Natural Language Engineering (2024), **30**, pp. 577–601
doi:10.1017/S1351324923000219

ARTICLE

Recommending tasks based on search queries and missions

Darío Garigliotti[1], Krisztian Balog[2], Katja Hose[1] and Johannes Bjerva[1]

[1]Department of Computer Science, Aalborg University, (Aalborg$^{\epsilon}$ | Copenhagen$^{\sigma}$), Denmark and [2]Department of Electrical Engineering and Computer Science, University of Stavanger, Stavanger, Norway
Corresponding author: D. Garigliotti; Email: dariog@cs.aau.dk

(Received 17 August 2021; revised 17 March 2023; accepted 17 April 2023; first published online 17 May 2023)

Abstract
Web search is an experience that naturally lends itself to recommendations, including query suggestions and related entities. In this article, we propose to recommend specific tasks to users, based on their search queries, such as planning a holiday trip or organizing a party. Specifically, we introduce the problem of query-based task recommendation and develop methods that combine well-established term-based ranking techniques with continuous semantic representations, including sentence representations from several transformer-based models. Using a purpose-built test collection, we find that our method is able to significantly outperform a strong text-based baseline. Further, we extend our approach to using a set of queries that all share the same underlying task, referred to as search mission, as input. The study is rounded off with a detailed feature and query analysis.

Keywords: Information Retrieval; Machine Learning

1. Introduction

When issuing a query online, people often have an underlying goal that is concerned with the completion of some task, such as planning a holiday trip or organizing a birthday party. This task is often nontrivial and instead can be decomposed into concrete steps. For instance, if the goal is to organize a birthday party, concrete steps might, for example, include *planning the party* or *baking a cake*. Task-based searches actually correspond to a considerable portion of query volume (Donato et al., 2010) and can have a large impact on search behavior (Byström, 2002). The support for more *complex* search tasks, including exploratory, multisearch, multisession, and multistep tasks, has been identified as an area where search engines should provide better assistance to their users (Kelly, Arguello, and Capra, 2013; Hassan Awadallah et al., 2014b). Indeed, current search engines support a small set of basic tasks, and most of the knowledge-intensive workload for supporting more complex tasks is left to the user (Choi et al., 2021). The ultimate challenge is to build useful systems "to achieve work task completion" (Toms, Villa, and Mccay-Peet, 2013) and to assist the users in "outlining the necessary steps to accomplish a complex task" (Hassan Awadallah et al., 2014b). In this article, we propose to recommend specific tasks to users, based on their search queries. An illustrative scenario is depicted in Figure 1, where an envisaged interface answers the user query "birthday party" by directly providing a list of recommended tasks.

Web search is an experience that naturally lends itself to recommendations, including providing query suggestions (Szpektor, Gionis, and Maarek, 2011; Bonchi et al., 2012; Ozertem et al., 2012; Dehghani et al., 2017) and finding-related entities (Blanco et al., 2013; Bi et al. 2015;

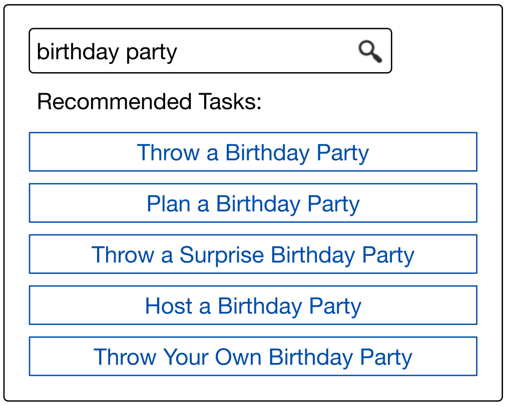


Figure 1. Query-based task recommendations.

Fernández-Tobías and Blanco, 2016). We introduce the problem of query-based task recommendation: returning a list of recommendable tasks for a given task-based search query. In a scenario such as the one depicted in Figure 1, the results ranked for the prompt search query can in turn be used to provide recommendations to the user by triggering a dedicated panel in the envisioned interface. One peculiarity of this problem is the lack of dedicated training data, which hinders the applicability of collaborative filtering and representation learning approaches when based directly on such data. While a small number of query-task pairs may be obtained as training instances by manual labeling (which is indeed what we also do in this study), this is inherently limited in scale. Given the practically unconstrained space of possible tasks and queries, for any given query, this essentially remains a cold start problem. Therefore, we approach task recommendation as a content-based recommendation problem, which boils down to the challenge of scoring tasks in response to input queries. For that reason, we start with well-established term-based ranking approaches and combine them with semantic matching signals developed for the task at hand. Beyond a single input query, we are also interested in the scenario of a set of queries that all share the same underlying task, referred to as search mission. Hence, we introduce a second problem: mission-based task recommendation. In this article, we aim to answer the following research question: *How can we recommend tasks based on search queries and missions?*

One of the main contributions of this study addresses the lack of a standard test collection to evaluate our novel problem. Building on a dataset of detected search missions developed by Hagen et al. (2013), we create and release the first test collection to assess task recommendation. The collection comprises 59 queries, annotated via crowdsourcing with relevance assessments for tasks in a repository of WikiHow articles. Another central contribution is establishing baseline methods for the query-based and mission-based task recommendation problems. We propose a query-based task recommendation approach, by combining well-established term-based matching methods with semantic matching signals based on continuous representations. Further, we extend our approach to deal with an input search mission. Task recommendation results in a search component that can be applied on top of any method proposed for detecting queries that belong to the same underlying task (Jones and Klinkner, 2008; Lucchese et al., 2011; Hagen et al., 2013; Lucchese et al., 2013, 2016).

In summary, these are the main contributions of this work:

- we introduce the problems of query-based task recommendation and mission-based task recommendation,
- we create and release the first test collection for assessing task recommendation,

- we establish baseline methods for the query-based and mission-based task recommendation problems, and
- we conduct an in-depth quantitative and qualitative analysis of our experimental results.

The resources developed within this work are made publicly available[a] under CC-BY-SA 4.0 license.

This article is structured as follows. After reviewing related work (Section 2), in Section 3, we formally introduce the two research problems we are addressing in this paper: query-based and mission-based task recommendation. Section 4 then outlines our proposed solutions to these problems while Section 5 details our efforts to build a test collection, consisting of a task repository, a set of search missions, representative "best" queries for each mission, and corresponding relevance assessments. Section 6 then presents and discusses our experimental results including a detailed feature and query analysis with insights concerning the importance and effectiveness of the various semantic signals. Finally, Section 7 concludes the paper.

2. Related work

In this section, we first discuss a body of related work on search sessions and missions. Then, we study research literature on procedural knowledge. Throughout the section, we complement these discussions by motivating the decisions behind our working notion of search mission, our operationalization of the concept of task, and our conceptualization of the problems introduced in this article as instances of recommendation for search queries and missions.

2.1 Search queries, sessions, and missions

The seminal high-level categorization of search behavior by Broder (2002), classifying the queries in navigational, informational, or transactional, has been subsequently extended as more online search activities are identified (Rose and Levinson, 2004; Marchionini, 2006; Jansen, Booth, and Spink, 2008). Of particular interest for our work is the multi-step search task (Hassan Awadallah et al., 2014a), which encompasses the behavior where a complex task, involving multiple particularities, aims to be fulfilled. A large body of work targets the detection of search sessions in query logs, with the ultimate aim of identifying search patterns in user behavior and of providing additional support on them. In the field of query log analysis, a *search session* is usually defined as a sequence of queries where the elapsed time between two consecutive queries is no larger than a delta value, which ranges from 30 min (He, Göker, and Harper, 2002; Radlinski and Joachims, 2005) to 90 min (Hagen et al., 2013). Beyond session detection, focus was also placed on the challenge of understanding the task-based user interaction with search engines. Here, "task" refers to the overall goal ("work task") of the user that motivated her to search in the first place. A notion of *search mission* is established by Jones and Klinkner (2008). They introduce the concept of hierarchical search, instituting the idea of a search mission as a set of topically related queries, not necessarily consecutive, composed by simpler search goals. Their work also discusses a method for detecting whether a given query pair belongs to the same goal. We adhere to their concept of a search mission, to emphasize the coherent nature of search queries motivated by a single underlying goal, as opposed to other lines of work that focus on multitasking behavior across sessions. Moreover, and for the sake of simplicity, we relax the definition of mission by discarding the notion of sequence and only considering a search mission as an unordered set of queries all with the same underlying task.

Several works have extended the problem of detecting search missions, and proposed approaches accordingly. In the work by Mei et al. (2009), a method is proposed for finding

[a]https://github.com/dariogarigliotti/task-recommendation

user search sequences at multiple levels of granularity within the hierarchical search approach. Furthermore, Donato et al. (2010) address on-the-fly identification of search tasks, that is as the user interacts with an engine. Intent-aware query similarity is introduced by Guo et al. (2011), aiming to capture the search intent behind a query and using these underlying tasks for recommending related queries. Kelly et al. (2013) propose a supervised model to analyze user search behavior spanning multiple sessions. They address two problems: identifying related queries for a query in previous sessions by that user, and predicting whether the user will return to a given multiquery task. The problem of mission detection framed as search task discovery within a user session is introduced in the work by Lucchese et al. (2011), where the authors also propose a method based on graph query representation. Hagen et al. (2013) extend task detection to the cross-session modality, using a cascading approach, to account for the user behavior of carrying on a search task across multiple sessions. Further approaches in cross-session search task detection include modeling intrinsic and extrinsic diversity (Raman, Bennett, and Collins-Thompson, 2013), temporal closeness (Li et al., 2014), topic membership (Li et al., 2016), and embedded trees in query similarity graph (Wang et al., 2013). Lucchese et al. (2016) propose a framework for finding search tasks combining a Learning-to-Rank-based query similarity and graph-based query clustering. Lugo et al. (2020b) address task-based query log segmentation with a method that does not depend on information such as clicked URLs. The authors also approach search task detection in a user-agnostic, unsupervised fashion (Lugo, Moreno, and Hubert, 2020a). We assume that we have access to a mission detection oracle that correctly identifies a set of queries all belonging to the same (unknown) task. In practical applications, this assumption would correspond to having a component in place that detects search missions, on top of which our task recommendation approach can be performed. The assumption of such a mission oracle in place avoids any potential issues with the errors of the mission detection method when observing task recommendation performances.

Recommendations based on search queries have been studied for specific domains, such as movies or celebrities (Yu et al., 2014; Bi et al., 2015). Such model-based methods typically rely on manually designed domain-dependent features, related to specific properties of entities, and usually deal with queries that consist of names of entities. Instead, we address search queries that express the need for solving an underlying task. We recommend elements from a repository of task descriptions using different attributes in the document representation of a task, such as title or explanation. We argue that our problem of task recommendation based on search queries is a significant conceptual transition from an elementary approach of ad hoc document retrieval. An analogous and very successful transition can be seen in entity retrieval, this is, the problem of obtaining a ranked list of entities relevant to a search query. The INEX 2007–2009 Entity Retrieval track (de Vries et al., 2008; Demartini et al., 2009, 2010) was the first benchmark campaign on studying entity retrieval, using Wikipedia as an entity repository of reference. In other words, the operational representation of entities was based on their corresponding repository of Wikipedia articles. Hence, our operationalization of the abstract notion of a task via a repository of task descriptions to be recommended for search queries resembles the one utilized in the conceptualization of entity retrieval.

2.2 Procedural knowledge

Task recommendations do not necessarily benefit from descriptive or declarative knowledge bases (KBs). Indeed, such KBs do not contain sufficient *procedural* knowledge. Procedural knowledge (Georgeff and Lansky, 1986) is the knowledge involved in accomplishing a task. A few *procedural KBs* are specialized in semi-structured knowledge (Pareti, Klein, and Barker, 2014; Chu, Tandon, and Weikum, 2017), that has been extracted from corpora of how-to guides collaboratively created in projects like WikiHow or eHow.[b] Jung et al. (2010) automatically create a

[b]http://www.ehow.com

situation ontology from how-to instructions through a model of daily situational knowledge with goals, action sequences, and contextual items. They also envisage a schema mapping actions to actual service providers. As mentioned above, for pragmatic reasons, we choose to work with a collection of procedural articles from WikiHow as our repository to recommend tasks from. Pareti et al. (2014) identify properties that define community-centric tasks. They then design a framework to represent the semantic structure of procedural knowledge in web communities. The obtained representation operates distributedly across different KBs and comprises a process ontology to express tasks, subtasks, requirements, and execution status.

Chu et al. (2017) propose to build a procedural KB by normalizing the how-to tasks from WikiHow into a hierarchical taxonomy that also represents the temporal order of subtasks, as well as the attributes of involved items. Their method relies on hierarchical agglomerative clustering of task frames obtained by open-domain semantic role labeling. They show the usefulness of this KB in expanding individual, task-oriented queries for retrieving relevant YouTube videos. Although we address a different problem, namely explicit task recommendation rather than leveraging tasks for recommending other kind of items, we use their dataset of WikiHow articles as our task repository.

Other more recent works approach the procedural nature of WikiHow in a data-driven fashion using representation learning. Problems addressed in these works include the correspondence between a subtask and its main task, approached as a single-class prediction (Zhang, Lyu, and Callison-Burch, 2020a) or as a binary classification (Zhou, Shah, and Schockaert, 2019), and the reasoning about the possible temporal ordering between two subtasks (Zhou et al., 2019; Zhang, Lyu, and Callison-Burch, 2020b). Zhang et al. (2020b) have studied the ability of WikiHow to be a resource for knowledge transfer to other tasks in natural language understanding. To the best of our knowledge, we are the first in using WikiHow as the main resource of task descriptions and building a corresponding dataset suited for the novel problem of task recommendation based on search queries and missions, that we also introduce in this work.

3. Problem statement

In this section, we formally introduce two specific task recommendation problems.

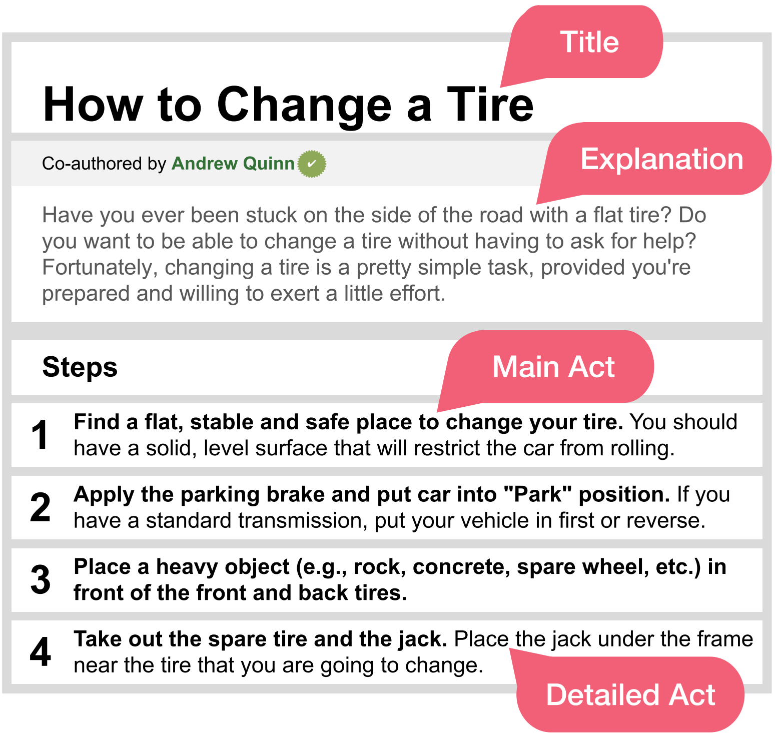
3.1 Query-based task recommendation

The first problem is concerned with the recommendation of tasks in response to a web search query. A key step in addressing this problem has to do with the operationalization of the abstract notion of a *task*. We need to establish the universe of possible tasks as well as a computational representation of tasks. For that, we introduce the concept of a *task repository*.

Definition 1.*A task repository $\mathcal{T}$ is a catalog of task descriptions. A task description $t \in \mathcal{T}$ is a semi-structured document that explains the steps involved in the completion of a given task.*

A task description can be, for example, a how-to article illustrating how to change a tire (see Figure 2). It typically contains parts such as a title and a brief explanation of the task, then enumerates the steps needed to accomplish the task. We say that the task description is semi-structured to emphasize that some of these document elements may be missing. Each element may be viewed as a *task attribute*, similar to attributes of entities in a KB. Throughout the article, when we say *task*, we simply refer to this structured representation of a task description.

Our envisioned system would need to handle any input query and accordingly address task recommendation in the case where the query indeed corresponds to an instance of task-based search. In this work, we assume that there is an intent classifier component in place that determines whether the query has indeed a task intent. When the type of the query has been determined as having a task-based intent, a subsequent component performs the ranking of relevant tasks for



How to Change a Tire

Co-authored by Andrew Quinn

Have you ever been stuck on the side of the road with a flat tire? Do you want to be able to change a tire without having to ask for help? Fortunately, changing a tire is a pretty simple task, provided you're prepared and willing to exert a little effort.

Steps

1 **Find a flat, stable and safe place to change your tire.** You should have a solid, level surface that will restrict the car from rolling.

2 **Apply the parking brake and put car into "Park" position.** If you have a standard transmission, put your vehicle in first or reverse.

3 **Place a heavy object (e.g., rock, concrete, spare wheel, etc.) in front of the front and back tires.**

4 **Take out the spare tire and the jack.** Place the jack under the frame near the tire that you are going to change.

Figure 2. WikiHow article corresponding to the task "How to change a tire." The task has a *title* and a *brief explanation*. Each step has a *main act* corresponding to a subtask (here, the first sentence of each step), and is possibly complemented with a *detailed act*.

the query. The problems that we address in this work correspond to this particular component, while assuming that the other parts of the system are in place. These relevant task descriptions are then expected to be recommended to the user accordingly.

Now, we can formally define the problem of *query-based task recommendation*.

Definition 2. *Given a search query q, the problem of query-based task recommendation is to return a ranked list of tasks from a task repository $\mathcal{T}$ that correspond to the task underlying q.*

An illustration of this problem is given by Figure 1, where an example query "birthday party" leads to the recommendation of five tasks.

3.2 Mission-based task recommendation

We now define the *mission-based task recommendation* problem. It generalizes the query-based task recommendation to recommend tasks for a set of queries, all of which are motivated by the same (unknown) underlying task. We refer to such query sets as *search missions*. Specifically, a search mission, or simply *mission*, is a set of queries $m = \{q_1, \ldots, q_n\}$, possibly spanning multiple search sessions, that all share the same underlying task.

We do not address the problem of search mission detection, and instead assume that there is a process in place that identifies and groups queries that share the same underlying task (Jones and Klinkner, 2008; Hagen et al., 2013; Lucchese et al., 2013).

Definition 3. *Given a search mission m as input, the problem of mission-based task recommendation is to return a ranked list of recommended tasks, corresponding to the queries in m.*

Similar to the previous example, given a single query "cake recipe," a highly relevant task to be recommended is $t_1 =$ "Make a cake." Now, assuming that the search mission also contains a second query "cake decorating ideas," the task $t_2 =$ "Make a birthday cake" would also be relevant. Further, if the mission had a third query "birthday invitation cards," then the recommended tasks could also include $t_3 =$ "Organize a birthday party."

4. Approach

This section presents the methods we developed for the problems stated in Section 3. It is important to point out that for these novel problems, we do not have large volumes of query-task pairs to be exploited as training data. This essentially leaves us with a cold start problem and influences the choice of methods to consider. We approach task recommendation as a content-based recommendation problem, which boils down to the challenge of scoring tasks in the repository in response to input queries or missions. The user is then presented with the top-N highest scoring tasks as recommendations. We furthermore expand on this limited data setting, by making use of recent advances in pre-training with transformer-based language models (Devlin et al., 2019; Liu et al. 2019).

4.1 Query-based task recommendation

Our approach for query-based task recommendation is based on document-ranking techniques. By considering individual task attributes as documents, we can apply traditional term frequency-based techniques to rank tasks. Further, we propose to take semantic matching signals into account as well, to extend the capabilities of these already strong traditional techniques.

4.1.1 Term-based matching

We consider every task $t \in \mathcal{T}$ as a document and build an inverted index for each task attribute t_a (Title, Explanation, MainAct, and DetailedAct) for efficient computation. We then score each task attribute t_a against the input query q using the well-established BM25 retrieval method (Robertson and Walker, 1994; Robertson et al., 1996; Robertson and Zaragoza, 2009), obtaining $\text{score}_{\text{BM25}}(t_a, q)$:

$$\text{score}_{\text{BM25}}(t_a, q) = \sum_{w \in q} \frac{f_{w,t_a} \cdot (1 + k_1)}{f_{w,t_a} + k_1 \left(1 - b + b\frac{|t_a|}{\text{avgl}}\right)} idf_w, \tag{1}$$

where f_{w,t_a} is the number of occurrences of word w in the (document corresponding to) task attribute t_a, idf_w is the inverse document frequency for w, avgl is the average document length, k_1 is the calibrating term frequency scaling, and b is the document length normalization factor.

4.1.2 Semantic matching

Consider the query "how to loan shark" (taken from our test corpus, cf. Section 5.1). When matching potential task descriptions, a traditional retrieval method based on term frequencies would fail to capture that the term "shark" does not refer to the fish but to the financial practice of predatory lending with extremely high-interest rates. Another example is the query "how do i put photos in my ipod." A standard document-based scoring method would not be able to capture that "pictures" is a synonym of "photos" in the task title "How to add pictures to my ipod." This kind of vocabulary mismatch is the motivation behind our choice for additionally leveraging semantic representations of terms provided by continuous word embeddings.

Table 1. Features used for learning to recommend tasks

Group	Word embedding	Word functions	#
BM25	-	-	4
W2V, selected	word2vec	SELECTED	28
W2V, complement	word2vec	COMPLEMENT	16
DESM, selected	DESM IN-OUT	SELECTED	28
DESM, complement	DESM IN-OUT	COMPLEMENT	16
RoBERTa	-	-	4
MPNet	-	-	4
Total number of features			100

We use the cosine similarity between a query q and a given task attribute t_a as the semantic score for q and t_a:

$$\text{score}_{\text{sem}}(t_a, q) = \cos(\boldsymbol{t_a}, \boldsymbol{q}). \tag{2}$$

The query and task attribute are each represented by a vector, $\boldsymbol{q}$ and $\boldsymbol{t_a}$, respectively, which are taken to be the centroids of the word vectors corresponding to the individual terms making up q and t_a. The set of words may be further restricted to a word function, that is filtered according to the part-of-speech (POS) tag of a term. Formally

$$\boldsymbol{q} = \frac{1}{|\{w \in q \wedge F(w)\}|} \sum_{w \in q \wedge F(w)} \boldsymbol{w}, \tag{3}$$

where a term vector $\boldsymbol{w}$ is given by some word embedding, and a query term w is only considered if it satisfies a given word function F. The task attribute vector $\boldsymbol{t_a}$ is calculated analogously. As discussed next in Section 4.1.3, regarding word functions, we propose to use combinations of POS tags that correspond to retaining only content words (nouns, verbs, adjectives). This selection by POS is shown to be beneficial as the best-performing feature subsets are based on these selected combinations (cf. Section 6.2).

We propose to combine these semantic matching signals together with the term-based matching score in a learning-to-recommend (LTR) approach. LTR has been widely used for various recommendation problems in the past (Shi et al., 2012a, 2012b, 2013a, 2013b; Rafailidis and Crestani, 2017). The novelty of our work lies in developing features that are appropriate for the task at hand. Specifically, we instantiate the semantic matching score in Equation (2) for each task attribute using multiple pre-trained word embeddings and combinations of word functions.

4.1.3 Features

Next, we detail the set of features we consider; these are summarized in Table 1. The first group of features is based on term-based matching using the BM25 method (cf. Section 4.1.1), computed for each of the four task attributes. The next four groups of features (lines 2–5 in Table 1) are based on semantic matching, using all 4-way combinations of word embeddings (2) and sets of word functions (2), computed for each of the four task attributes.

Embeddings. We consider two pre-trained word embeddings.

- **W2V**: We employ the widely used embeddings provided by the *word2vec*[c] toolkit (Mikolov et al., 2013). These vectors are trained on part of the Google News dataset (about 100 billion words) using a CBOW model with negative sampling. The embedding contains 3 million 300-dimensional vectors. These embeddings correspond to the input matrix in the CBOW model (Mikolov et al., 2013), that is the matrix between the input and hidden layers.
- **DESM**: The Dual Embedding Space Model (Mitra et al., 2016) provides two embedding spaces, IN and OUT, each with 200-dimensional vectors for around 2.74 million terms. This model is the same as in word2vec but (i) is trained on a large unlabeled query corpus and (ii) retains both projections IN and OUT, corresponding to the input and output matrices in CBOW model, respectively. By using IN embedding for the query terms and OUT embedding for the task terms, the semantic similarity is higher between words that often co-occur in the same query or task (topical similarity) (Mitra et al., 2016), that is overcoming part of the vocabulary mismatch problem. We distinguish between these IN and OUT embeddings when taking vectors for words, but refer to this setting altogether as vectors from the DESM model.

We employ the embeddings as corresponding to full form words, that is for a given word embedding, we assign a vector from the embedding to each word of interest in a query or a task attribute. In the case that the word of interest w is not present in the vocabulary of the given embedding, we ignore this word, that is we take the null vector as $\boldsymbol{w}$ in Equation (3).

Word functions. We propose to leverage signals based on POS tags of words. Consider, for example, the query "put a photo in my ipod." Intuitively, the verb "put" should be a strong indicator of the underlying task, with respect to other similar tasks like "take a photo with my ipod." Similarly, the noun "photo" should distinguish the corresponding task from others like "put a video in my ipod." We consider the following word POS-tags: verbs (V), nouns (N), and adjectives (A). Each of these three POS tags is taken as a word function, for example, we say that a term satisfies the word function V if the term is a verb. With these building blocks, we define combinations, such as $V+N$, meaning that a word is a verb or a noun, and consider these combinations as word functions as well. We use ALL to refer to any word function. The possible combinations of word functions are divided into two disjunct sets:

- **SELECTED** $= \{V, N, A, V+N, V+A, N+A, V+N+A\}$.
- **COMPLEMENT** $= \{ALL-V, ALL-N, ALL-A, ALL\}$.

The last two groups of features (lines 6 and 7 in Table 1) are also based on semantic matching. But in this case, we obtain sentence representations for each query and task attribute, using the Sentence Transformers library (Reimers and Gurevych, 2019). This gives us a vector representation for each sentence, encoding its semantic content. In this case, we use cosine distance as our feature of relatedness between each q and t_a pairing. We use two transformer-based architectures (Vaswani et al., 2017), one based on MPNet (Song et al., 2020) and one based on RoBERTa (Liu et al., 2019), as such models perform well in various semantically and syntactically oriented NLP tasks (Nooralahzadeh et al., 2020; Bjerva et al., 2020; Muttenthaler, Augenstein, and Bjerva, 2020; Bjerva and Augenstein, 2021).

In order to calculate every feature described above, all queries and task descriptions are preprocessed by lowercasing and punctuation stripping. In the particular case of retrieval-based features, they are additionally analyzed by stop-word removal, using a built-in stop-word list that is part of the indexing software.

[c]https://code.google.com/archive/p/word2vec/

In total, our feature set includes 100 features. As we show in Section 6.2, it is possible to get a competitive level of performance with just one feature, and a clear performance improvement with just two features, while maximal effectiveness is reached when using 35 features.

4.2 Mission-based task recommendation

We turn now to the problem of mission-based task recommendation. Given a search mission $m = \{q_1, \ldots, q_n\}$, let R_i denote the ranked list of recommended tasks for q_i, obtained by a query-based task recommender method (cf. Section 4.1). Further, we let T be the set of all tasks in all rankings, $T = \bigcup_{i=1}^{n} \{t : t \in R_i\}$.

We introduce a general approach for mission-aware task recommendation that aggregates the individual query-based rankings for each query $q_i \in m$ into a mission-level task ranking. Specifically, let s be a scoring function, such that $s(q_i, t)$ is the score corresponding to task t in the ranking R_i. Then, for each task $t \in T$, the mission-based recommendation score of t is obtained by aggregating the query-based recommendation scores with a given aggregator function aggr:

$$\text{score}_{\text{aggr}}(m, t) = \text{aggr}_{q_i \in m}\{s(q_i, t)\}. \tag{4}$$

We propose two ways to estimate $s(q_i, t)$, each leading to a particular family of mission-based task recommendation methods.

- **Score-based method**: we take $s(q_i, t)$ to be the raw score of the query-based task recommender, $\text{score}(q_i, t)$ (with $s(q_i, t)$ equal to 0 if $t \notin R_i$).
- **Position-based method**: we define $s(q_i, t)$ to be the reciprocal rank position of t in the ranking for q_i. That is, $s(q_i, t) = 1/r_i(t)$, where $r_i(t)$ is the rank position of recommended task t in the ranking R_i (with indexing starting from 1 for the top task, and setting $r_i(t) = |R_i| + 1$ if $t \notin R_i$, where $|R_i|$ is the length of the ranked list R_i).

The aggregator function is one of {sum, max, avg}. Accordingly

- if $\text{aggr} = \text{sum}$, we have $\text{score}_{\text{sum}}(m, t) = \sum_{q_i \in m} s(q_i, t)$;
- $\text{aggr} = \max$ leads to $\text{score}_{\max}(m, t) = \max_{q_i \in m}\{s(q_i, t)\}$;
- and when $\text{aggr} = \text{avg}$, $\text{score}_{\text{avg}}(m, t) = \frac{1}{n} \sum_{q_i \in m} s(q_i, t)$.

5. Dataset construction

In Section 3, we have introduced two task recommendation problems. Given their novelty, there is no standard test collection available to evaluate methods addressing these problems. As one of the main contributions of this study, we create the first test collection for task recommendation. Establishing a task repository that would cover any arbitrary task appears to be very challenging. For this reason, we focus our efforts on a particular subset of tasks, referred to as *procedural tasks*, which are of general interest and are also well covered in a public resource, namely WikiHow.[d] A procedural task is a search task that can be accomplished by following a nontrivial sequence of specific actions or subtasks. Correspondingly, a *procedural mission* is a search mission whose underlying task is procedural.

To illustrate this concept of a procedural task, consider two user goals: (1) reading reviews about a movie and (2) finding out how to tie a tie. Our notion of a procedural task aims to capture only (2) as procedural, since it is this latter goal that could naturally be explained as a nontrivial sequence of steps. By nontrivial, we mean that we expect the user to know how to perform

[d]http://www.wikihow.com

an informational search that itself consists in cognitive activities such as reading, aggregating, comparing, prioritizing information objects, just like the example goal (1). In other words, it is reasonable to assume that the user knows how to search for information, hence the family of tasks about search itself is not considered procedural. The nontriviality criterion also aims for representing the situation where, for a user who knows how to tie a tie, goal (2) is rather trivial, but not for most users in general, so this kind of task is considered procedural. An analogous consideration can be made for the tasks described by how-to articles such as the ones in the corpus that we make use of in this work.

As we will see in Section 5.2, procedural missions make 4% of our sample, a considerable proportion of corpus, verifying in this way the research importance of focusing on procedural tasks. In this section, we describe our corpus of search queries and missions (Section 5.1), the identification of procedural missions (Section 5.2), and the usage of WikiHow as our task repository $\mathcal{T}$ (Section 5.3). The resources developed within this work are made publicly available[e] under CC-BY-SA 4.0 license.

5.1 Corpus of search queries and missions

We base our experiments on the Webis Search Mission Corpus 2012 (Webis-SMC-12)[f] developed by Hagen et al. (2013). It contains 8840 queries issued by 127 users, corresponding to a 3-month sample from the 2006 AOL query log. These queries have been manually assigned to a total of 1378 search missions by two human assessors in full inter-annotator consensus (Hagen et al., 2013). By using this dataset, we eliminate potential errors associated with mission detection and thus ensure that our evaluation is not affected by noisy input from an upstream processing step.

In the original dataset, consecutive queries that belong to different missions are delimited by a special "logical break" marker. Since we are not interested in the multi-tasking aspect of search behavior (i.e., switching between missions in a same session), it is only the set of all queries within a given mission that matters to us. Hence, we post-process the original corpus to merge all queries that belong to the same mission into a single set, by ignoring the logical breaks. We further remove duplicate queries when they correspond to consecutive queries with the same timestamp (effectively, we ignore multiple consecutive clicks for the same query). After this removal, the relative order among the queries in each mission is still preserved. This results in a corpus of 1378 missions, as originally, but now with a total of 3873 queries.

5.2 Corpus of procedural search missions

In the next step, three human experts assessed each of the search missions to decide whether a mission are procedural or not. The Fleiss' Kappa inter-annotator agreement for this assessment was 0.55, which is considered moderate (Fleiss, 1971). For each mission, we take the majority vote to be the final label. In the case of a mission being assessed as procedural, each assessor was also requested to choose a single query from the mission that is the most representative for the task. Further, we take the union of all queries that were chosen as most representative by at least one assessor and refer to this set as the *best queries* of the mission. As a result, 54 out of the 1378 missions were labeled as procedural. This amounts to approximately 4% of all missions in our sample, which is a considerable proportion. Among the 54 procedural missions, the vast majority has a single best query each (six missions have two best queries and two missions have three best queries).

Figure 3 shows the distribution of missions in our corpus according to the number of queries they contain. 724 out of the 1378 missions (i.e., around 52%) consist of a single query, and 280

[e]https://github.com/dariogarigliotti/task-recommendation
[f]https://webis.de/data/webis-smc-12.html

Table 2. Examples of queries for two procedural missions in our corpus

Mission ID: 707848_8	Mission ID: 1045635_22
how to put music on your ipod	**find adress by phone number**
add songs to ipod	acton adress by phone number
put music on your ipod	public records adress by phone
	free adress lookup by phone
	free phone book phone and adress
	acton california adress by phone

Best queries in boldface.

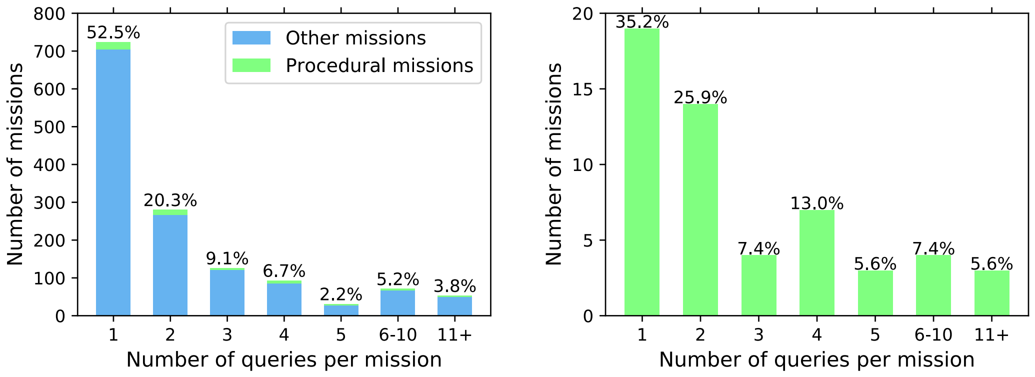


Figure 3. Distribution of (a) missions and (b) procedural missions according to the number of queries they contain.

missions (20%) contain each exactly two queries (Figure 3(a)). As for the procedural missions, 19 out of the 54 missions (35%) consist of a single query (Figure 3(b)). This leaves us with 35 procedural missions with 2+ queries, which is a suitable sample to be used for evaluating our second problem (mission-based task recommendation). Two example procedural missions are shown in Table 2.

5.3 Tasks repository

As our tasks repository $\mathcal{T}$ (cf. Section 3), we use the HowToKB dataset[g] developed by Chu et al. (2017). This dataset contains 168,697 articles crawled from WikiHow. Figure 2 shows an excerpt from a WikiHow article. Each article describes a procedural task and contains the following elements: title, explanation, and sequence of steps (each made of a main act, and possibly also of a detailed act) corresponding to its subtasks. Thus, each task $t \in \mathcal{T}$ is uniquely identified by its WikiHow ID, and represented by a set of four task attributes: Title, Explanation, MainAct, and DetailedAct.

5.4 Relevance assessments

In order to build a test collection for task recommendation, we collect annotations for query-task pairs via crowdsourcing. For each annotation instance (query-task pair), crowd workers were

[g]http://people.mpi-inf.mpg.de/cxchu/projects/howtokb/readme-howtokb.html

Table 3. Examples of queries, recommended tasks, and corresponding relevance assessments in our test collection

Query	Recommended task	Relevance
"opening a small business"	Open a Small Business	Highly relevant
	Brainstorm Small Business Ideas	Relevant
	Compete for Talent As a Small Business	Non-relevant
"how do i put photos in my ipod"	Put Pictures on an iPod	Highly relevant
	Transfer Photos from iPod to PC	Relevant
	Take Screenshots With iPod Touch	Non-relevant

presented with a search query, and a candidate WikiHow article consisting of a title and an explanation. We annotate every best query for all the procedural missions. We asked crowd workers to decide whether a task is highly relevant, relevant, or nonrelevant for the query. A document is highly relevant if it is the exact task that the user wants to accomplish when issuing the query. A document is relevant if it is somehow related to the task behind the query, but not exactly the searched task. Otherwise, it is nonrelevant. Table 3 lists a number of specific examples from our test collection, illustrating the relevance of tasks for queries. Our annotation process uses the well-established top-N pooling technique (Voorhees, 2014); further details are provided below.

Pooling. For each query, we rank tasks using three document ranking baselines: BM25, RM3, and SDM (cf. Section 6.1 for details). Each method was used to generate rankings based on two task attributes, Title and Explanation, resulting in a total of six rankings. We take the top 10 tasks from each of these six rankings to form the pool of tasks that will be annotated. All tasks outside this pool are assumed to be nonrelevant for a given query.

Collecting judgments. Using the Figure Eight platform,[h] a total of 933 unique instances (query-task pairs) were annotated, each by at least three judges (five at most, if necessary to reach a majority agreement, using dynamic judgments). We paid 8¢ per batch, comprising five annotation instances. We ensured quality by requiring a minimum accuracy of 80%, a minimum time of 35 s per batch, and a minimum confidence threshold of 0.8. The final label is taken to be the majority vote among the assessors. The Fleiss' Kappa inter-annotator agreement for this assessment was 0.4116, which is considered moderate (Fleiss, 1971).

In summary, the resulting test collection comprises 59 queries and corresponding relevance assessments for query-based task recommendation (based on the best queries in the missions and removing those for which no relevant task was found in the task catalog). For mission-based task recommendation, the collection has 50 missions (again, after the removal of those without any relevant tasks in the catalog). Relevance labels for missions are obtained by taking the union of the corresponding assessments for the best queries in the mission. In case of conflicting labels for a given task, we take the maximal relevance value assigned to the task.

6. Experimental results

In this section, we describe the experimental setup designed to assess our proposed approaches for the problems of query-based and mission-based task recommendation, and report and analyze the results we obtained.

[h]Figure Eight was acquired by Appen: https://appen.com/figure-eight-is-now-appen/

6.1 Experimental setup

Evaluation metrics. We report on the following rank-based metrics: Normalized discounted cumulative gain (NDCG) and precision (P), each at a cutoff of 10 (indicated as @10), and mean average precision (MAP). These metrics provide high robustness and discriminative power for assessing top-N recommendations (Valcarce et al., 2018).

Evaluation methodology. The relevance level ("gain") of a task to calculate NDCG is set to 2 if it is highly relevant, 1 if relevant, and 0 otherwise. For the LTR results, we used 5-fold cross-validation.

Baseline rankings. We obtain the baseline rankings using Anserini (Yang, Fang, and Lin, 2018). This software toolkit, built around the open-source Lucene search engine, provides state-of-the-art bag-of-words ranking models (Lin et al., 2016). We build an inverted index per task attribute (cf. Section 4.1.1) and consider three baselines: (i) BM25, where parameters in Equation (1) are set as usual to be $k_1 = 1.2$ and $b = 0.75$ (Robertson and Zaragoza, 2009), (ii) SDM (Sequential Dependence Model) (Metzler and Croft, 2005), and (iii) RM3 variant (Jaleel et al., 2004) of relevance models (Lavrenko and Croft, 2001). SDM and RM3 are used with the default parameters in Anserini.

Learning-to-recommend settings. Semantic features, for each task attribute, are computed only for the top-N results retrieved using the corresponding BM25 methods, with N set to 200. We employ the Random Forest algorithm for regression as our supervised ranking method. Following standard parametrization, we set the number of trees (iterations) to 1000, and the maximum number of features in each tree, m, to (the ceiling of) the 10% of the size of the feature set.

6.2 Query-based task recommendation

We begin our experimental investigation by assessing the extent to which content-based recommendation is able to recommend tasks for search queries, and how much we are able to improve by incorporating distributional semantic features. Hence, we start by identifying a ranking baseline. Our first research question is (RQ1) *Which approach is a solid text-based baseline for task recommendation?*

The top block of Table 4 presents the document ranking methods we experiment with. We apply the three ranking models on each of the indexed task attributes. The main observation we make is that the performances of the three models are very close to each other, for all metrics and task attributes. Indeed, using a two-tailed paired t-test with Bonferroni correction (Miller, 1981), we find that no statistical significance exists either between SDM and BM25 or between RM3 and BM25 (for any metric or task attribute). While SDM and RM3 are known to be effective techniques across a broad range of ranking problems (Dalton, Dietz, and Allan, 2014; Guo et al., 2016), for task recommendation they fail to bring any improvements over BM25. If we focus on contrasting by attributes, we observe that Title is consistently the best performing attribute by a clear margin in all metrics and all models. This is reasonable for some of our procedural missions, since their best queries are rather well-formed, and WikiHow article titles are syntactically correct phrases, mostly adhering to the pattern "How to ⟨verb⟩ ⟨noun phrase⟩ ⟨complement⟩," which often matches the query very well. Examples of these queries in our test collection are "how to develop a web site," "how to quit smoking," and "opening a small business." The Title attribute is followed by Explanation, in terms of importance, while the other two attributes do not have a clear order. Based on the above observations, we answer RQ1 by identifying BM25-Title as a solid term-based matching method, and will consider that henceforth as our *baseline* for query-based task recommendation.

Table 4. Performance of query-based task recommendation, measured in terms of NDCG@10, P@10, and MAP

	NDCG@10			
Method	Title	Explanation	MainAct	DetailedAct
BM25	0.3634	0.2926	0.1723	0.1723
RM3	0.3500	0.2736	0.1881	0.1761
SDM	0.3676	0.3019	0.1686	0.1884
LTR	**0.5173**‡			
	P@10			
Method	Title	Explanation	MainAct	DetailedAct
BM25	0.2237	0.1881	0.1220	0.1339
RM3	0.2305	0.1831	0.1390	0.1508
SDM	0.2237	0.1915	0.1220	0.1458
LTR	**0.3085**‡			
	MAP			
Method	Title	Explanation	MainAct	DetailedAct
BM25	0.2731	0.2327	0.1420	0.1332
RM3	0.2551	0.2267	0.1476	0.1429
SDM	0.2752	0.2408	0.1364	0.1356
LTR	**0.4510**‡			

Best scores are in **boldface**. The symbol ‡ denotes statistical significance of the LTR method against the baseline (BM25-Title), tested using a two-tailed paired *t*-test with Bonferroni correction (Miller, 1981) at $p < 0.00\widehat{03}$.

Next, we wish to assess the effectiveness of our semantic matching features that are combined in a LTR approach. This gives rise to the following research question: (RQ2) *How effective are our semantic matching features and our LTR approach for query-based task recommendation?*

Evaluation results for our LTR approach are shown in the bottom line of Table 4. Clearly, the improvements w.r.t. the baseline, BM25-Title, are substantial (42% in terms of NDCG@10, 38% in terms of P@10, and 65% in terms of MAP). Moreover, we use a two-tailed paired t-test with Bonferroni correction at $p < 0.00\widehat{03}$ and verify that these improvements are statistically highly significant for every metric. Hence, the answer to RQ2 is that leveraging semantic matching signals in a LTR approach highly significantly constitutes a more effective method than any of the existing term-based baselines. These results attest that we are able to help substantially and significantly reduce the vocabulary mismatch problem between queries and tasks.

The semantic features proposed in our approach (cf. Section 4.1.3) encompass several different signals, taken from all task attributes, word embeddings, and different combinations of possible word functions. To understand how each of these features contribute to the overall performance, we formulate the following question to guide our analysis: (RQ3) *What are the most important features when learning to recommend tasks?*

We start answering this research question by performing an ablation study on the semantic feature groups. Table 5 presents the results of this additional experiment. The first three rows of

Table 5. Performance of query-based task recommendation, in terms of NDCG@10, P@10, and MAP

	Metric		
Method	NDCG@10	P@10	MAP
BM25-Title	0.3634	0.2237	0.2731
LTR	$0.5173^{‡}$	$0.3085^{‡}$	$0.4510^{‡}$
LTR without W2V features	$0.5105^{‡}$	$0.3119^{‡}$	$0.4423^{‡}$
LTR without DESM features	$0.5072^{‡}$	$0.3119^{‡}$	$0.4525^{‡}$
LTR without W2V and DESM features	$0.4835^{†}$	$0.3102^{†}$	$0.4245^{‡}$
LTR without RoBERTa features	$0.4645^{†}$	$0.2847^{†}$	$0.4239^{‡}$
LTR without MPNet features	$0.4465^{†}$	$0.2780^{†}$	$0.4065^{‡}$
LTR without RoBERTa and MPNet features	0.4221	0.2576	$0.3912^{‡}$

The best performing configuration (LTR) is compared to its corresponding configurations without the features from one or more feature groups. The symbols † and ‡ denote statistical significance of a method against the baseline (BM25-Title), tested using a two-tailed paired *t*-test with Bonferroni correction (Miller, 1981) at $p < 0.01\widehat{6}$ and $p < 0.000\widehat{3}$, respectively.

the ending block correspond to studying the removal of one or both embedding-based feature groups from LTR. The experimental results show that even after removing either W2V or DESM groups entirely, the respective method still outperforms the baseline highly significantly across all metrics, and in some cases achieves performances comparative to or even slightly better than LTR. The removal of both W2V and DESM feature groups affects the performance with higher impact, yet retains significance in its improvements w.r.t. the baseline. We also observe the performance of the LTR configuration when we remove one or both transformer-based feature groups from it. This ablation results are reported in the last three rows of the ending block of Table 5. Across all metrics, it can be seen the large contribution of these features for LTR to improve w.r.t. the baseline with high significance. It also shows that these two underlying transformer-based models complement each other, with MPNet features contributing more than RoBERTa ones to the overall performance of the full set. The configuration where any of the two transformer-based feature groups is removed is significantly better than the baseline, in terms of NDCG@10 and P@10, but these improvements are not highly significant. The removal of both transformer-based feature groups produces a loss of all significance in the improvements measured with any of these two metrics. However, in terms of MAP, the improvement of any of these LTR configurations with less features is always highly significant. Hence, in a scenario where the interest is in the whole ranking, a smaller, simpler set of features may be enough.

Furthermore, we analyze the discriminative power of our features by sorting them according to their individual information gain, measured in terms of Gini importance; these are shown as the vertical bars in Figure 4. Then, we incrementally add features, one by one, according to their importance, to obtain corresponding feature sets made of the I most important features ($I \in \{1, \ldots, 100\}$), and report on performance, in terms of NDCG@10; this is shown as the line plot in Figure 4. Since the number of features changes in each iteration, we set the m parameter of the Random Forests algorithm to (the ceiling of) the 10% of the size of the feature set.

Our findings are as follows. The top three features are, in order, MPNet-Title, RoBERTa-Title, and BM25-Title. Moreover, 11 out of the 15 most important features are all obtained from the Title attribute. This confirms the importance of Title as the most informative task attribute. Still observing the 15 most important features, after the top three, the following 12 include RoBERTa

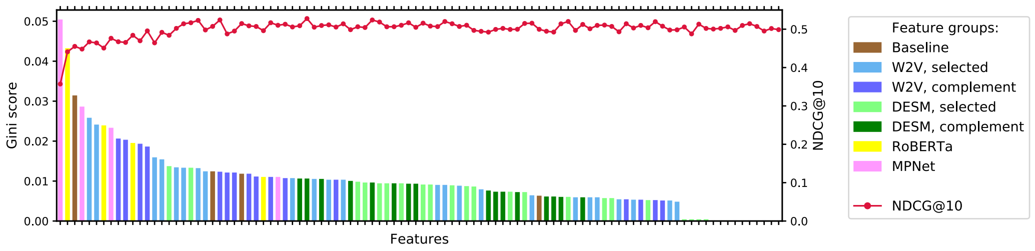


Figure 4. Performance of our LTR approach, in terms of NDCG@10, when incrementally adding features according to their individual information gain, measured by Gini score. The feature grouping is described in Table 1.

and MPNet each obtained from Explanation and DetailedAct attributes, and another eight semantic features which all use the W2V embedding. The expected importance of the features based on transformer-based models is confirmed again in these results, as well as the complementary contributions of these two models. Also, the prominence of W2V features shows that, contrarily to what we expected as explained in Section 4.1.2, the *typical* similarities that dominate the way this embedding is usually employed contributes more to the semantic matching than the *topical* similarity in the IN-OUT strategy with DESM. These semantic features use mainly the word functions $V + N + A$ and $V + N$, which shows, as expected, the importance of selected words as verbs and nouns, yet achieving practically same contributions since very few adjectives occur in the queries or task titles. These are followed by COMPLEMENT word functions.

Looking at how performance changes when adding features one-by-one, we can already achieve a result (NDCG@10 of 0.3571) comparable to the baseline with only the most important feature. Furthermore, we can actually outperform the baseline by adding just a single feature (NDCG@10 of 0.4411). The addition of the third most important feature, the baseline itself BM25-Title, slightly keeps improving the performance (NDCG@10 of 0.4552). From this point on, we observe a low fluctuation in performance, where each new feature sometimes leads to an improvement and other times produces a small deterioration. We find that performance peaks at an NDCG@10 of 0.5277, which is achieved with 35 features. Adding more features slightly hurts performance, but not significantly so. In answer to RQ3, we remark the effectiveness of (i) textual similarity, based on distributional semantic representations, in particular, from transformer-based models, (ii) selected word functions, and with (iii) highest contributions from the Title task attribute.

Since the task attributes are well-formed expressions in natural language, we conduct an additional experiment aiming to shed light on how task recommendation performs for queries that might not fall under this kind of expressions. We approach this observation by partitioning the query set into disjoint subsets of queries, according to the number of words in each query. In our test collection, the largest number of words for a query is 10. This is, given the set Q of 59 queries, we measure the LTR performance in each element of the partition $\mathcal{Q} = \{Q_i \subseteq Q | \text{each query in } Q_i \text{ has } i \text{ words}\}_{i=1}^{10}$. Figure 5 shows the the results obtained in this experiment. We can observe that, with the exception of the single query made of one word, for very short queries, the performance is lower across all the metrics. LTR performs better as the word counter increases; this should correspond with queries made of four or more words, such as “how to mount tires” or “learn to remodel a house,” which are the ones that can present a verb and a direct object, as well as some opening segment (e.g., “how to,” “learn to,” “how do I”) related to the user intent for a procedural task. Some queries of these lengths present slightly different structure (e.g., no opening segment before the verb in “block incoming email addresses”) but in general these are queries long enough to express their underlying intents in a more specific level of detail. For queries with seven or more words, task recommendation increasingly deteriorates, which makes sense as longer queries seem to become too specific to get appropriate recommendations from the task repository.

Table 6. Performance of mission-based task recommendation, compared against a query-based method, in terms of NDCG@10, P@10, and MAP

		Metric		
Method	Aggregator	NDCG@10	P@10	MAP
Query-based	-	**0.5207**	0.3140	**0.4716**
Mission-based by score	Sum	0.5106	**0.3320**	0.4581
	Max	0.4819	0.3220	0.4288
	Avg	0.5034	0.3280	0.4496
Mission-based by position	Sum	0.4881	0.3120	0.4450
	Max	0.4926	0.3200	0.4303
	Avg	0.4845	0.3100	0.4397

Best scores for each metric are in **boldface**; improvements over the query-based method are underlined. None of these performance differences are found to be statistically significant.

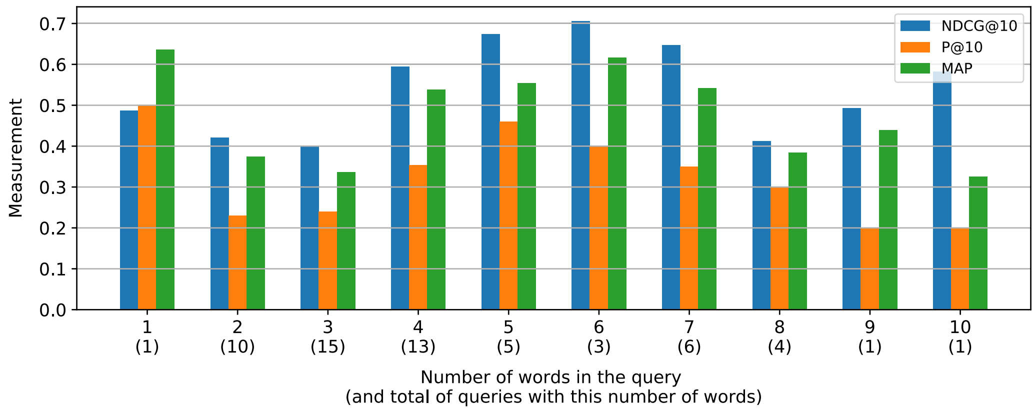


Figure 5. Performance of our LTR approach for query-based task recommendation (in terms of NDCG@10, P@10, and MAP), measured on each of the subsets given by partitioning the query set according to the number of words in each query.

6.3 Mission-based task recommendation

In this second part of the analysis, our experiments aim to answer the research question (RQ4) *Can task recommendation performance benefit from having access to all search queries in a mission, in addition to the best query?*

We report on all combinations of scoring methods (score-based vs. position-based) and aggregator functions (sum, max, or avg) in Table 6. Building on our feature analysis, we conduct this part of the experiments learning to recommend with only the 35 most important features. In particular, the first row of Table 6 reports the performance of query-based task recommendation using only such an optimized features set. For mission-based task recommendation, in order to make a fair comparison, we randomly select a single best query per mission and compare against the query-based LTR restricted to only these selected best queries. Overall, the results are mixed. In terms of NDCG@10 and MAP, none of the mission-based methods improve over the query-based approach. On the other hand, when observing P@10, most cases in the mission-based approach exhibit an improvement. However, none of these differences is statistically significant for any metric.

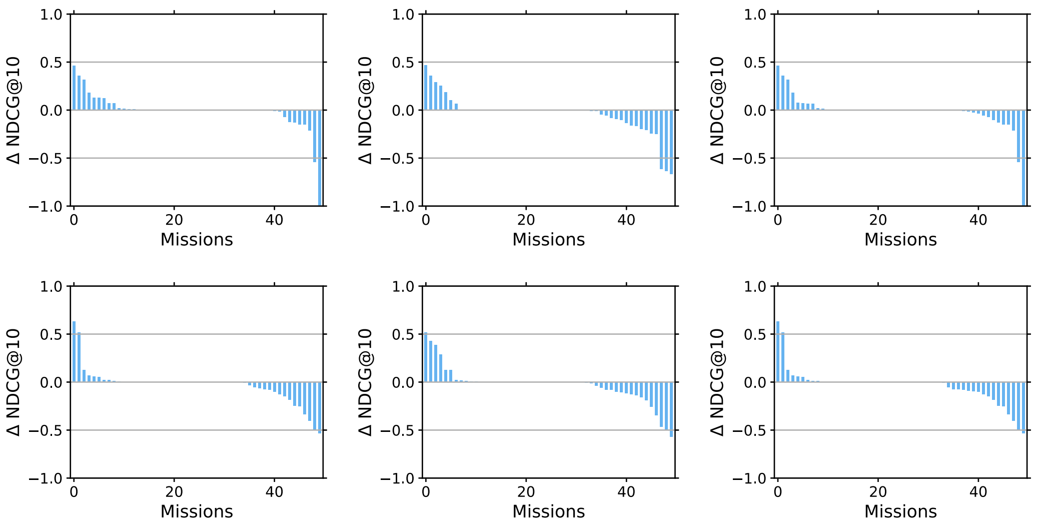


Figure 6. Differences in NDCG@10 per query between a given mission-based method and the query-based method.

To shed some light on the reasons behind this, we perform the following analysis. In Figure 6, we plot the differences in NDCG@10, for each individual mission, between a given mission-based configuration and the query-based method. Each positive bar corresponds to a mission where the mission-based method improved over the query-based one, and each negative bar represents a mission where the opposite is true. A key observation we make here is that there are indeed substantial differences on some missions, and that in some cases there are about as many wins as losses, with other cases having a tendency to the latter. There are several possible reasons for this. Since the query-based method uses the best query in a mission, which is identified by an oracle, it may be that, on average, it simply does not help to consider other queries in the mission as those often only bring in noise. Weighing queries in a mission may help to rectify it to some extent; we leave this exploration to future work. Another reason may lie in how the test collection was created. Recall that relevance labeling was done based only on the best queries, and not on other queries in the mission. Those other queries could possibly yield more relevant tasks, which have not been labeled as such.

We now report some of the particular examples when analyzing results of individual missions. Table 7 presents best and worst performing missions per configuration, as well as examples of relevant tasks in the test collection, and their rank position in each ranking. A mission with an NDCG@10 difference of 0.46 on each of the score-based methods with sum or avg aggregator (NDCG@10 difference of 0.47 in the max-aggregator method) has “how to create a petition not online” as one of its two best queries, the other best query being “how to create a petition.” For this mission, all score-based methods rank one of the only two highly relevant tasks, “Write a Petition,” as the top result while LTR ranks it as its third best result. This mission comprises three queries. Only the two best queries have a verb, “create,” but one of them does not include “not online.” When these two queries in the mission are taken into account, alongside its third query, “petition,” it becomes possible for the highly relevant task to improve its ranking with the mission-based method. The score-based method also exhibits a mission with a negative NDCG@10 difference of -1.0 for sum (Figure 6a) and avg (Figure 6c) aggregators (NDCG@10 difference of -0.67 for max; Figure 6b) as its worst performing mission. Represented by “michigan teacher certification,” the mission has a single relevant task in the test collection, “Become Certified to Teach Hunter Safety in Michigan,” ranked as top result by query-based LTR, but only as low ranked as the eleventh or seventh result by a score-based method. This mission has five queries, most of them mentioning terms such as “teaching license exam” and “gid promotions,” so these additional queries in

Table 7. Queries with the largest ΔNDCG@10 differences, using a mission-based configuration, in comparison with the query-based method

			Rankposition	
Δ	Mission ID (and best query)	Relevant task	MB	LTR
Score-based, sum or avg				
+0.46	3389546_26 ("how to create a petition not online")	Write a Petition	1	3
−1.0	10733023_1 ("michigan teacher certification")	Become Certified to Teach Hunter Safety in Michigan	11	1
Score-based, max				
+0.47	3389546_26 ("how to create a petition not online")	Write a Petition	1	3
−0.67	10733023_1 ("michigan teacher certification")	Become Certified to Teach Hunter Safety in Michigan	7	1
Position-based, sum or avg				
+0.63	1045635_18 ("install spyware")	Remove Spyware Manually (Windows)	2	14
−0.5	2161465_3 ("using the normal curve to find probability")	Calculate Probability	3	1
−0.54	1602009_1 ("how to pass urine drug test")	Pass a Urine Drug Test	3	1
Position-based,max				
+0.52	3389546_26 ("how to create a petition not online")	Write a Petition	2	3
−0.5	2161465_3 ("using the normal curve to find probability")	Calculate Probability	3	1
−0.57	10733023_1 ("michigan teacher certification")	Become Certified to Teach Hunter Safety in Michigan	4	1

Examples of relevant tasks in the test collection, as well as their rank position in the respective mission-based (MB) and query-based LTR ranking, are also presented.

the mission should help to express the actual user intent searching for general (e.g., primary and secondary school) teaching. And in practice, they contribute to lower the rank for the task of getting certified for hunter safety, which is realistically only tangentially relevant for teacher certification. However, since this is the only relevant task in the test collection, in our evaluation, the mission-based method performs as if it is introducing noise for this query.

For position-based methods (Figure 6d–6f), the phenomena regarding best and worst differences are very similar to score-based methods. The positive differences for position-based methods are slightly larger than for score-based methods, while the negative differences for position-based methods are slightly smaller than their score-based counterpart. The largest NDCG@10 difference (+0.63) for a position-based method with either sum or avg aggregator corresponds to a mission with "install spyware" as its best query. For this mission, both position-based methods, as well as query-based LTR, rank a non-relevant task, "Find Spyware," as their top result. However, the only relevant task, "Remove Spyware Manually (Windows)," is ranked by query-based LTR beyond the rank 10 cutoff, whereas each position-based method ranks it as its second best result. This mission comprises 10 queries, most of them with an occurrence of the term "spyware." Only the best query has a verb, "install," antonym of the verb "remove" in the relevant task. This vocabulary drift hinders the query-based ranking for this query, whereas the mission-based method is able to address it.

A mission with one of the worst differences in all position-based methods is represented by the best query "using the normal curve to find probability" and has a total of 17 queries drifting through several topics related to statistics, such as "linear regression" and "sample size." The single relevant task for this mission, "Calculate Probability," is the top recommended task by query-based LTR but only ranked as the third result at highest in all position-based methods. The best-performing mission for position-based method with max has a best query "how to create a petition not online," and corresponds to the same mission that performed best for the score-based methods previously analyzed. The position-based method with max aggregator ranks one of the only two highly relevant tasks, "Write a Petition," as its second best result, whereas LTR ranks it at rank 3.

In answer to RQ4, we find that our extensions of the query-based approach are promising, but more work is needed to be able to benefit from additional queries in a mission and to reach significant improvements.

7. Conclusions

In this article, we have introduced the problem of recommending tasks to users, based on their search queries. Specifically, we have defined the query-based task recommendation problem, and approached it as a content-based recommendation problem, which boils down to scoring tasks from a task repository in response to input queries. We have proposed a method for this problem that combines a strong term-based ranking technique with continuous semantic representations in a LTR framework. Further, we have extended our approach to deal with a search mission, that is a set made of an arbitrary number of queries that all share the same underlying task. Specifically, we have introduced the problem of mission-based task recommendation and proposed methods that aggregate the individual query-based recommendations for each query in the mission into mission-level recommended tasks. In this way, task recommendation can be coupled as a support component on top of any existing method for detecting the queries that belong to the same underlying task.

To overcome the lack of a standard test collection to evaluate our novel problems, we have created the first test collection to assess task recommendation. The collection comprises 59 queries, annotated via crowdsourcing with relevance assessments for tasks in a repository of WikiHow articles. Our experimental results have demonstrated the effectiveness of our approach, and in this way, the suitability of the proposed method as baseline for the query-based task recommendation problem. Finally, feature and query analyses have brought insights concerning the importance and effectiveness of various semantic signals leveraged from distributional representations.

In future work, we would address the utilization of other queries in the user's search mission, possibly by weighting them appropriately or considering their temporal order. In particular, these alternatives could benefit scenarios like the example we introduced when stating the problem of mission-based task recommendation. Another line of future investigation concerns the integration of the act-related task attributes with information corresponding to the subtasks of the task at hand. A third possible research direction could be to study aspects that are reasonable as specific within task-based search, in particular, related to the domain of how-to articles, aspects that could otherwise manifest very differently in other domains of search. An example of these kind of aspects is the relatively low impact of adjectives here observed, which could be rather much more sensitive for other flavors of search or recommendation. A fourth area of possible future work corresponds to studying variations of approaches (e.g., BM25f which allows different weights for different fields, or character-level or subword-level embedding techniques especially successful for morphologically rich languages), as well as experimenting with dataset extensions by the means of obtaining and using paraphrases of the existing queries. A fifth line of future investigation is the proposal and study of related research problems concerning the diversity of the task recommendation results. Second and fifth lines of future work pertain considering the possible subtasks within a task and their usefulness for diversity, aspects of task-based search that we have already studied accordingly (Garigliotti and Balog, 2017; Ding et al., 2018).

Acknowledgements. This research was partially funded by the Poul Due Jensen Foundation.

Competing interests. The authors declare none.

References

Bi B., Ma H., Hsu B.-J. P., Chu W., Wang K. and Cho J. (2015). *Learning to recommend related entities to search users.* Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM '15, Association for Computing Machinery, pp. 139–148.

Bjerva J. and Augenstein I. (2021). *Does typological blinding impede cross-lingual sharing?.* Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, pp. 480–486.

Bjerva J., Bhutani N., Golshan B., Tan W.-C. and Augenstein I. (2020). *SubjQA: A dataset for subjectivity and review comprehension.* Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 5480–5494.

Blanco R., Cambazoglu B. B., Mika P. and Torzec N. (2013). *Entity recommendations in web search.* Proceedings of the 12th International Semantic Web Conference, ISWC '13, Association for Computing Machinery, pp. 33–48.

Bonchi F., Perego R., Silvestri F., Vahabi H. and Venturini R. (2012). *Efficient query recommendations in the long tail via center-piece subgraphs.* Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, Association for Computing Machinery, pp. 345–354.

Broder A. (2002). A taxonomy of web search. *SIGIR Forum* **36**(2), 3–10.

Byström K. (2002). Information and information sources in tasks of varying complexity. *Journal of the American Society for Information Science and Technology* **53**(7), 581–591.

Choi B., Arguello J., Capra R. and Ward A. R. (2021). *OrgBox: A knowledge representation tool to support complex search tasks.* Proceedings of the 2021 Conference on Human Information Interaction and Retrieval, CHIIR, ACM, vol **21**, pp. 219–228.

Chu C. X., Tandon N. and Weikum G. (2017). *Distilling task knowledge from how-to communities.* Proceedings of the 26th International Conference on World Wide Web, WWW '17, Association for Computing Machinery, pp. 805–814.

Dalton J., Dietz L. and Allan J. (2014). *Entity query feature expansion using knowledge base links.* Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Association for Computing Machinery, pp. 365–374.

de Vries A. P., Vercoustre A.-M., Thom J. A., Craswell N. and Lalmas M. (2008). Overview of the INEX 2007 entity ranking track, *Focused Access to XML Documents, INEX '07*, Association for Computing Machinery, pp. 245–251.

Dehghani M., Rothe S., Alfonseca E. and Fleury P. (2017). *Learning to attend, copy, and generate for session-based query suggestion.* Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, pp. 1747–1756.

Demartini G., de Vries A. P., Iofciu T. and Zhu J. (2009). Overview of the INEX 2008 entity ranking track, *Advances in Focused Retrieval, INEX '08*, Association for Computing Machinery, pp. 243–252.

Demartini G., Iofciu T. and de Vries A. P. (2010). Overview of the INEX 2009 entity ranking track, *Focused Retrieval and Evaluation, 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, Revised and Selected Papers, INEX '09*, Association for Computing Machinery, pp. 254–264.

Devlin J., Chang M.-W., Lee K. and Toutanova K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN: Association for Computational Linguistics, pp. 4171–4186.

Ding H., Zhang S., Garigliotti D. and Balog K. (2018). *Generating high-quality query suggestion candidates for task-based search.* Advances in Information Retrieval, Proceedings of the 40th European Conference on IR Research, ECIR '18, Springer-Verlag, pp. 625–631.

Donato D., Bonchi F., Chi T. and Maarek Y. (2010). *Do you want to take notes?: Identifying research missions in Yahoo! search pad.* Proceedings of the 19th International Conference on World Wide Web, WWW '10, Association for Computing Machinery, pp. 321–330.

Fernández-Tobías I. and Blanco R. (2016). *Memory-based recommendations of entities for web search users.* Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, Association for Computing Machinery, pp. 35–44.

Fleiss J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin* **76**(5), 378–382.

Garigliotti D. and Balog K. (2017). *Generating query suggestions to support task-based search*. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, Association for Computing Machinery, pp. 1153–1156.

Georgeff M. P. and Lansky A. L. (1986). Procedural knowledge. *Proceedings of the IEEE* **74**, 1383–1398.

Guo J., Cheng X., Xu G. and Zhu X. (2011). *Intent-aware query similarity*. Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11, Association for Computing Machinery, pp. 259–268.

Guo J., Fan Y., Ai Q. and Croft W. B. (2016). *Semantic matching by non-linear word transportation for information retrieval*. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, Association for Computing Machinery, pp. 701–710.

Hagen M., Gomoll J., Beyer A. and Stein B. (2013). *From search session detection to search mission detection*. Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, OAIR '13, pp. 85–92.

Hassan Awadallah A., White R. W., Dumais S. and Wang Y.-M. (2014a). *Struggling or exploring? Disambiguating search sessions*. Proceedings of the 7th Annual International ACM Conference on Web Search and Data Mining, WSDM '14, Association for Computing Machinery, pp. 53–62.

Hassan Awadallah A., White R. W., Pantel P., Dumais S. T. and Wang Y.-M. (2014b). *Supporting complex search tasks*. Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14, Association for Computing Machinery, pp. 829–838.

He D., Göker A. and Harper D. J. (2002). Combining evidence for automatic web session identification. *Information Processing and Management* **38**(5), 727–742.

Jaleel N. A., Allan J., Croft W. B., Diaz F., Larkey L. S., Li X., Smucker M. D. and Wade C. (2004). *UMass at TREC 2004: Novelty and HARD*. Proceedings of the Thirteenth Text REtrieval Conference, TREC '04.

Jansen B. J., Booth D. L. and Spink A. (2008). Determining the informational, navigational, and transactional intent of web queries. *Information Processing and Management* **44**(3), 1251–1266.

Jones R. and Klinkner K. L. (2008). *Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs*. Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, Association for Computing Machinery, pp. 699–708.

Jung Y., Ryu J., Kim K.-m. and Myaeng S.-H. (2010). Automatic construction of a large-scale situation ontology by mining how-to instructions from the web. *Web Semantics: Science, Services and Agents* **8**(2-3), 110–124.

Kelly D., Arguello J. and Capra R. (2013). NSF workshop on task-based information search systems. *SIGIR Forum* **47**(2), 116–127.

Lavrenko V. and Croft W. B. (2001). *Relevance based language models*. Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01, Association for Computing Machinery, pp. 120–127.

Li L., Deng H., Dong A., Chang Y. and Zha H. (2014). *Identifying and labeling search tasks via query-based Hawkes processes*. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, Association for Computing Machinery, pp. 731–740.

Li L., Deng H., He Y., Dong A., Chang Y. and Zha H. (2016). *Behavior driven topic transition for search task identification*. Proceedings of the 25th International Conference on World Wide Web, WWW '16, Association for Computing Machinery, pp. 555–565.

Lin J., Crane M., Trotman A., Callan J., Chattopadhyaya I., Foley J., Ingersoll G., Macdonald C. and Vigna S. (2016). *Toward reproducible baselines: The open-source IR reproducibility challenge*. Advances in Information Retrieval, Proceedings of the 38th European Conference on IR Research, ECIR '16, Springer-Verlag, pp. 408–420.

Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv: 1907.

Lucchese C., Nardini F. M., Orlando S. and Tolomei G. (2016). *Learning to rank user queries to detect search tasks*. Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR '16, Association for Computing Machinery, pp. 157–166.

Lucchese C., Orlando S., Perego R., Silvestri F. and Tolomei G. (2011). *Identifying task-based sessions in search engine query logs*. Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM '11, pp. 277–286.

Lucchese C., Orlando S., Perego R., Silvestri F. and Tolomei G. (2013). Discovering tasks from search engine query logs. *ACM Transactions on Information Systems* **31**(3), 43–43.

Lugo L., Moreno J. G. and Hubert G. (2020a). *A multilingual approach for unsupervised search task identification*. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, Association for Computing Machinery, pp. 2041–2044.

Lugo L., Moreno J. G. and Hubert G. (2020b). *Segmenting search query logs by learning to detect search task boundaries*. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, Association for Computing Machinery, pp. 2037–2040.

Marchionini G. (2006). Exploratory search: From finding to understanding. *Communications of the ACM* **49**(4), 41–46.

Mei Q., Klinkner K. L., Kumar R. and Tomkins A. (2009). *An analysis framework for search sequences.* Proceedings of the 18th ACM International Conference on Information and Knowledge Management, CIKM '09, Association for Computing Machinery, pp. 1991–1994.

Metzler D. and Croft W. B. (2005). *A Markov Random Field model for term dependencies.* Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05, Association for Computing Machinery, pp. 472–479.

Mikolov T., Sutskever I., Chen K., Corrado G. S. and Dean J. (2013). *Distributed representations of words and phrases and their compositionality.* Proceedings of the 27th Annual Conference on Neural Information Processing Systems, NIPS '13, pp. 3111–3119.

Miller R. G. J. (1981). *Simultaneous Statistical Inference.* New York: Springer-Verlag.

Mitra B., Nalisnick E. T., Craswell N. and Caruana R. (2016). A dual embedding space model for document ranking. *CoRR*, abs/1602.01137.

Muttenthaler L., Augenstein I. and Bjerva J. (2020). Unsupervised evaluation for question answering with transformers, *Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, pp. 83–90.

Nooralahzadeh F., Bekoulis G., Bjerva J. and Augenstein I. (2020). *Zero-shot cross-lingual transfer with meta learning.* Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 4547–4562.

Ozertem U., Chapelle O., Donmez P. and Velipasaoglu E. (2012). *Learning to suggest: A machine learning framework for ranking query suggestions.* Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, SIGIR '12, Association for Computing Machinery, pp. 25–34.

Pareti P., Klein E. and Barker A. (2014). *A semantic web of know-how: Linked Data for community-centric tasks.* Proceedings of the 23rd International Conference on World Wide Web, WWW '14 Companion, Association for Computing Machinery, pp. 1011–1016.

Radlinski F. and Joachims T. (2005). *Query chains: Learning to rank from implicit feedback.* Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '05, Association for Computing Machinery, pp. 239–248.

Rafailidis D. and Crestani F. (2017). *Learning to rank with trust and distrust in recommender systems.* Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys '17, Association for Computing Machinery, pp. 5–13.

Raman K., Bennett P. N. and Collins-Thompson K. (2013). *Toward whole-session relevance: Exploring intrinsic diversity in web search.* Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13, Association for Computing Machinery, pp. 463–472.

Reimers N. and Gurevych I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.* Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 3980–3990.

Robertson S., Walker S., Beaulieu M., Gatford M. and Payne A. (1996). *Okapi at TREC-4.* Proceedings of the 4th Text REtrieval Conference (TREC-4), pp. 73–96.

Robertson S. and Zaragoza H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval* **3**(4), 333–389.

Robertson S. E. and Walker S. (1994). *Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval.* Proceedings of SIGIR'94, Springer-Verlag, pp. 232–241.

Rose D. E. and Levinson D. (2004). *Understanding user goals in web search.* Proceedings of WWW, Association for Computing Machinery, pp. 13–19.

Shi Y., Karatzoglou A., Baltrunas L., Larson M. and Hanjalic A. (2013a). *GAPfm: Optimal top-n recommendations for graded relevance domains.* 22nd ACM International Conference on Information and Knowledge Management, CIKM '13, Association for Computing Machinery, pp. 2261–2266.

Shi Y., Karatzoglou A., Baltrunas L., Larson M. and Hanjalic A. (2013b). *xCLiMF: Optimizing expected reciprocal rank for data with multiple levels of relevance.* Proceedings of the Seventh ACM Conference on Recommender Systems, RecSys '13, Association for Computing Machinery, pp. 431–434.

Shi Y., Karatzoglou A., Baltrunas L., Larson M., Hanjalic A. and Oliver N. (2012a). *TFMAP: Optimizing MAP for top-n context-aware recommendation.* The 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, Association for Computing Machinery, pp. 155–164.

Shi Y., Karatzoglou A., Baltrunas L., Larson M., Oliver N. and Hanjalic A. (2012b). *CLiMF: Learning to maximize reciprocal rank with collaborative less-is-more filtering.* Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12, Association for Computing Machinery, pp. 139–146.

Song K., Tan X., Qin T., Lu J. and Liu T.-Y. (2020). MPNet: Masked and permuted pre-training for language understanding, *Advances in Neural Information Processing Systems*, volume 33 of NeurIPS '20, Curran Associates, Inc, pp. 16857–16867.

Szpektor I., Gionis A. and Maarek Y. (2011). *Improving recommendation for long-tail queries via templates.* Proceedings of the 20th International Conference on World Wide Web, WWW '11, Association for Computing Machinery, pp. 47–56.

Toms E. G., Villa R. and Mccay-Peet L. (2013). How is a search system used in work task completion? *Journal of Information Science* **39**(1), 15–25.

Valcarce D., Bellogín A., Parapar J. and Castells P. (2018). *On the robustness and discriminative power of information retrieval metrics for top-n recommendation*. Proceedings of the Twelfth ACM Conference on Recommender Systems, RecSys '18, Association for Computing Machinery, pp. 260–268.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł. and Polosukhin I. (2017). Attention is all you need, *Advances in Neural Information Processing Systems*, pp. 5998–6008.

Voorhees E. M. (2014). *The effect of sampling strategy on inferred measures*. Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Association for Computing Machinery, pp. 1119–1122.

Wang H., Song Y., Chang M.-W., He X., White R. W. and Chu W. (2013). *Learning to extract cross-session search tasks*. Proceedings of the 22nd International Conference on World Wide Web, WWW '13, Association for Computing Machinery, pp. 1353–1364.

Yang P., Fang H. and Lin J. (2018). Anserini: Reproducible ranking baselines using Lucene. *Journal of Data and Information Quality* **10**(4), 20.

Yu X., Ma H., Hsu B.-J. P. and Han J. (2014). *On building entity recommender systems using user click log and Freebase knowledge*. Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM '14, Association for Computing Machinery, pp. 263–272.

Zhang L., Lyu Q. and Callison-Burch C. (2020a). *Intent detection with WikiHow*. Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Suzhou, China: Association for Computational Linguistics, pp. 328–333.

Zhang L., Lyu Q. and Callison-Burch C. (2020b). *Reasoning about goals, steps, and temporal ordering with WikiHow*. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 4630–4639.

Zhou Y., Shah J. and Schockaert S. (2019). *Learning household task knowledge from WikiHow descriptions*. Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5), Macau, China: Association for Computational Linguistics, pp. 50–56.

Cite this article: Garigliotti D, Balog K, Hose K and Bjerva J (2024). Recommending tasks based on search queries and missions. *Natural Language Engineering* **30**, 577–601. https://doi.org/10.1017/S1351324923000219