INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

RESULT PAPER ON IMPLEMENTATION OF DOCUMENT ANNOTATION USING CONTENT AND QUERYING VALUE

MISS. ANUPAMA V. ZAKARDE[1], DR. H. R. DESHMUKH[2]

1. Dept. of Computer Science, I.B.S.S. College of Engineering, Amravati (Maharashtra).
2. Prof. & Head IBSS college of Engineering, Amravati.

Accepted Date: 15/03/2016; Published Date: 01/05/2016

Abstract: Annotation plays a major role in a user's reading of a document: from elementary school students making notes on text books to professors marking up their latest research papers. A common place for annotations to appear is in the margin of a document. Surprisingly, there is little systematic knowledge of how, why and when annotations are written in margins or over the main text. This project investigates how margin size impacts the ease with which documents can be annotated, and user annotation behavior. The research comprises of a two part investigation: first, a survey which examines margins and their use in physical documents; secondly, we evaluate document reader software that supports an extended margin for annotation in digital documents. This work present a novel alternative approach that facilitates 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. The approach relies on the idea that humans are more likely to add the necessary metadata during creation time, if prompted by the interface; or that it is 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.

Keywords: Document annotation, adaptive forms, collaborative platforms

Corresponding Author: MISS. ANUPAMA V. ZAKARDE

Access Online On:

www.ijpret.com

How to Cite This Article:

Anupama V. Zakarde, IJPRET, 2016; Volume 4 (9): 647-654

PAPER-QR CODE

INTRODUCTION

There are many application domains where users create and share information; for instance, news blogs, scientific networks, social networking groups, or disaster management networks. Current information sharing tools, like content management software (e.g., Micro soft Share-Point), allow users to share documents and annotate (tag) them in an ad hoc way. Similarly, Google Base allows users to define attributes for their objects or choose from predefined templates. This annotation process can facilitate subsequent information discovery. Many annotation systems allow only "untyped" keyword annotation: for instance, a user may annotate a weather report using a tag such as "Storm Category 3." Many systems, though, do not even have the basic "attribute-value" annotation that would make a "pay-asyou- go" querying feasible. Annotations that use "attributevalue" pairs require users to be more principled in their annotation efforts. Users should know the underlying schema and field types to use; they should also know when to use each of these fields. With schemas that often have tens or even hundreds of available fields to fill, this task becomes complicated and cumbersome. This results in data entry users ignoring such annotation capabilities. Even if the system allows users to arbitrarily annotate the data with such attribute-value pairs, the users are often unwilling to perform this task. The task not only requires considerable effort but it also has unclear usefulness for subsequent searches in the future: who is going to use an arbitrary, undefined in a common schema, attribute type for future searches? But even when using pre deter- mined schema, when there are tens of potential fields that can be used, which of these fields are going to be useful for searching the database in the future? Such difficulties results in very basic annotations, if any at all, that are often limited to simple keywords. Such simple annotations make the analysis and querying of the data cumbersome. Users are often limited to plain keyword searches, or have access to very basic annotation fields, such as "creation date" and "owner of document."

In this paper, we propose Collaborative Adaptive Data Sharing platform (CADS), which is an "annotate-as-youcreate" infrastructure that facilitates fielded data annotation. A key contribution of our 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, we are trying to prioritize the annotation of documents toward generating attribute values for attributes that are often used by querying users.

Example: Our motivating scenario is a disaster management situation, inspired by the experience in building a Business Continuity Information Network for disaster situations in

South Florida. During disasters, we have many users and organizations publishing and consuming information. For example, in a hurricane situation, local government agencies report shelter locations, damages in structures, or structural warnings. Meteorological Agencies report the status of the hurricane, its position, and particular warnings. Business owners describe the status and needs of their stores and personnel. Volunteers share their activities and look for critical needs. The information produced and consumed in this domain is dynamic and unpredictable, and agencies have their own protocols and formats of sharing data, for example, the Miami-Dade County Emergency Office publishes hourly document reports. Further, learning the schema from previous disasters is hard, as new situations, needs, and requirements arise.

Fig. Adaptive insertion form

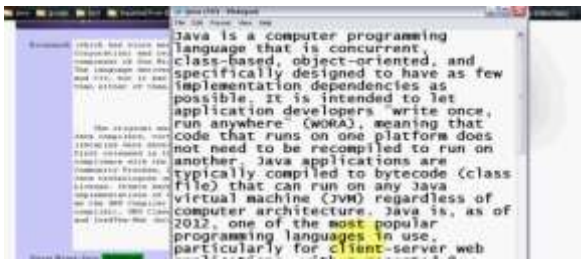

Fig. Sample Document and Annotation

2. LITERATURE REVIEW: Jain, P.G. Ipeirotis [1] Introduced a rigorous model for estimating the quality of the output of an information extraction system when paired with a document retrieval strategy.

R.T. Clemen & R.T. Winkler [2] In this paper, variety of approaches for combining probabilities. The inquiry has been how these models relate to principles of unanimity and compromise. Those models that provide for the most general patterns of dependence among sources are the most complex in terms of their conformance to the principles.

B. C. Russell , A. Torralba, K. Murphy, W. Freeman [3] In this paper, A web-based image annotation tool that was used to label the identity of objects and where they occur in images. We collected a large number of high quality annotations, spanning many different object categories, for a large set of images, many of which are high resolution.

M.J. Cafarella, J. Madhavan, and A. Halevy [4] In this paper, there are three systems that perform information extraction in a domain-independent fashion, and therefore can applied to the entire Web. In all three cases, a side result of the extraction is a set of entities, relationships and schemata that can be used as building blocks for the Web knowledge base and for additional semantic services.

O. Etzioni, M. Banko, S. Soderland, and D.S. Weld [5] This paper introduces Open IE from the Web, an unsupervised extraction paradigm that eschews relation-specific extraction in favor of a single extraction pass over the corpus during which relations of interest are automatically discovered and efficiently stored.

V. Uren, P. Cimiano, J.Iria , S. Handschuh, M. V. Vera , E. Motta , F. Ciravegna [6] In this paper, documents are central to KM, but intelligent documents, created by semantic annotation, would bring the advantages of semantic search and interoperability. These systems need

automation to support annotation, automation to support ontology maintenance, and automation to help maintain the consistency of documents, ontologies and annotations.

3. DOCUMENT ANNOTATION AND CALCULATION:

3.1 Conditional Independence Given Aj and Aj: We denote with $p(A_j \mid W, d_t, \mathcal{P})$ be the posterior probability that document d is annotated with Aj, given the forecast of W, d, and a prior belief P of CADS about the probability of adding Aj in any document. We define the score of attribute Aj as the odds that the attribute should appear in da. Using the Bayes theorem:

$$Score(A_j) = \frac{p(A_j \mid \mathcal{P}, W, d_t)}{p(\overline{A_j} \mid \mathcal{P}, W, d_t)} = \frac{p(\mathcal{P}, W, d_t \mid A_j) \cdot p(A_j)}{p(\mathcal{P}, W, d_t \mid \overline{A_j}) \cdot p(\overline{A_j})}.$$

The numerator and denominator are equivalent to the joint distributions $p(\mathcal{P}, W, d_t, A_j)$ and $p(\mathcal{P}, W, d_t, \overline{A_j})$, respectively. Using the chain rule on both terms:

$$Score(A_j) = \frac{p(\mathcal{P}) \cdot p(A_j \mid \mathcal{P}) \cdot p(W \mid A_j, \mathcal{P}) \cdot p(d_t \mid A_j, \mathcal{P}, W)}{p(\mathcal{P}) \cdot p(\overline{A_j} \mid \mathcal{P}) \cdot p(W \mid \overline{A_j}, \mathcal{P}) \cdot p(d_t \mid \overline{A_j}, \mathcal{P}, W)}$$

If W is independent of P, given A, and dt is independent of W, P, we simplify

$$Score(A_j) = \frac{p(A_j \mid \mathcal{P}) \cdot p(W \mid A_j) \cdot p(d_t \mid A_j)}{p(\overline{A_j} \mid \mathcal{P}) \cdot p(W \mid \overline{A_j}) \cdot p(d_t \mid \overline{A_j})}$$

Our prior belief P is independent of p(Aj), as we are not using any external knowledge to affect the estimates. So, the above equation can be further simplified to

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

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.

4. EXPERIMENTAL & RESULT: In particular, we show how pipelined algorithms can be employed to compute the top-k attributes with the highest scores, where scores are defined using Bayes strategy or Bernoulli strategy. In both strategies, we need to find efficient ways to calculate the Querying Value and Content Value components, which are defined in similar ways for the two strategies. We observe that in both strategies the score is a monotonically increasing

function $f(QV, CV) = CV \cdot QV$ for Bayes and $f(QV, CV) = \beta_1 \cdot QV + \beta_2 CV$ for Bernoulli.

4.1 QV Computation: A key observation is that the QV of an attribute is independent of the submitted document, QV only depends on the query workload. Hence, we maintain a precomputed list L^{QV} of QVs of the attributes in D_A, ordered by decreasing QV values. Since the query workload does not change significantly in real time, we update L^{QV} only periodically, as new queries arrive, since it is not critical for the QV metrics to be absolutely up-to-date: approximations suffice.

4.2 CV Computation: In contrast, it is expensive in terms of time and space to maintain all the CVs for all pairs of documents and attributes, where CV is defined. For that, we compute the CVs at runtime when a document arrives. The goal is to minimize the number of such computations when computing the top-k attribute suggestions. Given a document dt, we compute CV as follows: We first parse dt. For each term w € d_t, we compute its contribution. For that, we exploit two indexes: the inverted index It indexes the text of all documents, and the inverted index Ia stores for each attribute name Aj the list of documents for which Aj € d_a. To compute the numerator $DAj_{,w}$, we intersect the lists for Aj from the two indexes It and Ia. The denominator DAj is computed directly using I_a. We refer to this algorithm as GetCV (Aj).

4.3 Combining QV and CV: We employ a variation of the Threshold Algorithm with

Restricted Sorted Access (TA_Z). The pipelining algorithm performs sequential access on L^{QV} and for each seen attribute Aj it performs a "random access" to compute CV by executing GetCV (Aj).

The algorithm executes as follows:

1. Retrieve next Aj from L^{QV}.

2. Get the Content Value for attribute Aj.

3. Calculate the threshold value $\tau = F(CV, QV(A_j))$, 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. Add Aj to R if possible.

5. If the kth attribute A_k has Score $(A_k) > \tau$, we return R. Else, we go back to Step 1.

Note that instead of using TA_Z to combine CV and QV, we could have used the MPro algorithm, where the key difference is that sequential accesses has cost 0, and the execution is scheduled such that the number of random accesses are minimized. For simplicity, and since the efficiency of such computations is not the core contribution of this paper, we only present the results that we observed using the TA_Z algorithm.

5. RESULT:

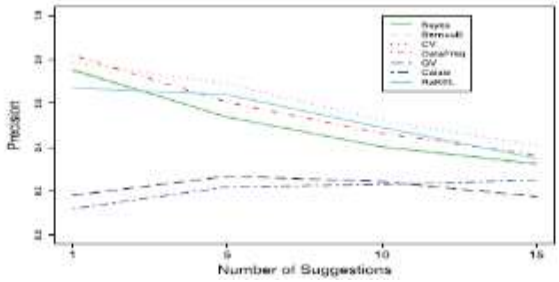

Fig. (a) Precision for the Emergency Dataset

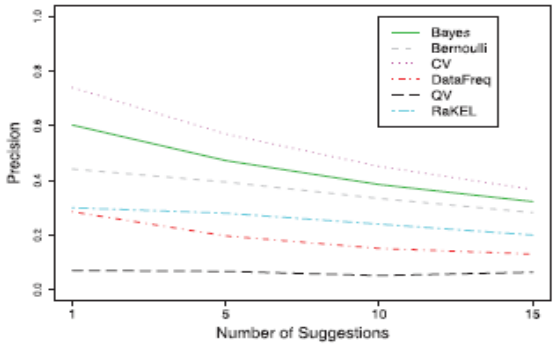

Fig. (b) Precision for the CNET Dataset

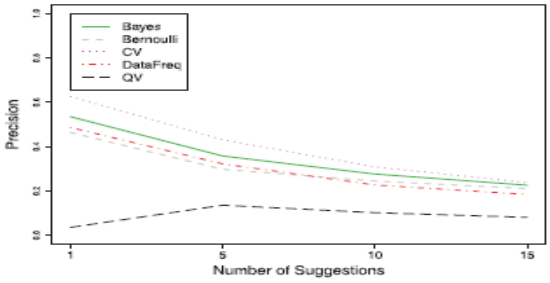

Fig. (c) Precision for the Amazon Dataset

6. CONCLUSION: A large number of organizations today generate and share textual descriptions of their products, services, and actions. Such collections of textual data contain significant amount of structured information, which remains buried in the unstructured text. While information extraction algorithms facilitate the extraction of structured relations, they are often expensive and inaccurate, especially when operating on top of text that does not contain any instances of the targeted structured information.

7. REFERENCES:

1. A. Jain and P.G. Ipeirotis, "A Quality-Aware Optimizer forInformatio-n Extraction," ACM Trans. Database Systems, vol. 34,article 5, 2009

2. R. T. Clemen and R.L. Winkler, "Unanimity and Compromise among Probability Forecasters," Management Science, vol.36, pp.767779, http://portal.acm.org/citation.cfm?id =81610.81609, July 1990.

3. B. Russell, A. Torralba, K. Murphy, and W. Freeman, "LabelMe: A Database and Web-Based Tool for Image Annotation," Int'l J.Computer Vision, vol. 77, pp. 157-173, http://dx.doi.org /10.1007/s11263-007-0090-8, 2008, doi: 10.1007/s11263-007-0090-8.

4. 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.

5. 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.

6. V. Uren, P. Cimiano, J.Iria , S. Handschuh ,M. V. Vera , E. Motta , F. Ciravegna, "Semantic annotation for knowledge management: Requirementsand a survey of the state of the art", 1570-8268/$ doi:10.1016/j.websem.2005.10.002

7. Eduardo J. Ruiz, Vagelis Hristidis, and Panagiotis G. Ipeirotis," Facilitating Document Annotation Using Content and Querying Value", ieee transactions on knowledge and data engineering,vol. 26, no. 2, february 2014.