International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

A Survey on Facilitating Document Annotation Using Content and Querying Value

Patil Vishal A.[1], Khambre Pankaj[2]

[1]Master of Computer Engg, Savitribai Phule Pune University, G. H. Raisoni Collage of Engineering and Technology, Wagholi, Pune, India

[2]Professor, Savitribai Phule Pune University, G. H .Raisoni Collage of Engineering and Technology, Wagholi, Pune, India

Abstract: ***Now-a-days many organizations generate and share descriptive and textual data of their products, services, and actions. Such type of collections of textual data contain significant amount of structured information, which remains saved in the unstructured text. While information extraction algorithms facilitate the extraction of structured relations in an very expensive and inaccurate way especially when operating on top of text that does not contain any instances of the targeted structured information. There are many alternative approaches that facilitate the generation of the structured metadata by identifying documents that are likely to contain information of interest and this information is going to be subsequently useful for querying the database which relies on the idea that humans are more likely to add the necessary metadata during creation time. This can be done like prompting by the interface; or that it should made much easier for humans (and/or algorithms) to identify the metadata when such information actually exists in the document, instead of naively prompting users to fill in forms with information that is not available in the document. There are different algorithms that identify structured attributes that are likely to appear within the document, by jointly utilizing the content of the text and the query workload.***

Keywords: Document annotation, Adaptive forms, Collaborative Adaptive Data Sharing platform

1. Introduction

Now-a-days a lot of users create and share information on the different application domains. We can consider the examples of Social Networking Groups, different kind of blogs, Scientific Networks as well as some very sensitive areas like Disaster Management Networks. There are many existing tools for example Microsoft SharePoint which is the best Content Management Tool for sharing documents. Similarly Google Drives, Microsoft One Drive allow users to share documents and annotate them in an ad hoc manner. One of the popular network i.e. Google, Google Base allows users to define attributes for their objects or choose from predefined templates. Such annotations process can facilitate subsequent information discovery. The existing annotation systems allow only "untyped" keywords annotation: for example, "Climate Category 3" is the tag which can be used to annotate a weather report.

The Annotations containing untyped approaches can be considered less expressive than those strategies which use attribute-value pairs. So in attribute-value pair approaches, the information can be expressed as (Climate Category, 3). The "pay-as-you-go" querying strategy in DataSpaces is being used in the recent works toward using more expressive queries that advantage to such annotations. The DataSpaces are widely used to provide clues for integration of data while performing queries. The existing systems assumes that the data sources already contain structured data/information also the problem to match the query attributes with the source attributes. The "pay-asyou-go" querying is much more feasible than "attribute-value" annotation systems. The users should provide the accurate annotations while using the "attributevalue" pairs. They should have knowledge about the different schemas and field types which can be used for the annotations. The schemas may have a lot of fields which may become hectic & complicated work for the users which may result into ignorance towards entering annotations which may lead to unclear usefulness towards subsequent searches.

The Collaborative Adaptive Data Sharing platform (CADS) is an "annotate-as-youcreate" infrastructure that facilitates the annotations using fielded data. The CADS system is uses direct query to direct the annotation process as well as to examine the content of the documents. The purpose of CADS is to create nicely annotated documents in cheaper manner which are useful for commonly issued semistructured queries. The purpose of CADS is to create nicely annotated documents in cheaper manner which are useful for commonly issued semistructured queries. The authors generate a new document and upload it to the repository. After uploading that document, CADS analyzes the text and creates an adaptive insertion form. The form contains the best attribute names given the document text and the information need and the most probable attribute values given the document text.

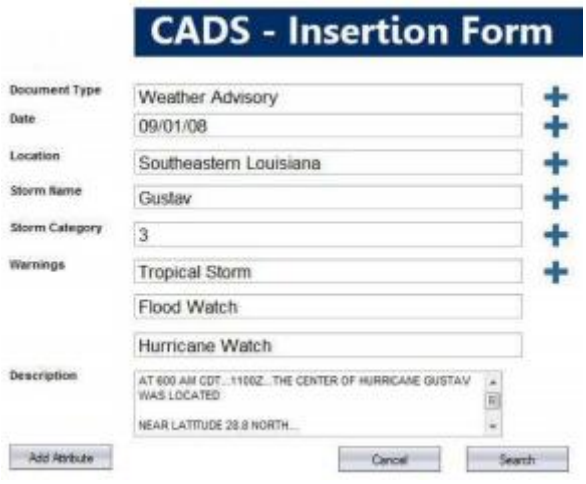

Figure 1: The Adaptive Insertion Form

Numerous annotation frameworks permit just "untyped" catchphrase annotation: for example, a client may clarify a climate report utilizing a tag, for example, "Climate Category 3". Annotation systems that utilization trait quality sets are for the most part more expressive, as they can contain more data than untyped methodologies. In such settings, the above data can be entered as (ClimateCategory, 3). A late profession towards utilizing more expressive questions those influence such annotations, is the "pay-as-you-make a go at" questioning methodology in DataSpaces.

In DataSpaces, clients give information reconciliation insights at inquiry time. The supposition in such frameworks is that the information sources as of now contain organized data and the issue is to coordinate the question traits with the source properties. Numerous frameworks, however, don't even have the essential "characteristic quality" annotation that would make a "pay-as-you make a go at" questioning possible. Annotations that utilization "property estimation" sets oblige clients to be more principled in their annotation endeavors. Clients have to know the fundamental outline and field sorts to utilize; they have to additionally know when to utilize each of these fields. With mappings that frequently have tens or even many accessible fields to fill, this errand get to be confounded and unwieldy which outcomes in information section clients disregarding such annotation capacities.

2. Literature Survey

2.1 Eduardo J. Ruiz, Vagelis Hristidis, and Panagiotis G. Ipeirotis, "Facilitating Document Annotation Using And Querying Value"

In this paper, they proposed CADS (Collaborative Adaptive Data Sharing platform) which is an "annotate-as-you-create" infrastructure that facilitates fielded data annotation. A key contribution of this system is the direct use of the query workload to direct the annotation process, in addition to examining the content of the document. In other words, they are trying to prioritize the annotation of documents towards generating attribute values for attributes that are often used by querying users.

Disadvantage:

1) Unavailability of proper information to diff erent levels of query. "Coordinate matching" by inner product similarity.
2) It does not provide more accurate data only get the files from only exact name required.
3) Unavailability of proper information rarely differentiates the search results.

2.2. K. Saleem, S. Luis, Y. Deng, S. C. Chen, V. Hristidis, and T. Li, "Towards a Business Continuity Information Network for Rapid Disaster Recovery"

Most of the recent work has been conducted for crisis management under terrorist attacks and emergency management services under natural disasters with private business continuity and disaster recovery a secondary concern. In this paper, they proposed a model for pre-disaster preparation and post-disaster business continuity/rapid recovery. The model is utilized to design and develop a web based prototype of our Business Continuity Information Network (BCIN) system facilitating collaboration among local, state, federal agencies and the business Community for rapid disaster recovery. They have presented a model and prototype with Hurricane Wilma as the case study.

2.3. Ronald Fagin, Amnon Lotem, Moni Naor, "Optimal aggregation algorithms for middleware":

Assume that each object in a database has m grades, or scores, one for each of m attributes. For example, an object can have a color grade, which tells how red it is, and a shape grade, that tells how round it is. For each attribute, there is a sorted list, which lists each object and its grade under that attribute, sorted by grade (highest grade first). Each object is assigned an overall grade, which is obtained by combining the attribute grades using a fixed monotone aggregation function, or combining rule, such as min or average. To determine the top k objects, that is, k objects with the highest overall grades, the naive algorithm must access every object in the database, to find its grade under each attribute. Fagin has given an algorithm (Fagins Algorithm, or FA) that is much more efficient. For some monotone aggregation functions, FA is optimal with high probability in the worst case. They analyzed an elegant and remarkably simple algorithm (the threshold algorithm, or TA) that is optimal in a much stronger sense than FA. They showed that TA is essentially optimal, not just for some monotone aggregation functions, but for all of them, and not just in a high-probability worst-case sense, but over every database. Unlike FA, which requires large buff ers (whose size may grow unboundedly as the database size grows), TA requires only a small, constant-size buff er. TA allows early stopping, which yields, in a precise sense, an approximate version of the top k answers. They distinguished two types of access: sorted access (where the middleware system obtains the grade of an object in some sorted list by proceeding through the list sequentially from the top), and random access (where the middleware system requests the grade of object in a list, and obtains it in one step). They considered the scenarios where random access is either impossible, or expensive relative to sorted access, and provide algorithms that are essentially optimal for these cases as well.

2.4. Collaborative Annotation

Many systems use the collaborative annotation of objects and use previous annotations or tags to annotate new objects. It has been a significant amount of work in predicting the tags for documents or other resources like webpages, images, and videos [7], [8]. These approaches have diff erent assumptions on the expected inputs depending on the object and the user involvement. Otherwise, the goals are similar to find missing tags that are related with the object. We can say that this approach is diff erent as workload to augment the document visibility uses the tagging process. When we compare this with the other approaches and the precision is a secondary goal that the annotator can improve the annotations on the process.

2.5. DataSpaces and pay-as-you-go integration:

The Integration Model of CADS and DataSpaces are similar, where a loosely integration model is proposed for heterogeneous sources. The DataSpaces integrate existing annotations for data sources and to answer queries [9]. The previous works suggest the appropriate annotations during insertion time. This also considers the query workload to identify the most promising attributes to be added. The Google Base is another related data model, where users can specify their own attribute/value pairs. And such proposed attributes in Google Base [2] are hard-coded for each item category (e.g., real-estate property). In CADS, the goal is to learn which attributes to be suggested. "Payas-you go" integration techniques like "PayGo" [3] [10] are useful to suggest candidate matching at query time. No previous works consider this problem at the time of insertion similar to CADS.

2.6. Content Management Softwares

Softwares like Microsoft SharePoint [11], and SAP NetWeaver [12] allow users to share documents, annotate them, and perform simple keyword queries. Using specialized insertion forms hard-coded attributes can be added. CADS basically used to improve these platforms by learning the user information demand and adjusting the insertion forms accordingly.

2.7. Information Extraction:

IE is related to the context of value suggestion for the computed attributes [13]. IE can be separated into two main efforts: 1. Closed IE 2. Open IE. In Closed IE, the user should define the schema and then the system populates the tables with relations extracted from the text. The recent work on attribute suggestion naturally complements closed IE identifying which attributes are likely to appear within a document. The information is then retracted; IE can be employed to extract the values for the attributes. Open IE [14] is closer to the needs of CADS. Open IE generates RDF like triplets, for example, (Gustav is category 3) with no input from the user. Open IE leads to a very large number of triplets, which means 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 execution of structured queries that require knowledge of the attribute names and values that appear within a document. In principle, we could use Open IE, and then pay-as-you-go solutions for identifying equivalency relations across attribute names; however, it is much better to deal with the problem early-on, during document generation, instead of trying to fix issues that could be prevented with proper design. The CIMPLE project uses IE techniques to create and manage data-rich online communities, like the DBLife community. In contrast to CIMPLE, where data are extracted from existing sources and a domain expert must create a domain schema, CADS is a data sharing environment where users explicitly insert the data and the schema automatically evolves with time. Nevertheless, the IE and mass collaboration techniques of CIMPLE can help in creating adaptive insertion forms in CADS.

2.8. Schema Evolution:

The adaptive annotation in CADS can be viewed as semiautomatic schema evolution [15]. Previous work on schema evolution did not address the problem of what attribute to add to the schema, but how to support querying and other database operations when the schema changes.

2.9. Query Forms:

Existing work on query forms can be leveraged in creating the CADS adaptive query forms. Jayapandian and Jagadish [16] propose an algorithm to extract a query form that represents most of the queries in the database using the "querability" of the columns, while they extend their work discussing forms customization. Nardi and Jagadish [17] use the schema information to autocomplete attribute or value names in query forms. Keyword queries are used to select the most appropriate query forms. Recent work can be considered a dual approach: instead of generating query forms using the database contents, they create the schema and contents of the database by considering the content of the query workload (and the contents of the documents, of course). The work in USHER [18] is also related: in USHER, the system automatically decides which questions in a survey are the most important to ask, given past experience with the completion of past surveys. In a sense

3. Conclusion

In this survey, we studied adaptive techniques to suggest relevant attributes to annotate a document, while trying to satisfy the user querying needs. Many papers are based on a probabilistic framework that considers the evidence in the document content and the query workload. There present two ways to combine these two pieces of evidence, content value and querying value: a model that considers both components conditionally independent and a linear weighted model. Experiments show that using these techniques, attributes that can improve the visibility of the documents with respect to the query workload by up to 50%. That is, we conclude that using the query workload can greatly improve the annotation process and increase the utility of shared data.

References

[1] Eduardo J. Ruiz, Vagelis Hristidis, and Panagiotis G. Ipeirotis, "Facilitating Document Annotation Using Content and Querying Value" [1] IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 2, February 2014.
[2] "Google," Google Base, http://www.google.com/base, 2011.
[3] S.R. Jeffery, M.J. Franklin, and A.Y. Halevy, "Pay-as You-Go User Feedback for Dataspace Systems" Proc. ACM SIGMOD Intl Conf. Management Data, 2008.
[4] K. Saleem, S. Luis, Y. Deng, S. C. Chen, V. Hristidis, and T. Li, "Towards a Business Continuity Information Network for Rapid Disaster Recovery," Proc. Intl Conf. Digital Govt. Research (dg.o 08), 2008.

[5] A. Jain and P.G. Ipeirotis, "A Quality-Aware Optimizer for Information Extraction" ACM Trans. Database Systems, vol. 34, article 5, 2009.
[6] R. Fagin, A. Lotem, and M. Naor, "Optimal Aggregation Algorithms for Middleware," J. Computer Systems Sciences, vol. 66, pp. 614-656, http://portal.acm.org/citation. cfm?id= 861182.861185, June 2003.
[7] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W.-C. Lee, and C.L. Giles, "Real-Time Automatic Tag Recommendation," Proc. 31st Ann. Intl ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 08), pp. 515-522, http://doi.acm.org/10.1145/ 1390334.1390423, 2008.
[8] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green, "Automatic Generation of Social Tags for Music Recommendation," Proc. Advances in Neural Information Processing Systems 20, 2008.
[9] M. Franklin, A. Halevy, and D. Maier, "From Databases to Dataspaces: A New Abstraction for Information Management," SIGMOD Record, vol. 34, pp. 27-33, http://doi.acm.org/10.1145/ 1107499.1107502, Dec. 2005
[10] J. Madhavan et al., "Web-Scale Data Integration: You Can Only Afford to Pay as You Go," Proc. Third Biennial Conf. Innovative Data Systems Research (CIDR), 2007.
[11] Sap Content Manager, https://www.sdn.sap.com/irj/sdn/nw-cm, 2011.
[12] Microsoft Sharepoint, http://www.microsoft.com/ sharepoint/, 2015.
[13] M.J. Cafarella, J. Madhavan, and A. Halevy, "Web-Scale Extraction of Structured Data," SIGMOD Record, vol. 37, pp. 55-61, http://doi.acm.org/10.1145/1519103.1519112, Mar. 2009.
[14] O. Etzioni, M. Banko, S. Soderland, and D.S. Weld, "Open Information Extraction from the Web," Comm. ACM,vol.51,pp.68-74, http://doi.acm.org/10.1145/1409360.1409378, Dec. 2008.
[15] J. Banerjee, W. Kim, H. J. Kim, and H.F. Korth, "Semantics and Implementation of Schema Evolution in Object-Oriented Databases," Proc. ACM SIGMOD Intl Conf. Management Data, 1987.
[16] M. Jayapandian and H.V. Jagadish, "Automated Creation of a FormsBased Database Query Interface," Proc. VLDB Endowment, vol. 1, pp. 695-709, http://dx.doi.org/10.1145/1453856.1453932, Aug. 2008.
[17] A. Nandi and H.V. Jagadish, "Assisted Querying Using Instant Response Interfaces," Proc. ACM SIGMOD Intl Conf. Management Data, 2007.
[18] K. Chen, H. Chen, N. Conway, J.M. Hellerstein, and T.S. Parikh, "Usher: Improving Data Quality with Dynamic Forms," Proc. IEEE 26th Intl Conf. Data Eng. (ICDE), 2010.