepts, standard forms based approaches break down. It is at this point that other techniques become necessary such as query construction tools, visual query languages, hierarchies of forms, natural language menus, etc. It stands to reason that such a point is also where natural language question answering may find a break through as well.

If, in violation of D2.2, the objects of user interest really don’t have complex properties associated with them other than their own names or textual content, then perhaps a key-word based search techniques is most appropriate. Consider for example a set of documents that have only very coarse descriptors and sets of associated keyword. Since there are only very simple predicates that objects may satisfy, an ‘advanced search’ option that mixes keyword and simple predication (e.g. date, domain name, etc.) will suffice.

Naturally, as noted in D2.3, if the number of entities in the domain is limited then the user would probably be better off just reading a list or navigating through a hypertext document.

2.3 Practical

Since we expect single sentence questions, naturally a user should be able to get something of value as an answer. Thus if a domain fails D3.1 and the content is either too well known or of very marginal interest, then who will be interested in querying it? For example if one has a geography database of the capitals of various countries, who, other than those interested in testing the system, would really use such an interface. A better option is to click through the CIA world fact book. The same is true for a database over relatively uninteresting data. If for instance, I put my contacts and calendar into a database, who other than myself would really be interested in querying it. Again we have to use our common sense to gauge the how well a domain meets D3.1.

The consideration in D3.2 is how feasible is it to build a representation of the domain. In general the cost of constructing the database and its interface’s linguistic configuration must be justified by the quantity of queries that are posed over the lifetime of the interface (D3.3). Factors such as the timeliness of the data and the speed at which users wish to know such data enter into judging a candidate domain on these measures.

2.4 Correlations among the Desiderata

Naturally there are correlations among the desiderata above. For example if a domain qualifies on D2.1 then it is likely to qualify on D2.2 and D2.3 as well. In fact all the sub-desiderata of the three major desiderata seem to positively correlate. Moreover D1 and D2 on a whole seem to be inversely correlated.

Of course these desiderata are simply a set of characteristics arrived at through introspection and are merely subjective judgments, overlapping and probably incomplete. Still they provide a convenient starting point to organize the search for the illusive ‘killer application’ of restricted domain questions answering.

3 Some Candidate Domains

We now evaluate a set of application domains. These domains are introduced with a simple description followed by an opinion of how well they rate on the above desiderata.

- A1: **NLI to a photo album:** Pictures are classified based on when and where they are taken, by who, of what, of whom, etc. This application scores well on D1, but poorly on D2.1 and D2.2. A hybrid approach based on key word search of picture captions and forms based predications on picture date, size, location is deemed to be a superior approach.
- A2: **Simple geography facts:** Users may ask questions about cities, countries, languages, religions and ethnic groups. This application scores relatively well on D1 and D2, though it could be argued that D1.2 is a little weak. The problem is mostly D3.1. Still there is some hope that with more complete data and integration of maps, etc. that some type of useful interface will emerge. An initial example of such a domain may be accessed through the interface [6] at www.cs.umu.se/~mjm/step which is backed by the Mondial data set[5].
- A3: **Bus schedules:** An interface allowing residents to query for times and destination of busses within the local community. This application scores well on D1, but perhaps less well on D2, especially D2.1 and D2.3. For D3 it seems like it could generate a fair number of queries especially if the interface was readily accessible via mobile devices. An example interface of this type may be accessed at www.idi.ntnu.no/~tagore/bustuc/
- A4: **City events information:** An interface to city events, hotels, restaurants and other attractions. This type of application does a fair job on D1 and a good job on D2 if the city is large. Some issues surround D1.2. If the city is a popular tourist destination it could do well on D3.1 and D3.3 as well. D3.2 is something of a weak spot, but it could be offset by D3.3. The work [7] discusses such an interface for the city of Cottbus.
- A5: **Natural language assistant to a GIS:** Allow for natural language querying of a wide variety of locations on a map. For example “show the houses that are worth between 100k and 200k dollars that are less than 10km from a lake.” Such an application scores OK on D1, though there are some questions surrounding spatial representation in D1.3. Provided that there are numerous types of objects represented on the map with complex properties, it is likely that the application will score well on D2 as well. The site could generate enough interest to justify the considerable cost associate with D3.2. One specific idea is to build an interface to the UNESCO world heritage sites database.
- A6: **Nobel prize database:** A database that contains information on the winners of the Nobel prizes including nationality, pictures, age, academic affiliation, etc. This idea rates well on D1, and receives a perhaps just passing score on D2. The database itself does not seem to be difficult to build and does not need to be updated often. Finally the domain might be of interest to the public.
- A7: **World Cup scores and highlights:** Allows for users to query for games, player and team statistics for the world cup tournament. This domain rates well on D1 and D2. Although the cost of building a database may be considerable, if the interface worked well, the expected number of queries may also be expected to be high.
- A8: **Software catalog:** Allows for users to query for software with certain properties, running on different platforms, etc. This type of domain is a bit weak on D1. There are simply so many different concepts that might be involved. The system is certainly complex enough (D2) and for D3 there seems to be a great need for this type of service, though it must be said that building and maintaining the database is likely to be a very difficult task.
- A9: **Continental European travel information:** Allows users to pose queries over train schedules, attractions, hotels, etc. This domain has difficulties with D1 but does well on D2. It may have difficulties with D3.2, but this could be offset with D3.3. A fragment of this domain has been treated in [2].

4 A Proposed Development Methodology

The following is a proposed methodology to roll out a restricted domain question answering system:

- S1: Identify a domain with the above desiderata
- S2: Collect a large number of candidate questions over the domain.
- S3: Build a conceptual model that covers the bulk of questions from S2.
- S4: Define a representational model that corresponds to the conceptual model of S3.
- S5: Populate the representational model with 'documents' and facts
- S6: Configure the natural language interface over the representational model
- S7: Offer the system to a user community

Note that S2 may be carried out through a brainstorming session or through hidden operator (“wizard of Oz”) experiments with a sample user group. Note that steps S3, S4 and most importantly S5 are carried out after the selection of the domain; in general we view the prospects of using legacy databases without significant restructuring as overly optimistic. Step S6 is the configuration of the natural language interface over the restricted domain. Note that this step will probably involve significant work even if general linguistic grammars are used; mapping domain independent grammars to restricted domain models is very challenging and can only be partly automated. Finally in S7 the system is presented to a user community. At this point S7 becomes a

source of queries and the development iterates through steps S2 through S7 ad infinitum.

5 Some Challenges

Of course each domain of section 3 has different ontological requirements, and thus have separate challenges, not addressed here, that must be confronted. The challenges sited in this section apply to all the candidate domains.

5.1 Natural Language Processing

Needless to say, the natural language interfaces to restricted domains must be sophisticated. One key issue is that interfaces must support both generation and understanding; a system must be able to paraphrase user questions either during answer presentation or in the case of a low confidence parses. Another issue is that the systems will need to be able to support some special types of questions. In particular it seems that superlatives are a difficult type of question which will need to be parsed (e.g. “Give the person who has won the most Nobel prizes.”). A variant of this type of question is to request the top ranking set of answers (e.g. “Give the 10 most populated countries in Asia.”). Mapping such requests to the logic that expressed them is non-trivial. Finally it should be noted that the natural language interfaces must be linguistically complete enough to offer to parse the vast majority of questions over the domain. This includes some method to cope with non syntactic inputs as well as a tolerance for simple spelling errors.

5.2 Cooperative Query Answering

It seems that most if not all domains will need to have at least some support for cooperative query answering [4]. At its core this means support for automated or semi automated query generalization and specialization. These capabilities enable relaxation of queries which obtain no answers and query refinement when user queries are too broad. A capability of identifying false presuppositions within user queries is also important (e.g. “List cities in Grance with more than 1 million people” should be answered with “There is no country or region named ‘Grance.’”).

Somewhat outside the traditional topics of cooperative querying answering, systems must in one way or another handle conceptual and data incompleteness. Conceptual incompleteness may perhaps be handled by extending the domain model with dummy concepts near to the domain topic. Meta-level responses would be generated when user queries touch upon these out of range concepts. Additionally logical meta-data may be used in cases that there is data incompleteness (e.g. “Give the cities in Sweden” could be answered with “the cities in Sweden are ..., but the database only contains cities in Sweden with more than 20,000 people.”)

5.3 Open Evaluation

To facilitate better comparison of systems, restricted domain query answer prototypes should be available

for anonymous querying on the web. The page at www.cs.umu.se/~mjm/step lists all the systems of which I am aware (including my own system STEP [6]). Additionally, for those who do offer web demonstrations, it would be helpful to publish the system’s complete configuration. Naturally it would also be helpful if all domain data was available as well. If more groups were to do this, perhaps more systematic comparisons of various techniques could be carried out.

6 Conclusions

This position paper has proposed that concerted efforts be made to promote restricted domain question answering on the web. Web interfaces make no special assumptions about the user group (visually impaired, talking on a phone, etc.) and because of this, we must confront the fact that other interface techniques, such as forms, hyper-links, menus, key-word searches will compete directly with natural language interfaces. Still, it is argued here that there is a set of circumscribed, complex and practical domains that are best served by natural language question answer interface. Such applications constitute a break through point for natural language access to structured data/knowledge. If a group can deploy such a solution that always beats forms based and keyword based rivals, this will go a long way toward vitalizing work in natural language interfaces and ontologies for that matter.

References

- [1] I. Androutsopoulos and G.D. Ritchie. Database interfaces. In R. Dale, H. Moisl, and H. Somers, editors, *Handbook of Natural Language Processing*, pages 209–240. Marcel Dekker Inc., 2000.
- [2] F. Benamara. Cooperative question answering in restricted domain : the WEBCOOP experiment. In *ACL04 workshop on Question Answering in Restricted Domains*, 2004.
- [3] A. Copestake and K. Sparck Jones. Natural language interfaces to databases. *The Natural Language Review*, 5(4):225–249, 1990.
- [4] T. Gaasterland, P. Godfrey, and J. Minker. An overview of cooperative answering. *Intelligent Information Systems*, 1(2):127–157, 1992.
- [5] W. May. Information extraction and integration with FLORID: The MONDIAL case study. Technical Report 131, Universität Freiburg, Institut für Informatik, 1999.
- [6] M. Minock. A phrasal approach to natural language access over relational databases. In *Proc. of NLDB*, Alicante, Spain, 2005.
- [7] B. Thalheim and T. Kobienia. Generating DB queries for web NL requests using schema information and DB content. In *Proc. of NLDB*, pages 205–209, 2001.