LatestEval: Addressing Data Contamination in Language Model Evaluation through Dynamic and Time-Sensitive Test Construction

Yucheng Li[1], **Frank Guerin**[1], **Chenghua Lin**[2*]

[1]Department of Computer Science, University of Surrey, UK
[2]Department of Computer Science, University of Manchester, UK
{yucheng.li, f.guerin}@surey.ac.uk, chenghua.lin@manchester.ac.uk

Abstract

Data contamination in evaluation is getting increasingly prevalent with the emergence of language models pre-trained on super large, automatically crawled corpora. This problem leads to significant challenges in the accurate assessment of model capabilities and generalisations. In this paper, we propose LatestEval, an automatic method that leverages the most recent texts to create uncontaminated reading comprehension evaluations. LatestEval avoids data contamination by only using texts published within a recent time window, ensuring no overlap with the training corpora of pre-trained language models. We develop the LatestEval automated pipeline to 1) gather the latest texts; 2) identify key information, and 3) construct questions targeting the information while removing the existing answers from the context. This encourages models to infer the answers themselves based on the remaining context, rather than just copy-paste. Our experiments demonstrate that language models exhibit negligible memorisation behaviours on LatestEval as opposed to previous benchmarks, suggesting a significantly reduced risk of data contamination and leading to a more robust evaluation. Data and code are publicly available at: https://github.com/liyucheng09/LatestEval.

Introduction

Recent years have seen the ubiquity of pretrained language models in natural language processing (NLP) due to their strong performance and generalisation capability. These models are usually pre-trained on super large internet-crawled corpora. However, many widely used benchmarks are also largely constructed from web resources (Hendrycks et al. 2020), which are very likely to be unintentionally included in the pretraining stage. This leads to a major emerging issue called *data contamination*.

Recent analysis has revealed that data contamination is widespread in model evaluations (Achiam et al. 2023; Sainz et al. 2023), which greatly undermines the credibility of evaluation results (Marie 2023; Li, Guerin, and Lin 2023) and prevents fair comparisons between models (Dickson 2023). Moreover, the massive scale of training data makes decontaminating existing benchmarks extremely difficult (Kreutzer et al. 2022). For many closed models, training data is considered as trade secret and thus confidential, eliminating any possibility for the community to address contamination by decontaminating benchmarks. One potential solution to avoid contaminated evaluation is to create new test data constantly or use human evaluation (Liu, Zhang, and Liang 2023; Jacovi et al. 2023), just like how examination for human works. However, this is extremely inefficient and costly, requiring huge human efforts periodically.

In this paper, we propose LatestEval, an automatic method that leverages the most recent texts to create novel uncontaminated reading comprehension evaluations. Before starting to create the data, we conduct the *first* manual analysis of real-world Human-AI chat data on document comprehension, to identify the most frequently asked types of knowledge and thus determine the scope of LatestEval. As for data construction, LatestEval, as a reading comprehension benchmark, typically consists of three components: *passage, query, and answer*. Here we start by 1) collecting recently created texts across various time-sensitive sources as the passages. Three different sources are used in our experiments: arXiv, BBC, and GitHub. We then 2) extract key information from these texts using various tools as the answers. At last we 3) construct questions targeting the extracted information. The questions are generated automatically via template-filling or large language models. During the testing stage, we remove the extracted answers from the original context, which encourages models to infer answers through reasoning based on the remaining context rather than short-cutting via copy-paste. Overall, LatestEval avoids data contamination by using the latest materials for testing, ensuring that it does not overlap with models' pretraining data.

Experimental results show that, in contrast to existing reading comprehension benchmarks where language models show significant memorisation, models exhibit negligible memorisation behaviour on LatestEval. Additional human evaluation and performance experiments further demonstrate LatestEval is reliable and effective in benchmarking state-of-the-art language models.

Data Contamination

What is data contamination? Data contamination refers to the phenomenon that examples from the evaluation set are also found in the training data. This might lead to the eval-

*Corresponding author

uation failing to accurately reflect models' capabilities, as models can cheat by memorising instead of learning to generalise. There are two primary types of data contamination (Dodge et al. 2021): (i) *input-only contamination* refers to only the input appearing in the pretraining corpus, and (ii) *input-and-label contamination* is when both inputs and their labels are present. The latter is generally more problematic, as models can directly memorise input-output pairs. However, the former can still cause issues as models may gain an advantage by learning extra information from the context (Li, Guerin, and Lin 2023).

How common is data contamination? Data contamination appears to be quite widespread across commonly used NLP benchmark datasets based on findings from recent studies. Dodge et al. (2021) revealed exact match contamination rates ranging from under 2% to over 50% on various GLUE benchmarks when compared to the C4 pretraining data. The GPT-3 study (Brown et al. 2020) found over 90% of examples in Quac, SQuADv2, and DROP were flagged as contaminated. FLAN (Wei et al. 2021) evaluations identified 7 out of 26 datasets exhibiting a serious contamination ratio of 50% and over. LLaMA 2 (Touvron et al. 2023) reported over 16% of MMLU examples are contaminated and about 11% are seriously contaminated (more than 80% token leakage). GPT-4 (Achiam et al. 2023) uses academic exams instead of NLP benchmarks for model evaluation. While 4 out of 34 exams are found to have zero contamination (e.g., Leetcode and Bar Exam), 9 out of 34 showed over 20% of instances are marked as dirty examples. Li, Guerin, and Lin (2023) illustrate varying level of contamination ranging from 2% to 45% in MMLU, C-Eval (Huang et al. 2023), Hellaswag (Zellers et al. 2019) and other popular QA benchmarks.

How to identify data contamination? Dodge et al. (2021) takes a straightforward approach to detect exact matches between test set examples and the pretraining data after normalising for capitalisation and punctuation. The *exact match here* means the entire input of an evaluation text is found in the training data. The GPT-3 paper (Brown et al. 2020) uses n-gram overlap to identify contamination, treating any examples with 13-gram co-occurrence in both test sets and training data as dirty examples. LLaMA-2 matches on tokenized prompts and takes a bottom-up, token-level approach to identify contamination. Overall, existing approaches usually use substring matching between evaluation examples and training data to identify data contamination. However, if we have no access to the training data, which is often the case for most recent closed models, it is extremely difficult to reveal contamination by observing the models themselves. Pioneering studies propose to identify benchmark data contamination by using search engine (Li, Guerin, and Lin 2023), measuring perplexity of test examples (Li 2023), or asking models to reconstruct test examples verbatim (Carlini et al. 2021, 2022).

To what extent does data contamination affect model evaluation? While contaminated data can potentially inflate scores, models do not necessarily perform worse on clean subsets or better on dirty subsets across all datasets. The degree of impact likely depends on factors like the dataset characteristics, model scale, and nature of the pretraining data. For instance, GPT-3 (Brown et al. 2020) showed a small 1-2% performance drop on clean subsets for PIQA and ReCoRD, compared to a significant 6% drop on clean set of SQuAD as 94% of its test examples were contaminated. The LLaMA model (Touvron et al. 2023) did not show significant gaps between clean and dirty subset performance. On HellaSwag, LLaMA's 70B model showed a 15.3 point gap between clean (63.5) and dirty (78.8) subsets. Li, Guerin, and Lin (2023) reveal contamination can inflate benchmark score even if the contamination does not give away the answer (i.e., *input-only contamination*). It also find larger models obtain more advantages from data contamination due to their more powerful memorisation.

Benchmarks for Language Models

Clean and robust benchmarks are the key to guide further progress of various models in NLP. Popular benchmarks used to evaluate large language models include:

- Comprehensive: MMLU, Big Bench Hard, AGI Eval
- Commonsense reasoning: PIQA, SIQA, HellaSwag, WinoGrande, ARC, OpenBookQA, CommonsenseQA
- World knowledge: NaturalQuestions, TriviaQA
- Reading comprehension: SQuAD, QuAC, BoolQ
- Math: GSM8K, MATH
- Code: HumanEval, MBPP

where most of them are collected from freely available online sources, which makes them very susceptible to data contamination. For instance, BoolQ (Clark et al. 2019) heavily relies on Wikipedia articles in their instances construct, which leads to a significant contamination rate of 60% as shown in (Brown et al. 2020) because Wikipedia serves as a key part of GPT-3's pretraining data.

Jacovi et al. (2023) propose three strategies to alleviate data contamination issues including encrypting test data, refusing derivative distribution, avoiding using internet data that appears with its solution, etc. Our method introduces a novel possibility motivated by examination for humans, where new tests are created dynamically to avoid cheating. Via updating periodically, LatestEval ensures to use of only the most recent texts on the web that were created after any training data were constructed, inherently mitigating the contamination risks.

The Scope of LatestEval

To determine the scope of LatestEval and make it a meaningful and realistic benchmark, we conduct the first manual analysis on *real-world conversations* between users and AI assistants chatting about documents. This analysis is to reveal which types of information humans find most essential when comprehending texts. We greatly thank `paperpersichat.tech` for providing real world conversation data about user interaction with chatbots aiming to understand academic papers. We sample 1000 real user queries and manually categorise their intention. We identified six major query categories as illustrated in Table 1.

As shown in the table, queries requesting explanations or details are dominant, highlighting the importance of models'

Query Type	Frequency	Examples
Asking for explanation, details or comparison	471	Can you provide more information on how GANs jointly optimize both privacy and utility? Do both approaches improve with more training data? What is the Accelerated Failure Time (AFT) model?
Asking for summary, or key insight	272	What is this paper about? What is the main conclusion of the analysis section?
Asking for reason, purpose or benefit	144	What is the purpose of the two-level model? What is the role of landmarks in the proposed method? How does RANSAC contribute to the multispectral photometric stereo method?
Asking for examples or demonstrations	48	Can you give an example in terms of how action-value functions works Can you explain what is Async-cloud update with an example?
Asking for future prediction	40	What are the potential application of the MIPS method? How could highway connection be used in further architecture design?
Asking for whether something is presented	21	Did the authors mentioned any rule-based method? Have the authors included SeqGAN in their experiments?

Table 1: Six most frequent categories of user query to comprehend papers.

capability to locate definitions and elaborate key information to aid user comprehension. Summary requests were also very prevalent, suggesting summarising skills are also critical for reading comprehension. There are also queries focusing on purpose analysis, giving examples, making future predictions, and judging whether something is presented. Note that although this finding is based on conversations about papers, we believe these categories also generalise to other sources in LatestEval i.e., news articles and GitHub readme documents. As a result, we focus LatestEval on the most frequent *five* categories of the query to align with practical needs: 1) terminology explanations and comparison; 2) summarisation; 3) finding the purpose; 4) providing examples; 5) predicting about the future. As a result, LatestEval will extract these corresponding types of key information to allow construction questions that assess models' abilities on these aspects.

LatestEval Pipeline

LatestEval is a *dynamic* reading comprehension benchmark, similar to SQuAD (Rajpurkar, Jia, and Liang 2018) and BoolQ, where each test instance is composed of three essential components: a passage, a query, and an answer. Models are required to answer the query based on the passage given. The LatestEval involves a three-stage pipeline: (i) the collection of the most recent texts from the web as the passages, (ii) the extraction of key information as answers, and (iii) the generation of questions corresponding to answers.

Collecting Latest Created Texts

We collect our passages from three different sources: arXiv, BBC, and GitHub. These three sources are continually updated providing real-time created content, which can effectively avoid data contamination in evaluation. In addition, they cover a wide range of topics and areas, offering both formal and informal language usage. This allows a diversified evaluation of model capabilities. At last, their popularity can ensure a sufficient volume of fresh content within even a limited time window.

To collect texts from arXiv, we request latex source of papers via its web API[1], followed by latex parser `pylatexenc` to extract plain text from `.tex` files. We collect news articles that appear on the front page of bbc.com to make sure they are fresh. GitHub repositories are open-source programming project, which normally has a `readme.md` file as a manual. We take the `readme` files as the passages of read comprehension tests. We use GitHub API[2] to obtain the latest created repositories and their `readme` files, but filter out repositories with no or very short (less than 100 words) readme files.

Considering the context window of current language models, we choose to set 1800 words as the max length of LatestEval instances. Therefore, we use only the first section (usually introduction) for arXiv papers, and filter out texts collected from BBC and GitHub that are longer than 1800 words. For a time span of 7 days from 01/07/2023 to 08/07/2023, we have collected 2,150 valid papers from arXiv, 525 articles from BBC, and 951 `readme` files from GitHub that meet our requirements. This demonstrates that there is sufficient content for the next steps.

Answer Construction

To construct meaningful questions for reading comprehension, we need to identify key pieces of information from the collected passages that can serve as the answers. Based on

[1]http://export.arxiv.org/api/query

[2]https://api.github.com/search/repositories

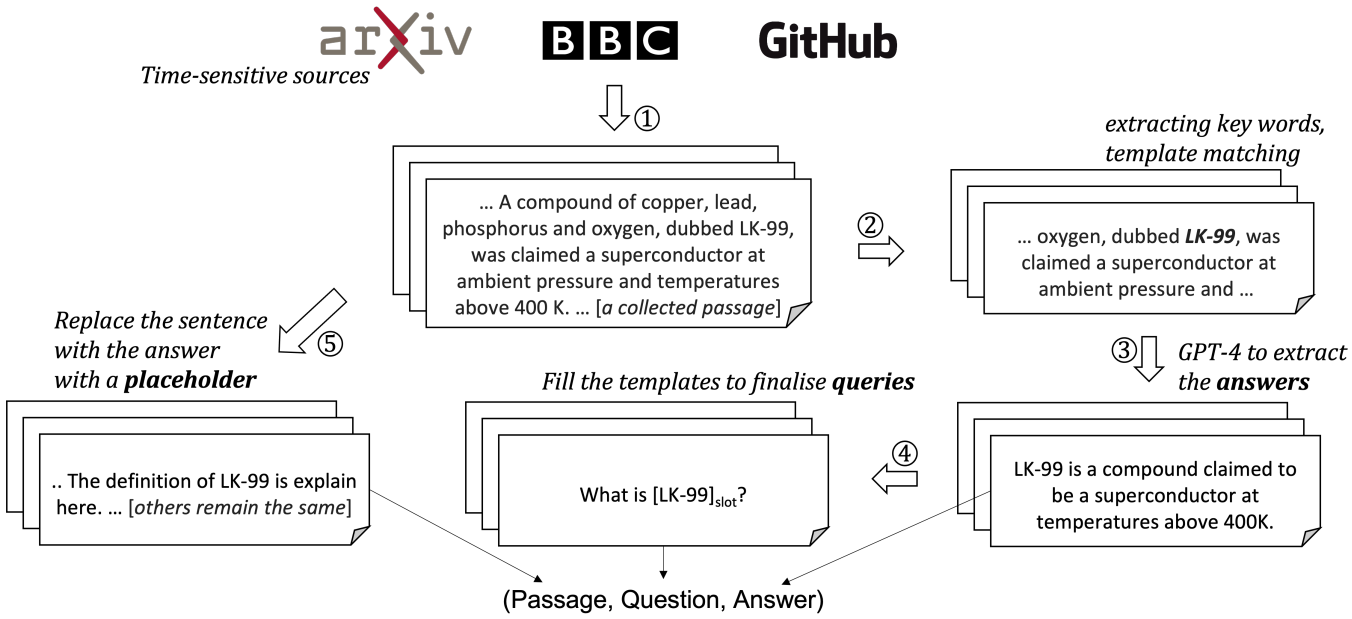


Figure 1: The overall pipeline of LatestEval. Step 1 is for collecting the latest texts; 2,3 are to construct the answers; 4 is to construct corresponding queries; and 5 is to prepare the passages.

the scope analysis above, we focus on extracting the five major types of key information.

Terminology explanations. We employ a two-step strategy to extract terminology explanations and comparisons from the input passage: keyword extraction + large language models. We first process the input passage via the KeyBERT (Grootendorst 2020) algorithm to extract candidate keywords and phrases. For passages exceeding 512 tokens (max length of BERT), we use `gpt-3.5-turbo`[3] to extract key terms. Then, we take the key terms and the first few sentences containing the terms as input, and ask GPT-4[4] to identify their definition from these sentences. Here we only consider terms with an explicit definition in the passage, then we ask GPT-4 to extract the definition as the target answers.

Summarisation. We use template-matching to directly identify summative content from passages. For papers, we extract summative content indicated with the *abstract* latex environment, key findings at the end of the introduction section, and sentences start with *In summary*. For news, we use *Tl;dr* as the indicator or take the title plus article description (written by editors) as the summary. For GitHub repositories, we take the repository description plus the first few lines of the readme file as the summary of the project.

Others. We use a pipeline consisting of phrase-matching plus GPT-4 post-processing to construct answers for purpose, examples, and future prediction types of queries. We first use phrase matching to locate these answers in the given passage. For instance, we locate answers for purpose types of queries by retrieving sentences containing the following phrases: *because*, *aim to*, *allow .. to* , *contribute to*, *lead to*, and *motivation*. For examples types of queries, we choose phrases of *for example*, *e.g.,*, *such as*. Note that we notice examples in GitHub readme files are usually involving code generation. We do not consider providing examples type of query for the source of Git Hub. For queries about making future predictions, we use phrases of *future works*, *Forecasts show*, *upcoming features*. Then we ask GPT-4 to extract the analysis about purposes as the target answers.

After extracting answers from the passages, we replace sentences containing these answers with placeholder sentences. For instance, we replace the sentence *"We chose to look at loss because it tends to be less noisy than other measures."* with *"The reason to look at loss is explained here."* Here the placeholder sentence is obtained by GPT-4 and template filling. We do this replacement to encourage models to infer answers instead of copy-past.

Query Construction

We propose a hybrid approach with templates and GPT-4 to construct natural queries targeting the extracted answers in the previous steps. For the summarisation queries, we directly apply simple templates like "*what is this passage about?*", "*what are the main points raised in this article?*", which already do a good job in providing clear questions asking for analysis main points. For other question types, we generate questions by providing GPT-4 with the extracted answers and template candidates and asking it to produce questions by filling the templates with appropriate terms. First, we manually create 3-5 templates for each query type. We then feed GPT-4 the answer text along with the template options. GPT-4 selects the most suitable template and fills it with appropriate terms or phrases from the answer, generat-

[3]https://platform.openai.com/examples/default-keywords

[4]GPT-4 is a closed source model, you can access it via OpenAI API. Check https://platform.openai.com/docs/guides/gpt

	Arxiv	BBC	Github	ALL
Terminology	1792	98	825	2715
Summary	570	504	466	1540
Purpose	793	565	421	1779
Example	421	16	113	550
Future	147	105	63	315
ALL	3723	1288	1888	6899

Table 2: Statistics of LatestEval, July week 1 2023.

ing a custom question targeting the specific information.

Examples and Statistics

Here we present the statistics of LatestEval on time slice of week 1 July 2023 as we use this version in our experiments section. In addition, we show the prompts we use in the pipeline of LatestEval, and show some of the templates we use in the query construction. As shown in Table 2, we have constructed 6899 reading comprehension tests in total from 500 passages for each of the three sources, and 3723, 1288, and 1888 tests from arXiv, BBC and GitHub, respectively. We do not use all generated data in our experiments, we sample 3k examples based on the distribution of question types we find in the real-world data analysis, i.e., Table 1. We have shown an example of how the overall procedure for a test case construction in Figure 1. We explain the prompt we use in steps 3 and 5, as GPT-4 is heavily used in these two steps. In step 3, we use the prompt: *"*{*relevant_sentences*}*. Please extract the definition of the term* {*term*} *that the above sentences explicitly defined. You must just do copy-pasting and be faithful to the given passage."*. The prompt in step 5 is *"*{*the_query*}*,* {*relevant_sentence*}*. Based on the above given information, generate a placeholder sentence considering the following examples* {*placeholder_examples*}*."*

Experiments

We performed three experiments to evaluate our LatestEval: contamination test, performance test, and human evaluation.

Contamination Test

First, we conduct a contamination test on common reading comprehension benchmarks to measure the extent that models memorise their test instances during pretraining. Although most existing analyses identify contamination by computing the overlapping between training and test sets, we cannot use the same approach as the pretraining data of current language models are mostly not publicly available. Therefore, we propose a novel method to quantify test contamination by observing whether models exhibit memorisation behaviour on test instances, which can serve as a strong signal of test contamination. Carlini et al. (2021, 2022) have defined the "memorisation" of models that a sequence is considered as memorised if the model has considerably smaller perplexity on that sequence. The idea is that sequences leak in the training data will tend to have lower perplexity (i.e., higher likelihood) than sequence models never seen before. This is well demonstrated in Figure 2, where we compare the perplexity of two Wikipedia subsets: the first is `Wikitext` (Merity et al. 2016) which is widely used in language model pretraining, the second is `NewWiki`, latest Wikipedia texts that created after April 2023, after all tested models were released. We find that Wikipedia content learned during pretraining has a much lower perplexity than recently created Wikipedia content.

Now we compute the perplexity of the validation set on the three existing reading comprehension benchmarks SQuADv2 (Rajpurkar, Jia, and Liang 2018), BoolQ (Clark et al. 2019), QuAC (Choi et al. 2018) and compare them to LatestEval. We involve five foundation models here: `opt-350m`, `opt-1.3b` (Zhang et al. 2022), `Llama-7b`, `Llama-13b`, and `Llama-30b` (Touvron et al. 2023). OPT models are run with fp32 and others are run with fp16. We make sure the input texts of each benchmark have the same length so that we can obtain a fair comparison.

Results. As shown in Figure 2, the perplexities of BoolQ and QuAC are very close to `wikitext` and are significantly lower than freshly created Wiki content across all language models. This reflects clear memorisation of language models on test examples in these two benchmarks, and thus can be considered as a signal of data contamination. The perplexities of SQuADv2 are divergent, which has a rather high perplexity and is close to freshly created Wiki texts on small models such as `llama-7b` and `opt`s. The perplexity decreases significantly on larger models such as `llama-13b,30b`. Based on this observation, we believe test examples in SQuADv2 were not memorised by small models but the extent of being contaminated increases with model scale. At last, we find a clear trend that LatestEval constantly has a perplexity very close to or even higher than fresh Wiki text, which shows that models have no prior knowledge of our constructed benchmark and thus have considerably less contamination risk.

In Figure 3, we also show how `GPT-4` memorise benchmark test examples by real cases. We prompt the model with a prefix and check whether they are able to reproduce the test examples verbatim, following the setting in (Carlini et al. 2022). We find BoolQ and QuAC are heavily contaminated, followed by SQuADv2 that is also partly affected. LatestEval tends to not be affected by the contamination issue. The results here well align with our finding in the perplexity-based analysis (Figure 2).

Performance Test

We test some leading large language models using our LatestEval. We include both foundation models as well as language models after supervised fine-tuning and human feedback reinforcement learning, including `GPT-3.5-turbo`[5], `GPT-4`, `llama-13b`, `llama-30b`, and `vicuna-13b` (Chiang et al. 2023). We are aware that the `llama-2` series has been recently released. However, our experiments were based on the July week 1 version of LatestEval. Given what `llama-2` has reported that their

[5]also known as ChatGPT

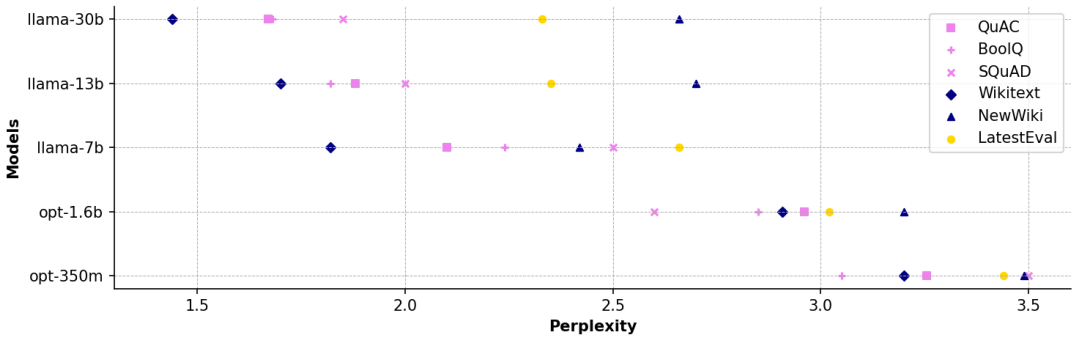


Figure 2: The comparison of datasets' perplexities indicates the contamination extent on various language models.

squad

Beyoncé Giselle Knowles–Carter (/biːˈjɒnseɪ/ bee–YON–say) (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in

the late 1990s as lead singer of R&B girl–group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best–selling girl groups of all time. Their hiatus saw the release of Beyoncé's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100

boolq

Persian (/ˈpɜːrʒən, –ʃən/), also known by its endonym Farsi (فارسی fārsi (fɒrˈsiː) (listen)), is one of the Western Iranian languages within the Indo–Iranian branch of the Indo–European language family. It is primarily

spoken in Iran, Afghanistan (officially known as Dari since 1958),[3] and Tajikistan (officially known as Tajiki since the Soviet era),[4] and some other regions which historically were Persianate societies and considered part of Greater Iran. It is written right to left in the Persian alphabet, a modified variant of the Arabic script.

quac

The number of Malayalam speakers in Lakshadweep is 51,100, which is only 0.15% of the total number, but is as much as about 84% of the population of Lakshadweep. In all, Malayalis made up 3.22% of the total Indian population in 2001. Of the total 33,066,

392 Malayalam speakers in India in 2001, 33,015,420 spoke the standard dialects, 19,643 spoke the Yerava dialect and 31,329 spoke non–standard regional variations like Eranadan. As per the 1991 census data, 28.85% of all Malayalam speakers in India spoke a second language and 19.64% of the total knew three or more languages.

LatestEval

The British have a love–hate relationship with the NHS. According to researchers at the King's Fund, the public gave the NHS its worst rating since records began 40 years ago. Just 29% said they were satisfied with the NHS in 2022. And yet we still love it. A whopping 90% of

of Britons believe that the NHS is a crucial institution that should be preserved. The juxtaposition between dissatisfaction with the current state and overall reverence for the institution speaks volumes about the complex relationship the British public has with their healthcare system.

Figure 3: Memorisation test of `GPT-4` model on four benchmarks. Coloured text refers to the text generated by `GPT-4` that matches the original test text. The four examples shown are just the first instance of each benchmark, so no cherry picking.

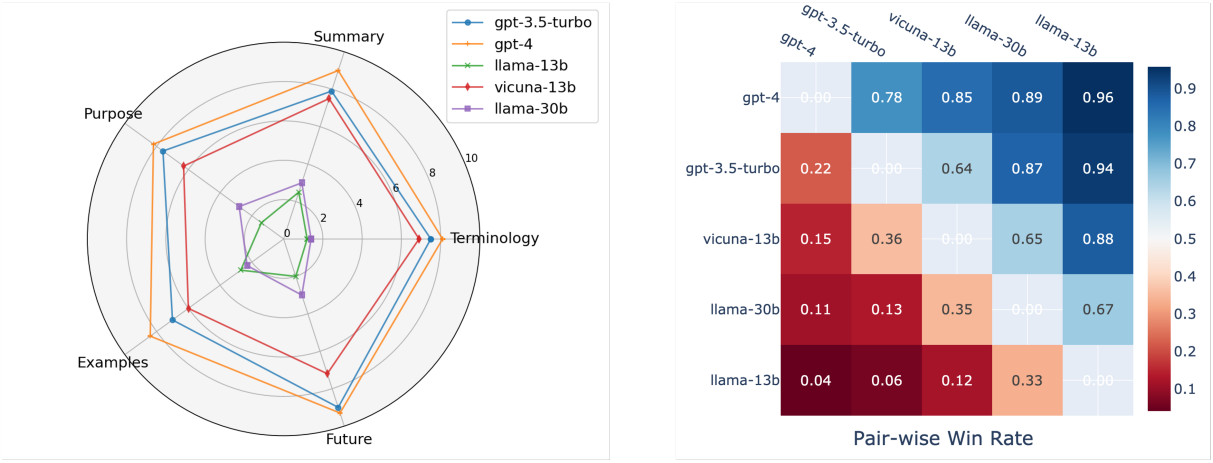
models incorporated data from July 2023 in their training, including `llama-2` would not ensure a fair comparison since they might have been exposed to passages from LatestEval. We will definitely add the `llama-2` series in future evaluations.

Instead of relying on n-gram based metrics such as BLEU and ROUGE to compare models' answers to reference answers, we utilise a LLM-as-a-judge method (Zheng et al. 2023) which has proven to be effective and robust for evaluating models' generations. LLM-as-a-judge provides two grading systems, where the first is single-answer grading and the second is pair-wise win rate. We report the results of both and visualise the results with their software. We adopt the prompt template from InstructGPT (Ouyang et al. 2022) designed for reading comprehension test and test models on a zero-shot basis. Due to the monthly limit of `GPT-4` quota, we do not use all pair-wise data, instead, we construct 6k pair-wise data to calculate the pair-wise win rate.

As shown in Figure 4, LatestEval is effective in differentiating large language models. The pair-wise win rate show that `GPT-4`, and `gpt-3.5-turbo` beat other models significantly. Foundation models without further tuning such as `llama-13b` and `llama-30b` hardly generate reasonable outputs, and thus perform rather poorly. Compared to previous reading comprehension benchmarks, LatestEval enables a fine-grained analysis on five categories of questions. `GPT-4` demonstrates better performance across all types of queries. `Vicuna` is good at summary and terminology explanation, but less competitive on providing future predictions and examples. In addition, we find models struggle most with terminology explanation and identifying purpose, which indicates room for future improvement.

Human Evaluation

To further validate the robustness of LatestEval, we conducted human evaluations. LatestEval, due to its automatic construction nature, can potentially lead to three issues: 1) as the use of GPT-4 for extracting answers and filling query templates, we shall evaluate how faithful GPT-4 can be to the given passage and measure how reliable are the extracted answers; 2) to encourage reasoning instead of copy-pasting, we delete the explicit answers from the original passage, which might lead to unanswerable questions; 3) even we have deleted the existing answers, the answers might be mentioned multiples times in the passage, so we also evaluate whether models can cheat by finding the answers explicitly somewhere else in the input after the answer deletion. Our human evaluation is to measure and analyse the above



(a) (b)

Figure 4: (a) : single answer scores across five types of queries; (b) : pair-wise win rate, y-axis indicates the winner.

	Num.	unfaithful	unanswerable	copyable
Arxiv	402	12	22	14
BBC	220	7	13	2
Github	161	17	11	1
Term.	309	18	27	6
Summary	190	0	0	5
Purpose	187	13	19	6
Example	62	3	0	0
Future	35	2	0	0

Table 3: Human evaluation on the faithfulness, answerability, and copyability of LatestEval.

three aspects. We sample 200 documents from LatestEval July week 1 and analyse their corresponding 783 reading comprehension pairs. Five annotators from Amazon Turk are employed and each test example is annotated at least three times on three dimensions: *faithfulness*, *answerability*, and *copyability*.

Results. We show our human evaluation results in Table 3, where we analyse the number of unfaithful, unanswerable, and copyable cases for each source and each category of question-answer pair. First, we have identified the faithful issues among all three sources, affecting 4.5% of examples in total. The issue of faithfulness is mainly from to the answer construction process, which usually happens when GPT-4 is unintended to use its inner knowledge instead of the given context to extract the answers. For example, in *"It has many obvious applications for outdoor scene understanding, from city mapping to forest management."*, the authors explain the potential application of the techniques. But GPT-4 rephrases or adds new things based on its inner knowledge, which we cannot ensure correctness and thus are not expected: *"... potential applications include urban planning, autonomous vehicles, environmental protection, etc."* From the view of query categories, faithfulness issues are mainly from Terminology explanation and Purpose analysis questions. To avoid model solving questions by just copy-pasting, we delete the explicit answers and ask models to infer the answer from the context. But this approach can somehow lead to unanswerable questions. We discovered unanswerable questions among all sources and two types of questions that affect 5.7% of questions. However, an interesting finding is that large language models can guess the correct answers even the questions are flagged as unanswerable, e.g. guessing the meaning of a term even the given content has no relevance to the term at all. At last, we analyse whether the model can search and copy the answer from somewhere in the given context, which prevents the reasoning of models. We have detected 2.1% of examples affected by the copy issue, in terminology, summary, and purpose types of queries.

Based on the above analysis, we believe LatestEval could benefit from 1) a post-processing stage with quality assessment; and 2) use few-shot setting in data construction instead of the current zero-shot setting. By involving a quality assessment procedure, LatestEval could conduct a self-review procedure for each example and then filter out uncertain cases. We could easily customise the quality assessment procedure to further adjust the benchmark. In addition, including demonstrations in the input prompt might enhance the overall robustness of data construction, which can be useful in mitigating the above-mentioned issues. We leave them for future study.

Conclusion

This paper proposes LatestEval, an automatic test construction pipeline that utilises the latest materials to avoid data contamination in language model evaluation. To determine the scope of LatestEval, we conduct the first analysis of real-world human-AI conversational data to find the most frequently asked queries. We also propose a novel perplexity-based approach to evaluate the extent of benchmark contamination, and find popular language models exhibit negligible memorisation on LatestEval.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Carlini, N.; Ippolito, D.; Jagielski, M.; Lee, K.; Tramer, F.; and Zhang, C. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, U.; et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2633–2650.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/. Accessed: 2024-01-15.

Choi, E.; He, H.; Iyyer, M.; Yatskar, M.; Yih, W.-t.; Choi, Y.; Liang, P.; and Zettlemoyer, L. 2018. QuAC: Question answering in context. *arXiv preprint arXiv:1808.07036*.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Dickson, B. 2023. Why data contamination is a big issue for LLMs. https://bdtechtalks.com/2023/07/17/llm-data-contamination/. Accessed: 2023-07-28.

Dodge, J.; Sap, M.; Marasović, A.; Agnew, W.; Ilharco, G.; Groeneveld, D.; Mitchell, M.; and Gardner, M. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *arXiv preprint arXiv:2104.08758*.

Grootendorst, M. 2020. KeyBERT: Minimal keyword extraction with BERT. https://github.com/MaartenGr/KeyBERT. Accessed: 2024-01-15.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Huang, Y.; Bai, Y.; Zhu, Z.; Zhang, J.; Zhang, J.; Su, T.; Liu, J.; Lv, C.; Zhang, Y.; Lei, J.; Fu, Y.; Sun, M.; and He, J. 2023. C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models. *arXiv preprint arXiv:2305.08322*.

Jacovi, A.; Caciularu, A.; Goldman, O.; and Goldberg, Y. 2023. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. *arXiv preprint arXiv:2305.10160*.

Kreutzer, J.; Caswell, I.; Wang, L.; Wahab, A.; van Esch, D.; Ulzii-Orshikh, N.; Tapo, A.; Subramani, N.; Sokolov, A.; Sikasote, C.; et al. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10: 50–72.

Li, Y. 2023. Estimating Contamination via Perplexity: Quantifying Memorisation in Language Model Evaluation. *arXiv preprint arXiv:2309.10677*.

Li, Y.; Dong, B.; Lin, C.; and Guerin, F. 2023. Compressing Context to Enhance Inference Efficiency of Large Language Models. *arXiv preprint arXiv:2310.06201*.

Li, Y.; Guerin, F.; and Lin, C. 2023. An Open Source Data Contamination Report for Large Language Models. *arXiv preprint arXiv:2310.17589*.

Li, Y.; Guo, Y.; Guerin, F.; and Lin, C. 2024. Evaluating Large Language Models for Generalization and Robustness via Data Compression. *arXiv preprint arXiv:2402.00861*.

Liu, N. F.; Zhang, T.; and Liang, P. 2023. Evaluating verifiability in generative search engines. *arXiv preprint arXiv:2304.09848*.

Marie, B. 2023. The Decontaminated Evaluation of GPT-4. https://benjaminmarie.com/the-decontaminated-evaluation-of-gpt-4/. Accessed: 2023-07-28.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822*.

Sainz, O.; Campos, J. A.; García-Ferrero, I.; Etxaniz, J.; de Lacalle, O. L.; and Agirre, E. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. *arXiv preprint arXiv:2310.18018*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*.