Collaborative Exploratory Search in Real-World Context

Naoki Tani
Danushka Bollegala
Shuhei Iitsuka
Yutaka Matsuo
The University of Tokyo,
Hongo 7-3-1, Tokyo
113-8656, Japan
tani@mi.ci.i.u-tokyo.ac.jp
danushka@iba.t.u-tokyo.ac.jp
(iitsuka|matsuo)@biz-model.t.u-tokyo.ac.jp

Naiwala Chandrasiri,
Kazunari Nawa
Toyota InfoTechnology Center
Aksasaka 6-6-20, Tokyo
107-0052, Japan
(np-chandrasiri|nawa)@jp.toyota-itc.com

Keisuke Okamoto
Toyota Motor Corp.
3-3-25 Higashi Ikebukuro
Tokyo, 170-8405, Japan
keisuke@okamoto.tec.toyota.co.jp

ABSTRACT

We propose *Collaborative Exploratory Search (CES)*, which is an integration of dialog analysis and web search that involves multi-party collaboration to accomplish an exploratory information retrieval goal. Given a real-time dialog between users on a single topic; we define CES as the task of automatically detecting the topic of the dialog and retrieving task-relevant web pages to support the dialog. To recognize the task of the dialog, we apply the *Author–Topic* model as a topic model. Then, *attribute extraction* is applied to the dialog to obtain the attributes of the tasks. Finally, a specific search query is generated to identify the task-relevant information. We implement and evaluate the CES system for a commercial in-vehicle conversation. We also develop an iPad application that listens to conversations among users and continuously retrieves relevant web pages. Our experimental results reveal that the proposed method outperforms existing methods, which demonstrates the potential usefulness of collaborative exploratory search with practically usable accuracy levels.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms

Algorithms

Keywords

Personalized and Collaborative Search, Information Extraction

1. INTRODUCTION

During the last decade, web searching has become commonplace in our everyday lives. Currently, most search engines supply query-based searching that is designed to navigate a user to those web pages that are most relevant to the user's query. The use of web searching has become very common in our lives. Therefore, many new scenarios of searching arise.

For example, as cars become smarter, a driver might want to search web pages while driving a car to obtain information related to destination or nearby events. Although current navigation systems cannot not usually access the web, some car manufacturers, such as Toyota, already provide a premium service for customers to search information from the web assisted by a dedicated telephone operator. The following conversation illustrates such a situation.

> *Driver: "Hello. I want to eat something special in Kyoto this evening."*
> *Operator: "OK. What kind of cuisine would you like?"*
> *Driver: "Well..., traditional Japanese."*
> *Operator: "OK, sir. Let me search for traditional Japanese restaurants in Kyoto."*
> *(Operator searches the web.)*
> *Operator: "There is a good restaurant named Kitcho in Kyoto near your location."*
> *Driver: "Is it good?"*
> *(Operator searches the web.)*
> *Operator: "I can not guarantee the taste, but it's been awarded three stars by Michelin."*
> *Driver: "OK. Could you set the location into the GPS?"*
> *Operator: "Certainly. I sent that information to your car."*

In this scenario, the driver has an information need and the operator makes a search, the search is conducted during a dialog, keywords are extracted from the dialog and fed into the search engine, and the search results are provided promptly and help the task in the dialog. In this paper, we specifically address such a situation: multiple persons collaboratively search the web during the conversation. Our system automatically analyzes the discourse, makes a customized query, and shows the results promptly.

As described in this paper, we propose *Collaborative Exploratory Search (CES)*. CES can be considered as the combination of multi-domain exploratory search and collaborative search. It is an integration of dialog analysis and web search that involves multiparty collaboration to accomplish an exploratory information retrieval goal. The goal of CES is to support users' collaboration in searching. Given a real-time dialog between users on a single topic, CES extracts topic and attribute–value pairs of a dialog and seeks relevant web pages automatically. Using real-time dialog as input data, it enables unconscious search and does not block collaboration.

The outline of the proposed method is shown in Figure 1. As shown in Figure 1, we break down the CES into three parts: (i)

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

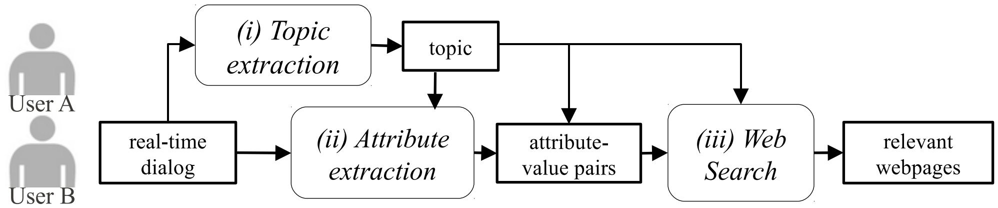

Figure 1: Outline of the proposed system.

topic extraction, (ii) attribute extraction, and (iii) web search. For topic extraction, we apply the *Author–Topic (AT)* model, an unsupervised topic detection algorithm to detect the topic of a dialog between users. For attribute extraction, we apply a two-stage method to obtain attribute–value pairs. Then the customized query is generated and the search results are shown in part (iii). We evaluate our CES approach using an actual dialog dataset by drivers and operators in Toyota. The results show that we can detect the task in a conversation with 50% accuracy in unsupervised topic extraction, and attain a 0.67 F-measure in attribute extraction.

2. RELATED WORK

Exploratory search is defined as the situation in which the user starts from a not-so-well-defined information need and progressively discovers more on his need and on the information available to address it [6]. Kotov et al. [4] proposed a questionguided search, in which a retrieval system would generate potentially interesting questions to the users automatically. The question-guided search system returns "Who is John Kennedy?" or "When was Kennedy sworn as the President of the United States?" if users entered query "john kennedy". Rajaraman [10] produced Kosmix web service[1]: Kosmix is a generalpurpose topic discovery engine, which responds to keyword search using a topic page that summarizes the most relevant information related to the subject associated to the search. Bozzon et al. [3] proposed Liquid Query, a paradigm that exploits the power of underlying search services and provides the user with a multi-domain exploratory search environment (e.g., travels, music, shows, food, movies, and so on). Recent studies have revealed that most people want to search with other people [5, 12]. A survey [7] of 204 people revealed that 53.4% of participants do some collaborative search tasks: travel planning, shopping, literature search, technical information, and so on. For example, TeamSearch [9] is a tabletop application that enables small, co-located groups to search for digital photographs from a metadata-tagged repository. S3 [8] allows users to share useful sites found during a web search asynchronously by representing search results in a persistent file format that can be sent to and augmented by several people. Amershi et al. introduced CoSearch [1], a system that is intended to improve the experience of co-located collaborative web search by leveraging readily available devices such as mobile phones and extra mice.

3. PROBLEM DEFINITION

Let us consider the situation where two agents, A and B, are having a dialog D about a particular topic T. Dialog D includes some attributes V. D can be represented as a vector (hereinafter designated as a dialog vector)

$$d = (a_i, b_j, ...)^T$$

where a_i and b_j are sequences of words (may or may not be complete sentences) uttered respectively by A and B. Elements in d are sorted in the order they are uttered by A and B. Only relative ordering of utterances is considered, thereby ignoring the absolute time information. Given a dialog vector d, we define collaborative exploratory search (CES) as the problem of retrieving a set of web documents L, that is relevant to the topic T and attributes V discussed in the dialog D. To develop a CES system we must perform three tasks: (a) extract the topic being discussed by the speakers in the dialog (Section 4), (b) extract attributes and their values regarding the main topic discussed in the dialog (Section 5), and (c) form queries using the extracted attributes and their values to domain-specific search engines to retrieve information to support the conversation (Section 6).

[1]http://www.kosmix.com/

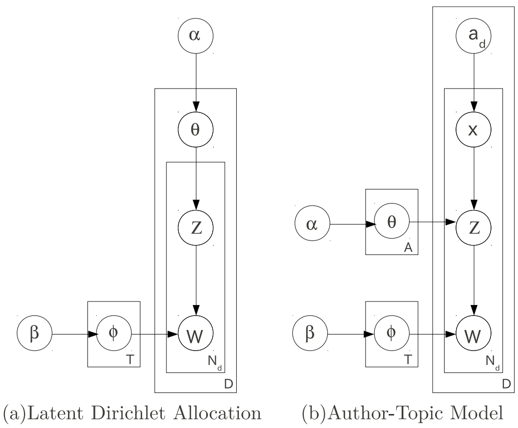

Figure 2: Topic models: LDA and AT.

4. TOPIC EXTRACTION

We briefly review the Latent Dirichlet Allocation (LDA) model and Author–Topic (AT) model, which we use in our system. LDA is a Bayesian network that generates a document using a mixture of topics [2]. In its generative process, for each document d, a multinomial distribution θ_d over topics is randomly sampled from a Dirichlet distribution with parameter α; then, to generate each word, a topic z_{di} is chosen from this topic distribution, and a word, w_{di}, is generated by random sampling from a topic-specific multinomial distribution $\phi_{z_{di}}$. Figure 2 (a) shows the graphical model for LDA.

Despite its simplicity and numerous applications, LDA cannot capture the authorship of documents. For example, multiple authors might contribute to a particular document and we might want to extract topics considering both word distribution as well the authors who wrote those words in a document. The Author–Topic (AT) model [11] extends LDA by incorporating author information for topic extraction. Figure 2 (b) shows an AT model. Specifically, each author is associated with a multinomial distribution over topics, represented by θ. Each topic is associated with a multinomial distribution over words, represented by ϕ. The multinomial distributions θ and ϕ have a symmetric Dirichlet prior with hyperparameters α and β. For each word in a document, we sample an author x uniformly from a_d, then sample a topic z from the multinomial distribution θ associated with author x and sample a word w from a multinomial topic distribution ϕ associated with topic z. This sampling process is repeated N_d times to generate a document d (here, N_d is the total number of words in document d). The AT model has two sets of unknown parameters – A author-topic distributions θ, and T topic distributions ϕ – as well as the latent variables corresponding to the assignments of individual words to topics z and authors x. It has been shown that parameter estimation via Expectation-Maximization (EM) algorithm engenders local maxima and is computationally inefficient in topic models. Consequently, Gibbs sampling has been used to estimate the parameters in AT models. Once we have estimated the parameters θ and ϕ, we can use those values to compute the posterior probability of $p(t|D)$, to assign a topic to a given document D. We model a multi-party dialog using AT model as follows. Each author in the AT model is replaced by a speaker who participates in a dialog. A document in the AT model is replaced by a single dialog. Moreover, the words that comprise a document are replaced by the utterances in the corresponding dialog. By modeling a multi-party dialog as an AT model, we leverage on existing work on topic models to solve the collaborative explorative search problem.

5. ATTRIBUTE EXTRACTION

Once we have identified the topic of the conversation, we must extract keywords related to that topic to retrieve information to support the conversation from a Web search engine. We extract attributes and values for the main topic as keywords. For example, if the topic is about a hotel reservation, then we extract attribute values such as the location of the hotel, the number of people staying, the check-in date, the check-out date, desired price range, Extracting attributes from texts has received much attention lately as a result of the increasing demand in information extraction systems. Watanabe et al. proposed a two-stage attribute extraction method [13], which we use in our CES system. First, given a dialog D, we mark all occurrences that are likely to be values of some attributes. We designate such occurrences as *candidates*. The candidate identification is done using a named entity recognizing tool and a gazetteer. Entity types such as personal names, locations, and organizations are selected as candidate attributes. In addition, we create lists of awards, universities, nationalities, etc. and use those lists in a gazetteer to mark candidate attributes. Having input from both the named entity recognition tool as well as the gazetteer enables an attribute extraction system to recognize a large number of attributes in texts. In the second stage, from the set of identified candidates we select the ones that are likely to be attributes relevant to the topic of the dialog. For this purpose we employ a distance-based heuristic. Specifically, given a list of candidate attributes and their occurrences in a text, we compute the distance between each attribute and the topic of the document. Candidates that are in close proximity to the topic of the document are selected as attributes for that document.

6. MULTI-DOMAIN SEARCH

After identifying the topic in a dialog and extracting appropriate keywords for that topic, we retrieve information relevant to the topic using domain specific search engines. Here, the input is the topic and attribute-value pairs extracted from the target dialog, and the output is a list of web pages which is most relevant to the task, which satisfies the requirement of attribute-value pairs. We obtain relevant information from not only surface web but also from the deep web using many web service APIs. We can thereby obtain the largest range of information and can mutually compare the results obtained using several web services. For example, the system gets HOTEL dialog among users and extracts attribute–value pairs such as (place, "Tokyo"), (heads, 2), (price, \$100). In this case, the system searches for a hotel in Tokyo with a twin-bed room for \$100 or less. In order for a user to compare results, web pages are selected widely from various perspectives. For instance, if users are searching for a restaurant, then we show the most popular restaurant and high-class restaurants. Users can choose from them depending on their tastes.

Table 1: Basic statistics for each task in Toyota dialog dataset.

task	#dialogs	duration	#of turns	#of utterances
HOTEL	30	158 s	110	611
NAVIGATION	53	154 s	67	354
RESTAURANT	73	148 s	91	500
SHOP	20	143 s	71	381
PARKING	18	131 s	69	397
CALL	17	156 s	87	502

7. EXPERIMENTS

We evaluate the performance of CES in this section. As shown in Figure 1, CES are decomposed into three tasks: topic extraction, attribute extraction, and web search. We conduct two sets of experiments to evaluate our CES system. First, we measure the performance of topic extraction. Then we evaluate the attribute extraction method and show how many attributes are correctly chosen.

We use a dialog dataset provided by Toyota Motor Corp. The dataset includes records of conversations between a driver and an operator: A call center operator provides a concierge service to drivers and provides information services such as recommendation and reservation of restaurants, hotels, or shops, and providing a phone number or the location of landmarks and so on. The data include a speaker, utterances obtained by speech recognition plus manual fixing, and time. For privacy reasons, the names of drivers and operators are anonymized. We target Japanese real-time dialog, although our method does not depend on any specific language. The Toyota dialog dataset includes 604 dialogs and 47,322 utterances, consisting of 356,618 words. We extract attribute-value pairs and divide dialogs into six tasks as shown in Table 1. The RESTAURANT and NAVIGATION tasks are the most numerous. Drivers often want to know the locations of places, or want to search for restaurants. Although the durations of the dialogs are not so varied among tasks, the average number of turns presents some tendencies: HOTEL and RESTAURANT needs more turns than others. Drivers must clarify their preferences and a higher degree of interaction might be necessary.

We first measure the performance of topic extraction. The task is to classify the dialog into one of the six task types. We devide 604 dialogs into 404 training data and 200 test data. We annotated the task type of each dialog manually. We use pLDA[2] as a baseline with the number of topics T=10. LDA parameters are set to $\alpha = 0.1$ and $\beta = 0.01$. Table 3 presents the performance of AT model, used in our CES system, and a baseline. Overall, the precision and recall of AT are higher than those of LDA. Table 2 portrays a confusion matrix of classification. The upper part is the result of LDA; the lower part is the result of AT model. Apparently, LDA classifies dialog mostly to NAVIGATION and RESTAURANT, whereas AT model classifies into all tasks properly. From the AT result, we can find that HOTEL and RESTAURANT tasks are confusing because both

[2]http://code.google.com/p/plda/

Table 2: Confusion matrix of the topic extraction. Legend: H (HOTEL), N (NAVIGATION), R (RESTAURANT), S (SHOP), P (PARKING), C (CALL).

(a) LDA: Baseline

	predicted label						
true label	H	N	R	S	P	C	Recall
H	**0**	21	2	3	1	0	0.00
N	0	**43**	1	2	5	0	0.84
R	0	43	**14**	7	6	0	0.20
S	0	12	3	**3**	0	0	0.17
P	0	2	3	2	**2**	0	0.22
C	0	5	0	0	0	**0**	0.00
Precision	0.00	0.34	0.60	0.18	0.14	0.00	

(b) AT model: Proposed method

	predicted label						
true label	H	N	R	S	P	C	Recall
S	**16**	3	4	2	0	3	0.57
N	3	**23**	9	9	0	8	0.44
R	18	6	**25**	11	0	6	0.38
S	2	1	4	**9**	0	1	0.53
P	0	1	4	2	**2**	4	0.15
C	2	1	3	0	0	**4**	0.40
Precision	0.39	0.66	0.51	0.27	1.00	0.15	

Table 3: Performance of the proposed method and baselines.

Method	Precision	Recall	F-score
LDA	0.21	0.24	0.22
AT (proposed)	**0.50**	**0.41**	**0.45**

include conversations about the location, number of people, prices, and reservation. Next, we evaluate attribute extraction after obtaining the task of each dialog. We use 200 labeled dialogs as test data. We compare five methods: AE-clueword, AE-clueword-driver, AE-clueword-operator, AE-driver, and AE-operator. The AE-clueword is the proposed method, which extracts attribute values based on clue words. The AE-clueword-driver extracts information only from the utterances from a driver, and AE-clueword-operator extracts information only from those of an operator. The AE-driver and AE-operator extract attribute values using candidates of attribute values. Each chooses randomly one candidate value from either the driver's utterances or from the operator's utterances.

Table 4 presents results of the experiment. The proposed clue word methods score high. In dialog, an attribute and an attribute value often appear as an adjacency pair, so clue words to represent attributes are efficient to find attribute values for each task. Consequently, the AE-clueword-driver and AE-driver outperform AE-clueword-operator and AE-operator. It is apparent that a user's utterances are likely to include attribute values rather than an operator's because CES in this context is mainly intended to fulfill the drivers information needs. Overall, the proposed methods on topic extraction and attribute extraction outperform the baseline methods, demonstrating the effectiveness of our approach. Experimental results show that the CES model performs better than LDA in terms of both precision and recall on the topic extraction task. In the Toyota dialog dataset, there are two parties: an operator and a driver. Modeling those two speakers in the AT model enables us to detect the topic of the dialog accurately. Although there exist standard benchmark test collections such as TREC evaluation benchmark for Web information retrieval systems and standard evaluation measures such as average precision or normalized discounted cumulative gain (NDCG), evaluating a collaborative search system

Table 4: Performance of the proposed method and baselines in the attribute extraction task.

Method	Precision	Recall	F-score
AE-clueword (proposed)	**0.65**	**0.73**	**0.68**
AE-clueword-driver	0.63	0.71	0.67
AE-clueword-operator	0.58	0.65	0.61
AE-driver	0.52	0.59	0.55
AE-operator	0.50	0.56	0.53

is a challenging task because of the lack of any gold standard for the collaborative search task or evaluation measures that take into account the collaborative nature of the task. Developing test collections and evaluation measures are important future research directions in this field. To underscore the effectiveness of the CES model, we developed a prototypical CES application for use on iPad.

8. CONCLUSION

We proposed collaborative exploratory search (CES), which involves multiparty collaboration to accomplish an exploratory information retrieval goal. Experimental results from a topic extraction and attributes extraction task showed that the proposed method outperforms baselines.

9. REFERENCES

[1] S. Amershi and M. R. Morris. Cosearch: A system for co-located collaborative web search. *CHI'08*, April 2008.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1022, 2003.

[3] A. Bozzon, M. Brambilla, S. Ceri, and P. Fraternali. Liquid query: Multi-domain exploratory search on the web. *WWW'10*, pages 161–170, April 2010.

[4] A. Kotov and C. Zhai. Towards natural question-guided search. *WWW'10*, pages 26–30, April 2010.

[5] A. Large, J. Beheshti, and T. Rahman. Gender differences in collaborative web searching behavior: an elementary school study. *Information Processing and Management*, 38:427–443, 2002.

[6] G. Marchionini. Exploratory search: from finding to understanding. *Communications ACM*, 49(4):41–46, 2006.

[7] M. R. Morris. A survey of collaborative web search practices. *In proc. of CHI*, pages 1657–1660, April 2008.

[8] M. R. Morris and E. Horvitz. S3: Storable, shareable search. *Interact*, pages 120–123, 2007.

[9] M. R. Morris, A. Paepcke, and T. Winograd. Teamsearch: Comparing techniques for co-present collaborative search of digital media. *TABLETOP' 06*, pages 97–104, 2006.

[10] A. Rajaraman. Kosmix: Highperformance topic exploration using the deep web. *VLDB'09*, August 2009.

[11] M. Rosen-Zvi, T. Griffths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. *UAI'04*, 2004.

[12] M. B. Twidale, D. M. Nichols, and C. D. Paice. Browsing in a collaborative process. *Information Processing and Management*, 33(6):761–783, 1997.

[13] K. Watanabe, D. Bollegala, Y. Matsuo, and M. Ishizuka. A two-step approach to extracting atrtibutes for people on the web. In *Proc. Second Web People Search Task (WePS2) Workshop*, 2009.