Clustering for BYOP Toolkit

Dhruvkumar Patel
DA-IICT, Gandhinagar
200901130@daiict.ac.in
Supervisor
Prof. Prasenjit Majumder
Dr. Nisheeth Shrivastava
Bell Labs, Alcatel-Lucent, Manyata TechPark, Bangalore.

***Abstract*— Tracking of internet users is very concerning privacy issue of the current times. OSP (Online Service Providers) track users in order to provide them with personalized and behaviorally targeted content. There are some methods and tools to stop this behavioral tracking and personalized services, but they don?t provide users total control over their privacy and thus stops all the personalized content delivered to users. Many users may want personalized content on internet, but they might want to stop being tracked on few sensitive topics only. So, a tool which can show users their past internet usage in summarized form and let them choose on which topics they don?t want to be tracked, is much needed. Here, I present such tool?s functionality and implementation.**

***Index Terms*— Clustering, Summarization, Privacy, Categorization, Text-extraction.**

I. INTRODUCTION

Online service providers track lots of web users data, mainly being Search engines like Google and 3rd party advertisement networks like Google ad-sense and Yahoo advertisement network. Search Engines like Google, by default, keeps history of all the search queries made by user in past and search results clicked by user. Using this data Google makes online profile of the user. When user enter new search query, Google will use this data to provide personalized search results to user. For a particular use case, if some user who has Cancer disease searches a lot many queries related to cancer on Google, then if the person searches for some other query on Google, Google will rank query related to Cancer higher in the search results for this other query based on the past user data. But Cancer can be a sensitive topic for user and he/she may not want to be tracked on that topic and may not want personalized search results related to Cancer. 3rd party advertisement networks also do similar kind of users tracking. They have contracts with different websites and blogs, so whenever user visits the website, these ad-networks know that user has visited that webpage. These ad-networks practice this tracking all the time. Now when user browses to other sites to which these ad-networks are affiliated with, users get the personalized advertisement based on user?s past browsing history tracked by the ad-network. So if user visits some webpages which he/she thinks are sensitive, these third party ad-networks will still track user on those webpages and afterwards, show user personalized advertisements on other websites user visit.
There are some tools and Method available to stop this tracking. For Example,there are some browser plugins available using which you can remove all the ads from the webpages you visit,e.g AdBlock Plus. But such tools only stops advertisements being shown on browser. They don't stop ad networks and Search engines from tracking you. Also a new HTTP header has been proposed called 'DO NOT TRACK' which tells 3rd party advertisers not to track the user and current browsers has this setting which lets users to opt out of online tracking. Again,this is totally up to advertisement provider whether they want to follow this protocol or not. Very few popular advertisement networks are currently support DNT. Even if they do support this header in future, it will stop all kind of personalized content users view on the web, instead of only sensitive content on which users don't want to be tracked.
So we need a tool which can take user's past history, be it browsing history, search queries history, ads-clicked history. This can be very big collection of data items. So this tool should be able to show these data items in a summarized form to user, which means these data must be clustered into different categories. From these categories, user should be able to mark some of the categories as sensitive,on which he/she doesn't want to be tracked.

II. BYOP TOOLKIT: PRIVACY CONTROL FOR USER

BYOP is an acronym for 'Bring Your Own Privacy'. It's programmed in Java programming language. BYOP toolkit's primary aim is to let User choose sensitive topics on which user does not want to be tracked. We categorize user's past history into different clusters and put this clusters into categories.Then User can choose which categories or even clusters are sensitive. BYOP toolkit will stop tracking by 3rd Party Advertisers and Search Engine like Google on the sensitive topics chosen by User.

A. Functionality

BYOP toolkit lets user mark their past history as sensitive or non-sensitive by showing them the summary of the history. For this functionality, we need to cluster user's past history to reduce no of items, as user's past history can contain tens of thousands of different website urls and search queries. To be able to cluster this urls and search queries, we need text content for each of the data item to process on. Then we can

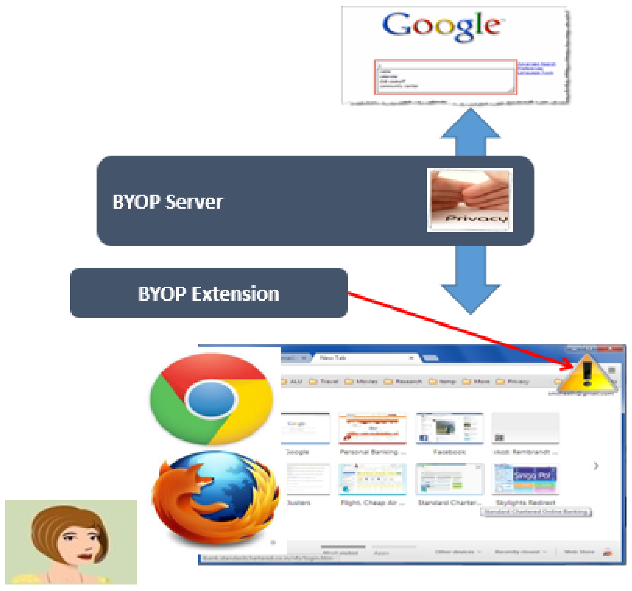


Fig. 1. BYOP Toolkit

apply clustering algorithm on this title and text content pair for each item to cluster them. After getting different clusters formed out of similar documents, we need to label each cluster in intuitive and user-understandable way so that user can see, what kind of documents the cluster contains, just by looking at the label. Then, we need to categorize these clusters into one of the predefined categories like Health, Finance or Movies. For categorization purpose, we can use some of the online directory project datasets like Google ad-categories or DMoz online directory project. Then we show this summarized view of history to user, where user can mark the clusters or documents or even categories as sensitive or non-sensitive. Now when user enters new search query or visits new webpage, we need to find its sensitivity in real-time. For this, we first fetch text content for the new document and then use some classifier to classify it into one of the pre-made clusters. Then we can look at which clusters this new document belongs to and whether user has marked those clusters sensitive or not, to define sensitivity of this new document. Based on this sensitivity score, we decide whether user should be marked on this new website visit or search query or not. So BYOP toolkit contains two components:

- Browser Extension on User's device: This performs data collection and user interaction. It send user's past history to server and show the clusters to user. It allows user to mark sensitivity of different clusters and send this settings to BYOP server.
- BYOP Server: It performs data analytics task like getting user's history and cluster it and send it back to browser extension. It also performs real-time classification task when user enters new search query or visits new webpage.

Thus, there are two main tasks BYOP toolkit should be able to perform:

- Summarize user's past browsing and search history.
  - Clustering and classifying these clusters into categories.
  - User can mark sensitivity of clusters.
- Find sensitivity of new url being visited in real-time.

For the summarization task, following are the main steps to be performed. We will look into each step in the same order.

1) Extract text from web documents or search queries.
2) Apply clustering algorithm.
3) Find intuitive and user-understandable labels.
4) Categorize the clusters into categories like Google ad categories or DMoz categories.

B. *Text Extraction*

Very important aspect to get good clustering results is to have good text content to cluster from. In order to get impressive clustering results, one must extract as good text content one can from the given data source.
For the clustering task in BYOP toolkit, we have following three different kind of data items to cluster:

- Links of webpages in User's browsing history
- User's search history queries
- Ads which had been shown to user in past while visiting these browsing history links.

For each of the data sources mentioned above, we want to extract text and make a new document object per item (item being a history link, a search query or an advertisement link/image) which will have two fields - a title and a text snippet or say text summary. For each of three kind of data items, we can perform different schemes to extract text content for clustering.

a) Extracting text content from browsing history links: For browsing history links, first we fetch html webpages using HTTP GET calls to all urls in java. After fetching html content for all the webpages of user's past history, we extract text content from html webpage using Jsoup. Jsoup is a java library which performs text extraction task on html webpages. It extracts all text from html,head, title and body tags. We get title tag of html webpage and set to a title of new document object. As for setting text content of new document object for the item, we have two options:

1) Whole body tag content from html document
2) Summary of body tag generated using some standard summarization algorithm

We tried both the methods. For second method, we used Lucene (version 4.2.0) search engine library to summarize the text content of html document. Algorithms is as following. We first add all the html documents into lucene index. Then we get interesting terms for this particular docummment out of Lucene index using Lucene's inbuilt method called getInterestingTerms(). These terms basically mean that they are rare and very unique to this particular document. Then we search for this terms in the document, and find out the complete sentences which have any of these terms. We weight these sentences by the frequency of the terms and no of the terms from interesting term-set. Then we pickup set of highest weighed top five sentences and tell it the summary of the document. This method gives very good results as for summary extraction task. But this summary does not improve clustering quality, but instead reduces clustering quality. The reason for this non-intuitive behavior being, most or all available clustering algorithms we have tried, basically cluster documents according to their semantics and their text content. Summarization of documents remove much of the text content from the documents and only contain unique and important sentences for the particular document. Thus after summarization, there hardly remains any semantic similarity among documents which are topic-wise similar and should fall into same cluster. So using summarization, clustering algorithms fail to put documents into same cluster. Because of this reason, we are using all the text content from body tag of html document as summary of document object for clustering.

b) Extracting text content from search queries of user's search history: For search queries, we want make the query itself the title of the new document object. To get summary field of the new document object, we make a query to Google search engine with the given query. Using HTTP GET call, we fetch Google's search results page for the given query, which contains around 10 search results for given query. Then we use our own html parser class to extract search results' titles and snippets from the fetched Search Results page. Now we concatenate all the search results snippets into one String and make it a summary of new Document object for the given query. This method works fairly well to get good clustering results of search queries.

c) Extracting text content from advertisements on user's browsing history pages: As another part of BYOP project, we have developed an ad-extraction tool which extracts advertisement data from webpages, which contains text on the advertisement as well as landing page links. Landing page is the webpage, where you will be redirected to if you click on that ad on the webpage. We can either use advertisement text content for clustering ads,or we can use landing page links' html content for clustering ads. As we have observed, using html content of landing page links works much better for clustering advertisements, as it contained lot more text and represents the category in which ad is supposed to belong.

C. Clustering of document objects

Clustering is the most crucial task of summarization of all these data items. After extracting document objects with two fields: title and summary for each of these data items (webpage, query or ad), we need to cluster these document objects. This task is crucial because it reduces no of document objects, which can be as higher as thousands, to fewer no of clusters. Thus it represents user's history in much better way for user to visualize his/her past browsing behavior. We have tried and implemented many different text clustering algorithms available, which I list below:

1) Bottom-up merging using cosine similarity
2) KMeans clustering
3) Topic Modeling
4) Lingo clustering Algorithm
5) Lingo3G
6) Clustering using Lucene

d) Bottom-up Merging: This is the most simple hierarchical clustering method. In this, we take set of all the documents to be clustered and extract all terms out of them. After finding list of terms, we remove standard set of stopwords from it. We used stopwords list provided by MIT on their website. After finding a vector of terms, we find tf(term frequency) of these terms in each documents, and thus for each document we make a term-frequency vector. Now we pickup a document randomly and compute its cosine similarity with all other documents and cluster it with top similarity documents above some threshold. Cosine similarity is computed by following formula:

$$S_{d1,d2} = V_{d1} \cdot V_{d2} \tag{1}$$

We do it until all documents are clustered. I implemented Bottom-up merging algorithm and applied to sample users' dataset we have and found following results:
Pros:

- Good clusters.

Cons:

- Very slow.
- Very high processing is needed.

e) KMeans Clustering: KMeans clustering is most prevalent and popular clustering scheme in text clustering area. In this algorithm, you take all documents and compute their term-frequency vectors just like Bottom-up merging. After getting tf-vectors for all documents, we define initial no of clusters as k. Total no of documents is n. Now we want to distribute these n documents into k clusters. So, initially we divide all documents randomly in these k clusters in equal numbers. Then we compute for each of the cluster, cluster-mean vectors, which are mean of tf-vectors of all documents contained in the cluster.
Now we repeat following process. We take each document and find out its Euclidian distance from each of the clusters. Here Euclidian distance is computed between document's tf-vector and cluster's cluster-mean vector. Then we remove the document from its current cluster and put it in the cluster having lowest distance to the document. We do it for all the documents in the set. After one such iteration, we keep iterating over this method until no more documents are moved. Then we take these new clusters as output.
I implemented KMeans clustering algorithm and applied to sample users' dataset we have, and foudn following results:
Pros:

- Tight and small clusters.
- All the semantically similar documents are clustered well.
- Fast enough.

Cons:

- Some 2-3 very big clusters get generated with random documents.

f) Topic Modeling: Topic Modeling is also a technique to divide set of documents into initially defined no of clusters (Here, topics). There are different versions of topic models implementations available including Explicit Semantic Analysis, Latent Semantic Analysis, Latent Dirichlet Allocation, Hierarchical Dirichlet Process. We experimented with LDA (Latent Dirichlet Allocation) implementation provided by Mallet tool, which is a machine learning toolkit developed by ML group at University of Massachusetts, Amherst. We did not get that good results with LDA implementation of Topic Modeling for the kind of dataset we have (Webpages, search queries and advertisements). We measured the output's clustering score with Weka toolkit, which again is a machine learning algorithms toolkit developed at University of Waikato, New Zealand. Weka is a standard toolkit for scoring clustering scores. We found 35% clustering Score for LDA's output of topics, which is very low for the clustering quality desired in BYOP toolkit. So Topic Models didn't turn out to be very useful for BYOP toolkit.

g) Lingo Clustering Algorithm: Lingo clustering algorithm is an algorithm for clustering of search results. It was introduced by S.Osinksi as part of his Masters thesis. Most classical text clustering algorithms first finds clusters of documents and then find labels of these clusters. Whereas Lingo algorithm first finds frequent phrases from documents set. Then it computes terms-document matrix and apply SVD (single value decomposition) to it. Then it chooses only first k elements from the SVD's first vaector (vector of basis), where k is chosen by selecting the Frobeinus Norms of term document matrix. And assign these k classed to top k phrases in phrase extraction task. It has its own labeling scheme, so we don't need to apply any labeling scheme afterwards. Implementation of Lingo clustering algorithm is available in Carrot Clustering workbench. We used its java library and checke results. *Pros:*

- Good labels for clusters.

Cons:

- Most clusters contain many outliers.
- Many documents (around 30%) remain unclustered.

h) Lingo3g Clustering Algorithm: Lingo3g is a closed source algorithm commercially available from Carrot Search tool. It is paid and algorithm behind the tool is not publicly available. We have tried a trial licensed version of Lingo3g and it gave most impressive results of all other algorithms we have tried. It has its own very good labeling scheme for clusters. *Pros:*

- Very good labels.
- Most of the clusters are good and tight.

Cons:

- Some clusters contain outliers.
- Many documents (around 30%) remain unclustered.

i) Clustering using Lucene: Lucene is a search engine library. So it is not primarily intended for clustering purpose, but is intended for search engine implementation, for example Nutch. But we can use Lucene for clustering purpose. In this method, you add all documents into a Lucene index. Then, you pick a document randomly and find Lucene query containing top terms from that document using Lucene's inbuilt method. Now we perform this query on Lucene index of all documents and get search results with search scores. We cluster up all the documents from search results having search scores above some constant threshold with the given document. Now we pick random document from remaining unclustered documents again and repeat the process until all documents are clustered. This clustering method is very fast because it totally uses Lucene's faster and optimal searching and comparing capibilities for tf-idf vectors stored in Lucene index. We get clusters for thousands of documents in millisecond time span. Since we have a method to get similar documents for a given text content, it is useful for real-time classification task too later. *Pros:*

- Very good and well-summarized clusters.
- Very fast.
- Maintaining Lucene index is useful for real-time classification task too.

Cons:

- Not too good labels.

j) Comparison between Lingo3G and Clustering using Lucene: We are using clusterin using Lucene method for BYOP toolkit. Since Lingo3G is closed source,it cant be tweaked for better clusters and labels. It also has outliers problem and problem of many documents remaining unclustered. Below I show clustering results using both the methods:

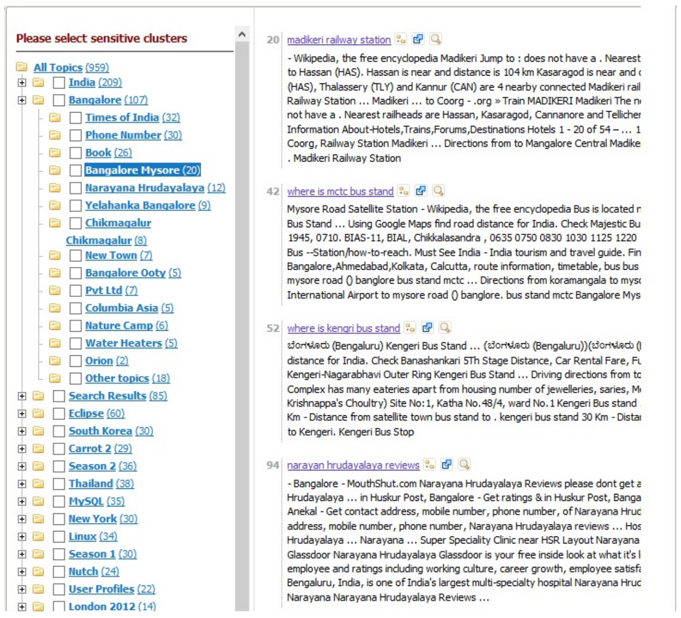
D. Labeling schemes

We have tried several methods to get god labels for given clusters like term frequencies, tf-idf scores etc. Tf-idf works just good enough. Tf-idf can be defined as:

$$tf - idf = tf \cdot idf where idf = l\frac{|D|}{df} \tag{2}$$

E. Categorization

This is very crucial task in summarization of user's history. After getting clusters from user's history documents and assign labels to the clusters, we want to put them into several predefined categories like Travel, Finance, Health etc. To be able to categorize into one these categories, we need a classifier and dataset to train the classifier. We can use online available categories and their datasets for example Google ad-categories, DMOz categories. We tried three approaches for classification task, which are as following:



Please select sensitive clusters

- All Topics (959)
  - India (209)
  - Bangalore (107)
    - Times of India (32)
    - Phone Number (30)
    - Book (26)
    - Bangalore Mysore (20)
    - Narayana Hrudayalaya (12)
    - Yelahanka Bangalore (9)
    - Chikmagalur Chikmagalur (8)
    - New Town (7)
    - Bangalore Ooty (5)
    - Pvt Ltd (7)
    - Columbia Asia (5)
    - Nature Camp (6)
    - Water Heaters (5)
    - Orion (2)
    - Other topics (18)
  - Search Results (85)
  - Eclipse (60)
  - South Korea (30)
  - Carrot 2 (29)
  - Season 2 (36)
  - Thailand (38)
  - MySQL (35)
  - New York (30)
  - Linux (34)
  - Season 1 (30)
  - Nutch (24)
  - User Profiles (22)
  - London 2012 (14)

20 madikeri railway station

- Wikipedia, the free encyclopedia Madikeri Jump to : does not have a . Nearest
to Hassan (HAS). Hassan is near and distance is 104 km Kasaragod is near and
(HAS), Thalassery (TLY) and Kannur (CAN) are 4 nearby connected Madikeri rail
Railway Station ... Madikeri ... to Coorg - .org » Train MADIKERI Madikeri The n
not have a . Nearest railheads are Hassan, Kasaragod, Cannanore and Tellicher
Information About-Hotels,Trains,Forums,Destinations Hotels 1 - 20 of 54 – ... 1
Coorg, Railway Station Madikeri ... Directions from to Mangalore Central Madike
. Madikeri Railway Station

42 where is mctc bus stand

Mysore Road Satellite Station - Wikipedia, the free encyclopedia Bus is located r
Bus Stand ... Using Google Maps find road distance for India. Check Majestic Bu
1945, 0710. BIAS-11, BIAL, Chikkalasandra , 0635 0750 0830 1030 1125 1220
Bus --Station/how-to-reach. Must See India - India tourism and travel guide. Fin
Bangalore,Ahmedabad,Kolkata, Calcutta, route information, timetable, bus bus
mysore road () banglore bus stand mctc ... Directions from koramangala to mys
International Airport to mysore road () banglore. bus stand mctc Bangalore Mys

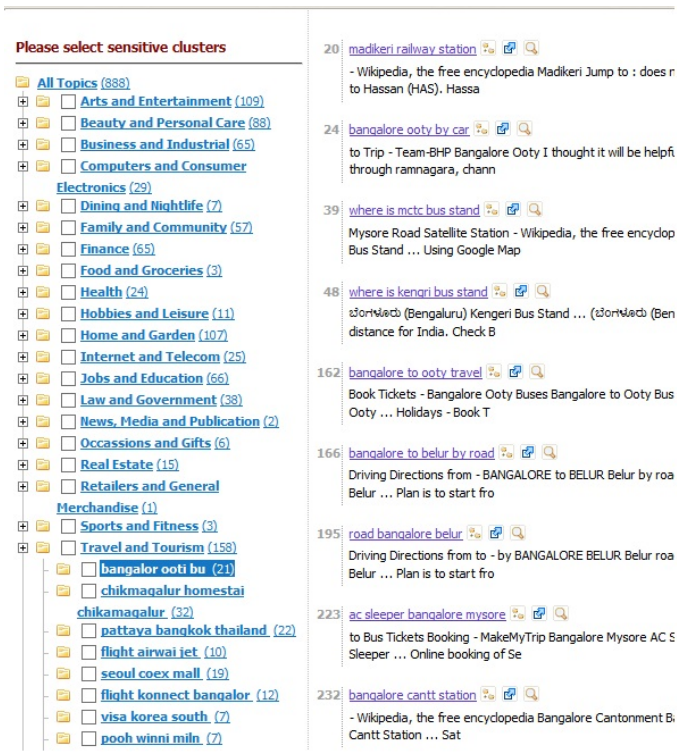
52 where is kengri bus stand

ಬೆಂಗಳೂರು (Bengaluru) Kengeri Bus Stand ... (ಬೆಂಗಳೂರು (Bengaluru))(ಬೆಂಗಳೂರು (
distance for India. Check Banashankari 5Th Stage Distance, Car Rental Fare, F
Kengeri-Nagarabhavi Outer Ring Kengeri Bus Stand ... Driving directions from t
Complex has many eateries apart from housing number of jewelleries, saries, M
Krishnappa's Choultry) Site No:1, Katha No.48/4, ward No.1 Kengeri Bus stand
Km - Distance from satellite town bus stand to . kengeri bus stand 30 Km - Dista
to Kengeri. Kengeri Bus Stop

94 narayan hrudayalaya reviews

- Bangalore - MouthShut.com Narayana Hrudayalaya Reviews please dont get a
Hrudayalaya ... in Huskur Post, Bangalore - Get ratings & in Huskur Post, Banga
Anekal - Get contact address, mobile number, phone number, of Narayana Hruc
address, mobile number, phone number, Narayana Hrudayalaya reviews ... Hos
Hrudayalaya ... Narayana ... Super Speciality Clinic near HSR Layout Narayana
Glassdoor Narayana Hrudayalaya Glassdoor is your free inside look at what it's l
employee and ratings including working culture, career growth, employee satisf
Bengaluru, India, is one of India's largest multi-specialty hospital Narayana Hruc
Narayana Narayana Hrudayalaya Reviews ...

Fig. 2. Clusters using Lingo3G



Please select sensitive clusters

- All Topics (888)
  - Arts and Entertainment (109)
  - Beauty and Personal Care (88)
  - Business and Industrial (65)
  - Computers and Consumer Electronics (29)
  - Dining and Nightlife (7)
  - Family and Community (57)
  - Finance (65)
  - Food and Groceries (3)
  - Health (24)
  - Hobbies and Leisure (11)
  - Home and Garden (107)
  - Internet and Telecom (25)
  - Jobs and Education (66)
  - Law and Government (38)
  - News, Media and Publication (2)
  - Occassions and Gifts (6)
  - Real Estate (15)
  - Retailers and General Merchandise (1)
  - Sports and Fitness (3)
  - Travel and Tourism (158)
    - bangalor ooti bu (21)
    - chikmagalur homestai chikamagalur (32)
    - pattaya bangkok thailand (22)
    - flight airwai jet (10)
    - seoul coex mall (19)
    - flight konnect bangalor (12)
    - visa korea south (7)
    - pooh winni miln (7)

20 madikeri railway station

- Wikipedia, the free encyclopedia Madikeri Jump to : does r
to Hassan (HAS). Hassa

24 bangalore ooty by car

to Trip - Team-BHP Bangalore Ooty I thought it will be helpf
through ramnagara, chann

39 where is mctc bus stand

Mysore Road Satellite Station - Wikipedia, the free encyclop
Bus Stand ... Using Google Map

48 where is kengri bus stand

ಬೆಂಗಳೂರು (Bengaluru) Kengeri Bus Stand ... (ಬೆಂಗಳೂರು (Ben
distance for India. Check B

162 bangalore to ooty travel

Book Tickets - Bangalore Ooty Buses Bangalore to Ooty Bus
Ooty ... Holidays - Book T

166 bangalore to belur by road

Driving Directions from - BANGALORE to BELUR Belur by roa
Belur ... Plan is to start fro

195 road bangalore belur

Driving Directions from to - by BANGALORE BELUR Belur roa
Belur ... Plan is to start fro

223 ac sleeper bangalore mysore

to Bus Tickets Booking - MakeMyTrip Bangalore Mysore AC S
Sleeper ... Online booking of Se

232 bangalore cantt station

- Wikipedia, the free encyclopedia Bangalore Cantonment B
Cantt Station ... Sat

Fig. 3. Clusters using Lucene

k) K Nearest Neighbors: We have in input set of categories and set of example webpage links for each of the category. We want to put given cluster or document in one of the category. In this method, we took set of all the example webpages and found some K nearest neighbors for given cluster using Euclidian distance measure. We assign the cluster to most occurring category with low distance. But since this method uses computing Euclidean distance with each of example webpages, it is too slow for BYOP toolkit.

l) Naive Bayesian Classifier: In this experiment, we have,in input,a set of categories and for each category, a set of keywords. We compute each keyword's probability of occurring into each of the categories. We compute feature vector for the given cluster or document, from the set of all keywords from all categories. Then we use Naive Bayesian Algorithm to find a category into which this feature vector has highest probability of occurring.

m) SVM: SVM (Support Vector Machines) are another classifier which we can train using same feature vectors described as above. SVM is much more accurate than Naive Bayesian. Instead of example set of keywords, we can train the classifier on users' real data. I have made a tool, using which we can put user's data into one of the categories manually and then train the classifier from it. The tool is ready, but manual training experiment is yet to be done.

F. Real-time Classification

After getting summarized view of past history, user marks sensitive clusters and categories and stores it. Now when user visits new webpage or makes new search query, we want to know its sensitivity. In order to do that, we need to classify the document into one of the pre-made clusters. Since we are using clustering using Lucene algorithms, it is very easy and fast to perform this task. We can get text out of the document (webpage or search query) as mentioned in Text Extraction task, search for the text using Lucene on whole index of user's history. From search results, we assign the document to cluster which occurs most times with high search score. Then we look up sensitivity of the cluster as marked by user in past, and decide whether newly being visited document is sensitive or not. Thus, whether user should be tracked upon that webpage/search query or not.

III. CONCLUSIONS

A tool which can show users their past internet usage in summarized form and let them choose sensitive categories and which they don't want to be tracked, is very much needed. BYOP Toolkit tried to solve this problem by providing the needed solution in a very efficient and secure manner.

ACKNOWLEDGMENT

I wish to acknowledge Dr. Anirban Majumder, Bell Labs, Bangalore for his contribution in Topic Modeling related experiments as well as constant feedback during whole project interim.

REFERENCES

[1] Anirban Majumder, Nisheeth Shrivastava, Know Your Personalization: Learning Topic level Personalization in Online Services, CoRR, 2012.
[2] S Osinski, An Algorithm for Clustering of Web Search Results, Master Thesis,2003.

[3] J MacQueen, Some Methods for Classification and Analysis of Multivariate Observations, 1967

[4] McDonald, A.M., and Cranor, L.F. Beliefs and behaviors: Internet user's understanding of behavioral advertising, In TPRC, 2010.

[5] David M. Blei, Introduction to Probabilistic Topic Models, Princeton University.