Web Intelligence 13 (2015) 31–41
DOI 10.3233/WEB-150308
IOS Press

Assisting web document retrieval with topic identification in tourism domain

Rajendra Prasath [a,*], Vijai Kumar [b] and Sudeshna Sarkar [a]

[a] *Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur - 721 302, India*
E-mails: rajendra@cse.iitkgp.ernet.in, sudeshna@cse.iitkgp.ernet.in
[b] *Cisco Systems, inc., Bangalore, India*
E-mail: viz.kec2009@gmail.com

Abstract. In this work, we present a domain specific Information Retrieval (IR) system that identifies query and document topics and use them for better documents retrieval. We focus on retrieving documents having the specific types of information as that of the user query related to the tourism domain. Based on our past experience in handling tourism specific information, we observed that the query intent in the tourism domain largely span over a few major types. Based on this observation, we present an approach for document retrieval based on query and documents type identification. To do this, we have identified the major types (topics) in the tourism domain and built an ontology of the tourism domain. We developed a document classifier to identify the topic of web documents, and a query classifier to identify the topic of the user query, both pertaining to the tourism domain. The proposed IR system performs document retrieval by matching the type of user query with the matching type of documents. The experimental results show that the tourism specific topic identification of queries and documents improves the retrieval of documents having more specific information to satisfy user queries in the tourism domain.

Keywords: Topic identification, query classifier, document classifier, tourism specific retrieval, retrieval efficiency

1. Introduction

Information Retrieval (IR) Systems accept a sequence of words as a query, perform the retrieval task on the specified collection of documents and retrieve a ranked list of documents as search results. Each retrieved document matches with the information needs specified by the user query only if the user perceives it to be relevant and useful. Queries are used to specify the user information needs, but they are often short having 2–3 keywords on an average [20] and are often ambiguous. IR systems have to identify and fill the gaps in such queries. This reduces the effectiveness of IR systems [7]. An IR system may be considered to be effective if a large proportion of top k ($k > 0$) retrieved documents are relevant. Because of under-specified or ambiguous queries, the retrieved top k documents may not contain documents that have information relevant to the user needs. Additionally, ranking of the retrieved documents with more specific information to the user query becomes really challenging in IR systems.

In this paper, we focus on the retrieval of documents with specific information needs related to the tourism domain. User queries in the tourism domain often ask for information about a place of interest, and contain the place name. For example, consider the query, "cheap comfortable way to reach Ooty". The user intention in this query is to find: "how to reach Ooty (comfortably by bus / train with cheap or economical fares)?" type of information. All documents that contain the actual information on different modes of transportation like "buses", "trains" with the cheapest or economical fares to Ooty, may not contain the actual query terms. Hence the query type needs to be identified so as to retrieve documents having the terms "bus", "train" pertaining to "how to reach" type of information specific to the place "Ooty".

*Corresponding author. E-mail: drrprasath@gmail.com.

The ambiguous query: "bus services in Java" submitted to Google,[1] with the user intent to know about the transportation services, especially the information on bus facilities in Java Island, fetched no documents pertaining to the query topic, and all retrieved documents pertained to "JAVA programming language". In this query, if we could identify the query intent "how to reach" from the phrase "bus services", we could look for documents that contain information about travel means, and could have given more importance to the phrases of "buses", "travel", "how to reach" during the retrieval of documents.

Consider the query: "staying in Alwar city". This query seeks information related to hotels / guest houses / hostels in the Alwar city and falls under the topic "accommodation". So more importance has to be given to those pages having accommodation related terms "hotels", "guest houses", "hostels" of the Alwar city rather than considering the importance of each of the direct query terms.

Let us consider a few more examples. The query, "kanchipuram varadharaja perumal temple", seeks the details about a particular topic: "places to visit", and specifically "attractions" in the city of "kanchipuram". Hence the query terms: "varadharaja perumal temple" pertaining to the specific topic, "attractions" in the city of "Kanchipuram" need to be given more importance.

Lastly, consider the user query, "climbing nanda devi". Documents that contain the query term, "devi", may refer to the temple related information. But the query term "climbing" seeks the information specific to the topic: "how to reach" the place called "nanda devi". So retrieving documents having the matching type information on "how to reach" is more important than the details of a particular "attraction".

In each of these queries, the specific focus of user information needs has to be identified with the right query term(s) pertaining to a specific topic and then the retrieval has to be performed to fetch the documents having the matching topic of the query. Since user information needs focus on a specific aspect of tourism related information either like places or services or hospitality, the supplied query terms are alone not sufficient to capture the underlying information needs and hence the additional information has to be identified and incorporated with user queries.

In this paper, we present a system to search for tourism specific documents pertaining to tourist places in India.

[1] Searched in Google on 26.06.2014

2. Review of literature

Since web queries are very short, consisting of 2–3 words on an average [1] and ambiguous [4], a query may belong to multiple topics. Goker [6] described a machine learning approach to infer the actual context behind user queries in an incremental way. This approach tried to learn the "problem situations" that represent the context of the query and the context learner helps to 'learn' from one query to another incrementally. Similar mechanisms exploited the interest and context of users in various topics and have the potential to improve web retrieval systems [21,22]. In 2003, Kang and Kim [10] showed that category information can be used to trigger the most appropriate vertical searches corresponding to a query, to improve topic relevance tasks (informational) and home page finding tasks (navigational). They used 'and' and 'sum' operators for matching query terms. In the case, 'and' operator means that the result document has all query terms in it. 'sum' operator means that a result document has at least one query term in it. Ozmutlu and Cavdur [15] investigated the properties of a specific topic identification methodology with Excite Web Search Engine data logs. In this method, the parameters (term weights and a threshold) for the topic identification algorithm are determined using topic shift and continuation probabilities. Carmel et al. [2] presented a model that captures the main components of a topic and the relationship between those components and topic difficulty. Again in 2008, Ozmutlu et al. [14] proposed a topic identification algorithm without considering the context of queries, but rather by using the statistical characteristics of the transaction log queries.

Web Query Classification (QC) has been studied for its wide usage in domain specific web search, personalized IR, online advertisement, etc. Many of the methods used follow the bag of words approach [12,18,24]. Lin and Chao [13] presented an algorithm that retrieves tourism related opinions which are then used to determine tourist attractions, that is, given an opinionated sentence, the algorithm determines whether it is tourism-related or not, and then decides which tourist attraction is the focus of the given opinion.

In order to understand user intentions, Pu et al. described an evaluation framework, namely ResQue (Recommender systems' Quality of user experience), to measure the qualities of the recommended items, the system's usability, usefulness, interface and interaction qualities, user satisfaction with the systems, and the influence of these qualities on users' behavioral

intentions, including their intention to purchase the products recommended to them [17]. Since the accuracy of the recommendation systems depends on the (partially identified) user experience, Knijnenburg et al. [11] proposed a user-centric approach to recommender system evaluation by linking objective system aspects to objective user behavior through a series of perceptual and evaluative constructs. This work also incorporates the behavioural correlates between the influence of personal and situational characteristics on the user experience.

Hull [8] proposed a query structuring method. The basic idea of query structuring is to group query keys and to use query operators in such a way that more weight is being assigned to important or correct keys than the other keys. In this method, query input format consists of a series of attribute-value pairs, each is considered as a concept and all terms, entered in that specific attribute-value pair, are combined using OR operator. The user may designate the importance of each concept and these concepts are combined using a weighted AND operator. Pirkola [16] showed that applying the query structuring on cross lingual information retrieval with translation equivalents of query terms with $n-$grams captures the clue on the intention of users' information need. Recently, D'hondt et al. [5] proposed a technique to automatically identify the topics present in a document, based on the presence of lexical chains. Xiang et al. [23] conducted a study focusing on understanding the representation of tourism related information through current search technologies on the Internet by analysing 1) the size and visibility of the tourism specific information provided by Google and, 2) the representation of tourism specific unique websites on different pages of search results.

3. Objectives

Our objective is to develop a domain specific IR system for tourism domain especially to retrieve documents pertaining to the tourist places in India. With the rapid expansion of the web, the volume of documents in the tourism domain grows very fast. All these documents may not have useful information as required by the users. Based on the study of user information needs in the tourism domain, we observed that the users often search for the most specific information centered around the places of their interest. Due to the popularity of information demands of the users in the tourism domain, a need was felt to develop a good vertical search engine for this domain.

Since the major types of query intent in this domain are a few, an effective retrieval engine can be built based on dentifying the types of information in the web documents and categorizing them under major tourism specific topics. If the query intent can be identified, this may assist the retrieval of relevant documents. Hence our goal is to develop an IR system that first identifies the type of user queries, the types of information in the documents and then performs retrieval of documents having topic specific information pertaining to the type of the user query.

4. The overview of the proposed IR system

In this section, we describe the overview of the proposed tourism specific document retrieval system. The proposed system consists of the following components: *tourism ontology*, *document topic classifier*, *query topic classifier* and *the proposed IR system*. The proposed IR system, in turn, consists of the components: *Content extraction with topic identification*, *Indexer* and *Searcher*. The work flow of the proposed system, represented in Fig. 1, is divided into the following phases:

1. **Initial Phase:** During this phase, tourism ontology is built by identifying the set of major query topics and the set of document topics, in the tourism domain. Using this tourism ontology and tourism queries collected from the web users, a query topic classifier is built. This query classifier can be used to identify the topic of the user query in the tourism domain. Similarly, using the text documents tagged with tourism related topics, the document topic classifier model is built using Naive Bayes approach. This model can be used to identify the topic(s) of a text document in the tourism domain.
2. **Offline Process:** The crawler obtains web documents. A web document in this crawled data may have noisy contents like advertisements, banners, forms and so on. Noise removal heuristic is applied on these documents and the values of various fields like title, meta tags, description, hyperlinks, the extracted text content, etc. are filtered out in the form of attribute-value pairs. Additionally, the tourism specific domain classifier checks the extracted text content, filters out the

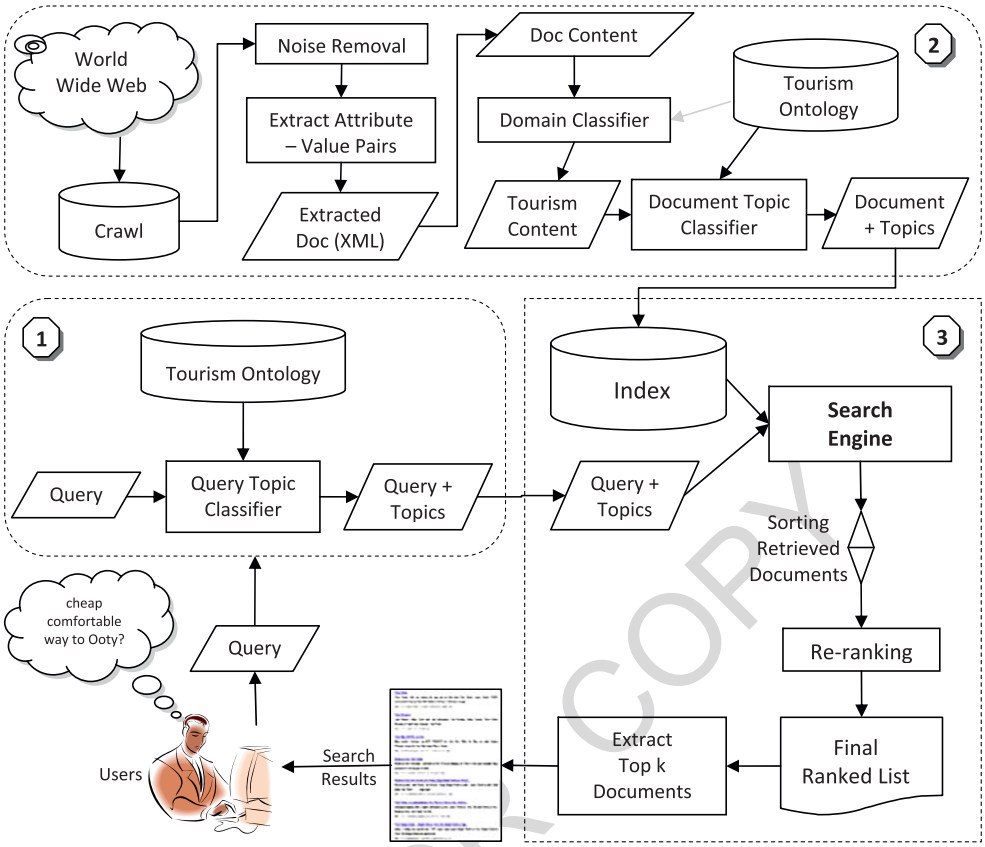

Fig. 1. The Architecture of the Proposed Tourism Specific IR System.

non-tourism text contents and allows the tourism related text content in the pipeline. This text content is split into coherent segments. Then document topic classifier is applied on each of these text segments and the topic of each segment is identified. The identified topic information along with other attribute-value pairs is stored in an XML format. Then the proposed IR system performs the indexing of the documents represented in XML formats and stores the index into the disk.

3. **Online Process:** When the proposed IR system receives the user query, it invokes the query topic classifier which in turn identifies and returns the topic(s) of the query. This topic information along with the supplied query terms is used by the searcher to retrieve the subset of documents (from the index) pertaining to the user query topic. Then re-ranking is applied on the retrieved set of documents with the matching topics associated boost factors. This finally produces the ranked list of documents from which top k documents pertaining to the query topic(s) are returned to the user as search results.

We next describe each component in detail:

4.1. Tourism ontology

Based on the analysis of tourism related web pages, we have identified a list of major topics in the tourism domain which are listed below.

1. *Attractions* – monuments and places to see in and around a specific tourism spot
2. *Activities* – events and festivals related information
3. *Accommodations* – "where to stay" and details about hotels / guest houses / hostels / dormitories /etc.
4. *Climate* – weather related information pertaining to the specific place
5. *Food* – where to eat / restaurants, cafeteria, motels related information and specific food items of that region

6. *Ideal Time* – the best time to [or not to] visit the specific place
7. *History* – importance of a place in ancient days, details about the rulers of that region
8. *Travel* – "how to reach" and "local transportation" related information
9. *Tour Packages* – tour operators details, guide services and sightseeing related information and
10. *Others* – travel guides / maps and other informations not pertaining to any of the above topics.

These tourism specific topics were used to identify the topics of documents and topics of the query.

4.2. Document topic classifier

The document topic classifier is a subsystem that identifies the tourism specific topic(s) of the extracted content of a web document and then tags the extracted content with one or more identified topic(s). The document topic classifier pertaining to the tourism domain can be built using either a simple bag of words approach [9,12,19,24] or decision rules [3] or probability estimations on the manually tagged tourism documents collection.

We created a document topic classifier using a collection of 458 tourism related web documents for which the topics were manually identified and the documents were tagged with the topics. Using this as the training data, we have built a document topic classification model using Naive Bayes method. This model, given the tourism related text content, identifies the most matching topic of that content with its matching score – *score(d, t)* – the score assigned by the classifier to the document d pertaining to the topic t. This identified topic with *score(d, t)* is stored in the index during the offline process. Later the searcher uses this information to retrieve the matching type of documents pertaining to the user query topic.

The offline processing module associated with the document topic classifier is pictorially represented by the component labelled as (2) in Fig. 1.

4.3. Query topic classifier

In this section, we describe the query topic classifier and its role in identifying the topics of user queries in the tourism domain.

The query topic classifier takes a query (a sequence of terms) describing the user information needs in the tourism domain, as an input. It identifies the likely query topic(s), and then enriches the query with the identified topic(s). The tourism specific query topic classifier can be built using the domain knowledge specific to the tourism domain.

In the tourism domain, most queries have been found to be information centric focusing on either popular attractions or a cheap hotel or a famous monument or a voyage or a similar entity, pertaining to that specific *place*. The query topic classifier, on receiving such user queries, uses the tourism ontology to identify the matching topic(s) of interest and expands the given query with the identified topic associated information. The online processing module associated with the query topic classifier is pictorially represented by the component labelled as (1) in Fig. 1.

The query topic classifier identifies the topic(s) of the tourism specific query either by matching:

- lists of topic related keywords with query terms, or
- user query patterns with the sequence of query terms, or
- using the tourism specific classification model developed by machine learning approaches.

Now let us describe each of these ways in detail.

4.3.1. Matching query terms with topic related keywords

We have identified the lists of keywords pertaining to each of the tourism specific topics. The user query having k terms is compared with each of the lists having the keywords specific to one topic in the tourism domain. The topic of the list having the maximum number of terms matching with the query terms is assigned as the identified topic to the given query. Basically this is a *Bag of Words*(BOW) approach. For example, if a query is related to the transportation seeking "how to reach a place by air / train / bus?", then the list having the keywords – route, reach, nearest, flights, airport, best, driving, road, bus, station, train, railways, from, to, till, upto – would be identified as the list with most matching keywords and hence the topic of this list: "travel" is assigned to the query as its topic. Similarly, for a food related query, the list having the keywords – restaurants, motels, mess, eat, food, vegetarian, non-vegetarian, dining, dinner, lunch, breakfast, taste, snacks, coffee, tea, cakes, juices – would be identified as the primary list with more matching keywords with the query terms. So "food" is assigned to that query as its topic. The matching score, between

query terms and the topic associated keywords, is calculated as follows:

$$\text{Score} = \frac{\text{\#of match between query terms and keywords}}{\text{\#of query terms}}$$

Using this, we will find the score for each query topic. Finally the topic of the query is determined by the topic of the list having the maximum number of matching keywords.

4.3.2. Matching user queries with query patterns

Since user queries in the tourism domain follow specific patterns centered around place names, regular expressions can be used to identify patterns pertaining to each of the tourism specific topics. For example, consider the query patterns: "how to reach ___ from ___ ?", "attractive ___ (around|near) ___", "best ___ to visit ___", "cheap ___ to stay in ___" and so on. While using these patterns, the blank spaces like ___ in a pattern, have to be replaced with the name of the place, a user is interested in to visit or search for relevant information. Hence we rearrange the query terms in a meaningful way so as to find the matching query pattern pertaining to the topic of the user query. The topic of the pattern, whose frames are well filled in by the query terms, is assigned to the user query as its topic. However a query may be expressed in many ways all leading to same information needs. Additionally, each user may also use a different set of query terms for searching the same information. So the query patterns used here may not be able to include all possible variations of the user query patterns. However, the matching score, between query patterns associated with the topics and the user query terms, is calculated as follows:

$$\text{Score} = \frac{\text{\#of frames filled in with matching query terms}}{\text{\#of query terms}}$$

We list below a few regular expressions for the topics: *Climate*, *Ideal time* and *Accommodation*

⊘ **Climate**

a) (climate|weather|rainfall)* (of|in) (<place>)
b) (climate | weather | temperature) (of|in)* (<place>) (during|in)* (autumn|spring | summer | winter)
c) [cloudy|monsoon] (seasons|time)* (of|in) (<place>)
d) [average](temparature)* [var(ies|ying) | hover(s|ing)]* [from] (<degree_celsius>) [to] (<degree_celsius>) (during)* (autumn | spring | summer | winter)
e) (<place>)(weather|monsoon)(forecast|ing)* [during|in] <month|season>

⊘ **Ideal time**

a) (time | season)*(to | for)* (visit | (tour | stay(| ing))) [in] (<place>)
b) [Best | good] (season) to (see | visit | tour) (< place>) (from) (<Month | dates>) (to)* (<Month|dates>)
c) (ideal | pleasant)* (time | season)* for (sight seeing | outings | tourist activities) [in | around] (<place>)
d) ([perfect | best] time)*) to (feel | enjoy)* (chilly | warm | rainy)* [climate] for (honeymooners | trekkers | family tours)* in (<place>)
e) which is (|not)* the (best|peak|festival)* (time | season) to (view | see | visit) (<place>) from (<place>)

⊘ **Accommodation**

a) [luxury | cheap | best] (< Hotel(|s) | Guest houses | hostel(| s)>) to (stay | lodge) in (<place(|s)>)
b) (accomodation(| s) | hotel) available near (<place>) during (<month | season>)
c) (room(|s) | dormitory) for (<count>) (person|day)(|s)
d) (lodge | resort | dharamshala | hotel | accomodation)(|s) near (<place>)
e) [star | budget | cheap] (hotel(|s))(to | for) stay(|ing) in (<place>) near (<place>)

These regular expressions are used to find the pattern of the underlying query terms whose topic is decided by the most matching query pattern. For example, the query – **weather** in **goa** during **may** – belongs to the topic: "Climate"; another query – mighty **forts** of chhatrapati **shivaji** in **maharastra** – belongs to the topics: "Attractions" and / or "History".

4.3.3. Naive Bayes approach

We have manually identified the list of web queries, each tagged with its specific topic related to the tourism domain. This tagged set of queries is used as the training data and the query topic classifier model is built using Naive Bayes method with 10-fold cross validation. This classification model is then used to predict the topic of new user queries. The output will be the query tagged with the tourism specific topic(s) having the highest matching score(s).

Using any one of the above ways, the query topic classifier is built. Then this classifier is added in the

proposed architecture as a subsystem and used to identify the topic(s) of the new user query during an online process.

4.4. The proposed IR system

The proposed IR system consists of the following components: *Content extraction with topic identification*, *Indexer* and *Searcher*. From web documents, textual descriptions are extracted by eliminating noisy contents. Then this extracted textual content is split into segments and the topic of each segment is identified. The Indexer is used to store the identified topics in the index. Then the searcher, on receiving a user query, performs the retrieval of documents by matching the query and document topics. We will now describe each component in detail:

4.4.1. Content extraction with topic identification

Web documents contain noisy contents like advertisements, banners, forms, apart from the core textual content. Many web documents use different layouts with HTML markups and their own style sheets. Most web pages are semi-structured or ill-structured. So at first, we create a tree like document structure using the underlying markups of a web document. By traversing through the nodes in the well formed document structure, advertisements, images, banners, forms, etc. are identified by their tags and filtered out. Among the rest of nodes, some may contain only navigational links and we filter out all such nodes.

Next the document structure is suitably split into different segments using div class tags and table tags as nodes. Then each node content is validated with the heuristic: *link-to-text* ratio, which is defined as the ratio between the size of the text tagged with and without hyperlinks. The text segments which exceed this threshold will be extracted by starting at div or table nodes. The filtered content is converted into the unicode, if not already. The filtered text segments are then sent to the document topic classifier.

The filtered text segments may have one or more overlapping topics pertaining to the tourism ontology. The document topic classifier built in Section 4.2 is invoked on each extracted segment and its topic is identified. The identified topic of the extracted web content is stored in the XML structure and indexed subsequently as an offline process using the indexing system powered by Lucene.[2]

[2]Lucene: www.apache.org/dist/lucene/java/

4.4.2. Indexer

The filtered text segments are fetched from the parsed contents of the web documents and converted into a lucene document object with suitablly chosen fields that could be either stored or indexed or both. The information organized in each field looks like an attribute-value pair. Then the lucene document is indexed and used for faster retrieval. This process is done in offline.

Algorithm 1 The searcher of the proposed IR system

Input:

A query having n terms: $Q = \{t_1, t_2, \ldots, t_n\}, n > 0$
A query topic classifier
Index – Documents indexed with their topics

Description:

1: On receiving a user query, invoke the query topic classifier and identify the most matching topic(s) (with its matching score) of that query
2: Reformulate the user query with the identified topic(s) and feed to the search engine
3: The search engine retrieves an initial set of documents each with its *doc_score*
4: Apply re-ranking of the retrieved documents as follows: consider each document d in the retrieved list; If the document topic(s) matches with the query topic(s), then obtain the document weighting factor w using:

$$w = \sum_t \alpha * score(d, t) \tag{1}$$

where α is the weighting parameter; $score(d, t)$ – score of the document d belonging to the topic t. Then combine w with the document score to get the updated *doc_score*:

$$doc\_score(q, d) = doc\_score(q, d) + w \tag{2}$$

5: Generate the final list of documents based on the updated *doc_score* sorted in the decreasing order
6: **return** top k documents as search results

Output: The ranked list of top k retrieved documents

4.4.3. Searcher

The searcher is the core module in the proposed document retrieval system. It uses the topic information of the query and documents to assist the retrieval task in the tourism domain. At first, on receiving the user query, we invoke the query topic classifier and get the reformulated query – the query with the identified topic(s). Then using this reformulated query, the initial list of documents is retrieved by computing $doc\_score(q, d)$ (as in Eq. 2) – the cosine similarity score between q and d; the doc_score of the retrieved document, whose topic matches with the topic of the query, is updated with the document weighting factor w; and then the final list is generated by the updated doc_score. The document weighting factor w (as given in Eq. 1) is computed as the product of the preference factor α and $score(d, t)$ over the top m topics taken into account, where the preference factor α is the parameter to support the documents with more / less scores independent of the topic information (here we assume $\alpha = 0.85$); $score(d, t)$ is the classifier score for the topic t of the retrieved document and $score(d, t) \in [0, 1]$. Thus the system retrieves an initial list of documents whose similarity scores are further enhanced with the topic(s) associated document weighting factor. Then the final ranked list of top k documents pertaining to the topic of the user query is presented as the search results.

The online processing module associated with the searcher is pictorially represented by the component labeled as (3) in Fig. 1. The proposed document retrieval procedure is given in Algorithm 1. The effectiveness of the proposed topic assisted document retrieval method is given in the next section.

5. Experimental results

5.1. Dataset and queries

We have collected 288 query patterns in the tourism domain from different sources like *Yahoo! questions*, *travel blogs*, *web forums* and *web users*. The collected queries belong to specific tourism related topics and additionally we have included most of the possible variations to each topic. We used these query patterns to build a query topic classifier. We tagged 458 tourism related text documents with one or more identified topics as listed in Section 4.1. This collection has been used to build the document topic classifier. Finally for the evaluation of the proposed document retrieval system, we have used a corpus of 20,482 documents related to tourism domain and used the stemmed queries for evaluation. We have extracted multiple, noise-free, text segments from each document in the corpus and used a very limited resource to build the document classifier. Since the number of text segments extracted is nearly 5 times higher than the total number of documents on an average, we have limited our experiment with this collection of documents. We have also manually collected the topicwise list of keywords to build the keyword based classifier.

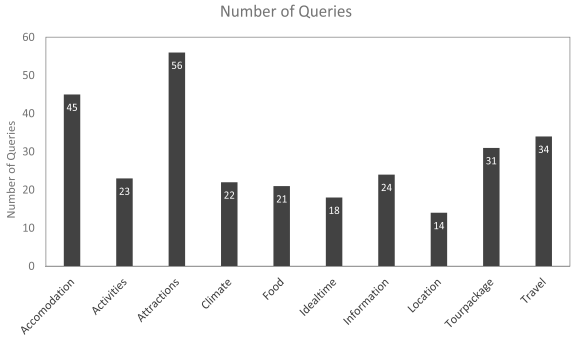

Fig. 2. Statistics on the number of queries.

First we present the experimental results obtained for query classification. Figure 2 shows the number of queries manually collected for each topic. A study of the collected queries in the tourism domain shows that the main interest of the users lies in querying for the content mostly related to the following topics: *attractions*, *accommodation* and *travel (how to reach)*.

Using the number of tourism related queries / patterns collected and the tagged web data, we have built 3 types of classifiers(and their accuracy is given in brackets) based on: i) *topic keywords matching* (0.46) ii) pattern matching (0.53) and iii) Naive Bayes (0.65) approach. Among these 3 approaches, Naive Bayes classifier performs better on an average as it has been applied to the collection of text documents tagged with tourism specific topics. Naives Bayes method assumes the documents as the *Bag of Words* (BoW) in which the terms are assumed to be independent of each other. Figure 3 shows topicwise query classification accuracy.

5.1.1. Document classification

This section presents the experimental results of document classification using Naive Bayes Approach. We have used Weka[3] APIs (Application Programming

[3]Weka APIs: www.cs.waikato.ac.nz/ml/weka/

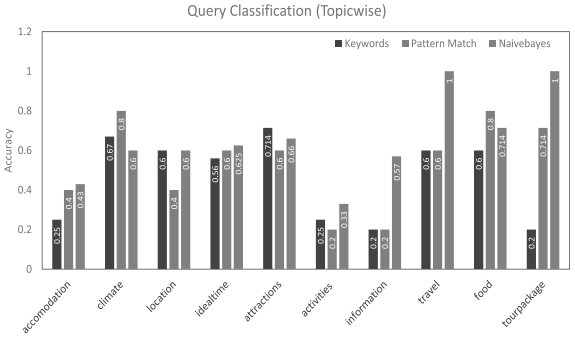

Fig. 3. Query Classification Accuracy (Topicwise).

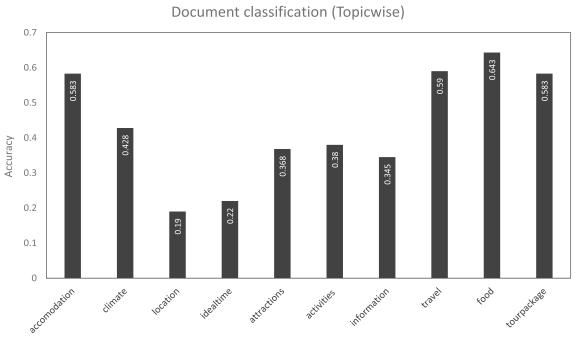

Fig. 4. Classification accuracy [topicwise] with Naive Bayes Algorithm.

Interfaces) to build the Naive Bayes classifier on the tagged collection of 458 tourism related web documents. The document topic classification model is built by considering 60% data for training and 40% data for testing. The classifier is built once with this training data and subsequently be used to classify the topic of the newly extracted web content. We list the distribution of documents in the tourism domain (# of documents in each topic): Accommodation (53), Activities (46), Attractions (50), Climate (41), Food (46), Idealtime (36), History (70), Location (37), Tour Package (39), Travel (40). Figure 4 shows the accuracy of the documents classifier with Naive Bayes on tourism data.

In order to evaluate the proposed document retrieval method, we have conducted the experiments in a similar way like TREC[4] evaluations. We have considered the actual user queries having 10 types of user information needs in the tourism domain and evaluated the proposed document retrieval method with these 10 queries.

[4]Text REtrieval Conference @ http://trec.nist.gov/

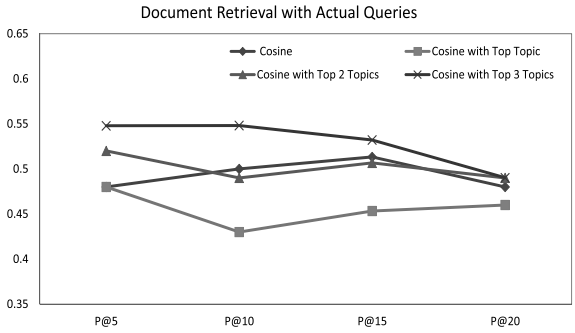

Fig. 5. Retrieval scores comparison of top k documents.

We have used the Vector Space Model(VSM) [18] supported in Lucene as the base line (with no topic related information). In this experiment, given a query, we compute the matching score using cosine similarity between query and documents and then rank them by decreasing order of their similarity score. Documents with the highest similarity score are obtained and evaluated based on how many relevant documents are present among the retrieved top k documents.

5.2. Documents retrieval assisted by topics

We have taken the VSM with information on topic(s) of query and documents. The results were compared against the standard VSM results. We have initially considered the most matching topic of the query to retrieve documents pertaining to the same topic. As some of the queries belong to more than one topic, we have made a variation to the retrieval method by including the topics of the next most matching query topic for assisting the document retrieval. Then all retrieved documents were taken into account and based on their cosine similarity score, the ranked list of documents is presented to the user. The top 5, 10, 15 and 20 documents were evaluated manually for each of 10 queries and computed the retrieval scores in terms of precision @ top k documents.

Figure 5 shows the retrieval scores of documents for the selected 10 queries pertaining to the tourism domain. We have shown the Averaged Precision @ top k (= 5, 10, 15, 20) documents.

Let us discuss some examples which describe the effects of using top 3 classes: consider the query: “Amritsar and the golden temple”. For this query, the query classifier returns the top 3 topics as: *activities*, *accomodation*, *attractions*. The retrieved set of documents consists of news documents covering information related to the activities in and around Amritsar and the

Table 1
Queries with the identified top 3 topics

User query	Top 3 topics
Toy train of Shimla	*location*, *travel*, *information*
Mumbai Ganesh festival	*activities*, *information*, *food*
Sunderbans national park and its tourist	*attractions*, *information*, *activities*
Special and common Tamil food idly dosai	*food*, *location*, *tourpackage*
Goa & its beautiful sun drenched beaches	*activities*, *attractions*, *accomodation*
Konark sun temple & its architecture	*activities*, *location*, *tourpackage*
Meghalaya and its virgin beauty	*activities*, *information*, *accomodation*
Trekking expedition in Darjeeling	*activities*, *attractions*, *travel*

Golden Temple. Here the user intent is to know the general information like "places to see", "hotels to stay", in and around the city of Amritsar. Similarly, for the query, "Water rafting in Rishikesh", the retrieved documents contain information related to various sport activities in the location Rishikesh. Additionally, Rishikesh is a popular tourist place and hence "attractions" related information also identified in top 10 documents. In Table 1, We have listed a few more queries and the identified topics that are included for effective document retrieval. The top most topic identified for the query may be not a suitable topic and the precision goes down when documents are retrived only with this topic. Combining top 3 topics helps the retrieval of documents with diverse information representing different aspects of the actual user intent.

Consider the query: "cheap comfort way to ooty". The query topic classifier identifies the topics (the topics are given in the decreasing order of their topic scores): *Accommodation*; *tour packages*; *travel*; *food*, *attractions*. The terms "cheap" and "comfort" are mostly used to describe "budget hotels" / "cheap / economical hotels" / "cheap guest houses" in tourism related web pages. Since the query topic classifier identifies the topic "how to reach" with low topic score, the documents having travel related information are not boosted sufficiently. With one topic, more documents pertaining to the topic *accommodation* are boosted to bring them in top 5 places. The precision goes down below the base line method. This may be due to two facts: the *query topic classifier accuracy* which directly influences the boosting of topic associated information and the *topic score* which is multiplied with the document score and directly proportional to the final document score. Instead of using one topic, we have considered top 3 topics and analysed the retrieved documents. Each of the topic score is used to boost the document score and then combined to get the final document score. The top 10 retrieved results were analyzed based on this. Among these 10 results, 5 documents contain *how to reach* related information, 1 document contains *tour packages* related information and rest of them fall in *accommodations* and *attractions*. In overall observations, we have noticed that $p@5$ for the actual queries gives 14.1458% improvement over the base line retrieval. However for $p@10$, recall is achieved at the loss of precision which has come down to 9.62% improvement over the standard baseline. Since the accuracy of the query topic identification is marginal, we plan to work on improving the retrieval of documents by improving the performance of the query topic classifier in future. Additionally, we would like to perform scalable experiments with large datasets of TREC and FIRE.

6. Conclusion

We presented a system for document retrieval pertaining to the tourist places in India. We make use of tourism ontology to build the proposed system. The proposed system consists of the document topic classifier which identifies the topics of the given text segment and the query topic classifier to identify the topic of the user query, both pertaining to the tourism domain. Then the tourism specific retrieval engine performs document retrieval by matching the type of user query with the matching type of documents in the index. The experimental results show that the topic identification helps the retrieval of relevant documents at the top of the ranked list. Subsequently, we plan to measure the effectiveness of the proposed method on a larger collection of multilingual web documents pertaining to the tourism domain.

Acknowledgements

A part of this work was supported by Cross Lingual Information Access (CLIA) project funded by Ministry of Communications and Information Technology(MCIT), Government of India. Also the first author would like to thank Dr. Philip O'Reilly of UCC, Ireland for his support in providing time and computing facilities during the revision of this paper.

References

[1] S.M. Beitzel, E.C. Jensen, O. Frieder, D. Grossman, D.D. Lewis, A. Chowdhury and A. Kolcz, Automatic web query classification using labeled and unlabeled training data, in: *Proc. of the 28th ACM SIGIR Conference on Research and Development in IR*, SIGIR '05, ACM, New York, NY, USA, 2005, pp. 581–582.

[2] D. Carmel, E. Yom-Tov, A. Darlow and D. Pelleg, What makes a query difficult? in: *Proc. of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, ACM, New York, NY, USA, 2006, pp. 390–397.

[3] W.W. Cohen and Y. Singer, Context-sensitive learning methods for text categorization, in: *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, ACM, New York, NY, USA, 1996, pp. 307–315.

[4] H. Cui, J.-R. Wen, J.-Y. Nie and W.-Y. Ma, Probabilistic query expansion using query logs, in: *Proc. of the 11th International Conference on World Wide Web*, WWW '02, ACM, New York, NY, USA, 2002, pp. 325–332.

[5] J. D'hondt, P.-A. Verhaegen, J. Vertommen, D. Cattrysse and J.R. Duflou, Topic identification based on document coherence and spectral analysis, *Information Sciences* **181**(18) (2011), 3783–3797.

[6] A. Goker, Context learning in okapi, *Journal of Documentation* **53** (1997), 80–83.

[7] D. He and D. Wu, Enhancing query translation with relevance feedback in translingual information retrieval, *Inf. Process. Manage.* **47** (2011), 1–17.

[8] D.A. Hull, Using structured queries for disambiguation in cross-language information retrieval, in: *Working Notes of AAAI Spring Symposium on Cross-language Text and Speech Retrieval*, 1997, pp. 84–98.

[9] T. Joachims, *Learning to Classify Text Using Support Vector Machines*, Kluwer, 2002.

[10] I.-H. Kang and G. Kim, Query type classification for web document retrieval, in: *Proc. of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, ACM, New York, NY, USA, 2003, pp. 64–71.

[11] B.P. Knijnenburg, M.C. Willemsen, Z. Gantner, H. Soncu and C. Newell, Explaining the user experience of recommender systems, *User Modeling and User-Adapted Interaction*, **22**(4–5) (2012), 441–504.

[12] E. Leopold and J. Kindermann, Text categorization with support vector machines. How to represent texts in input space?, *Mach. Learn.* **46** (2002), 423–444.

[13] C.-J. Lin and P.-H. Chao, Tourism-related opinion mining, in: *ROCLING*, 2010.

[14] H. Cenk Ozmutlu, F. Cavdur and S. Ozmutlu, Cross-validation of neural network applications for automatic new topic identification, *J. Am. Soc. Inf. Sci. Technol.* **59** (2008), 339–362.

[15] H. Cenk Ozmutlu and F. Çavdur, Application of automatic topic identification on excite web search engine data logs, *Inf. Process. Manage.* **41** (2005), 1243–1262.

[16] A. Pirkola, D. Puolamäki and K. Järvelin, Applying query structuring in cross-language retrieval, *Inf. Process. Manage.* **39** (2003), 391–402.

[17] P. Pu, L. Chen and R. Hu, A user-centric evaluation framework for recommender systems, in: *Proc. of the Fifth ACM Conference on Recommender Systems*, RecSys '11, ACM, New York, NY, USA, 2011, pp. 157–164.

[18] G. Salton, A. Wong and C.S. Yang, A vector space model for automatic indexing, *Commun. ACM* **18** (1975), 613–620.

[19] Sebastiani, Machine learning in automated text categorization, *ACM Computing Surveys* **34** (2002), 1–47.

[20] A. Singhal and M. Kaszkiel, A case study in web search using trec algorithms, in: *Proc. of the 10th International Conference on World Wide Web*, WWW '01, ACM, New York, NY, USA, 2001, pp. 708–716.

[21] A. Spink, D. Wolfram, M.B.J. Jansen and T. Saracevic, Searching the web: the public and their queries, *J. Am. Soc. Inf. Sci. Technol.* **52** (2001), 226–234.

[22] S. Talja, H. Keso and T. Pietilainen, The production of context in information seeking research: a metatheoretical view, *Inf. Process. Manage.* **35** (1999), 751–763.

[23] Z. Xiang, K. Wober and D.R. Fesenmaier, Representation of the online tourism domain in search engines, *Journal of Travel Research* **47**(2) (2008), 137–150.

[24] L. Zhang, D. Zhang, S.J. Simoff and J. Debenham, Weighted kernel model for text categorization, in: *Proc. of the Fifth Australasian Conference on Data Mining and Analystics - 61*, AusDM '06, Australian Computer Society, Inc., Darlinghurst, Australia, 2006, pp. 111–114.