Query-Task Mapping

Michael Völske
Ehsan Fatehifar
Benno Stein
<firstname>.<lastname>@uni-weimar.de
Bauhaus-Universität Weimar
Weimar, Germany

Matthias Hagen
matthias.hagen@informatik.uni-halle.de
Martin-Luther-Universität Halle-Wittenberg
Halle (Saale), Germany

ABSTRACT

Several recent task-based search studies aim at splitting query logs into sets of queries for the same task or information need. We address the natural next step: mapping a currently submitted query to an appropriate task in an already task-split log. This query-task mapping can, for instance, enhance query suggestions—rendering efficiency of the mapping, besides accuracy, a key objective.

Our main contributions are three large benchmark datasets and preliminary experiments with four query-task mapping approaches: (1) a Trie-based approach, (2) MinHash LSH, (3) word movers distance in a Word2Vec setup, and (4) an inverted index-based approach. The experiments show that the fast and accurate inverted index-based method forms a strong baseline.

ACM Reference Format:
Michael Völske, Ehsan Fatehifar, Benno Stein, and Matthias Hagen. 2019. Query-Task Mapping. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331286

1 INTRODUCTION

Users often turn to a search engine to fulfill an underlying task that led to the information need expressed in a query. The field of task-based search aims to understand the tasks behind information needs, in order to develop better support tools. Recent research has focused on observing user behavior during task-based search [15] or on splitting query logs into tasks and subtasks [24]. Given a task-split query log, we focus on the natural next step: map a new query to the most appropriate task. Query-task mapping may be used to derive task-based query embeddings [25] or to identify query suggestions [33]. Since query suggestions have to be derived in milliseconds, efficiency is a crucial factor besides effectiveness. Hence, our study analyzes runtime along with accuracy.

We create three benchmarking datasets:[1] one based on search session and mission detection corpora [8, 21], another based on the TREC Sessions and Tasks tracks [4, 14] combined with a corpus of TREC-based search sessions [9], and the third built from wikiHow questions. We enlarge each dataset with query suggestions from Google and Bing to reach several tens of thousands of queries and annotate the task information. In a preliminary study, we test four query-task mapping methods on our new datasets: (1) a Trie-based approach, (2) Minhash LSH Forest, (3) word movers distance in a Word2Vec setup, and (4) an Elasticsearch-based BM25 retrieval. In our experiments, the fast and accurate retrieval approach turns out to be a strong query-task mapping baseline.

2 RELATED WORK

Research on matching queries with the same information need has recently shifted focus from single-user oriented session and mission detection [6–8, 10, 12, 13, 15, 18, 21] to the more multi-user oriented problem of splitting search logs into tasks [3, 11, 17, 22–24].

The studies on search sessions aimed to either match a current query to one of the previous queries of the same user submitted either within the same (time-based) physical session, or to some set of queries (search mission) from before [13, 30]. The goal then was to better support a user with their search, either by better understanding the information need based on the directly preceding queries (as in the TREC Sessions tracks [4]), or by helping a user resume a previously abandoned information need [15, 29]. Already then, session and mission detection techniques recognized runtime as important to the online setting [8].

Recently, the focus has shifted away from the notion of individual users' search missions towards one of complex tasks that can re-occur across users. A complex search task is a multi-aspect or multi-step information need comprising subtasks which might recursively be complex; planning a journey is a typical example [1], and studies have aimed to subdivide query logs into clusters of same-task queries [19]. As before, the goal is to support individual users, but this time by leveraging what others have done in similar situations. One idea is to suggest related queries from the identified query-task clusters like in the TREC Task tracks' setting [14].

Grouping the queries of some larger log into tasks and potentially subtasks has been tackled in different ways ranging from Hawkes processes [17], Bayesian Rose trees [24], entity relations [31], to DBpedia categories [32]. However, no large annotated datasets of logs split into tasks are available. And, maybe even more importantly, the problem of quickly mapping a currently submitted query to an appropriate task in a task-split background log has not been really studied in the literature so far. We address both issues by providing three large benchmarking datasets of task-split queries and an empirical study of four approaches for query-task mapping.

[1] Data available from: https://webis.de/data/webis-qtm-19.html

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331286

3 BENCHMARKING DATASETS

We provide three new datasets of queries split into tasks with different "characteristics:" (1) based on available search session / mission corpora, (2) based on queries for TREC topics, and (3) based on wikiHow questions. Table 1 gives a high-level overview of the dataset construction and basic statistics.

3.1 Session-based Dataset

Research on session and mission detection has produced three publicly available corpora of queries sampled from the AOL query log [28] annotated with individual users' search sessions [7, 8, 20]. Since the newer corpus of Hagen et al. [8] and Gayo-Avello's corpus [7] are based on the same sample, we use the corpora of Lucchese et al. [20] and of Hagen et al. [8] as our basis.

Lucchese et al. [20] sampled from the 500 time-based sessions with the largest number of queries from the first week of the AOL log. The 1424 queries from 13 users in this corpus are manually annotated with logical session information (i.e., which queries in a time-based session of one user were probably submitted for the same information need). Note that the corpus does not contain all queries from the sampled users. We manually annotated the 771 unique queries from the corpus (clicks in the AOL log are often logged as "another" query) with task information across users, taking into account Lucchese et al.'s session information. Altogether, we identified 223 tasks with 3.5 unique queries on average.

Hagen et al. [8] took all queries from the 215 AOL users in Gayo-Avello's corpus [7], removed 88 users with fewer than 4 queries, and annotated the remaining 8840 queries from 127 users with per-user logical session and search mission information (1378 missions). We manually annotated the 3750 unique queries from the corpus with task information across users, taking into account existing session and mission information. Altogether, we identified 1298 tasks (some search missions indeed were identical between users) with 2.9 queries on average. We then merged the unique queries from both corpora (4502 queries) and manually checked whether some tasks were similar enough to be merged. This resulted in 1423 tasks for both corpora combined with an average of 3.2 queries.

To enlarge the dataset to tens of thousands of queries, we submit each original query to Google and Bing and scrape the query suggestions that we then add to the same task. We discard suggestions that have the original query as a prefix, but do not continue with a new term.[2] Manual spot checks showed the task assignment to be reasonable for the remaining suggestions, with a small number of exceptions where the search engines returned suggestions in a different language, which were removed semi-automatically; further spot checks showed the remaining suggestions to be accurate. We gathered 24,939 unique suggestions from Google (queries from the original data were not added again) and 12,339 from Bing (queries already suggested by Google were not taken twice), resulting in a larger corpus of 41,780 queries for 1,423 tasks (30 queries per task).

Table 1: Statistics of the benchmark datasets. Rows with "+" are cumulative, omitting duplicate task-query pairs.

	Tasks	Queries	Queries per Task		
			min	avg	max
Session-based dataset					
Lucchese et al. [20]	223	771	1	3.5	55
+ Hagen et al. [8]	1,423	4,502	1	3.2	147
+ Google suggestions	1,423	29,441	1	20.7	924
+ Bing suggestions	1,423	41,780	1	29.4	1,368
TREC-based dataset					
Webis-TRC-12 [9]	150	3,848	1	25.7	122
+ TREC	276	7,771	1	28.2	144
+ Google suggestions	276	38,478	8	139.4	858
+ Bing suggestions	276	47,514	8	172.2	997
WikiHow-based dataset					
WikiHow	7,202	15,914	1	2.2	22
+ Google suggestions	7,202	119,283	1	16.6	197
+ Bing suggestions	7,202	119,292	1	16.6	197

3.2 TREC-based Dataset

Our TREC-based dataset uses the queries from the TREC Session tracks 2012–2014 [4], from the TREC Tasks tracks 2015 and 2016 [14], and from the Webis-TRC-12 [9]. The Webis-TRC-12 is based on the search logs of writers who wrote essays on the 150 topics used at the TREC Web tracks 2009–2011 while doing their background research using a search engine (13,881 submitted queries, 3848 unique). At the TREC Session tracks, 4666 queries (3248 unique) were collected as user search sessions on 60 different topics. The TREC Tasks tracks 2015 and 2016 each had 50 topics with 547 and 405 unique queries, respectively. We merged the 7771 unique queries from all the above sources and manually checked whether some of the potentially 310 tasks are identical, resulting in 276 tasks (28 queries per task). We again collected 30,707 query suggestions from Google and 9,036 from Bing, resulting in 47,514 unique queries for 276 tasks (172 queries per task).

3.3 WikiHow-based Dataset

Our third dataset is based on crawling 198,163 questions from wikiHow,[3] inspired by Yang and Nyberg's idea of extracting steps for completing task-based search intents from the procedural knowledge collected at this platform [33]. However, we do not aim to extract steps, but to identify different questions on the same task.

On wikiHow, each question is linked to other recommended questions, but spot checks showed that only those questions that mutually link to each other can be considered as on the same task such that we restrict the extraction to these cases. This way, we gathered 15,914 questions split into 7202 tasks. As before, we enlarge the dataset by obtaining 103,369 suggestions from Google and 9 additional ones Bing (for these long and specific questions, the suggestions were usually identical) for every question; this results in 119,292 queries for 7202 tasks (17 queries per task).

4 EXPERIMENTAL ANALYSIS

We compare four straightforward query-task mapping methods on our new benchmarking datasets with respect to their accuracy and efficiency, both in terms of preprocessing and online query processing.[4] Table 2 summarizes the results.

[2] For instance, for the original query [how to open a can], we would discard [how to open a canadian bank account] if returned as a suggestion.

[3] www.wikihow.com

[4] Experiment machines had Intel Xeon 2608L-v4 CPUs and 128GB of DDR4 memory

4.1 Query-Task Mapping Approaches

Our experiments take inspiration from the taxonomy of Metzler et al. [26] to cover a range of different short-text retrieval paradigms: (1) a Trie-based approach (lexical match on a surface representation), (2) MinHash LSH (stemmed representation), (3) word movers distance in a Word2Vec setup (expanded representation), and (4) an inverted index-based approach (probabilistic matching).

Trie-based Approach. The trie data structure, first described by De La Briandais [5], matches strings based on prefixes. We construct a trie for all queries within the task-split dataset during pre-processing, and for query-task mapping assign a new query q to the task associated with the query found as the longest prefix of q. If queries from multiple tasks qualify, we choose the majority vote. We use the implementation from the Google Pygtrie library.[5]

MinHash LSH. MinHash (the min-wise independent permutations locality sensitive hashing scheme) is a technique for estimating the similarity between sets via representation as a compact signature. During preprocessing, we hash the queries' binary term vectors. To efficiently find the most similar entries for a new query, we employ the implementation from the datasketch library[6] which combines MinHash with Bawa et al.'s Locality-Sensitive Hashing (LSH) Forest scheme [2].

Word Movers Distance. Using word embeddings, the word movers distance [16] measures the distance of two strings (i.e., queries in our case) as the minimum distance that the embedded words of one string need to "travel" to reach the embedded words of the other string. We employ the pre-trained Word2Vec embeddings [27] from the publicly available GoogleNews-vectors-negative300 corpus to embed all terms in a query, which is then assigned to the task of its closest WMD neighbor. We use the Fast Word Mover's Distance implementation from the wmd-relax library.[7]

Index-based Search. As a final approach, we use an inverted index-based method, whereby we store the queries in the log in an Elasticsearch index, with a field for their task.[8] To perform query-task mapping for a new query, we simply submit it to the index, and assign the task of the top result.

4.2 Pre-Processing and Mapping Efficiency

As for the pre-processing efficiency, we just measured the time needed to build the necessary data structures on the full datasets. Building the trie took about 10 seconds for the smaller datasets and 25 seconds for the largest. This is very close to the time needed to look up all query terms in the pre-trained word-embedding model for the WMD method, but quite a bit faster than computing the hashes for MinHash LSH Forest, which takes about one minute for the smaller datasets and three minutes for the largest. Building the inverted indexes takes about 30 seconds for the smaller corpora and about one minute for the largest.

Query-task mapping runtime was averaged over 10,000 test queries left out from the pre-processing. Mapping a query to its tasks is a matter of milliseconds using the trie approach or MinHash LSH Forest. Compared to these runtimes, using WMD it took 23 seconds on average to map a single query to its task on the largest dataset—prohibitively slow for an online setup without any further efficiency tweaks that were beyond the scope of our study. Using the index-based method, determining the task of a query again only takes a few milliseconds.

Table 2: Summary of our experimental results. Accuracy values shown with 95% confidence intervals.

Dataset	Trie	LSH	WMD	Index
Preprocessing time (entire dataset)				
Session-based	10.03s	53.79s	9.60s	24.14s
TREC-based	13.26s	62.09s	11.14s	26.90s
Wikihow-based	28.00s	141.65s	26.50s	53.48s
Query-task mapping time (per query)				
Session-based	0.46ms	2.42ms	7.16s	2.80ms
TREC-based	0.51ms	2.50ms	9.24s	2.95ms
Wikihow-based	0.33ms	2.28ms	22.65s	4.21ms
Query-task mapping accuracy				
Session-based	$0.69^{\pm0.02}$	$0.66^{\pm0.02}$	$0.67^{\pm0.03}$	$0.78^{\pm0.03}$
TREC-based	$0.66^{\pm0.03}$	$0.68^{\pm0.03}$	$0.73^{\pm0.03}$	$0.80^{\pm0.03}$
Wikihow-based	$0.48^{\pm0.02}$	$0.41^{\pm0.02}$	$0.55^{\pm0.03}$	$0.63^{\pm0.02}$

4.3 Query-Task Mapping Accuracy

We measure accuracy on every dataset as the ratio of correct task mappings across 50 runs of 100 independently sampled test queries in a leave-one-out manner: each test query is removed from its task individually, the datasets without that one query are pre-processed, and the methods are asked to map the now "new" query to a task. Overall, our approaches map at least one in three, and at most four out of five test queries to the correct task. The index-based method clearly performs best on all three datasets while the slow WMD approach is second best twice.

Out of our three datasets, the smaller Session- and TREC-based ones pose easier query-task mapping problems, with all methods getting at least two thirds of the test queries correct. This is explained in part by the smaller datasets having fewer tasks, and comparatively more queries per task; beyond that, previous research on one of the underlying query logs [9] found related queries to often share prefixes, boosting not just the Trie-based method, but the other exact-word-match based ones (Index and LSH), as well.

By contrast, all four methods exhibit their worst query-task mapping performance on the WikiHow-based dataset—the largest both in terms of tasks and total number of queries, but with the smallest average number of queries per task. The fact that the distributional similarity (rather than exact match) based WMD method declines comparatively less in accuracy here points to the prevalence of tasks with less-directly related queries, and some spot checks in the data bear this out: queries with the same task often share synonyms, rather than exactly identical terms.

To elaborate on this insight, Figure 1 shows the results of an additional experiment on two of our datasets. Here, we retrieve the top k (where $1 \leq k \leq 11$) results with each method, and then assign

[5] https://github.com/google/pygtrie

[6] https://ekzhu.github.io/datasketch/lsh.html

[7] https://github.com/src-d/wmd-relax

[8] https://www.elastic.co/, version 5.6.14, retrieval model: BM25

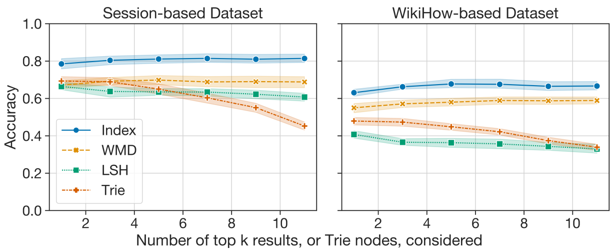

Figure 1: Query-task mapping accuracy under a majority voting scheme. Bands show 95% confidence intervals.

the majority task. Since the Trie data structure does not induce any ordering among the results in the same trie node, we use the deepest k trie nodes on the path to the input query that contain a larger number of individual result queries; with increasing k the Trie method thus approaches a simple majority (or Zero-Rule) classifier. On the less noisy Session-based dataset, this has a more detrimental effect on the Trie method's accuracy. Conversely, the Index and WMD methods benefit a bit more from a majority vote among a few selectively chosen top results on the noisier WikiHow-based dataset than they do on the Session-based one.

5 CONCLUSION

We consider the problem of query-task mapping: given a query log split into tasks and a new query, identify the most appropriate task for the query. This problem is not as well studied as the problem of splitting a log into tasks while also larger datasets of task-split queries are missing. To close this gap and to foster research on query-task mapping, our first main contribution are three large publicly available benchmarking datasets (two with about 50,000 queries and one with about 120,000 queries annotated with tasks). As our second contribution, we compare accuracy and efficiency of four mapping approaches on these datasets in a preliminary study. Our experiments show that an inverted index-based method forms a strong baseline (accuracy above 0.6 at under 6 milliseconds per query).

Interesting directions for future research include the development of more accurate fast methods, and the generalization of our experiments to even larger datasets. Such larger datasets will most likely contain highly similar tasks that turned out to be the hardest for the tested baselines to distinguish; all methods performed worse on the bigger corpus compared to the smaller ones. In our experiments, all queries had an annotated ground truth task that was also shared by other queries. Also including queries not part of any task may form an interesting addition to the experimental setup (mapping methods should then return that the query is unrelated to any known task). For instance, the index-based method could employ a retrieval score threshold as a guidance in that direction.

REFERENCES

[1] Ahmed Hassan Awadallah, Ryen W. White, Patrick Pantel, Susan T. Dumais, and Yi-Min Wang. 2014. Supporting complex search tasks. In *Proceedings of CIKM 2014*, 829–838.

[2] Mayank Bawa, Tyson Condie, and Prasanna Ganesan. 2005. LSH Forest: Self-tuning indexes for similarity search. In *Proceedings of WWW 2005*, 651–660.

[3] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. 2008. The query-flow graph: Model and applications. In *Proceedings of CIKM 2008*, 609–618.

[4] Ben Carterette, Evangelos Kanoulas, Mark M. Hall, and Paul D. Clough. 2014. Overview of the TREC 2014 Session track. In *Proceedings of TREC 2014*.

[5] Rene De La Briandais. 1959. File searching using variable length keys. In *Proceedings of IRE-AIEE-ACM 1959*, 295–298.

[6] Debora Donato, Francesco Bonchi, Tom Chi, and Yoëlle S. Maarek. 2010. Do you want to take notes?: Identifying research missions in Yahoo! search pad. In *Proceedings of WWW 2010*, 321–330.

[7] Daniel Gayo-Avello. 2009. A survey on session detection methods in query logs and a proposal for future evaluation. *Information Sciences* 179, 12 (2009), 1822–1843.

[8] Matthias Hagen, Jakob Gomoll, Anna Beyer, and Benno Stein. 2013. From search session detection to search mission detection. In *Proceedings of OAIR 2013*, 85–92.

[9] Matthias Hagen, Martin Potthast, Michael Völske, Jakob Gomoll, and Benno Stein. 2016. How writers search: Analyzing the search and writing logs of non-fictional essays. In *Proceedings of CHIIR 2016*, 193–202.

[10] Daqing He, Ayse Göker, and David J. Harper. 2002. Combining evidence for automatic web session identification. *Information Processing & Management* 38, 5 (2002), 727–742.

[11] Wen Hua, Yangqiu Song, Haixun Wang, and Xiaofang Zhou. 2013. Identifying users' topical tasks in web search. In *Proceedings of WSDM 2013*, 93–102.

[12] Bernard J. Jansen, Amanda Spink, Chris Blakely, and Sherry Koshman. 2007. Defining a session on web search engines. *JASIST* 58, 6 (2007), 862–871.

[13] Rosie Jones and Kristina Lisa Klinkner. 2008. Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs. In *Proceedings of CIKM 2008*, 699–708.

[14] Evangelos Kanoulas, Emine Yilmaz, Rishabh Mehrotra, Ben Carterette, Nick Craswell, and Peter Bailey. 2017. TREC 2017 Tasks track overview. In *Proceedings of TREC 2017*.

[15] Alexander Kotov, Paul N. Bennett, Ryen W. White, Susan T. Dumais, and Jaime Teevan. 2011. Modeling and analysis of cross-session search tasks. In *Proceedings of SIGIR 2011*, 5–14.

[16] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *Proceedings of ICML 2015*, 957–966.

[17] Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, and Hongyuan Zha. 2014. Identifying and labeling search tasks via query-based Hawkes processes. In *Proceedings of KDD 2014*, 731–740.

[18] Zhen Liao, Yang Song, Yalou Huang, Li-wei He, and Qi He. 2014. Task trail: An effective segmentation of user search behavior. *IEEE Trans. Knowl. Data Eng.* 26, 12 (2014), 3090–3102.

[19] Zheng Lu, Hongyuan Zha, Xiaokang Yang, Weiyao Lin, and Zhaohui Zheng. 2013. A new algorithm for inferring user search goals with feedback sessions. *IEEE Trans. Knowl. Data Eng.* 25, 3 (2013), 502–513.

[20] Claudio Lucchese, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Gabriele Tolomei. 2011. Identifying task-based sessions in search engine query logs. In *Proceedings of WSDM 2011*, 277–286.

[21] Claudio Lucchese, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Gabriele Tolomei. 2013. Discovering tasks from search engine query logs. *ACM Trans. Inf. Syst.* 31, 3 (2013), 14.

[22] Rishabh Mehrotra, Prasanta Bhattacharya, and Emine Yilmaz. 2016. Deconstructing complex search tasks: A Bayesian nonparametric approach for extracting sub-tasks. In *Proceedings of NAACL 2016*, 599–605.

[23] Rishabh Mehrotra and Emine Yilmaz. 2015. Terms, topics & tasks: Enhanced user modelling for better personalization. In *Proceedings of ICTIR 2015*, 131–140.

[24] Rishabh Mehrotra and Emine Yilmaz. 2017. Extracting hierarchies of search tasks & subtasks via a Bayesian nonparametric approach. In *Proceedings of SIGIR 2017*, 285–294.

[25] Rishabh Mehrotra and Emine Yilmaz. 2017. Task embeddings: Learning query embeddings using task context. In *Proceedings of CIKM 2017*, 2199–2202.

[26] Donald Metzler, Susan T. Dumais, and Christopher Meek. 2007. Similarity measures for short segments of text. In *Proceedings of ECIR 2007*. 16–27.

[27] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv* abs/1301.3781 (2013).

[28] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A picture of search. In *Proceedings of Infoscale 2006*, 1.

[29] Procheta Sen, Debasis Ganguly, and Gareth J. F. Jones. 2018. Tempo-lexical context driven word embedding for cross-session search task extraction. In *Proceedings of NAACL 2018*. 283–292.

[30] Amanda Spink, Minsoo Park, Bernard J. Jansen, and Jan O. Pedersen. 2006. Multitasking during web search sessions. *Inf. Process. Manage.* 42, 1 (2006), 264–275.

[31] Manisha Verma and Emine Yilmaz. 2014. Entity oriented task extraction from query logs. In *Proceedings of CIKM 2014*, 1975–1978.

[32] Manisha Verma and Emine Yilmaz. 2016. Category oriented task extraction. In *Proceedings of CHIIR 2016*, 333–336.

[33] Zi Yang and Eric Nyberg. 2015. Leveraging procedural knowledge for task-oriented search. In *Proceedings of SIGIR 2015*, 513–522.