# EFFICIENT DATA PRE-PROCESSING TECHNIQUE FOR INFORMATION RETRIEVAL

**M.Aarthy**

*Abstract*— **A probabilistic topic model for analyzing and extracting the content-related annotations data from noisy annotated discrete data such as web pages stored using social Add bookmark services. With these services, because users can attach annotations data at freely, some annotations data details did not describe the semantics of the content of the data, thus they are noisy data, i.e., not content related data's. The extraction of content-related data or keyword can be used as a prepossessing step in machine learning tasks such as text classification and image file recognition, or can improve information retrieval performance. Social annotation is an inwards data, on-line, collaborative process through which each element of a collection of resources files (e.g., URLs, pictures, videos, etc.) is associated with a group of descriptive keywords, widely known as tagged files. Each group is a query keyword and accurate summary of the relevant resource's content and it is obtained via aggregating the opinion of users, as expressed to the form of short tag sequences process. The availability of this information gives rise to new searching keywords paradigm where resources are retrieved and ranked based on the similarity of a keyword query to their accompanying tags. The proposed model is a generative model for content and social annotation data's, in which the annotations are assumed to originate either from topics that generated the content or from a general distribution unrelated to the content. We demonstrate to make effectiveness of the proposed method by using synthetic data and real social annotation data for text and images**.

***Index Terms***— Social Annotation, Search and replace Algorithm, Tag Allocation Model, Weighting.

## I. INTRODUCTION

Social annotations offer us a huge amount of user generated labeled data, see Yahoo! Delicious1 for examples. However, unlike expert-annotated data set, social annotations expose two liabilities: ambiguity and noise. Ambiguity rises when the users assign multiple tags to a single document. Figure 1 shows an example. In this example, the document has tags such as "*AI*" and '*international*", but we don't know which part of the document solicits each tag. To the computer, every word in the document is related to all the tags. Noise is the nature of most user-generated content, and social annotation data is not an exception [1].

Social annotation allows any string to be used as tags. In the meantime, the users are not professional annotators; they hold no responsibility to keep accuracy and consistency either. When we use social annotation as a labeled data set, reducing ambiguity and identifying noise tags [2] are important. First, ambiguity reduction helps us to find out the real intention behind a tag, which leads to higher accuracy in related applications. On the other hand, the benefit of identifying and removing noise is straightforward. For example, if another user posts the news about web intelligent, we would not like to recommend "*my favorite*" to him, since he may be more interested in some other discipline. When dealing with Web-scale information, an automatic solution that can separate noise from good tags is more appropriate.

### *A. Annotation*

Annotation is typically defined as "extra information asserted with a particular point in a document or other piece of information". Prior research has demonstrated that making annotation is an important accompanying activity to reading, with annotations used for diverse purposes.

Patrick et al (2004) discussed three attributes that are used to describe annotations, which are content, form, and functionality. Annotation content could be either very understandable to an occasional reader, or very personal in meaning. Annotation forms (types) include styles such as underlining and colouring, and different positions such as within the document and stand alone. Annotation functionalities include reading, editing, linking, and sharing.

Various types of annotations could be made on hardcopy documents, such as highlighting, commentary, link making, reading records, etc. (Marshall 1998). Marshall also noted that annotations are a primary vehicle for supporting collaboration around documents.

Manuscript received Aug 24, 2014

**M.Aarthy**, Research Scholar, Department of Computer Science and Engineering, Bharathidasan University, Tiruchirappalli, India

Many purposes of making annotations have been identified. Marshall (1998) found that annotations were used to bookmark important sections, to make interpretive remarks, and to fine-grain highlight to aid memory. O'Hara and Sellen (1997) discovered that people use annotations to help them understand a text and to make the text more useful for future tasks. Annotations are often helpful for other readers as well, even when they are not made with others in mind.
Glover, Xu, and Hardaker (2007) point out the two key advantages of inserting annotations into the web page, which are being able to share those notes with others and the ability to access the annotations from any web enabled computer. What's more, "annotations also provide third party, subjective metadata about the content of a web page that can be analysed to provide additional information for use in web searching and dynamic link generation".

### *B. Web annotation*

Web annotation has been a popular research topic since the invention of hypertext technology and accompanying web technologies as well as the steady increase in web-based materials. As stated earlier, this paper will use the following definition of web annotations:
"A Web annotation is an online annotation associated with a web resource, typically a web page. With a Web annotation system, a user can add, modify or remove information from a Web resource without modifying the resource itself."
Heck et al. (1999) concluded that the solution to the "incomplete information and wasted time" problem would be the "instantiation of an annotation tool that can be used to make private, public, or shared annotations, or notes, on already existing web pages."
Fu et al (2005) identified four types of annotation systems, which are annotation functionality built into web browsers, personalized web information organization systems, interactive web publication forums, as well as annotation engines. They concluded that "the four types of annotation tools provide almost all the functions that can be accomplished on paper", furthermore, these tools "also provide some functions which are difficult to realize in the paper environment". However, they also mentioned that no single approach is available to support all of the features.
According to Denoue & Vignollet (2000), an annotation system usually consists of three modules: the first is used to view existing annotations, the second to create new annotations, and the third to store the annotations. Vasudevan and Palmer (1999) reviewed web annotation system's architecture, and suggested that new technologies such as the Document Object Model (DOM) level 2 will be desirable to design high-quality annotation systems. Patrick et al. (2004) described a "Conceptual architecture of the individual mode of WATs (Web useful for future tasks. Annotations are often helpful for other readers as well, even when they are not made with others in mind.
Glover, Xu, and Hardaker (2007) point out the two key advantages of inserting annotations into the web page, which are being able to share those notes with others and the ability to access the annotations from any web enabled computer. What's more, "annotations also provide third party, subjective metadata about the content of a web page that can be analysed to provide additional information for use in web searching and dynamic link generation".

### c. *Text Annotation*

Text annotation is the practice and the result of adding a note or gloss to a text, which may include highlights or underlining, comments, footnotes, tags, and links. Text annotations can include notes written for a reader's private purposes, as well as shared annotations written for the purposes of collaborative writing and editing, commentary, or social reading and sharing. In some fields, text annotation is comparable to metadata insofar as it is added post hoc and provides information about a text without fundamentally altering that original text. Text annotations are sometimes referred to as marginalia, though some reserve this term specifically for hand-written notes made in the margins of books or manuscripts. This article covers both private and socially shared text annotations, including hand-written and information technology-based annotation, as well as Web-based text annotation.

### *D. Annotation structure*

The structural components of any annotation can be roughly divided into three primary elements: a body, an anchor, and a marker. The body of an annotation includes reader-generated symbols and text, such as handwritten commentary or stars in the margin. The anchor is what indicates the extent of the original text to which the body of the annotation refers; it may include circles around sections, brackets, highlights, underlines, and so on. Annotations may be anchored to very broad stretches of text (such as an entire document) or very narrow sections (such as a specific letter, word, or phrase). The marker is the visual appearance of the anchor, such as whether it is a grey underline or a yellow highlight. An annotation that has a body (such as a comment in the margin) but no specific anchor has no marker.

## II. FRAMEWORK FOR COMPARINNG OF ANNOTATIONS SCHEMS

Seven typical functionalities/features that are supported by various annotation systems will be discussed below [5]:

### *A. Highlighting*

Highlighting is typical when users make annotations on a document, either traditional paper or web pages. With the support of highlighting, users can select a portion of web pages, for example, a range of texts, part of a paragraph, etc.

### *B. Annotating*

Annotating is the most basic function of a web annotation system, which allows users to post textual annotations to web pages. All of the web annotation systems (tools) that were talked in this paper support annotating.

### *C. Tagging*

Some web annotation systems provide the functionality of tagging. Users can associate some keywords or terms with the annotations they made, or the pages they bookmarked. What's more, people can flexibly organize and share their own libraries of annotated web pages.

### *D. Searching*

The ability to search within the annotation repository is useful and may offer higher precision and faster response times than search using search engines.

### *E. Bookmarking*

Bookmarking is an essential part of some web annotation systems, such as Diigo, JumpKnowledge, and SharedCopy. Users can save URL of web pages which they have highlighted or annotated for future reference, or for sharing with friends and colleagues.

### *F. Sharing/Collaborative*

Sharing/collaborating in web annotation systems is the ability to let users share links, comments, annotations or the annotated web page with their friends or colleagues for collaborative research or study.

### *G. Page Capturing*

Different from bookmarking, page capturing allows users to save a copy of the web page. Some systems use page capturing to provide the functionalities of annotating and bookmarking, such as Fleck and SharedCopy.

## III. ROPOSED WORK

Suppose that, we have a set of documents, and each document consists of a pair of words annotations, where is the set of words in a document that represents the content, and is the set of assigned annotations, or tags. Our notation is summarized as follows.

The proposed topic model first generates the content, and then generates the annotations. The generative process for the content is the same as basic topic models, such as latent Dirichlet allocation. Each document has topic proportions id that are sampled from a Dirichlet distribution. For each of the words in the document, a topic is chosen from the topic proportions, and then word is generated from a topic-specific multinomial distribution. In the generative process for annotations, each annotation is assessed as to whether it is related to the content or not. In particular, each annotation is associated with a latent variable with value if annotation is not related to the content otherwise.

If the annotation is not related to the content, annotation is sampled from general topic-unrelated multinomial distribution. If the annotation is related to the content, annotation is sampled from topic-specific multinomial distribution where is the topic for the annotation. Topic is sampled uniform randomly from topics that have previously generated the content. This means that topic is generated from a multinomial distribution, in which, where is the number of words that are assigned to topic in the document in summary.

### *A. Search and Replace Algorithm*

It certainly would be possible to carefully define an algorithm to search for text that spans runs, noting where the searched text intersects bookmarks, comments, and the like. However, this algorithm would be pretty complicated, and to be done properly, a test team would need to write extensive test specs, and supply a plethora of sample documents that exercise all edge cases. It is a non-trivial project.

However, there is another approach that we can take that is pretty simple, easy to test, and yields the correct results in all cases. The algorithm consists of:

- Concatenate all text in a paragraph into a single string, and search for the search string in the concatenated text. If the search text is found, then continue with the following steps.
- Iterate through all runs in the paragraph, and break all runs into runs of a single character. There are a variety of special characters, such as carriage return, hard tab, break, and the non-breaking hyphen character. Normally, these special characters will coexist in runs with text elements. When breaking runs into runs of a single character, these special characters should also be placed into their own run. At the end of this process, no run will contain more than a single character, whether it is a character of text, or one of the special characters that is represented by an XML element.
- After breaking runs of text into multiple runs of single characters, it is then pretty easy to iterate

through the runs looking for a string of runs that match the characters in the search string.

- If the algorithm finds a string of runs that match the search string, then it inserts a new run into the document. This new run contains the run properties of the first run in the string of runs that match the search string. In addition, the algorithm deletes the set of single-character runs that matched the search string. This process is repeated until no strings of runs are found that match the search string.
- After the algorithm replaces the single-character runs with a new run containing the replacement text, then the algorithm coalesces adjacent runs with the same formatting into a single run.

### *B. SEARCHING*

We now have all the machinery in place for ranking a collection of tagged resources. The first step required is to train a bigram model for each resource, which involves the bigram and unigram probability computation and the optimization of the interpolation parameters. At query time we can compute the probability of the query keyword sequence being "generated" by each resource's bigram model. More precisely, the score of each resource R, given a query Q = q1. . . qk, is

$$PR(q1,\dots,qk) - \prod_{j=1}^{K} P(qj|qj-1) \quad \dots (1)$$

### *C. Noise Detection*

As mentioned earlier, our definition of noise is based on the assumptions that the more presentation styles that are used to compose an element node the more important the element node is and that the more diverse that the actual contents of an element node are, the more important the element node is. We now define what we mean by noises and give an algorithm to detect and to eliminate them.

**Definition (**noisy**)**: For an element node *E* in the SST, if all of its descendent and itself have composite importance less than a specified threshold *t*, then we say element node *E* is *noisy*. The algorithm *Mark Noise* (*E*) to identify noises in the SST. It first checks whether all *E*.s descendants are noisy or not. If any one of them is not noisy, then *E* is not noisy. If all its descendants are noisy and *E*.s composite importance is also small, then *E* is noisy.

**Input:** *E*: root element node of a SST

**Return**: *TRUE* if *E* and all of its descendants are noisy, else *FALSE*

**ALGORITHM: Mark Noise** (*E*)

1: **for** each *S* □*E.Ss* **do**
2: **for** each *e* □ *S.Es* **do**
3: **if** (**Mark Noise** (*e*) == *FALSE*) **then**
4: **return** *FALSE*
5: **end if**
6: **end for**
7: **end for**
8: **if** (*E.CompImp* ≤ *t*) **then**
9: mark *E* as .noisy.
10: **return** *TRUE*
11: **else return** *FALSE*
12: **end if**`

**Definition (**maximal noisy element node**)**: If a noisy element node *E* in the SST is not a descendent of any other noisy element node, we call *E* a maximal noisy element node. In other words, if an element node *E* is noisy and none of its ancestor nodes is noisy, then *E* is a maximal noisy element node, which is also marked by the algorithm.

**Definition (**meaningful**)**: If an element node *E* in the SST does not contain any noisy descendent, we say that *E* is meaningful.

**Definition (**maximal meaningful element node**)**: If a meaningful element node *E* is not a descendent of any other meaningful element node, we say *E* is a maximal meaningful element node.

### *D. Term-Weighting Scheme*

One of the best evolved schemes for identifies the illegal words is as follows:

$$R \Lambda N \quad \dots\dots\dots\dots\dots (2).$$

The term-frequency factor is normalised (tf) as follows:

$$wN \ulcorner wR \quad \dots\dots\dots\dots\dots (3).$$

## CONCLUSION

In this paper, we have proposed a topic model for extracting content related annotations from noisy annotated data. The proposed model can be applied in both implicit and partially explicit relevance settings, and it can also be used as the preprocessing for different classifiers as well as for modeling noisy annotated data. We have confirmed experimentally that the proposed method can extract content-related annotations appropriately, and can be used for analyzing social annotation data. We proposed a technique to clean Web pages for Web data mining. Observing that the Web pages in a given Web site usually share some common layout or presentation styles, we propose a new tree structure, called Style Tree (ST) to capture those frequent presentation styles and actual contents of the Web site. The site style tree (SST) provides us with rich information for analyzing both the structures and the contents of the Web pages. We also proposed an information based measure to evaluate the importance of element nodes in SST so as to detect noises. To clean a page from a site, we simply map the page to its SST. Our cleaning technique is evaluated using two data

mining tasks. Our results show that the proposed technique is highly effective.

## REFERENCE

[1] Jiang Bian, Ding Zhou, Shuyi Zheng, Hongyuan Zha, C. Lee Giles." Exploring Social Annotations for Information Retrieval". Social Networks & Web 2.0 - Applications & Infrastructures for Web2.0.

[2] Xiance Si ; Maosong Sun." Tag Allocation Model: Model Noisy Social Annotations By Reason Finding". 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.

[3] Kashoob.S. ; Caverlee,J. ; "Probabilistic Generative Models of the Social Annotation Process", IEEE xplore, 2009.

[4] Shenghua Ba, Xiaoyuan Wu, Ben Fei, Guirong Xue1, Zhong Su2, and Yong Yu1; "Optimizing Web Search Using Social Annotations"; In Proc. WWW 2007, 501–510.

[5] Paul Heymann, Georgia Koutrika, Hector Garcia-Molina; " **Can social bookmarking improve web search?" ;** WSDM '08 Proceedings of the 2008 International Conference on Web Search and Data Mining.

[6] VU Thanh Nguyen; "Using social annotation and web log to enhance search engine" IJCSI International Journal of Computer Science Issues, Vol. 6, No. 2, 2009.

[7] Jennifer Fernquist, Ed H. Chi;" Re-testing the Perception of Social Annotations in Web Search"; In *Proc. CIKM 2009*, 1227–1236.

[8] C. J. Van Rijsbergen, Information Retrieval, Butterworth-Heinemann, Newton, MA, 1979

[9] Charniak E., Berland M.: Finding parts in very large corpora. In Proceedings of the 37th Annual Meeting of the ACL, pages 57-64, 1999.

[10] Glover E., Tsioutsiouliklis K., Lawrence S., Pennock D., and Flake G.: Using web structure for classifying and describing web pages. In Proc. of the 11thWWW Conference, pages 562-569. ACM Press, 2002.

[11] Reeve L., Hyoil Han: Survey of semantic annotation platforms. In SAC '05, pages 1634-1638, NY, USA, 2005. ACM Press. ISBN 1-58113-964-0.

[12] Handschuh S., Staab S.: Authoring and annotation of web pages in cream. In WWW '02, pages 462-473,NY, USA, 2002. ACM Press. ISBN 1-58113-449-5.

[13] Marti Hearst. Multi-paragraph segmentation of expository text. In 32nd. Annual Meeting of the Association for Computational L inguistics, pages 9–16, 1994.

[14] G. Hirst and D. St-Onge. Lexical chains as representation of context for the detection and correction of malapropisms. In C. Fellbaum,editor,WordNet: An electronic lexical database and some of its ap-plications.The mit press. edition, 1997.

[15] Japan Electronic Dictionary Research Institute, Ltd., http://www.iijnet.or.jp/edr.EDR Electronic Dictionary Technical Guide (2nd edition), 1995.

[16] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of ROCLING X, 1997