DEIM Forum 2017 F4-4

Finding and Suggesting Alternative Verbal Queries from Community Q&A Corpus

Suppanut POTHIRATTANACHAIKUL†, Takehiro YAMAMOTO†, Sumio FUJITA††, Akira TAJIMA††, and Katsumi TANAKA†

† Graduate School of Informatics, Kyoto University
†† Yahoo Japan Corporation
E-mail: †{suppanut,tyamamot,tanaka}@dl.kuis.kyoto-u.ac.jp, ††{sufujita,atajima}@yahoo-corp.jp

Abstract Web searchers often use a Web search engine to find a way or means to achieve his/her goal. For example, a user intending to solve his/her sleeping problem, the query "sleeping pills" may be used. However, there may be another solution to achieve the same goal, such as "have a cup of hot milk" or "stroll before bedtime." The problem is that the user may not be aware that these solutions exist. Thus, he/she will probably choose to take a sleeping pill without considering these solutions. In this study, we define and tackle the *alternative action mining* problem. In particular, we attempt to develop a method for mining alternative actions for a given query. We define alternative actions as actions which share the same goal and define the alternative action mining problem as similar in the search result diversification. To tackle the problem, we propose leveraging a community Q&A (cQA) corpus for mining alternative actions. The cQA corpus can be seen as an archival dataset comprising dialogues between questioners, who want to know the solutions to their problem, and respondents, who suggest different solutions. We propose a method to compute how well two actions can be alternative actions by using a question-answer structure in a cQA corpus. Our method builds a question-action bipartite graph and recursively computes how well two actions can be alternative actions. We conducted experiments to investigate the effectiveness of our method using two newly built test collections, each containing 50 queries. The experimental results indicated that our proposed method outperformed the query suggestion methods provided by the commercial search engines in terms of D#-nDCG.

Key words information retrieval, search result diversification, query suggestion

1. Introduction

Web searchers often use a Web search engine to find a way or means to achieve his/her real-world *goal*. For example, a user who is suffering from a sleeping problem may issue the query "sleeping pills," intending to find a good sleeping pill to *solve his/her sleeping problem*. According to a survey on 1,000 Web searchers, reported by Nakamura *et al.* [15], approximately 57.5% of the users answered that one motivation for using Web search engines is to find a way or means to solve their goal. Such Web search has more recently started being referred to as task-oriented Web search [26], and many researchers have started tackling the problem of supporting task-oriented Web search.

In task-oriented search, the searcher faces the problem that he/she may not be aware of another existing solution that could help achieve the same goal behind the query. For example, for the searcher issuing the query "sleeping pills," other solutions such as "have a cup of hot milk" or "stroll before bedtime" exist as well, which can also help resolve the "solve his/her sleeping problem." Since the searcher often believes in the mean he/she initially comes up with, he/she

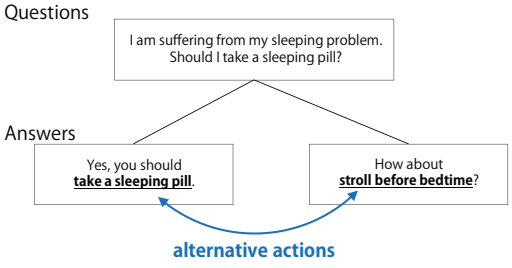

Figure 1 Example question-answer pairs in cQA corpus. Our method extracts alternative actions by using question-answer structure.

may decide to take a sleeping pill without considering the other solutions that can solve the same problem. Although the current search engines provide query suggestions for supporting a searcher to reformulate his/her query, it is hard for the searcher to find alternative solutions.

In this study, we tackle the *alternative action mining* problem, where a system is required to find alternative actions for a given query. An alternative action for a query is defined as an action that can solve the same problem (See Section 3. 2). For example, given the query "sleeping pills," our objective is to find alternative actions such as "have a cup of hot milk" or "stroll before bedtime," both these alternative actions can achieve the same goal behind the query, i.e., "solve the sleeping problem." Mined alternative actions can be utilized for supporting a searcher in a task-oriented Web search. For example, by suggesting the alternative actions to the searcher issuing the query "sleeping pills," he/she is able to notice different solutions and make an improved decision on how to solve his/her sleeping problem.

To tackle the alternative action mining problem, we propose leveraging a community Q&A (cQA) corpus. We hypothesize that the cQA corpus can be seen as an archival dataset comprising dialogues between questioners, who want to know the solutions to their problem, and respondents, who suggest good solutions for it. Figure 1 shows an example of a question-answer pair in a cQA corpus. The fundamental idea of using a cQA corpus is that, as can be seen in the figure, the two actions "take a sleeping pills" and "stroll before bedtime" are proposed by the respondents to satisfy the same goal of a questioner, which means "take a sleeping pills" and "stroll before bedtime" can be alternative actions. We also propose a method for computing how well two actions can be alternative actions using the question-answer structure of a cQA corpus. Our method constructs a question-action bipartite graph from a set of question-answer pairs and recursively computes how well two actions can be alternative actions (See Section 4.).

We prepared two test collections, each containing 50 queries, for our evaluation. The experimental results using the test collections showed that our method outperformed the conventional query suggestions provided by the commercial search engines in terms of D#-nDCG.

The main contributions of this study are as follows: (1) We identified and defined the *alternative action mining* problem. We defined the problem in terms of search result diversification, and provided the definitions regarding the problem to make our work reliable (See Section 3.). To our knowledge, our work is the first to address this problem. (2) We proposed utilizing a cQA corpus to address the problem. We revealed that the questions-answer relationship can be effective for identifying how well two actions can be alternative actions. (3) We prepared the test collections for the alternative action mining problem. Our two test collections, each of which contains 50 queries, are constructed from two different services, which enabling us to investigate the applicability of our method (See Section 5.).

2. Related Work

2. 1 Task-Oriented Web Search

Hassan *et al.* studied on supporting the *complex search task*, in which a searcher has to accomplish several subtasks to satisfy his/her information need [7, 8]. They proposed a method that includes grouping queries into the same task through the query log mining and query syntactic analysis. Jones and Klinkner proposed the *mission-goal* hierarchical relationship [10] between information needs, and proposed a method for classifying a pair of queries into the same mission/goal (referred to as *task* in their study) or not. Although, in the present study, a hierarchical relation is assumed between actions as in these previous studies; our study focuses on the users' real-world behavior rather than on other types of searches such as covering many aspects of a topic.

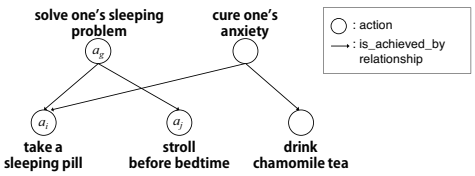

Figure 2 Example structure among actions. In the figure, actions a_i (*take a sleeping pill*) and a_j (*stroll before bedtime*) are called *alternative actions* when they share the same action a_g (*solve one's sleeping problem*) as their goal. Note that the actions "take a sleeping pill" and "drink chamomile tea" are also alternative actions, since they share the other action "cure one's anxiety."

The studies that are most relevant to the present study are those by Yamamoto *et al.* [23] and Yang *et al.* [26]. Yamamoto *et al.* defined the *goal-subgoal* relationship and proposed a method for clustering queries into subgoals by leveraging the sponsored search data. Yang *et al.* defined the *task-subtask* relationship and proposed a method for connecting search queries with task descriptions written in wikiHow(注1). Although they used the different terminologies for defining the hierarchical relationship, both definitions were based on the *is-achieved-by* relationship. In this study, we also use the is-achieved-by relation to define alternative actions.

The key difference between the above studies and ours is that most of the existing studies focused on finding *sub* tasks for a given query. For example, given the query "lose weight," the desired outputs are "do physical exercise" and "control calorie intake," each of which can achieve the query [13].

2. 2 Connecting cQA with Web Search

Liu *et al.* extensively analyzed the behavior logs obtained from a Web search engine and a cQA service, and revealed the typical patterns when a Web searcher gives up his/her search and asks a question in the cQA service. According to their study, a searcher who issues a query containing terms such as "how," "can," or "do," *etc.*, tended to ask a question. Another study showed that one popular type of questions on a cQA service is a how-to question [6]. With regard to these studies, Webner *et al.* focused on a Web search query related to how-to information, and proposed a method for extracting its answer from the cQA corpus [20]. These results suggest that a cQA corpus contains much how-to information, which can be effectively used for mining alternative actions.

3. Problem Definition

In this section, we define the *alternative action mining*

(注1) : http://wikihow.com

problem addressed by the present study. As discussed in the literature [8, 26], different terminologies were used in many of the exiting studies to represent similar concepts, such as *mission-goal* [10], *goal-subgoal* [23] or *task-subtask* [8,26]. In this study, we basically follow the definitions proposed by Yang *et al.* [26], except that the use of the term *action* instead of using *task*. This is done because we focus on a verbal phrase as our retrieval unit.

We first introduce several concepts including the concepts of the action, the is-achived-by relationship and the alternative actions relationship. We then define the alternative action mining problem. Finally, we discuss the relation of our study to the existing studies.

3.1 Alternative Action

Definition 1 (action): An *action* is an activity that a user wants to achieve. In our study, we represent an action as a verbal phrase, as in the work [23]. For example, "take a sleeping pill," and "have a cup of hot milk," are actions.

Definition 2 (is-achieved-by relationship): For two actions a_i and a_g, we call a_g *is-achieved-by* a_i when achieving a_i helps to achieve a_g. Figure 2 illustrates the example actions for the is-achieved-by relationship. In the figure, action a_g is-achieved-by a_i since achieving "take a sleeping pill" helps to achieve "solve one's sleeping problem." For the convenience, in this study, we also call "a_g is a *goal* of a_i", whose meaning is "a_g is-achieved-by a_i."

Definition 3 (alternative actions relationship): For two actions a_i and a_j, we call a_i and a_j are *alternative actions* when they are different actions and there exists at least one other action a_g which is their common goal.

As shown in Figure 2, the actions "take a sleeping pill" and "stroll before bedtime" are alternative actions since they share the same goal "solve one's sleeping problem." Also, note that, actions "take a sleeping pill" and "drink chamomile tea" are also alternative actions since they share the other goal "cure one's anxiety."

3.2 Alternative Action Mining Problem

As introduced in Section 1., our objective is to automatically mine alternative actions for a given query. One thing we have to consider is the ambiguity of the goals behind the query. As shown in Figure 2, the action "take a sleeping pill" can be used to achieve two different goals. Thus, for a searcher who issues the query "sleeping pills," it is hard to predict which goal the searcher wants to achieve, i.e., "solve his/her sleeping problem" or "cure his/her anxiety," and the desired alternative actions depend on it. To solve this ambiguity, we follow an approach similar to the one used in the search result diversification [1, 3, 13, 22], where, for a given query, the system is required to generate a *diversified* ranked list of documents satisfying as many different *search intents* behind the query as possible. The alternative action mining problem is defined as follows:

Alternative action mining problem: Given a query q, the alternative action mining problem refers to returning a diversified ranked list of k alternative actions $a_1, a_2, \ldots, a_k$ for that query, which can satisfy as many different goals of searchers who issue q. In this study, we assume that the query is an action, even if it is not represented as a verbal phrase, and that the searcher focuses on achieving the action represented by the query. For example, for the query "sleeping pills", our objective is to automatically generate a ranked list of actions (e.g., "stroll before bedtime", "drink chamomile tea"), which can satisfy two different goals "solve one's sleeping problem" and "cure one's anxiety," of the action represented by the query "take a sleeping pill."

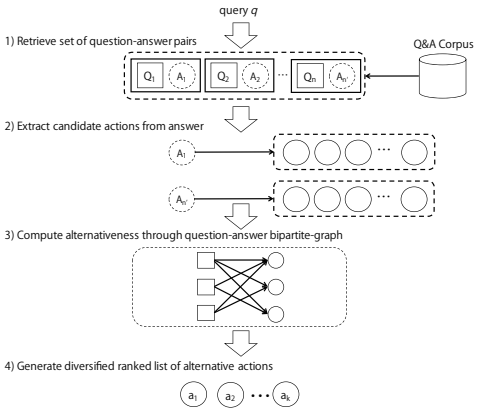

Figure 3 Method overview.

4. Our Approach

In the previous section we defined the alternative mining problem addressed by this study. In this section we explain our proposed method, which utilizes a cQA corpus to automatically find alternative actions to a given query. We first discuss a technical challenge of the problem. We then give the details of the proposed method.

4.1 Technical Challenge

One key challenge of the alternative action mining problem is measuring how well two actions a_i, and a_j are alternative actions. In other words, we have to compute how well the two actions can achieve the same goal. In this study, we refer to the strength of this relationship as *alternativeness* between a_i and a_j, denoted as $\mathrm{alt}(a_i, a_j)$. If we could measure the alternativeness between query q and action a_i, $\mathrm{alt}(q, a_i)$, the rank of the action for the query should be basically determined by its value. The difficulty in measuring the alternativeness is that using textual similarity (e.g., the Levenshtein distance) or semantic similarity (e.g., word embedding [14]) is not effective. For example, the two actions "take a sleeping pill" and "stroll before bedtime" should have high alternativeness although they are neither textually nor semantically similar.

4.2 Method Overview

Given a query q, our objective is to retrieve a ranked list of k alternative actions $a_1, a_2, \ldots a_k$ that satisfy as many goals behind the query as possible. Figure 3 shows the overview of our method. Given a query q, our first step is to retrieve a set of question-answer pairs related to the query from the community Q&A corpus. Then, we extract the candidate actions from the retrieved answers. Our third step, which is the core of our method, is to compute the alternativeness between actions through the question-action bipartite graph. Finally, we apply the search result diversification algorithm to the actions and generate a diversified ranked list of alternative actions for the query.

4.3 Retrieve Question-Answer Pairs

Given a query q, we first retrieve a set of question-answer pairs from a cQA corpus. We hypothesize that some questions are likely to receive many suggestions by the respon-

Table 1 Manually predefined terms for retrieving question-answer pairs.

Terms
effect, should I, disadvantage, try, want to, worst, need to, begin, beginner, why, risk of, prefer, alternative

dents because of their content. For example, when a question contains "*Should I* use sleeping pills?" in its text, its answers will likely to contain many actions other than "take a sleeping pill" because the questioner is unsure about his/her idea. Thus, retrieving such questions may help us find alternative actions from their answers.

To this end, we manually prepare some terms that are likely to indicate that a questioner is unsure about his/her idea. Table 1 lists up the terms we prepared. By using these terms we retrieve questions and answers related to the query. More specifically, we first retrieve answers containing q. We then obtain the questions from these answers. Next, we rank the questions by using the terms show in Table 1 by the Okapi BM25 algorithm and obtain the top n questions ($n = 10,000$ in our experiments). Finally, we retrieve all the answers of these questions, and obtain a set of question-answer pairs for the n questions.

4.4 Extract Candidate Actions from Answers

After we obtain the set of question-answer pairs, we extract candidate actions from their answers. We extract all the verbal phrases from the answers. We apply the standard text-chunking approach using Conditional Random Field (CRF) to extract verbal phrases from the answers. We prepare 500 sentences by sampling answers in the cQA corpus, and annotate the verbal phrases in the sentences. We then learned the classifier which classifies terms in a sentence into a verbal phrase or not. We use the standard 19 features for text chunking [16] including bag-of-words and parts-of-speech of the target word and its surroundings. We apply the learnt classifier to the texts of the answers and extract a set of actions.

4.5 Measure Alternativeness between Actions

Obtaining the set of actions from the previous step, we compute the alternativeness between actions. As we mentioned in Section 4.1, it is hard to compute the alternativeness between two actions simply using the usual textual or semantic similarity. To compute the alternativeness between two actions we make the following two hypotheses: (1) **H1 (question → action):** If two questions are likely to *represent the same goal*, actions in their answers are likely to be *alternative actions*. (2) **H2 (action → question):** If two actions are likely to be *alternative actions*, questions of the answers containing these actions are likely to *represent the same goal*.

Take, as an example, two different questions "*I am suffering from my sleeping problem. What should I do*" and "*How can I sleep well?*" We can expect that answers to these different questions are intended to satisfy the same goal, which we can obtain **H1**. Also, for two answers containing the different actions "take a sleeping pill" and "have a cup of hot milk", we can expect that their questions are likely to address the same problem, which we can obtain **H2**.

Since **H1** and **H2** are recursive – the alternativeness between two actions depends on how likely two questions represent the same goal, and this depends on the alternativeness between actions in their answers – we apply the SimRank algorithm [9], which is designed to compute the similarity between nodes on a graph, on the question-action bipartite graph. We first prepare the question-action bipartite graph from the questions and actions extracted in the previous steps Let $\mathcal{Q} = \{Q_i\}_{i=1}^{n}$ be the set of questions retrieved by the step described in Section 4.3, $A = \{a_j\}_{j=1}^{m}$ be the set of actions extracted in the step described in Section 4.4, and $\mathcal{A} = A \cup \{q\}$ be the union of these actions and query, we construct a bipartite graph $G = (\mathcal{Q} \cup \mathcal{A}, E)$, where $E \subseteqq \mathcal{Q} \times \mathcal{A}$ and edge $e_{ij} = (Q_i, a_j) \in E$ in G represents that action a_j appears in at least one answer of question Q_i.

Let $\mathrm{sim}_{\mathrm{goal}}(Q_i, Q_j)(Q_i, Q_j \in \mathcal{Q})$ represent how well two questions represent the same goal, and $\mathrm{alt}(a_i, a_j)(a_i, a_j \in \mathcal{A})$ represent how well two actions are alternative actions. If $Q_i = Q_j$, the initial value for $\mathrm{sim}_{\mathrm{goal}}(Q_i, Q_j)$ is set to 1, otherwise the initial value for $\mathrm{sim}_{\mathrm{goal}}(Q_i, Q_j)$ is 0. We use the same condition to initialize the value for $\mathrm{alt}(a_i, a_j)$. After we assign the initial values to $\mathrm{sim}_{\mathrm{goal}}(\cdot,\cdot)$ and $\mathrm{alt}(\cdot,\cdot)$, we update these two measures by iteratively computing the following two formulae:

$$\mathrm{sim}_{\mathrm{goal}}(Q_i, Q_j) = \frac{C}{|O(Q_i)||O(Q_j)|} \sum_{k=1}^{|O(Q_i)|} \sum_{l=1}^{|O(Q_j)|} \mathrm{alt}(O_k(Q_i), O_l(Q_j)) \ , \tag{1}$$

$$\mathrm{alt}(a_i, a_j) = \frac{C}{|I(a_i)||I(a_j)|} \sum_{k=1}^{|I(a_i)|} \sum_{l=1}^{|I(a_j)|} \mathrm{sim}_{\mathrm{goal}}(I_k(a_i), I_l(a_j)) \tag{2}$$

where C is a constant value and $O(Q_i)(\subseteqq \mathcal{A})$ is a set of out-neighbors of Q_i and $I(a_i)(\subseteqq \mathcal{Q})$ is a set of in-neighbors of a_i. We use $C = 0.8$ and the values of $\mathrm{alt}(\cdot,\cdot)$ obtained after the five iterations, as suggested in [9].

From the alternativeness between query q and action a_i $\mathrm{alt}(q, a_i)$, we can know that how well an action a_i can be the alternative actions for a query q, which is used to determine the relevance of the action to the query. Moreover, the alternativeness between two actions $\mathrm{alt}(a_i, a_j)$ indicates how well two actions share the same goal; high $\mathrm{alt}(a_i, a_j)$ means they are similar in terms of their goals and low $\mathrm{alt}(a_i, a_j)$ means they are dissimilar. This information can be used for the diversification of the ranked list.

4.6 Measure Effectiveness of Action by Community Evaluation

To improve the performance of measuring the relevance between query and action, we also propose utilizing the quality of answers evaluated by the community in the service. Most cQA services enable their users to evaluate the quality of answers by, for e.g., selecting best answers or up-voting good answers. Our idea is that such evaluations by the community can help find actions that many people believe in their effects.

We compute the effectiveness of action a_i as the probability that an answer containing a_i be selected as a best answer:

$$\mathrm{effect}(a_i) = \frac{|\mathrm{BestAnswers}(a_i)| + \theta}{|\mathrm{Answers}(a_i)| + 2\theta} \ , \tag{3}$$

where $\mathrm{BestAnswers}(a_i)$ and $\mathrm{Answers}(a_i)$ represent the set of best answers and answers in the question-answer pairs retrieved by the step in Section 4.3, respectively, and θ is the Laplace smoothing parameter ($\theta = 8$ in our experiments).

4.7 Generate Diversified Ranked List

Once we compute the alternativeness between actions and their effectiveness, we generate a ranked list of actions. As described in Section 3.2, the purpose of the ranked list is to achieve as many different goals behind the query as possible.

To this end, we apply the result diversification technique to diversify the ranked list.

We apply the Maximal Marginal Relevance (MMR) algorithm [2] to generate the diversified ranked list of actions. MMR iteratively chooses the relevant items considering both the relevance and diversity. Letting $A = \{a_j\}_{j=1}^{m}$ be the set of candidate actions to be ranked, MMR selects a^r, an action ranked at the r-th position using:

$$a^r = \underset{a \in A_q \setminus S^{r-1}}{\arg\max} \left[\lambda \cdot \mathrm{rel}(q, a) - (1 - \lambda) \max_{a' \in S^{r-1}} \mathrm{alt}(a, a') \right] , \tag{4}$$

where

$$\mathrm{rel}(q, a) = \alpha \cdot \mathrm{alt}(q, a) + (1 - \alpha) \cdot \mathrm{effect}(a) \ . \tag{5}$$

S^{r-1} denotes a set of $r-1$ actions that MMR has already selected, λ is a parameter balancing the relevance and the diversity, and α is a parameter balancing the alternativeness and the effectiveness. By applying the MMR algorithm, we obtain the diversified ranked list of k alternative actions for the query.

5. Experimental Setup

In this study, we address the following research questions by conducting the experiments: (1) Does our proposed method outperform the query suggestions provided by the commercial search engines in terms of the providing alternative actions for a query? (2) Can our method work for the cQA corpora on different services? (3) How do the parameters λ, which balance the relevancy and diversity, and α, which combine the alternativeness and effectiveness, affect the performance? (4) For what kinds of queries does our method work effectively? We prepared two test collections for Japanese and English, which we refer to **JaCollection** and **EnCollection**.

5.1 Dataset

We use the corpus archived in Yahoo! Chiebukuro(注2), which is the most popular community Q&A service in Japan, for JaCollection. Table 2 shows the statistics of the Yahoo! Chiebukuro corpus we use in this evaluation. We build the search system on Elasticsearch to retrieve questions and answers from the corpus.

We also use other data in the evaluation to investigate whether our method works on different data. We use the data archived in Reddit(注3) as another cQA corpus for EnCollection. Reddit is one of the most popular online communities, where users communicate by making posts and giving comments to them. Although the purposes of the Reddit users are not only for community-based Q&A, in this study we view Reddit as the community Q&A service; assuming a post made by a user as a question and the comments to the post as its answers. We use the APIs(注4) provided by Reddit to retrieve posts and their comments.

One difference between Yahoo! Chiebukuro and Reddit, which affects our method, is that Yahoo! Chiebukuro allows users to vote for the best answer whereas Reddit does not have option. Instead, Reddit allows users to provide a positive or negative voting to a comment. Thus, when computing Equation (3) for the Reddit data, instead of computing the best answer probability we compute the probability that a comment receives positive votes.

(注2) : http://chiebukuro.yahoo.co.jp/

(注3) : https://www.reddit.com/

(注4) : https://www.reddit.com/dev/api/

Table 2 Data statistics of Yahoo! Chiebukuro corpus.

# of questions	84,123,965
# of answers	224,969,857
Avg. # of answers/question	2.67
Archive period	April 2004 - December 2014

Table 3 Example queries used in experiment (EnCollection).

Domain	**Queries**	
Health	kettlebell workout	acupuncture
	chamomile tea	pilates
Recreation	airbnb	uber taxi
	cheap flight	youth hostel
Education	coursera	vocational school
	public university	free certification

5.2 Proposed and Baseline Methods

To measure the effectiveness of our method, we prepare the following methods:

(1) **Query Suggestion (QS)**: We extract the query suggestions from a Web search engine to investigate whether the current query suggestions provide alternative actions. We use the query suggestions provided by the two commercial search engines, which we refer to as **QS1** and **QS2**, respectively. (2) **RelDivQA**: This is our proposed method which generates a ranked list of actions based on Equations (4) and (5). Equations (4) and (5) contain two parameters λ and α. To fairly compare with the baselines and our method, we use the optimum λ and α for EnCollection when evaluating JaCollection, In addition, we use the ones for JaCollection when evaluating EnCollection. (3) **RelOnlyQA**: This is the same as RelDivQA, except that we set $\lambda = 1.0$. Thus, RelOnlyQA only considers the relevance but not the diversity of the ranked list.

5.3 Test Collection Construction

JaCollection used Yahoo! Chiebukuro and EnCollection used Reddit as the cQA corpus. We first prepare 50 queries to be used as the input to alternative action mining. We chose three domains (Health, Recreation and Education) to select queries. Table 3 shows example queries we use for EnCollection. Both test collections contain 23 queries from health, 14 from recreation and 13 from education.

We view our alternative mining problem as similar to the search result diversification. To evaluate our method, we need the following ground truth: (1) A set of goals $G^q = \{g_1^q, \ldots g_n^q\}$ for query q, where n is the number of goals for q. E.g., for query "sleeping pills," $G = \{$"solve one's sleeping problem", "cure one's anxiety"$\}$. (2) Goal-level action relevance $\mathrm{rel}^q(a, g)$, which represents how well an action a is an alternative action to the query q in terms of achieving its goal g.

In order to prepare goal-level action relevance, three assessors for each language are used in this experiment. We first pool the results of both baseline and proposed methods at the pool depth size at 10. For the proposed method, we generate the ranked result for each combination of the two parameters λ and α, by changing their parameters from 0.1 0.2, ..., to 1.0. Then, for each query, an assessor was asked to annotate goal-level action relevance for the pooled actions.

Table 4 D#-nDCG@k for each test collection (highest values among methods are in bold).

		D#-nDCG@k			
		$k=1$	$k=3$	$k=5$	$k=8$
JaCollection	QS1	0.000	0.000	0.006	0.017
	QS2	0.000	0.000	0.000	0.006
	RelOnlyQA	**0.116**	**0.292**	0.368	0.411
	RelDivQA	**0.116**	0.291	**0.369**	**0.412**
EnCollection	QS1	0.055	0.066	0.092	0.109
	QS2	0.000	0.070	0.087	0.124
	RelOnlyQA	**0.168**	0.218	0.307	**0.415**
	RelDivQA	**0.168**	**0.300**	**0.329**	0.390

The annotation is conducted in the following step. For each (query, goal, action), we ask the assessors to annotate its relevance with three-graded scores according to the following criteria: (1) **highly relevant (2)**: action a strongly helps to achieve goal g, and also a is another solution which differs from the action represented by the query itself. (2) **relevant (1)**: action a may help to achieve goal g and also a is another solution which differs from the action represented by the query itself. (3) **irrelevant (0)**: otherwise. The cases when an action receives a relevance value of 0 are (a) the action does not help to achieve the goal, or (b) achieving the action solves the action indicated by the query, which means the action does not provide any alternative. (e.g., for the query "sleeping pill", "take Xanax" or "Xanax" (Xanax is a popular sleeping pill) was assigned the relevance of 0 since taking a Xanax achieves taking a sleeping pill).

The Fleiss' kappa coefficients [5] for these relevance judgments among three assessors were 0.290 (JaCollection) and 0.565 (EnCollection). The Fleiss' kappa coefficients for the binary relevance judgments were 0.368 (JaCollection), which is a fair agreement, and 0.626 (EnCollection), which is a substantial agreement. Finally, for each goal-level action relevance, we merge the results of three assessors as follows: (1) if two or more assessors give a relevance of 0, then we regard it as a relevance of 0, (2) otherwise, we regard it as a relevance of 1.

5.4 Evaluation Metric

We use D#-nDCG [19], which was proposed by Sakai *et al.*. and has been used in the NTCIR INTENT [18] and IMine [13, 22] tasks. The purpose of D#-nDCG is to intuitively evaluate a ranked-list in terms of both its diversity and relevance. We use D#-nDCG@8 as our primary metric since many of the conventional query suggestions provide eight suggestions to a query. The ranked list containing more relevant and diverse (in terms of q's goals) actions achieves higher D#-nDCG@8.

6. Experimental Results

6.1 Comparison with Baselines

Table 4 shows the results of D#-nDCG@k of the baseline and our methods described in Section 5.2 for two test collections. Here, we use $\lambda = 0.7$ and $\alpha = 0.6$ as parameters, which is the optimum D#-nDCG@8 for EnCollection, for evaluating JaCollection. We also use $\lambda = 0.4$ and $\alpha = 0.5$, which achieves the optimum D#-nDCG@8 for JaCollection, for evaluating EnCollection.

From the table, we can see that both RelDivQA and RelOnlyQA outperform QS1 and QS2 for both test collections. In addition, it can be seen that D#-nDCG of both QS1 and QS2 are quite low, compared with RelDivQA and RelOnlyQA. This result indicates that conventional query suggestions rarely provide alternative actions for a query, whereas cQA is an effective resource for mining alternative actions. Also, having that our methods achieved the similar performance on different test collections, the experimental results suggest that our method is able to applicable to many cQA services.

When we compare the results of RelDivQA and RelOnlyQA, we observe that the results of RelDivQA and RelOnlyQA are similar. The two-sided Randomized Tukey's HSD test [17] revealed that we observed the significant differences between all the pairs of baselines and the proposed methods at the significant level $\alpha = 0.01$, whereas we did not observe any significant difference between RelDivQA and RelOnlyQA for all the metrics on both test collections. This implies that the combination of the relevance and diversity does not always help to improve the performance in our evaluation. One possible reason of this would be that the number of goals were small. As described in Section 5.3, each query has at most three goals, which is relatively small number compared with the existing test collection [18]

Table 5 D#-nDCG@8 of RelDivQA ($\lambda = 0.4$, $\alpha = 0.5$ for JaCollection, $\lambda = 0.7$, $\alpha = 0.6$ for EnCollection) for different domains.

	JaCollection	**EnCollection**
Health	0.566	0.479
Recreation	0.300	0.413
Education	0.235	0.414
ALL	0.414	0.448

6.2 Effect of Domain

To investigate the effectiveness of our method with the different domains, Table 5 summarizes D#-nDCG@8 for three domains. Note that we used the optimum parameters for each test collection when computing D#-nDCG@8. From the table, we can observe that the results of the health domain achieved the best performance for both JaCollection and EnCollection. The possible explanation of this would be, in the health domain, people discuss about many possible solutions for solving their problems since they want to choose the effective and credible solution for their health. We thus could find many alternative actions from the cQA corpus.

6.3 Examples

Table 7 shows examples of the alternative actions retrieved by our method and baselines (QS1 and QS2). For example, for the query "chamomile tea," our method successfully ranked the alternative action "drink a cup of hot milk," which can achieve the goal behind the query "promote falling asleep" at the first rank, while the baselines QS1 and QS2 suggested the queries which specialize the input query (e.g., "chamomile tea effect"). Since the conventional query suggestions are not designed for providing alternative actions for a query, suggesting the alternative actions obtained by our method can complement the existing query suggestions

Table 6 D#-nDCG@8 for different λ and α for both JACollection and EnCollection (highest value in bold).

	JaCollection											EnCollection									
$\alpha\backslash\lambda$	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	$\alpha\backslash\lambda$	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
0.1	0.413	0.413	0.414	0.414	0.414	0.414	0.414	0.414	0.414	0.414	**0.1**	0.308	0.349	0.304	0.326	0.315	0.320	0.333	0.337	0.353	0.347
0.2	0.413	0.413	0.414	0.414	0.414	0.414	0.414	0.414	0.414	0.414	**0.2**	0.295	0.340	0.307	0.308	0.301	0.345	0.361	0.359	0.375	0.370
0.3	0.413	0.413	0.414	0.414	0.414	0.414	0.414	0.414	0.414	0.414	**0.3**	0.321	0.335	0.329	0.348	0.323	0.341	0.383	0.389	0.395	0.360
0.4	0.413	0.413	0.413	0.414	0.414	0.414	0.414	0.414	0.414	0.413	**0.4**	0.322	0.350	0.350	0.365	0.345	0.379	0.415	0.420	0.373	0.392
0.5	0.413	0.413	0.413	**0.414**	0.414	0.414	0.414	0.414	0.414	0.414	**0.5**	0.370	0.377	0.372	0.390	0.387	0.429	0.443	0.426	0.409	0.415
0.6	0.406	0.406	0.406	0.412	0.412	0.412	0.412	0.412	0.412	0.412	**0.6**	0.350	0.351	0.359	0.375	0.389	0.420	**0.448**	0.418	0.412	0.435
0.7	0.402	0.406	0.406	0.411	0.412	0.412	0.412	0.412	0.412	0.411	**0.7**	0.362	0.356	0.353	0.378	0.419	0.429	0.420	0.413	0.401	0.420
0.8	0.402	0.406	0.406	0.406	0.412	0.412	0.412	0.412	0.411	0.411	**0.8**	0.366	0.354	0.345	0.386	0.389	0.414	0.414	0.401	0.412	0.425
0.9	0.402	0.402	0.406	0.406	0.412	0.412	0.411	0.403	0.402	0.402	**0.9**	0.356	0.359	0.346	0.352	0.425	0.399	0.416	0.403	0.408	0.403
1.0	0.398	0.398	0.398	0.398	0.398	0.410	0.410	0.410	0.410	0.410	**1.0**	0.323	0.301	0.297	0.371	0.443	0.399	0.410	0.413	0.419	0.415

and help a searcher make an improved decision on how to achieve his/her goal.

On the other hand, from the table we can see that our method ranked the action "put it on your eyes," which seems a meaningless phrase, at the second rank of the query "chamomile tea." We found that many of the irrelevant actions retrieved by our method were such meaningless verbal phrases, which affected the performance of our method. We will discuss how to improve our method in Section 7..

7. Limitations

Our work has several limitations that we should acknowledge. First, we take a search result diversification approach to generating a ranked list of alternative actions. The problem of this approach is that the ranked list contains actions which achieve different goals and it would be difficult for a searcher to find the alternative actions which achieve *his/her* actual goal. Several possible solution to solve this problem would be clustering the alternative actions according to their goals or predicting the goal of the searcher by analyzing his/her behavior log.

Second, as shown in Table 4, D#-nDCG obtained by our method is relatively low, compared with the standard search result diversification problems [18] Currently our method just extract actions (verbal phrases) from the answers in the cQA corpus and use them as the candidates for the ranked list. The problem is that these actions contain lots of irrelevant actions which cannot be alternative actions for a query. Some researchers proposed to use syntactic patterns to extract target entities from a text corpus [12]. Applying such a method would enable us to extract the candidate alternative actions rather than actions from the cQA corpus.

Lastly, we should acknowledge the belief of a searcher. People usually favor information that confirms their preexisting beliefs and biases [11]. Recently, White also revealed that the existence of the search biases in which users preferred affirmative information to their beliefs [21]. His findings implies that, even if we successfully provide alternative actions to a searcher, he/she would not take them into consideration because he/she believes in the solution expressed by the query. Although it is challenging to change the belief of a searcher, many researchers attempted to support a searcher's credible or careful search, *e.g.*, by suggesting disputed sentences [24], [4] or providing scores according to credibility criteria [25]. By applying such methodologies, we may raise the awareness of the searcher.

8. Conclusions

In this study, we addressed the alternative action mining problem. We defined the alternative mining problem as similar in the search result diversification. To our knowledge, our work is the first to study this problem. Also, we proposed leveraging a cQA corpus to address the alternative action mining problem. Our method iteratively computes two measures; (1) $\mathrm{alt}(\cdot,\cdot)$, which measures the alternativeness between two actions, and (2) $\mathrm{sim}_{\mathrm{goal}}(\cdot,\cdot)$, which measures how well two questions represent the same goal, by applying the SimRank algorithm to the question-action bipartite graph. Our method generates the diversified ranked list of alternative actions by applying the MMR algorithm according to the alternativeness between actions.

The experimental results using our in-house two test collections showed that our method outperformed the conventional query suggestions provided by the commercial search engines in terms of D#-nDCG. We believe that our method can complement the conventional query suggestions and help a searcher make an improved decision on how to achieve his/her goal. We also found that the combination of the alternativeness and the effectiveness improved the performance for the English test collection, whereas we could not find this trend for the Japanese test collection.

As we discussed in Section 7., we have several limitations that affect the performance of our method. One possible direction would be improving the step for extracting candidate actions so that we can obtain more relevant alternative actions. Another interesting direction would be designing a new search interaction so that a search system can encourage a searcher to carefully compare the solutions suggested by the system and the one he/she initially comes up with.

Acknowledgement This work was supported in part by JSPS KAKENHI Grant Numbers 15H01718, 16H02906, 16K16156, Microsoft Research CORE 12 program, and Yahoo Japan Corporation.

References

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proc. of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14, 2009.

[2] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336, 1998.

Table 7 Example top 2 alternative actions retrieved by proposed and baseline methods for queries "chamomile tea" and "youth hostel". Relevant actions are in bold.

	query = "chamomile tea"		
	RelDivQA	**QS1**	**QS2**
1st.	**drink a cup of hot milk**	chamomile tea effect	chamomile tea effect
2nd.	put it on your eyes	how to make chamomile tea	camomile tea cough
	query = "youth hostel"		
	RelDivQA	**QS1**	**QS2**
1st.	**check out airbnb**	youth hostel dublin	youth hostel paris france
2nd.	stayed at a hostel	youth hostels in london	youth hostel association of india

[3] C. L. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2009 Web track. In *Proc. of the 18th Text REtrieval Conference*, 2009.

[4] R. Ennals, B. Trushkowsky, and J. M. Agosta. Highlighting disputed claims on the web. In *Proc. of the 19th International conference on World Wide Web*, pages 341–350, 2010.

[5] J. L. Fleiss and J. Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, 33(3):613–619, 1973.

[6] F. M. Harper, D. Moy, and J. A. Konstan. Facts or friends?: distinguishing informational and conversational questions in social q&a sites. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 759–768, 2009.

[7] A. Hassan and R. W. White. Task tours: helping users tackle complex search tasks. In *Proc. of the 21st ACM International Conference on Information and Knowledge Management*, pages 1885–1889, 2012.

[8] A. Hassan Awadallah, R. W. White, P. Pantel, S. T. Dumais, and Y.-M. Wang. Supporting complex search tasks. In *Proc. of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 829–838, 2014.

[9] G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. In *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 538–543, 2002.

[10] R. Jones and K. L. Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proc. of the 17th ACM Conference on Information and Knowledge Management*, pages 699–708, 2008.

[11] D. Kahneman. A perspective on judgment and choice: mapping bounded rationality. *American Psychologist*, 58(9):697, 2003.

[12] S. Li, C.-Y. Lin, Y.-I. Song, and Z. Li. Comparable entity mining from comparative questions. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 650–658, 2010.

[13] Y. Liu, R. Song, M. Zhang, Z. Dou, T. Yamamoto, M. P. Kato, H. Ohshima, and K. Zhou. Overview of the ntcir-11 imine task. In *Proc. of the 11th NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, 2014.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[15] S. Nakamura, S. Konishi, A. Jatowt, H. Ohshima, H. Kondo, T. Tezuka, S. Oyama, and K. Tanaka. Trustworthiness analysis of web search results. In *Proc. of the 11th European Conference on Research and Advanced Technology for Digital Libraries*, pages 38–49, 2007.

[16] N. Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs), 2007.

[17] T. Sakai. Metrics, statistics, tests. In *PROMISE Winter School 2013: Bridging between Information Retrieval and Databases*, 2014.

[18] T. Sakai, Z. Dou, T. Yamamoto, Y. Liu, M. Zhang, R. Song, M. Kato, and M. Iwata. Overview of the NTCIR-10 INTENT-2 task. In *Proc. of the 10th NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, 2013.

[19] T. Sakai and R. Song. Evaluating diversified search results using per-intent graded relevance. In *Proc. of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1043–1052, 2011.

[20] I. Weber, A. Ukkonen, and A. Gionis. Answers, not links: extracting tips from Yahoo! answers to address how-to web queries. In *Proc. of the 5th ACM International Conference on Web Search and Data Mining*, pages 613–622, 2012.

[21] R. White. Beliefs and biases in web search. In *Proc. of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 2013.

[22] T. Yamamoto, Y. Liu, M. Zhang, Z. Dou, K. Zhou, I. Markov, M. P. Kato, H. Ohshima, and S. Fujita. Overview of the NTCIR-12 IMine-2 task. In *Proc. of the 12th NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, pages 94–123, 2016.

[23] T. Yamamoto, T. Sakai, M. Iwata, C. Yu, J.-R. Wen, and K. Tanaka. The wisdom of advertisers: mining subgoals via query clustering. In *Proc. of the 21st ACM International Conference on Information and Knowledge Management*, pages 505–514, 2012.

[24] Y. Yamamoto and S. Shimada. Can disputed topic suggestion enhance user consideration of information credibility in web search? In *Proc. of the 27th ACM Conference on Hypertext and Social Media*, pages 169–177, 2016.

[25] Y. Yamamoto and K. Tanaka. Enhancing credibility judgment of web search results. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1235–1244, 2011.

[26] Z. Yang and E. Nyberg. Leveraging procedural knowledge for task-oriented search. In *Proc. of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 513–522, 2015.