Query Understanding with Large Language Models

Eric Oliver Schmidt[1], Matthias Hagen[2], and Maik Fröbe[2]

[1] Leipzig University, Germany
[2] Friedrich-Schiller-Universität Jena, Germany

Abstract. Query understanding techniques can improve the effectiveness of search engines. Rule- or feature-engineering-based approaches for query understanding are highly efficient but require much engineering. Large language models can reduce the engineering effort to implement certain query understanding techniques. To compare large language models against traditional query understanding approaches, we use three query understanding tasks, namely query intent prediction, query segmentation, and query interpretation. We evaluate the query understanding approaches in intrinsic and extrinsic experiments. In intrinsic experiments, we use dedicated query-understanding test collections to measure accuracy in the corresponding query understanding task. For extrinsic experiments, we include the query understanding approaches in retrieval pipelines to measure the end-to-end effectiveness on the Robust04 collection. We find that large language models do not outperform traditional methods in query understanding evaluations. For the end-to-end retrieval effectiveness, we find that only a few of our query-understanding approaches improve the effectiveness. Potential reasons why most query understanding approaches do not improve effectiveness in a TREC-style evaluation might be that those tasks often contain only a few queries that are then focused on only a few retrieval tasks.

1 Introduction

Query understanding can help to improve the effectiveness of retrieval systems by refining and interpreting user queries [10, p. 190ff]. Consequently, query understanding received a big attention by the community so that many approaches have been published,[3] making it difficult to decide which approach could be helpful for a given retrieval scenario. Many of the traditional query understanding techniques are rule-based or use feature-engineering, and likely need to be adjusted with high effort to a new retrieval scenario. For this reason, comparing many approaches would require high engineering effort. Given that large language models (LLMs) can serve as foundation models [6], they might reduce the effort to identify which query understanding approaches could be interesting for a retrieval pipeline without high engineering effort. If a query understanding approach shows promise in exploratory experiments with LLMs, the approach could then be re-implemented with higher efficiency requirements for production.

[3] 4000 papers of the IR Anthology [28] match "query understanding" via ChatNoir [27].

Table 1. Overview of query understanding tasks that we compare in this paper.

Task	Description	Example	
		Query	**Output**
Intent Prediction	Analyze queries to determine user intent (e.g., navigational, transactional, informational).	`Symptoms of flu`	Informational
		`Buy iPhone`	Transactional
		`Google`	Navigational
Segmentation	Split multi-word queries into meaningful chunks.	`New York Times subscription`	[`New York Times`, `subscription`]
Interpretation	Understand ambiguous queries by considering context, user history, and linguistic structure.	`apple benefits`	`Health benefits (fruit)` or `stock benefits (Apple Inc.)`

Listing 1 Minimal workflow for evaluating query intent via our Python API.

```
evaluator = QueryIntentEvaluator()
results = evaluator.evaluate_frames(predictions_df,truths_df)
```

To verify if large language models can help low-effort prototyping for query understanding, we apply them to several query understanding tasks that we select from the WOWS 2024 workshop [12, 13] (that had a dedicated shared task on query understanding) and the query understanding blog of Daniel Tunkelang.[4] Table 1 shows an overview of our selected query understanding tasks. We assess the intrinsic effectiveness of large language models for each of the three tasks, as well as the extrinsic impact of these methods on end-to-end retrieval effectiveness. We find that LLM-based methods are almost as effective as the traditional query understanding approaches, but do not outperform the traditional approaches. Consequently, large language models can still help for low-effort prototyping to decide which query understanding approach could be promising for a given retrieval scenario. For extrinsic evaluations, we evaluate the end-to-end retrieval effectiveness on the Robust04 test collection, comparing our LLM approaches against standard baselines and an oracle upper bound. For query intent prediction, we find that TREC-style shared tasks mostly have informational queries, so that intent prediction does not help to improve retrieval effectiveness in the shared tasks that we considered. For the query segmentation and query interpretation task (see Table 1), we find that the oracle use of query understanding in a retrieval pipelines can improve the retrieval effectiveness of a baseline substantially, showing that those tasks can complement retrieval pipelines. However, neither LLM-based approaches nor traditional methods currently enhance retrieval across the complete query set of Robust04 (as one query understanding approach helps for one query but not for a second query, making fine-grained selections of query understanding a requirement). Finally, to address the high effort required to compare different query understanding approaches, we implement our query understanding evaluation in an re-usable API (see Listing 1 for an example). We release our code and all LLM predictions.[5]

[4] `https://queryunderstanding.com`
[5] `https://github.com/webis-de/CLEF-26`

2 Related Work

Query understanding approaches can be applied in interactive scenarios for direct interactions with searchers or in ad-hoc settings when the query understanding is solely applied in the backend of a search engine. Anand et al. [3] presented a generic framework for interactive scenarios to rewrite ambiguous queries as large language models can help users to better formulate their queries. Given that large language models can help users to better formulate their queries in such interactive settings, we investigate if they also can help for ad-hoc query understanding approaches when solely used in the backend of a search engine. The 2024 edition of the WOWS workshop [12, 13] had a dedicated shared task on ad-hoc query understanding for which we will next describe three tasks (see Table 1) and solutions that we re-use for our comparisons to verify if large language models can also help for ad-hoc query understanding.

Query intent prediction classifies the implicit goal of a query, which can be informational, navigational, or transactional [7]. Alexander et al. [2] presented a rule-based approach that was evaluated on ORCAS [9] and TIRA [15]. While their method demonstrated a high generalizability across a range of queries, its effectiveness may still be constrained in specific scenarios such as argumentative questions, where a deeper understanding of the context is required.

Query segmentation decomposes a user query into semantically meaningful phrases (e.g., segmenting `new york times subscription` into `[new york times] [subscription]`). These concepts can then be used to improve retrieval precision or query reformulation [18]. Hagen et al. evolved query segmentation from naïve n-gram frequencies [18] to the integration of Wikipedia titles [19], already noting that effective retrieval needs to properly select a suitable segmentation approach per query and that no single segmentation works for all query types [20]. Because a query may yield multiple plausible segmentations depending on user context, Hagen et al. [20] introduce new evaluation measures for segmentation that we re-implement in our experiments.

Query interpretation aims to disambiguate a query by mapping textual query segments to knowledge base entries (e.g., mapping the query `apple` to either *Apple Inc.* or the *fruit*). Kasturia et al. [24] presented a recall-oriented query interpretation approach that found ca. 90,000 Wikipedia-linked entities across 31 standard IR datasets [16], showing that most queries can be interpreted in more than one valid way. Erben et al. [11] investigated models such as GPT-3.5, Llama2, and FLAN-UL2 for reformulating the original query into similar ones without entity linking and query interpretation. Even though their approach improved recall, conventional baselines outperformed LLM-based approaches.

Ye et al. [31] investigated the intrinsic efficacy of LLMs such as GPT4 for a set of query understanding tasks. They found that while LLMs outperformed traditional baseline models in higher-level semantic tasks like query expansion and intent prediction, they were outperformed by non-LLMs in term-level tasks such as word segmentation. To mitigate the high inference latency of LLMs in real-world search scenarios, the authors proposed using LLMs to generate and filter unsupervised training samples to fine-tune smaller, faster models. Rather

Task details:
Classify the intent of the following search query:
`<QUERY>`

Categories:

- Informational
- Navigational
- Transactional

Instructions:
Only give the intent category in one word.

Fig. 1. Basic prompt for intent prediction.

than addressing query understanding sequentially, unified models execute them simultaneously to prevent error propagation. Guo et al. [17] proposed a discriminative conditional random field (CRF-QR) model for unified query refinement, demonstrating improved accuracy and retrieval relevance. Extending this idea, Bendersky et al. [4] showed that jointly modeling multiple linguistic features improves annotation accuracy compared to independent annotation methods.

3 Query Understanding with LLMs

We build our experiments to (i) include different LLM models, (ii) properly select prompt templates, and (iii) infrastructure constraints (API limits, reproducibility). We use the free tier of Groq[6] for our experiments using only open-source large language models in their default settings.

Choice of Models. To assess differences in model capacity and inference costs, we evaluate a selection of open-access LLMs via the Groq free-tier API. Our selection included two 8B parameter models optimized for throughput and cost-efficiency: Llama-3-8B-8192 and its successor, Llama-3.1-8B-Instant. We also evaluated two 70B parameter models: Llama-3.3-70B-Versatile from Meta, and DeepSeek-R1 Distill. While the 70B models offer stronger capacity, they also come with the highest latencies and most restrictive API usage limits, which impacts their practical deployment and large-scale usage.

Prompt Design and Formulation. For our query understanding task, we evaluated prompts across three strategies: zero-shot, few-shot, and Chain-of-Thought (CoT) prompting. For query intent prediction, we designed ten prompt templates to classify queries into the categories informational, navigational, and transactional [7]. These templates systematically vary in complexity (e.g., prompt

[6] `https://groq.com`

System Instructions:
You are an expert at understanding the intent behind search queries. Classify the following queries into:

- **Informational:** Looking for knowledge or answers
- **Navigational:** Trying to reach a specific website or brand
- **Transactional:** Intending to act, buy, or download

Examples:
- "how to make a pie crust" → Informational
- "facebook login" → Navigational
- "buy iPhone 15 pro max" → Transactional
- "download adobe photoshop" → Transactional
- "tesla careers" → Navigational
- "tesla share price" → Informational
- "amazon" → Navigational
- "buy amazon prime membership" → Transactional

Task details:
Provide explanation for the intent and give your answer with one of the categories: informational, navigational or transactional only.
One of the categorical values is compulsory in the answer.
Instructions: Think about the query. Explain your reasoning first. Only provide the intent category in the last line.

Input:
Query: `<QUERY>`
Explanation:

Fig. 2. Prompt with detailed examples and explanation for intent prediction.

length, role-prompting), the number of in-context examples, and CoT instructions. Figure 1 shows a zero-shot prompt which requests a strict classification output. Figure 2 demonstrates a more complex template with expert role-prompting and CoT reasoning before generating the final classification. Our prompts for query segmentation explicitly target ambiguous queries or retrieval-oriented optimal segmentation by adopting the "in doubt without" principle proposed by Hagen et al. [20], leaving many queries unsegmented in favor of retrieval effectiveness. Finally, to generate query interpretations, the LLM must first extract the relevant named entities. We compared two strategies: a joint approach where the model outputs both entities and the interpretation in a single step, and an approach where the model is constrained to return either entity names or their corresponding Wikipedia URLs, depending on the prompt, without the interpretation. For the latter, we re-implemented the greedy interpretation finding (GIF) algorithm [21] to derive interpretations from the extracted entities, aligning with the methodology of Kasturia et al. [24].

Reproducibility and Output Parsing. To ensure reproducibility, all LLM requests and full API responses were logged in an OpenAI-compatible format. Extracting the classification or final answer from the LLM's raw output proved difficult, as the models frequently generated plausible but verbose responses. Therefore, we added an instruction to the prompts requiring the model to state its final answer on the concluding line (inspired by umBRELA [29]).

Hyperparameter Tuning and Prompt Selection. We select our prompt for each LLM on a validation dataset. For each task, we used a pilot study with a few examples to design ten prompts. Then, all ten prompt templates per task were evaluated across all four models on task-specific validation sets with task-specific evaluation measures. For query intent prediction, we selected the prompt for a model via the Macro-F1 classification score. For query segmentation, we selected the prompt for a model based on the average of the five query segmentation evaluation metrics proposed by Hagen et al [20]. For query interpretation, we used the F1 score for prompt selection. Given that performing query understanding on real-world queries comprises ambiguity, we did not choose a temperature of zero for inference. Instead, we used the default temperature to sample five responses per query, allowing us to obtain a distribution of outputs and leverage multiple reasoning paths rather than relying on deterministic greedy decoding [30]. To select a prompt template for each task, we employed a hold-out validation strategy. We evaluated all candidate prompts on a dedicated validation set by taking the majority vote of these five samples. We then selected for each model and task the prompt that achieved the best effectiveness per task on the validation set. This best-performing prompt was then applied unchanged to the test set. In the remaining part of the paper, we report the results for the model–prompt combination that we selected this way. We note that there was no prompt that worked well on our validation set for each model, indicating that models prefer different ways of describing the query understanding task.

4 Intrinsic Evaluation

We evaluate the large language model query understanding approaches in an (1) intrinsic evaluation to measure the effectiveness in the specific query understanding task, and an (2) extrinsic evaluation to assess its impact on retrieval effectiveness. For the intrinsic evaluation, we compare our LLM-based intent prediction with the results from WOWS 2024 [13], and for segmentation and interpretation, we compare with the approaches of Hagen et al. [18, 19, 20] and Kasturia et al. [24]. We use the baselines as submitted to the WOWS workshop and our LLM approaches in the variant selected via hyperparameter tuning.

4.1 Query Intent Prediction

To evaluate our large language model-based query intent predictor, we use the ORCAS-I dataset [1],[7] a labeled extension of the publicly available ORCAS dataset [9] that augments Bing search queries with three intent categories: informational, navigational, and transactional. From the provided 2-million-record subset (ORCAS-I-2M), we sampled 150 queries for prompt engineering and hyperparameter tuning (50 per intent class). For the final evaluation, we utilized the manually annotated ORCAS-I-gold test set with 1,000 queries. Because of

[7] `https://researchdata.tuwien.ac.at/records/pp7xz-n9a06`

Table 2. Intrinsic macro-averaged precision (P), recall (R), and F1 scores for query intent prediction on ORCAS-I-gold. For intent prediction, the most frequent intent from five runs was chosen. LLM results are compared against BERT and Snorkel baselines [2]. Darker cells indicate higher scores, and bolded cells highlight the best score per metric.

Model	Intent results		
	P	R	F1
llama3-8b	0.608	0.590	0.596
llama-3.1	0.632	0.631	0.630
llama-3.3	0.603	0.698	0.613
deepseek	0.618	**0.716**	0.633
BERT	0.742	0.705	**0.717**
Snorkel	**0.771**	0.648	0.667

the class imbalance within ORCAS-I, in which informational intent dominates (81.33%) [2], we report macro-averaged scores for precision, recall and F1. Table 2 presents the evaluation scores. To mitigate the stochastic effects of LLMs, each query was evaluated five times, and the final intent was determined via majority voting. While the 70B parameter models showed noticeably higher variance across individual runs compared to the 8B models, majority voting stabilized their predictions, and also improved the effectiveness of the 8B models. Table 2 also benchmarks our approach against the WOWS-baselines [2]. The highest Macro-F1 score achieved by our LLM-based approach was approximately 0.63. While this is slightly below the weakly supervised Snorkel (0.67) and below the supervised BERT (0.72), it is a competitive result (as no fine-tuning was needed).

4.2 Query Segmentation

For the query segmentation task, our experiments rely on the Webis Query Segmentation Corpus 2010 (Webis-QSeC-10) [19],[8] a large-scale evaluation corpus derived from the AOL query log. For prompt engineering, we sampled 200 random queries from the training split; for intrinsic evaluation, we sampled 1,000 queries from the test split. Table 3 shows the segmentation accuracy across the evaluated LLMs. To mitigate variability, we applied majority voting across five runs per query. Llama-3.3-70B-Versatile consistently outperformed the other models across all metrics. While the larger 70B models yielded higher accuracy on individual outputs, the smaller 8B models benefit more from majority voting, reducing the accuracy gap through aggregation. In Table 3, we also benchmark the most accurate LLM, Llama-3.3-70B-Versatile, against traditional and hybrid baselines [20]. While the LLM exceeds early baselines such as pointwise mutual information (PMI) [18] and naive segmentation [19], it falls short of sophisticated hybrid algorithms like the Wikipedia titles and strict noun phrases (WT+SNP) baseline and accuracy-oriented hybrid segmentation (HYB-A) [20].

[8] `https://webis.de/data/webis-qsec-10.html`

Table 3. Segmentation accuracy on 1,000 test queries from Webis-QSeC-10 and comparison against traditional and hybrid baseline algorithms [20]. LLM outputs are aggregated via majority voting across five runs. Darker cells indicate higher accuracy.

Accuracy measure		Model				Algorithm			
Selector	Acc. level	llama 3-8b	llama 3.1	llama 3.3	deepseek	PMI	WT+ SNP	WT	HYB-A
weighted best fit unless absolute majority	query	0.351	0.360	0.501	0.443	0.473	0.555	0.589	**0.606**
	seg prec	0.585	0.590	0.691	0.629	0.627	0.720	0.732	**0.747**
	seg rec	0.636	0.637	0.668	0.625	0.616	0.698	**0.758**	0.751
	seg F	0.602	0.604	0.671	0.620	0.617	0.705	0.740	**0.744**
	break	0.696	0.705	0.784	0.741	0.772	0.807	0.826	**0.831**
break fusion	query	0.315	0.324	0.440	0.396	0.399	0.469	**0.536**	0.524
	seg prec	0.554	0.561	0.655	0.600	0.578	0.664	**0.696**	0.692
	seg rec	0.607	0.607	0.620	0.591	0.559	0.627	**0.716**	0.685
	seg F	0.572	0.574	0.629	0.587	0.563	0.640	**0.701**	0.683
	break	0.671	0.679	0.744	0.707	0.716	0.753	**0.792**	0.780

Table 4. Interpretation results on a 200-query sample of the Webis-QInC-22 test set, restricted to interpretations rated at least "moderately likely" [24]. The sample has comparable average difficulty to the full test set. Metrics are calculated for complete matches. For the LLMs, we report the metrics of the majority vote of five iterations. Darker cells indicate higher scores, and bolded cells highlight the best score per metric.

(a) Results for LLM-based approaches

Model	Precision	Recall	F1
llama-3.1	0.228	0.270	0.214
llama3-8b	0.301	**0.295**	0.263
llama-3.3	**0.345**	0.283	**0.292**
deepseek	0.314	0.258	0.263

(b) Results for traditional approaches

Approach	Precision	Recall	F1
Kasturia et al. [24]	**0.501**	**0.470**	**0.446**
Dexter [8]	0.295	0.223	0.237
Nordlys EL [21, 22, 23]	0.280	0.211	0.225
FEL [5]	0.195	0.161	0.168

Consequently, for standalone query segmentation, simple heuristic approaches remain the state-of-the-art in both accuracy and computational efficiency.

4.3 Query Interpretation

For our evaluation of query interpretation, we used the Webis Query Interpretation Corpus 2022 (Webis-QInC-22) [24].[9] The corpus comprises 2,234 queries, each of which contains at least one entity and, on average, two interpretations assigned a probability (i.e., plausible, moderate, very likely). Each query is also annotated with a difficulty level (1–5, average 1.77) that we used to randomly sample two subsets that maintain the average difficulty, each containing 200 queries, for prompt engineering and testing, respectively. Table 4 presents the

[9] `https://webis.de/data/webis-qinc-22.html`

Table 5. Predicted distributions of informational (I), navigational (N), and transactional (T) query intents for each LLM across Web and news tracks corpora.

	Web ad-hoc												Financial News			Q&A		
LLM	**ClueWeb09**			**ClueWeb12**			**GOV**			**GOV2**			**Disks45**			**MSMARCO**		
	I	N	T	I	N	T	I	N	T	I	N	T	I	N	T	I	N	T
deepseek	147	31	22	242	2	7	296	23	6	144	2	4	346	1	3	97	0	0
llama3-8b	164	32	4	250	1	0	313	12	0	147	1	2	349	1	0	97	0	0
llama-3.1	164	31	5	249	1	1	312	12	1	146	2	2	349	1	0	97	0	0
llama-3.3	156	27	17	243	2	6	281	18	26	141	2	7	340	1	9	95	0	2

interpretation effectiveness on the test sample. We report precision, recall, and F1 scores for complete matches. A predicted interpretation is only a complete match if both its segmentation and its contained entities align with a ground truth interpretation [24], whereas for partial matches at least the entities of a ground truth interpretation would be enough. Following Kasturia et al. [24], we restrict our ground truth target to interpretations rated at least "moderately likely", i.e., more or most searchers would mean this interpretation. Consistent with our findings in intent prediction and segmentation, aggregating predictions via majority voting improved effectiveness across all models, with 8B models benefiting the most. Finally, we benchmarked our approach against the Fast Entity Linker (FEL) [5], Nordlys EL [21, 22, 23], Dexter [8] and the approach from Kasturia [24]. Llama-3.3-70B-Versatile achieved a Macro-F1 of 0.292, which exceeded the Dexter, Nordlys and FEL baselines, but was clearly outperformed by the approach from Kasturia [24] with a score of 0.446.

5 Extrinsic Evaluation

In the previous section we saw that LLMs can perform query understanding tasks (at high costs). In this chapter we evaluate the extrinsic impact of our query intent prediction, segmentation, and interpretation approaches on the TREC Robust04 benchmark (Disks 4 & 5), a standard collection of newswire documents chosen for its exhaustive human relevance judgments and its status as a baseline for hard retrieval topics. The documents, queries, and relevance judgments are loaded with `ir_datasets` [25], which provides standardized access. All experiments are implemented using PyTerrier [26]. Where applicable, we incorporate pre-computed results from the TIRA [15] evaluation platform,[10] including retrieval runs or traditional baselines, to ensure reproducibility.

5.1 Query Intent Prediction

Our initial motivation was an observation that BM25 can be substantially more effective for navigational queries whereas monoT5 can be highly effective for informational queries [14]. Hence, our hypothesis was that query intent prediction

[10] `https://github.com/tira-io/tira/blob/main/python-client/tira/tirex.py`

Table 6. Extrinsic retrieval effectiveness (nDCG@10) for query segmentation on Robust04. Superscripts indicate Cohen's d relative to the no-segmentation baseline (None), and statistically significant improvements are marked bold. Underlined values indicate the highest score per Retriever, except the Oracle (best approach by design).

	Approach	a) Oracle			b) Realistic		
		BM25	PL2	Tf	BM25	PL2	Tf
BL	None	.414	.411	.080	.414	.411	.080
	OPT	**.539**$^{\uparrow 0.4}$	**.526**$^{\uparrow 0.4}$	.391$^{\uparrow 1.2}$	**.539**$^{\uparrow 0.4}$	**.526**$^{\uparrow 0.4}$	.391$^{\uparrow 1.2}$
LLMs	deepseek	.354$^{\downarrow 0.3}$	.345$^{\downarrow 0.3}$	**.204**$^{\uparrow 0.3}$	.330$^{\downarrow 0.3}$	.319$^{\downarrow 0.4}$	.149$^{\uparrow 0.2}$
	llama-3.3	.359$^{\downarrow 0.3}$	.342$^{\downarrow 0.3}$	**.228**$^{\uparrow 0.4}$	.341$^{\downarrow 0.3}$	.324$^{\downarrow 0.4}$	**.187**$^{\uparrow 0.3}$
	llama3	.408$^{\downarrow 0.1}$	.406$^{\downarrow 0.1}$	.167$^{\uparrow 0.2}$	.389$^{\downarrow 0.1}$	.389$^{\downarrow 0.1}$	.104$^{\uparrow 0.1}$
	llama-3.1	.402$^{\downarrow 0.1}$	.395$^{\downarrow 0.1}$	.158$^{\uparrow 0.1}$	.374$^{\downarrow 0.2}$	.371$^{\downarrow 0.2}$	.091$^{\uparrow 0.1}$
Traditional	HYB-A	.330$^{\downarrow 0.3}$	.320$^{\downarrow 0.3}$	**.238**$^{\uparrow 0.4}$	.294$^{\downarrow 0.3}$	.272$^{\downarrow 0.3}$	**.151**$^{\uparrow 0.3}$
	HYB-B	.454$^{\uparrow 0.1}$	.439$^{\uparrow 0.1}$	**.278**$^{\uparrow 0.7}$	.433$^{\uparrow 0.1}$	.398$^{\uparrow 0.0}$	**.156**$^{\uparrow 0.4}$
	HYB-I	.418$^{\downarrow 0.0}$	.405$^{\downarrow 0.0}$	**.261**$^{\uparrow 0.6}$	.411$^{\downarrow 0.0}$	.390$^{\downarrow 0.1}$	**.183**$^{\uparrow 0.4}$
	WIKI	.294$^{\downarrow 0.4}$	.286$^{\downarrow 0.4}$	**.221**$^{\uparrow 0.4}$	.274$^{\downarrow 0.4}$	.263$^{\downarrow 0.5}$	**.171**$^{\uparrow 0.3}$
	WT	.412$^{\downarrow 0.0}$	.400$^{\downarrow 0.1}$	**.278**$^{\uparrow 0.7}$	.404$^{\downarrow 0.0}$	.387$^{\downarrow 0.1}$	**.221**$^{\uparrow 0.5}$
	WT+SNP	.317$^{\downarrow 0.3}$	.311$^{\downarrow 0.4}$	**.200**$^{\uparrow 0.3}$	.296$^{\downarrow 0.4}$	.285$^{\downarrow 0.4}$	.143$^{\uparrow 0.2}$

could help to route between different retrieval models. We did run all our query intent predictors on a set of TREC-style shared tasks. Table 5 shows the results, indicating that Robust04 (Disks45) has almost only informational queries. This aligns with previous findings [14], and highlights that it does not make sense to apply query intent prediction on our extrinsic evaluation on Robust04.

5.2 Query Segmentation

Following Hagen et al. [20], we filtered the evaluation dataset to include only queries with at least three keywords and at least one relevant document. Table 6 presents the extrinsic retrieval effectiveness (nDCG@10) on the TREC 2004 Robust Track (Robust04) collection. We evaluated our LLM-generated segmentations against traditional algorithmic approaches [20], loading pre-computed baseline segmentations from TIRA to ensure standardized comparisons. As proposed in Hagen et al. [20], we employ two baselines: the optimum segmentation oracle (OPT) that is the theoretical upper bound and the no-segmentation (None) which is the unmodified query. The OPT oracle selects the segmentation that maximizes nDCG@10 per query. For all approaches, we report an "optimistic" assessment by selecting the formulation with the highest nDCG score. These formulations included applying Boolean operators (`AND`, `OR`), phrase quoting, query term boosting, and optimizing the target index field (e.g., title, default text, or main content) for phrase search on a positional index. The results show that while the OPT oracle achieved substantial improvements over the None baseline, the practical gains from real-world segmentations are modest. Segmentation generally led to slight decreases in nDCG for BM25 and PL2, though therm frequency (Tf) retrieval was improved by both LLM and traditional approaches.

Table 7. Extrinsic retrieval effectiveness (nDCG@10) for query interpretation on TREC Robust04. The oracle represents the theoretical upper bound. Underlined values indicate the highest score per retriever. Superscripts indicate Cohen's d relative to the baseline (None), and statistically significant improvements are highlighted in bold.

	Approach	a) Oracle			b) Realistic		
		BM25	PL2	Tf	BM25	PL2	Tf
BL	None	.414	.411	.080	.414	.411	.080
LLMs	deepseek	**.554**$^{\uparrow 0.5}$	**.543**$^{\uparrow 0.4}$	**.213**$^{\uparrow 0.7}$	.334$^{\downarrow 0.3}$	.336$^{\downarrow 0.2}$	.020$^{\downarrow 0.5}$
	llama3	**.558**$^{\uparrow 0.5}$	**.546**$^{\uparrow 0.5}$	**.215**$^{\uparrow 0.7}$	.319$^{\downarrow 0.3}$	.320$^{\downarrow 0.3}$	.022$^{\downarrow 0.5}$
	llama-3.1	**.581**$^{\uparrow 0.6}$	**.569**$^{\uparrow 0.5}$	**.223**$^{\uparrow 0.7}$	.327$^{\downarrow 0.3}$	.327$^{\downarrow 0.3}$	.019$^{\downarrow 0.5}$
	llama-3.3	**.554**$^{\uparrow 0.5}$	**.544**$^{\uparrow 0.5}$	**.212**$^{\uparrow 0.7}$	.343$^{\downarrow 0.2}$	.346$^{\downarrow 0.2}$	.019$^{\downarrow 0.5}$
WOWS	Gohsen et al. [16]	**.525**$^{\uparrow 0.4}$	**.517**$^{\uparrow 0.4}$	**.199**$^{\uparrow 0.6}$	.298$^{\downarrow 0.4}$	.302$^{\downarrow 0.4}$	.032$^{\downarrow 0.4}$

A key challenge for the LLM approach was the generation of invalid segmentations, which were retained in the evaluation and scored as incorrect. Traditional algorithms such as HYB-B and the WT-baseline achieved higher nDCG scores than LLM-based approaches, confirming that algorithmic query segmentations remain a strong standard. As Hagen et al. [20] previously pointed out, intrinsically accurate segmentations do not guarantee improved retrieval effectiveness, which is an interesting starting point for future work.

5.3 Query Interpretation

While previous research have significantly advanced query interpretation methodologies [16, 24], their extrinsic impact on retrieval effectiveness remains largely unexplored. We benchmark our LLM-based approach against the interpretations generated by Gohsen et al. [16], which are publicly available for the Robust04 collection. Analogous to query segmentation, query interpretation introduces a complex hyperparameter space which includes the choice of retriever, target index field (e.g., default text, main content, title), and the scope of the entity expansion (e.g., restricting to the Wikipedia title versus appending introductory sentences). Table 7 presents our results. The oracle represents an optimistic upper bound, selecting the hyperparameter combination that maximizes nDCG@10 per query. Conversely, the realistic condition applies a static, naive heuristic: searching the default text index and expanding queries using both the Wikipedia title and its first three sentences. The results show that there is a big gap between the oracle and the realistic retrieval pipeline. While the oracle exhibits substantial improvements over the baseline, the realistic heuristic severely degraded retrieval effectiveness across all approaches. This indicates that while query interpretation can improve retrieval effectiveness, applying it in practice is difficult, as a production deployment would require sophisticated routing.

Table 8. LLM efficiency comparison for query intent prediction, segmentation, and interpretation tasks on Robust04. Metrics show the average accumulated total tokens and completion time (seconds) per query across five iterations for majority voting.

	Intent Prediction		**Segmentation**		**Interpretation**	
LLM	Tokens	Time (s)	Tokens	Time (s)	Tokens	Time (s)
deepseek	1,648.5	6.21	1,612.0	7.33	2,504.2	6.10
llama-3.1	1,873.9	1.37	354.7	0.06	6,882.4	2.67
llama-3.3	1,623.4	2.82	516.2	1.33	4,189.3	4.63
llama3-8b	260.4	0.03	248.3	0.05	6,293.4	1.48

5.4 Efficiency

A primary disadvantage of LLMs in search pipelines is their computational overhead compared to traditional approaches. Table 8 shows the average token usage and inference latency collected from the API responses (summed across the five iterations required for majority voting per query). The results on Robust04 reveal strong variations across tasks and models. For instance, the smaller Llama-3-8B-8192 model demonstrates high efficiency in intent prediction (0.02 s, 255.1 tokens) and segmentation (0.05 s, 233.7 tokens). However, query interpretation leads to a higher average latency of 1.17 s. Conversely, larger models like DeepSeek-R1 Distill show substantial higher latency, showing that only prototypical usage of LLMs in online query understanding is possible.

6 Conclusion and Future Work

We studied LLM-based query intent prediction, query segmentation, and query interpretation in TREC-style information retrieval experiments, alongside intrinsic evaluation of system effectiveness. From an intrinsic perspective, our results demonstrate that LLMs can achieve effectiveness comparable to more traditional baselines, even if they do not reach the effectiveness of the traditional baselines.

Extrinsically, however, we found that intent prediction has a limited impact on retrieval effectiveness because most queries in TREC-style shared tasks are informational queries. We also observed that, in our setups, applying single query segmentation and interpretation approaches did not lead to the improved retrieval effectiveness seen for oracle upper bounds. This raises the question of which query understanding tasks are most useful in which scenarios, and how they should be combined. Our results show that large language models can help in this scenario, as they can allow for fast prototyping.

Future research could focus on bridging the gap between intrinsic query understanding and actual end-to-end retrieval effectiveness. As Alexander et al. [2] hypothesize, transformer-based models may struggle to express their full capability on short search queries. To address this, it could be interesting to, rather than considering each task in isolation, perform all tasks holistically with the goal to improve the end-to-end retrieval effectiveness.

Bibliography

[1] D. Alexander, W. Kusa, and A. P. de Vries. ORCAS-I: queries annotated with intent using weak supervision. In E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, and G. Kazai, editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 3057–3066. ACM, 2022. https://doi.org/10.1145/3477495.3531737. URL https://doi.org/10.1145/3477495.3531737.

[2] D. Alexander, W. Kusa, and A. P. de Vries. ORCAS-I query intent predictor as component of TIRA. In S. M. Farzana, M. Fröbe, G. Hendriksen, M. Granitzer, D. Hiemstra, M. Potthast, and S. Zerhoudi, editors, *Proceedings of the first International Workshop on Open Web Search co-located with the 46th European Conference on Information Retrieval ECIR 2024, Glasgow, UK, March 28, 2024*, volume 3689 of *CEUR Workshop Proceedings*, pages 23–29. CEUR-WS.org, 2024. URL https://ceur-ws.org/Vol-3689/WOWS_2024_paper_3.pdf.

[3] A. Anand, V. V, A. Anand, and V. Setty. Query understanding in the age of large language models. *CoRR*, abs/2306.16004, 2023. https://doi.org/10.48550/ARXIV.2306.16004. URL https://doi.org/10.48550/arXiv.2306.16004.

[4] M. Bendersky, W. B. Croft, and D. A. Smith. Joint annotation of search queries. In D. Lin, Y. Matsumoto, and R. Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 102–111. The Association for Computer Linguistics, 2011. URL https://aclanthology.org/P11-1011/.

[5] R. Blanco, G. Ottaviano, and E. Meij. Fast and space-efficient entity linking for queries. In X. Cheng, H. Li, E. Gabrilovich, and J. Tang, editors, *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015, Shanghai, China, February 2-6, 2015*, pages 179–188. ACM, 2015. URL https://doi.org/10.1145/2684822.2685317.

[6] R. Bommasani, D. A. Hudson, E. Adeli, R. B. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. S. Chatterji, A. S. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. E. Gillespie, K. Goel, N. D. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. S. Krass, R. Krishna, R. Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021. URL https://arxiv.org/abs/2108.07258.

[7] A. Z. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002. URL https://doi.org/10.1145/792550.792552.

[8] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani. Dexter: an open source framework for entity linking. In P. N. Bennett, E. Gabrilovich, J. Kamps, and J. Karlgren, editors, *ESAIR'13, Proceedings of the Sixth International Workshop on Exploiting Semantic Annotations in Information Retrieval, co-located with CIKM 2013, San Francisco, CA, USA, October 28, 2013*, pages 17–20. ACM, 2013. URL https://doi.org/10.1145/2513204.2513212.

[9] N. Craswell, D. Campos, B. Mitra, E. Yilmaz, and B. Billerbeck. ORCAS: 18 million clicked query-document pairs for analyzing search. *CoRR*, abs/2006.05324, 2020. URL https://arxiv.org/abs/2006.05324.

[10] W. B. Croft, D. Metzler, and T. Strohman. *Search Engines - Information Retrieval in Practice*. Pearson Education, 2009. ISBN 978-0-13-136489-9. URL http://www.search-engines-book.com/.

[11] L. Erben, M. Hampel, M. Kuns, V. Melisch, P. Natzschka, W. Pertsch, L. Razouk, R. Stolle, R. T. Thoss, T. G. Trinh, J. Gonsior, and A. Reusch. Assembling four open web search components: TU dresden at WOWS 2024. In S. M. Farzana, M. Fröbe, G. Hendriksen, M. Granitzer, D. Hiemstra, M. Potthast, and S. Zerhoudi, editors, *Proceedings of the first International Workshop on Open Web Search co-located with the 46th European Conference on Information Retrieval ECIR 2024, Glasgow, UK, March 28, 2024*, volume 3689 of *CEUR Workshop Proceedings*, pages 73–93. CEUR-WS.org, 2024. URL https://ceur-ws.org/Vol-3689/WOWS_2024_paper_8.pdf.

[12] S. M. Farzana, M. Fröbe, M. Granitzer, G. Hendriksen, D. Hiemstra, M. Potthast, A. P. de Vries, and S. Zerhoudi. Report on the 1st international workshop on open web search (WOWS 2024) at ECIR 2024. *SIGIR Forum*, 58(1):1–13, 2024. https://doi.org/10.1145/3687273.3687290. URL https://doi.org/10.1145/3687273.3687290.

[13] S. M. Farzana, M. Fröbe, G. Hendriksen, M. Granitzer, D. Hiemstra, M. Potthast, and S. Zerhoudi, editors. *Proceedings of the first International Workshop on Open Web Search co-located with the 46th European Conference on Information Retrieval ECIR 2024, Glasgow, UK, March 28, 2024*, volume 3689 of *CEUR Workshop Proceedings*, 2024. CEUR-WS.org. URL https://ceur-ws.org/Vol-3689.

[14] M. Fröbe, S. Günther, M. Probst, M. Potthast, and M. Hagen. The power of anchor text in the neural retrieval era. In M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvåg, and V. Setty, editors, *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part I*, volume 13185 of *Lecture Notes in Computer Science*, pages 567–583. Springer, 2022. https://doi.org/10.1007/978-3-030-99736-6_38. URL https://doi.org/10.1007/978-3-030-99736-6_38.

[15] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, and M. Potthast. Continuous Integration for Reproducible Shared Tasks with TIRA.io. In *Advances in Information Retrieval.*

45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Berlin Heidelberg New York, Apr. 2023. Springer.

[16] M. Gohsen and B. Stein. Integrating query interpretation components into the information retrieval experiment platform. In S. M. Farzana, M. Fröbe, G. Hendriksen, M. Granitzer, D. Hiemstra, M. Potthast, and S. Zerhoudi, editors, *Proceedings of the first International Workshop on Open Web Search co-located with the 46th European Conference on Information Retrieval ECIR 2024, Glasgow, UK, March 28, 2024*, volume 3689 of *CEUR Workshop Proceedings*, pages 63–72. CEUR-WS.org, 2024. URL https://ceur-ws.org/Vol-3689/WOWS_2024_paper_7.pdf.

[17] J. Guo, G. Xu, H. Li, and X. Cheng. A unified and discriminative model for query refinement. In S. Myaeng, D. W. Oard, F. Sebastiani, T. Chua, and M. Leong, editors, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 379–386. ACM, 2008. https://doi.org/10.1145/1390334.1390400. URL https://doi.org/10.1145/1390334.1390400.

[18] M. Hagen, M. Potthast, B. Stein, and C. Bräutigam. The power of naive query segmentation. In F. Crestani, S. Marchand-Maillet, H. Chen, E. N. Efthimiadis, and J. Savoy, editors, *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 797–798. ACM, 2010. https://doi.org/10.1145/1835449.1835621. URL https://doi.org/10.1145/1835449.1835621.

[19] M. Hagen, M. Potthast, B. Stein, and C. Bräutigam. Query segmentation revisited. In S. Srinivasan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, and R. Kumar, editors, *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, pages 97–106. ACM, 2011. https://doi.org/10.1145/1963405.1963423. URL https://doi.org/10.1145/1963405.1963423.

[20] M. Hagen, M. Potthast, A. Beyer, and B. Stein. Towards optimum query segmentation: in doubt without. In X. Chen, G. Lebanon, H. Wang, and M. J. Zaki, editors, *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 1015–1024. ACM, 2012. https://doi.org/10.1145/2396761.2398398. URL https://doi.org/10.1145/2396761.2398398.

[21] F. Hasibi, K. Balog, and S. E. Bratsberg. Entity linking in queries: Tasks and evaluation. In J. Allan, W. B. Croft, A. P. de Vries, and C. Zhai, editors, *Proceedings of the 2015 International Conference on The Theory of Information Retrieval, ICTIR 2015, Northampton, Massachusetts, USA, September 27-30, 2015*, pages 171–180. ACM, 2015. https://doi.org/10.1145/2808194.2809473. URL https://doi.org/10.1145/2808194.2809473.

[22] F. Hasibi, K. Balog, and S. E. Bratsberg. Exploiting entity linking in queries for entity retrieval. In B. Carterette, H. Fang, M. Lalmas, and J. Nie, editors, *Proceedings of the 2016 ACM on International Conference on the Theory*

of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12-6, 2016, pages 209–218. ACM, 2016. URL https://doi.org/10.1145/2970398.2970406.

[23] F. Hasibi, K. Balog, D. Garigliotti, and S. Zhang. 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*, SIGIR '17, pages 1289–1292. ACM, 2017. https://doi.org/10.1145/3077136.3084149.

[24] V. Kasturia, M. Gohsen, and M. Hagen. Query interpretations from entity-linked segmentations. In K. S. Candan, H. Liu, L. Akoglu, X. L. Dong, and J. Tang, editors, *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 449–457. ACM, 2022. https://doi.org/10.1145/3488560.3498532. URL https://doi.org/10.1145/3488560.3498532.

[25] S. MacAvaney, A. Yates, S. Feldman, D. Downey, A. Cohan, and N. Goharian. Simplified data wrangling with ir_datasets. In F. Diaz, C. Shah, T. Suel, P. Castells, R. Jones, and T. Sakai, editors, *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2429–2436. ACM, 2021. https://doi.org/10.1145/3404835.3463254. URL https://doi.org/10.1145/3404835.3463254.

[26] C. Macdonald, N. Tonellotto, S. MacAvaney, and I. Ounis. Pyterrier: Declarative experimentation in python from BM25 to dense retrieval. In G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, and H. Tong, editors, *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 4526–4533. ACM, 2021. https://doi.org/10.1145/3459637.3482013. URL https://doi.org/10.1145/3459637.3482013.

[27] J. Merker, J. Bevendorff, M. Fröbe, T. Hagen, H. Scells, M. Wiegmann, B. Stein, M. Hagen, and M. Potthast. Web-scale Retrieval Experimentation with chatnoir-pyterrier. In C. Hauff, C. Macdonal, D. Jannach, G. Kazai, F. M. Nardini, F. Pinelli, F. Silvestri, and N. Tonellotto, editors, *Advances in Information Retrieval. 47th European Conference on IR Research (ECIR 2025)*, volume 15576 of *Lecture Notes in Computer Science*, pages 96–104, Cham, Switzerland, Apr. 2025. Springer Nature. https://doi.org/10.1007/978-3-031-88720-8_17.

[28] M. Potthast, S. Günther, J. Bevendorff, J. P. Bittner, A. Bondarenko, M. Fröbe, C. Kahmann, A. Niekler, M. Völske, B. Stein, and M. Hagen. The Information Retrieval Anthology. In *Lernen. Wissen. Daten. Analysen. (LWDA 2021)*, Sept. 2021.

[29] S. Upadhyay, R. Pradeep, N. Thakur, N. Craswell, and J. Lin. Umbrela: Umbrela is the (open-source reproduction of the) bing relevance assessor. *arXiv:2406.06519*, 2024.

[30] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[31] D. Ye, Y. Qin, B. Tian, J. Fan, J. Liu, H. Liang, and J. Ma. Can llms really help query understanding in web search? A practical perspective. In M. Cha, C. Park, N. Park, C. Yang, S. B. Roy, J. Li, J. Kamps, K. Shin, B. Hooi, and L. He, editors, *Proceedings of the 34th ACM International Conference on Information and Knowledge Management, CIKM 2025, Seoul, Republic of Korea, November 10-14, 2025*, pages 5433–5438. ACM, 2025. URL `https://doi.org/10.1145/3746252.3760842`.