Explaining Results from Information Retrieval and Integration

Honglei Zeng, Deborah L. McGuinness, Paulo Pinheiro da Silva, and Richard Fikes

Computer Science Department, Knowledge System, AI Laboratory, Stanford University
email: {hlzeng, dlm, pp, fikes}@ksl.stanford.edu

Abstract

Information retrieval and integration systems typically must handle incomplete and inconsistent data. Current approaches attempt to reconcile discrepant information by leveraging data quality, user preferences, or source provenance information. Such approaches may overlook the fact that information is interpreted relative to its context. Therefore, discrepancies may be explained and thereby resolved if contexts are taking into account. In this paper, we describe an information integrator that is capable of explaining its results. We focus on using knowledge of an assumption context learned through decision tree-based classification to inform the explanations. We further discuss some benefits and difficulties of applying assumption context in information retrieval. Finally, we indicate how to use Inference Web to explain discrepancies resulting from information retrieval and integration applications.

1. Introduction

Information retrieval (*IR*) techniques can be used to retrieve documents relevant to user queries. In Web search settings, IR systems, such as Google, process a search query and return a list of web pages ranked according to relevancy measures defined by the IR systems. Information integration (*IIT*) techniques can be used to integrate data from multiple heterogeneous data sources. For example, TAP [1] is an integration framework that facilitates knowledge aggregation and semantic search. IR and IIT are vastly different in the ways they provide results to users: IR typically returns web pages or text files. Users must read the documents to find the desired information. IIT attempts to extract information from sources and produce unified and structured data that may be presented as answers (instead of requiring user review and analysis). IR and IIT are also highly related, not only because they face many similar challenges such as the co-reference problem but also because they share a number of techniques, for instance, document clustering.

Dealing with incomplete and inconsistent data is one of the most critical issues in information retrieval and integration (IRI) [2]. Current approaches attempt to reconcile discrepant information by leveraging data quality, user preferences, or source provenance information [3] [4] [5]. However, such efforts often fail to produce consistent or complete knowledge and may leave users mystified concerning missing or conflicting data.

Information is interpreted relative to its context. We believe one major source of discrepancies come from misinterpretation of information. Although there is no consensus on the definition of context, we informally refer it to be a collection of provenance, situations, assumptions, biases, domain, prior events and other information relevant to a source. Discrepancies may be understandable and resolvable if source contexts are accessible.

In this paper, we propose an approach called assumption context knowledge (ACK) to describe assumptions used in generating or updating an information source. We focus on ACK for two main reasons: first, it is capable of supporting explanations for several types of data incompleteness and inconsistency issues and therefore it is useful in IRI. Secondly, though learning general context is a complex task, we are able to build a decision tree based classifier to partially learn ACK.

In Section 2.1 and 2.2, we present the learning method and the applications of ACK in IIT. In Section 2.3, we show the important role of ACK in IR and the difficulties in learning ACK in IR. We discuss Inference Web, a general framework for managing explanations and proofs in Section 2.4 and conclude with a discussion of future work in Section 3.

2. Assumption Knowledge

In IIT systems, source wrappers are typically designed to convert unstructured or semi-structured source data to structured data that an integrator can process. In other cases, source wrappers may also translate data from one format to another. Without loss of generality, we therefore assume that a relational model is used in IIT, in which data are described by n-tuples of attribute values: (v_1, v_2, …, v_n), where each v_i is a value of a certain attribute T_i. We also assume that each information source can be characterized as a collection of tuples.

Table 1 contains several tuples from the Internet Movie Database (IMDB) and the Yahoo Movies website. For example, data tuple t4: (“t4,” “Midwives,” “TV,” 2001, 92, IMDB) denotes the movie named “Midwives”, whose type is TV, whose release year is 2001 and whose running time is 92 minutes. *Class*, the classification label, decides which website the movie is on. The values of *Class* could be “IMDB” if the movie only appears on IMDB, or “Yahoo” if it only appears on Yahoo Movies, or “IMDBYahoo” if on both websites. A question mark in

Table 1 means that the value of corresponding attribute is unavailable. Only attributes and values relevant to our discussion are presented in Table 1.

Table 1. A partial list of movies from IMDB and Yahoo

Information source A (IMDB.com)					
ID	*Name*	*Type*	*Year*	*Run time*	*Class*
t1	The Black Dahlia	Film	2005	?	IMDB
t2	Little Fish	Film	2005	?	IMDB
t3	Band of Brothers	TV	2001	600	IMDB
t4	Midwives	TV	2001	92	IMDB
t5	A Beautiful Mind	Film	2001	135	IMDBYahoo
t6	Shrek	Film	2001	90	IMDBYahoo
Information source B (movies.yahoo.com)					
ID	*Name*	*Type*	*Year*	*Run time*	*Class*
t7	A Beautiful Mind	Film	2001	135	IMDBYahoo
t8	Shrek	Film	2001	90	IMDBYahoo
t9	Solaris/ Cast Away	DVD	2000	?	Yahoo
t10	The Terminal/ Catch me if you can	DVD	2005	?	Yahoo

2.1 Explaining incomplete data in IIT

One key observation in Table 1 is that some data tuples on IMDB (i.e. t1, t2, t3, and t4) are not on Yahoo and vice versa (i.e. t9 and t10), even though both websites are credible sources for movie information. A simple approach would be to list any movie that appears on either IMDB or Yahoo. A more sophisticated approach may leverage data quality, user preferences, or source provenance information when combining partial results. For example, this approach may reject tuples from less credible sources. We believe an information integrator with explanation capacities can provide much more accurate and trustworthy data than previous approaches. For example, a conservative user may accept information only when it is confirmed by *all* sources. However, users of our system, may be willing to accept information that is only partially confirmed but information from is accompanied by satisfactory explanations.

To address the need for explanations, we introduced the concept of ACK in [6]. ACK refers to a set of implicit rules about assumptions on which a source is based. For example, we know the assumption that the Yahoo Movies website does not list TV mini series. This can explain why "Band of Brothers" is missing from Yahoo. In another example, "The Black Dahlia" is missing from Yahoo because of the assumption that the list of upcoming movies on Yahoo is incomplete.

We can a build a decision tree from Table 1 using the c4.5 decision tree generator [7]. Once the tree has been constructed, it is a simple matter to convert it into an equivalent set of rules. This set of rules is a partial set of ACK that we are seeking. The rules are represented in terms of available attributes. As simple as it is, the tree in Figure 1 is an interesting characterization of contexts of the tuples from IMDB and Yahoo in Table 1. For example, tuples t3 & t4 are missing from Yahoo because of the causal rule from the decision tree: "*(Type x TV)* → *(Missing x Yahoo)*".

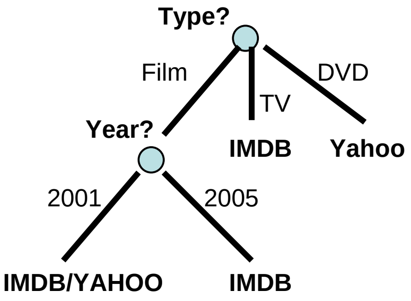

Figure 1. Decision tree generated from Table 1

Decision tree classification is a widely used data mining technique. For example, a loan approval system may classify millions of customer records from multiple financial sources. The records may fall into two categories: acceptance or rejection, based on the attributes of the records, such as level of income, outstanding debts, and credit history. The novelty of our approach is that the sources themselves can be the classification categories; therefore, our loan system may learn the contextual differences between multiple financial sources in addition to making a loan decision, while the contexts learned may improve the quality of loan decisions.

Choosing a good set of attributes is critical in building a decision tree because bad attributes cannot capture assumption contexts of sources. A related discussion on choosing attributes and other ACK issues (such as detecting incomplete data and the co-reference problem) can also be found in [6]. The integrator we built has 87% accuracy in explaining tuples missing from IMDB and Yahoo. Finally, the integrator can also explain missing values. For example, the runtime of "Solaris/ Cast Away" is missing because it is a DVD combo.

2.2 Explaining inconsistent data in IIT

In Table 1, the running time of "Shrek" is different in IMDB and Yahoo: one is 90 minutes and the other is 100 minutes. This inconsistency seems rather odd because we would expect the runtime of a movie to be a unique value. However, this is not entirely true. The length of a movie could vary due to film cuts for different countries, or new scenes inserted for DVD release. Many other explanations are also possible. For example, the frame rate of PAL (European DVD format) is 4% faster than the US format; thus PAL movies are typically 4% shorter in length. A decision tree-based approach may generate a number of interesting explanations if proper attributes are chosen. However, choosing good attributes may be challenging.

The principle of *learning from discrepancies* is vital in dealing with inconsistent data. For example, with a sample size of 100, 28% of Yahoo movie runtime values are rounded to tens (e.g. 140 minutes), while only 12% of IMDB runtime values do so. This result strongly suggests that many runtime values on Yahoo are imprecise; however such explanations are difficult to generate without knowing the differences between IMDB and Yahoo. If we only know the Yahoo data, a possible assumption is that the actual runtime happens to be a rounded number.

2.3 Assumption context knowledge in IR

The decision tree approach, though successful in IIT, is difficult to apply directly in IR, because the data that IR returns are not in a relational model (in fact, the data are usually unstructured). In addition, unlike IIT which typically works with predefined sources, IR retrieves data from millions of sources. Building ACK for every source is very challenging and expensive yet it still may be useful for IR and may warrant the effort. .

First, users often seek explanations of divergent answers. For example, the query: “kill bill volume 1 runtime” yields 111 using IMDB while dvdtown.com returns 107 minutes. The explanation may be useful when users are interested in finding deleted scenes from that movie. In general, we believe explanation can help users find information more efficiently and more accurately.

Second, ACK may be used to expand user queries. This is a widely used technique and has been shown to improve the precision of IR. User query expansion typically works by adding additional relevant words to the user query thereby narrowing the search space. For example, a query “jaguar” can be augmented with “car” or “animal”, depending on the sense of the word the user is looking for. Query expansion can also work to improve recall by adding more words to the query that are more specific. For example, a query of “car” may be enhanced with the words “Porsche 911” and then documents containing only the model, but not the word “car”, could be returned.

Finally, explainable information retrieval itself may be important. Users may want to understand IR ranking criteria so they can adjust certain ranking parameters. For example, the need for time-dependent information that is on the Web often lasts significantly longer than the availability of that information on the Web. If a user wants to find information about an old laptop model, he may want to weigh older documents more heavily.

Although this paper focuses on assumption context, context in general has significant value in IR. Recently much work has been done in clustering and classification of retrieved web pages, which in some sense, is building contexts about sources. Also, as Semantic Web technology grows, automated information extraction and classification is becoming feasible thus generating more possibilities for the use of ACK in the Semantic Web.

2.4 Inference Web

Inference Web (IW) [8] is a general framework that enables applications to generate, check, present, browse, summarize, share and distribute explanations. IW contains data for representing proofs, explanations and metadata about proofs and explanations. IW proofs and explanations are encoded in the Proof Markup Language (PML) [9], which is built using proof elements referring to provenance elements. ACK rules and explanations can be encoded in PML and IW can be used to build, maintain, check, abstract and present ACK-based proofs and their explanations for IRI applications.

3. Conclusion

We have discussed the importance of ACK and explanation generation in IRI. We presented a decision tree based approach for learning ACK from incomplete and inconsistent data. We presented Inference Web as a general framework for managing explanations. We intend to make IRI explanation-aware thus improving the accuracy, efficiency, and user-friendliness of IRI.

We are investigating ways to extend our work: learning and reasoning with ACK with emerging Semantic Web technology. We are also exploring other machine learning techniques for building ACK in IRI and extracting other types of context knowledge in IRI.

Acknowledgement

This work is partially supported by ARDA’s NIMD program and DARPA’s IPTO Office.

References

[1] Guha, R. V. and McCool, R. *TAP: a Semantic Web platform*, Computer Networks 42(5): 557-577 (2003).

[2] Information Integration Research Summary, NSF Information and Data Management workshop (IDM 2003)

[3] Motro, A., Anokhin, P. and Acar, A. *Utility-based Resolution of Data Inconsistencies*, IQIS 2004.

[4] Greco, G. and Lembo, D. *Data Integration with Preferences Among Sources*, ER 2004: 231-244.

[5] Buneman, P., Khanna, S., and Tan, W.C: *Why and Where: Characterization of Data Provenance*. ICDT 2001.

[6] Zeng, H. and Fikes, R. *Explaining data incompleteness in knowledge aggregation*. Under review (2005).

[7] Quinlan, J.R. *C4.5: Programs for Machine Learning* (1993).

[8] McGuinness, D.L. and Paulo Pinheiro da Silva. Explaining Answers from the Semantic Web: The Inference Web Approach. Journal of Web Semantics. Vol.1 No.4., pp. 397-413 (2004)

[9] Pinheiro da Silva, P., McGuinness, D.L. and Fikes, R.. *A Proof Markup Language for Semantic Web Services*. Information Systems (to appear).