LaQuE: Enabling Entity Search at Scale

Negar Arabzadeh[1], Amin Bigdeli[1], and Ebrahim Bagheri[2]

[1] University of Waterloo,[2] Toronto Metropolitan University
{narabzad, abigdeli}@uwaterloo.ca; bagheri@torontomu.ca

Abstract. Entity search plays a crucial role in various information access domains, where users seek information about specific entities. Despite significant research efforts to improve entity search methods, the availability of large-scale resources and extensible frameworks has been limiting progress. In this work, we present LaQuE (Large-scale Queries for Entity search), a curated framework for entity search, which includes a reproducible and extensible code base as well as a large relevance judgment collection consisting of real-user queries based on the ORCAS collection. LaQuE is industry-scale and suitable for training complex neural models for entity search. We develop methods for curating and judging entity collections, as well as training entity search methods based on LaQuE. We additionally establish strong baselines within LaQuE based on various retrievers, including traditional bag-of-words-based methods and neural-based models. We show that training neural entity search models on LaQuE enhances retrieval effectiveness compared to the state-of-the-art. Additionally, we categorize the released queries in LaQuE based on their popularity and difficulty, encouraging research on more challenging queries for the entity search task. We publicly release LaQuE at https://github.com/Narabzad/LaQuE.

1 Introduction

The importance of entity search has grown significantly in various information access domains, where users seek to find information on specific entities such as individuals, organizations, and places and their associated attributes [24, 16, 25]. Research suggests that more than 40% of web search queries revolve around entities, prompting search engines to rely on knowledge graphs to provide relevant and reliable responses [5, 36, 44, 15]. Entity search finds applications in diverse areas, including vertical search, which may only display a limited number of entities due to space constraints on the screen; enterprise search, focusing on entities within a specific organization; and social networks, emphasizing on the relationships between people [26, 6], among others. The task of entity search is defined as retrieving a ranked list of entities from a knowledge graph, such as Wikipedia, in response to an input keyword query. The entity search task differentiates itself from the more traditional ad hoc retrieval task by capitalizing on additional knowledge graph semantics such as relations, types, and attributes [22, 57, 51, 35, 52].

The entity search task has witnessed a growing range of methods including traditional bag-of-word-based approaches to more complex methods that incorporate neural embeddings, anchor texts, structural components of Wikipedia, and associated categories, to name a few [24, 16, 57, 12, 8, 56, 26]. While there has been significant research focused on improving method performance on this task [28, 29, 20], the development of large-scale frameworks consisting of query collections and resources has not kept up the pace. One of the main reasons for this relates to the time and resource-intensive nature of identifying and maintaining a comprehensive set of user queries and their relevant entities. As recent advances in ad hoc retrieval and the experience with the MS MARCO dataset show, neural methods are data-hungry and are often effective when trained on large relevance judgement collections [42]. However, the scarcity of such resources makes it challenging to develop and benchmark strong and generalizable entity search models [42, 34, 3].

From among the available resources curated specifically for the entity search task, TREC Complex Answer Retrieval (CAR)[1] stands out in terms of its size and coverage [18, 17]. TREC CAR was designed initially for the complex question-answering task that involves retrieving a ranked list of relevant entities and their supporting passages for each section of a given complex topic query. To create the TREC CAR dataset, topics, outlines, and paragraphs were extracted from the English version of Wikipedia. In addition to the manual ground truth, automatic ground truths were also curated in the CAR collection. The automatic ground truth is released for all training sets and is determined by whether a paragraph is contained within the page/section, making it relevant, or if it is not contained, making it non-relevant. The ground truth is provided at three levels of granularity: paragraph contained in the section (hierarchical), paragraph contained in section hierarchy below the top-level section (top-level), and paragraph contained anywhere in the page (article). While the TREC CAR collection constitutes a valuable resource for the community, it *is not designed to include real-world user queries*. Given the nature of the CAR task, the queries in this collection are section titles from Wikipedia pages; therefore, these queries are not considered to be real-world user-generated queries and differ substantially in characteristics from real user queries.

There are however other available entity search datasets that have user-generated queries and manually-labelled ground truths such as the DBpedia-Entity (v1 and v2)[2] dataset, which is a widely recognized and standard test collection for evaluating entity search methods [27, 7]. The purpose of this test collection is to assess the performance of retrieval systems in generating ranked lists of entities in response to user queries expressed in free text. *The limitation of the two versions of DBpedia-Entity is that they only include 485 and 467 queries* from INEX, QALD, SemSearch and TREC Entity benchmarks, respectively. The low number of queries prevents researchers from training deep learning models for this task.

[1] `https://trec-car.cs.unh.edu/`

[2] `https://github.com/iai-group/DBpedia-Entity`

In this paper, our main focus is to propose a large-scale publicly accessible framework, called LaQuE (pronounced as lɛ́jk - layk), along with supporting code and dataset for entity search. LaQuE is based on an intuitive idea that helps with the automated generation of a large-scale dataset for entity search with two important characteristics: (1) LaQuE includes real-user queries; and, (2) LaQuE is large-scale such that neural methods can be trained and tested on it.

In order to enable these two main characteristics, we propose and implement an intuitive idea to leverage the Open Resource for Click Analysis in Search (ORCAS) dataset. ORCAS is a large-scale click-based resource curated for the TREC Deep Learning Track. The extensive set of queries within ORCAS has already facilitated research in various areas of information retrieval (IR) and natural language processing (NLP), including query autocompletion and web mining [1, 40, 41, 13, 21]. We intuitively propose that the clickthrough data between user queries and their corresponding relevant web documents in the ORCAS dataset can be considered to be a form of *pseudo-relevance feedback*. On this basis, we establish specific criteria for filtering queries from the ORCAS dataset by only considering those queries whose relevant clicked web pages refer to links pointing to Wikipedia entities. This way, we identify the subset of queries from the ORCAS dataset where the users have determined the relevant document to be a Wikipedia entity. This subset of queries are those that require the retrieval of an entity to be satisfied.

On the basis of this idea, we incorporate such queries and their relevant clicked entities within LaQuE and offer supporting methods and code to work with the query collection. LaQuE delivers over 2 million pairs of queries and clicked Wikipedia entities. To facilitate training neural models, LaQuE offers separate standard train, test and development sets. Furthermore, for benchmarking purposes, LaQuE offers implementation for state-of-the-art first-stage retrievers, ranging from traditional bag-of-words-based retrievers to more complex pre-trained neural models in order to offer out-of-the-box strong baselines.

In this paper, we empirically illustrate that training on datasets intended for other tasks, such as ad hoc retrieval, is not as effective as training neural models on the collection offered through the LaQue framework when it comes to the task of entity retrieval. In addition, we report on our detailed investigative studies on different cuts of queries offered through LaQuE. We categorize queries based on *popularity*, and *difficulty*. LaQuE offers information on entity popularity by collecting page views of the relevant entities on Wikipedia (number of times the relevant entity page was viewed on Wikipedia) and categorizes them based on the number of views they received. Based on LaQuE, and in this paper, we investigate whether popular entities are easier to retrieve and whether language models have an inherent bias towards more popular entities rather than rare ones. Finally, inspired by previous work [2, 10, 19, 4], LaQuE offers an additional categorization of entities based on their difficulty. This categorization encourages the research community to not only focus on improving overall performance but also specifically tackle more challenging queries [2].

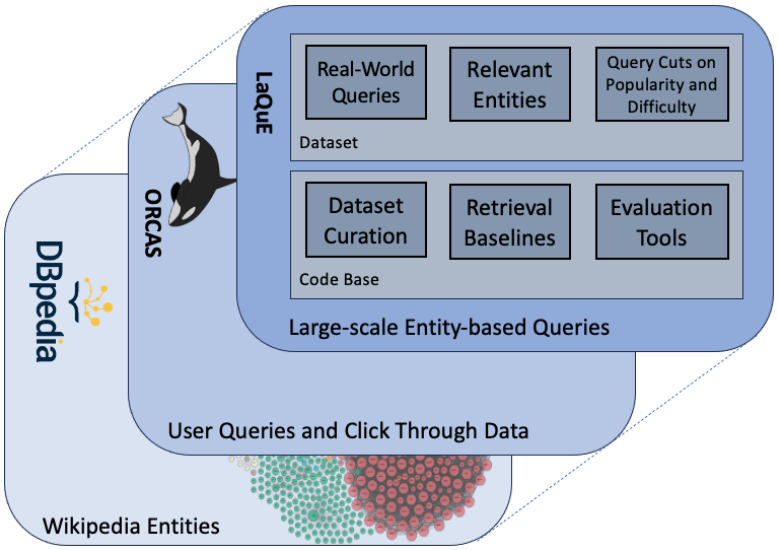

Fig. 1: Overview of the LaQuE Framework.

Licensing: In compliance with open data principles and to facilitate transparent and accessible research, we have made the LaQuE framework openly available on Zenodo. It has been assigned a Digital Object Identifier (DOI) for easy identification and citation, ensuring its long-term accessibility and proper attribution. We have chosen to license it under the Creative Commons license, which allows for broad use and redistribution, provided proper attribution is given. The dataset is released under anonymity given this stage of peer-review. As such, the authors' identities are not disclosed at this time. Researchers interested in accessing and utilizing LaQuE at this time can access it through the anonymized link: `https://anonymous.4open.science/r/LaQuE-0CDD/`.

2 The LaQuE Framework

In this section, we will discuss the intuitive idea behind the resources curated and made available through LaQuE as well as provide some of their statistical characteristics.

2.1 Dataset Curation in LaQuE

The data curated by LaQuE are derived from the ORCAS dataset, which consists of a vast collection of 18 million relations between 10 million distinct queries and relevant document URLs. There are at least two main advantages to our approach for using the ORCAS queries: First, the query set is diverse as it was

Table 1: The statistics of the dataset from LaQuE in terms of queries and their relevant entities in train, dev, and test sets.

Split	#Queries	Avg $\|q\|$	Std $\|q\|$	#Related Entities	Avg #Entity
Train	2,019,183	2.793	1.344	2,176,400	1.08
Dev	112,176	2.792	1.345	120,081	1.07
Test	112,176	2.796	1.349	119,200	1.06
Total	2,243,535	2.793	1.345	2,415,681	1.08

Table 2: Sample queries and their relevant entities.

Query	Related Entity
what is phylogeny	`<dbpedia:Phylogenetics >`
who was melchizedek parents	`<dbpedia:Melchizedek >`
nashville actors	`<dbpedia:List_of_Nashville_cast_members >`
first iphone released	`<dbpedia:IPhone_(1st_generation) >`
fisher river	`<dbpedia:Fisher_River_Cree_Nation >`

curated from millions of users, encompassing different topics. Second, the abundance of data in the ORCAS dataset allows for different cuts of the dataset for various purposes, and even after applying these cuts, there remain a significant number of data points in each cut for model training and evaluation purposes. The LaQuE framework offers the possibility to extract queries and their relevant clicked documents from ORCAS based on an intuitive idea: From the URIs of the clicked documents in ORCAS, LaQuE applies a filtering strategy to retain only those URLs connected to the English version of Wikipedia. This allows LaQuE to identify entities from Wikipedia that were able to satisfy the information need behind a specific user query. This idea is inspired by previous work [14, 37, 11, 50], where user clicks in search log files are considered implicit feedback for relevance.

Similar to prior works [27, 30, 23, 43], LaQuE leverages the English subset of DBpedia version 3.7[3] as its main collection of entities. LaQuE ensures that all selected entities must have a title and abstract, specifically the *rdfs:label* and *rdfs:comment* predicates, and have excluded any category, redirect, and disambiguation pages. This provides LaQuE with access to a set of 4.6 million entities, each uniquely identifiable through their URI. By intersecting the filtered ORCAS dataset that is linked with documents that have a valid Wikipedia URI with the filtered DBpedia dataset consisting of 4.6 million entities, LaQuE curates and offers a dataset, which includes user-generated queries that are related to Entities in DBpedia.

In summary, the ORCAS dataset serves as a foundational resource for LaQuE. This is a strong advantage for LaQuE as ORCAS boasts a rich collection of user-generated queries, sourced from millions of users across various topics and domains. This diversity ensures that LaQuE also encompasses a wide spectrum

[3] `http://downloads.dbpedia.org/wiki-archive/Downloads2015-10.html`

Table 3: Sample queries with more than one relevant entity.

Query	Relevant Entity
the temptations	<dbpedia:Paul_Williams_(The_Temptations)> <dbpedia:The_Temptations>
the texas rangers	<dbpedia:Texas_Rangers_(baseball)> <dbpedia:Texas_Ranger_Division>
the tracey ullman show	<dbpedia:The_Tracey_Ullman_Show> <dbpedia:The_Simpsons_shorts>
project management	<dbpedia:Project_management_triangle> <dbpedia:Project_management> <dbpedia:Project_manager> <dbpedia:Project_management_software>
progressivism definition	<dbpedia:Progressivism> <dbpedia:Progressive_education> <dbpedia:Progressivism_in_the_United_States>
prague	<dbpedia:Prague,_Oklahoma> <dbpedia:Czech_Republic> <dbpedia:Prague> <dbpedia:Prague_astronomical_clock>

of information needs, making it a valuable resource for training and evaluation purposes. We emphasize that LaQuE includes a careful selection of queries from ORCAS that explicitly exhibit a clear intent to retrieve Wikipedia entities. This intent-filtering approach is designed to ensure that the queries offered through LaQuE are relevant to the entity retrieval task. The process supported by LaQuE involves intersecting ORCAS queries with relevant entities in DBpedia. LaQuE ensures that relevant entities are available on DBpedia as this will allow entity retrieval methods to benefit from additional external sources of information such as content on DBpedia and knowledge graph embeddings. In summary, LaQuE is designed to benefit from the effective integration and intersection of ORCAS and DBpedia for entity retrieval. We believe that this approach strengthens the foundations of our work and ensures that LaQuE is a valuable framework for the research community.

2.2 LaQuE Statistics

The statistics of the data provided through the LaQuE framework in terms of the number of queries as well as their relevant entities are shown in Table 1. LaQuE offers over 2.2 million queries and their relevant entities. It randomly splits the queries into training, development (dev), and test sets, with a distribution of 90%, 5%, and 5%, respectively. As a result, the training set contains 2 million queries, while both the development and test sets consist of over 100,000 queries each. Furthermore, Table 1 presents the average number of query terms and the standard deviation of the number of query terms. These statistics show LaQuE ensures that the data is well-distributed among the three splits. On

Table 4: Sample entities with more than one related query. Individual queries are separated by a semicolon;

Relevant Entity	Submitted Queries
<dbpedia:Aaron>	Aaron; aaron and moses; aaron bible; aaron brother of moses; aaron budjen; aaron from the bible; aaron high priest; aaron in bible; aaron in the bible; aaron in the bible facts; aaron meaning; aaron moses; aaron moses brother; aaron of the bible; aaron old testament
<dbpedia:Belly_dance>	arab belly dance; arabian dance; bally dance
<dbpedia:Ballston,_New_York>	Ballston; ballston lake; ballston lake new york; ballston lake ny
<dbpedia:Lumbricus_terrestris>	Anecic; canadian nightcrawlers; classification of earthworm; common earthworm
<dbpedia:Common_krait>	blue krait; blue krait snake; bungarus caeruleus; common krait; common krait snake

average, each query is associated with 1.08 relevant entities, indicating that the majority of queries have only one relevant entity. Sparse labels, where queries have few relevant entities, are also observed in other well-known and widely-used benchmarks such as the MS MARCO dataset [42, 45, 3, 9, 39]. However, this does not undermine the reliability of the evaluation process as appropriate strategies can be employed to handle sparse labels [3, 9, 39]. In Table 2, we show a few sample queries from LaQuE accompanied with their relevant entities, generated from real users and adopted from the ORCAS dataset.

We have also conducted a detailed analysis of the distribution of entities on a per-query basis. While, on average, there are only 1.08 entities per query, LaQuE offers a substantial number of queries with multiple entities, owing to its large size. In Figure 2(a), we present a logarithmic representation of the number of entities per query. We note that to avoid noise, we filter a number of queries that have more than 20 entities. Figure 2(a) shows that there are over 145,000 queries with 2 entities, over 10,000 queries with 3 entities, over 1,600 queries with more than 4 entities and so on. This abundance of queries with multiple relevant entities in LaQuE enables us to achieve better and more diverse training for entity search purposes. In Table 3, we present a few examples of queries that have more than one entity. We additionally investigate the mapping between individual entities and variations of queries with which they are associated. To do so, we demonstrate the histogram of the number of unique queries per entity in Figure 2(b). LaQuE filters any noisy entities that have more than 100 different queries (such as 'www'). As shown in this Figure, there are an abundant number of entities with different queries mapped to them. This will allow such entities to be used in methods such as query transformation, query refinement, and query expansion. In Table 4, we provide a few examples of a single entity that have been considered relevant for different queries.

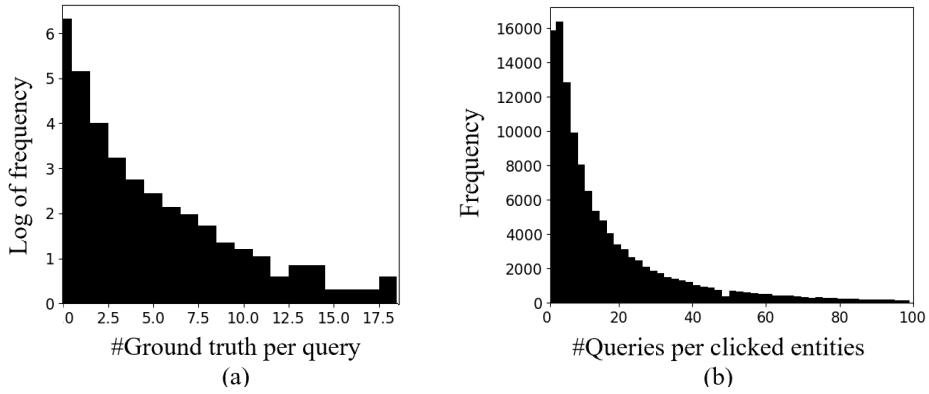

Fig. 2: (a) Distribution of the number of relevant entities per query. (b) Distribution of the number of unique queries per entity.

2.3 State of the Art Baselines in LaQuE

LaQuE offers a comprehensive set of retrievers, including both high-dimensional bag-of-word-based sparse retrievers and neural-based dense retrievers, for benchmarking purposes. In this paper, we only report the results of various retrievers based on the LaQuE dev set due to limited space. However, all complete results are available on our GitHub repository. For the sparse retrievers, LaQuE offers two methods, namely BM25 [49] and QL [54]. To enhance these sparse retrievers and investigate the impact of pseudo-relevance feedback and query expansion on the entity retrieval task, LaQuE incorporates the RM3 framework to create BM25-PRF and QL-PRF variants with pseudo-relevance feedback. For the dense retrievers, LaQuE provides a bi-encoder-based siamese network, as used in numerous previous studies, including [32, 46, 38, 58]. The model consists of two separate encoder towers, with one encoding the query and the other encoding the candidate content. LaQuE provides the means to evaluate the performance using various transformer models, such as BERT, DistilBERT, DistilRoBERTa, and MiniLM, all of which exhibit promising results across different downstream tasks, including passage retrieval, entity retrieval, and question answering [32, 46, 38, 58, 47, 55, 48]. During the training phase, LaQuE optimizes multiple negatives ranking loss function, encouraging higher similarity scores for relevant query-candidate pairs and lower scores for irrelevant pairs. In the training set, LaQuE considers the query and entity pairs as positive (relevant) data points. It adopts a training strategy from the Sentence Transformer library [46], where negative pairs are randomly sampled from the top-1000 retrieved entities using BM25, similar to [32, 58, 47]. For the purposes of the experiments reported in this paper, we customize LaQuE to consider only one negative sample per pair and train the models for one epoch. During each epoch, a batch size of 64 is employed, and the process is initiated with a warm-up phase spanning 1,000 steps. Upon completing the training of the bi-encoder model, LaQuE proceeds

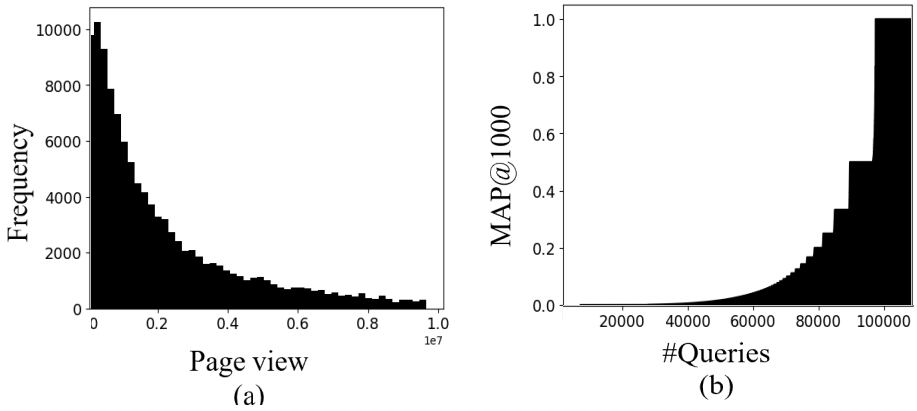

Fig. 3: Distribution of queries based on (a) number of page views (b) performance of BM25 in terms of MAP@1000.

Table 5: Performance of sparse and dense retrievers on the development set from LaQuE in terms of MAP@10, nDCG@10 and Recall@10 as well as MAP@1000, nDCG@1000 and Recall@1000.

			Cut-off@10			Cut-off@1000		
	Retriever	Training	MAP	nDCG	Recall	MAP	nDCG	Recall
Sparse Retrievers	BM25	-	0.2234	0.2662	0.3953	0.2338	0.3369	0.7965
	BM25 + RM3	-	0.2050	0.2624	0.3869	0.2157	0.3222	0.7965
	QL	-	0.2156	0.2581	0.3869	0.2260	0.3291	0.7891
	QL+RM3	-	0.1952	0.2414	0.3833	0.2058	0.3135	0.7918
Dense Retrievers	BERT	MS MARCO	0.3742	0.4218	0.5600	0.3820	0.4748	0.8607
		LaQuE	0.6018	0.6489	0.7801	0.6069	0.6781	0.6069
	DistilBERT	MS MARCO	0.4078	0.4553	0.5915	0.4155	0.5025	0.8505
		LaQuE	0.6179	0.6636	0.7900	0.6229	0.6920	0.9417
	DistilRoBERTa	MS MARCO	0.3335	0.3778	0.5068	0.3412	0.4272	0.7828
		LaQuE	0.5569	0.6056	0.7418	0.5629	0.6404	0.9289
	MiniLM	MS MARCO	0.4226	0.4664	0.5902	0.4294	0.5081	0.8184
		LaQuE	0.5731	0.6195	0.7481	0.5785	0.6501	0.9110

to build an index for the collection, which involves storing the embedding vectors of entities within the collection. To accomplish this, LaQuE leverages the capabilities of FAISS [31]. The choice of FAISS is motivated by its efficiency in conducting approximate nearest neighbor retrieval, a feature that significantly enhances the speed and effectiveness of retrieving relevant entities in response to queries during the inference phase. When a query is received, the trained model initially encodes it into a vector representation. This vector is then employed to locate relevant entities within the constructed index, utilizing the L^2 distance function as recommended in [33].

Table 5 demonstrates the performance of the various retrievers in terms of Mean Average Precision (MAP), normalized Discounted Cumulative Gain (nDCG), and Recall on the top-10 and top-1000 retrieved entities, specifically on

Table 6: Performance of established baselines on popularity-based query subsets in terms of MAP@1000

	Retriever	Train Set	Unpopular	Somewhat Popular	Popular	Highly Popular
Sparse Retrievers	BM25	-	0.3464	0.2545	0.1821	0.1217
	BM25 + RM3	-	0.3283	0.2447	0.1789	0.1229
	QL	-	0.5101	0.4292	0.3746	0.3193
	QL+RM3	-	0.5738	0.4927	0.4383	0.3781
Dense Retrievers	BERT	MS MARCO	0.5109	0.4146	0.3346	0.2377
		LaQuE	0.6102	0.6076	0.6030	0.5825
	DistilBERT	MS MARCO	0.5101	0.4293	0.3746	0.3193
		LaQuE	0.6104	0.6170	0.6228	0.6179
	DistilRoBERTa	MS MARCO	0.4231	0.3604	0.3130	0.2425
		LaQuE	0.5559	0.5612	0.5590	0.5488
	MiniLM	MS MARCO	0.5218	0.4381	0.3867	0.3397
		LaQuE	0.5840	0.5790	0.5704	0.5593

the development set offered by LaQuE. As shown in this table and aligned with the performance of sparse versus dense retrievers on other downstream tasks, dense retrievers outperform sparse retrievers by a large margin. We also note that pseudo-relevance feedback (RM3) would not help addressing the queries since it did not lead to any consistent significant improvement on the results. Among the dense retrievers, we conducted experiments using a pre-trained model on the MS MARCO passage collection dataset and compared it with the model trained on LaQuE. While both models perform better than the set of sparse retrievers, the dense retrievers fine-tuned on the language model using LaQuE outperform the MS MARCO model. For example, taking DistilBERT as an example, it achieves a MAP@10 of 0.4078 and a recall@10 of 0.8505, whereas the same model trained on LaQuE for one epoch obtained a MAP@10 of 0.6179 and a recall@10 of 0.9417. This observation confirms how training on a large-scale entity retrieval task can significantly boost the performance of entity retrievers.

3 Query Subsets in LaQuE

We delve deeper into entity retrieval task by examining queries based on their characteristics and the attributes of the related entities. LaQuE categorizes queries based on 1) popularity of the related entities; and 2) performance of the queries. This categorization approach encourages the research community to not only focus on the query set as a whole but also tackle more challenging and diverse queries.

3.1 Popularity-based Query Subsets

To determine the popularity of entities, LaQuE collects the total page views for each entity on Wikipedia from January 1, 2018, to December 31, 2022, spanning

Table 7: Performance of established baselines on difficulty-based query subsets in terms of MAP@1000

	Retriever	Train Set	Easy	Medium	Hard	Very Hard
Sparse Retrievers	BM25	-	0.8171	0.1585	0.0155	0.0004
	BM25 + RM3	-	0.7602	0.1716	0.0200	0.0007
	QL	-	0.7015	0.5264	0.3511	0.1597
	QL+RM3	-	0.7851	0.6035	0.4099	0.1921
Dense Retrievers	BERT	MS MARCO	0.6611	0.4711	0.3094	0.1579
		LaQuE	0.7599	0.7213	0.6251	0.3899
	DistilBERT	MS MARCO	0.7016	0.5265	0.3512	0.1597
		LaQuE	0.7632	0.7331	0.6471	0.4188
	DistilRoBERTa	MS MARCO	0.5604	0.4189	0.2861	0.1526
		LaQuE	0.6973	0.6553	0.5632	0.3769
	MiniLM	MS MARCO	0.7137	0.5472	0.3655	0.1639
		LaQuE	0.7415	0.6940	0.5959	0.3593

a period of five years. Analyzing the number of views received by these entities provides insights into whether popular entities are more likely to be retrieved by the retrieval systems. Figure 3(a) illustrates the histogram of page views for related entities in LaQuE. As depicted in the figure, the distribution of entity view counts follows a long-tailed pattern. Based on the range of page views, LaQuE divides the queries into four equally sized buckets, creating four query subsets: "Unpopular", "Somewhat Popular", "Popular", and "Highly Popular". In Table 7, we present the performance results of retrievers on these query subsets.As observed in the table, we consistently notice that the less popular a related entity is to a query, the higher the performance achieved by both sparse and pre-trained dense retrievers. However, this observation does not hold on models that were trained on LaQuE. We hypothesize that this trend occurs because, in cases where the information need of a user is less well-known, users tend to provide more detailed and elaborate queries. For instance, queries related to unpopular queries include examples such as 'apostrophe figure of speech' and 'which president moved thanksgiving up a week'. Such elaborate queries result in higher retrieval performance. Conversely, for popular queries, users often enter shorter queries, examples of which include 'alphabet' and 'lightning 2'. For these short queries, industry-scale search engines can find relevant entities by utilizing various user personalization, and trending information. However, without access to such information, retrieval methods will find it challenging to retrieve the relevant information for those queries.

3.2 Difficulty-based Query Subsets

LaQuE also provides the means to evaluate query subsets based on their difficulty. Following the approach of previous studies [19, 2, 10], LaQuE classifies queries based on their level of difficulty. By evaluating query performance across

different difficulty levels, LaQuE offers insights into the strengths and limitations of existing entity retrieval systems and identifies areas for improvement [53].

We report the performance of the widely used BM25 model on the queries from the LaQuE development set, using MAP@1000 as shown in Figure 3(b). The figure reveals a long-tail distribution of retriever performance. This means that while the retrievers are effective for a subset of queries, their performance is poor for others, resulting in an imbalanced distribution of performance across all queries. For instance, more than 20,000 queries in the LaQuE development set have a MAP@1000 value of zero when being retrieved with BM25. This underscores the need for the research community to focus on addressing more challenging queries. Building upon this observation, LaQuE categorizes queries into four equal-sized buckets based on their performance in terms of MAP@1000 with the BM25 model. These categories are labeled as "Very Hard", "Hard", "Medium", and "Easy". The results for these query subsets are reported in the right side of Table 7. It is shown that query difficulty remains consistent across all the retrievers. In other words, the "very hard" subset of queries, which exhibits the lowest performance by BM25, also demonstrates the lowest performance even with dense retrievers. The poor performance on this subset compared to the other query subsets emphasizes the importance for the research community to focus on addressing more challenging queries.

4 Concluding Remarks

We have introduced the LaQuE framework, which offers an extensible code base as well as real-user queries and large-scale training data for the entity search task. LaQuE offers access to more than 2.2 million query-entity pairs divided into train, development, and test sets, facilitating the training and evaluation of neural models for entity search. Additionally, LaQuE categorizes query sets based on popularity and difficulty, encouraging researchers to tackle challenging queries and explore biases associated with popular entities. We believe that LaQuE has the potential to extend its utility beyond the entity retrieval task, as its large-scale nature can be adapted for entity linking, query refinement, query generation, and other downstream tasks in information retrieval and natural language processing. Lastly, we note that as it is originally mentioned by the ORCAS team, this dataset may exhibit biases related to race, gender, and other factors. These biases can stem from inherent biases in the original queries, user clicks, and search algorithms. While studying these biases can be valuable, it is crucial for researchers to be aware of these potential biases when using the data, as they can impact the learning of models and subsequent analyses. We encourage the research community to adopt the LaQuE framework with a critical but constructive perspective. We recognize that bias is a complex issue, and our dataset represents an opportunity to engage in meaningful discussions and research on this topic.

References

1. Alexander, D., Kusa, W., P. de Vries, A.: Orcas-i: queries annotated with intent using weak supervision. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 3057–3066 (2022)
2. Arabzadeh, N., Mitra, B., Bagheri, E.: Ms marco chameleons: challenging the ms marco leaderboard with extremely obstinate queries. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 4426–4435 (2021)
3. Arabzadeh, N., Vtyurina, A., Yan, X., Clarke, C.L.: Shallow pooling for sparse labels. Information Retrieval Journal **25**(4), 365–385 (2022)
4. Bagheri, E., Ensan, F., Al-Obeidat, F.: Neural word and entity embeddings for ad hoc retrieval. Information Processing & Management **54**(4), 657–673 (2018)
5. Balog, K.: Entity retrieval. (2018)
6. Balog, K., Neumayer, R.: Hierarchical target type identification for entity-oriented queries. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 2391–2394 (2012)
7. Balog, K., Neumayer, R.: A test collection for entity search in dbpedia. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. pp. 737–740 (2013)
8. Balog, K., Serdyukov, P., Vries, A.P.d.: Overview of the trec 2010 entity track. Tech. rep., NORWEGIAN UNIV OF SCIENCE AND TECHNOLOGY TRONDHEIM (2010)
9. Büttcher, S., Clarke, C.L., Yeung, P.C., Soboroff, I.: Reliable information retrieval evaluation with incomplete and biased judgements. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 63–70 (2007)
10. Carmel, D., Yom-Tov, E., Darlow, A., Pelleg, D.: What makes a query difficult? In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 390–397 (2006)
11. Carterette, B., Jones, R.: Evaluating search engines by modeling the relationship between relevance and clicks. Advances in neural information processing systems **20** (2007)
12. Chatterjee, S., Dietz, L.: Entity retrieval using fine-grained entity aspects. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1662–1666 (2021)
13. Chen, T., Zhang, M., Lu, J., Bendersky, M., Najork, M.: Out-of-domain semantics to the rescue! zero-shot hybrid retrieval models. In: Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I. pp. 95–110. Springer (2022)
14. Chuklin, A., Serdyukov, P., De Rijke, M.: Click model-based information retrieval metrics. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. pp. 493–502 (2013)
15. Cuzzola, J., Jovanović, J., Bagheri, E.: Rysannmd: a biomedical semantic annotator balancing speed and accuracy. Journal of Biomedical Informatics **71**, 91–109 (2017)
16. De Cao, N., Izacard, G., Riedel, S., Petroni, F.: Autoregressive entity retrieval. arXiv preprint arXiv:2010.00904 (2020)
17. Dietz, L., Foley, J.: Trec car y3: Complex answer retrieval overview. In: Proceedings of Text REtrieval Conference (TREC) (2019)

18. Dietz, L., Verma, M., Radlinski, F., Craswell, N.: Trec complex answer retrieval overview. In: TREC (2017)
19. Ensan, F., Bagheri, E.: Document retrieval model through semantic linking. In: Proceedings of the tenth ACM international conference on web search and data mining. pp. 181–190 (2017)
20. Feng, Y., Zarrinkalam, F., Bagheri, E., Fani, H., Al-Obeidat, F.: Entity linking of tweets based on dominant entity candidates. Social Network Analysis and Mining **8**, 1–16 (2018)
21. Fetahu, B., Fang, A., Rokhlenko, O., Malmasi, S.: Gazetteer enhanced named entity recognition for code-mixed web queries. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1677–1681 (2021)
22. Fetahu, B., Gadiraju, U., Dietze, S.: Improving entity retrieval on structured data. In: The Semantic Web-ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I 14. pp. 474–491. Springer (2015)
23. Gerritse, E.J., Hasibi, F., de Vries, A.P.: Graph-embedding empowered entity retrieval. In: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42. pp. 97–110. Springer (2020)
24. Gillick, D., Kulkarni, S., Lansing, L., Presta, A., Baldridge, J., Ie, E., Garcia-Olano, D.: Learning dense representations for entity retrieval. arXiv preprint arXiv:1909.10506 (2019)
25. Hasibi, F., Balog, K., Bratsberg, S.E.: Exploiting entity linking in queries for entity retrieval. In: Proceedings of the 2016 acm international conference on the theory of information retrieval. pp. 209–218 (2016)
26. Hasibi, F., Balog, K., Garigliotti, D., Zhang, S.: Nordlys: A toolkit for entity-oriented and semantic search. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1289–1292 (2017)
27. Hasibi, F., Nikolaev, F., Xiong, C., Balog, K., Bratsberg, S.E., Kotov, A., Callan, J.: Dbpedia-entity v2: a test collection for entity search. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1265–1268 (2017)
28. Hosseini, H., Mansouri, M., Bagheri, E.: A systemic functional linguistics approach to implicit entity recognition in tweets. Information Processing & Management **59**(4), 102957 (2022)
29. Hosseini, H., Nguyen, T.T., Wu, J., Bagheri, E.: Implicit entity linking in tweets: An ad-hoc retrieval approach. Applied Ontology **14**(4), 451–477 (2019)
30. Jafarzadeh, P., Amirmahani, Z., Ensan, F.: Learning to rank knowledge subgraph nodes for entity retrieval. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2519–2523 (2022)
31. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Transactions on Big Data **7**(3), 535–547 (2019)
32. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906 (2020)
33. Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., Lewis, M.: Generalization through memorization: Nearest neighbor language models. arXiv preprint arXiv:1911.00172 (2019)

34. Lin, J., Nogueira, R.F., Yates, A.: Pretrained transformers for text ranking: BERT and beyond. CoRR **abs/2010.06467** (2020), https://arxiv.org/abs/2010.06467
35. Lin, X., Lam, W., Lai, K.P.: Entity retrieval in the knowledge graph with hierarchical entity type and content. In: Proceedings of the 2018 acm sigir international conference on theory of information retrieval. pp. 211–214 (2018)
36. Macdonald, C., Ounis, I.: Voting for candidates: adapting data fusion techniques for an expert search task. In: Proceedings of the 15th ACM international conference on Information and knowledge management. pp. 387–396 (2006)
37. Macdonald, C., Ounis, I.: Usefulness of quality click-through data for training. In: Proceedings of the 2009 workshop on web search click data. pp. 75–79 (2009)
38. Macdonald, C., Tonellotto, N.: On approximate nearest neighbour selection for multi-stage dense retrieval. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 3318–3322 (2021)
39. Magdy, W., Jones, G.J.: Examining the robustness of evaluation metrics for patent retrieval with incomplete relevance judgements. In: Multilingual and Multimodal Information Access Evaluation: International Conference of the Cross-Language Evaluation Forum, CLEF 2010, Padua, Italy, September 20-23, 2010. Proceedings 1. pp. 82–93. Springer (2010)
40. Malmasi, S., Fang, A., Fetahu, B., Kar, S., Rokhlenko, O.: Multiconer: a large-scale multilingual dataset for complex named entity recognition. arXiv preprint arXiv:2208.14536 (2022)
41. Meng, T., Fang, A., Rokhlenko, O., Malmasi, S.: Gemnet: Effective gated gazetteer representations for recognizing complex entities in low-context input. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1499–1512 (2021)
42. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: Ms marco: A human generated machine reading comprehension dataset. choice **2640**, 660 (2016)
43. Nikolaev, F., Kotov, A.: Joint word and entity embeddings for entity retrieval from a knowledge graph. In: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42. pp. 141–155. Springer (2020)
44. Pound, J., Mika, P., Zaragoza, H.: Ad-hoc object retrieval in the web of data. In: Proceedings of the 19th international conference on World wide web. pp. 771–780 (2010)
45. Qu, C., Yang, L., Chen, C., Qiu, M., Croft, W.B., Iyyer, M.: Open-Retrieval Conversational Question Answering. In: SIGIR (2020)
46. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
47. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2020), https://arxiv.org/abs/2004.09813
48. Reimers, N., Gurevych, I.: The curse of dense low-dimensional information retrieval for large index sizes. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). pp. 605–611. Association for Computational Linguistics, Online (8 2021), https://arxiv.org/abs/2012.14210
49. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M., et al.: Okapi at trec-3. Nist Special Publication Sp **109**, 109 (1995)

50. Scholer, F., Shokouhi, M., Billerbeck, B., Turpin, A.: Using clicks as implicit judgments: Expectations versus observations. In: Advances in Information Retrieval: 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings 30. pp. 28–39. Springer (2008)
51. Sciavolino, C., Zhong, Z., Lee, J., Chen, D.: Simple entity-centric questions challenge dense retrievers. arXiv preprint arXiv:2109.08535 (2021)
52. Shehata, D., Arabzadeh, N., Clarke, C.L.A.: Early stage sparse retrieval with entity linking (2022). https://doi.org/10.48550/ARXIV.2208.04887, https://arxiv.org/abs/2208.04887
53. Shehata, D., Arabzadeh, N., Clarke, C.L.: Early stage sparse retrieval with entity linking. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 4464–4469 (2022)
54. Song, F., Croft, W.B.: A general language model for information retrieval. In: Proceedings of the eighth international conference on Information and knowledge management. pp. 316–321 (1999)
55. Thakur, N., Reimers, N., Daxenberger, J., Gurevych, I.: Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 296–310. Association for Computational Linguistics, Online (6 2021), https://arxiv.org/abs/2010.08240
56. Van Gysel, C., de Rijke, M., Kanoulas, E.: Semantic entity retrieval toolkit. arXiv preprint arXiv:1706.03757 (2017)
57. Wu, L., Petroni, F., Josifoski, M., Riedel, S., Zettlemoyer, L.: Scalable zero-shot entity linking with dense entity retrieval. arXiv preprint arXiv:1911.03814 (2019)
58. Zhan, J., Mao, J., Liu, Y., Zhang, M., Ma, S.: Repbert: Contextualized text embeddings for first-stage retrieval. arXiv preprint arXiv:2006.15498 (2020)