International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 10 094 - 097

Adaptive Technique for Document Annotation to Identify Attributes of Interest

Nupoor Gade
Computer Engineering,
Alard College of Engineering and Management,
Pune, India
nupoor.gade@gmail.com

Prof. Rugraj
Computer Engineering,
Alard College of Engineering and Management,
Pune, India

***Abstract*—** Many application domains generate and share information which describes their products and services. Such description contains unstructured information. So, it is always difficult to find the useful metadata. The information extraction algorithms are very expensive or inaccurate when operating on such unstructured information. This paper proposes adaptive technique for document annotation process to retrieve the useful information. This approach is based on Collaborative Adaptive Data Sharing (CADS) platform for document annotation. A CADS uses query workload to direct the annotation process. A key attribute of CADS is that it identifies important data attributes of the application. Further it uses this information to direct the data insertion and querying**.**

Keywords*- Annotation, Adaptive technique, Collaborative platform, CADS, Information extraction*

*****

I. INTRODUCTION

Today, users create and shares useful information by using various application domains like social networking, blogs, news, various management groups etc. There are many information sharing tools like content management tools, goggle base and many more. Some annotation systems allow users to share documents and annotate them in an ad hoc way. Some allows users to define attributes for their objects or choose from predefined templates. Many annotations systems allow only untyped keyword annotations. Some annotation strategies use attributes value pairs which is expensive as it contains more information than untyped annotations.

Annotations systems, which do not have the basic "attribute-value" annotation makes a "pay-as you- go" querying feasible. Annotations that use "attribute -value" pairs require users to be more principled in their annotation efforts. In some schemas, users needs to fill more than hundreds of fields, which makes the task tedious. Such difficulties provide basic annotations making annotations limited to simple keywords. It makes analysis and querying of data cumbersome.

To improve quality of searching, documents needs to be properly annotated. Inappropriate annotations make it hard to retrieve the information and rank it properly. This paper proposes, Collaborative Adaptive Data Sharing platform i.e. annotate-as-you-create infrastructure for annotating the document, which is fielded data annotation. The key novelty of the system is to lower the cost of document annotation. It provides query workload to direct the process of annotation. A CADS facilitates annotation of desired document at creation time.

II. RELATED WORK

Many systems support collaborative annotations of an object. There have been a significant amount of work in predicting the tags for documents or other resources (webpages, images, videos) [3], [4]. Aim of CADS is similar i.e. to find missing tags related to that particular object. The ability to use query workload after tagging process makes this approach unique.

The model of CADS is somewhat similar to dataspaces[12], where model is proposed for heterogeneous sources. A CADS considers the query workload to identify attributes and suggests annotations where author is in initial phase i.e. insertion time whereas dataspaces combines existing annotations.

Similarly, CADS is related to Google Base[2]. Here user can specify their own attributes and their values in addition to attribute values provided by user. But these values are hard coded.Pay-as-you go integration techniques like PayGo [5] and [6] are useful to suggest candidate matching at query time. However, no previous work considers this problem at insertion time, as in CADS.

Similarly, Microsoft Sharepoint [7] and SAP NetWeaver also allow sharing of documents, annotating the document and perform simple queries. A CADS improves these platforms by learning the user information demand and adjusting the insertion forms accordingly. This approach uses information extraction [8] for the value suggestion for the computed attributes.

III. PROPOSED WORK

CADS System Design

The aim of CADS is to lower the cost of document annotations and encourage the document annotations at time creation, when the author or document generator is in initial phase. As shown in figure 1, in CADS generator creates new document and uploads it in repository. A CADS analyzes all information present in the document and prepares an adaptive

insertion form. This form not only contains all important attribute names given in the document but also the information need (query workload). It also contains the attribute values present in the document. It provides modification facility to the author of the document where producer (or author) can inspect the form, modify if required. Finally, CADS stores the document.

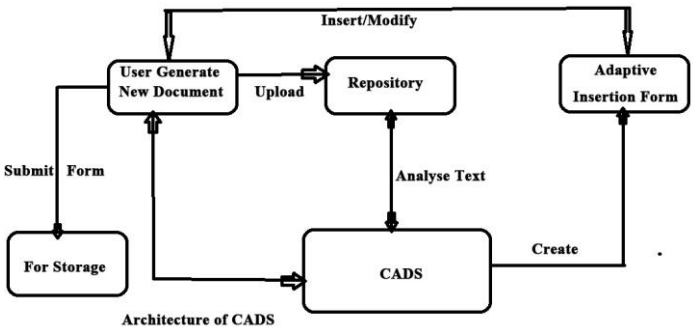

Fig. 1. Architecture of CADS

Basically, the system of CADS has two actors Consumers and Producers. Producers generate documents and upload data into the system. For this producer uses interactive insertion forms. By using adaptive query forms consumers searches for the relevant information. Insertion and query are the two major modules in CADS design as shown in figure 2.

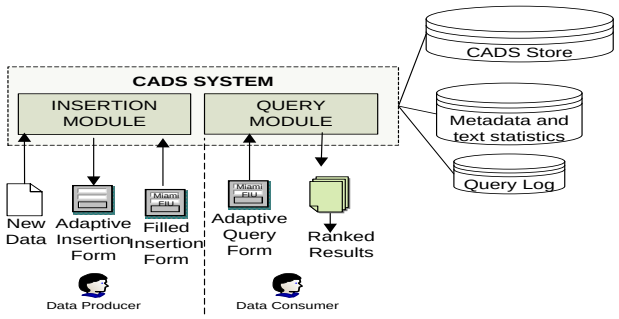

Fig. 2. CADS Workflow

Insertion phase: As stated, after storing the document, CADS analyzes the information and creates an adaptive insertion form (figure 3). It consists of the set of the most probable <attribute name, attribute value> pairs. The user fills this form and submits it. Now, the next and final stage is to store associated document and all metadata in the CADS repository.

Query phase: Here user is presented with adaptive query form. Initially, the query form consists of the default attributes. The user can specify additional attribute name, attribute value.It also contains Description attribute where user can specify keywords if he doesn't know how to use <attribute names , attribute values>.

As we are using query based searching, it helps to improve the quality of searching. This Query based searching can be performed in other formats also like .docx, .pdf, .xml etc. It also improves the performance which yields fast, better and accurate results. It also supports addition of images in order to understand the document.

Fig. 3. Adaptive insertion form

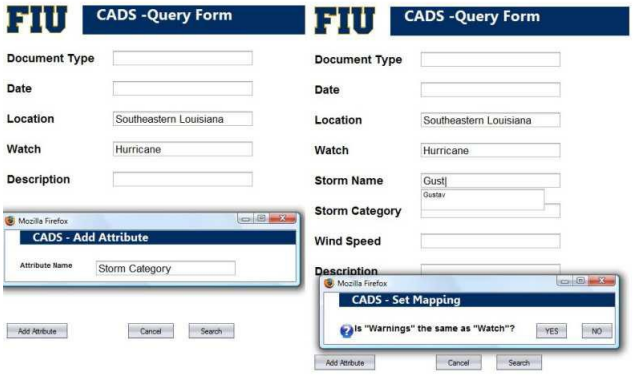

Fig. 4. Adaptive query form

The key contribution of this work is the "attribute suggestion" problem, which accounts for the query workload, and identifies the attributes that are present in the document, but not their values. There are two conflicting properties for indentifying and suggesting attributes for a document d.

- The attribute must have high querying value (QV) with respect to the query workload W.
- The attribute must have high content value (CV) With respect to d.

IV. MATHEMATICAL MODEL

Here we describe mathematical model for the entire system. Here, the goal is to compute a set of candidate annotation fields ^da, such that the conditional probability p(^da|W,dt) is maximized. The value p(da|W,dt) measures how probable a set of annotations is for a document, given the overall query workload for the database and the text of the specific document. Adopting this probabilistic framework, we can redefine the Attributes Suggestion problem as: (Probabilistic attribute suggestion)

Given a query workload W and a new document d, for which we only know its content dt, find a candidate set ^da of k attributes that maximize p(^da|W,dt).

When estimating language models, we consider each attribute Aj independently, and we compute the k attributes that maximize p(Aj|W,dt).

Here, d is the document, da is document annotations for d, W is the query workload and dt is document text for the document d, A is attributes used in the union of W and D(Repository), Aj is Attribute in A.
We define the score of attribute Aj as the odds that the attribute should appear in da.

$$Score(A_j) = \frac{p(A_j \mid W)}{1 - p(A_j \mid W)} \cdot \frac{p(d_t \mid A_j)}{p(d_t \mid \overline{A_j})} \quad (1)$$

Equation (1) is our score function. The first term represents the likelihood of producing Aj, given the workload W. We refer to that term as querying value as it expresses the "relevance" of the attribute to the query workload. The second term, which we refer to as content value, is the likelihood of observing the content dt given that the attribute Aj appears in the document.

Querying value: Let WAj ={Q € W : use(Q,Aj)} be the set of queries in W that use Aj as one of the predicate conditions. We use Laplace smoothing to avoid zero probabilities for the attributes that do not appear in the
workload, we have

$$p(A_j \mid W) = \frac{|W_{A_j}| + 1}{|W| + 1}$$

Content value: For the content value p(dt|Aj), our probabilistic model assume independence between the
terms in dt, which is a typical assumption when dealing with textual data (e.g., in probabilistic information retrieval,
text classification, language models, etc.) We have

$$p(d_t \mid A_j) = \prod_{w \in d_t} p(w \mid A_j)$$

IE(Information Extraction) Algorithm :
Step 1: Select a text file which is created by user
Step 2 : Parse the same file. Ignore stop words from it and count frequency of high querying keywords which will be important for content based search. Maintain the count of all these keywords.
Step 3: Upload the file on to the server
Step 4: Then fill all the annotations which are relevant to the document which can be useful for query based searching.

QV, CV Computation:
Combining QV and CV
The algorithm executes as follows:
1. Retrieve next Aj from LQV.
2. Get the Content Value for attribute Aj.
3. Calculate the threshold value T = F(CV,QV (Aj)), where CV is the maximum possible CV for
the unseen attributes and QV (Aj) is the QV of Aj.
4. Let R be the set of k attributes with highest score that we have seen. Add Aj to R if possible.
5. If the kth attribute Ak has Score(Ak)> T , we return R. Else, we go back to Step 1.

V. RESULT

First author needs to select unstructured document as a file to upload into the system database. To upload author needs to login and only registered person has access to the system. Its Document Annotation Using Content, the end user has to download the file on the system. For downloading the file end user has to enter content or query as there are two techniques for searching the annotation document first one is Content Search, second one is Query Search. In the content search document will be downloaded by giving the content search which is present in the corresponding annotation document. If its search result present the corresponding document will be downloaded, otherwise it is not downloaded. And second one is query search that the document will be downloaded by using simple query which has present in the base paper. If the result matches with the document then this document will be downloaded otherwise it rejected.

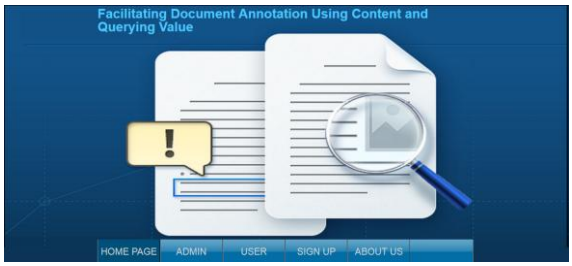

Fig. 5. Main page

A CADS analyzes all information present in the document and generates adaptive insertion form which is having annotations suggested by the system. Along with the system suggestions user can add his own annotations for particular document.

Using this system workload of the application can be reduced. As we are using query based searching it also increases efficiency of searching. Query based searching is also applied on other formats also like .docx, .pdf, .xml etc.

VI. CONCLUSION

Our system provides solution for document annotation. This proposes an adaptive technique in order to suggest relevant attributes for the document annotation. The solution is based on CADS i.e. collaborative adaptive data sharing platform. The aim of CADS is to minimize the cost of annotating the document. The Query based searching provides best results

where users can get accurate and faster results. It also increases performance.

VII. REFERENCES

[1] Eduardo J. Ruiz, Vagelis Hristidis, and Panagiotis G. Ipeirotis,Facilitating Document Annotation Using Content and Querying Value, IEEE TRANSACTIONS, VOL. 26, NO. 2, FEBRUARY 2014

[2] P. Heymann, D. Ramage, and H. Garcia-Molina, Social Tag Prediction, Proc. 31st Ann. Intl ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 08), pp. 531-538, http://doi.acm.org/10.1145/1390334.1390425, 2008.

[3] 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.

[4] 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.

[5] 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

[6] Microsoft, Microsoft Sharepoint, http://www.microsoft.com/ sharepoint/, 2012.

[7] 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.

[8] M. Jayapandian and H.V. Jagadish, Automated Creation of a Forms-Based Database Query Interface, Proc. VLDB Endowment, vol. 1, pp. 695-709, http://dx.doi.org/10.1145/1453856.1453932,Aug. 2008.

[9] A. Jain and P.G. Ipeirotis, A Quality-Aware Optimizer for Information Extraction, ACM Trans. Database Systems, vol. 34, article 5, 2009

[10] J.M. Ponte and W.B. Croft, A Language Modeling Approach to Information Retrieval, Proc. 21st Ann. Intl ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 98), pp. 275-281, http://doi.acm.org/10.1145/290941.291008, 1998.

[11] D. Yin, Z. Xue, L. Hong, and B.D. Davison, A Probabilistic Model for Personalized Tag Prediction, Proc. ACM SIGKDD Intl Conf. Knowledge Discovery Data Mining, 2010. ACEM

[12] 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.