Understanding the User: An Intent-Based Ranking Dataset

Abhijit Anand
L3S Research Center
Hannover, Germany
aanand@L3S.de

Jurek Leonhardt
Delft University of Technology
Delft, The Netherlands
L.J.Leonhardt@tudelft.nl

Venktesh V
Delft University of Technology
Delft, The Netherlands
v.Viswanathan-1@tudelft.nl

Avishek Anand
Delft University of Technology
Delft, The Netherlands
avishek.anand@tudelft.nl

ABSTRACT

As information retrieval systems continue to evolve, accurate evaluation and benchmarking of these systems become pivotal. Web search datasets, such as MS MARCO, primarily provide short keyword queries without accompanying intent or descriptions, posing a challenge in comprehending the underlying information need. This paper proposes an approach to augmenting such datasets to annotate informative query descriptions, with a focus on two prominent benchmark datasets: TREC-DL-21 and TREC-DL-22. Our methodology involves utilizing state-of-the-art LLMs to analyze and comprehend the implicit intent within individual queries from benchmark datasets. By extracting key semantic elements, we construct detailed and contextually rich descriptions for these queries. To validate the generated query descriptions, we employ crowdsourcing as a reliable means of obtaining diverse human perspectives on the accuracy and informativeness of the descriptions. This information can be used as an evaluation set for tasks such as ranking, query rewriting, or others.

CCS CONCEPTS

• **Information systems → Information retrieval**.

KEYWORDS

Intent Dataset; Ad-hoc retrieval; Ranking; User Intents; Web Search; Diversity; Data collection

1 INTRODUCTION

In information retrieval (IR), a core challenge in building ranking models is to explicitly or implicitly *aligning* the actual user intent with the machine intent, i.e., the intent as understood by the ranker. This misalignment stems from the inherent complexity and variability in how users articulate their information needs versus how these needs are interpreted and processed by retrieval systems. This misalignment might be due to multiple reasons – ambiguity, poorly formulated queries, complex queries, or a retrieval set that lacks relevant documents [5, 13].

Most current research on ranking models in IR is based on training parameterized models over large training datasets from MS MARCO [15]. However, to the best of our knowledge, there exist no recent datasets that attempt to measure the chasm between user intent and machine intent. The current practice of measuring ranking performance is through sparsely [15] or densely annotated ad-hoc ranking test sets [7–11] that provide queries and corresponding

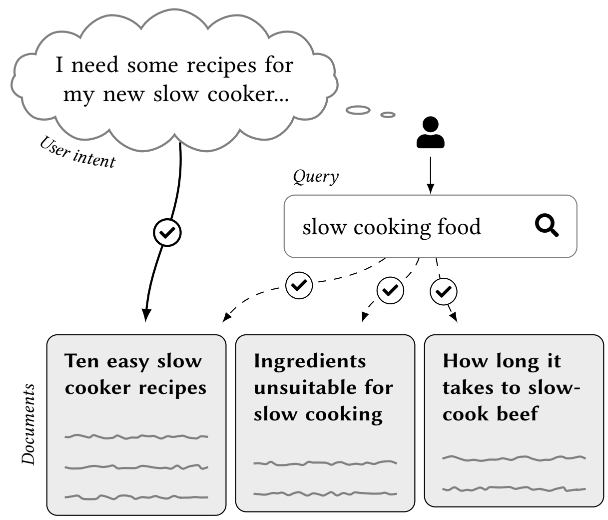


Figure 1: An illustration of a user querying a search engine. The user has a specific intent in mind, but formulates the query in a more ambiguous way. As a result, there is a discrepancy between the documents relevant to the query and the documents relevant to the actual user intent.

relevance annotations. While these test sets allow for determining the overall effectiveness of a ranker, they fail to provide a way of measuring the extent to which the ranking models understand the true intent of the user. For example, consider the query "*what are the three countries in 1984*". While the intent—to identify the three countries mentioned in George Orwell's novel "1984"—seems clear, it remains difficult to rank effectively because it requires specific contextual knowledge that may not be directly available in the retrieved documents. Another example is the query "*slow cooking food*" (cf. Fig. 1). Although this query appears to be straightforward, it can have multiple intents. This multiplicity of potential intents complicates the ranking process, as the system needs to correctly infer and prioritize the user's actual intent to provide relevant results. Knowing the user's intent allows the model to retrieve and rank documents most relevant to that intent, thereby addressing a critical challenge in handling ambiguous queries.

In this paper, we specifically focus on a subset of these challenges: *queries that contain multiple intents*. We propose a new dataset named DL-MIA (**M**S MARCO **I**ntent **A**nnotations), which is a derivative of the TREC-DL test sets. The DL-MIA dataset contains 2655 tuples of (query, intent, passage, label) over a small yet challenging set of 24 queries from the TREC-DL '21 and '22 datasets. To construct DL-MIA, the key challenge was to accurately formulate user intents, as only queries are available in the TREC-DL test sets. Toward this, we used a combination of LLM-generated query-specific intents and sub-intents that are post-processed through a carefully designed crowd-sourcing process to ensure human supervision and quality control. DL-MIA mainly aims at measuring the gap between user intent and query by *fine-grained intent annotation*, but can be used in multiple ranking scenarios, such as re-ranking, diversification, intent coverage, or query suggestion tasks.

Our contributions are twofold – first, we introduce a comprehensive dataset DL-MIA that meticulously documents the variations and complexities of user intent; second, we provide an analysis of this dataset's impact on ranking performance by applying it to several baseline models. DL-MIA is publicly available at https://zenodo.org/doi/10.5281/zenodo.11471482.

2 RELATED WORK

Several ranking datasets have been published that consider the concept of what we refer to as *user intents*. Most notably, the data provided for the TREC-Web track [5] customarily includes topics (queries) along with *topic descriptions* as well as, in many cases, *subtopics*. These subtopics represent various distinct aspects that each topic may have. The data further includes relevance judgments for documents from the ClueWeb collections w.r.t. the topics and subtopics. However, the TREC-Web track has been discontinued after 2014, and ClueWeb corpora are not freely available. Our dataset is similar, as the subtopics are essentially user intents.

The MS MARCO ranking dataset [15], which has emerged as one of the most widely used collections for IR-related tasks in recent years, contains a large number of training and evaluation queries. Furthermore, the TREC-DL track [7, 10] provides annotated test sets of queries and corresponding relevance annotations. More recently, the second version of the MS MARCO corpus, which is significantly larger than the first version, was released to be used in the TREC-DL 2021 track and onward [8, 9, 11].

Mackie et al. [13] showed that queries (topics) within TREC-DL vary with respect to their complexity (and, hence, difficulty) and released the DL-HARD dataset. Along with relevance annotations, this dataset assigns *intent categories* to each query. Similarly, *intent taxonomies* have been proposed for web search in general [3] as well as legal case retrieval [20]. The difference compared to our work is that we annotate specific user intents rather than categories.

Another related line of work deals with the *reformulation* of complex queries. Mackie et al. [14] recently released the CODEC collection for document and entity ranking, which also contains query reformulations. Salamat et al. [18] showed that the way queries are worded has an impact on their corresponding ranking performance. Our proposed user intents can be seen as reformulations that focus on specific aspects of the original query.

3 THE DL-MIA DATASET

In this section, we introduce the DL-MIA dataset by outlining the creation and annotation process and presenting some statistics.

3.1 Dataset Creation

The process of creating the dataset comprises several key stages: generating candidate intents using an LLM (Section 3.1.1), clustering and manual refinement of intents (Section 3.1.2), crowdsourcing annotations (Section 3.1.3), merging similar intents (Section 3.1.4) and QRel creation (Section 3.1.5). This process is illustrated in Fig. 2.

3.1.1 Generating Candidate User Intents. For all queries in the TREC-DL-21 and '22 test sets, we retrieve all relevant passages using their respective QRel files. We then cluster similar passages per query. To achieve this, we first obtain passage embeddings using Sentence-BERT [16] and then group passages into the same cluster if their pairwise cosine similarity exceeds a threshold of 0.8. In the next step, we select the query and passages from the clusters to give to the LLM to generate five distinct intents relevant to the query-passage pairs. We employ the GPT-4 model with the prompt given below. We use a temperature value of 0.6 to control randomness which helps in getting diverse intents.

LLM Prompt: Intent candidate generation

A person wants to find out distinct intention behind the question **{query}**. Give five descriptive (max. 15 words) distinct intentions which are easy to understand. Consider all documents in your response. Response should be in this format:
Intention:: <intention> , **Doc_list**::<list of documents with the intention>
Documents: **{list of input documents}**

3.1.2 Clustering and Intent Selection. After generating intents, we cluster similar intents using the SBERT embedding and cosine similarity approach as described above. We group intents that are similar in meaning if their pairwise cosine similarity exceeds a threshold of 0.9. This clustering process helps in reducing redundancy and coming up with distinct intents. After clustering, we do manual selection, where we examine the clustered intents and choose the most relevant ones for each cluster. We do this to remove irrelevant intents or hallucinated text by the LLM. If any intents are found to be incomplete or poorly written, they are manually rewritten to improve their clarity and comprehensiveness. This ensures that the intents are well-defined and useful for the next stages of the dataset creation process. After this process, only queries with 2 or more intents were selected which resulted in 26 queries.

3.1.3 Crowdsourcing Annotation. The next step involves crowdsourcing to annotate the intents with the relevant passages. Our pool of annotators comprises volunteers who are computer scientists and graduate school students familiar with ranking tasks for search. Annotators are presented with a query and a passage and are asked to determine which of the provided intents the passage satisfies. Additionally, annotators are given the option to add or modify intents if they find that the existing ones do not capture the

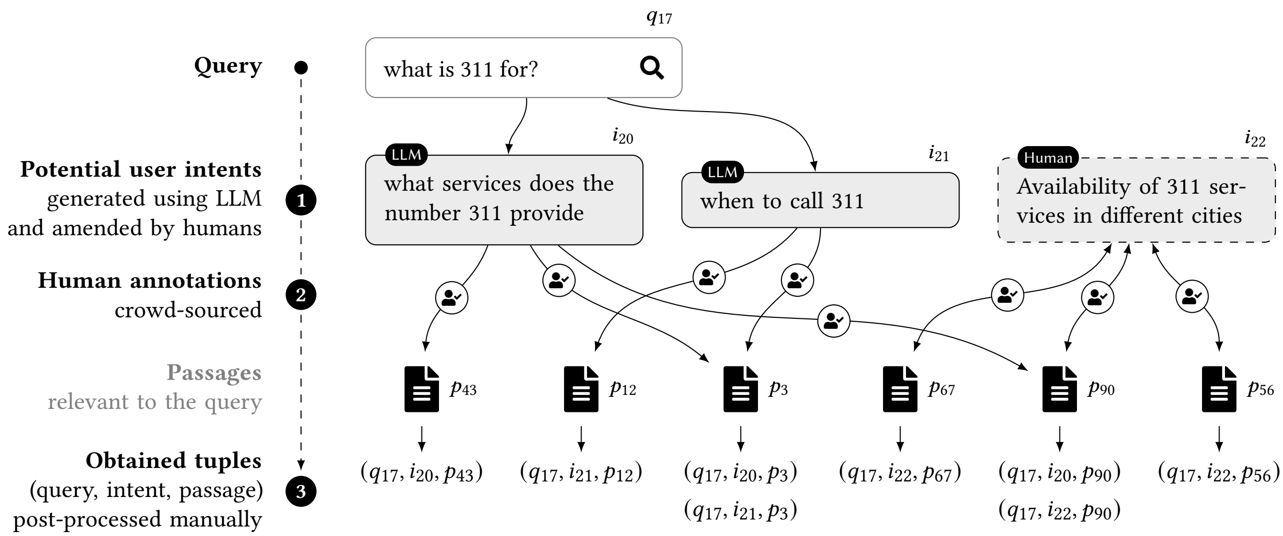
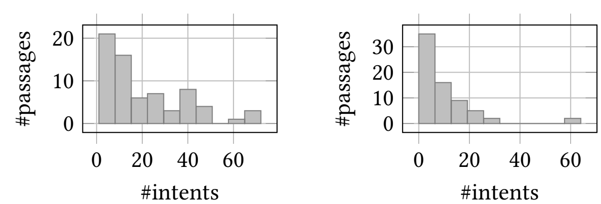


Figure 2: A high-level overview of how DL-MIA is created: Given a query, an LLM is used to generate candidate user intents. The query and its relevant passages (according to the original QRels), along with the candidate intents, are presented to human annotators, who can add, modify, or remove candidate intents and assign passages to them.

passage's intent. To manage queries with a large number of relevant passages (more than 30), the passages are divided into smaller chunks of 30. This division creates subqueries, making the annotation process more manageable for the annotators. Each subquery is annotated separately, ensuring that the workload is distributed and the annotators can focus on a smaller set of passages. In total, 22 sets of annotations are done by 16 distinct annotators and each set consist of 5 rounds (queries or subqueries), such that each query is annotated at least twice.

3.1.4 Manual Review and Merging of Intents. In order to improve data quality and avoid redundancy, we conduct a manual review and merge intents. We evaluate the intents suggested by the annotators and integrate them into the existing set of intents where appropriate. E.g., in Fig. 2 we merge "when to call 311" and "when to call 311 rather than 911" into a single intent. Any passage-intent pair which does not have at least two annotators is dropped to ensure that the final set of intents reflects a consensus among multiple annotators. The merging process also helps in consolidating similar intents and removing any redundant or less relevant ones. After this process, we end up with 24 queries. We further elaborate on different scenarios we encountered during this phase in Appendix D

3.1.5 Scoring and Creating QRel File. Finally, we score the intent-passage pairs and create a QRel file for ranking. The scoring is based on the annotations provided by the participants. Each intent-passage pair is scored as follows: a score of 0 is assigned if no annotator marked the intent, a score of 1 is assigned if at least one annotator marked the intent, and a score of 2 is assigned if all annotators marked the intent. These scores reflect the level of agreement among the annotators and the relevance of the intent to the passage. The final query-intent-passage-score mappings are compiled into a QRel file, which is used for ranking. This QRel file serves as ground-truth for evaluating information retrieval systems,



(a) rel ≥ 1 **(b) rel** $= 2$

Figure 3: Histograms illustrating the number of relevant passages per intent for (a) all relevant passages and (b) only passages with relevance label 2**.**

ensuring that the dataset can be effectively used for further research and application.

3.2 Statistics

Initially, the dataset included 118 queries from TREC-DL-21 and '22. Through a process of clustering and intent selection, 26 queries were identified as suitable for annotations, as these queries had two or more distinct intents (69 in total). After annotation (Sec. 3.1.3), a manual review and merging of intents were performed (Sec. 3.1.4). This process was necessary because the number of intents increased from 69 to 171 due to annotators adding custom intents. Hence, this review process was crucial in refining the dataset and ensuring the accuracy and clarity of the intents. After this rigorous review, 24 queries and 69 intents were finalized for inclusion in the dataset with **2655 relevance annotations** present in the final QRel file. The distribution of relevant passages per intent is shown in Fig. 3.

Because annotators were able to add custom intents, computing established agreement measures is difficult as the intents annotated by humans may have different granularities; however, the relevance

	Intent Ranking	Diversity
	nDCG@10	α-nDCG@10
Original queries, user intent QRels		
BM25	0.073	0.144
BERT	0.060	0.114
User intents as queries, user intent QRels		
BM25	0.116	0.250
BERT	0.169	0.375
ColBERTv2	**0.261**	**0.532**

Table 1: DL-MIA ranking performance. Best performing models are in bold. Re-rankers use the corresponding BM25 runs. Diversity is calculated at query level in both cases.

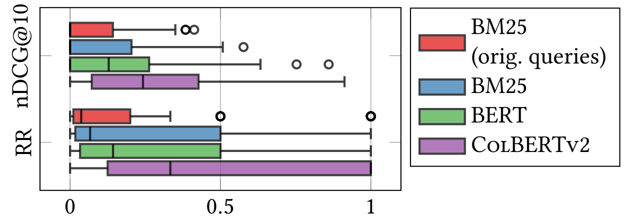


Figure 4: Performance comparison on a per-intent level. The boxplots show the distribution of the ranking performance of individual intents.

scores we obtained in Sec. 3.1.5 are determined by the overlap of judgments and can therefore be seen as an indication of agreement among annotators.

3.3 Tasks and Evaluation

The DL-MIA dataset can be used for several tasks, such as:

Intent-based ranking aims at improving the document ranking by understanding different user intents and ensuring that the returned documents are relevant to the intent. This can be evaluated using metrics like nDCG@10.

Diversity of search results aims at ensuring that document rankings provide diverse sets of responses that cover various aspects of the query to satisfy users information needs, evaluated using metrics like α-nDCG@10.

Intent-based summarization aims at generating a summary that covers multiple intents of a query, evaluated using metrics such as ROUGE or BLEU.

User and machine intent alignment aims at bridging the gap between user and machine intent through query rewriting to fully specify the intent [2]. DL-MIA aids in training generative models that can generate intents more aligned with real-world user intents.

4 EXPERIMENTS

In order to demonstrate the utility of DL-MIA, we conduct experiments using a number of simple baselines: **BM25** [17] is a lexical model which is also used as a first-stage retriever for re-rankers. **BERT** [12] is a cross-attention re-ranker (BERT-base, 12 layers). The input length is restricted to a maximum of 512 tokens. The model is trained on MS MARCO passage data using a pointwise ranking loss objective with a learning rate of 1e-5. **ColBERTv2** [19] is a multi-vector late-interaction re-ranking model that computes token-wise representations for the query and document and estimates relevance using the MaxSim operation.

4.1 Results

We report results on two of the tasks outlined in Section 3.3, namely *intent-based ranking* and *diversity of search results*. We present these results in Table 1. Note that we evaluate two settings: First, we use the original queries, but evaluate using the user intent-based QRels (i.e., assuming that the user had one specific intent in mind). Second, we treat the user intents as queries directly. The results show that, unsurprisingly, specifying the actual user intent as the query results in better performance than using the (more general) original queries. We additionally demonstrate the diversity ranking performance of various models using the α-nDCG@10 metric. To achieve this in the second setting (where user intents are treated as queries), we employ reciprocal rank fusion [6] with $k = 60$. This technique is applied to the intent-based rankings to generate a unified ranking for the original query. Overall, ColBERTv2 shows the best performance. Finally, we closely examine the ranking performance corresponding to each user intent in Fig. 4. The results are in line with Table 1.

The key takeaway from these results is the necessity of specifying concrete user intents; in other words: if a user has a specific information need, it is necessary to provide that intent as a clear, unambiguous query to a search engine.

5 CONCLUSION

In this paper, we have created the DL-MIA dataset to understand user intents, thereby satisfying information needs more effectively. We have used queries from TREC-DL-21 and TREC-DL-22, generated intents using an LLM, and crowd-sourced relevance annotations. DL-MIA can be used for a variety of tasks; we present performance of different models on ranking and diversity tasks, showing the importance of this dataset for fulfilling user information needs. For future work, we plan to extend DL-MIA to include queries from TREC-DL-19, TREC-DL-20, and DL-HARD.

REFERENCES

[1] Abhijit Anand, Vinay Setty, Avishek Anand, et al. 2023. Context aware query rewriting for text rankers using llm. *arXiv preprint arXiv:2308.16753* (2023).

[2] Avishek Anand, Venktesh V, Abhijit Anand, and Vinay Setty. 2023. Query Understanding in the Age of Large Language Models. arXiv:2306.16004 [cs.IR]

[3] B. Barla Cambazoglu, Leila Tavakoli, Falk Scholer, Mark Sanderson, and Bruce Croft. 2021. An Intent Taxonomy for Questions Asked in Web Search. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval* (Canberra ACT, Australia) *(CHIIR '21)*. Association for Computing Machinery, New York, NY, USA, 85–94. https://doi.org/10.1145/3406522.3446027

[4] Daniel L. Chen, Martin Schonger, and Chris Wickens. 2016. oTree—An open-source platform for laboratory, online, and field experiments. *Journal of Behavioral and Experimental Finance* 9 (2016), 88–97. https://doi.org/10.1016/j.jbef.2015.12.001

[5] Kevyn Collins-Thompson, Craig Macdonald, Paul N Bennett, Fernando Diaz, and Ellen M Voorhees. 2014. TREC 2014 Web Track Overview.. In *TREC*, Vol. 13. 1–15.

[6] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 758–759.

[7] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. arXiv:2102.07662 [cs.IR]

[8] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2022. Overview of the TREC 2021 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC. https://www.microsoft.com/en-us/research/publication/overview-of-the-trec-2021-deep-learning-track/

[9] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2023. Overview of the TREC 2022 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC. https://www.microsoft.com/en-us/research/publication/overview-of-the-trec-2022-deep-learning-track/

[10] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. arXiv:2003.07820 [cs.IR]

[11] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Hossein A. Rahmani, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2024. Overview of the TREC 2023 Deep Learning Track. In *Text REtrieval Conference (TREC)*. NIST, TREC. https://www.microsoft.com/en-us/research/publication/overview-of-the-trec-2023-deep-learning-track/

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). http://arxiv.org/abs/1810.04805

[13] Iain Mackie, Jeffrey Dalton, and Andrew Yates. 2021. How Deep is your Learning: the DL-HARD Annotated Deep Learning Dataset. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (<conf-loc>, <city>Virtual Event</city>, <country>Canada</country>, </conf-loc>) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2335–2341. https://doi.org/10.1145/3404835.3463262

[14] Iain Mackie, Paul Owoicho, Carlos Gemmell, Sophie Fischer, Sean MacAvaney, and Jeffrey Dalton. 2022. CODEC: Complex Document and Entity Collection. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (<conf-loc>, <city>Madrid</city>, <country>Spain</country>, </conf-loc>) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 3067–3077. https://doi.org/10.1145/3477495.3531712

[15] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. (November 2016). https://www.microsoft.com/en-us/research/publication/ms-marco-human-generated-machine-reading-comprehension-dataset/

[16] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[17] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (apr 2009), 333–389. https://doi.org/10.1561/1500000019

[18] Sara Salamat, Negar Arabzadeh, Shirin Seyedsalehi, Amin Bigdeli, Morteza Zihayat, and Ebrahim Bagheri. 2023. Neural Disentanglement of Query Difficulty and Semantics. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (<conf-loc>, <city>Birmingham</city>, <country>United Kingdom</country>, </conf-loc>) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 4264–4268. https://doi.org/10.1145/3583780.3615189

[19] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 3715–3734. https://doi.org/10.18653/v1/2022.naacl-main.272

[20] Yunqiu Shao, Haitao Li, Yueyue Wu, Yiqun Liu, Qingyao Ai, Jiaxin Mao, Yixiao Ma, and Shaoping Ma. 2023. An Intent Taxonomy of Legal Case Retrieval. *ACM Trans. Inf. Syst.* 42, 2, Article 62 (dec 2023), 27 pages. https://doi.org/10.1145/3626093

A GENERATING INTENTS USING LLM

The objective of this step is to generate a diverse set of user intents that accurately reflect the informational needs expressed by the user query. To achieve this, we first retrieve all relevant passages for each query from the QRel file. We observed a significant amount of redundancy among these passages, which could lead to duplication in intent generation when using a large language model (LLM). To mitigate this, we cluster similar relevant passages for each query before proceeding with intent generation. Several methods were explored, including entailment-based approaches , but we found that clustering using cosine similarity with Sentence-BERT [16] (detailed in Appendix B) yielded the best results. For the query "What is 311 for" in Table 2, there are 53 relevant passages. After applying passage clustering, these were reduced to 18 distinct clusters. Next we give the query and 18 passages too LLM for intent generation. Subsequently, we generated intents using an LLM, resulting in 10 intents, as shown in Table 2.

We experimented with various prompts (some examples are shown in Fig below) following the Context-Aware Query Expansion (CAQE) method [1], which generates intents based on a query and a relevant document/passage pair. However, this approach resulted in a large number of intents, many of which were duplicates, as they were derived from individual <*query, passage*> pairs. To address this issue, we expanded the context to include a list of passages, which allowed us to generate a smaller number of high-quality intents by considering the collective context of all relevant passages (prompt in Section 3.1.1).

LLM Prompt using CAQE: Intent generation

Prompt 1: Given a query and document generate the intention of the query given the document. The generation should be like a human would write an intention in more than 3 words and less than 10 words. Use 'UNKNOWN' if there is no intent found. Response should as human like with minimum 3 words and maximum 10 words and not the answer to the query, only intention. **query**:{query} **document**:{doc}

Prompt 2: Being a intent generator, the task is to generate multiple intents of min 3 words. Given a query and a document, generate multiple distinct intents from the document that answers the query. Below are the rules to be followed. Answer in one short sentence per intent of minimum 3 words. Generate only distinct intents and not answers. Make sure intent is not the Query. Generate multiple intents but limit to maximum 3. Use 'UNKNOWN' if there is no intent found. Response should be in this specific format Query:: <query> Query_Intent:: <intent>. **query**:{query} **document**:{doc}

B INTENT CLUSTERING

As outlined in Section A, we perform clustering both before (passage clustering) and after (intent clustering) intent generation using a large language model (LLM) to eliminate redundancy. The same clustering approach is applied in both stages. For a given query, we first obtain embeddings for all passages or intents using Sentence-BERT (SBERT). Next, we select a passage/intent P_i/I_i and identify all other passages/intents that have a pairwise cosine similarity above a predefined threshold. Specifically, we use a threshold of 0.8 for passages and 0.9 for intents. All passages/intents that meet this similarity criterion are grouped into the same cluster, and then removed from the passage/intent list. This process is repeated iteratively until no passages/intents remain in the list. For the query "What is 311 for," we generated intents, which are listed in the "LLM generated intents" section of Table 2. Upon clustering these intents, we obtained two clusters: one consisting of 9 intents and another consisting of a single intent. Next we select intent representative of the cluster and reformulate it for the next step. So "differentiate between emergency and non-emergency numbers" is reformulated to "when to call 311" and representative reformulated intent from cluster 1 is "what services does the number 311 provide".

In addition to using SBERT with cosine similarity, we experimented with alternative methods for similarity scoring, including out-of-the-box and fine-tuned entailment models with both uni-directional and bi-directional entailment. To assess the quality of the clustering, we constructed an evaluation set on which all the clustering methods, for both passages and intents, were systematically evaluated. SBERT-based clustering approach performed better than the alternative methods.

C ANNOTATION VIA CROWDSOURCING

The collection of user intent annotations for DL-MIA is performed using a custom web application we implemented using the oTree framework [4].

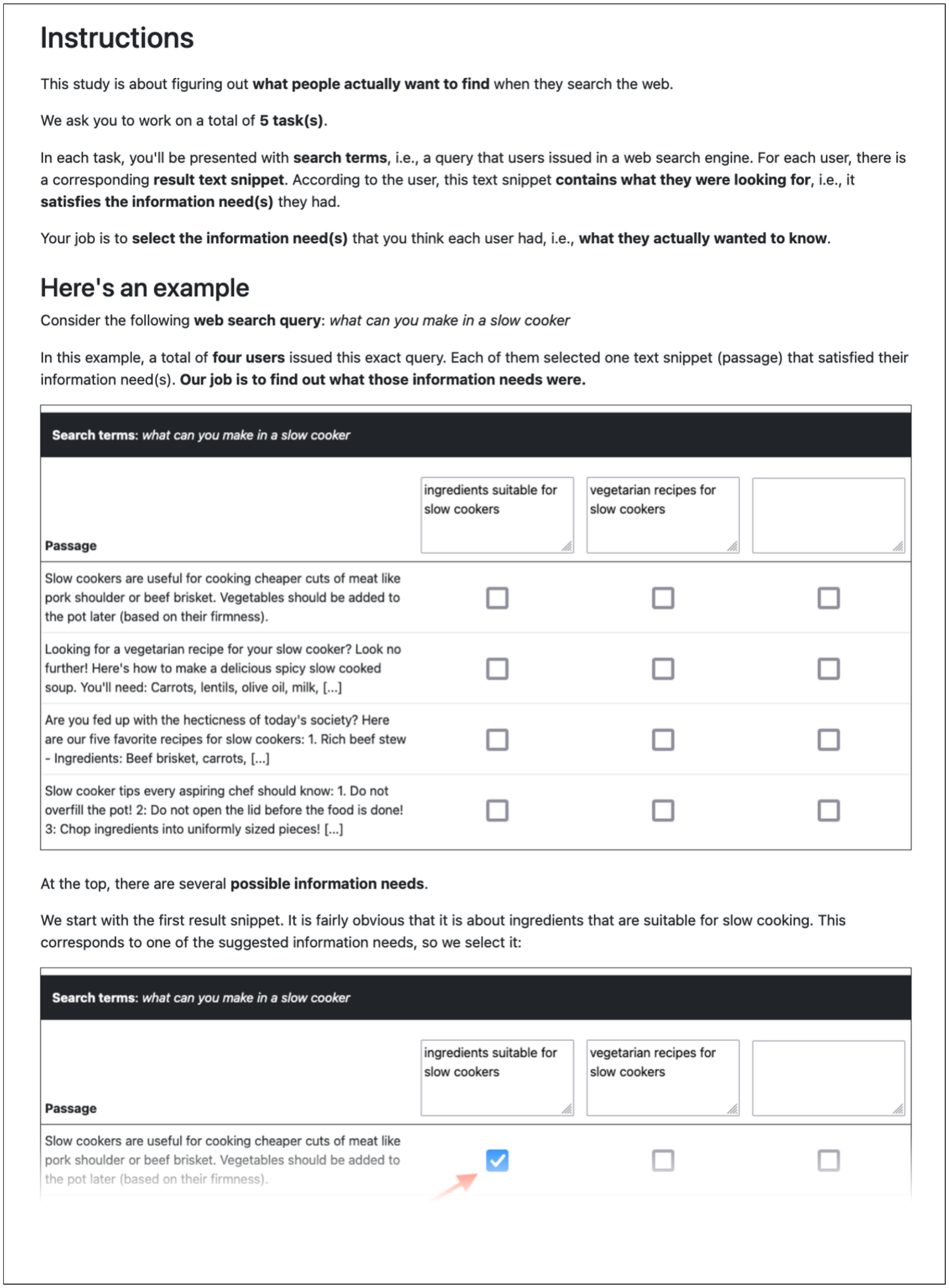
Each participant is presented with detailed instructions how to perform the task (cf. Fig. 5) in the beginning. The subsequent pages display **one (sub)query each** along with a list of the corresponding passages and intent candidates (cf. Fig. 6). The interface ensures that each passage either has at least one annotated relevant intent or it is specifically indicated that the passage in question is not relevant to the query at all. By displaying all passages and intents within the same page we make sure that the participant always maintains a mental overview over the distinct intents (i.e., aspects) of the current query.

The collected data is stored in a PostgreSQL database. After the collection is complete, oTree provides functionality to export the relevant data in CSV format. Our web application for data collection is publicly available.

D CLEANUP AND MERGING OF INTENTS POST-ANNOTATION

After the annotation phase, we perform a manual analysis of intents and corresponding relevance annotations for the passages. We show some of the scenarios and related intents in Table 3. We observe from example one that Intent 2 and Intent 3 are semantically similar and can be considered as redundant and hence are merged. When merging the intents, we also combine the ratings suing the following guidelines:

- We set the score based on number of annotations for the intents to be merged that indicate the level of agreement between annotators and relevance to the query
- For instance, if two annotators annotate 1 for both intents paired with a certain passage, we assign the relevance score as 2 for the merged intent.



Instructions

This study is about figuring out **what people actually want to find** when they search the web.

We ask you to work on a total of **5 task(s)**.

In each task, you'll be presented with **search terms**, i.e., a query that users issued in a web search engine. For each user, there is a corresponding **result text snippet**. According to the user, this text snippet **contains what they were looking for**, i.e., it **satisfies the information need(s)** they had.

Your job is to **select the information need(s)** that you think each user had, i.e., **what they actually wanted to know**.

Here's an example

Consider the following **web search query**: *what can you make in a slow cooker*

In this example, a total of **four users** issued this exact query. Each of them selected one text snippet (passage) that satisfied their information need(s). **Our job is to find out what those information needs were.**

Search terms: *what can you make in a slow cooker*

Passage	ingredients suitable for slow cookers	vegetarian recipes for slow cookers	
Slow cookers are useful for cooking cheaper cuts of meat like pork shoulder or beef brisket. Vegetables should be added to the pot later (based on their firmness).	☐	☐	☐
Looking for a vegetarian recipe for your slow cooker? Look no further! Here's how to make a delicious spicy slow cooked soup. You'll need: Carrots, lentils, olive oil, milk, [...]	☐	☐	☐
Are you fed up with the hecticness of today's society? Here are our five favorite recipes for slow cookers: 1. Rich beef stew - Ingredients: Beef brisket, carrots, [...]	☐	☐	☐
Slow cooker tips every aspiring chef should know: 1. Do not overfill the pot! 2: Do not open the lid before the food is done! 3: Chop ingredients into uniformly sized pieces! [...]	☐	☐	☐

At the top, there are several **possible information needs**.

We start with the first result snippet. It is fairly obvious that it is about ingredients that are suitable for slow cooking. This corresponds to one of the suggested information needs, so we select it:

Search terms: *what can you make in a slow cooker*

Passage	ingredients suitable for slow cookers	vegetarian recipes for slow cookers	
Slow cookers are useful for cooking cheaper cuts of meat like pork shoulder or beef brisket. Vegetables should be added to the pot later (based on their firmness).	☑	☐	☐

Figure 5: The instructions displayed to each crowdsourcing worker prior to the annotation process. Note that this screenshot is cropped and does not include the entire instructions.

Query: What is 311 for		
LLM generated intents (Section 3.1.1)		
LLM generated intents	To identify 311 as a number for non-emergency law enforcement related complaints, to explain 311 as access to various non-emergency municipal services, to highlight 311 for city information and non-emergency service requests, to outline the general usage of 311 for non-emergency information and services, to showcase the origin and adoption of 311 in various cities, understand the utility of 311 as a contact number, learn about 311's role in specific cities or counties, discover how to report specific issues with 311, find out about the origin and development of the 311 system, differentiate between emergency and non-emergency numbers.	10
Clustering (Section 3.1.2)		
Cluster 1	to identify 311 as a number for non-emergency law enforcement related complaints., to explain 311 as access to various non-emergency municipal services., to highlight 311 for city information and non-emergency service requests., to outline the general usage of 311 for non-emergency information and services., to showcase the origin and adoption of 311 in various cities., understand the utility of 311 as a contact number., learn about 311's role in specific cities or counties., discover how to report specific issues with 311., find out about the origin and development of the 311 system.	9
Cluster 2	differentiate between emergency and non-emergency numbers.	1
Crowdsourcing Intents (Section 3.1.3)		
Intents shown to Annotators	what services does the number 311 provide, when to call 311	2
Annotator generated intents	What services are provided by 311 in different cities, Availability of 311 system in city, Who answers the call when 311 is dialed, Availability of 311 services in different cities, when to call 311 rather than 911, what happens when one dials 311	6
Final Intents (Section 3.1.4)		
Final Intents	when to call 311, Availability of 311 services in different cities	2

Table 2: The table illustrates the various stages in refining LLM-generated intents to final intents for a given query. The right column displays the number of intents at each stage.

- If only one annotator assigned a score of 1 to both intents for the same passage we assign the score as 1.

Apart from redundant intents, we also observed cases where the machine generated or the custom intents from the user were not relevant to the core aspects of the query. For instance, in example 2 in Table 3, "the cost of Tuk-tuks" is irrelevant to the query which deals with aspects related to cost of living in Bangkok. Such intents are removed.

We also observe cases where the intents are same as query as shown in the table and these intents are removed. Finally, we also observe a scenario where the generated intents answer the query instead of conveying the explicit or latent aspects of the query as shown in Example 4 in Table 3. We remove such instances as they are actually not intents.

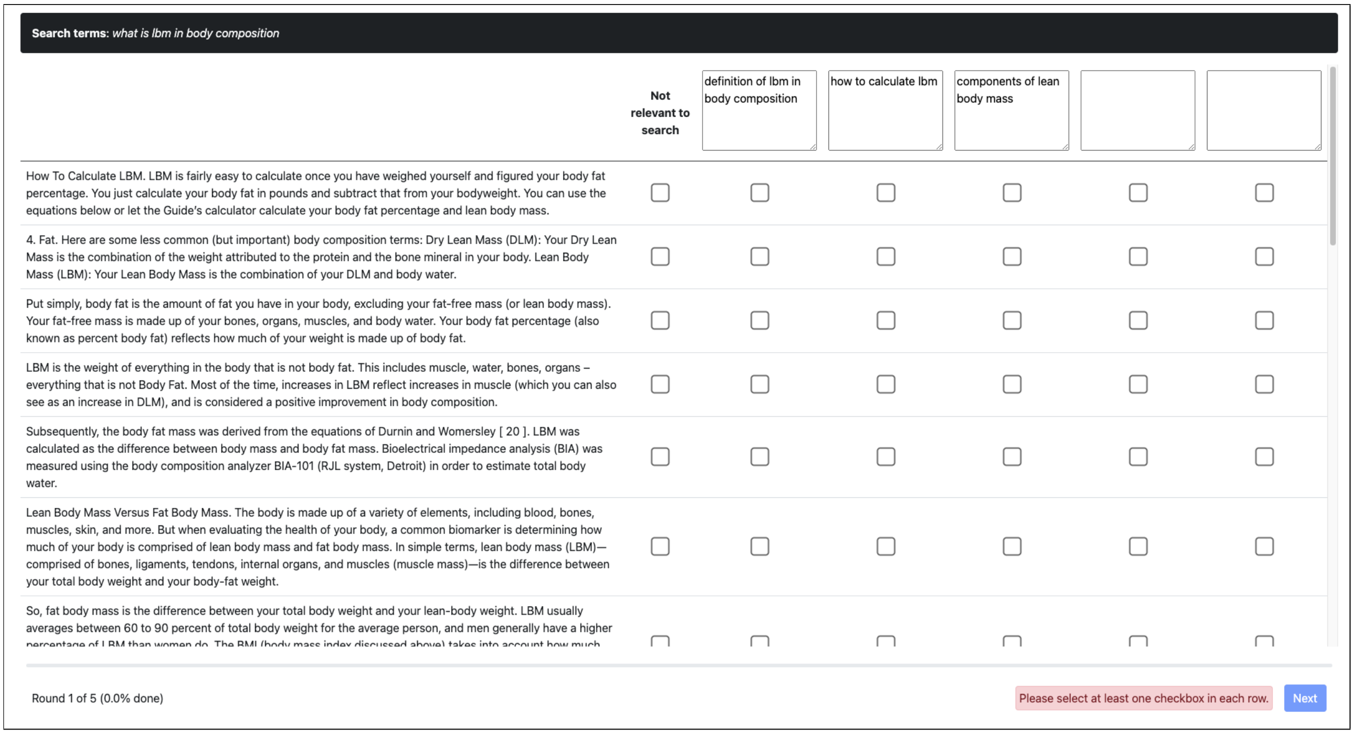


Figure 6: The user intent annotation interface for crowdsourcing workers. Each participant is asked to complete several *rounds*, where a round corresponds to one (sub)query and the corresponding list of passages to annotate. The page presents the original query (*search terms*) and the intent candidates (generated by the LLM) to the participant. At the top, the suggested intents can be modified and new intents can be added. Alternatively, it is possible to indicate that a given passage is not relevant to the search query at all.

Type	**Query with Intents**	Decision
Redundant Intents	**Query**: what vaccination should u give show piglets? **Intent 1:** available vaccinations for show piglets **Intent 2:** optional vaccinations for show piglets **Intent 3:** non-essential vaccination for show piglets	Merge intent 2 and intent 3
Intents not relevant to query	**Query**: How much money do i need in bangkok? **Intent 1:** how expensive is daily life in bangkok **Intent 2:** how expensive is tourism in bangkok ... **Intent 4:** cost of taxi/tuk-tuks	Remove Intent 4
Same as query	**Query**: when a house goes into foreclosure what happens to items on the premises? **Intent 1:** what happens to personal items when a house goes into foreclosure? **Intent 2:** what happens to fixtures when a house goes into foreclosure ... **Intent 8:** what physically happens to items after a house goes into foreclosure?	Remove Intent 8
Answers the query	**Query:** what is the name of the triangular region at the base of the bladder? **Intent 1:** Description of the trigone region? **Intent 2:** Synonyms of the term trigone in bladder	Remove Intent 1 and intent 2

Table 3: Examples of different scenarios for post-cleanup or merging of intents.