

# Large language models can accurately predict searcher preferences

PAUL THOMAS, Microsoft, Australia

SETH SPIELMAN, Microsoft, USA

NICK CRASWELL, Microsoft, USA

BHASKAR MITRA, Microsoft Research, Canada

Relevance labels, which indicate whether a search result is valuable to a searcher, are key to evaluating and optimising search systems. The best way to capture the true preferences of users is to ask them for their careful feedback on which results would be useful, but this approach does not scale to produce a large number of labels. Getting relevance labels at scale is usually done with third-party labellers, who judge on behalf of the user, but there is a risk of low-quality data if the labeller doesn't understand user needs. To improve quality, one standard approach is to study real users through interviews, user studies and direct feedback, find areas where labels are systematically disagreeing with users, then educate labellers about user needs through judging guidelines, training and monitoring. This paper introduces an alternate approach for improving label quality. It takes careful feedback from real users, which by definition is the highest-quality first-party gold data that can be derived, and develops an large language model prompt that agrees with that data.

We present ideas and observations from deploying language models for large-scale relevance labelling at Bing, and illustrate with data from TREC. We have found large language models can be effective, with accuracy as good as human labellers and similar capability to pick the hardest queries, best runs, and best groups. Systematic changes to the prompts make a difference in accuracy, but so too do simple paraphrases. To measure agreement with real searchers needs high-quality “gold” labels, but with these we find that models produce better labels than third-party workers, for a fraction of the cost, and these labels let us train notably better rankers.

CCS Concepts: • **Information systems** → *Test collections*; **Relevance assessment**; • **Computing methodologies** → *Natural language generation*.

Additional Key Words and Phrases: large language models, offline evaluation, labelling

## 1 LABELLING RELEVANCE

Relevance labels—annotations that say whether a result is relevant to a searcher's need—are essential for evaluating and improving information retrieval systems. Labels can come from (in decreasing order of both reliability and difficulty to obtain): (i) actual users, (ii) subject-matter experts, (iii) professional assessors (without subject-matter expertise), or (iv) crowd workers (without extensive training in the relevance assessment tasks). Label quality can be evaluated by comparing them to some *gold standard* labels [Saracevic 2008].

This paper defines gold standard labels as those from the query topic originator [Bailey et al. 2008]. The originator could be a relevance assessor who develops their own query topic, then labels the results. Even better, the originator could be a real user who did the query in situ, knows exactly what they were trying to find, and gives careful feedback on what's relevant. If each search only has one originator, then their gold labels are the ones that all other labels should be evaluated against. Given a set of first-party labels, other parties (human or machine) can at best perfectly agree, but can never “outperform” the given gold labels.

Third-party assessors may disagree with gold because they misunderstand the user's preference. If workers are systematically misunderstanding user needs—if the labels are biased—this cannot be fixed by getting more data. For

---

Authors' addresses: Paul Thomas, Microsoft, Adelaide, Australia, pathom@microsoft.com; Seth Spielman, Microsoft, Boulder, USA, sethspielman@microsoft.com; Nick Craswell, Microsoft, Seattle, USA, nickcr@microsoft.com; Bhaskar Mitra, Microsoft Research, Montreal, Canada, bmitra@microsoft.com.

example, consider a pool of workers who do not understand which queries are navigational [Broder 2002]. When a first-party user wants to navigate to a site, the third-party labels do not reward retrieval of that site. The resulting labels do not help us build a search system that performs well on navigational queries, and this can’t be solved by getting more labels from the biased worker pool. Working with human labellers, especially crowd workers, can also lead to other well-documented problems including mistakes, other biases, collusion, and adversarial or “spammy” workers [Clough et al. 2013; Inel et al. 2023; Thomas et al. 2022]. The resulting labels can be low-quality, and using them for training or making decisions will develop a retrieval system that makes similar errors.

The standard path to obtaining higher-quality labels involves multiple steps. The first is to learn about real users through interviews, user studies, direct feedback on their preferences and implicit feedback on their preferences such as clicks [Dumais et al. 2014]. Studying associated relevance labels, and looking for systematic mistakes, can indicate patterns where labellers are misunderstanding what users want. The final step is to educate labellers, by reference to guidelines or examples, to minimise future errors: for example, Google uses over 170 pages of guidelines to educate their search quality raters on what makes a good Google result [Google LLC 2022]. Asking labellers to follow guidelines should lead to improvements in their output, and that improvement can be measured against ground truth that either comes from real users (did labellers agree with real users?) or is based on our best understanding of user preferences (did labellers agree with examples carefully chosen by experts to agree with our best understanding of users?).

This paper introduces a new way of reaching very high-quality labels, that match real user preferences, by leveraging large language models (LLMs). In practice, LLM performance on any task can vary depending on the wording of the prompt [Zhang et al. 2022; Zhou et al. 2022]. Our approach is to get a small sample of feedback that perfectly reflects real user preferences, because they come from real users who did a careful job of giving feedback. We then choose a prompt for the LLM that generates labels, such that the labels have the best match with first-party ground truth.

Using machine learning for labelling raises the question of circularity, since labels are used for training and optimising retrieval systems, which may use machine learning. Machine-learned models have long been employed for relevance estimation. **These predicted or automatic relevance models are often trained on human relevance labels, and have historically been inferior in quality to the labels they were trained on.** Because they are cheap to run, the machine learned models are employed as rankers, estimating relevance at a scale that would be impractical to achieve with human assessors, and focusing on optimising the relative ordering of items, particularly in top ranks. With GPT-4 [OpenAI 2023] and similar large language models, we are now observing a new opportunity—the ability to augment relevance estimators with assessment guidelines as part of the prompt—as well as a different kind of trade-off whereby LLM labels may match first-party gold labels more closely than some third-party human labels do. GPT-4 is still too inefficient to be deployed as a real-time ranking model serving web-scale query loads, where even a tenth of a second increase in query processing latency has been shown to negatively impact searchers [Brutlag 2009; Schurman and Brutlag 2009]. This creates a new opportunity to employ these automatic relevance assessments from GPT-4 for training and evaluating more efficient ranking models, which may be seen as a form of knowledge distillation [Hinton et al. 2015].

For other annotation tasks there is evidence that LLMs can be comparable to crowd workers, using standard metrics such as agreement or correlation [Alizadeh et al. 2023; Gilardi et al. 2023; Törnberg 2023]. However, we argue it is more interesting to compare labels to a relatively small set of first-party ground truth, from real searchers. We can then ask how well different labellers do—human or LLM—in generating labels that match real user preferences. Our study shows that *LLM labellers can do better on this task than several populations of human labellers*. The worst are the crowd labellers, who are least diligent and least knowledgeable about user preferences. Better are human raters who are more knowledgeable and diligent, as demonstrated by better agreement with first-party ground truth (gold). LLMs perform

Large language models can accurately predict searcher preferences

better on this metric than any population of human labellers that we study. Our results demonstrate the potential for LLMs as a tool for obtaining high-quality relevance labels that match what users think.

## 2 EXPERIMENTS: TREC-ROBUST

To illustrate these ideas, we have experimented with queries, documents, and labels from TREC-Robust 2004 [Voorhees 2004]. Our main question was whether LLMs could replicate the original TREC labels, assigned by expert human assessors.

### 2.1 Machinery and data

TREC-Robust includes 250 topics (each with one canonical query, so “query” and “topic” are synonymous in what follows)<sup>1</sup>. We took queries from the TREC title field; description and narrative were also included in some prompts, as discussed below.

Official labels were taken from the TREC-Robust qrel file. These labels were assigned by trained assessors, who had also provided the queries and topic descriptions, so although these are not “real” in situ search scenarios with a real product, they fit our definition of gold Bailey et al. [2008]: the person who labelled each document is the single best judge of what the query and topic mean, and what sort of document was responsive. If and when a third-party labeller (human or LLM) deviates from gold, it is considered an error with respect to the first-party data.

The original qrels files had 1031 “highly relevant” labels, 16 381 “relevant”, and 293 998 “not relevant”. In the first experiments below we used a stratified random sample of 1000 qrels for each label, 3000 labelled topic : document pairs in total. In later experiments we used all documents returned in Robust 2004 runs at ranks 1–100, where those documents were judged in TREC.

The experiments here used an in-house version of GPT-4 [OpenAI 2023], running on the Azure service. Temperature was set at zero, so the model would select the single most likely output; other parameters were top  $p = 1$ , frequency penalty 0.5, presence penalty 0, with no stopwords.

### 2.2 Prompting

Having carefully selected our gold data, we consider a number of prompt template variants (determining LLM inputs) which is generally a cheap and fast way to improve quality [Karpathy 2023].

Figure 1 gives an overall schema for the prompts. *Italicised* words are placeholders, which were filled differently for each topic and document, or otherwise varied to match the rest of the prompt. **Shaded** text is optional and was included in some prompt variants.

The prompt has four parts. The first part gives task instructions. These are closely based on instructions given to TREC assessors with two changes: First, the TREC instructions included material on consistency in labels, which is not relevant to an LLM case so was dropped here. Second, the phrase “you are a search engine quality rater...” replaces some of the TREC text which discusses the assessors’ past experience developing TREC tracks. The phrase “search engine quality rater” is used by Google in its labelling efforts, and the phrase is widely used on the web, making it a useful shorthand.

---

<sup>1</sup>One query had no relevant documents. It is included in our analysis but will always score zero, on any metric, using the official labels.

The second part of the prompt gives the query/document pair to be labelled: we include the query that the “searcher” issued; in some configurations we include a more detailed version of their intent from the TREC narrative field; and we give the text of the document itself.

The third part of the prompt restates the task, including the instruction to “split this problem into steps” by explicitly considering the searcher’s intent as well as the document. This follows observations by Wei et al. [2022] and Kojima et al. [2022] that “chain of thought” or “step by step” prompts can produce more reliable results (something we have also observed, informally, in other work). In some variants, we expanded this to explicitly ask for scores for two aspects—topicality and trust—as well as an overall score. In some variants, we also ask the model to simulate several human judges (here five) and give scores from each.

The final part of the prompt specifies an output format and includes a snippet of JSON to encourage correct syntax.

This is a “zero-shot” prompt, in that it does not include any examples of the task. Liang et al. [2022] report remarkably mixed results across tasks and models, so it is certainly possible that we could improve with one or more examples; it is also possible we could see some regression. The length of TREC documents means it is hard to include even one entire example, let alone more, and we leave experimentation with one- or few-shot prompts as future work.

Note that we do not claim that this is the best prompt, or the best prompt format; indeed, in Section 4.4 we will see that even minor paraphrases can make a material difference. Our interest here is in the range of results we see with a reasonable prompt (as opposed to the minimal prompts of Faggioli et al. [2023] or Liang et al. [2022]), in the practical impact of disagreements, and in which features of a prompt seem to help or hinder LLM accuracy.

## 2.3 Variations

We varied the prompt in four ways:

**Describing the role** The simplest version of our instructions asks for a score for a query and a web page. Web page quality is a complex notion, but search providers frequently publish hints of what they are looking for. In particular, Google’s labelling guidelines use the phrase “search quality rater” [Google LLC 2022]. Some prompts therefore include the phrase “you are a search quality rater evaluating the relevance of web pages”, as a shorthand way to reference both the guidelines (which are generally useful) and surrounding discussion.

**Varying topical description** Queries alone are an impoverished representation of an information need, but TREC topics have additional text describing what the query means (description) and which documents should be considered responsive (narrative). For example, for the query *hubble telescope achievements*, the description restates that the query is about achievements of the space telescope since its launch in 1991, and the narrative clarifies that this is about scientific achievement so results that only talk about shortcomings and repairs would not be considered relevant. In some prompts, we include this text as the “description” and “narrative” fields.

**Varying aspects** A straightforward approach, following the TREC guidelines, would be to ask for an overall label for each query : document pair. In past work with human labelling, we have found it more useful to spell out several aspects, and ask for ratings against these, before asking for an overall label. These extra questions have been useful to help anchor judge assessments, without constraining the final label (i.e. the overall label need not be a simple average of the per-aspect labels). Similarly, with large language models there has been demonstrated success with splitting problems into steps with prompts such as “think step by step” [Kojima et al. 2022].

Large language models can accurately predict searcher preferences

role	<p><b>You are a search quality rater evaluating the relevance of web pages.</b> Given a query and a web page, you must provide a score on an integer scale of 0 to 2 with the following meanings:</p> <p>2 = highly relevant, very helpful for this query 1 = relevant, may be partly helpful but might contain other irrelevant content 0 = not relevant, should never be shown for this query</p> <p>Assume that you are writing a report on the subject of the topic. If you would use any of the information contained in the web page in such a report, mark it 1. If the web page is primarily about the topic, or contains vital information about the topic, mark it 2. Otherwise, mark it 0.</p>
description, narrative	<p><b>Query</b> A person has typed [<i>query</i>] into a search engine. <b>They were looking for:</b> <i>description narrative</i></p> <p><b>Result</b> Consider the following web page.</p> <p>—BEGIN WEB PAGE CONTENT—</p> <p><i>page text</i></p> <p>—END WEB PAGE CONTENT—</p> <p><b>Instructions</b> Split this problem into steps:</p> <p>Consider the underlying intent of the search.</p>
aspects	<p><b>Measure how well the content matches a likely intent of the query (M).</b></p>
aspects	<p><b>Measure how trustworthy the web page is (T).</b></p> <p><i>Consider the aspects above and the relative importance of each, and decide on a final score (O).</i></p>
multiple	<p><b>We asked five search engine raters to evaluate the relevance of the web page for the query. Each rater used their own independent judgement.</b></p> <p><i>Produce a JSON array of scores without providing any reasoning. Example: [{"M": 2, "T": 1, "O": 1}, {"M": 1 . . .</i></p> <p><b>Results</b> [{"</p>

Fig. 1. General form of the prompts used in our TREC Robust experiments. *Italicised* words are placeholders, filled with appropriate values. **Shaded** text is optional, included in some prompt variants.

Inspired by these ideas, in some variants we explicitly ask for labels over aspects of “relevance” as well as for an overall label. For TREC Robust, we ask for labels for topicality (“how well the content matches a likely intent”—note that this captures likely intents that aren’t captured elsewhere) and for trustworthiness (“how trustworthy the page is”). There are no further definitions of either aspect.

**Varying number of “judges”** People naturally vary in their labels, and aggregating several labels for each result can reduce noise and increase sensitivity due to law of large numbers. In some prompts we ask the model to simulate several judges, generating the output of five simulated judges from one LLM call. Since the outputs are generated in sequence they are not really independent labellers, but we previously found it useful to generate and aggregate multiple labels in this way, so we include it as a prompt variant here.

### 3 EVALUATING THE LABELS, EVALUATING THE LABELLERS

How are we to choose between labels, or rather between labelling processes? The main criterion is validity, in particular that labels from any new source should agree with gold labels [Faggioli et al. 2023]. We can measure this in two ways: by looking at the labels themselves or by looking at preferences between documents. Additionally, labels are typically aggregated to derive query-level or system-level scores, and we may care whether machine labels would lead to similar conclusions at these aggregated levels.

Further criteria include cost, in both dollars and time; throughput; and how easily we can measure new types of result, such as results in different languages or different media types.

#### 3.1 Document labels

The simplest way to evaluate a machine labelling process is to ask: does it produce the same labels as would human labellers? Evidently, if the labels are the same for any document, then the machine process can be directly substituted without any quality concerns.

We can summarise differences between the machine and human labels with a confusion matrix. The labels are on an ordinal scale (not an interval scale), but if we assign scores 0 and 1 to the two levels then we can further compute the mean difference between the human and machine labels. In what follows we report accuracy with the mean absolute error (MAE), where 0 means the two sources always agree on labels and 1 means they are maximally different.

In an earlier study, Faggioli et al. [2023] report Cohen’s  $\kappa$  between TREC assessors and GPT-3.5 and YouChat LLMs, and we report  $\kappa$  here as well.  $\kappa$  ranges from 1 (complete agreement) through 0 (agreement only by chance) to  $-1$  (complete disagreement). For direct comparison with Faggioli et al. we report  $\kappa$  over binarised labels, where partly- and highly-relevant are conflated.

#### 3.2 Document preference

Minimising document-level MAE gives us scores which are calibrated across queries, and interpretable for debugging and development. Ranking, however, can use preferences between documents rather than calibrated scores; this is also sufficient for many learning-to-rank algorithms [Liu 2009]. On this view it is the relative ordering of any two documents that is important, and we can measure this with pairwise accuracy or AUC: the chance that, given any two documents with a human preference, the model’s preference is the same. A score of 1 means the model’s preferences are always the same as the human’s, a score of 0 means they always disagree, and a score of 0.5 is chance alone.

Large language models can accurately predict searcher preferences

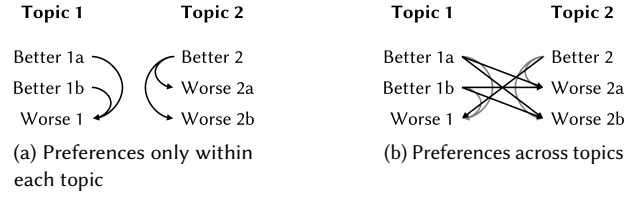


Fig. 2. Options for document preference. If we form preferences only within each topic, there is no constraint on how, for example, “better 1a” is scored relative to “worse 2a”: labels can vary per topic. If we form preferences across topics, we add the constraint that “better 1a” should score higher than “worse 2a”, so labels are consistent. We also generate many more pairs.

(Another consideration is that two scoring schemes may differ in scale and location: for example, one source may give scores 1–5 while another gives 1–10 or 0–99. MAE in this case is misleading, even if there is a completely consistent mapping from one source to another. Pairwise preferences are robust to this sort of difference.)

There are two ways to form pairs of documents (Figure 2). If we choose pairs of documents only from the same topic, we can use a topic-dependent labelling scale: the worse document for one topic might still be better than the better document from another, for example if one topic is especially hard. The set of pairs will also be smaller. Choosing pairs of documents from all topics, that is, from all documents ever labelled, enforces a query-independent scale as the “better” document from one query should score higher than the “worse” document from any other. The set of pairs formed this way will also be bigger. In our evaluation, we choose the second approach; in other circumstances, the flexibility of per-topic ordering might be preferable.

### 3.3 Query ordering

Our primary interest is in generating (and evaluating) labels for documents. However, past work has shown that errors in document labels can be washed out when labels are aggregated to query-level or system-level scores [Bailey et al. 2008]. It is certainly possible that differences in labels are not relevant to query- or system-level evaluations.

In consideration of this we can also order result lists (SERPs) by some metric (e.g. RBP or MAP), according to the labels produced by humans and with regard to some fixed search engine; order the same result lists, on the same metric, according to the labels produced by a model; and ask how similar the two orderings are.

With this query-level analysis we are likely to be looking for queries which do badly (i.e. where a system scores close to zero), so here we measure correlation with rank-biased overlap (RBO) [Webber et al. 2010] after sorting the queries from lowest to highest scores. This means that (dis)agreements about which queries score worst—which queries we want to investigate—count for more than (dis)agreements about those queries that score well.

In our case, since the two rankings are permutations, there is a well-defined lower bound<sup>2</sup> for RBO:

$$(1 - \phi) \sum_{d=\lfloor N/2 \rfloor + 1}^N \left( \phi^{d-1} (2d - k) / d \right)$$

with  $N$  queries and a discount parameter  $\phi$ . For ease of interpretation we use this minimum to normalise RBO scores into the range 0 to 1, so 0 is an exactly opposite ranking and 1 is an identical ranking. We use set  $\phi = 0.9$ , corresponding to an experimenter looking (on average) at the first ten queries.

<sup>2</sup>Alistair Moffat, personal communication, July 2023.

### 3.4 System ordering

The primary use of query:document scores is of course to score a whole system, first by accumulating document scores to query scores then by accumulating query scores to system scores. To see the effect of disagreements between our human and LLM judges, we report RBO over those systems which ran the same queries. Again, since there are a fixed set of systems, we can calculate the minimum RBO score and normalise. An experimenter might look seriously at the top three or four systems, so we set  $\phi = 0.7$ .

### 3.5 Ground-truth preferences between results

An alternative view is that, since human-assigned labels may themselves be biased or noisy, labels should instead accurately predict real searcher preferences.

Evaluating machine labels by their agreement with human labels is useful, because in many situations we can use a large corpus of existing labels. However, it does not speak to the validity of the labels: that is, whether the labels (or a metric derived from the labels) reflects some true searcher experience. If machine labels agree with human labels to (e.g.) 80%, then the 20% disagreement might be a fault with the machine, or poor-quality labels from the humans, or some combination. We expand on this idea in Section 5.

### 3.6 Other criteria

Besides the above, we can imagine other criteria for choosing a labelling process. These might include cost per label; time, per label or end-to-end; reliability; scalability; difficulty of adapting to new languages, markets, or evaluations; and ease of debugging the labelling process. We do not address these criteria here, but in our experience labelling with LLMs is superior to labelling by crowd workers on all these grounds and is superior to labelling by experts (employees or specially-qualified workers) on all grounds except debuggability.

## 4 RESULTS

After running the prompt, the generated label was converted to a score in  $[0, 2]$ . Where we generated multiple labels, the final score is simply the mean. In keeping with the TREC guidelines, if we prompted for aspects we still considered only the overall label. If the model generated unparseable output, we dropped the result entirely: this happened in 90 out of 96 000 cases.

TREC-Robust included two sets of topics. Topics up to 650 came from earlier editions of TREC, and had only binary relevance judgements (“relevant” or “non-relevant”; 1 or 0). Topics 651–700 were developed for the track, and have three-level judgements (adding “highly relevant”, 2). Our prompts generated a scores from 0 to 2 for all documents, in line with instructions to TREC-Robust assessors for the new topics. Since comparisons are difficult between a three- and a two-level scale, we follow TREC and Faggioli et al. [2023] by considering “relevant” and “highly relevant” together, i.e. by binarising the scores in all cases.

We evaluate the quality of these *labels* (not the documents) in three ways: by comparing the model’s labels for each document to the labels from TREC assessors, by comparing the aggregated scores for each query, and by comparing the overall system rankings that result.



Large language models can accurately predict searcher preferences

TREC assessor		Model	
		0	1 or 2
	0	866	95
	1 or 2	405	1585

Table 1. Results from the best-performing prompt of Figure 1—i.e. with descriptions, narrative, and aspects, prompt “-DNA-”—over a stratified sample of the TREC Robust data. Overall, the LLM is more likely to say “not relevant” than were TREC assessors; an LLM assessment of “relevant” or “highly relevant” is reliable. Some qrels are missing due to unparsable LLM output, a rate of 1.6%.

#### 4.1 Comparing scores

Similar to Faggioli et al. [2023], we compare these model-generated scores to scores from the TREC assessors. As an example, Table 1 gives a confusion matrix for one prompt and all 3000 query:document pairs. (There are 32 such matrices, one for each set of prompt features or equivalently one for each row of Table 2.) We can see that in this case, the LLM is more likely to say “not relevant” than were TREC assessors (44% vs 33%), and is correspondingly inaccurate (68% agreement with TREC, when the LLM says “not relevant”). An LLM assessment of “relevant” or “highly relevant” however, is reliable (94% agreement).

Table 2 summarises the models’ agreement with human judges, over the 3000 query:document pairs, as we manipulate the prompt as above: there is one row for each prompt, identified by which optional features are included. For example, the row labelled “-N-M” corresponds to the prompt with **n**arrative and **m**ultiple judges, but not **r**ole statement, **d**escription, or **a**spects. For each prompt, we report the three document-level, one query-level, and one system-level metrics described above, plus a 95% confidence interval based on 20 bootstraps over documents. The best-performing prompt for each metric is labelled with a ★, and these are significantly better than any other ( $t$  test,  $p < 0.05$ ).

Performance is highly variable as we change the features—that is, the quality of the labelling depends a great deal on the prompt structure or template. For example, Cohen’s  $\kappa$  varies from as low as 0.20 (prompt “R---M”) to 0.64 (prompt “-DNA-”). We need to be accordingly careful interpreting any claim based on a single prompt, especially where that prompt has not been tuned against some existing labels; we also observe this in the variable performance reported in Liang et al. [2022], for example.

The performance here ( $\kappa$  0.20 to 0.62) compares favourably to that seen by Damessie et al. [2017], who re-judged 120 documents from TREC-Robust and saw  $\kappa$  of 0.24 to 0.52 for crowd workers, and  $\kappa$  of 0.58 for workers in a controlled lab. In particular, 6/32 prompts here to better than 0.58 and only 3/32 do worse than 0.24. Our agreement also compares favourably to reports from Cormack et al. [1998], who labelled TREC ad-hoc documents a second time, using a second group of assessors. From their data we can compute Cohen’s  $\kappa = 0.52$  between two groups of trained human assessors.

On other data sets, Castillo et al. [2006] report  $\kappa = 0.56$  labelling web pages for spam; Hersch et al. [1994] report  $\kappa = 0.41$  on relevance in the OHSUMED collection; Agarwal et al. [2019] saw  $\kappa = 0.44$  for news sentiment; and Scholer et al. [2013] reported that assessors seeing a document for a second time only agreed with their first label 52% of the time. Faggioli et al. [2023] reported  $\kappa$  from 0.26 to 0.40 on binarised labels from TREC-8 and TREC Deep Learning. Faggioli et al. used another LLM but with relatively simple prompt, reinforcing LLMs’ sensitivity to their prompt.

On this metric, at least, we can conclude that with minimal iterations LLMs are already at human quality for this collection and for some prompts. In Section 5 we will see that, in a common setting, LLMs can perform substantially better than third-party judges.

Prompt features					Document scores MAE	Document scores $\kappa$	Document preference AUC
—	—	—	—	—	0.34± 0.01	0.38± 0.02	0.73± 0.01
R	—	—	—	—	0.38± 0.02	0.32± 0.02	0.71± 0.01
—	D	—	—	—	0.36± 0.02	0.35± 0.03	0.72± 0.01
—	—	N	—	—	0.35± 0.02	0.37± 0.03	0.73± 0.01
—	—	—	A	—	0.19± 0.02	0.60± 0.03	0.82± 0.02
—	—	—	—	M	0.46± 0.02	0.22± 0.02	0.65± 0.01
R	D	—	—	—	0.40± 0.02	0.30± 0.03	0.69± 0.01
R	—	N	—	—	0.38± 0.02	0.33± 0.02	0.71± 0.01
R	—	—	A	—	0.21± 0.02	0.56± 0.03	0.81± 0.02
R	—	—	—	M	0.49± 0.02	0.20± 0.02	0.64± 0.01
—	D	N	—	—	0.35± 0.02	0.37± 0.02	0.74± 0.01
—	D	—	A	—	0.19± 0.01	0.59± 0.03	0.83± 0.01
—	D	—	—	M	0.45± 0.01	0.24± 0.02	0.66± 0.01
—	—	N	A	—	0.18± 0.01	0.62± 0.02	0.84± 0.01
—	—	N	—	M	0.41± 0.02	0.29± 0.02	0.69± 0.01
—	—	—	A	M	0.31± 0.02	0.42± 0.04	0.80± 0.02
R	D	N	—	—	0.37± 0.02	0.34± 0.03	0.72± 0.02
R	D	—	A	—	0.22± 0.01	0.53± 0.03	0.82± 0.01
R	D	—	—	M	0.46± 0.02	0.23± 0.02	0.66± 0.01
R	—	N	A	—	0.20± 0.01	0.59± 0.03	0.83± 0.01
R	—	N	—	M	0.42± 0.02	0.28± 0.02	0.69± 0.01
R	—	—	A	M	0.38± 0.02	0.32± 0.02	0.78± 0.01
—	D	N	A	—	0.17± 0.01	0.64± 0.02★	0.85± 0.01★
—	D	N	—	M	0.40± 0.02	0.31± 0.02	0.70± 0.01
—	D	—	A	M	0.31± 0.01	0.42± 0.02	0.80± 0.01
—	—	N	A	M	0.27± 0.02	0.49± 0.03	0.82± 0.02
R	D	N	A	—	0.19± 0.01	0.61± 0.02	0.84± 0.01
R	D	N	—	M	0.41± 0.01	0.29± 0.02	0.69± 0.01
R	D	—	A	M	0.37± 0.02	0.34± 0.02	0.80± 0.01
R	—	N	A	M	0.33± 0.01	0.39± 0.02	0.80± 0.01
—	D	N	A	M	0.26± 0.01	0.50± 0.02	0.82± 0.01
R	D	N	A	M	0.16± 0.02★	0.51± 0.06	0.77± 0.03

Table 2. Performance of the variant prompts of Figure 1, compared to human labels on a stratified sample of the TREC Robust data. R = include role, D = include description, N = include narrative, A = include aspects, M = include multiple “judges”. Accuracy of document scores is measured with mean absolute error and with Cohen’s  $\kappa$  against TREC assessors on binary labels. Accuracy of document preference is measured with AUC. Accuracy of query and system ordering is measured with RBO, normalised to the range 0–1. Uncertainty is reported as a 95% confidence interval based on 20 bootstraps. ★ marks the best prompt in each case (significantly better than the next-best performer, one-sided  $t$  test,  $p < 0.05$ ).

Feature	Change in $\kappa$
Role, R	−0.04
Description, D	+0.01
Narrative, N	+0.06
Aspects, A	+0.21
Multiple “judges”, M	−0.13

Table 3. Performance impact of the optional prompt features in Figure 1, measured using  $\kappa$  against TREC assessors. All changes are statistically significant and effects are  $\pm 0.005$  at a 95% CI.

## 4.2 Effect of prompt features

Table 2 gives results for 32 prompt templates, made from turning five features on or off. To try to summarise the effect of each feature individually, Table 3 reports the effect of each feature on  $\kappa$ —that is, the effect of including a prompt feature independent of any other features being on or off.

Contrary to our expectations, there is a statistically significant *negative* effect due to role (R) and multiple “judges” (M):  $\kappa$  decreases by an average 0.04 and 0.13 respectively. Adding description (D) gives an insubstantial boost (only 0.01 points of  $\kappa$ ). Adding a narrative (N) leads to a boost of 0.04; this is modest, but perhaps the background knowledge of LLMs (especially on well-used public data like this) is enough that the narrative adds little information beyond the query terms.

Aspects (A) give a substantial improvement in  $\kappa$  against TREC assessors, +0.21. Topicality and trustworthiness are the two aspects we used here, but of course that are not the only aspects that might matter, and we do not claim they are the best selection; in Bing we use several aspects, and measure the LLM’s performance on all of these with good results. In this case it seems likely, in fact, that it is the step-by-step nature of labelling with aspects that gives rise to these improvements rather than the particulars of the aspects themselves.

Note that this presents features in isolation, when in fact any prompt could have zero, one, two, three, four, or all five of these features at once and the effects are not necessarily additive. The best-performing prompt in Table 2 is, however, of the form “-DNA-” which is expected from this analysis.

## 4.3 Effect of prompt length

Using an LLM to compare texts, Wang et al. [2023] saw an effect of prompt length—the longer the text, the more positive the LLM’s assessment. We checked for similar effects in our data by modelling the LLM’s *signed error* as a response to prompt length. This controls for any effect of length on true relevance; if longer documents are just more (or less) likely to be relevant, then the LLM should not be penalised for reflecting this. Replicating Wang et al.’s effect would require a positive effect: that is, errors should get more positive (the LLM should overestimate more, or be more optimistic) as prompts got longer.

Controlling for prompt features, we saw no substantial correlation between prompt length and signed error. Effects varied according to prompt features, with modelled score shifting between  $-9 \times 10^{-6}$  and  $1 \times 10^{-5}$  per character of prompt. This corresponds to only a shift in score of -0.05 to 0.06 at the median prompt length, which (although statistically significant) is of no practical significance given the MAEs of Table 2.

#### 4.4 Effect of paraphrasing prompts

We have seen that LLM performance varies considerably as the prompt is varied, even when the task and the input data are fixed. This raises a question: how sensitive is the LLM not just to coarse prompt features, such as asking for aspects, but to quirks of phrasing? In other words, if we rephrased “assume that you are writing a report” to “pretend you are collecting information for a report”, or to “you are collecting reading material before writing a report”, would the labels change? If so, then our LLM is highly sensitive to such apparently trivial considerations. That would mean that, first, the results above are only representative of a wide range of possible performance; and second, any serious attempt to use LLMs at scale needs to explore a large and unstructured prompt space.

To test this, we took the “-DNA-” prompt—the best above—and generated 42 paraphrases by rewriting the text “Given a query and a web page ... Otherwise, mark it 0” and by rewriting the text “Split this problem into steps: ... Produce a JSON array of scores without providing any reasoning”. Figure 3 gives some examples.

Figure 4 shows the resulting spread of label quality, measured again as Cohen’s  $\kappa$  against the labels from TREC assessors and across our stratified sample of 3000 documents. Each paraphrase is represented by one dark line, showing the mean  $\kappa$  and a 95% confidence interval derived from 20 bootstraps over documents. There is a large range, from mean  $\kappa = 0.50$  (moderate agreement) to mean  $\kappa = 0.72$  (substantial agreement, and better than the reference values cited above [Agarwal et al. 2019; Castillo et al. 2006; Cormack et al. 1998; Faggioli et al. 2023; Hersh et al. 1994]). The empirical 95% confidence interval, over all bootstraps and all paraphrases, is 0.50–0.71 (plotted at the left-hand edge of Figure 4).

This is a wide range from a single prompt design, and from Figure 3 it is not at all apparent which versions would score higher or why. The outsized effect of simple paraphrases has been observed in other domains as well [Zhang et al. 2022; Zhou et al. 2022].

This leads to two observations. First, the measured performance of any prompt—including those in Table 2—should be taken as a single sample from a wider range of potential performance. Small tweaks to the wording could result in noticeably different performance, even without any changes to the prompts’ overall design. Second, it is prudent to fix an overall design, and then explore rephrasing and other options. Because it is not clear what leads to better or worse performance, taking paraphrases is a reasonable approach, but we note work by Pryzant et al. [2023], Yang et al. [2023], Zhou et al. [2022], and others that suggests alternatives for fine-tuning prompts.

#### 4.5 Effect of document selection

Given the different performance of the different prompts, and indeed the different paraphrases, it is tempting to choose the best-performing variant and commit to using it for future labelling. This of course carries a risk: performance on these topics and documents might not predict performance on other, unseen, topics and documents. The conventional guard against this is a train:test split. Here, we can interpret “training” as the choice of prompt, and we used repeated splits to understand the risk of choosing the best variant. For each of 1000 iterations, we randomly split our 3000 TREC and LLM labels into two sets of 1500 documents. We measured  $\kappa$  for each prompt (or paraphrase) over the first 1500, noted the best performer (highest  $\kappa$ ), and measured again on the second 1500.

The results were consistent. When scoring prompts (Table 2), in all 1000 iterations the best-performing prompt on the first split also beat the baseline “-----” on the second split. That means that, starting from the baseline prompt, if we chose an alternative because it was the best improvement on one set of documents, we can be almost certain that prompt would still be an improvement on another set. In 829/1000 first splits, the best-performing variant was -DNA-,

## Large language models can accurately predict searcher preferences

### Original

$\kappa = 0.64$

Given a query and a web page, you must provide a score on an integer scale of 0 to 2 with the following meanings:

2 = highly relevant, very helpful for this query

1 = relevant, may be partly helpful but might contain other irrelevant content

0 = not relevant, should never be shown for this query

Assume that you are writing a report on the subject of the topic. If you would use any of the information contained in the web page in such a report, mark it 1. If the web page is primarily about the topic, or contains vital information about the topic, mark it 2. Otherwise, mark it 0.

...

Split this problem into steps:

Consider the underlying intent of the search.

Measure how well the content matches a likely intent of the query (M).

Measure how trustworthy the web page is (T).

Consider the aspects above and the relative importance of each, and decide on a final score (O).

Produce a JSON dictionary of scores without providing any reasoning.

### Paraphrase 1

$\kappa = 0.72$

Rate each web page for how well it matches the query, using these numbers: 0 = no match, 1 = some match, 2 = great match. Think of writing a report on the query topic. A web page gets 2 if it is mainly about the topic or has important information for the report. A web page gets 1 if it has some information for the report, but also other stuff. A web page gets 0 if it has nothing to do with the topic or the report.

...

To score this problem, follow these steps:

- Think about what the search query is trying to achieve.

- Assign a score from 0 to 2 for how well the content addresses the query's goal (M). Higher scores mean better matches.

- Assign a score from 0 to 2 for how reliable the web page is (T). Higher scores mean more trustworthiness.

- Combine the scores for M and T, and give more weight to the more important aspect. Assign a final score from 0 to 2 (O). Higher scores mean better overall quality.

- Write a JSON dictionary with the keys M, T, and O, and their corresponding scores. Do not explain your scores.

### Paraphrase 2

$\kappa = 0.50$

To rate a web page for a query, use 0, 1, or 2. Use 0 if the page has nothing to do with the query. Use 1 if the page has some useful information, but also other stuff. Use 2 if the page is mainly about the query or has important information.

...

For this problem, you need to do the following:

- Think about what the searcher wants to find out.

- Rate how well the content answers the query, from 0 (poor) to 2 (excellent) (M).

- Rate how reliable the web page is, from 0 (low) to 2 (high) (T).

- Based on the ratings and their importance, give a final score from 0 to 2 (O).

- Write a JSON dictionary of the scores without explaining them.

Fig. 3. Examples of paraphrased prompts, based on prompt format “-DNA-” (description, narrative, and aspects). Each paraphrase was run with each of our 3000 sampled documents, to gauge the model's sensitivity to changes in the prompt text.

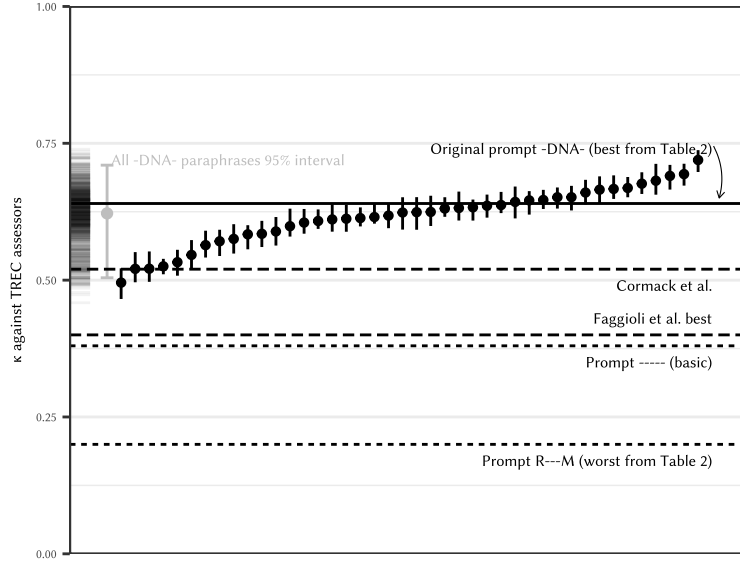


Fig. 4. Variation in Cohen’s  $\kappa$  between LLM labels and human labels, over a stratified sample of 3000 documents from TREC-Robust. Small changes in the wording of the prompt, while keeping the structure the same, lead to substantial changes in  $\kappa$ . Each vertical line is one paraphrased prompt, with empirical 95% CI from 20 bootstraps over documents. Grey interval at left is the empirical 95% CI over all bootstraps and paraphrases.

which is again consistent with the above but also suggests the choice is reliable. (The next best performer was --NA-, 139 times out of 1000; of course in practice these two prompts are very similar.)

Looking at the 42 paraphrases of Figure 4, in 989/1000 iterations the best-performing paraphrase on the first 1500 documents still beat the initial -DNA- prompt on the second 1500. The best-performing paraphrase was again consistent: variant #13 had the highest  $\kappa$  on the first split in 838/1000 iterations. This is marginally less consistent than the choice of overall prompt design.

These observations suggest that while performance is variable, there is little chance of regret. That is, if we start with a baseline prompt and generate variants—e.g., by adding features or by paraphrasing—and choose to switch to the best variant, that is a safe choice. If we choose the best variant on some set of documents, performance on unseen documents will almost never turn out to be worse than the baseline.

#### 4.6 Measuring query difficulty and run effectiveness

Document labels themselves are not the goal of most evaluations. Instead, we typically map these labels to numeric values (0 and 1 for binary labels) and then use a metric such as average precision to aggregate to scores for each query and run. The scores for queries let us investigate instances where we do badly, meaning where there is scope for improvement; the scores for runs let us choose which combination of algorithms and parameters performs the best overall.

Accordingly, another way to judge a labelling scheme is by whether (under some metric) it gives the same ranking of queries or runs. If we swapped labelling schemes, would we still identify the same queries as hard? Would we still identify the same runs as top performers?

Large language models can accurately predict searcher preferences

	Hardest queries	Best runs	Best groups
	RBO, $\phi = 0.9$	RBO, $\phi = 0.7$	RBO, $\phi = 0.7$
P@10	0.40	0.79	0.97
RBP@100, $\phi = 0.6$	0.42	0.63	0.91
MAP@100	0.48	0.50	0.58
(Random permutation)	0.04	0.03	0.21

Table 4. Consistency of rankings on LLM labels compared to human labels, replicating all qrels in TREC-Robust to a depth of 100. Queries, runs, and groups were scored with each of three metrics, based on each of two sets of labels. Higher numbers mean the rankings based on LLM labels are more like those based on human labels. We report normalised RBO, ranging from zero (LLMs and humans put queries/runs/groups in opposite order) to one (LLMs and humans give scores putting queries/runs/groups in the same order).

In Table 4 we report the consistency of query and run rankings as we switch from human-assigned to LLM-assigned labels. In each case we score all the queries with one metric—e.g. P@10—based on TREC’s human labels, and score them again based on our LLM labels. (We collected additional labels so that every document retrieved to depth 100, in every run, was labelled with prompt -DNA- *except* those which were never labelled at TREC. For consistency with TREC, we assume these unlabelled documents are not relevant.) This gives two rankings of queries. The consistency between these rankings is measured with RBO, normalised so that a score of 0 represents an inverted order and a score of 1 represents an identical ordering. We assume an experimenter would be willing to look at the worst ten queries, so set  $\phi = 0.9$ . To help interpret the figures we also report the RBO scores for random permutations, i.e. the consistency between the TREC ordering and a random re-ordering of the same queries.

The exercise is repeated for all 110 runs, assuming we want to find the best three or four runs ( $\phi = 0.7$ ). Since runs from the same group are likely very similar, we also repeat the exercise for the best run for each group—this simulates choosing the best approach (or perhaps vendor), rather than the best parameter settings. Again we assume we want to find the best three or four for further examination.

The consistency of rankings, in all three cases, depends on the metric being used: ordering by MAP is more consistent for queries, and ordering by average P@10 is more consistent for runs and groups. Group-level rankings are more consistent than runs or queries, no matter the metric. It is harder to be consistent when ranking 250 queries than when ranking 110 runs or 14 groups, and small perturbations make a larger difference in ranking since many queries have similar scores. Nonetheless we see that for any problem and choice of metric, labels from LLMs lead to overall rankings which are at least similar to those from human labels, and our imagined experimenters would make similar choices. For example, under all metrics the top three runs are the same; the top five groups are consistent under P@10, the top three under RBP@100, and three of the top four under MAP@100. The worst-performing query is the same under TREC or LLM labels for P@10 and RBP@100, and two of the top three are the same under MAP@100.

Of course perfect agreement is unlikely even with humans labelling. By way of comparison, Voorhees [1998] reports  $\tau = 0.94$  across runs, using labels from different assessors. This is on a different data set, with correspondingly different judgements (and only 33 runs), but give a rough upper bound for how consistent runs could ever be. Faggioli et al. [2023] demonstrate  $\tau$  from 0.76 to 0.86 on TREC Deep Learning data, again under slightly different circumstances (notably, shorter documents and fewer runs). We see  $\tau$  from 0.77 (MAP@100) to 0.86 (P@10) for our 110 runs with full documents. Given the  $\kappa$  and AUC figures in Table 2, this is at least promising and plausibly as good as most human labellers.

	Relative accuracy	Latency	Relative throughput	Relative cost
Employees	+24%	hours to days	$\times 1/100$	$\times 8$
Best crowd	+19%	hours to days	$\times 1/15$	$\times 5$
Typical crowd	—	hours	$\times 1$	$\times 1$
LLM (GPT-4)	+28%	minutes to hours	$\times 10$	$\times 1/20$

Table 5. Labelling schemes compared. “Crowd” are crowd workers via our in-house platform, “LLM” is the best-performing prompt from private experiments. “Latency” is the time to the first usable labels, “cost” is the dollar cost alone. These figures give an overall comparison, but please note that they depend on our particular computing resources, crowd contracts, assessor training, and other details.

#### 4.7 Observations

We see somewhat better results than those reported by Faggioli et al. [2023], particularly in agreement on the raw labels ( $\kappa$ ). There are at least two factors at work. First, we are using a more capable model (GPT-4 with local modifications, compared to stock GPT-3.5); and second, our prompts are based on our experiences in Bing, and relatively long, whereas those of Faggioli et al. are simpler. Even small wording changes can make a difference (Figure 4), and selecting prompt features makes a bigger difference still (Table 2). Again, this demonstrates that time spent on this configuration—which is comparable to time spent on instruments and instructions for crowd or in-house workers—can pay dividends.

These results show that LLMs are competent at labelling—at the minimum, with GPT-4 and in the TREC-Robust setting. The labels are as close to those from humans as we could expect, given the disagreement between people to begin with, and we can reasonably consistently identify the hardest queries, best runs, and best groups.

We now turn to LLM labelling at scale, in the context of a running search engine, where LLMs have proved not just more efficient but more accurate than the status quo.

### 5 LLM LABELLING IN USE: WEB SEARCH AT BING

The results above are on one corpus—TREC-Robust’04, based on documents from the TREC ad-hoc collections—and labels from trained assessors working over simulated information needs. At Bing we have also seen good results with our web corpus, queries from real Bing use, and labels from searchers with real needs. Accordingly we have been using LLMs, in conjunction with a reduced number of human labellers, for most of our offline metrics since late 2022.

#### 5.1 Experience with LLMs at Bing

At Bing we have made heavy use of crowd workers, for many years, to scale to the number of labels, languages, and markets we need. Despite systems for detecting and removing low quality labels and workers, this scale has come at a cost of natural biases, mistakes, and adversarial workers.

In Table 5 we summarise our experiences with labelling to date, considering (top to bottom) full-time Bing employees (mainly scientists and engineers working on metrics); our best crowd workers, recruited and trained specifically for metrics problems and with close oversight; our general pool of crowd workers, subject to quality control but minimal training; and our LLM models, based on GPT-4. LLM models give us better accuracy at vastly reduced latency and cost. In current work with newer models and prompts, we expect to see a further increase in accuracy of 8–10% in some languages, with around five times the throughput.



Large language models can accurately predict searcher preferences

The prompts in use are confidential. In our case we include the URL, since this is always defined for web documents; we also include date, location, language and other information available from our logs. In our experience LLMs do remarkably well. They have proved more accurate than any third-party labeller, including staff; they are much faster end-to-end than any human judge, including crowd workers; they scale to much better throughput; and of course are many times cheaper. This has let us measure many more results than previously, with associated gains in sensitivity (we can see smaller effects if we label more things). The end-to-end speed, also much improved, is helping Bing engineers try more things and get more done.

## 5.2 Evaluating labellers and prompts

In Bing’s case we have found breadth preferable to depth: that is, we prefer small data for many queries to the TREC-Robust approach of more data for fewer queries. All else being equal, we also prefer queries which resemble a real web search workload rather than the invented needs of TREC-Robust.

Our gold labels are, therefore, largely gathered in situ: from employees and contractors in the context of their normal search activity, and also from feedback from the general public. This data is collected at or close to the time of need, by people who had the need, and in view of a full SERP (including e.g. images, maps, and advertisements). These properties mean the data is very reliable: if a label says some document is good (or bad), it is almost certainly so in the eyes of the person who issued the query.

Our ground truth corpus comprises queries, descriptions of need, metadata like location and date, and at least two example results per query. Results are tagged—again, by the real searcher—as being good, neutral, or bad and these tags may be reviewed by Microsoft staff prior to inclusion in our corpus. Similar to the TREC experiments above, from this we can derive pairs of preferred and non-preferred results and then treat labelling and scoring as a binary classification problem: the preferred result should score higher than the non-preferred, for all queries and pairs of results. Again, we can use pairwise agreement to evaluate the labels. At the time of these experiments our ground corpus comprised about 2.5 million such pairs, in about ten languages and from about fifty countries.

Using three labels does conflate small distinctions (“it’s a little bit better”, e.g. good vs neutral results) and large distinctions (“it’s a lot better”, good vs bad results), but our ground truth corpus has distinct advantages in that we can collect preferences from real searchers in their own context, and providing a preference is easier than providing absolute labels [Carterette et al. 2008]. Moreover, the focus on general labels maximises the reuse of the corpus as the definition of a good or bad result is unlikely to evolve over time, whereas subtle distinctions might be subject to change.

Our user-generated ground truth corpus gives us an evaluation which is independent of the labels from third-party judges. In particular, by measuring against user-generated labels we can identify cases where the model is more accurate than third-party human judges; if we only had third-party labels, we could identify labelling disagreements but not resolve them one way or the other. For AUC scores to be useful, of course the data must represent some population of interest: at Bing we stratify the triples by language and by important result attributes (for example recency, authority, or topicality). This is not a uniform sample but instead lets us identify areas of particular concern.

## 5.3 Monitoring the LLM system

The results above give us a good deal of confidence that a large language model, appropriately prompted, can produce high-quality labels for at least some of the aspects important to our ongoing evaluation. As an additional safety check, we routinely compare the LLM’s labels to those from (trained and qualified) assessors. Every week, we take a stratified sample of query:document pairs labelled by the model, chosen from amongst those that our experiments have used

recently. Those are re-labelled by our reviewers, and we monitor for shifts either in disagreement rate or patterns of disagreement; any changes are investigated by a dedicated metrics team with expertise in both the crowd and LLM processes. In practice, large changes are rare, and resolved in favour of the LLM as often as in favour of the humans. Since we use a highly skilled set of judges this remains an expensive process, but it is relatively lightweight and to date has needed less than a day a week of employee time.

In addition to the human oversight of our LLM based labels we have a large set of queries that we consistently relabel. On a day-to-day basis we expect no change in the labels associated with this set; that is, the expected value of day  $n$  labels – day  $n + 1$  labels is zero. This automated system is designed to monitor the health of labelling systems and provides a more rapid response than the human based evaluation.

Our system therefore sits somewhere between Clarke et al.’s “manual verification” and “fully automated” options [2023], with the scale of a fully automated system but some degree of control and quality assurance from manual verification. Disagreements, and analyses of these, can inform future developments of the metrics and the gold set as well as the LLM labeller.

We note, too, that although LLM labels are important to our evaluation they are only one part of a web-scale search system. Amongst other things, web search needs to account for spam, misinformation, piracy, and other undesirable material; needs to treat some topics carefully and with editorial input (health, finance, and others); and needs to account for diversity in the final ranking. Our LLM prompts are not intended to replace these or other safety systems.

## 6 POTENTIAL LIMITATIONS AND PITFALLS

Using LLMs for automated relevance labelling is a recent phenomenon, and initial evidence is promising to say the least. The field would, however, also benefit from acknowledging how little we understand potential limitations and negative externalities of these approaches. Language models are known to reproduce and amplify harmful stereotypes and biases of social import [Bender et al. 2021; Blodgett et al. 2020; Bolukbasi et al. 2016; Caliskan et al. 2017; Gonen and Goldberg 2019] and therefore there is an immediate need to study if and how these biases may also manifest in relevance labelling. These biases may further intensify existing representational and allocative harms from search systems [Noble 2018; Sweeney 2013]. Other forms of bias unrelated to concerns of demographic fairness—such as under-estimating the relevance of longer documents [Hofstätter et al. 2020]—may also manifest more systemically when relevance labels are solicited from LLMs rather than crowd-workers. It may be tempting to suggest employing a variety of different prompts and underlying LLMs to address this issue—similar to employing a diverse group of crowd-workers—but that may or may not have the desired effect if the outputs across these variations are correlated and exhibit similar biases. The quality of LLM-generated relevance labels may also vary disproportionately for content that is in different languages, from different geographical locations, and for different demographic groups due to disparate availability of data across these dimensions that have been employed for LLM training. Efforts to address these biases may further create undesirable incentives for more pervasive data collection and user surveillance.

Developers of search systems who evaluate using and optimise towards these LLM-based labels also risk falling into the trap of over-fitting to the idiosyncrasies of the LLM rather than towards improving true relevance, in line with Goodhart’s law [Chrystal and Mizen 2001; Goodhart 1975; Hoskin 1996; Thomas and Uminsky 2022]. Agreement with our in-situ or TREC gold labels suggests this is not yet a problem—we are closer to the ground truth with LLMs than with third-party assessors—but this may change as large models play a bigger role in ranking or as web authors start optimising for LLM labels. LLM-generated relevance labels may also show bias towards ranking models that themselves

Large language models can accurately predict searcher preferences

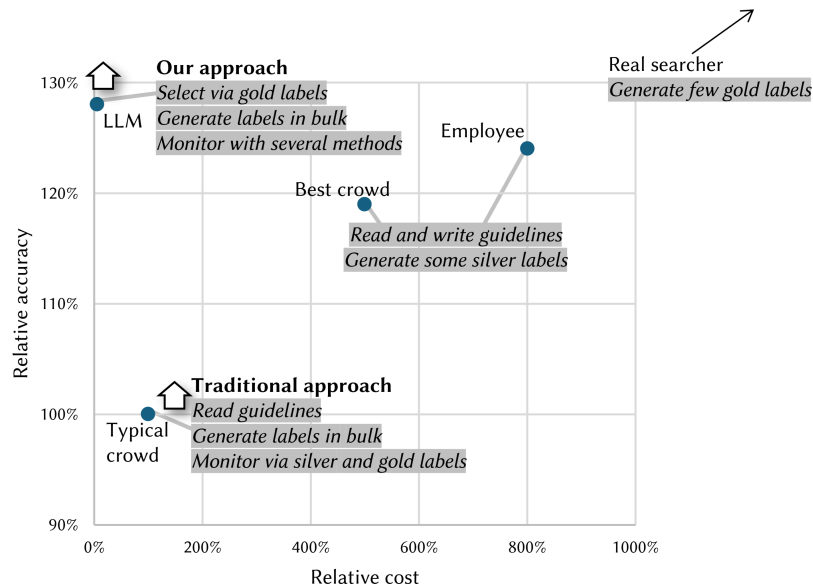


Fig. 5. Labelling options discussed in this work, along with the cost and accuracy we see at Bing. All else being equal, as experimenters we would like to move up and left in this space. A traditional approach uses gold and silver labels to improve crowd workers; we use gold labels to select LLMs and prompts.

incorporate LLMs, although if we are to truly embrace the lens of knowledge distillation in describing the evaluation and optimisation using these labels then those biases may at least be partially justified.

Biases may arise not just from LLMs learning spurious correlations with respect to its inputs, but due to the absence of certain information that human annotators would have access to (e.g. images and other non-textual content), and more subtly due to differences in what these models and humans pay attention to [Bolotova et al. 2020; Kazai et al. 2022]. Whether website designers can take advantage of such biases in LLMs-for-labelling systems to unfairly gain more exposure for their content, or whether large chunks of the web optimising towards what LLMs deem important leads to undesirable shifts in trends and homogenisation of online content, are also important topics for future research. Examples of the latter can be witnessed in other domains such as the impact of online streaming services on length of songs in the music industry.<sup>3</sup>

Lastly, the ecological costs of these LLMs are still heavily debated [Bender et al. 2021; Bommasani et al. 2021; Dodge et al. 2022; Patterson et al. 2022, 2021; Wu et al. 2022] but represent an important aspect in which these models should continue to be studied and scrutinised as appropriate in near future and beyond.

## 7 CONCLUDING REMARKS

Evaluating information retrieval typically relies on relevance labels, and we have several options for collecting these. Figure 5 illustrates the options discussed in this paper, with the cost and accuracy we see at Bing. As experimenters, our goal is to move up and left, to greater accuracy and lower cost. Traditionally the goal has been to improve crowd labels, that is to move the bottom-left point higher up, and this has involved (i) collecting insight from real users (or

<sup>3</sup><https://www.theverge.com/2019/5/28/18642978/music-streaming-spotify-song-length-distribution-production-switched-on-pop-vergecast-interview>

from experimenters themselves), (ii) turning these into guidelines, (iii) using trusted workers to read these guidelines and generate “silver” labels, and (iv) giving the same guidelines to crowd workers. The crowd workers are monitored against the silver labels, and improvements largely come from improving the guidelines.

Our approach is different: we collect high-quality gold labels from searchers themselves (searchers in situ at Bing, topic developers in TREC) and use these labels to evaluate and select prompts for a large language model. The labels we get from our model are high quality, and in practice are more useful than those from even trained assessors. They are of course cheaper to acquire, and easier to collect for new languages or other new context; but they are also more accurate than third-party labels at predicting the preference of real searchers. This has had a tangible effect on our operations: retraining parts of our ranker using labels from this model, while keeping all else constant, resulted in about six months’ relevance improvement in a single step.

Of the options described by Faggioli et al. [2023], our labelling is closest to “human verification: LLMs are considered crowdworkers, ... controlled by a human”, although we do not deliberately vary the LLM’s characteristics. We do retain human oversight and audit examples of LLM output, although we do not audit every label. Quality control, and indeed measuring LLM quality in general, is (as anticipated by Faggioli et al.) difficult as in most cases our LLM is “beyond human” quality and we can no longer rely on third-party assessors. Our gold collection, with queries and labels from real searches and real searchers, helps a great deal but of course searchers can still be swayed by distracting captions or unreliable results. (We review every query and URL in the corpus, but this only adds another human to the loop.) Contra Clarke et al., we do not see machine-made assessments degrading quality at all; nor do we consider them “very expensive”, at least compared to trained annotators.

In some ways, this is an easy case: the language model was trained on web text and we are labelling web text. The notion of judging web pages is likely already encoded, although we do not have clear evidence for this. Further, the topics can be addressed in the corpus: they do not need any personal, corporate, or otherwise restricted data, nor any particular domain-specific knowledge not already found in the text. Using LLMs for labelling suggests new and more difficult applications, for example labelling private corpora where we cannot give human assessors access. From the experiments above, we cannot verify this will be effective, and this remains for future work. We have also measured our labels in part with test sets—both TREC, and Bing’s corpus—which have clear task descriptions. If we were to sample a query load from a running system, we would not have these descriptions and our labels would be less accurate. We also have a capable model: Liang et al. [2022] saw large differences from model to model over a range of tasks, although given our observations in Section 4 this could also be due to model:prompt interactions. As new models emerge, their performance will of course need to be tested.

As our models improve, we are also faced with increasing difficulty measuring our labels as our measures start to saturate [Faggioli et al. 2023]. We have found it necessary to build “harder” gold sets over time, encoding finer distinctions to better distinguish labellers and prompts. There is no equivalent mechanism in TREC or other open data sets, and this may become pressing if and when LLM-based labelling becomes commonplace.

It is certainly possible to use large language models to label documents for relevance and therefore to evaluate search systems; it is possible to get performance comparable to TREC judges and notably better than crowd judges. There are many choices that make a difference, meaning we need metrics-for-metrics to distinguish a good from a bad system, as well as ongoing audits and human verification. True “gold” judgements (e.g. from TREC assessors or our ground-truth set) make it possible to experiment with prompt and metric design. We have found the approach productive at Bing, and have used it for greater speed, reduced cost, and substantial improvements in our running system.

Large language models can accurately predict searcher preferences

## ACKNOWLEDGMENTS

We thank David Soukal and Stifler Sun for their effort developing and testing many iterations of Bing’s LLM labelling system. Ian Soboroff kindly provided TREC-Robust judging guidelines. Dave Hedengren, Andy Oakley, and colleagues at Bing provided useful comments on the manuscript.

## REFERENCES

- Aashish Agarwal, Ankita Mandal, Matthias Schaffeld, Fangzheng Ji, Jhiaio Zhan, Yiqi Sun, and Ahmet Aker. 2019. Good, neutral or bad news classification. In *Proceedings of the Third International Workshop on Recent Trends in News Information Retrieval*. 9–14.
- Meysam Alizadeh, Maël Kubli, Zeynab Samei, Shirin Dehghani, Juan Diego Bermeo, Maria Korobeynikovo, and Fabrizio Gilardi. 2023. Open-source large language models outperform crowd workers and approach ChatGPT in text-annotation tasks. arXiv:2307.02179 [cs.CL]
- Peter Bailey, Nick Craswell, Ian Soboroff, Paul Thomas, Arjen P. de Vries, and Emine Yilmaz. 2008. Relevance Assessment: Are Judges Exchangeable and Does It Matter. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 667–674.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*.
- Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of “bias” in NLP. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 5454–5476.
- Valeria Bolotova, Vladislav Blinov, Yukun Zheng, W Bruce Croft, Falk Scholer, and Mark Sanderson. 2020. Do people and neural nets pay attention to the same words: studying eye-tracking data for non-factoid QA evaluation. In *Proceedings of the ACM International Conference on Information and Knowledge Management*. 85–94.
- Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advances in neural information processing systems* 29 (2016).
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. arXiv:2108.07258 [cs.LG]
- Andrei Broder. 2002. A taxonomy of web search. In *ACM Sigir forum*, Vol. 36. ACM New York, NY, USA, 3–10.
- Jake Brutlag. 2009. Speed matters for Google web search. Online: [https://services.google.com/fh/files/blogs/google\\_delayexp.pdf](https://services.google.com/fh/files/blogs/google_delayexp.pdf). Downloaded 2023-09-14..
- Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186.
- Ben Carterette, Paul N Bennett, David Maxwell Chickering, and Susan T Dumais. 2008. Here or there: Preference judgments for relevance. In *Proceedings of the European Conference on Information Retrieval*. 16–27.
- Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini, and Sebastiano Vigna. 2006. A reference collection for web spam. *SIGIR Forum* 40, 2 (Dec. 2006), 11–24.
- K. Alec Chrystal and Paul D. Mizen. 2001. Goodhart’s law: Its origins, meaning and implications for monetary policy. Prepared for the Festschrift in honour of Charles Goodhart.
- Charles L A Clarke, Gianluca Demartini, Laura Dietz, Guglielmo Faggioli, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Ian Soboroff, Benno Stein, and Henning Wachsmuth. 2023. HMC: A spectrum of human-machine-collaborative relevance judgment frameworks. In *Frontiers of Information Access Experimentation for Research and Education*, Christine Bauer, Ben Carterette, Nicola Ferro, and Norbert Fuhr (Eds.). Vol. 13. Leibniz-Zentrum für Informatik. Issue 1.
- Paul Clough, Mark Sanderson, Jiayu Tang, Tim Gollins, and Amy Warner. 2013. Examining the limits of crowdsourcing for relevance assessment. *IEEE Internet Computing* 17, 4 (2013).
- Gordon V Cormack, Christopher R Palmer, and Charles L A Clarke. 1998. Efficient construction of large test collections. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 282–289.
- Tadele T. Damessie, Taho P. Nghiem, Falk Scholer, and J. Shane Culpepper. 2017. Gauging the quality of relevance assessments using inter-rater agreement. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Jesse Dodge, Taylor Prewitt, Remi Tachet des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A Smith, Nicole DeCario, and Will Buchanan. 2022. Measuring the carbon intensity of AI in cloud instances. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*. 1877–1894.
- Susan Dumais, Robin Jeffries, Daniel M. Russell, Diane Tang, and Jaime Teevan. 2014. Understanding user behavior through log data and analysis. In *Ways of knowing in HCI*, Judith S. Olson and Wendy A. Kellogg (Eds.). Springer, New York, 349–372.
- Guglielmo Faggioli, Laura Dietz, Charles Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on large language models for relevance judgment. arXiv:2304.09161 [cs.IR]
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd-workers for text-annotation tasks. arXiv:2303.15056 [cs.CL]
- Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

- 609–614.
- Charles A E Goodhart. 1975. Problems of monetary management: The UK experience. In *Papers in Monetary Economics*. Vol. 1. Reserve Bank of Australia.
- Google LLC. 2022. General Guidelines. <https://guidelines.raterhub.com/searchqualityevaluatorguidelines.pdf>, Downloaded 29 July 2023..
- William Hersch, Chris Buckley, TJ Leone, and David Hickam. 1994. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 192–201.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*. <http://arxiv.org/abs/1503.02531>
- Sebastian Hofstätter, Hamed Zamani, Bhaskar Mitra, Nick Craswell, and Allan Hanbury. 2020. Local self-attention over long text for efficient document retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2021–2024.
- Keith Hoskin. 1996. The ‘awful’ idea of accountability: Inscribing people into the measurement of objects. In *Accountability: Power, ethos and technologies of managing*, R Munro and J Mouritsen (Eds.). International Thompson Business Press, London.
- Oana Inel, Tim Draws, and Lora Aroyo. 2023. Collect, measure, repeat: Reliability factors for responsible AI data collection. arXiv:2308.12885 [cs.LG]
- Andrej Karpathy. 2023. State of GPT. Seminar at Microsoft Build. <https://build.microsoft.com/en-US/sessions/db3f4859-cd30-4445-a0cd-553c3304f8e2>.
- Gabriella Kazai, Bhaskar Mitra, Anlei Dong, Nick Craswell, and Linjun Yang. 2022. Less is Less: When are Snippets Insufficient for Human vs Machine Relevance Estimation?. In *Proceedings of the European Conference on Information Retrieval*. 153–162.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. arxiv:2205.11916 [cs.CL]
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic evaluation of language models. arXiv:2211.09110 [cs.CL]
- Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.
- Safiya Umoja Noble. 2018. Algorithms of oppression. In *Algorithms of oppression*. New York University Press.
- OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]
- David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluís-Miquel Munguia, Daniel Rothchild, David R So, Maud Texier, and Jeff Dean. 2022. The carbon footprint of machine learning training will plateau, then shrink. *Computer* 55, 7 (2022), 18–28.
- David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluís-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. (2021). arXiv:2104.10350 [cs.LG]
- Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with “gradient descent” and beam search. arXiv:2305.03495
- Tefko Saracevic. 2008. Effects of inconsistent relevance judgments on information retrieval test results: A historical perspective. *Library Trends* 56, 4 (2008), 763–783.
- Falk Scholer, Diane Kelly, Wan-Ching Wu, Hanseul S. Lee, and William Webber. 2013. The effect of threshold priming and need for cognition on relevance calibration and assessment. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 623–632.
- Eric Schurman and Jake Brutlag. 2009. Performance related changes and their user impact. In *Velocity web performance and operations conference*.
- Latanya Sweeney. 2013. Discrimination in online ad delivery. *Commun. ACM* 56, 5 (2013), 44–54.
- Paul Thomas, Gabriella Kazai, Ryen W. White, and Nick Craswell. 2022. The crowd is made of people: Observations from large-scale crowd labelling. In *Proceedings of the Conference on Human Information Interaction and Retrieval*.
- Rachel L. Thomas and David Uminsky. 2022. Reliance on metrics is a fundamental challenge for AI. *Patterns* 3, 5 (2022).
- Petter Törnberg. 2023. ChatGPT-4 outperforms experts and crowd workers in annotating political Twitter messages with zero-shot learning. arXiv:2304.06588 [cs.CL]
- Ellen M Voorhees. 1998. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 315–323.
- Ellen M Voorhees. 2004. Overview of the TREC 2004 Robust Retrieval Track. In *Proceedings of the Text REtrieval Conference*.
- Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? Exploring the state of instruction tuning on open resources. arXiv:2306.04751 [cs.CL]
- William Webber, Alistair Moffat, and Justin Zobel. 2010. A Similarity Measure for Indefinite Rankings. *ACM Transactions on Information Systems* 28, 4, Article 20 (Nov. 2010).
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. arXiv:2201.11903 [cs.CL]
- Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable AI: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems* 4 (2022), 795–813.
- Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimisers. arXiv:2309.03409 [cs.LG]

## Large language models can accurately predict searcher preferences

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E. Gonzalez. 2022. TEMPRA: Test-time prompt editing via reinforcement learning. [arXiv:2211.11890 \[cs.CL\]](#)

Yongchao Zhou, Andrei Ioan Muresanu, Ziwon Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. [arXiv:2211.01910 \[cs.LG\]](#)