Dynamic data sharing for facilitating communication during emergency responses

Fiona McNeill
Heriot-Watt University
f.mcneill@hw.ac.uk

Andriana Gkaniatsou
University of Edinburgh
a.e.gkaniatsou@sms.ed.ac.uk

Alan Bundy
University of Edinburgh
a.bundy@ed.ac.uk

ABSTRACT

This paper describes the CHAIn system, which is designed to facilitate data sharing between disparate organisations during emergency response situations by resolving mismatches in their data. It uses structured data matching to reformulate failed queries in cases where these failed because of incompatibilities between the query (derived from the source schema) and the schema of the queried datasource (the target schema). This reformulation is done by developing matches between the source schema and the target schema. These matches are then used to reformulate the query and retrieve responses relevant to those expected by the original query. Despite the growing interest in intelligent query answering, integration of data matching into query answering is novel, and allows users to successfully query datasources even if they do not know how the data in that source is organized, which is often the case during emergency responses. We describe the proof-of-concept system we have developed and an encouraging initial evaluation.

Keywords: Matching, query, data interpretation, communication

INTRODUCTION

Fast, effective data sharing is essential in many situations; one of the fields in which this is particularly pressing is emergency response. Emergency response situations are characterised by the coming together of large numbers of organisations, each of which is likely to have large amounts of data, much or some of which may be pertinent to the current emergency. Some of these organisations may be well known to each other, but others will be unknown and potentially untrusted. Datasources, even of known organisations, may be highly dynamic. Reports into previous emergency responses frequently cite poor communication and failure to effectively share information as a significant barrier to an effective response (HMSO, 2007). There is much interest in providing automated assistance to address these problems. There are many problems surrounding such automation; the one in which we are particularly interested is that of mismatch between datasources. Successful querying of a datasource depends on a good understanding of that datasource, thereby ensuring that the schema of the query correctly aligns with the target schema. If data querying is part of an automated process, such knowledge depends on being able to anticipate in advance exactly what datasources will be relevant and knowing accurately what the schema and data representations of that source will be at the time of querying. If such knowledge is possible, then effective communication is best addressed by pre-alignment of data sources, where ideally these datasources use the same fixed vocabulary for easy integration. However, in the general case such an approach is unrealistic. In a highly dynamic environment, such as emergency response, we will not usually know exactly whom we will need to interact with in advance, or exactly what the context of this will be. Since speed is of the essence in emergencies, and since humans cannot efficiently align large data sources, relying on human ability to identify these problems and update queries accordingly is not feasible; in addition, this depends

Proceedings of the 11th International ISCRAM Conference – University Park, Pennsylvania, USA, May 2014
S.R. Hiltz, M.S. Pfaff, L. Plotnick, and P.C. Shih, eds.

374

on the ability to access the schema of other people's data, which is not always permitted or practical. We, therefore, believe that automatic reformulation of queries based on matching between the source schema and the target schema is a necessary part of providing automated assistance to this communication process.

This paper introduces the CHAIn (Combining Heterogeneous Agencies' Information) system, which can be used by an owner of a datasource to formulate appropriate responses to incoming queries, even when these queries fail to match the datasource at the schema level and/or the data level[1]. We can assume that the target schema is available without loss of privacy, because CHAIn is a system that is installed by data owners to facilitate the controlled sending out of their own data, and details of their schema need not be passed to any other party. Potential responses are ranked, and the process may be fully automated or may be used as a basis for fast, efficient human interaction with large datasources. The main contribution of this paper is in the adaptation of this successful matching process (based on the structure-preserving semantic matching (SPSM) algorithm (Giunchiglia, McNeill, Yatskevich, Pane, Besana and Shvaiko, 2008)) to the problem of dynamic query matching in (potentially) large data. The use of automated data matching significantly advances the state-of-the-art in database querying (see Related Work section for details). Our hypothesis is that *we can use dynamic matching to provide meaningful results to queries that would otherwise fail*.

WORED EXAMPLE

Our techniques are designed primarily with an emergency response scenario in mind. Such situations are data rich, and are characterised by the need to share data quickly and effectively with collaborators who may be previously unknown, and may not be trusted. Formulating correct queries under these circumstances is extremely difficult, and there is, therefore, a pressing need for automated query reformulation.

Consider a flooding scenario. Imagine an environmental organisation which is trying to determine how full the rivers are to anticipate the course of the floods. They will have their own sensors monitoring this, but may want to get additional information from other organisations which also have sensors - for example, other environmental agencies from upstream regions and private companies who monitor water levels for their own commercial purposes but may be agreeable to sharing limited data during an emergency. Perhaps in their own datasources they frame readings from their sensors as follows: *measurement(Reporter_ID,Node,Level,Date)*.

They will then send this query to other organisations that they believe may have relevant information. If the organisation were automatically developing queries to populate this table, the query that would be formulated, in SQL-like format, would be:

> **SELECT** *reporter_ID, node, level, date*
>
> **FROM** *measurement*

If there is no mismatch at either the data or the schema level, the query will succeed and appropriate responses will be returned without the need to invoke the CHAIn system. However, the queried organisation may organise their data differently on many levels. For instance *reading(Node,Date,Water_level)*. There are various mismatches here, in the words used, in the organisation of the arguments and in the number of arguments, so returning an appropriate response will require alignment with the datasource schema.

QUERY RESPONSE PROCESS

The goal of the CHAIn system is to return an appropriate response to a query on a datasource, despite the fact that the query is not correctly formulated for that datasource. Whenever a query fails, CHAIn will first investigate whether the schema used by the query is correct with reference to the schema of the queried datasource. If it is not, matching is used to determine whether there is anything in the schema of the datasource that approximates to the schema of the query, and, if so, reformulates the query accordingly. If a query fails when there is no mismatch in the schema - either because the original schema of the query was correct according to the target schema, or because the query has been reformulated to reflect the target schema - the CHAIn system will then consider potential mismatches at the data level. Ultimately, the query will be responded to with a set of responses using the data that CHAIn considers the most appropriate response to the original query, or it will fail if there is nothing in the target datasource that is considered sufficiently relevant.

---

[1] A mismatch at the *schema level* is one where the structure of the data differs: for example, the columns of a database are in a different order. A mismatch at the *data level* is one where the schemas match at this particular point, but the specific data differs: for example, dates are given as DDMMYY rather than MMDDYY.

Currently, orthogonal issues, such as provenance of the data, are disregarded, though a full solution to this problem would need to incorporate such information. In this section, we describe the various aspects of the process, with reference (where appropriate) to the worked example.

The role of Humans: CHAIn can operate completely automatically because the matching process ranks potential matches. It is, therefore, possible to produce a single best-ranked response to the original query. Setting CHAIn to be fully automated is the best approach where responses are needed very quickly and quality of matching is not important. However, we primarily view CHAIn as a method for allowing human users to interact with large datasources quickly and effectively, providing them with the tools to make informed decisions. It is therefore the general expectation that when the matching process is completed, a set of results passing a given threshold for matching quality, or constrained by the number of required results, is presented to a user that owns, or belongs to an organisation that owns, the target datasource. These results are presented with clear information about where approximations were made, where parts of the query were left unanswered, and so on. This allows for the input of local knowledge into context-free matching, and helps datasources where localised terminology is used to become more universal comprehensible. This user (from the target organisation) will be able to decide to send back one or more (or none) of the responses suggested by CHAIn to the querier. The current implementation simply presents a ranked list of possible responses to a user, with information about the matching. However, a more usable system would need to focus on HCI considerations to allow the user from both source and target organisations to navigate possible responses easily and effectively. On the querying side, the querying organisation could automatically accept this response if desired, or it could also be filtered by a human user within that organisation to determine whether or not to accept the response, or which of a list of potential responses is most appropriate. This will depend on what the information is needed for; for example, a response missing one particular attribute of the query may be useless, whereas one missing another attribute may be of some value.

Lifecycle of the Process

1.   A query to a particular datasource is received by the target organisation (or individual).
2.   If the query fails, it is first determined whether this is due to mismatch at the schema level. If not, go to step 6. If so, naive matching is done on the target schema to narrow down the search space to likely matches.
3.   Potential matches are sent pair-wise with the query to the SPSM matcher.
4.   The SPSM matcher returns matches together with a score, which is used to rank and reject matches.
5.   The query is reformulated according to the matches and resent to the target datasource.
6.   If the (now correctly formulated) query fails, we look for mismatches at the data level.
7.   Potential responses to the query passing a given threshold are presented, with appropriate annotation describing the match, to a user who is knowledgable about the target datasource. This user then choses any number of matches to be returned to the querier. (Fully automated ranking can be used if desired).
8.   The querier receives the response(s) to their original query, together with information about the matches that were used to produce the response, and uses these as they feel appropriate.

Format of the Datasource: The schema-matching part of the process matches first-order terms: that is, terms of the form $predicate(Arg_1, Arg_2, ..., Arg_n)$, where each Arg_i may itself be a function. This process is therefore applicable to data formats that can be translated into this format. Since this is a very general format, this subsumes a vast number of common representations. Currently, CHAIn can be used for SPARQL queries to RDF datasources and for SQL queries to RDB datasources, but it could easily be extended to more formats.

Narrowing Down the Search Space: The matching part of the process is done by the SPSM algorithm, which does pairwise matching of structured terms. This is an expensive process, and it is not feasible to do it between a single query and every datum in the possible (potentially large) target datasources. We therefore do a fairly naive keyword matching on the source datasource name to the target datasource to narrow down to a subset of likely candidates. There is, of course, a tension in determining how wide this net should be cast: we do not wish to exclude potentially good matches; nor do we wish to do structured data matching on a very large dataset. Initial results (see Evaluation section) indicate our approach is workable, but extensive simulation is necessary to determine more precisely where this sweet spot lies.

Returning Query Responses: Helping users, who may have a good understanding of the data used by their organisation but are unlikely to be experts in representation and matching, to appropriately and efficiently evaluate a set of potential responses to a query, weighing them against each other with reference to the particular ways in which each individual response fails to be an exact response to the original query, is a complex and demanding task, requiring high-quality techniques in human-computer interaction. Our current approach is to present the relevant information to the user in a simple manner. Building a more sophisticated, user-oriented solution to this is an important aspect of future work.

EVALUATION

A robust evaluation of this approach requires these queries to be generated during the kind of automated process we envisage this work be used in, and the effectiveness of these matches determined through large-scale simulation, to determine what the actual impact of using these matches are. There is a growing interest in automated facilitation of communication during emergency events (e.g., Fitzgerald, Bryans and Payne, 2012). Integrating our work with such a system is an important past of future work, but has not been attempted yet. We are therefore considering individual queries and responses, rather than queries and responses generated organically during automated interaction. We have carried out the evaluation as follows: *(i)* We have sourced appropriate data from a collaborator, SEPA (Scottish Environmental Protection Agency). *(ii)* We have sourced online different datasources which discuss similar things, such that we might expect a query from such a source datasource to find an appropriate response in our target SEPA data. It is not immediately apparent from the behavior of the system what constitutes success or failure. The system returning results constitutes failure if these responses are not appropriate; similarly, the failure of the system to return any results constitutes success if there was no relevant data in the target datasource. We therefore needed to examine results and datasources carefully to judge the performance of the system.

For each evaluation test we selected a dataset and formed a query based on the schema of that dataset, which was sent to the other datasets. We have carried out 26 evaluation tests. Of these, 14 returned responses to the queries. All these responses were judged appropriate: answers were relevant to the query. 12 did not return a result. In 4 of these cases, this was because the narrow-down process returned no results, and in all cases this was judged an appropriate response because there was nothing in the target datasources which approximated the query predicate name (not returning irrelevant results is almost as important as returning relevant results). 8 of these failed at the SPSM matching after receiving input from the narrow-down process. Of these, 3 were appropriate failures, because the arguments of the narrowed-down predicates were not similar enough to the query to justify return data; 5 failed because of SPSM implementational issues which we believe can be fixed without difficulty. There is some difficulty establishing exactly what an *appropriate response* is. Our approach has been to consider the kind of data that would have been generated by the query on the source datasource compared to the kind of data returned by the target datasource, to determine whether they are reasonably similar. We have also presented many of our results to SEPA, and they agreed with our judgements in these cases.

RELATED WORK

The work in this paper is concerned with the extension and adaptation of existing matching techniques to a new domain: that of dynamic query reformulation, specifically in an emergency response scenario. SPSM differs from other matchers in that it is applicable to *structured* terms, built on standard tree-edit-distance techniques (e.g., in (Shasha and Zhang, 1997) and using semantic matching to determine what impact on the overall *meaning* of a tree alteration or removal of nodes will have. Previously, it has been used to perform service integration and peer selection in a peer-to-peer network (Osman, Sierra, McNeill, Pane and Debenham, 2013). The novelty of our work is in developing a system that can use this structured matching technique to address the problem of returning useful results to a query formulated according to a source datasource and sent to a target datasource.

There is a considerable body of work on the problem of intelligent query answering, which aims to do more than simple look-up. However, this work tends to focus on how to return maximum, or maximally useful information and assumes that the format of the datasource is known: the need to use matching techniques is not considered. (Han, Huang, Cercone and Fu, 1995) describes various approaches to this issue, where intelligent query answering is used to provide more (relevant) detail than the query originally asked for, to provide summaries of data, and to reformulate queries into sub-queries but does not consider the issue of data matching. (Han et al, 1995) also raises interesting questions about how to appropriately interact with the user and how to ensure that sensitive data is released in an appropriate way to appropriate people, which is certainly relevant to a more developed version of our system. (Chu and Chen, 1992) proposes the DB-Pattern-KB framework and the type abstraction hierarchy to improve query answering. Work has also been done on combining data from different sources to provide intelligent responses to queries (see, for example (Hagedorn and Sattler, 2013)). (Buscaldi, Rosso and Arnal, 2005; Hollink, Guus and Wielinga, 2007) investigate how semantic relations based on Wordnet can improve query expansion. However, the problem of data mismatch between the query and the target datasource is not considered. In (Calvanese, de Giacomo, Lembo, Lenzerini, Poggi, Rodriguez-Muro, Rosati, Ruzzi and Savo, 2011) the Ontology-Based Data Access framework proposes the use of an ontology to create views of the datasources, so a user can have access to the data without knowing the data organization. In our approach, instead of creating views of the datasources we align the query on the fly to the most appropriate target datasources to retrieve answers, a much more dynamic solution.

Proceedings of the 11th International ISCRAM Conference – University Park, Pennsylvania, USA, May 2014
S.R. Hiltz, M.S. Pfaff, L. Plotnick, and P.C. Shih, eds.

377

There is some work on query answering over distributed, mismatched data sources; however, this makes more assumptions than our work. For example (Lee, Park, Park, Chung and Min, 2010) describe a rewriting system that is in some ways similar to ours, but assume that the user is provided with all available schemata and generates their queries with this in mind. (Haase and Wang, 2007) address the problem of distributed query answering over heterogeneous ontologies using semantic mappings, but assume that all such mappings are predefined and not generated on the fly (unrealistic in the emergency response domain). There is also a significant body of work on schema mapping (see (Kolaitis, 2005) for a summary), but this generally assumes pair-wise mappings between full schema, and is thus addressing a different problem to us.

CONCLUSIONS

The problem of successfully querying unknown datasources is an important one in terms of facilitating data sharing during emergency response events. The general assumption in database querying is that the querier knows the datasource and can use the correct schema and terminology when developing their query, but this is a massively limiting assumption as it restricts querying to situations where: 1) one knows in advance what datasources one will wish to query; 2) one has the time to devote to examining the datasource and correctly formulating any queries one may have; 3) the datasource is open and one can examine its schema. We believe this discounts a large number of situations in which one or more of these assumptions are not valid, particularly during emergency response situations, where collaborations are formed dynamically and quickly, where information needs to be passed rapidly, and where the size, number and potential sensitivity of available datasources ensure that it is impossible to adequately understand all relevant data.

The CHAIn system described in this paper uses an established structural matching algorithm - SPSM - to address the problem of mismatched database queries. We are centrally interested in how techniques developed for ontology matching can be used to address other situations in which data mismatch of some sort is a problem. CHAIn is currently a proof-of-concept system and requires significant further development and extensive evaluation based on simulation before we expect it to be usable in the field. However, the implementation and evaluation we have produced so far have demonstrated that this approach to queries which are mismatched with the datasource they are querying is valid and potentially useful.

REFERENCES

1.   Buscaldi, D., Rosso, P. and Arnal, E.S. (2005) A word-net based query expansion method for geographical information retrieval, In *Working notes for the CLEF workshop*.

2.   Calvanese, D., de Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M. and Savo, D. F. (2011), The Mastro system for ontology-based data access. *Semantic Web Journal,* 2(1):43-53.

3.   Chu, W.W. and Chen, Q. (1992) Neighbourhood and associative query answering, *Journal of Intelligent Information Systems,*1(3-4):355-382.

4.   Fitzgerald, J., Bryans, J. and Payne, R. (2012) A Formal Model-based Approach to Engineering Systems-of-Systems, *Collaborative Networks in the Internet of Services*.

5.   Giunchiglia, F., McNeill, F., Yatskevich, M., Pane, J., Besana, P. and Shvaiko, P (2008) Approximate Structure-Preserving Semantic Matching, In *Proceedings of the 7th International Conference on Ontologies, Databases and Applications of Semantics (ODBASE)*, 1217-1234.

6.   Haase, P. and Wang, Y. (2007) A Decentralised Infrastructure for Query Answering over Distributed Ontologies. In *Proceedings of the 2007 ACM symposium on Applied Computing*

7.   Hagedorn, S. and Sattler, K.-U. (2013) Discovery querying in linked open data, In *EDBT/ICDT Workshops*, 38-44.

8.   Han, J., Huang, Y., Cercone, N. and Fu, Y (1995) Intelligent query answering by knowledge discovery techniques. *IEEE Transactions on Knowledge and Data Engineering,* 8:373 390.

9.   Hollink, L., Guus, S. and Wielinga, B. (2007) Patterns of semantic relations to improve image content search, *Web Semantics: Science, Services and Agents on the World Wide Web,* 5.3:195-203.

10. HMSO (2007) The Pitt review: lessons learned from 2007 floods. London.

11. Kolaitis, P. (2005) Schema Mappings, Data Exchange and Metadata Management. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems*, 61-75.

12. Lee, J., Park, J-H,. Park, J-M., Chung, C-W. and Min, J-K. (2010). An intelligent query processing for distributed ontologies. *Journal of Systems and Software.* 83 (1) 85-95.

13. Osman, N., Sierra, C., McNeill, F., Pane, J. and Debenham, J. (2013) Trust and matching algorithms for selecting suitable agents, *ACM Transactions on Intelligent Systems and Technology (TIST)*.

14. Shasha, D. and Zhang, K. (1997) Approximate Tree Pattern Matching. In *Pattern Matching Algorithms, 341-371.*

Proceedings of the 11th International ISCRAM Conference – University Park, Pennsylvania, USA, May 2014
S.R. Hiltz, M.S. Pfaff, L. Plotnick, and P.C. Shih, eds.

378