Modeling User Search Tasks with a Language-Agnostic Unsupervised Approach

Luis Lugo(✉), Jose G. Moreno, and Gilles Hubert

IRIT UMR 5505, CNRS,U. de Toulouse, Toulouse, France
{luis.lugo,jose.moreno,gilles.hubert}@irit.fr

Abstract. Conversational information seeking is a major emerging research area because of the increasing popularity of conversational AI systems users utilize to perform their search tasks. Search systems and multiple other user supporting applications benefit from modeling the search tasks users carry out to satisfy their information needs. Most existing search task modeling methods are monolingual, and few methods leverage user clicks even though clicked URLs are crucial for modeling user intent. We propose a language-agnostic, user intent aware approach to model search tasks from user interactions with search systems. The proposed approach leverages user intent modeling from clicked query-document pairs, latent representations of queries in a language-agnostic space, and graph-based clusteringto model search tasks in an unsupervised approach. Experimental results demonstrate the proposed approach outperforms recent work in search task modeling, supporting user queries in multiple languages. It can also produce search task modeling results in the order of milliseconds, an essential aspect for conversational systems and user support applications requiring realtime results.

Keywords: Conversational search · User intent modeling · Language-agnostic query representation

1 Introduction

Conversational AI systems are becoming increasingly popular because of advances in speech recognition, natural language understanding, text-to-speech synthesis, and the availability of digital personal assistants [15,24,26]. Personal assistants like Amazon Alexa, Apple Siri, Google Assistant, and Microsoft Cortana are now available in smartphones, tablets, desktops, and dedicated smart speakers [15,30]. Consequently, the increasing popularity and availability of conversational systems make conversational information seeking a major emerging area of research [1,30].

In conversational information seeking and other search systems, modeling the search tasks that users perform to satisfy their information needs is a crucial step [18,22]. Search task modeling is a step in the process to make search

D. Hiemstra et al. (Eds.): ECIR 2021, LNCS 12656, pp. 405–418, 2021.
https://doi.org/10.1007/978-3-030-72113-8_27

systems more coherent, natural, engaging, and conversational [15,21,26]. Multiple user supporting applications benefit from search task modeling, including conversational question suggestion, personalization in e-commerce, product recommendations, query term prediction, query suggestions, query reformulation, and results ranking [13,20,21,23,27]. Even informative conversations with digital assistants can benefit from correctly modeling the search tasks, as the subjective perception of the quality in the conversation is strongly related to the accurate tracking of the topic [26].

Users around the world access search systems in multiple languages, making it essential to process users' requests with language-agnostic models. Also, search systems and user supporting applications require realtime responses when processing user information needs. For instance, multimodal search in conversational systems runs multiple processes in parallel, post-processing their outputs to generate a message answering the user request; hence, modeling can not exceed the timeout periods set on the search system [30]. Similarly, user clicks are strongly related to the user intent [31]. Different queries with similar clicked URLs can pertain to the same information need [20], and analyzing clicked URLs can help disambiguate queries [5].

Our contributions are threefold. First, we propose a language-agnostic search task modeling (LASTM) approach to model search tasks from user interactions with search systems. Second, given the relationship between clicked URLs and user intent, we propose a user intent modeling technique leveraging a large scale query - clicked document collection in the query latent space. Third, to enable the utilization of LASTM in conversational search systems and user supporting applications requiring responses on the fly, we propose a realtime method for mapping incoming queries to the modeled user search tasks directly on the query latent space.

2 Related Work

Mining user interactions with search systems enable modeling the search tasks that users perform to satisfy their information needs [13]. In particular, search query logs can be mined for search task modeling using methods such as heuristics-based models, semi-supervised clustering, Bayesian approaches, and graph-based clustering. A model based on a cascade of heuristics [12] first segments the search query log in logical sessions; then, it performs a post-processing step to detect search tasks based on the queries pertaining to the logical sessions. However, several manually set thresholds in the heuristics make it challenging to adapt heuristics-based models to other datasets without manually adjusting them.

Semi-supervised clustering approaches combine a supervised component and an unsupervised method to model search tasks. Bestlink SVM [28] first trains a support vector machine to detect if a pair of adjacent queries from a user pertains to the same search tasks or not. Then, it clusters the related queries in the search log using the SVM output to establish links between queries.

Bestlink SVM uses a backward context from users' queries to improve the task clustering results. The two most important features from the query representation in Bestlink SVM includes the cosine similarity between query embeddings and the similarity between clicked URLs. Context Attention based LSTM (CA-LSTM) [9] relies on recurrent neural networks instead of SVMs, using both backward and forward queries to provide context while training the neural network to detect if a pair of adjacent queries pertain to the same task or not. CA-LSTM then uses graphs to cluster related queries.

Bayesian approaches include Latent Dirichlet Allocation with Hawkes processes (LDA-Hawkes) [16], Distance Dependent Chinese Restaurant Process (DD-CRP) [19], and Bayesian Rose Trees (BRTs) [20]. LDA-Hawkes combines LDA with Hawkes processes to identify and label search tasks from query logs. LDA performs topic modeling, identifying semantically related queries from different users, while Hawkes processes take into account time lapses between query timestamps in individual query sequences, assigning temporally close queries to the same search task. DD-CRP extracts a single-level hierarchy of tasks from query logs, linking related queries by word embedding distances. DD-CRP assumes a restaurant with an infinite number of tables. Customers enter the restaurant in tandem; they are assigned to a nonempty table based on the number of existing customers in the table, or an empty table depending on a hyperparameter. Entries in the query log are customers, while search tasks are tables. BRTs extend the single level hierarchy of DD-CRP to multiple levels, modeling search tasks by clustering related nodes in the hierarchical structure.

Graph-based clustering is used in several approaches for search task modeling [17,18,22]. QC-WCC [17] builds a graph where nodes correspond to queries, and edges are weighted according to the similarities between queries. Similarities are based on two features: one content-based from Jaccard similarity on tri-grams, and the other semantic-based exploiting Wikipedia and Wiktionary to infer the semantics. QC-HTC [17] is a computationally simpler algorithm based on QC-WCC, although less accurate. It exploits the sequential nature of queries to decrease the computational complexity of the graph based method. QC-HTC first builds sequences of queries according to the distance between them, creating the first set of clusters, then takes the first and last queries of the sequence to represent a cluster and group it with other clusters depending on query distances. Using only the first and last queries in each cluster avoids the computation of the full similarity graph required for QC-WCC, making QC-HTC less computationally expensive.

QRY-VEC [22] improves over the QC-WCC algorithm using word embedding similarities instead of lexically based similarities. Queries for the same task clusters tend to be semantically similar rather than lexically similar, as queries in the same tasks contain more synonym words than exact words [17,27]. Because of this, instead of relying on lexically based similarities and retrieved documents from the Wikipedia collection, QRY-VEC uses the cosine similarity on tempo-lexical word embeddings and documents retrieved from the ClueWeb12B collection [3]. Multilingual Graph-Based Clustering (MGBC) [18] outperforms

previous models by combining a multilingual query encoding with graph-based clustering, supporting queries in several languages through the use of the Multilingual Universal Sentence Encoder (MUSE) [29]

However, most search task modeling methods [9,12,16,17,19,20,22,28] are monolingual. Although MGBC supports several languages through MUSE, it can only process queries in sixteen languages. Additionally, when using ClueWeb12B for calculating query similarities, MGBC can only support user queries in English. By the same token, most search task modeling methods [9,16–19,22] fail to take into account clicked URLs when processing search query logs, even though clicked URLs have a critical correlation to the user intent [31]. Also, conversational information seeking systems and multiple applications supporting users search efforts require results on the fly. Building models from scratch when a user submits a query could create large processing times, forcing search systems to trigger timeout intervals [30]. Similarly, waiting for forward queries to provide context [9] can render models unfeasible in realtime setups. Also, some models requiring user identifiers [9,12,16,20] can not be used in user-independent [5,18,22] modeling scenarios.

3 User Search Task Modeling

LASTM is an unsupervised method that leverages latent representations of queries in a language-agnostic space, user intent modeling from clicked query-document pairs, and graph-based clustering to model user search tasks. It can also produce a realtime mapping of queries to modeled search tasks. In contrast with previous work [9,12,16,17,19,20,22,28], our proposed approach supports multiple languages through a language-agnostic latent space. The proposed approach is also independent of user identifiers, enabling the modeling of search tasks in both user-independent and personalized scenarios. It also differs from some prior methods [9,16–19,22] by leveraging clicked URLs to model user intent [31] in the query latent space.

3.1 Language-Agnostic Query Representation

Users worldwide submit queries in different languages to satisfy their information needs. Language-agnostic BERT Sentence Embedding (LABSE) [10] provides the sentence embeddings to represent user queries in a language-agnostic latent space. Using a 12-layer transformer architecture [7,25] in a dual configuration, LABSE takes the transformer's hidden state for the last token in the sentence to generate the query representation.

The query representation using LABSE has the ability to perform zero-shot cross-lingual transfer, supporting queries in languages that are not part of the training dataset. When performing tests with the TAOEBA dataset [2], LABSE obtains an 83.7% accuracy, while the baseline Language-agnostic Sentence Representations [2] gets 65.5%, even though more than 30 languages in the TAOEBA dataset were not part of the LABSE training data [10].

We use the cosine proximity [10,22] to compute the similarity between query representations in the language-agnostic latent space. Formally, given a pair of queries q_i, q_j with latent representations u_i, u_j, the similarity between query representations S_{lat} is calculated as follows [10,22]:

$$S_{lat}(u_i, u_j) = \frac{u_i u_j}{|u_i|\,|u_j|} \tag{1}$$

3.2 User Intent Modeling

User clicks play a critical role in modeling user intent – the information need the user wants to satisfy by performing the search task [31]. Query term match between queries for the same information need can be very low; even lexically different queries pertaining to the same search task can have similar clicked URLs [20,31]. Also, analysis of clicked URLs can help disambiguate queries, revealing which documents users clicked when performing their search tasks [5].

To model user intent, we use the Open Resource for Click Analysis in Search (ORCAS) [5], a collection containing 18.8 million clicked document - query pairs for 10.4 million unique queries. Clicked documents are represented using the TREC document identifier in the TREC Deep Learning document collection [6]. We encode queries in ORCAS in the language-agnostic latent space [10], creating a user intent database $\mathcal{D}_M$ with clicked document - query pairs. To retrieve the most relevant documents for a given user query in the database, we use Scalable Nearest Neighbor (ScaNN) [11], a state-of-the-art method for large-scale retrieval tasks. ScaNN performs maximum inner product search (MIPS) using an anisotropic vector quantization, which allows a fast rate of document scoring.

Even though ORCAS has queries exclusively in English, doing MIPS directly on the language-agnostic latent space enables user intent modeling in any language LABSE can support. Hence, we can leverage the existing relationship between clicked URLs and user intent [31] by searching the $\mathcal{D}_M$ database.

Formally, given a database $\mathcal{D}_M = \{m_i\}_{i=1,2,\ldots,n}$ formed from a clicked query-document dataset $\mathcal{D}_Q$ with n data points, where each data point $m_i \in \mathbb{R}^p$ is the latent representation of the query $q \in \mathcal{D}_Q$ in the p-dimensional language-agnostic latent space, we want to find the most relevant documents $\{d_j\}_{j=1,2,\ldots,k} \in \mathcal{D}_M$ for the user query $u \in \mathbb{R}^p$. Therefore, we search for the k points with the maximum inner product with the user query u as follows [11]:

$$MIPS(\mathcal{D}_M, u) = \{d_j\}_{j=1,2,\ldots,k} = \arg\max_{m_i \in \mathcal{D}_M} \langle u, m_i \rangle \tag{2}$$

Given a user query pair q_i, q_j with latent representations u_i, u_j, the similarity based on user intent S_{int} is calculated using the Jaccard coefficient for the top thousand relevant documents in the database $\mathcal{D}_M$ [18,22]:

$$D_i = MIPS(\mathcal{D}_M, u_i) \tag{3}$$

$$D_j = MIPS(\mathcal{D}_M, u_j) \tag{4}$$

$$S_{int}(u_i, u_j) = \frac{|D_i \cap D_j|}{|D_i \cup D_j|} \tag{5}$$

3.3 Unsupervised Search Task Modeling

We now integrate user intent modeling and language-agnostic query representations with graph-based clustering [4] to model search tasks (Algorithm 1). First, we encode queries in the latent space (Section 3.1); every query embedding becomes a node in the weighted graph. Then, we compute the similarities between pairs of queries to create the edges of the weighted graph. The similarity between queries S_{qry} is a convex combination of the similarity in the latent space S_{lat} and the similarity based on user intent S_{int}. Given a pair of queries q_i, q_j with latent representations u_i, u_j, query similarity S_{qry} is calculated as follows [18]:

$$S_{qry}(u_i, u_j) = \alpha * S_{lat}(u_i, u_j) + (1 - \alpha) * S_{int}(u_i, u_j) \tag{6}$$

After finishing edge weight calculations, we prune the weighted graph, deleting edges with $S_{qry} < \eta$. The resulting connected components $\mathcal{C}$ in the graph constitute the search tasks, so we assign a unique task label $task_i$ to every connected component. All the queries pertaining to a connected component receive the same task label. A grid search optimizes parameters η and α, using $\eta = k/10, \alpha = k/10, 0 < k \leq 10, k \in \mathbb{N}$ [4,17,18,22].

3.4 Realtime Mapping of New Queries

Most search systems and user supporting applications require results in realtime. Applications like contextual topic modeling in conversational search [15], query suggestion, or query reformulation can not afford to wait for large processing times. It is essential to return an answer in a few milliseconds. Hence, once the user performs a search request, we map the new incoming query to the labels extracted with Algorithm 1 so that we can model the search task in realtime. To do the mapping, we use the same MIPS method with anisotropic vector quantization [11] that we used in Section 3.2.

The search task database maps the latent representation of the queries in the search log $\mathcal{Q}_L$ to the extracted task labels $\mathcal{L}_T$. Formally, given a database $\mathcal{Q}_T = \{m_i\}_{i=1,2,\ldots,n}$ formed from the search query log $\mathcal{Q}_L$ with search task labels $\mathcal{L}_T$ returned from Algorithm 1, where each datapoint $m_i \in \mathbb{R}^p$ is the latent representation of the query $q \in \mathcal{Q}_L$ in the p-dimensional language-agnostic space. For an incoming query q_i, we compute the latent representation u_i; then, we retrieve the search task labels T of the k closest queries in the language-agnostic latent space using MIPS:

$$T = MIPS(\mathcal{Q}_T, u_i) \tag{7}$$

Once we have the search task labels T of the k closest queries, we return the task label with the highest number of occurrences in T.

Algorithm 1. LASTM

Inputs: Search query log $\mathcal{Q}_L$, Clicked query-document collection $\mathcal{D}_Q$
Output: Task labels $\mathcal{L}_T$

// Build database for user intent
$\mathcal{D}_M \leftarrow \{\}$
for all $q_i, d_i \in \mathcal{D}_Q$ **do**
  $x_i \leftarrow language\_agnostic\_space(q_i)$
  $\mathcal{D}_M \leftarrow \mathcal{D}_M \cup \{x_i, d_i\}$
end for

// Model search tasks
$V \leftarrow \{\}, E \leftarrow \{\}, G(V, E) \leftarrow (V, E)$
for all $q_i \in \mathcal{Q}_L$ **do**
  $u_i \leftarrow language\_agnostic\_space(q_i)$
  $V \leftarrow V \cup \{u_i\}$
end for

for all $v_i, v_j \in V$ **do**
  $S_{lat}(v_i, v_j) = cos(v_i, v_j)$
  $D_i, D_j \leftarrow$ document IDs for v_i, v_j from $\mathcal{D}_M$
  $S_{int}(v_i, v_j) = Jaccard(D_i, D_j)$
  $\mathbf{e}_k \leftarrow \alpha * S_{lat}(v_i, v_j) + (1 - \alpha) * S_{int}(v_i, v_j)$
  $E \leftarrow E \cup \{\mathbf{e}_k\}$
end for

for all $\mathbf{e}_k \in E$ **do**
  if $\mathbf{e}_k < \eta$ **then**
    $E \leftarrow E \setminus \{\mathbf{e}_k\}$
  end if
end for

for all $\mathcal{C}_i \in G(V, E)$ **do**
  $task_i \leftarrow i$
  for all $v_k \in \mathcal{C}_i$ **do**
    $\mathcal{L}_T[v_k] \leftarrow task_i$
  end for
end for

return $\mathcal{L}_T$

4 Results and Discussion

In this section, we analyze LASTM in user independent search task modeling and realtime mapping of incoming queries. Following previous work [9,18], we calculate model performance with the F_β score:

$$F_\beta = \frac{(1+\beta^2) * p * r}{\beta^2 * p + r} \tag{8}$$

where p is precision and r is recall. We consider both $\beta = 1.0$ and $\beta = 0.6$ [9], which gives more weight to the precision of the model. The Student's paired t-test provides statistical significance calculations [31].

We use open source implementations for ScaNN[1], NetworkX[2] in graph-based clustering, and the publicly available pretrained model for LABSE.[3]

Table 1. Search task modeling results for the CSTE dataset in all the languages supported by the MGBC method. Differences between MGBC and LASTM results have $p \leq 0.05$ for the Student's t-test.

Language	ISO 639-1	F_1		$F_{0.6}$	
		MGBC	LASTM	MGBC	LASTM
Arabic	ar	0.447	**0.521**	0.395	**0.490**
Chinese PRC	zh	0.480	**0.539**	0.473	**0.513**
Chinese Taiwan	zh-tw	0.482	**0.540**	0.476	**0.515**
Dutch	nl	0.449	**0.534**	0.431	**0.511**
English	en	0.456	**0.538**	0.437	**0.512**
German	de	0.450	**0.533**	0.432	**0.511**
French	fr	0.484	**0.539**	**0.547**	0.512
Italian	it	0.452	**0.540**	0.434	**0.517**
Portuguese	pt	0.458	**0.537**	0.438	**0.514**
Spanish	es	0.450	**0.541**	0.432	**0.516**
Japanese	ja	0.453	**0.522**	0.436	**0.495**
Korean	ko	0.451	**0.523**	0.396	**0.501**
Russian	ru	0.449	**0.533**	0.429	**0.508**
Polish	pl	0.460	**0.536**	**0.524**	0.512
Thai	th	0.444	**0.522**	0.427	**0.489**
Turkish	tr	0.429	**0.538**	0.401	**0.513**

4.1 Search Task Modeling

The Cross-Session Task Extraction (CSTE) dataset [22] and the Complex User Search Task Analysis (CUSTA) dataset [8] are used for experiments. CSTE has 1424 entries with 224 ground truth labels corresponding to cross-session search tasks. CUSTA has 2390 entries with 15 ground truth search task labels. As a

[1] https://github.com/google-research/google-research/tree/master/scann.
[2] https://networkx.github.io/.
[3] https://tfhub.dev/google/LaBSE/1.

baseline, we use MGBC, a state-of-the-art method for search task modeling, calculating metrics for all the languages supported by the baseline. Queries in the CSTE dataset are in English, while queries in the CUSTA dataset are mostly in French, with very few English entries. Hence, we perform machine translation with the Google Cloud Translation API[4] for evaluating LASTM in all the languages supported by MGBC.

Table 2. Search task modeling results for the CUSTA dataset in all the languages supported by the MGBC method. Differences between MGBC and LASTM results have $p \leq 0.05$ for the Student's t-test.

Language	ISO 639-1	F_1		$F_{0.6}$	
		MGBC	LASTM	MGBC	LASTM
Arabic	ar	0.595	**0.608**	0.648	**0.665**
Chinese PRC	zh	0.658	**0.667**	0.667	**0.688**
Chinese Taiwan	zh-tw	0.632	**0.672**	0.604	**0.694**
Dutch	nl	0.594	**0.648**	0.577	**0.761**
English	en	0.597	**0.657**	0.544	**0.705**
German	de	0.550	**0.642**	0.542	**0.715**
French	fr	0.656	**0.732**	0.748	**0.750**
Italian	it	0.559	**0.604**	0.492	**0.602**
Portuguese	pt	0.616	**0.622**	0.610	**0.636**
Spanish	es	0.641	**0.643**	0.593	**0.712**
Japanese	ja	**0.697**	0.619	**0.737**	0.571
Korean	ko	**0.573**	0.563	**0.639**	0.561
Russian	ru	0.633	**0.641**	0.742	**0.754**
Polish	pl	0.541	**0.598**	0.578	**0.605**
Thai	th	0.541	**0.603**	0.533	**0.636**
Turkish	tr	0.618	**0.653**	0.640	**0.711**

The proposed approach improves the search task modeling performance of the baseline method in the two datasets used for testing. Using the CSTE dataset (Table 1), LASTM surpasses MGBC in all the languages supported by the baseline, obtaining up to 10.9% ($p \leq 0.05$) improvement in the F_1 score for the Turkish language; similarly, LASTM obtains better $F_{0.6}$ scores in fourteen out of sixteen languages, getting an improvement of up to 11.2% ($p \leq 0.05$) in the Turkish language. Furthermore, the monolingual QRY-VEC method, which supports queries in English, obtains an F_1 score of 0.538 and an $F_{0.6}$ score of 0.488 [22]. Consequently, there is no loss in modeling performance when comparing LASTM to the QRY-VEC method. For the CUSTA dataset (Table 2),

[4] https://cloud.google.com/translate.

we observe improvements in fourteen out of the sixteen languages supported by MGBC; LASTM generates up to 9.2% ($p \leq 0.05$) improvement in the F_1 score for the German language and up to 18.4% ($p \leq 0.05$) improvement in the $F_{0.6}$ score for the Dutch language.

Both the similarity between query representations S_{lat} and the similarity based on user intent S_{int} contribute to the search task modeling results. In the grid search for the CSTE dataset, α values averaged 0.238 ± 0.099. For the CUSTA dataset, α values in the grid search averaged 0.731 ± 0.157. These α values indicate that the convex combination (Eq. 6) effectively relies on the two similarities to compute the edges for the weighted graph.

From a language coverage perspective, the query representation for LASTM is trained with 109 languages and can perform zero-shot cross-lingual transfer to multiple more languages [10]. In contrast, the baseline only supports sixteen languages, making LASTM coverage at least seven times larger when considering training languages only. The improvements in modeling results and language coverage highlight the importance of considering user intent along with language-agnostic query representation for modeling search tasks.

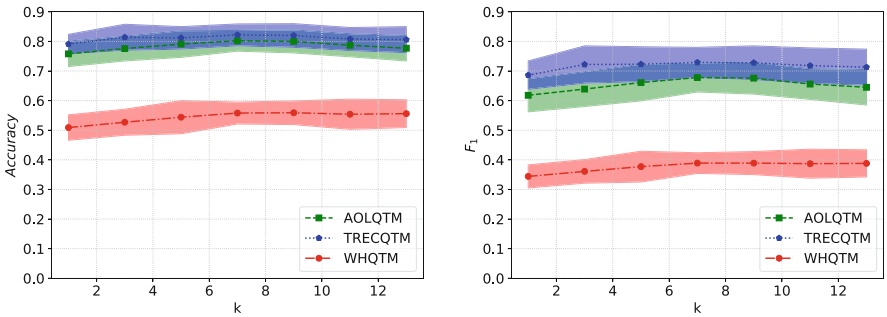

Fig. 1. Search task mapping results in the language-agnostic latent space for AOLQTM, TRECQTM, and WHQTM datasets. Results include several values of top k from the ScaNN index, considering (a) Accuracy (b) F_1.

4.2 Mapping of Incoming Queries

To analyze the performance of LASTM for mapping new incoming queries, we run the mapping method using three benchmark datasets previously proposed for query-task mapping [27]:

- AOL-based Query-Task-Mapping (AOLQTM) dataset, which has 41780 queries and labels for 1423 search tasks.
- TREC-based Query-Task-Mapping (TRECQTM) dataset, which has 47514 queries with labels for 276 search tasks.

– WikiHow-based Query-Task-Mapping (WHQTM) dataset, which has 119292 queries with labels for 7202 search tasks.

We use a leave-one-out evaluation, independently selecting one hundred random queries from the dataset and repeating the evaluation for fifty runs. Experiments run on a virtual machine instance with 8 CPUs of 3 GHz and 60 GB of RAM. Metrics include accuracy, F_1, $F_{0.6}$, and query time. To measure query time, we take the average time for mapping a single query, using 10^4 mappings to compute the average [18,27]. As a baseline, we use the MGBC approach for query task mapping. MGBC combines the Neighborhood Graph and Tree approximate nearest neighbor method [14] with the MUSE latent space for query encoding. For reference, we also include results using the Trie[5] data structure and the BM25[6] retrieval model [18,27,29].

Figure 1 depicts the optimization experiments for the number of top k results from ScaNN to consider. After running tests for $k = [1, 3, 5, 7, 9, 11, 13]$, we found that top $k = 7$ results from ScANN generates the optimal configuration, providing the best results for task mapping while keeping the time per query under a millisecond (Table 3). Low response time is an essential aspect for applications supporting users in realtime setups. Long answer times could affect the interaction of the search system with the users, especially in conversational and multimodal search systems, where a post-processing step is required to generate a response to the user request [15,30]. Similarly, long answer times could trigger internal timeout intervals [30], forcing search systems to ignore search task mapping results while doing internal post-processing.

Table 3. Realtime mapping of queries to search tasks. Differences against baseline MGBC results have $p \leq 0.05$ for the Student's t-test.

Dataset	Method	Accuracy	F_1	$F_{0.6}$	Query time
AOLQTM	Trie	0.693	0.543	0.543	0.029 ms
	BM25	**0.809**	**0.689**	**0.689**	0.947 s
	MGBC	0.751	0.608	0.607	0.308 ms
	LASTM	0.802	0.678	0.677	0.490 ms
TRECQTM	Trie	0.650	0.519	0.518	0.030 ms
	BM25	0.791	0.688	0.688	2.532 s
	MGBC	0.804	0.705	0.704	0.299 ms
	LASTM	**0.822**	**0.729**	**0.728**	0.481 ms
WHQTM	Trie	0.471	0.310	0.311	0.032 ms
	BM25	0.621	0.453	0.454	6.572 m
	MGBC	**0.648**	**0.481**	**0.481**	0.368 ms
	LASTM	0.558	0.389	0.389	0.982 ms

[5] https://github.com/google/pygtrie.
[6] https://github.com/nhirakawa/BM25.

LASTM surpasses the baseline and reference methods in the TREC-based dataset, improving the F_1 score by 2.4% ($p \leq 0.05$), while keeping processing times under a millisecond. For the AOL-based dataset, LASTM surpasses the baseline method, obtaining a 7.0% improvement in the F_1 score ($p \leq 0.05$); likewise, LASTM obtains similar results to BM25, but it is faster when comparing to the BM25 implementation used for experiments. For the WikiHow-based dataset, LASTM underperforms MGBC and BM25 (Table 3). Regarding the number of user queries per task, we find that the TREC-based dataset has an average of 28 user queries per search task, while the WikiHow-based dataset has an average of 2 user queries per task. Hence, the WikiHow-based dataset contains mostly simple tasks, which users can solve with a few queries [13]. Task mapping results suggest that LASTM is better than the baseline and reference methods when mapping search tasks containing multiple queries, while MGBC is better when mapping simple search tasks in realtime.

5 Conclusion

In this paper, we proposed LASTM, an unsupervised method for modeling search tasks from user interactions with search systems. The proposed model outperforms the baseline both in modeling performance as well as the number of languages it can support, highlighting the importance of language-agnostic latent spaces for query representation and the importance of considering clicked URLs to model user intent. Also, it is independent of user identifiers, enabling modeling search tasks in user-independent or personalized applications. The modeling performance of LASTM, its language-agnostic capacity, and its ability to support realtime modeling can benefit search systems and user supporting applications, constituting an essential step in the effort to make search more coherent, conversational, engaging, and natural. For future work, we plan to explore unsupervised alternatives for graph-based clustering to further improve search task modeling.

Acknowledgement. This work was supported by the Agence National de la Recherche (ANR), through project CoST, code ANR-18-CE23-0016.

References

1. Anand, A., Cavedon, L., Joho, H., Sanderson, M., Stein, B.: Conversational search (Dagstuhl seminar 19461). In: Dagstuhl Reports, vol. 9. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2020)
2. Artetxe, M., Schwenk, H.: Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. Trans. Assoc. Comput. Linguist. **7**, 597–610 (2019)
3. Callan, J.: The Lemur project and its ClueWeb12B dataset. In: Invited talk at the SIGIR 2012 Workshop on Open-Source Information Retrieval (2012)
4. Chen, Z., Ji, H.: Graph-based clustering for computational linguistics: a survey. In: Proceedings of the 2010 workshop on Graph-based Methods for Natural Language Processing, pp. 1–9. Association for Computational Linguistics (2010)

5. Craswell, N., Campos, D., Mitra, B., Yilmaz, E., Billerbeck, B.: ORCAS: 18 million clicked query-document pairs for analyzing search. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management. ACM (2020)
6. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the TREC 2019 deep learning track. arXiv preprint arXiv:2003.07820 (2020)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186 (2019)
8. Dosso, C., Chevalier, A., Tamine, L.: How to support search activity of users without prior domain knowledge when they are solving learning tasks? In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management. 1st International Workshop on Investigating Learning During Web Search. ACM (2020)
9. Du, C., Shu, P., Li, Y.: CA-LSTM: search task identification with context attention based LSTM. In: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1101–1104. ACM (2018)
10. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-agnostic BERT sentence embedding. arXiv preprint arXiv:2007.01852 (2020)
11. Guo, R., Sun, P., Lindgren, E., Geng, Q., Simcha, D., Chern, F., Kumar, S.: Accelerating large-scale inference with anisotropic vector quantization. In: Proceedings of the 37th International Conference on Machine Learning (2020)
12. Hagen, M., Gomoll, J., Beyer, A., Stein, B.: From search session detection to search mission detection. In: Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, pp. 85–92 (2013)
13. Hearst, M.: Search user Interfaces. Cambridge University Press, Cambridge, CB2 8BS, UK (2009)
14. Iwasaki, M., Miyazaki, D.: Optimization of indexing based on k-nearest neighbor graph for proximity search in high-dimensional data. arXiv preprint arXiv:1810.07355 (2018)
15. Khatri, C., Goel, R., Hedayatnia, B., Metanillou, A., Venkatesh, A., Gabriel, R., Mandal, A.: Contextual topic modeling for dialog systems. In: 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 892–899. IEEE (2018)
16. Li, L., Deng, H., Dong, A., Chang, Y., Zha, H.: Identifying and labeling search tasks via query-based Hawkes processes. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 731–740 (2014)
17. Lucchese, C., Orlando, S., Perego, R., Silvestri, F., Tolomei, G.: Identifying task-based sessions in search engine query logs. In: Proceedings of the 4th ACM International Conference on Web Search and Data Mining, pp. 277–286. ACM (2011)
18. Lugo, L., Moreno, J.G., Hubert, G.: A multilingual approach for unsupervised search task identification. In: The 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2041–2044. ACM (2020)

19. Mehrotra, R., Bhattacharya, P., Yilmaz, E.: Deconstructing complex search tasks: a Bayesian nonparametric approach for extracting sub-tasks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 599–605 (2016)
20. Mehrotra, R., Yilmaz, E.: Extracting hierarchies of search tasks and subtasks via a Bayesian nonparametric approach. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 285–294. ACM (2017)
21. Rosset, C., et al.: Leading conversational search by suggesting useful questions. In: Proceedings of The Web Conference 2020. pp. 1160–1170 (2020)
22. Sen, P., Ganguly, D., Jones, G.: Tempo-lexical context driven word embedding for cross-session search task extraction. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 283–292 (2018)
23. Tamine, L., Melgarejo, J.L., Pinel-Sauvagnat, K.: What can task teach us about query reformulations? In: Jose, J.M., et al. (eds.) ECIR 2020. LNCS, vol. 12035, pp. 636–650. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45439-5_42
24. Thomas, P., McDuff, D., Czerwinski, M., Craswell, N.: Expressions of style in information seeking conversation with an agent. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1171–1180 (2020)
25. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
26. Venkatesh, A., et al.: On evaluating and comparing open domain dialog systems. arXiv preprint arXiv:1801.03625 (2018)
27. Völske, M., Fatehifar, E., Stein, B., Hagen, M.: Query-task mapping. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 969–972 (2019)
28. Wang, H., Song, Y., Chang, M.W., He, X., White, R.W., Chu, W.: Learning to extract cross-session search tasks. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1353–1364. ACM (2013)
29. Yang, Y., et al.: Multilingual universal sentence encoder for semantic retrieval. In: Proceedings of the 58th Annual Meeting of the ACL: System Demonstrations, pp. 87–94. ACL (2020)
30. Zamani, H., Craswell, N.: Macaw: An extensible conversational information seeking platform. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2193–2196 (2020)
31. Zhang, H., et al.: Generic intent representation in web search. In: The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM (2019)