Asian Journal of Information Technology 15 (12): 1970-1974, 2016
ISSN: 1682-3915

The Efficient Technique to Protect and Maintain Content Based Queries of Mobile Networks

D. Teja and G. Rama Krishna
Department of Computer Science and Engineering, KL University, Andhra Pradesh, India

Abstract: Now-a-days several organizations generate and share descriptive and textual information of their product, services and actions. Such kind of collections of textual information contain vital quantity of structured data that remains saved in the unstructured text. While data extraction algorithms facilitate the extraction of structured relations in a very costly and in accurate manner especially when operating on top of text that doesn't contain any instances of the targeted structured data. There are several alternative approaches that facilitate the generation of the structured data by distinguishing documents that are possible to contain information of interest and this data is going to be subs equently helpful for querying the database that depends on the thought that humans are more possible to add the required data throughout creation time. This could be done like prompting by the interface or that it should made easier for humans (and/or algorithms) to identify the data when such data really exists within the document, rather than naively prompting users to fill in forms with data that's not out there within the document.

Key words: Annotation, CADS, information extraction, data quality, attribute value

INTRODUCTION

There are several application domains wherever users create and share information; for instance, news blogs, scientific networks, social networking teams or disaster management networks. Current data sharing tools, like content management software (e.g., Microsoft share point), allow users to share documents and annotate (tag) them in ad-hoc way. Similarly, google base permits users to define attributes for their objects or select from predefined templates. This annotation process will facilitate subsequent information discovery. Several annotation systems permit only "untyped" keyword annotation: for instance, a user may annotate a weather report using a tag such a "Storm Category 3". Annotation methods that use attribute-value pairs are generally additional expressive as they can contain a lot of data than utyped approaches. In such settings, the above data will be entered as (Storm Category, 3). A recent line of work towards using additional expressive queries that leverage such annotations is the "pay-as-you-go" querying strategy in dataspaces (Jeffery *et al*., 2008) in dataspaces, users provide information integration hints at query time. The assumption in such systems is that the information sources already contain structured data and drawback is to match the query attributes with the source attributes. Many systems, though, don't even have the essential "attribute-value" annotation that will create a "pay-as-you-go" querying feasible. Annotations that use "attribute-value" pairs users to be additional principled in their annotation efforts. Users should know the underlying schema and filed types to use; they should also recognize once to use each of these fields. With schemas that always have tens or even hundreds of available fields to fill this task becomes complicated and cumbersome. This leads to information entry users ignoring such annotation capabilities. Even if the system permits users to arbitrarily annotate the information with such attribute-value pairs, the users are usually unwilling to perform this task: the task not only needs considerable effort but it also has unclear usefulness for subsequent searches in the future: who is going to use an arbitrary undefined during a common schema when there are tens of potential fields that can be used which of these fields are going to be helpful for searching the database within the future? Such difficulties results in very basic annotations if any all that are usually restricted to easy keywords. Such simple annotations create the analysis and querying of the data cumbersome. Users are typically restricted to plain keyword searches or have access to very basic annotation fields such as "creation data" and "owner of document". In this study, we propose CADS

Corresponding Author: D. Teja, Department of Computer Science and Engineering, KL University, Axdhra Pradesh, India

ZCZC MIATCPAT2 ALL
TTAA00 KNHC DDHHMM
BULLETIN
HURRICANE GUSTAV INTERMEDIATE ADVISORY NUMBER 31A
NWS TPC/NATIONAL HURRICANE CENTER MIAMI FL AL072008
600 AM CDT MON SEP 01 2008
(a)
EYE OF GUSTAV NEARING THE LOUISIANA COAST...HURRICANE FORCE WINDS OVER PORTIONS OF SOUTHEASTERN LOUISIANA... A HURRICANE WARNING REMAINS IN EFFECT FROM JUST EAST OF HIGH ISLAND TEXAS EASTWARD TO THE MISSISSIPPI-ALABAMA BORDER...INCLUDING THE CITY OF NEW ORLEANS AND LAKE PONTCHARTRAIN. PREPARATIONS TO PROTECT LIFE AND PROPERTY SHOULD HAVE BEEN COMPLETED. A TROPICAL STORM WARNING REMAINS IN EFFECT FROM EAST OF THE MISSISSIPPI-ALABAMA BORDER TO THE OCHLOCKONEE RIVER. GUSTAV IS MOVING TOWARD THE NORTHWEST NEAR 16 MPH...26 KM/HR... ON THE FORECAST TRACK...THE CENTER WILL CROSS THE LOUISIANA COAST BY MIDDAY TODAY. MAXIMUM SUSTAINED WINDS ARE NEAR 115 MPH...185 M/HR...WITH HIGHER GUSTS. GUSTAV IS A CATEGORY THREE HURRICANE ON THE SAFFIR-SIMPSON SCALE.
(b)
Storm Name = 'Gustav'
Storm Category = 3
Warnings = 'tropical storm'
(c)
Q1: Storm Name = 'Gustav' AND Warnings = 'flood'
Q2: Storm Name = 'Gustav' AND Storm Category > 2
Q3: Document Type = 'advisory' AND Location = 'Louisiana' AND Date FROM 08/31/2008 TO 09/30/2008

Fig. 1: Sample document and annotations: a) example of an unstructured document; b) desirable annotations for the document above and c) queries that can benefit from the annotations

Fig. 2: Adaptive insertion form

(collaborative reconciling data sharing platform) that is an "annotate-as-you-to-create" infrastructure that facilitates fielded information annotation. A key contribution of our system is the direct use of the query workload to direct the annotation process, additionally to examining the content of the document. In alternative words we are trying to rank the annotation of documents towards generating attribute values for attributes that are often used by querying users.

If we tend to use automatic data extraction algorithms to extract targeted relations from the document (e.g., addresses of exhausted buildings) it's necessary to process only documents that really contain such information: when we process documents that don't contain the targeted information and we use automatic data extraction algorithm to extract such fields we often face a significant number of false positives which may lead to significant quality issues within the information (Jain and Ipeirotis, 2009). Similarly, if the documents are processed by humans (i.e., wherever there's slow probability of false positives), asking humans to inspect documents wherever the relevant data is gift is expensive and harmful. For example, if only 1% of the documents contains info regarding the address of evacuated buildings it's planning to be unnecessarily costly to raise humans to examine all documents to spot such information: it's far better to focus on and method solely promising documents with high chance of containing relevant data. Going back to our disaster management motivating situation, after the user submits the cyclone consultative document Fig. 1a, CADS analyzes the content and finds that the subsequent attributes types are relevant and present within the document: "Storm Name", "Storm Category" and "Warnings".

Figure 2 presents the adaptive insertion form for that document. The system adds the recommended attributes to a group of default attributes like: "Document Type",

"Date" and "Location" that are the essential data that the user always provides as defined by a domain expert. This reconciling generation of data forms permits for abundant more efficient data generation. As we are aiming to see later our CADS system prioritizes and suggests initial attribute varieties that are used frequently by users that issue queries against the database. In short the contributions of this study are: we present an adaptive technique for automatically generating information inputs forms for annotation unstructured textual documents such the utilization of the inserted information is maximized, given the user data needs.

LITERATURE REVIEW

Collaborative annotation: There are many systems that favor the collaborative annotation of objects and use previous annotations or tags to annotate new objects. Their visibility when the tagging method compared with the other approaches exactness may be a secondary goal as we expect that the annotator will improve the annotations on the method. On the other hand, the discovered tags assist on the tasks of retrieval rather than simply bookmarking.

Dataspaces and pay-as-you-go integration: The integration model of CADS is comparable to that of dataspaces (Franklin *et al*., 2005) where a loosely integration model is proposed for heterogeneous sources. The essential distinction is that dataspaces integrate existing annotations for information sources, so as to answer queries. This research suggests the suitable annotation during insertion time and also takes into thought the query work to identify the most promising attributes to add. Another related information model is that of google base where users will specify their own attribute/value pairs in addition to the ones proposed by the system. However, the proposed attributes in Google Base are hard-coded for every item category (e.g., land property). In CADS, the goal is to learn what attributes to suggest. Pay-as-you-go integration techniques like PayGo (Jeffery *et al*., 2008) are useful to suggest candidate matchings at query time. However, no previous work considers this drawback at insertion time as in CADS. The work on peer information management systems (Halevy *et al*., 2003) is a precursor of the above projects.

Content management Softwares: Microsoft Sharepount Sharepoint and SAP NetWeaver Management enable users to share documents, annotate them and perform easy keyword queries. Hard-coded attributes is added to specified insertion forms. CADS improves these platforms by learning the user data demand and adjusting the insertion forms consequently.

Information extraction: Data extraction is related to this effort, primarily within the context important suggestion for the computed attributes. Cafarella *et al*. (2009) for a summary of IE) we are able to broadly separate the area into 2 main efforts: closed that is and open that is. Closed that is needs the user to define the schema then the system populates the tables with relations extracted from the text. This research on attribute suggestion naturally enhances closed that is as we determine what attributes are likely to appear within a document.

Once we've the data, we can then use the IE system to extract the values for the attributes. Open IE (Etzioni *et al*., 2008) is nearer to the requirements of CADS, particulalr, open IE generates RDF-like triplets, e.g. (Gustav is calss, 3) with no input from the user. Open that is leads to a very large number of triplets which implies that even after the successful extraction of the attribute values we still have to deal with the problem of schema explosion that prevents the successful extraction of the attributes values we still have to deal with the problem of schema explosion that prevents the successful execution of structured queries that need knowledge of the attribute names and values that seem within a document in essence we may use open that is and then pay-as-you-go solutions for characteristic equivalency relations across attribute names: but it is much better to affect the problem early-on, throughout document generation, rather than trying to mend issues that would be prevented with correct design. The CIMPLE project (Doan *et al*., 2006; Chu *et al*., 2009) uses IE techniques to form and manage data-rich on-line communities, like the DBLife community. In distinction to CIMPLE where information is extracted from existing sources and a domain expert must produce a domain schema, CADS is a data sharing setting wherever users expressly insert the data sharing setting wherever users expressly insert the data and the schema automatically evolves with time. Nevertheless, the mass collaborative techniques of CIMPLE will facilitate in making adaptive insertion forms in CADS.

Schema evolution: The adaptive annotation in CADS may be viewed as semi-automatic schema evolution. Previous work on schema evolution (Banerjee *et al*., 1987) didn't address the problem of what attribute to feature to the schema but how to support querying and alternative information operations once the schema changes.

Query forms: Existing work on query forms is leveraged in making the CADS adaptive query forms. Jayapandian and Jagadish (2008a) propose an algorithm to extract a query type that represents most of the queries in the database using the "querability" of the columns while by Jayapandian and Jagadish (2008b) they extend their research discussing forms customization. Nandi and Jagadish (2007) use the schema information to auto-complete attribute or value names in query forms. In (Chu *et al*., 2009) keyword queries are used to choose the most appropriate query forms. Our work is thought of a dual approach: rather than generating query forms using the database contents, we produce the schema and contents of the database by considering the content of the query workload (and the contents of the documents, of course). The research in usher (Chen *et al*., 2011) is also related: in usher the system automatically decides which queries during a survey are the foremost necessary to ask, given past expertise with the completion of past surveys. In a sense, usher is complementary to CADS: once we determine the attributes and values within the documents using CADS we will then use usher to model the dependencies across attributes and minimize the quantity of queries asked.

Probabilistic models: Probabilistic tag recommendation systems (Liu *et al*., 2009; Yin *et al*., 2010) have the same goal like our system. However, the most distinction is that we use the question workload in our model, reflecting the user interest.

METHODOLOGY

This is based on CADS (Collaborative Adaptive Data Sharing Platform) which is associate "annotate-as-you-create" infrastructure that makes simple to present fielded kind of information of the document, a key contribution of their system is that the direct use of the type of query research to direct the annotation method. They were trying to rank the annotation of documents towards generating attribute-value pair of attributes that are usually used by querying users. The first goal of CADS infrastructure is to encourage, support and lower the price of creating sophisticated and nicely annotated documents that can be useful for normally issued and kind of queries entered semi-structured queries. Their primary key goal is to encourage, support and provide the annotation of the documents provided or entered at creation time, though the techniques even be used for post generation document annotation whereas the creator of a particular document is within the phase of "document creation". Facilitation of document annotation using content and querying value system architecture is shown in Fig. 3.

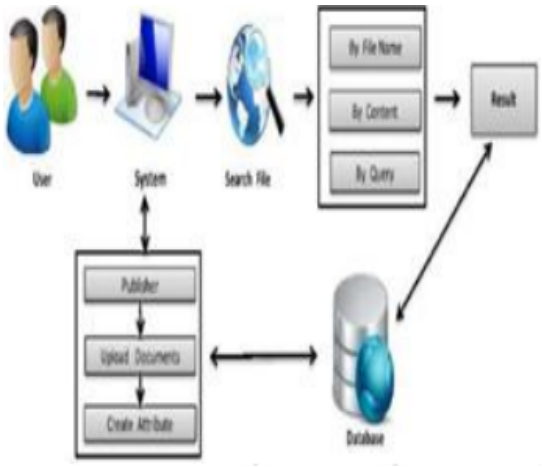

Fig. 3: System architecture of facilitating document annotation using content and querying value

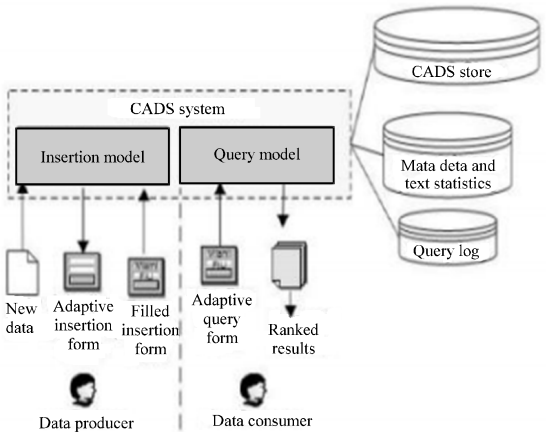

Fig. 4: CADS workflow

In their system the author generates a new document and uploads it to the repository. When the upload, CADS analyzes that and creates an adaptive insertion type. The form contains the simplest kind of attribute names provided to the document text and also the data want, i.e., query workload and the most of the probable attribute-values combine given the document text. The researcher, i.e., creator will determine the form of data, modify or change the generated metadata as necessary and needed and submit the annotated document for storage. CAD's model researches as shown in Fig. 4.

CONCLUSION

We proposed adaptive techniques to recommend relevant attributes to annotate a document whereas attempting to satisfy the user querying wants. Our

solution is predicated on a probabilistic framework that considers the proof within the document content and therefore the query workload. We present two ways that to combine these two items of evidence, content value and querying value: a model that considers each elements conditionally independent and a linear weighted model. Experiments shows that exploitation our techniques, we will suggest attributes that improve the visibility of the documents with respect to the query workload by up to 500th. That is, we show that using the query workload will greatly improve the annotation process and increase the utility of shared information.

REFERENCES

Banerjee, J., W. Kim, H.J. Kim and H.F. Korth, 1987. Semantics and implementation of schema evolution in object-oriented databases. ACM., 16: 311-322.

Cafarella, M.J., J. Madhavan and A. Halevy, 2009. Web-scale extraction of structured data. ACM. SIGMOD Rec., 37: 55-61.

Chen, K., H. Chen, N. Conway, J.M. Hellerstein and T.S. Parikh, 2011. Usher: Improving data quality with dynamic forms. IEEE Trans. Knowl. Data Eng., 23: 1138-1138.

Chu, E., A. Baid, X. Chai, A. Doan and J. Naughton, 2009. Combining keyword search and forms for ad hoc querying of databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, June 29-July 2, 2009, ACM, New York, USA., ISBN: 978-1-60558-551-2, pp: 349-360.

Doan, A., R. Ramakrishnan, F. Chen, P. DeRose and Y. Lee *et al*., 2006. Community information management. IEEE Data Eng. Bull., 29: 64-72.

Etzioni, O., M. Banko, S. Soderland and D.S. Weld, 2008. Open information extraction from the web. Commun. ACM., 51: 68-74.

Franklin, M., A. Halevy and D. Maier, 2005. From databases to dataspaces: A new abstraction for information management. ACM. Sigmod Rec., 34: 27-33.

Halevy, A.Y., Z.G. Ives, D. Suciu and I. Tatarinov, 2003. Schema mediation in peer data management systems. Proceedings of the 19th International Conference on Data Engineering, March 5-8, 2003, IEEE, USA., ISBN: 0-7803-7665-X, pp: 505-516.

Jain, A. and P.G. Ipeirotis, 2009. A quality-aware optimizer for information extraction. ACM Trans. Database Syst., Vol. 34,

Jayapandian, M. and H.V. Jagadish, 2008a. Automated creation of a forms-based database query interface. Proc. VLDB Endowment, 1: 695-709.

Jayapandian, M. and H.V. Jagadish, 2008b. Expressive query specification through form customization. Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology, March 25-30, 2008, ACM, New York, USA., ISBN: 978-1-59593-926-5, pp: 416-427.

Jeffery, S.R., Franklin, M.J. and A.Y. Halevy, 2008. Pay-as-you-go user feedback for dataspace systems. Proceedings of the International Conference on Management of Data, June 09-12, 2008, ACM, New York, USA., ISBN: 978-1-60558-102-6, pp: 847-860.

Liu, D., X.S. Hua, L. Yang, M. Wang and H.J. Zhang, 2009. Tag ranking. Proceedings of the 18th International Conference on World Wide Web, April 20-24, 2009, ACM, New York, USA., ISBN: 978-1-60558-487-4, pp: 351-360.

Nandi, A. and H.V. Jagadish, 2007. Assisted querying using instant-response interfaces. Proceedings of the International Conference on Management of Data, June 11-14, 2007, ACM, New York, USA., ISBN: 978-1-59593-686-8, pp: 1156-1158.

Yin, D., Z. Xue, L. Hong and B.D. Davison, 2010. A probabilistic model for personalized tag prediction. Proceedings of the 16th ACM International Conference on Knowledge Discovery and Data Mining, July 25-28, 2010, ACM, New York, USA., ISBN: 978-1-4503-0055-1, pp: 959-968.