Identifying Real-Life Complex Task Names with Task-Intrinsic Entities from Microblogs

Ting-Xuan Wang* and **Kun-Yu Tsai** and **Wen-Hsiang Lu**
National Cheng Kung University
Tainan, Taiwan
{P78981320,P76014460,whlu}@mail.ncku.edu.tw

Abstract

Recently, users who search on the web are targeting to more complex tasks due to the explosive growth of web usage. To accomplish a complex task, users may need to obtain information of various entities. For example, a user who wants to travel to Beijing, should book a flight, reserve a hotel room, and survey a Beijing map. A complex task thus needs to submit several queries in order to seeking each of entities. Understanding complex tasks can allow a search engine to suggest related entities and help users explicitly assign their ongoing tasks.

1 Introduction

The requirement of searching for complex tasks dramatically increases in current web search. Users not always search for single information need (Liao et al., 2012). To accomplish a real-life complex task, users usually need to obtain various information of distinct entities on the web. In this paper, we define the necessary entities for a complex task as task-intrinsic entities. For example, a complex task “travel to Beijing” has at least three task-intrinsic entities, including a flight ticket, hotel room, and maps. Therefore, users need submit several queries in order to seek all of the necessary entities. However, conventional search engines are careless of latent complex tasks behind a search query. Users are guided to search for each task-intrinsic entity one by one to accomplish their complex task inefficiently.

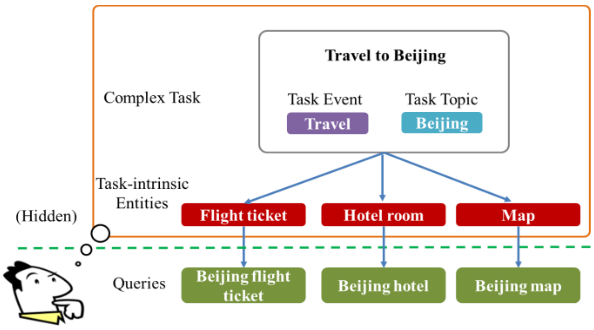

Figure 1. The structure of a complex task with task-intrinsic entities and related queries.

Figure 1 shows a complex task consisting of a task name “travel to Beijing” and several task-intrinsic entities. A task name is composed of a task event and a task topic. The task event triggers users to perform exploratory or comparative search behaviors such as “prepare something”, “buy something” or “travel to somewhere”. The task topic is the subject of interest in the complex task. Task-intrinsic entities are intrinsically demanded by the complex task. The three queries “Beijing flight ticket”, “Beijing hotel”, and “Beijing map” are driven by the information need of each of task-intrinsic entities with topic “Beijing” and event “travel” for the hidden complex task “travel to Beijing”.

According to our observation, users may describe details of a complex task to be done or already completed via microblogs, e.g., Twitter or Weibo[1]. Microblogs are a miniature version of traditional weblogs. In recent years, many users post and share their life details with others on microblogs every day. Due to the post length limitation (only 140 characters in case of Weibo), users tend to only describe key points. Table 1 shows an example of a microblog. We can find that the user, who has an ongoing complex task “北京旅遊(travel to Beijing)”, mentioned two task-intrinsic entities “機票(flight ticket)” and “飯店(hotel)”.

In this work, we address the problem of how to help users efficiently accomplish a complex task when submitting a single query or multiple queries.

[1] Weibo: http://weibo.com

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 470–475,
Baltimore, Maryland, USA, June 23-25 2014. ©2014 Association for Computational Linguistics

Chinese
今天已經**訂好機票**，只剩下**找間飯店**，就等著下禮拜去**北京旅遊**了~好期待!
English Translation
I have already booked a **flight** today, and I only have to find a **hotel**. I'm about to **travel to Beijing** next week - good anticipation!

Table 1. A microblog post from Weibo mentioning an ongoing complex task “北京旅遊 (travel to Beijing)”

We divide the problem into the following three major sub-problems.

1. Find task-intrinsic entities for the complex task.
2. Generate a task name for the complex task.
3. Suggest proper search results covering all desired entities for the complex task.

The above three problems are very important but non-trivial to solve. In this preliminary work, we only focus on first two sub-problems. We proposed an entity-driven complex task model (ECTM) to automatically generate complex task names and related task-intrinsic entities. To evaluate our proposed ECTM, we conducted experiments on a large dataset of real-world query logs. The experimental results show that our ECTM is able to identify a comprehensive complex task name with the task-intrinsic entities and help users accomplish the complex task with less effort.

2 Related Work

Recent studies show that about 75% of search sessions searching for complex tasks (Feild and Allan, 2013). To help users deal with their complex search tasks, researchers devoted their efforts to understand and identify complex tasks from search sessions. Boldi et al. (2002) proposed a graph-based approach to dividing a long-term search session into search tasks. Guo and Agichtein (2010) made the attempt to investigate the hierarchical structure of a complex task with a series of search actions based on search sessions. Cui et al. (2011) proposed random walk based methods to discover search tasks from search sessions. Kotov et al. (2011) noticed that a multi-goal task may require a user to issue a series of queries, spanning a long period of time and multiple search sessions. Thus, they addressed the problem of modeling and analyzing complex cross-session search tasks. Lucchese et al. (2011) tried to identify task-based sessions in query logs by semantic-based features extracted from Wiktionary and Wikipedia to overcome lack of semantic information. Ji et al. (2011) proposed a graph-based regularization algorithm to predict popular search tasks and simultaneously classify queries and web pages by building two content-based classifiers. White et al. (2013) improved the traditional personalization methods for search-result re-ranking by exploiting similar tasks from other users to re-rank search results. Wang et al. (2013) addressed the problem of extracting cross session tasks and proposed a task partition algorithm based on several pairwise similarity features. Raman et al. (2013) investigated intrinsic diversity (ID) for a search task and proposed a re-ranking algorithm according to the ID tasks.

A complex task consists of several sub-tasks, and each sub-task goal may be composed of a sequence of search queries. Therefore, modeling the sub-tasks is necessary for identifying a complex task. Klinkner (2008) proposed a classification-based method to divide a single search session into tasks and sub-tasks based on the four types of features, including time, word, query log sequence, and web search. Lin et al. (2012) defined a search goal as an action-entity pair and utilized web trigram to generate fine-grained search goals. Agichetin et al. (2012) conducted a comprehensive analysis of search tasks and classified them based on several aspects, such as intent, motivation, complexity, work-or-fun, time-sensitive, and continued-or-not. Jones and Yamamoto et al. (2012) proposed an approach to mining sub-tasks for a task using query clustering based on bid phrases provided by advertisers. The most important difference between our work and previous works is that we further try to generate task names with related task-intrinsic entities. To the best of our knowledge, there is no existing approach to utilizing microblogs in dealing with task identification and generating human-interpretable names.

3 Entity-driven Complex Task Model

3.1 Problem Formulation

Given a query q, we aim to identify the complex task for the query. Since the single query is not able to describe a complex task. Our proposed ECTM model introduces an expanded query set $\boldsymbol{Q}_t$ for helping identify the task t. Thus, $\mathrm{P}(t|q)$ can be formulated as follows:

$$\mathrm{P}(t|q) = \sum_{\boldsymbol{Q}_t} \mathrm{P}(\boldsymbol{Q}_t|q)\mathrm{P}(t|\boldsymbol{Q}_t, q) \qquad (1)$$

Since the expanded query set $\boldsymbol{Q}_t$ always contain the input query q, the Equation (1) can thus be approximated as:

$$\mathrm{P}(t|q) = \sum_{\boldsymbol{Q}_t} \mathrm{P}(\boldsymbol{Q}_t|q)\mathrm{P}(t|\boldsymbol{Q}_t), \quad (2)$$

where $\mathrm{P}(\boldsymbol{Q}_t|q)$ is the query expansion model. For $\mathrm{P}(t|\boldsymbol{Q}_t)$ we utilize a set of microblog posts $\boldsymbol{m}$ for identifying the complex task t and obtain the following equation:

$$\mathrm{P}(t|\boldsymbol{Q}_t) = \sum_{\boldsymbol{m}} \mathrm{P}(\boldsymbol{m}|\boldsymbol{Q}_t)\mathrm{P}(t|\boldsymbol{m},\boldsymbol{Q}_t). \quad (3)$$

For $\mathrm{P}(t|\boldsymbol{m},\boldsymbol{Q}_t)$ in Equation (3), the query set $\boldsymbol{Q}_t$ can be omitted since the microblog post set $\boldsymbol{m}$ contains $\boldsymbol{Q}_t$. The Equation (3) can thus be modified as follows:

$$\mathrm{P}(t|\boldsymbol{Q}_t) = \sum_{\boldsymbol{m}} \mathrm{P}(\boldsymbol{m}|\boldsymbol{Q}_t)\mathrm{P}(t|\boldsymbol{m}). \quad (4)$$

Finally, the ECTM can be obtained as follows:

$$\mathrm{P}(t|q) = \sum_{\boldsymbol{Q}_t} \mathrm{P}(\boldsymbol{Q}_t|q) \sum_{\boldsymbol{m}} \mathrm{P}(\boldsymbol{m}|\boldsymbol{Q}_t)\mathrm{P}(t|\boldsymbol{m}), \quad (5)$$

where $\mathrm{P}(\boldsymbol{Q}_t|q)$ is the query expansion model, $\mathrm{P}(\boldsymbol{m}|\boldsymbol{Q}_t)$ is microblog retrieval model, and $\mathrm{P}(t|\boldsymbol{m})$ is task identification model. In the following section, we will describe the three models in detail respectively.

3.2 Query Expansion Model

In fact, only using a single query is insufficient to identify the latent complex task. We thus try to extract task-coherent queries from search sessions. According to our observation, users may persistently search for the same complex task in a period of time. However, users may also simultaneously interleave search for multiple different tasks (MacKay and Watters, 2008; Liu and Belkin, 2010). Therefore, identifying task-coherent queries from search sessions is an important issue. We perform the following processes in order to extract task-coherent queries.

Given a query log and an input query q, we first separate queries in the log into search sessions with the time gap of 30 minutes. We extract search sessions containing the input query q and thus obtain a set of sessions $\boldsymbol{S}_q$. To extract task-coherent queries $\boldsymbol{Q}_t$ from the session set $\boldsymbol{S}_q$, we employ log-linear model (LLM) with the following three useful features:

Average Query Frequency: In most cases, the frequency of queries can reflect their importance. To avoid a long session resulting in high query frequency, we calculate the normalized query frequency as:

$$f_{AQFrequency}(q_t) = \frac{1}{|\boldsymbol{S}_{q_t}|} \times \sum_{s \in \boldsymbol{S}_{q_t}} \frac{freq(q_t, s)}{|s|}, \quad (6)$$

where $freq(q_t, s)$ is the frequency of the query q_t in session s, $\boldsymbol{S}_{q_t}$ is the sessions containing q_t, $|s|$ is the number of queries in session s, and $|\boldsymbol{S}_{q_t}|$ is the number of sessions containing query q_t in the set $\boldsymbol{S}_{q_t}$.

Session Coverage: The queries occurring in several sessions are possible candidates in terms of task-coherence. In order to favor queries occurring in many sessions, we use average session frequency, which can be calculated as follows:

$$f_{ASFrequency}(q_t) = \exp\left(\frac{|\boldsymbol{S}_{q_t}|}{|\boldsymbol{S}_q|}\right), \quad (7)$$

where $|\boldsymbol{S}_q|$ is the number of sessions containing the input query q in the set $\boldsymbol{S}_q$, $|\boldsymbol{S}_{q_t}|$ is the number of sessions containing query q_t in the set $\boldsymbol{S}_{q_t}$, and $\exp(\cdot)$ is the exponential function.

Average Query Distance: Since queries which close to the input query in a search session may have high task-coherence for the latent complex task. We thus use normal distribution to estimate the task-coherence for each query:

$$f_{AQDistance}(q_t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d^2}{2\sigma^2}}, \quad (8)$$

where σ is standard deviation (is empirically set 0.2 in this work), d is the average number of queries between q_t and input query q in sessions.

We employ log-linear model to calculate the probability of each candidate task-coherent query based on the features described above:

$$P(q_t; \boldsymbol{W}) = \frac{\exp(\sum_{i=1}^{|F|} w_i f_i(q_t))}{Z(\boldsymbol{Q}_t)}, \quad (9)$$

where $\boldsymbol{Q}_t$ is the set of all candidate queries in the session set $\boldsymbol{S}_q$, $|F|$ is the number of used feature functions $f_i(q_t)$, $\boldsymbol{W}$ is the set of weighting parameters w_i of feature functions, and $Z(\boldsymbol{Q}_t)$ is a normalizing factor set to the value $Z(\boldsymbol{Q}_t) = \sum_{q_t \in \boldsymbol{Q}_t} \exp(\sum_{i=1}^{|F|} w_i f_i(q_t))$.

3.3 Microblog Retrieval Model

Since the task names are not always observable in the expanded query set $\boldsymbol{Q}_t$, we thus need further expanding $\boldsymbol{Q}_t$ by retrieving microblog posts. The basic idea is that a microblog post containing all queries in $\boldsymbol{Q}_t$ may also contain the task name (see the example in Table 1). In fact, the queries in the query set $\boldsymbol{Q}_t$ usually consist of a topic name and a task-intrinsic entity. For example a query "北京機票(Beijing flight ticket)" contains a topic "北京(Beijing)" and an entity "機票(flight ticket)". Therefore, we first try to extract task-intrinsic entities from the query set $\boldsymbol{Q}_t$ by extracting all common nouns in each of queries. We can thus obtain a list of task-intrinsic entities $\boldsymbol{E}_t$ ordered by the occurrence frequency of each entity. Since a microblog post may only contain a part of entities for a complex task, we generate pseudo queries based on all subsets containing two or three entities from top-*n* entities of $\boldsymbol{E}_t$. Finally, we use all generated pseudo queries to retrieve microblog posts.

3.4 Task Identification Model

To identify a suitable task name from retrieved microblog posts, there are two steps in this model, including candidate task name extraction and correct task name determination.

Candidate Task Name Extraction

For each retrieved microblog post, we first extract all bigrams and trigrams which match the POS (part of speech) patterns listed in Table 2. According to our observation, the POS of a task topic is usually a proper noun (N_p) and the POS of a task event is usually a transitive verb (V_t) + common noun (N_c) or an intransitive verb (V_i). On the other hand, a task topic may be the most important term in related search sessions $\boldsymbol{S}$. More specifically, the term with the POS of proper noun and the highest occurrence count in the $\boldsymbol{Q}_t$. We thus consider the term as a candidate topic (notated as <T>) and adopt two related task POS patterns, i.e., V_t + <T> + N_c and <T> + V_i.

Topic POS	Event POS	Task POS Pattern
N_p	$V_t + N_c$	$V_t + N_p + N_c$
	V_i	$N_p + V_i$
<T>	$V_t + N_c$	V_t + <T> + N_c
	V_i	<T>+ V_i

Table 2. Adopted POS patterns for extracting candidate task names from microblog posts.

Correct Task Name Determination

Different from long-text documents (e.g., webpages), microblog posts are relatively short and hard to find features based on special sections in content (e.g., anchor text, title, or blocks). Therefore, we use five efficient features proposed by Zeng et al. (2004) to extract complex task names from short-text snippets, such as microblog post or search-result snippets. The features proposed by Zeng et al. including TFIDF, phrase length, intra-cluster similarity, cluster entropy, and phrase independence. Furthermore, in this work, we plus two practical features *task name coverage* (the percentage of microblog posts containing the candidate task name) and *chi-square score* (Manning, 1999).

Based on the set of extracted candidate task names $\boldsymbol{T}_q$ for the input query q, we also utilized LLM to select the potential task names with the highest likelihood. The LLM for identifying complex task names is given as follows:

$$P(t;\boldsymbol{\Gamma}) = \frac{\exp(\sum_{j=1}^{|K|} \gamma_j k_j(t))}{Z(\boldsymbol{T}_q)}, \qquad (10)$$

where $\boldsymbol{\Gamma}$ is the set of weighting parameters γ_j of feature functions $k_j(t)$, $|K|$ is the number of feature functions $k_j(t)$, $Z(\boldsymbol{T}_q)$ is a normalizing factor set to $\sum_{t\in\boldsymbol{T}_q} \exp(\sum_{j=1}^{|K|} \gamma_j k_j(t))$.

4 Experiments

4.1 Data

We use a one-month query logs from the Sogou search engine, which contains 21,422,773 records and 3,163,170 distinct queries. Each record contains user ID, query, clicked URL, user clicked order for the query, and the search-result rank of the clicked URL. We group query records into sessions according to user ID. Since a complex search task may take a long time to accomplish, we used one week as the time gap to split sessions, and finally obtained 264,360 sessions. For microblogs, we collected the top 50 posts for each pseudo query from Weibo.

To evaluate the performance of our proposed ECTM model, we manually selected 30 testing queries from sessions which are searching for complex tasks. For each query, we employ three annotators to label complex task names. Three annotators independently annotated 30 queries. We further examined the labeled results, and unified the similar task names. For instances, "北京旅遊 (travel to Beijing)" and "北京旅行 (trip to Beijing)" were be unified to "北京旅遊 (travel to Beijing)". Table 3 shows an example of testing query with labeled task name and task-intrinsic entities.

Query	Labeled Task Name	Labeled Task-Intrinsic Entities
Chinese		
北京旅行社	北京旅遊	地圖, 天氣, 飯店 機票, 行程表
English Translation		
Beijing travel agency	travel to Beijing	map, weather, hotel ,flight tickets, schedule

Table 3. An example query "北京旅行社 (Beijing travel agency)" with labeled task name and task-intrinsic entities.

4.2 Compared Methods

We compare our approach with the state-of-the art phrase extraction approach from short-text snippet (e.g., microblog posts or search result snippets):

- **Cluster_Q_RS (baseline)**: The method is proposed by Zeng et al. (2004), which try to identify important phrases from search result snippets. They proposed five features including TFIDF, phrase length, intra-cluster similarity, cluster entropy, and phrase independence.
- **Cluster_EQ_RS**: Since the above method only aim to identify important phrases from a single query, the result should be not fair for the problem addressed in this work. We try to enhance Cluster_Q_RS using expanded search-result snippets proposed in this work.
- **ECTM_RS**: This method further use our suggested POS patterns for extracting candidate task names and use all features proposed in Section 3.4.2.
- **ECTM_MB**: The only difference between this method and the above method is that the method try to identify task names from microblog posts.

4.3 Parameter Selection

The weights of feature functions are learned by five-fold cross-validation based on our labeled data. We use the same weights for the all of following experiments. Furthermore, determining the number of task-intrinsic entities used in generating pseudo queries is most critical in this work. We show the top *n* average coverage rate and average precision of extracted entities for our 30 testing queries in Figure 2.

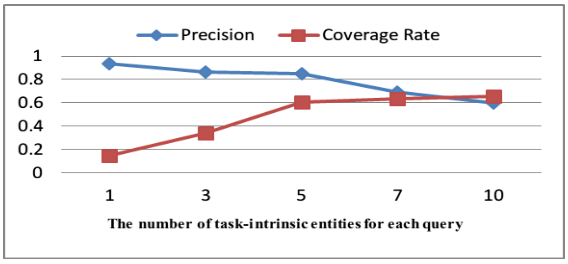

Figure 2. The precision and coverage rate of top *n* entities used in our microblog retrieval model

We found that using top 5 task-intrinsic entities can achieve the best results. Therefore, for each query, we will generate 20 (i.e., $C_2^5 + C_3^5$) pseudo queries and we retrieved top 10 microblog posts for each pseudo queries (totally 200 posts for each testing query).

4.4 Results of Task Name Identification

We use average top ***k*** inclusion rate as the metrics. For a set of queries, its top ***k*** inclusion rate is defined as the percentage of the query set whose correct task names occur in the first ***k*** identified task names. The overall results are shown in Table 4. We can see that our ECTM_MB outperform other methods. The ECTM_MB can identify correct task names within the first three recommendations. Unsurprisingly, Cluster_Q_RS achieved worst inclusion rate. The reason is that Cluster _Q_RS try to find comprehensive complex task name based on search results from only a single query. Most of task names suggested by Cluster_Q_RS are simple task names i.e., the sub-tasks for the latent complex task, such as "預訂機票(book flight tickets)". For ECTM_RS, which is a variation of Cluster_EQ_RS, it achieved slightly better performance by adding the restrictions of POS patterns for extracting candidate task names. Since some identified task names in Cluster_EQ_RS may not semantically suitable, ECTM_RS's approach can efficiently deal with this problem. Furthermore, we also found that using search-result snippets may generate worse task names than using microblog posts. According to our investigating on the two types of the short-text-snippet resources, the search-result snippets are very diverse and task-extrinsic while microblog posts are task-coherent in describing real-life tasks.

Top *k* inclusion rate	**Top1**	**Top3**	**Top5**	**Top10**
Cluster_Q_RS	0.28	0.33	0.37	0.47
Cluster_EQ_RS	0.40	0.43	0.50	0.73
ECTM_RS	0.43	0.43	0.57	0.83
ECTM_MB	**0.87**	**1**	**1**	**1**

Table 4. The results of compared methods

5 Conclusion

In this work, we proposed an entity-driven complex task model (ECTM), which addressed the problem of improving user experience when searching for a complex task. Experimental results show that ECTM efficiently identifies complex tasks with various task-intrinsic entities. Nevertheless, there are still some problems that need to be solved. In the future, we will try to investigate ranking algorithms for developing a novel complex-task-based search engine, which can deal with queries based on complex tasks in real life.

References

Agichtein, E., White, R. W., Dumais, S. T., and Bennett, P. N. Search, Interrupted: Understanding and Predicting Search Task Continuation. In *Proc. of SIGIR*, 2012.

Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., and Vigna, S. The Query-Flow Graph: Model and Applications. In *Proc. of CIKM*, 2008.

Cui, J., Liu, H., Yan, J., Ji L., Jin R., He, J., Gu, Y., Chen, Z., and Du, X. Multi-view Random Walk Framework for Search Task Discovery from Click-through Log. In *Proc. of CIKM*, 2011.

Feild, H. and Allan, J. Task-Aware Query Recommendation. In *Proc. of SIGIR*, 2013.

Guo, Q. and Agichtein, E. Ready to Buy or Just Browsing? Detecting Web Searcher Goals from Interaction Data. In *Proc. of SIGIR*, 2010.

Ji, M., Yan, J., Gu, S., Han, J., He, X., Zhang, W. V., and Chen, Z. Learning Search Tasks in Queries and Web Pages via Graph Regularization. In *Proc. of SIGIR*, 2011.

Jones, R., and Klinkner, K. Beyond the Session Timeout: Automatic Hierarchical Segmentation of Search Topics in Query Logs. In *Proc. of CIKM*, 2008.

Kotov, A., Bennett, P. N., White, R. W., Dumais, S. T., and Teevan, J. Modeling and Analysis of Cross-Session Search Tasks. In *Proc. of SIGIR*, 2011.

Liao, Z., Song, Y., He, L.-W., and Huang, Y. Evaluating the Effectiveness of Search Task Trails. In *Proc. of WWW*, 2012.

Lin, T., Pantel, P., Gamon, M., Kannan, A., and Fuxman, A. Active Objects: Actions for Entity-Centric Search. In *Proc. of WWW*, 2012.

Liu, J. and Belkin, N. J. Personalizing Information Retrieval for Multi-Session Tasks: The Roles of Task Stage and Task Type. In *Proc. of SIGIR*, 2010.

Lucchese, C., Orlando, S., Perego, R., Silvestri, F., and Tolomei, G. Identifying Task-based Sessions in Search Engine Query Logs. In *Proc. of WSDM*, 2011.

MacKay, B. and Watters, C. Exploring Multi-Session Web Tasks. In *Proc. of CHI*, 2008.

Manning, C. D., Schütze, H. Foundations of Statistical Natural Language Processing. The MIT Press. Cambridge, US, 1999.

Raman, K., Bennett, P. N., and Collins-Thompson, K. Toward Whole-Session Relevance: Exploring Intrinsic Diversity in Web Search. In *Proc. of SIGIR*, 2013.

Wang, H., Song, Y., Chang, M.-W., He, X., White, R. W., and Chu, W. Learning to Extract Cross-Session Search Tasks. In *Proc. of WWW*, 2013.

White, R. W., Chu, W., Hassan, A., He, X., Song, Y., and Wang, H. Enhancing Personalized Search by Mining and Modeling Task Behavior. In *Proc. of WWW*, 2013.

Yamamoto, T., Sakai, T., Iwata, M., Yu, C., Wen, J.-R., and Tanaka, K. The Wisdom of Advertisers: Mining Subgoals via Query Clustering. In *Proc. of CIKM*, 2012.

Zeng, H.-J., He, Q.-C., Chen, Z., Ma, W.-Y., and Ma, J. Learning to Cluster Web Search Results. In *Proc. of SIGIR*, 2004.