Information Retrieval Research 1 (2025) 29–46 Submitted 09/24; Revised 02/25; Published 03/25

Don't Use LLMs to Make Relevance Judgments

Ian Soboroff IAN.SOBOROFF@NIST.GOV
National Institute of Standards and Technology
Gaithersburg, Maryland, USA

Editor: Vanessa Murdock, Djoerd Hiemstra

Abstract

Relevance judgments and other truth data for information retrieval (IR) evaluations are created manually. There is a strong temptation to use large language models (LLMs) as proxies for human judges. However, letting the LLM write your truth data handicaps the evaluation by setting that LLM as a ceiling on performance. There are ways to use LLMs in the relevance assessment process, but just generating relevance judgments with a prompt isn't one of them.[1]

Keywords: Information Retrieval Evaluation, Large Language Models

1 Introduction

The Text Retrieval Conference (TREC) is a community evaluation and dataset construction activity sponsored by the U.S. National Institute of Standards and Technology (NIST). TREC has run annually since 1991. TREC is divided into *tracks* which embody specific search tasks. The canonical TREC task is *adhoc search*, with searches against a static set of documents, each search returning a single ranked list of documents. An individual search instance is called a *topic* and expresses the user's information need in long form rather than providing a query. The relevance judgments, or *qrels*, maps each topic to the documents that should be retrieved for it. The combination of the document collection, the topics, and the relevance judgments is called a *test collection*.

The relevance judgments representing ground truth are created collaboratively between participants and NIST using a process called *pooling* (Jones and van Rijsbergen, 1975). TREC participants use their IR systems to return the top K documents for each topic. The union of the top-ranked $k \ll K$ documents from each participant system is called the *pool*. The documents in the pool are reviewed by the person who invented the topic, and they decide which documents are relevant and which are not. Using the qrels as labels we can compute various measures of retrieval effectiveness such as precision and recall. The test collections let researchers rapidly innovate new search algorithms in a laboratory setting before deploying them to a live system. More information about TREC can be found in Voorhees and Harman (2005), a book covering the first ten years of the program.

The process used in TREC descends from the Cranfield indexing experiments conducted by Cyril Cleverdon in the 1960s, and so we say TREC is following the Cranfield paradigm (Cleverdon, 1967). Central to the Cranfield paradigm is a set of assumptions that simplify the search problem: the document collection and information needs are fixed,

1. This article reflects the views of the author and not necessarily those of NIST or of the U. S. Government.

 WWW: https://irrj.org DOI: 10.54195/irrj.19625

all documents are labeled as relevant or not relevant to every query, relevance is modeled by topical similarity, the relevance of a document is independent of the relevance of any other document, there is a single query that is answered with a single ranked list, and the relevance judgments are representative of the user population (Voorhees, 2001). TREC can be thought of as a community effort in pushing the bounds of the Cranfield paradigm. Complete judgments were replaced with the pooling procedure, which has been shown to be sufficient for measuring the pooled systems and also useful for measuring systems which were not pooled for evaluation, as long as certain properties are maintained (Harman, 1995; Zobel, 1998; Buckley et al., 2007). Likewise, many TREC tracks push back on the notion of static documents and information needs (Frank et al., 2014; Carterette et al., 2014), relevance as topical similarity (Craswell and Hawking, 2004; Balog et al., 2011), single rankings (Owoicho et al., 2022; Aliannejadi et al., 2023), and independent relevance (Soboroff and Harman, 2005).

Making the relevance judgments for a TREC-style test collection can be complex and expensive. Relevance assessing at NIST for a typical TREC track usually involves a team of six contractors working for 2-4 weeks. Those contractors need to be trained and monitored. Software has to be written to support recording relevance judgments correctly and efficiently. Experience in both the technical and human aspects of the process counts for a lot, which is why we run evaluation campaigns rather than everyone building their own test collections. Evaluation campaigns are infrastructure for IR research.

The recent advent of large language models that produce astoundingly human-like flowing text output in response to a natural language prompt has inspired IR researchers to wonder how those models might be used in the relevance judgment collection process (Bauer et al., 2023; Faggioli et al., 2024).

At the ACM SIGIR 2024 conference, a workshop "LLM4Eval" provided a venue for this work, and featured a data challenge activity where participants reproduced TREC deep learning track judgments, as was done by Thomas et al. (2024). I was asked to give a keynote at the workshop, and this paper presents that keynote in article form.

The bottom-line-up-front message is, don't use LLMs to create relevance judgments for TREC-style evaluations.

2 Automatic evaluation

The idea of automatic evaluation for information retrieval came from a paper I wrote with Charles Nicholas and Patrick Cahan in 2001 (Soboroff et al., 2001). I had been reading Ellen Voorhees well-known SIGIR paper from 1998 (Voorhees, 1998) which shows experimentally that while people differ in their judgments of relevance, those differences don't affect the relative ordering of systems in a TREC evaluation. Surprised by this result, I wondered what would happen if the relevance judgments were randomly sampled from the pools. Certainly, TREC relevance judgments aren't random, but how much can they vary towards random and still rank systems equivalent to the official system ranking?

A representative example result from that work is shown in Figure 1. The single +'s are official scores from TREC, and the ×'s with whiskers up and down are scores obtained with random documents from the pool labeled as relevant. The points are ordered according to their official TREC scores. The key point to notice is that, using random judgments, the

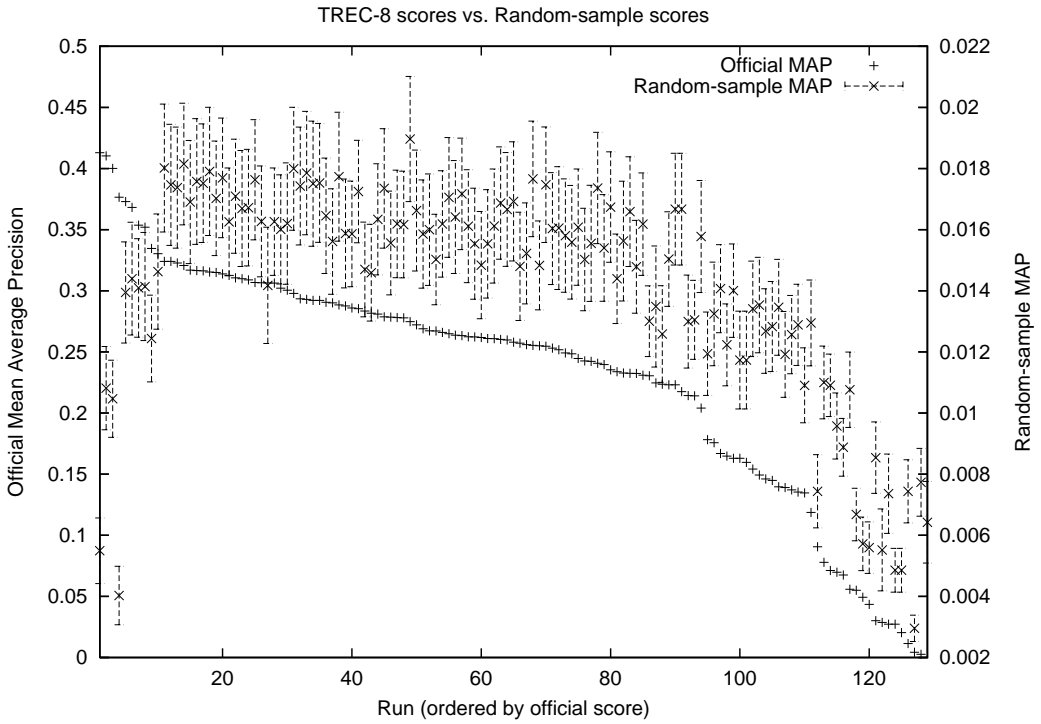

Figure 1: Sample result from Soboroff et al. (2001), TREC-8, TREC-style pooling to depth 100.

best systems (those with the highest MAP, on the left) look like the worst systems (on the right).

Automatic evaluation in this sense means making relevance judgments using an algorithm rather than people, as opposed to inducing relevance from implicit behavior cues or history. In published papers proposing automatic evaluation methods for IR, the quality of these methods is quantified by comparing the ordering of systems induced by the automatic method to the official TREC ranking based on manual assessments. That is, the qrels are used to score each system using some metric, and the systems are then ordered by their score. The common metric used is Kendall's tau (τ), a correlation measure between rankings. Some researchers also use the more familiar Spearman's rho (ρ), a correlation measure that takes into account the distance between the points and not just their rank order. Others have proposed versions of these correlations that emphasize the upper part of the ranking (i.e. the best systems) (Yilmaz et al., 2008). I have heard a discussion of a variation of tau where a swap in position between two systems only happens if they are

significantly different according to some statistical test, which would take advantage of the fact that the points in the ranking represent average performance over a set of topics.

Aslam and Savell (2003) published a short paper that explained my 2001 results. No matter how many systems retrieve a given document, it is only added to the pool once. There are many more nonrelevant than relevant documents, so a given relevant document was likely to have been retrieved by more than one system. (The mean number of systems retrieving a nonrelevant document in TREC-8 is 3, versus 11 for a relevant document.) By selecting the relevant documents at random, I was implicitly selecting documents retrieved by many systems. So my 2001 paper shows the results of a popularity contest. Under this approach, the worst systems and the best systems both look bad, because they fail (or succeed) by retrieving documents that other systems don't find. Another way to think about Aslam and Savell is that by using the output of a system as the ground truth, I am measuring the similarity of the two systems, how close the retrieval system is to the model that created the ground truth.

Around that time, the BLEU automatic metric for machine translation (Papineni et al., 2002) and the ROUGE automatic metric for summarization (Lin, 2004) were published. These measures compare system-generated outputs, such as translations and summaries, by the overlap of word n-grams with a model or reference output. ROUGE worked well for extractive summarization, where a summary is produced by cutting and pasting sentences from the source documents, but less well in a generative setting where word choice could vary quite a bit from the original documents.

3 Machine learning and predicting from examples

A more successful method of imputing relevance comes in the form of relevance feedback and more sophisticated machine learning algorithms. In these cases, examples of relevant (and possibly nonrelevant) documents are used to train a model to predict the relevance of other documents.

Relevance feedback (RF) in the vector space model, developed by Joseph Rocchio and Eleanor Ide in the mid to late 1960s as part of the SMART system (Salton and McGill, 1983), may be the first on-line machine learning algorithm.[2] In RF, the user executes an initial search and identifies one or more documents in the search results as being relevant or irrelevant. The terms in the query are augmented and reweighted based on the feedback, and the refined query is executed to rank the remaining documents in the collection. Since then, it has been adopted as a general IR technique rather than a specific algorithm and has been instantiated within nearly every ranking model. Currently, the most common implementation of RF uses the BM25 ranking algorithm with the RM3 method of term weighting (Jaleel et al., 2004). Pseudo-relevance feedback (PRF) is a modification in which instead of indicating relevance by the user, some number of documents ranked highest in the initial ranking are assumed to be relevant for a round of relevance feedback (Buckley et al., 1994). Relevance feedback is one of the most successful techniques in information retrieval, producing large improvements in performance. PRF is somewhat similar except that for some topics it fails because the initial retrieval is off base for some reason.

2. Rocchio did not describe his method as machine learning, but he did develop a theory that his relevance feedback method builds an optimal query.

Büttcher et al. (2007) proposed using the TREC relevance judgments to predict the relevance of unjudged documents retrieved by unpooled systems, and also as a method for expanding the set of relevance judgments overall. They use the qrels to train a binary classifier and then apply that to documents that were not judged but were ranked above the pool depth of TREC's pools. Anecdotally, this technique did not perform as well when all retrieved documents (down to rank 1000) were predicted, so there is something to restricting predictions to those documents that are already ranked highly by the search ranker.

Rajput et al. (2012) describe an iterative method using *nuggets*. A nugget here is a manually selected passage from a relevant document. Starting with manual nuggets, the process identifies new high-probability shingles as new nuggets and uses those to predict the relevance of other documents. Their use of the term "nugget" is different than how the term is used for evaluation of summarization and question answering; summarization nuggets are atomic pieces of information which must be manually aligned to the generated summary, whereas these nuggets, being strings or shingles, are automatically matched. Nuggets in (Rajput et al., 2012) are essentially lexical patterns that identify relevant documents.

The BERTScore metric (Zhang et al., 2020) computes token similarity between a generated and reference output using BERT embeddings. We can think of this as the LLM equivalent of BLEU, using embeddings instead of n-grams.

4 LLM-based predictions of relevance

Modern large language models (LLMs) have inspired a new approach, where a topic and document are embedded in a prompt, which is then fed to an LLM that outputs some indicator of relevance. LLMs may be fine-tuned with relevance examples, or other relevant documents may be included in the prompt, but otherwise no examples are used as in the supervised learning methods above.

Thomas et al. (2024) describe using LLMs to predict relevance in TREC collections as well as for search results from a major commercial search engine. They develop prompts at a number of levels of richness. In their web search results, they find that the generated judgments "have proved more accurate than any third-party labeler, 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." The paper describes results on TREC data in greater detail, and there is an extended discussion of their prompts and their evolution.

MacAvaney and Soldaini (2023) used nearest-neighbors, classifiers, and LLM prompts to elicit relevance judgments to supplement judgments in MS-MARCO (Nguyen et al., 2016), a collection where there is only a single manually-adjudicated relevance judgment per query. By basing a system's performance measurement on more than one document, IR metrics are found to be more stable.

Alaofi et al. (2024) investigated the agreement between LLMs and TREC assessors and found that LLM false positive decisions seemed to be related to the presence of query terms in the passage being assessed. A false positive is where the LLM votes that the passage is relevant, but the human assessor judged it to not be relevant. In many of these cases, the false positive passages included terms from the query, despite not being relevant. This

seems to imply that despite the richness of the language model, lexical cues can influence the decision more than the true meaning of the text.

Outside of the information retrieval domain, researchers seem to be eagerly jumping on a bandwagon for LLM-based automatic evaluation. As one example, Lin and Chen (2023) employ prompts to gauge generated responses in open-domain dialogues, and compare results to other automatic evaluation techniques, some of which use the LLM to identify properties of good responses (Mehri and Eskenazi, 2020b,a) and others which use the LLM to directly assess dialog responses (Chen et al., 2023; Fu et al., 2024). None of the comparison metrics is validated against manual labels of the dialogues in question.

The search for automatic metrics is long and has made use of new algorithms as they have been developed. There is a real need for automatic metrics, because manual assessment is slow and hard to scale. When the labels are created zero-shot, specifically meaning that the evaluation model is operating at the same degree of data exposure as the systems being measured, the evaluation reduces to comparing the performance of the system to the model, not to human performance. When the evaluation model has more knowledge than the systems being measured, for example relevance judgments on the topics in the test set, then the model may produce an evaluation that can stand as a useful measurement, a comparison to something more than just another system. When the evaluation model is making use of outside knowledge, for example in Mehri and Eskenazi (2020a), then the situation depends on the systems being measured. The following sections elaborate this argument.

5 Retrieval and evaluation are the same problem

Asking a computer to decide if a document is relevant is no different than using a computer to retrieve documents and rank them in order of predicted degree of relevance. In both cases, the algorithm makes the assessment of relevance.

A retrieval system, or a relevance model, is a model of relevance given available data. The system is trying to predict which documents are relevant and which documents are not. Even though real systems might try to optimize a pairwise or listwise output or compute a degree of relevance of a document or a search engine result page, it is useful to think of all these processes as predicting relevance.

During relevance assessment, we are asking the assessor to decide whether documents are relevant or not. This, too, is essentially a prediction of relevance. It's a well-informed prediction since the person is reading the document and often composed the information need, but since the task is artificial, the assessor is basically saying that they would include this document in a report on the topic, a report which they don't ever actually write. We can call that a prediction too.

We use one set of predictions, the relevance judgments, to measure the performance of the other set of predictions, the system outputs. In doing so, we declare the relevance judgments to be truth. In fact, you can switch the two sets of predictions, declare the system output to be truth, and measure the "effectiveness" of the assessor compared to that of the system. All evaluations which compare a system output to an answer key are making a measurement with respect to the answer key, not with respect to the universe.

Since both retrieval systems and relevance assessors are making predictions of relevance, evaluation and retrieval are the same problem. We can imagine a very slow system that would have a human read every document and assess its relevance given a query.

John Searle's "Chinese Room" thought experiment[3] posits a person in a box who receives questions through a slot and delivers answers out the slot. The questions and answers are in Chinese, a language which the person does not read or speak. Rather, the person follows a sophisticated set of instructions for generating an answer from a question, in Chinese, by manipulating symbols on the paper. Thus, the box appears to understand and communicate in human language but is basically a computer. A mechanical Chinese Room can be implemented with an LLM chatbot. Construct a prompt of a topic statement and a document and ask for the LLM to say relevant or not, for every document in the collection. Asking the language model about relevance is the mirror of evaluation.

If we believed that a model was a good assessor of relevance, then we would just use it as the system. Why would we do otherwise? We don't use human assessors that way, because it doesn't scale. LLMs in 2024 don't scale, but that feels like an engineering problem more than something fundamental; we will probably solve this with better hardware and smaller models.

Since both retrieval and evaluation are prediction activities it seems natural to apply machine learning to both. The predictions don't happen in isolation: systems know about collection frequencies and click patterns that inform the ranked list, and assessors have experience and world knowledge that informs their labels. Machine learning, the field where we train prediction systems by example, clearly has a role to play here.

As with any prediction, there are errors of omission and commission (or false negatives and false positives if you prefer), and those errors represent a maximum discriminative ability of those relevance judgments to distinguish systems. I will dive into this in more detail in the next section.

6 The ceiling on performance

Whatever we use as the answer key represents both an ideal solution and a ceiling on measurable performance. No system can outperform the evaluation's answer key. When the answer key is made up of human-assigned labels, then we are saying that human performance is the ideal we are aiming for, and we can't measure something better than that performance. Likewise, when the answer key is created by a machine learning model or some other mechanical process, we are saying that the model represents the idea we are aiming for, and we can't measure something better than the performance of that model. This is the critical flaw with LLM-sourced relevance judgments.

An IR test collection is a 3-tuple:

$$C = \{D, S, R\}$$

where D is a set of documents, S is a set of search needs, and R is a function $R : S \mapsto D$ that maps search needs to relevant documents. In the original Cranfield collections, there is a value of R for every document d and search need s. In the TREC collections, most of

3. https://en.wikipedia.org/wiki/Chinese_room

those pairs are unknown and pooling lets us assume that an unjudged document is likely not relevant.

A retrieval model produces a ranking of documents $d_n : d_n \in D$ in order of predicted relevance to the search need s:

$$A(s, D) = \{d_n \ \forall \ d_n \in D\}$$

where the set here is an ordered set, a sequence of documents where each document appears once. The entire document collection is ranked although in practice we cut off the ranking much earlier.

An evaluation function $E(A, R)$ computes a real number from a k-prefix of the ranked list A^k. Often in TREC $k = 1000$ but some measures set k much lower to focus on the top of the list. If the number of relevant documents for $sR_s >= k$, then a system can produce a ranking whose prefix consists only of relevant documents. Thus in practice we try to have search needs with many fewer relevant documents, and a fundamental difficulty with enormous collections like ClueWeb is that we can easily find thousands of relevant documents and still worry that we have not found them all (Buckley et al., 2007).

The relevance judgments in a Cranfield experiment are a model of human behavior, and since we are trying to build systems that understand information needs and documents as well as humans do, they model ideal retrieval performance. The evaluation function E is typically defined to be maximized by an ideal ranking, for example if all relevant documents are ranked ahead of any irrelevant documents. If you take the relevance judgments and turn it into a run by first listing all the relevant documents and then padding the listing to k with irrelevant documents, it gets perfect scores on the appropriate metrics.

Historically, this was the goal of IR performance. IR systems are meant to augment people by scaling up their ability to understand information, and so the performance of people is the ideal.

This ideal is also a limit on what Cranfield can measure. Under R, the best possible ranking

$$A(s, D) \mapsto \{+, +, +, +, ...-, -, -\}$$

orders the relevant documents ahead of any irrelevant documents. The order of relevant documents among themselves, and irrelevant documents among themselves, are not important: there is a very large number of equivalently ideal rankings by permutations among the relevant and irrelevant documents. For graded relevance regimes, this ranking orders documents by their rated degree of relevance, where those degrees are positive integers, zero for not relevant, and perhaps negative numbers for other poor outcomes like spam, and within each relevance degree or category the documents can be permuted to create equally ideal ranked lists. If two or more categories are equivalently valuable, we can replace them with a superset including all equivalently valuable documents. Without loss of generality, moving forward I will assume that rankings can have all the relevant documents ahead of all the irrelevant documents for any relevance construct.

Theorem 1 (ideal rankings) *Let C be a test collection (D, S, R) where $R : s \mapsto d$ maps search needs to relevant documents $\{+, +, +, ...\}$. Let $A(s, D)$ be a ranking function that produces a ranking of documents $\{d_n \in D\}$ for a search need s. Let $E : A(s, D), R \mapsto \mathbb{R}$ be*

an evaluation metric that computes a real number representing the quality of the ranking A given the relevance judgments R. Then, we can define the ***ideal ranking*** *as*

$$A(s, D) \mapsto \{+, +, +, +, ...-, -, -\}$$

the ranking the places the relevant documents ahead of any irrelevant documents. The ideal ranking maximizes E, and there does not exist any ranking A' that obtains a higher value than A of E subject to R.

Proof Suppose a ranking A' that has one or more relevant documents that are not in ideal ranking A. For A' to be ideal, these extra relevant documents must appear at the head of the ranking. However, the ideal is defined subject to R, the full set of relevance judgments, and A is already defined to be ideal. If there are extra relevant documents missing from A, then A is not ideal. If the new "relevant" documents are not in R, then A' can't be ideal either. So A and A' must have the same set of relevant documents in their ranking, and $E(A, R)$ and $E(A', R)$ are equal and maximize E. ■

If we imagine we have a system that is better than a human, for example by finding relevant documents that are not in the relevance judgments or correctly ranking a document which was assessed incorrectly, that system will score less than perfectly when we score it using the human relevance judgments. This must be the case, because unjudged documents are assumed to be not relevant, and the documents found by this novel system are either absent from the relevance judgments or judged non-relevant when they should have been marked relevant. The system has retrieved documents which **according to the evaluation relevance judgments** are not relevant. And so this top-performing system is measured as performing less well than it does.

This is reflected in my 2001 paper, and with other papers that came later, but exemplified by Figure 1 above. The (true) top systems are under-ranked by the "model" of relevance. This must be true for any model of relevance that generates relevance judgments, be it human or machine. We cannot measure a system that is better than the relevance judgments. Or, rather, the evaluation can't distinguish such a system from one that performs less than perfectly.

As a counterexample, Büttcher et al. (2007) trained a model using an incomplete set of manual judgments to classify a larger set of documents automatically, improving the collection. In this case the evaluation model is privileged in comparison to the runs, in that it has relevance information that they do not. Relevance feedback nearly always improves performance, so we would expect a hybrid set of judgments like these to have the possibility of outperforming evaluation using the shallow judgments. This is outperforming the original human but only doing so by retrospective use of human relevance data.[4]

And so when the relevance judgments are created by a person, the model can't exceed the human ideal. If we had a model that had "super-human" performance, we would just make our IR system use that model. In the current state of the art, the most advanced

4. We still have a grounding problem, in that you may not believe that the model makes accurate predictions. The process can be improved by doing a second stage of relevance assessments on the classifier outputs in order to estimate the classifier error rates.

LLMs are used as components of systems that may be hypothetically measured by relevance judgments generated from the same models. Those systems cannot perform better than the model generating the relevance judgments.

Obviously, the human that created the relevance judgments is not entirely ideal. The assessor is not all-knowing, all-seeing, all-reading with perfect clarity. Assessors make mistakes, and TREC participants are fond of finding them. More importantly, the assessor is only one person; someone else with the same information need would make different judgments. If we compared the assessor's judgments to those of a secondary assessor by pretending that secondary assessor is a run, it would necessarily perform less than perfectly.

That means that even if we imagine that systems exist which perform better at the task than humans do, we can't see that improved performance in a Cranfield-style evaluation. This follows from the ideal ranking theorem above. It must also be true for **any evaluation** where we are comparing a system output to a "gold standard," for example in machine learning or natural language processing, because the gold standard represents ideal performance, and by the ideal ranking theorem, no ranking can be measured as better than the truth data.

The so-called "super-human" performance observed on benchmark datasets[5] is actually just measurement error. Super-human performance would be scored as less than ideal by the established ground truth, because performing better than a human entails making different decisions than those in the ground truth.

Some benchmarks are capable of showing super-human performance by differentiating between the humans performing the task and the humans that create the answer key. For example, an LLM may perform better than many people on a standardized test, but we can measure that because the humans taking the test are not the source of the correct answers. Likewise tests of solving analogies or complex math problems. In IR evaluations, we are only comparing to the answer key, not another person's attempt to recreate the answer key.[6] To summarize, you should not create relevance judgments using a large language model, because:

- You are declaring the model to represent ideal performance, and so you can't measure anything that might perform better than that model.

- The model used to create relevance judgments is certainly also used as part of the systems being measured. Those systems will evaluate as performing poorly even if they actually improve on the model, because improving on the model means retrieving new relevant unjudged documents that aren't in the answer key.

- When the next shiny model comes out, it will measure as performing less well than the old model, because it necessarily must retrieve unjudged documents or ones judged incorrectly as not relevant. And so the relevance judgments can only measure systems

5. For example, `https://openai.com/index/planning-for-agi-and-beyond/` and `https://venturebeat.com/ai/google-deepmind-unveils-superhuman-ai-system-that-excels-in-fact-checking-saving-costs-and-improving-accuracy/` For a contrasting view, see Tedeschi et al. (2023).
6. We wouldn't do that because assessor disagreement is reality. There are no absolutely correct answers outside the Cranfield room.

that perform worse than the state of the art at the time the relevance judgments were created.[7]

7 Limitations

The argument in this paper makes the case that using an LLM to generate the ground truth for an IR evaluation results in a substandard evaluation. However, it could certainly be the case that LLMs could play different roles in the evaluation process than inventing the answer key.

LLMs can be used to create the answer key if they have more knowledge about relevance than the systems being measured. If the evaluation model has access to privileged information, for example by being fine-tuned on manual relevance judgments on the evaluation topics, then those relevance judgments should still be able to measure systems that use the untuned model. While it might be tempting to assume that the LLM has information about relevance in the training data, we should avoid this assumption since we don't have access to the training data.

Blessing the evaluation model with extra relevance information is what makes Büttcher et al's (Büttcher et al., 2007) results work: the model is trained on relevance data, and so the trained model has the advantage over any system it is measuring that doesn't have access to that relevance data.

We actually already knew this: the fact that relevance feedback improves retrieval is a basic result in IR. If we have relevance information gleaned from many systems as we do when pooling, the outputs will perform better than any individual system and thus we can measure any individual systems with less information about relevance than the collective pool.

This still has the problem that the evaluation isn't future-proof: we might have a new model that outperforms the relevance feedback of the prior generation. We haven't seen this yet when the collections are pooled from older systems only, if the collection is well-judged (Voorhees et al., 2022). Or new models might have TREC triples (topic, document, relevance) as an explicit component in their training data and be able to make use of that in a retrieval setting.

One simple idea that seems promising is to employ a LLM to follow the assessor and look out for mistakes. This can't be done by simply asking, "Did you mean 'relevant' instead?" since people are primed to trust the computer more than they should (Logg et al., 2019; Bogert et al., 2021).[8] But it may be possible to automate a quality-control process using a model.

There are evaluation activities that don't involve creating an answer key. For example, in a user study, researchers observe user behavior and analyze those observations to draw conclusions about the experiment. LLMs might be useful in supporting the observational

7. This might be a good thing. Since new models will look worse than old models, when their developers run them through leaderboard-style benchmarks they will cast them aside because they seem to perform poorly. Then we won't ever have to worry about having a new model.

8. Logg et al. (2019) is also interesting as much behavioral literature four to five years prior seemed to find that people distrusted algorithmic recommendations. Perhaps our perspectives are changing with exposure. But see also Sparrow et al. (2011) on search's effects on memory.

process (perhaps by transcribing mouse movements and clicks in a readable way) or the analysis process (much as we use statistical models to determine significance).

At the SIGIR workshop, a questioner asked if an LLM-generated evaluation might still be useful even given its flaws. For example, Thomas et al. (2024) found LLM judgments to be as useful as crowdsourced judgments, but not better than curated judgments from a trained team. In their setup, crowd judgments represented a low rung in a tiered hierarchy of relevance judgments and system measurements. If the judgments are not meant to support a rigorous evaluation but rather as noisy training data, then the LLM judgments may be useful. But if the LLM creating the truth data is part of the search system, or is of an older generation than the search system, the results may under-report performance and not be able to distinguish improvements, as shown above. In all cases it should be kept in mind that the ideal used as a comparison point is not human performance, but model performance.

8 Conclusions

I have discussed the limitations of using models to create relevance judgments. You don't want to do that, because then you have limited what you can measure to the level of the generating model. If the generating model is also part of the evaluated systems, you are stuck in a loop, or perhaps falling into a bottomless pit.

This is similar to model collapse (Guo et al., 2024). When you train the model using its own outputs, the performance of the model decreases. The collapse mechanism is the measurement error that comes from generating the truth data using the evaluated system. In this case, evaluation is a loss function computed based on the generated truth data.

This doesn't mean that LLMs can't allow us to do amazing things. As someone who got his start in IR working with LSI (Deerwester et al., 1990), which is essentially an optimal linear embedding, I am very excited by the idea of nonlinear embeddings. IR systems that use LLMs to surmount the vocabulary boundary have enormous promise for real users.

All models have limits, and humans do too. If we want to use the model to evaluate performance, we first need to consider if we are doing something past the ability of the model as used in that evaluation paradigm. The relevance judgments barrier is a fundamental limitation of evaluations that measure systems against ground truth.

Acknowledgments and Disclosure of Funding

I thank Ellen Voorhees and Rikiya Takehi for their comments on the talk and on early versions of this paper. I also thank the attendees of the SIGIR 2024 LLM4Eval workshop for their insightful questions and continuing discussions. No external funding was received in support of this work. The TREC activity has been annually reviewed by NIST's Research Protection Office and determined to not be human subjects research. Any company, product or service mentioned in this paper should not be taken as an endorsement of that company, product, or service by NIST. Nothing in this paper should be read as a comparison to or among commercial products.

References

Marwah Alaofi, Paul Thomas, Falk Scholer, and Mark Sanderson. LLMs can be fooled into labelling a document as relevant: best café near me; this paper is perfectly relevant. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP)*, page 32–41, 2024. Association for Computing Machinery. ISBN 9798400707247. doi: 10.1145/3673791.3698431. URL https://doi.org/10.1145/3673791.3698431.

Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffery Dalton, and Leif Azzopardi. TREC IKAT 2023: The interactive knowledge assistance track overview. In Ian Soboroff and Angela Ellis, editors, *The 32nd Text REtrieval Conference Proceedings (TREC)*, volume 1328 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2023. URL https://trec.nist.gov/pubs/trec32/papers/Overview_ikat.pdf.

Javed A. Aslam and Robert Savell. On the effectiveness of evaluating retrieval systems in the absence of relevance judgments. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 361–362, 2003. Association for Computing Machinery. ISBN 1581136463. doi: 10.1145/860435.860501. URL https://doi.org/10.1145/860435.860501.

Krisztian Balog, Pavel Serdyukov, and Arjen P. de Vries. Overview of the TREC 2011 entity track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of The Twentieth Text REtrieval Conference, (TREC)*, volume 500-296 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2011. doi: 10.6028/NIST.SP.500-296.entity-overview. URL http://trec.nist.gov/pubs/trec20/papers/ENTITY.OVERVIEW.pdf.

Christine Bauer, Ben Carterette, Nicola Ferro, Norbert Fuhr, Joeran Beel, Timo Breuer, Charles L. A. Clarke, Anita Crescenzi, Gianluca Demartini, Giorgio Maria Di Nunzio, Laura Dietz, Guglielmo Faggioli, Bruce Ferwerda, Maik Fröbe, Matthias Hagen, Allan Hanbury, Claudia Hauff, Dietmar Jannach, Noriko Kando, Evangelos Kanoulas, Bart P. Knijnenburg, Udo Kruschwitz, Meijie Li, Maria Maistro, Lien Michiels, Andrea Papenmeier, Martin Potthast, Paolo Rosso, Alan Said, Philipp Schaer, Christin Seifert, Damiano Spina, Benno Stein, Nava Tintarev, Julián Urbano, Henning Wachsmuth, Martijn C. Willemsen, and Justin Zobel. Report on the Dagstuhl seminar on frontiers of information access experimentation for research and education. *SIGIR Forum*, 57(1), 2023. ISSN 0163-5840. doi: 10.1145/3636341.3636351. URL https://doi.org/10.1145/3636341.3636351.

Eric Bogert, Aaron Schechter, and Richard T. Watson. Humans rely more on algorithms than social influence as a task becomes more difficult. *Scientific Reports*, 11(8028), 2021.

Chris Buckley, Gerard Salton, James Allan, and Amit Singhal. Automatic query expansion using SMART: TREC 3. In *Proceedings of The Third Text REtrieval Conference, (TREC)*, volume 500-225 of *NIST Special Publication*, pages 69–80. National Institute of Standards and Technology (NIST), 1994.

Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen Voorhees. Bias and the limits of pooling for large collections. *Information Retrieval*, 10(6):491–508, 2007. ISSN 1386-4564. doi: 10.1007/s10791-007-9032-x. URL https://doi.org/10.1007/s10791-007-9032-x.

Stefan Büttcher, Charles L. A. Clarke, Peter C. K. Yeung, and Ian Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 63–70, 2007. Association for Computing Machinery. ISBN 9781595935977. doi: 10.1145/1277741.1277755. URL https://doi.org/10.1145/1277741.1277755.

Ben Carterette, Evangelos Kanoulas, Mark M. Hall, and Paul D. Clough. Overview of the TREC 2014 session track. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of The Twenty-Third Text REtrieval Conference, (TREC)*, volume 500-308 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2014. doi: 10.6028/NIST.SP.500-308.session-overview. URL http://trec.nist.gov/pubs/trec23/papers/overview-session.pdf.

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. Exploring the use of large language models for reference-free text quality evaluation: An empirical study. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 361–374, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-ijcnlp.32. URL https://aclanthology.org/2023.findings-ijcnlp.32/.

C. W. Cleverdon. The Cranfield tests on index language devices. *Aslib Proceedings*, 19: 173–192, 1967. (Reprinted in *Readings in Information Retrieval*, K. Sparck Jones and P. Willett, eds, Morgan Kaufmann, 1997).

Nick Craswell and David Hawking. Overview of the TREC 2004 web track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the Thirteenth Text REtrieval Conference, (TREC)*, volume 500-261 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2004. URL http://trec.nist.gov/pubs/trec13/papers/WEB.OVERVIEW.pdf.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the Association for Information Science and Technology*, 41(6):391–407, 1990. doi: 10.1002/(SICI)1097-4571(199009)41:6⟨391::AID-ASI1⟩3.0.CO;2-9.

Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. Who determines what is relevant? Humans or AI? Why not both? *Communications of the ACM*, 67(4):31–34, 2024. ISSN 0001-0782. doi: 10.1145/3624730. URL https://doi.org/10.1145/3624730.

John R. Frank, Max Kleiman-Weiner, Daniel A. Roberts, Ellen M. Voorhees, and Ian Soboroff. Evaluating stream filtering for entity profile updates in TREC 2012, 2013, and 2014. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of The Twenty-Third Text REtrieval Conference, (TREC)*, volume 500-308 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2014. doi: 10.6028/NIST.SP.500-308.kba-overview. URL http://trec.nist.gov/pubs/trec23/papers/overview-kba.pdf.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. GPTScore: Evaluate as you desire. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.365. URL https://aclanthology.org/2024.naacl-long.365/.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. The curious decline of linguistic diversity: Training language models on synthetic text, *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024. URL https://arxiv.org/abs/2311.09807.

Donna Harman. Overview of the fourth Text Retrieval Conference (TREC-4). In *TREC*, volume 500-236 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 1995.

Nasreen Abdul Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah S. Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. UMass at TREC 2004: Novelty and HARD. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the Thirteenth Text REtrieval Conference (TREC)*, volume 500-261 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2004. URL http://trec.nist.gov/pubs/trec13/papers/umass.novelty.hard.pdf.

K. Sparck Jones and C. van Rijsbergen. Report on the need for and provision of an "ideal" information retrieval test collection. Technical report, British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Yen-Ting Lin and Yun-Nung Chen. LLM-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. In Yun-Nung Chen and Abhinav Rastogi, editors, *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI)*, pages 47–58, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.nlp4convai-1.5. URL https://aclanthology.org/2023.nlp4convai-1.5/.

Jennifer M. Logg, Julia A. Minson, and Don A. Moore. Algorithm appreciation: People prefer algorithmic to human judgment. *Organizational Behavior and Human Decision*

Processes, 151:90–103, 2019. ISSN 0749-5978. doi: https://doi.org/10.1016/j.obhdp.2018.12.005. URL https://www.sciencedirect.com/science/article/pii/S0749597818303388.

Sean MacAvaney and Luca Soldaini. One-shot labeling for automatic relevance estimation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2230–2235, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3592032. URL https://doi.org/10.1145/3539618.3592032.

Shikib Mehri and Maxine Eskenazi. Unsupervised evaluation of interactive dialog with DialoGPT. In Olivier Pietquin, Smaranda Muresan, Vivian Chen, Casey Kennington, David Vandyke, Nina Dethlefs, Koji Inoue, Erik Ekstedt, and Stefan Ultes, editors, *Proceedings of the 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 225–235, 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.sigdial-1.28. URL https://aclanthology.org/2020.sigdial-1.28/.

Shikib Mehri and Maxine Eskenazi. USR: An unsupervised and reference free evaluation metric for dialog generation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 681–707, 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.64. URL https://aclanthology.org/2020.acl-main.64/.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In *Workshop on Cognitive Computation (CoCo@NIPS)*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

Paul Owoicho, Jeff Dalton, Mohammad Aliannejadi, Leif Azzopardi, Johanne R. Trippas, and Svitlana Vakulenko. TREC CAsT 2022: Going beyond user ask and system retrieve with initiative and response generation. In Ian Soboroff and Angela Ellis, editors, *Proceedings of the 31st Text REtrieval Conference, (TREC)* volume 500-338 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2022. doi: 10.6028/NIST.SP.500-338.cast-overview. URL https://trec.nist.gov/pubs/trec31/papers/Overview_cast.pdf.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Shahzad Rajput, Matthew Ekstrand-Abueg, Virgil Pavlu, and Javed A. Aslam. Constructing test collections by inferring document relevance via extracted relevant information. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, page 145–154, 2012. Association for Com-

puting Machinery. ISBN 9781450311564. doi: 10.1145/2396761.2396783. URL https://doi.org/10.1145/2396761.2396783.

Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., USA, 1983. ISBN 0070544840. URL https://sigir.org/resources/museum/.

Ian Soboroff and Donna Harman. Novelty detection: The TREC experience. In Raymond Mooney, Chris Brew, Lee-Feng Chien, and Katrin Kirchhoff, editors, *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 105–112, 2005. Association for Computational Linguistics. URL https://aclanthology.org/H05-1014/.

Ian Soboroff, Charles Nicholas, and Patrick Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 66–73, 2001. Association for Computing Machinery. ISBN 1581133316. doi: 10.1145/383952.383961. URL https://doi.org/10.1145/383952.383961.

Betsy Sparrow, Jenny Liu, and Daniel M. Wegner. Google effects on memory: Cognitive consequences of having information at our fingertips. *Science*, 333(6043):776–778, 2011. doi: 10.1126/science.1207745. URL https://www.science.org/doi/abs/10.1126/science.1207745.

Simone Tedeschi, Johan Bos, Thierry Declerck, Jan Hajič, Daniel Hershcovich, Eduard Hovy, Alexander Koller, Simon Krek, Steven Schockaert, Rico Sennrich, Ekaterina Shutova, and Roberto Navigli. What's the meaning of superhuman performance in today's NLU? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12471–12491, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.697. URL https://aclanthology.org/2023.acl-long.697.

Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. Large language models can accurately predict searcher preferences. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1930–1940, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657707. URL https://doi.org/10.1145/3626772.3657707.

Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 315–323, 1998. Association for Computing Machinery. ISBN 1581130155. doi: 10.1145/290941.291017. URL https://doi.org/10.1145/290941.291017.

Ellen M. Voorhees. The philosophy of information retrieval evaluation. In *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems (CLEF)*, page 355–370, 2001. Springer-Verlag. ISBN 3540440429. URL https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=151546.

Ellen M. Voorhees and Donna K. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2005.

Ellen M. Voorhees, Ian Soboroff, and Jimmy Lin. Can old TREC collections reliably evaluate modern neural retrieval models? *CoRR*, abs/2201.11086, 2022. URL https://arxiv.org/abs/2201.11086.

Emine Yilmaz, Javed A. Aslam, and Stephen Robertson. A new rank correlation coefficient for information retrieval. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 587–594, 2008. Association for Computing Machinery. ISBN 9781605581644. doi: 10.1145/1390334.1390435. URL https://doi.org/10.1145/1390334.1390435.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 307–314, 1998. Association for Computing Machinery. ISBN 1581130155. doi: 10.1145/290941.291014. URL https://doi.org/10.1145/290941.291014.