Generating Benchmarks for Factuality Evaluation of Language Models

Dor Muhlgay* Ori Ram Inbal Magar Yoav Levine Nir Ratner
Yonatan Belinkov Omri Abend Kevin Leyton-Brown Amnon Shashua Yoav Shoham

AI21 Labs

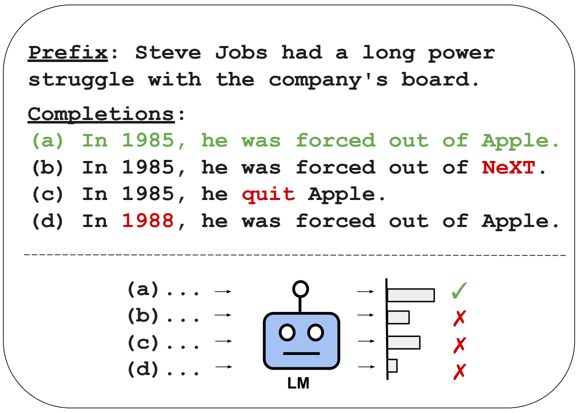
Abstract

Before deploying a language model (LM) within a given domain, it is important to measure its tendency to generate factually incorrect information in that domain. Existing methods for factuality evaluation of LLM generation focus on facts sampled from the LM itself, and thus do not control the set of evaluated facts and might under-represent domain specific or rare facts. We propose FACTOR: Factual Assessment via Corpus TransfORmation, a scalable approach for evaluating LM factuality. FACTOR automatically transforms a factual corpus of interest into a benchmark evaluating an LM's propensity to generate true facts from the corpus vs. similar but incorrect statements. We use our framework to create three benchmarks: *Wiki-FACTOR*, *News-FACTOR* and *Expert-FACTOR*. We show that: (i) our benchmark scores increase with model size and improve when the LM is augmented with retrieval; (ii) benchmark score and perplexity do not always agree on model ranking; (iii) when perplexity and benchmark score disagree, the latter better reflects factuality in open-ended generation, as measured by human annotators. We make our data and code publicly available[1].

1 Introduction

Despite rapid improvements in their capabilities, large Language Models (LMs) still tend to generate factually inaccurate or erroneous text (Lin et al., 2022; Maynez et al., 2020; Huang et al., 2020). Such phenomena can pose a significant hurdle to deploying LMs in important or sensitive settings, motivating the development of methods for evaluating LM factuality in open-ended generation.

Methods for directly evaluating an LM's propensity towards factual generation were recently proposed by Lee et al. (2022) and Min et al. (2023). These methods suggest sampling generations from a model, applying an automatic pipeline for fact verification, and then assigning a score corresponding to the percentage of factually correct generated statements. In task-specific domains, such as long-form question answering, evaluation is usually done by assessing the relevance of a sampled generation against a reference text (Lin, 2004; Fabbri et al., 2022). However, the sampling approach may introduce bias: by scoring the accuracy of facts that an LM tends to generate in an open-ended setting, high-likelihood facts are over-represented, while the "long-tail" of rare facts is under-represented.



Prefix: Steve Jobs had a long power struggle with the company's board.

Completions:

(a) In 1985, he was forced out of Apple.
(b) In 1985, he was forced out of NeXT.
(c) In 1985, he quit Apple.
(d) In 1988, he was forced out of Apple.

Figure 1: Each example in our evaluation task (dubbed FACTOR) consists of a *prefix* and four *completions*, of which only one is factually correct (completion (a) in this example). The non-factual completions (b), (c) and (d), marked in red, are generated according to different factual error types, detailed in Table 1. The evaluated model assigns likelihood scores to each completion separately. It is considered "correct" if it assigns the highest likelihood to the factually correct completion over all non-factual alternatives.

Currently, there are no metrics suited to measuring LM factuality with respect to a *controlled set of facts* in a *generation setting*. A common proxy is measuring LM perplexity; this was widely adopted to evaluate retrieval-augmented LMs (Khandelwal et al., 2020; Borgeaud et al., 2022; Ram et al., 2023; Shi et al., 2023). However, perplexity is affected

*Corresponding author: dorm@ai21.com

[1] https://github.com/AI21Labs/factor

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 49–66
March 17-22, 2024 ©2024 Association for Computational Linguistics

by many linguistic phenomena, and so cannot be directly linked to factuality.

This paper introduces a novel framework for testing a model's tendency to generate factual information from a given factual corpus: Factual Assessment via Corpus TransfORmation (*FACTOR*). The key idea is automatically perturbing factual statements taken from the corpus to create a constant number of similar but false variations for each true statement (Figure 1). We employed InstructGPT (Ouyang et al., 2022) to generate the false variations for each true statement. The LM's FACTOR accuracy on our benchmark is defined as the percentage of examples for which it assigns higher likelihood to the factual completion than to any of the false variations.

We applied FACTOR to the Wikipedia and News domains, as well as to a diverse collection of domain specific question-answer pairs (*e.g.*, medicine, technology, law); constructing new benchmarks dubbed *Wiki-FACTOR*, *News-FACTOR* and *Expert-FACTOR*. We used these datasets to evaluate a large suite of LMs from the OPT (Zhang et al., 2022), GPT-2 (Radford et al., 2019), and GPT-Neo (Black et al., 2021) families, ranging from 110M to 66B parameters. We show in §5.1 that, as expected, FACTOR scores increase with model size. However, even the largest models we evaluated achieved scores of only 58% for Wiki-FACTOR, 68% for News-FACTOR, and 55% for Expert-FACTOR, indicating that these benchmarks are challenging even for large LMs. In §5.2 we show that consistent FACTOR score improvements can be achieved by augmenting the LMs with the simple retrieval component used by Ram et al. (2023). This directly demonstrates that retrieval augmentation improves factuality in the LM setting; FACTOR is thus posed as a prominent approach for measuring retrieval-augmented LMs.

We further show that FACTOR accuracy and LM perplexity are correlted but can sometime induce different orderings between LMs (§5.3). This highlights that FACTOR and perplexity capture different aspects of the LMs' performance (see Figure 2). In §6, we report findings of a manual annotation effort over $1,200$ generated completions, which reinforces FACTOR accuracy as predictive of factuality in open-ended generation.

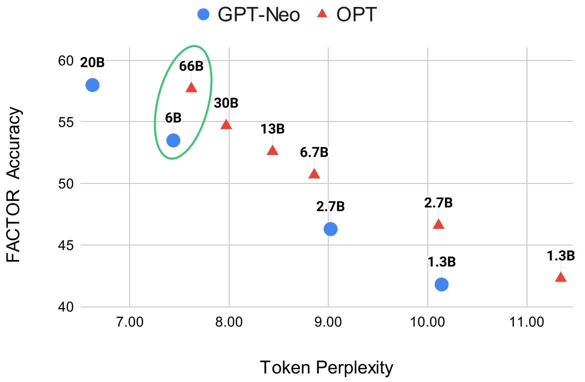


Figure 2: Wiki-FACTOR scores versus LM perplexity on Wikipedia for LMs from the GPT-Neo model family (blue circle, sizes 1.3B-20B) and the OPT model family (red triangle, 1.3B-66B). Labels indicate sizes (in billions). The two may disagree on ranking, *e.g.*, the OPT-66B LM has higher perplexity but better Wiki-FACTOR accuracy than the GPT-J-6B LM (marked in green circle). In §6 we annotate text generated out of both models and show that better Wiki-FACTOR is predictive of more factual text generation.

2 Related Work

Factuality Evaluation The subject of factuality evaluation has been extensively studied in downstream tasks such as summarization, fact-verification and dialog (Honovich et al., 2022; Huang et al., 2021; Chen et al., 2021; Tam et al., 2023). These works typically focus on *factual consistency*, evaluating whether a generated text is supported by a reference text or context (*e.g.*, source document and generated summary).

Another popular approach suggests probing LMs' internal factual knowledge by using slot filling tasks, *e.g.*, "Barack Obama was born is [MASK]" (Petroni et al., 2019, 2021; Roberts et al., 2020; Jiang et al., 2020; Elazar et al., 2021; Li et al., 2022; Zhong et al., 2021; Peng et al., 2022; Mallen et al., 2023). These works test LMs in a simplified, synthetic setting.

FACTOR differs from the above methods as it aims at evaluating factuality in a natural open-ended text generation setting. In such setting, the context may be needed to reason over the evaluated factual statement, while the factual statement may not be evident in the context (unlike summarization).

Recent works proposed scoring the factuality of free-form LM generations samples (Min et al., 2023; Lee et al., 2022). However, these approaches lack control over the evaluated facts and are biased towards common facts generated by the LM.

Contrastive Datasets Contrastive evaluation, in which a model is tested to discern between similar positive and negative examples, is widely used in various tasks (Sennrich, 2017; Burlot and Yvon, 2017; Glockner et al., 2018; Kaushik et al., 2020). For factuality evaluation, negative examples are obtained by perturbing factual claims. This is done through human annotation, rule-based or model based heuristics (Schuster et al., 2021; Liu et al., 2022; Gupta et al., 2022). Following recent works on benchmarks generation (Perez et al., 2023), we employed Instruct-GPT to generate non-factual claims, as described in the following section.

3 The FACTOR Evaluation Approach

This section outlines our proposed approach: Factual Assessment via Corpus TransfORmation, or FACTOR. Given a corpus, we define a multi-choice task where each example is comprised of a multi-sentence prefix, a single *factual* next sentence completion, and three *non-factual* alternative completions (Figure 1). In §3.1 we present several properties required of a FACTOR benchmark, and describe the error verticals along which we generate non-factual alternatives. We then explain our FACTOR dataset creation pipeline, which automatically generates a FACTOR benchmark from a given corpus (§3.2). Finally, we apply this pipeline to two corpora Wikipedia and news, and a long-form question answering dataset, creating Wiki-FACTOR, News-FACTOR and Expert-FACTOR. We verify the quality of these datasets through manual annotations against the required properties (§3.3).

3.1 The Evaluation Task: *FACTOR*

We describe the FACTOR multi-choice factual evaluation task. Each example of our task contains a prefix text t, along with four possible full sentence completions, of which only one is factually correct. We choose the original completion (*i.e.*, the continuation of t in the corpus) as the factually correct one. The correct completion is denoted as c^+, and the non-factual completions as $\mathcal{C}^- = \{c_1^-, c_2^-, c_3^-\}$. We evaluate models by measuring the percentage of examples where they assign the highest mean log-probability to c^+. Formally, a model is correct on a given example if:

$$c^+ = \operatorname*{argmax}_{c \in \{c^+\} \cup \mathcal{C}^-} \frac{\log p(c|t)}{|c|}, \qquad (1)$$

where $|c|$ is the length of completion c in tokens. We refer to the percentage of correct examples as the FACTOR accuracy.

We require each of the "incorrect" completions $c^- \in \mathcal{C}^-$ to satisfy the following properties:

1. *Non-factuality*: c^- contains a false claim;
2. *Fluency*: c^- is grammatical;
3. *Similarity to the factual completion*: c^- has a small edit-distance from c^+.

The second and third properties make it harder to distinguish between the factual and non-factual completions for reasons other than their factual correctness, such as fluency or style. Furthermore, it is desirable that the non-factual completions be logical and self-consistent, to make them more difficult to eliminate. For example, modifying $c^+ =$*"They got married in 2010 and divorced in* ***2017****"* by changing *2017* to *2009*, results in a non-factual completion which can be discarded by knowing the temporal relation between marriage and divorce.

Error Types Non-factual completions in a FACTOR dataset should cover diverse factuality error types. To do so, we adopt the error typology introduced in FRANK (Pagnoni et al., 2021). While they introduced their error typology to categorize factual inconsistencies of generated summaries w.r.t. the source document, we instead leverage this typology to vary the type of factual inconsistencies that hold between non-factual completions and the prefix and completion (t and c^+). We focus on the five error types from two error categories: semantic frame and discourse (examples in Table 1):

- *Predicate* error: a predicate that is inconsistent with c^+ or t.
- *Entity* error: The subject or object of a predicate are inconsistent with c^+ or t.
- *Circumstance* error: The completion contains information describing the circumstance of a predicate (*e.g.*, location, time, manner) that is inconsistent with c^+ or t.
- *Coreference* error: The contradiction is inconsistent with a pronoun/reference in c^+ or t, referring to a wrong or non-existing entity.
- *Link* error: c^- is inconsistent with c^+ or t in the way that different statements are linked together (causal/temporal links).

Original text (completion in bold)	*...In 1982, Donne was appointed as the first Queen's Representative to the Cook Islands.* ***After completing his term, he became Chief Justice of Nauru and Tuvalu in 1985.***
Error Type	**Example**
Entity	*After completing his term, he became the Queen's Representative to the Cook Islands in 1985.*
Predicate	*After completing his term, he declined the position of Chief Justice of Nauru and Tuvalu in 1985.*
Circumstance	*After completing his term, he became Chief Justice of Nauru and Tuvalu in 1987.*
Coreference	*After completing her term, she became Chief Justice of Nauru and Tuvalu in 1985.*
Link	*Before completing his term, he became Chief Justice of Nauru and Tuvalu in 1985.*

Table 1: Error types examples. The original text (top) consists of a prefix and a completion sentence (marked in bold). Each example introduce different perturbation over the original completion of different type (edit marked in red).

3.2 Generating FACTOR Benchmarks

Given an evaluation corpus, we generate a FACTOR benchmark automatically. The process is designed to meet the requirements presented in §3.1, and follows a four-stage pipeline: (1) prefix and completion selection, (2) non-factual completion generation, (3) non-factual completion filtering, and (4) non-factual completion selection.

3.2.1 Prefix and Factual Completion Selection

We select a single sentence from each document as a factual completion c^+. We exclude headlines and sentences with less than 10 words. The prefix t is the entire text preceding c^+ in the document.

3.2.2 Non-factual Completions Generation

Given a prefix t and its original completion c^+, we use InstructGPT (davinci-003; Ouyang et al. 2022) to generate a set of contradictory completions. We designed a specific prompt instructing the model to generate contradictions corresponding to each type of error.[2] We only apply each prompt to sentences that are relevant to its error type (determined through simple heuristics, see App. A.1). The prompts are designed as follows:

- Multiple contradiction generation: the model is prompted to generate multiple subsequent contradictions in each sampling operation. Preliminary experiments showed that this sampling practice improves diversity compared to multiple independent completion sampling.
- Edit planning: for each contradiction, the model first explicitly generates the planned edits over the original completion, and then applies those edits by writing the entire *modified* completion (similar to chain-of-thought prompting; Wei et al. 2022). For instance, the coreference error in Table 1 is generated by explicitly writing the edits ("Changes: 'his' to 'her'") and then the contradiction. This encourages the model to make minimal edits.

3.2.3 Non-factual Completions Filtering

We considered the set of generated completions as candidates for non-factual completions. We applied automatic tools to filter out (i) *non-contradictory* and (ii) *non-fluent* completions.

Non-Contradictory Completions Given a candidate completion c, we assert that it is indeed contradictory to the original completion c^+ by applying an NLI model.[3] The *premise* is set to be c^+ along with its near context (*i.e.*, the last tokens of the prefix t; denoted by t_{near}). The *hypothesis* is set to be c, also preceded by t_{near}. We selected generations classified as contradictory by the NLI model with a probability higher than τ_{NLI}, *i.e.*:

$$p_{\text{NLI}}(\text{contradiction} \mid [t_{\text{near}}; c^+], [t_{\text{near}}; c])) > \tau_{\text{NLI}}$$

We chose $\tau_{\text{NLI}} = 0.6$ (except for contradictions generated by the coreference error prompt, where we set $\tau_{\text{NLI}} = 0.3$) after using a manual validation process detailed App. A.2.

[2] App. D lists the full prompts for each error type.

[3] We used DeBERTa-large model (He et al., 2021) fine-tuned on the MNLI dataset (Williams et al., 2018) from Hugging Face: microsoft/deberta-large-mnli.

Property	Wiki	News	Expert
Non-factual	97.6	98.3	97.5
Fluent	94.0	97.0	96.7
Self-Consistent	87.4	87.3	83.8
Edit-Distance	2.3±(1.4)	2.1±(1.4)	4.0±(3.1)

Table 2: Validation results: percentage of generation that meet each desired property, estimated by manual annotation over sub-samples (top), and mean edit-distance between the generations and their factual completion (bottom).

Non-Fluent Completions To verify that c is a fluent completion we use GPT2-Small (Radford et al., 2019) scores, similar to Gupta et al. (2022): We filter out generations with mean log-likelihood lower than the original completion's by a fixed margin τ_{LM}. Using a manual validation, we set $\tau_{\text{LM}} = 0.2$ (see App. A.2). Formally, we selected a completion c if it satisfies:

$$\frac{\log p(c)}{|c|} > \frac{\log p(c^+)}{|c^+|} - \tau_{\text{LM}}$$

3.2.4 Non-factual Completion Selection

Finally, we select non-factual completions c_1^-, c_2^-, c_3^- from the filtered candidates. For increased error type diversity, we choose one completion per type, and repeat types only when not enough generations meet the §3.2.3's criteria.

3.3 Applying FACTOR to Knowledge Intensive Domains

We focused on three knowledge intensive domains: Wikipedia (encyclopedic knowledge), news (current events) and long-form question answering in specific domains. We constructed the following evaluation datasets:

- *Wiki-FACTOR:* based on the Wikipedia section of The Pile's validation split (Gao et al., 2021), containing 2994 examples.
- *News-FACTOR:* based on Reuters articles published after $1/10/2021$, extracted from The RefinedWeb Dataset (Penedo et al., 2023). The dataset consists of 1036 examples.
- *Expert-FACTOR:* based on the validation and test splits of ExpertQA (Malaviya et al., 2023), a long-form expert-curated question answering dataset spanning various fields, which suits the motivation of FACTOR to evaluate rare facts. Each document in the corpus is a concatenation of a question-answer pair. The dataset consists of 236 examples.

Type	Wiki	News	Expert
Predicate	25.4	31.3	47.1
Entity	42.8	48.0	38.8
Circumstance	24.2	16.0	7.1
Coreference	4.4	2.3	2.9
Link	3.2	2.3	4.2

Table 3: Annotated error type distribution for Wiki-FACTOR (Wiki), News-FACTOR (News), Expert-FACTOR (Expert).

3.3.1 Dataset Validation

To validate that our FACTOR benchmarks meet the required properties detailed in §3.1, we manually evaluated a sub-sample from each dataset. We sampled 138 examples from Wiki-FACTOR, 100 examples from News-FACTOR and 80 examples from Expert-FACTOR, containing 414, 300 and 240 generations overall. Each generation was annotated w.r.t. the properties manifested in §3.1, namely whether they were (1) non-factual, (2) fluent, and (3) self-consistent. To assess datasets diversity, we annotated the contradictions in accordance with the error typology of Pagnoni et al. (2021), described in §3.1. We verified that the non-factual completions are minimally edits variants of the factual completion by measuring mean edit distances.

Validation results in Table 2 show that for all datasets, almost every generated completion indeed contradicts the original one, was fluent, and was self consistent. Table 3 shows the error type distribution, indicating that FACTOR yields diverse contradiction types. Semantic frame errors (Entity, Predicate, and Circumstance) were more prevalent than discourse errors (Link and Coreference), as more sentences are suited for these type of errors.

4 Experimental Setup

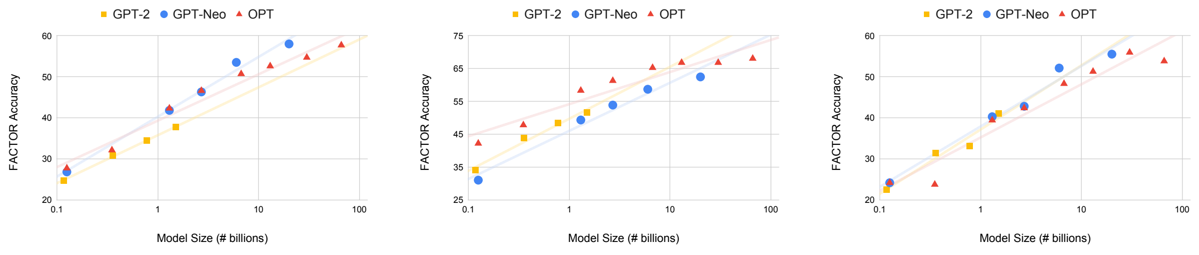
We used FACTOR benchmarks to evaluate factual knowledge of LLMs across varying model families. We describe the experimental setup below.

4.1 Datasets

The Wiki-FACTOR, News-FACTOR and Expert-FACTOR datasets are described in §3.3. For perplexity evaluation (§5.3), we selected a subset of 300 Wikipedia articles from the documents Wiki-FACTOR is based on (∼367K tokens).

4.2 Models

We performed our experiments over a set of open source models: four models of GPT-2 family (110M–1.5B; Radford et al. 2019), five models



(a) Wiki-FACTOR (b) News-FACTOR (c) Expert-FACTOR

Figure 3: Accuracy per model size for Wiki-FACTOR (left), News-FACTOR (center), and Expert-FACTOR (right) for models from GPT-2 (yellow square), GPT-Neo (blue circle), and OPT (red triangle) families.

from the GPT-Neo family (125M–20B; Black et al. 2021, 2022; Wang and Komatsuzaki 2021), and eight models of OPT (125M–66B; Zhang et al. 2022). We capped the sequence length at 1024 tokens to compare all models directly.

The corpora that our FACTOR benchmarks were constructed from were not used for training any of the examined models. News-FACTOR is based on articles published after 1/10/2021, while Expert-FACTOR is based on examples written in 2023. Both are beyond the models' data cutoff date. Wiki-FACTOR is based on Wikipedia documents from The Pile's validation split, which is not part in any of the models' training sets. (OPT and GPT-Neo models were trained on The Pile's training split, GPT-2 models were not trained on Wikipedia).

4.3 Retrieval-Augmented Models

In §5.2, we present evaluations of retrieval-augmented variants of the models. To that end, we adopted the In-Context RALM (IC-RALM) framework of Ram et al. (2023), where the retrieved document is prepended to the LLM's input, without any further training or specialized LLM architecture. In IC-RALM, a retriever is called every s tokens (*i.e.*, the *stride*), with a query comprised of the last ℓ tokens. The LLM is run with the concatenated input to assign log-probabilities to the next s tokens. We used the lexical BM25 (Robertson and Zaragoza, 2009) over Wikipedia corpus,[4] excluding the evaluated docs; and set $s = 8$, $\ell = 32$.

5 Factual Knowledge Evaluation Results

This section describes the experimental evaluation of LLM factuality using our FACTOR benchmarks. In §5.1 we show that FACTOR accuracy increases with model size but also depends on the training data (different model families differ in scores). In §5.2, we show that retrieval augmentation of the LM improves FACTOR accuracy, positioning it as the first automatic measure of factuality improvement for retrieval augmented LMs. Finally, in §5.3, we show that the pairwise model ranking of corpus perplexity and FACTOR accuracy can differ significantly. This outcome, along with manual validation of the correlation between FACTOR accuracy and factual generation in §6, solidifies FACTOR accuracy as a novel automatic measure for evaluating the proneness of an LM to generate factual information in a certain domain.

5.1 Factual Knowledge Improves with Model Size

We evaluate GPT-2, GPT-Neo, and OPT models on Wiki-FACTOR, News-FACTOR and Expert-FACTOR (Figure 3). Larger models generally outperform smaller ones within the same model family. However, even the largest models are capped at 58.0% (GPT-NeoX-20B), 68.1% (OPT-66B) and 55.9% (OPT-30B) on Wiki-FACTOR, News-FACTOR and Expert-FACTOR respectively, indicating the benchmarks are challenging. Recent works (Chuang et al., 2023; Kai et al., 2024) use Wiki-FACTOR and News-FACTOR to evaluate models from the LLaMA family (Touvron et al., 2023) and show similar trends.

We observe that all models achieve higher FACTOR accuracy on news comparing to the other two domains. This may be because news articles cover specific events, making the prefix more useful for detecting factual completions (further discussion in App. B.2). When comparing different model-families, we find that the OPT models leads on News-FACTOR, while the GPT-Neo family leads on Wiki-FACTOR. This implies that the different data sources used for training these two model fam-

[4] We used the Wikipedia corpus of Karpukhin et al. (2020), based on the dump from Dec. 20, 2018.

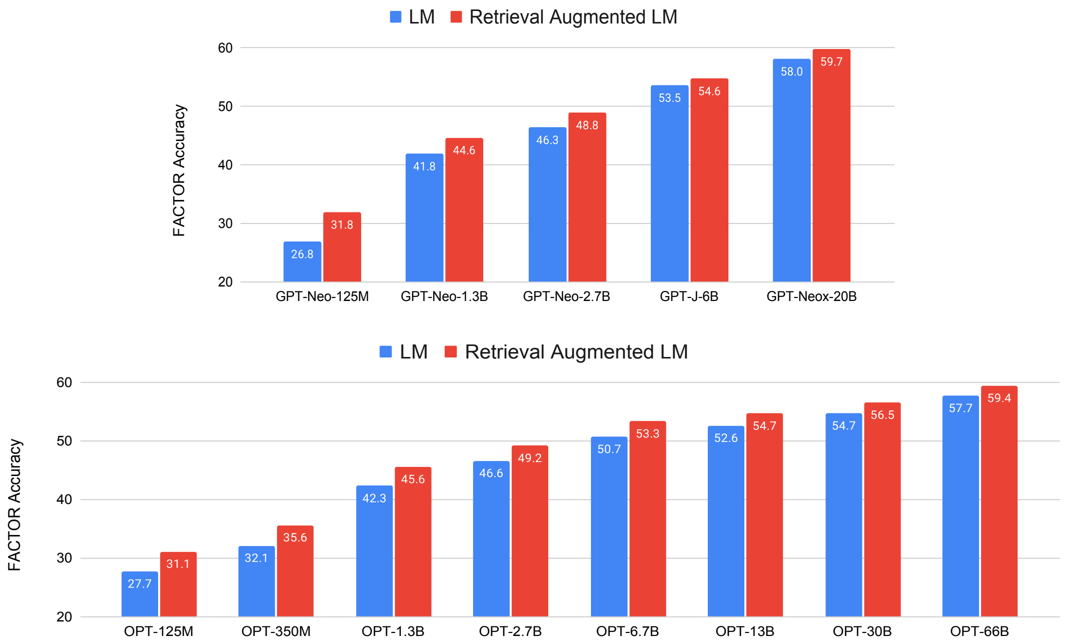


Figure 4: Factual accuracy over Wiki-FACTOR for GPT-Neo and OPT models, compared to their IC-RALM variants. IC-RALM leads to consistent improvement for all models.

ilies are suited to different domains.

5.2 The Effect of Retrieval Augmentation on Factual Knowledge

Next, we ask: *Can FACTOR accuracy be improved by augmenting models with a retrieval component?* Importantly, while a clear motivation for retrieval augmentation is factual grounding of LMs, no existing metrics allow direct measurement of it in a text generation setting. We propose FACTOR accuracy as an alternative to the course measure of LM perplexity, which is often used to assess these methods (Khandelwal et al., 2020; Borgeaud et al., 2022; Ram et al., 2023; Shi et al., 2023).

We compared the FACTOR accuracy of LLMs to that of their retrieval-augmented counterparts, implemented following the IC-RALM framework (§4.3; Ram et al. 2023). Figure 4 show the results for GPT-Neo and OPT Wiki-FACTOR. We observed consistent gains from augmenting the models with retrieval. These results highlight that grounding the model in an external corpus can improve its factuality. Since the retriever used in our experiments is used in an "off-the-shelf" manner, we speculate that further performance boosts may be gained by a retriever system specialized for this task (Izacard et al., 2022; Ram et al., 2023).

Another interesting finding is that the *relative* gains in FACTOR accuracy obtained by IC-RALM, are more moderate compared to the relative gains in perplexity over WikiText-103 (Merity et al., 2016), reported by Ram et al. (2023). We explore the connection between the two in the next section.

5.3 Perplexity Correlates but is not Always Aligned with FACTOR Accuracy

We investigate whether FACTOR accuracy adds additional information beyond perplexity, when used as a comparative metric for selecting which LM to use within a certain corpus. Figure 2 shows the FACTOR accuracy of models on Wiki-FACTOR, compared to their token-level perplexity on the Wikipedia section of The Pile's validation set (§4.1) (App. B.1 includes all evaluated models). Overall, we observe a high correlation between the two metrics. However, there are cases where they disagree (*i.e.*, a pair of models where one is better when measured by perplexity but worse in terms of FACTOR accuracy). For example, GPT-Neo-2.7B is significantly better than OPT-2.7B in terms of perplexity (9.0 vs. 10.1), but slightly worse in terms of FACTOR accuracy (46.3% vs. 46.6%). In addition, GPT-J-6B has lower perplexity compared to OPT-66B (7.4 vs. 7.6), while OPT-66B is significantly better in terms of FACTOR accuracy (57.7% vs. 53.5%). This finding suggests that (i) FACTOR accuracy offers a complementary view of models' performance, not necessarily captured by perplexity, and (ii) improvements in perplexity do not necessarily imply better factuality.

6 Factuality in Open-Ended Generation

This section explores the connection between FACTOR accuracy and factuality in open-ended generation, via human annotations.

6.1 Experimental Setup

We selected tuples of prefix, original completion and non-factual completion (t, c^+, c^-) from Wiki-FACTOR. We then manually identified the *minimal factual claim* modified by c^-, denoted by f. For example, the predicate error from Table 1, in which "*became*" was replaced with *"declined the position of"*, the edit relates to the minimal fact "*Donne became Chief Justice of Nauru and Tuvalu*".

We let LLMs generate free text, conditioned on the prefix and the completion until the edit induced by c^-. Formally, let c be the common prefix of c^+ and c^- (in the predicate error example, c is "*After completing his term, he*"). The LLM is conditioned on the concatenation of t and c. The LLM might generate the correct fact, text violating it, or other completion that does not refer to it. For each example we manually annotated whether the generated text is *true*, *false*, or *neutral* w.r.t. f.

We analyzed two models with a similar token-level perplexity but a significant gap in FACTOR accuracy: GPT-J 6B and OPT-66B (marked in a green circle in Figure 2). For each model, we considered two groups of examples: examples with c^+, c^- pairs for which the model was *right*, *i.e.*, the model assigns larger mean log-likelihood to c^+ compared to c^-, and pairs for which the model was *wrong* (the complement set). We sampled three generations per example for 100 examples from each group and for each model. Overall, we created 1200 generations. We filtered some of the samples due to ill-formatted generations or non-contradictory completions (14.5% of all samples).

6.2 Results

We assess model's knowledge of the minimal facts through manual annotation. We only considered relevant generations for their minimal fact f, excluding "neutral" generations (59.5% and 54.3% for GPT-J 6B and OPT-66B, respectively). For each model, we measure the percentage of generated texts that are true w.r.t. f in the "right" and "wrong" subsets separately. We obtained the overall FACTOR accuracy by weighting the subsets results according to their distribution in Wiki-FACTOR. Results in Table 4 (full results in App. B.2).

Model	Subset	Fact. Accuracy
GPT-J 6B	Right	30.0%
	Wrong	10.5%
	All (Weighted)	24.8%
OPT-66B	Right	46.6%
	Wrong	4.6%
	All (Weighted)	**38.8**%

Table 4: Manual factuality annotation results for OPT-66B and GPT-J 6B. For each model, we present the results per *right* and *wrong* subsets. Bottom row shows the weighted average between the *right* and *wrong* variants w.r.t to the *right/wrong* pairs of Wiki-FACTOR.

Accuracy over Wiki-FACTOR is linked with factuality in open-ended generation. For cases where models were *wrong*, they generated more false claims regarding their minimal fact. For example, OPT-66B only generated a true claim 4.6% of the times it was wrong, compared to 46.6% for when it was right. This suggests that FACTOR accuracy can shed light on the model's ability to generate factual claims accurately.

As a comparative metric, accuracy over Wiki-FACTOR aligns with factuality in open-ended generation. There were gaps in factuality annotation between OPT-66B and GPT-J 6B: OPT-66B generated *true* claims 38.8% of the time, while GPT-J 6B generated only 24.8%. This aligns with the models' performance over Wiki-FACTOR, despite sharing similar perplexity on Wiki. This suggests that FACTOR is a better proxy for measuring model factuality in a specific domain.

7 Discussion

This paper introduces FACTOR, a novel way to evaluate LMs' factuality. FACTOR creates an evaluation benchmark from a corpus, consisting of factual statements and non-factual variations. By comparing the LM's likelihood of factual claims with non-factual variants, FACTOR score captures the LM's propensity to generate factual information.

Metrics for measuring factual knowledge over a given corpus are lacking. Prior works used perplexity, which may be affected by factors other than factual knowledge and does not contrast facts with false statements. FACTOR focuses the language modeling task on factuality by taking a contrastive approach. Our experiments show that FACTOR ranks models differently than perplexity and is more aligned with factuality in open-ended generation. These findings highlight the importance of negative examples for evaluating factuality. Moreover, they indicate that incorporating negative examples into training sets might also help optimizing models to be more factual. We leave investigation of training with FACTOR style data to future work.

Our work joins recent studies on factuality evaluation in a text-generation setting, which proposed to evaluate models by fact-checking the model's generations (Lee et al., 2022; Min et al., 2023). As FACTOR focuses on evaluation over a controlled set of facts, we see these two approaches as complementary; together, they yield a more holistic assessment of LM factuality.

Limitations

We point to several limitations of our work. First, since FACTOR benchmarks are generated in an automated way, they may not fully comply with the requirements we define in §3.1, as analyzed in §3.3. Second, generating FACTOR benchmarks for different domains may pose new challenges. For instance, the selection of factual completions is straightforward in knowledge-intensive domains, where nearly every sentence in the corpus contains factual information. However, in general cases, a more intricate approach is needed to identify such sentences. Moreover, the generation of non-factual completions is based on a prompted model, specifically designed for the Wikipedia domain. While we observed those prompts applied well for the news domain, their effectiveness may vary in other, more specific domains.

Ethics Statement

Language models' tendency to generate factually inaccurate text raises significant issues. FACTOR allows automatic evaluation of factuality, which can be used to efficiently measure and develop methods for mitigating these risks. However, we stress that when deploying such models in sensitive settings, automatic evaluations may not be sufficient, and human evaluation is required.

References

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *ICML*.

Franck Burlot and François Yvon. 2017. Evaluating the morphological competence of machine translation systems. In *Proceedings of the Second Conference on Machine Translation*, pages 43–55, Copenhagen, Denmark. Association for Computational Linguistics.

Yiran Chen, Pengfei Liu, and Xipeng Qiu. 2021. Are factuality checkers reliable? adversarial meta-evaluation of factuality in summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2082–2095, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. DoLa: Decoding by contrasting layers improves factuality in large language models.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The Pile: An 800gb dataset of diverse text for language modeling.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Prakhar Gupta, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. DialFact: A benchmark for fact-checking in dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3785–3801, Dublin, Ireland. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 161–175, Dublin, Ireland. Association for Computational Linguistics.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Trans. Inf. Syst.*, 38(3).

Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. The factual inconsistency problem in abstractive text summarization: A survey. *arXiv preprint arXiv:2104.14839*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Atlas: Few-shot learning with retrieval augmented language models.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Jushi Kai, Tianhang Zhang, Hai Hu, and Zhouhan Lin. 2024. SH2: Self-highlighted hesitation helps you decode more truthfully.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. In *Advances in Neural Information Processing Systems*.

Shaobo Li, Xiaoguang Li, Lifeng Shang, Zhenhua Dong, Chengjie Sun, Bingquan Liu, Zhenzhou Ji, Xin Jiang, and Qun Liu. 2022. How pre-trained language models capture factual knowledge? a causal-inspired analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1720–1732, Dublin, Ireland. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2023. Expertqa: Expert-curated questions and attributed answers. In *arXiv*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for*

Computational Linguistics, pages 1906–1919, Online. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActscore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. *arXiv preprint arXiv:2305.14251*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only.

Hao Peng, Xiaozhi Wang, Shengding Hu, Hailong Jin, Lei Hou, Juanzi Li, Zhiyuan Liu, and Qun Liu. 2022. COPEN: Probing conceptual knowledge in pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5015–5035, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of ACL*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382, Valencia, Spain. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. 2023. Evaluating the factual consistency of large language models through summarization. In *Findings of ACL*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

A Technical Details of FACTOR Data Pipeline

A.1 Identifying Sentences' Relevant Error Types

For each sentence, we identify the types of edits we can apply to it. First, we use a part-of-speech tagger to detect relevance for entity error (detecting nouns), predicate error (detecting verbs) and coreference error (detecting pronouns). For circumstances errors, we use Named-Entity Recognition taggers to identify sentences containing locations, dates, and time entities. Finally, we search for temporal/causal link words from a predefined set of words, which implies relevance for link errors.

A.2 Setting Filters Thresholds

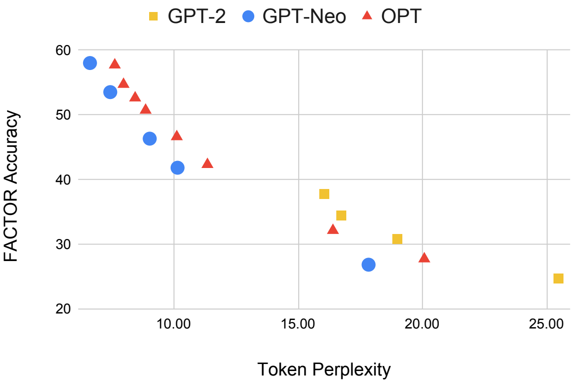


Figure 5: Accuracy per token perplexity over Wiki-FACTOR.

As discussed in §3.2.3, we applied two filters to ensure the quality of the potential completions–an NLI filter (to filter out non-contradictory completions) and an LM filter (to filter out non-fluent completions). To choose the thresholds τ_{NLI} and τ_{LM}, we manually annotated 40 samples w.r.t to the properties specified in §3.1 (*i.e.*, (1) contradictory and (2) fluent and self-consistent). We have tested thresholds 0.1-0.9, and chose the threshold which achieved highest precision without filtering out too many samples (max 35% of the samples). For the NLI filter we used DeBERTa-largs model fine-tuned on the MNLI dataset. Best threshold was $\tau_{\text{NLI}} = 0.6$, with precision of 0.96. Manually evaluating the different contradiction types we have noticed this threshold was too harsh for corefrence contradiction (87.5% of the completions were filtered out. Therefore we reduced its threshold to 0.3 which filtered out 75% of the samples). For the LM filter we used GPT2-Small. Best threshold was $\tau_{\text{LM}} = 0.2$, with precision of 0.78.

B Extended Results and Discussion

B.1 Comparison between Perplexity and FACTOR Accuracy over Wikipedia

Figure 5 presents Wiki-FACTOR scores versus LM perplexity on Wikipedia. The figure extends Figure 2, presenting all evaluated LMs: models from the GPT-Neo family (blue circle), OPT family (red triangle) and GPT2 family (yellow square).

B.2 Factuality in Open-ended Generation

Table 6 shows the extended results for the manual factuality annotation for open-ended generation experiment §6. In addition to the overall results, we include the distribution of Neutral/True/False annotations. Notably, most generations are neutral for both models. This highlights the limitation of sampled-based approach for assessing model's factual knowledge.

B.3 Knowledge of Unseen Facts

As seen in Figure 3 in §5.1, FACTOR-accuracy is often way above the random baseline of 25%, indicating that some models succeed in predicting unseen facts. It is possible that the knowledge of these facts is derived from another document in the training data (for example, Wikipedia contains many different articles related to each other, sharing similar factual statements). Another possibility is that an unseen fact is implied by the prefix. We hypothesize that this leads to higher FACTOR scores in the news domain, which often covers specific events, making the prefix more useful for detecting factual completions. Analysis of these cases is non-trivial, and is left for future work.

C Dataset Licenses

Table 5 details the license for each corpus we used in the paper:

Dataset	License
The Pile	MIT
The RefinedWeb	ODC-By 1.0
ExpertQA	MIT

Table 5: Datasets' licenses

D Prompts for Contradictions Generation

We prompted the model to generate multiple candidate completions, For each of the five error types: entity (Table 7), circumstance (Table 8), coreference (Table 9), predicate (Table 10 and 11) and link (Table 12). The prompts are concatenated to a given a completion and its near context, with the exception of link-prompt where only the completion is given (we found that the instruct model tends to repeat the context when it's appended to this particular prompt). The prompts instruct the model to first plan its local edits, and then generate the contradiction.

Model	Variant	Neutral	True (T)	False (F)	Fact. Accuracy $\left(=\frac{\mathrm{T}}{\mathrm{T+F}}\right)$
GPT-J 6B	Right	62.4%	11.3%	26.3%	30.0%
	Wrong	48.8%	5.4%	45.8%	10.5%
	All (Weighted)	59.5%	10.0%	30.5%	24.8%
OPT-66B	Right	54.1%	21.4%	24.5%	46.6%
	Wrong	55.1%	2.1%	42.8%	4.6%
	All (Weighted)	54.3%	17.7%	28.4%	38.8%

Table 6: Manual factuality annotation results for OPT-66B and GPT-J 6B. For each model, we present the results per *right* and *wrong* subsets. Bottom row shows the weighted average between the *right* and *wrong* variants w.r.t to the *right/wrong* pairs of Wiki-FACTOR.

Type	Prompt
Entity	Given a context and a completion, write diverse alternative completions that contradict the original completion meaning. First, identify if the completion contains an entity. Then, write the contradiction by modifying an entity or it's property, add additional modifications if necessary. Make sure the changes you make are minimal (so only change necessary details to make the sentence plausible). Do not modify dates or quantities. ## Context: "Sorry" is a song by American singer Madonna from her tenth studio album Confessions on a Dance Floor (2005). It was written and produced by Madonna and Stuart Price, and released as the second single from the album on February 7, 2006. It later appeared on Celebration, her 2009 greatest hits album. An uptempo dance song, " Sorry " was one of the first tracks developed for the album and had numerous remix treatments before the ultimate version of the track was finalized. Completion: One of the remixes was done by the known band the Pet Shop Boys, featuring added lyrics by the band. 1. Change: "Pet Shop Boys" to "Maddona". Contradiction: One of the remixes was done by the known singer Maddona, featuring added lyrics by the singer. 2. Change: "Pet Shop Boys" to "Depeche Mode". Contradiction: One of the remixes was done by the known band Depeche Mode, featuring added lyrics by the band. 3. Change: "known" to "unfamiliar". Contradiction: One of the remixes was done by the unfamiliar band Pet Shop Boys, featuring added lyrics by the band. 4. Change: "Pet Shop Boys" to "the Killers". Contradiction: One of the remixes was done by the known band the Killers, featuring added lyrics by the band. ## Context: {context} Completion: {completion}

Table 7: Prompt for entity-errors generation

Type	Prompt
Circumstance	Given a context and a completion, write diverse alternative completions that contradict the original completion meaning. First, identify if the completion describes the circumstances of an event (location or time). If circumstances are mentioned, modify it to contradict the completion. Do not add time or location if they didn't appear in the original completion. Make sure the changes you make are minimal. ## Context: The kingdom had been in long gradual decline since the early 13th century. Had Pagan possessed a stronger central government, the collapse could have been temporary, and the country "could have risen again". But the dynasty could not recover, and because the Mongols refused to fill the power vacuum, no viable center emerged in the immediate aftermath. As a result, several minor states fought it out for supremacy for the better part of the 14th century. Completion: It was only in the late 14th century that two relatively strong powers emerged in the Irrawaddy basin, restoring some semblance of normalcy. 1. Change: "14th" to "15th". Contradiction: It was only in the late 15th century that two relatively strong powers emerged in the Irrawaddy basin, restoring some semblance of normalcy. 2. Change: "Irrawaddy" to "Chindwin". Contradiction: It was only in the late 14th century that two relatively strong powers emerged in the Chindwin basin, restoring some semblance of normalcy. 3. Change: "late" to "mid". Contradiction: It was only in the mid 14th century that two relatively strong powers emerged in the Irrawaddy basin, restoring some semblance of normalcy. ## Context: {context} Completion: {completion}

Table 8: Prompt for circumstance-errors generation

Type	**Prompt**
Coreference	Given a context and a completion, write diverse alternative completions that contradict the original completion meaning. First, decide if the completion contains a pronoun (such as: he, she, it, they, his, her, its, theirs...) and write the entity it refers to. Write the contradiction by modifying the pronoun to contradict the original coreference. ## Context: His stance in favor of prohibition cost him the votes of four legislators in his own party and the seat went to Republican William O. Bradley. Six years later Beckham secured the seat by popular election, but he lost his re-election bid largely because of his pro-temperance views and his opposition to women's suffrage. Completion: Though he continued to play an active role in state politics for another two decades, he never returned to elected office, failing in his gubernatorial bid in 1927 and his senatorial campaign in 1936. 1. Pronoun: he Change: "he" to "Bradley". Contradiction: Though Bradley continued to play an active role in state politics for another two decades, he never returned to elected office, failing in his gubernatorial bid in 1927 and his senatorial campaign in 1936. 2. Pronoun: he Change: "he" to "Bradley". Contradiction: Though he continued to play an active role in state politics for another two decades, Bradley never returned to elected office, failing in his gubernatorial bid in 1927 and his senatorial campaign in 1936. 3. Pronoun: his Change: "his" to "Bradley's". Contradiction: Though he continued to play an active role in state politics for another two decades, he never returned to elected office, failing in Bradley's gubernatorial bid in 1927 and his senatorial campaign in 1936. ## Context: The early 6th century saw another queen ruling the city, known only as the "Lady of Tikal", who was very likely a daughter of Chak Tok Ich 'aak II. Completion: She seems never to have ruled in her own right, rather being partnered with other rulers. 1. Pronoun: She Change: "She" to "He" and "her" to "his". Contradiction: He seems never to have ruled in his own right, rather being partnered with other rulers. 2. Pronoun: She Change: "She" to "The king" and "her" to "his". Contradiction: The king seems never to have ruled in his own right, rather being partnered with other rulers. 3. Pronoun: She Change: "She" to "Chak Tok Ich". Contradiction: Chak Tok Ich seems never to have ruled in her own right, rather being partnered with other rulers. ## Context: {context} Completion: {completion}

Table 9: Prompt for coreference-errors generation

Type	Prompt
Predicate	Given a context and a completion, write diverse alternative completions, that contradict the original completion meaning by modifying verbs. First, Identify a verb in the original completion, and then write the contradiction by modifying it. Make sure the contradictions are grammatically correct, fluent and consistent. Make any necessary additional modifications to ensure that. ## Context: Homarus gammarus is a large crustacean, with a body length up to 60 centimetres (24 in) and weighing up to 5 – 6 kilograms (11 – 13 lb), although the lobsters caught in lobster pots are usually 23 – 38 cm (9 – 15 in) long and weigh 0.7 – 2.2 kg (1.5 – 4.9 lb). Completion: Like other crustaceans, lobsters have a hard exoskeleton which they must shed in order to grow, in a process called ecdysis (moulting). 1. Change: "shed" to "retain". Additional changes: "in order to grow" to "in order to survive". Contradiction: Like other crustaceans, lobsters have a hard exoskeleton which they must retain in order to survive, in a process called ecdysis (moulting). 2. Change: "grow" to "maintain their size". Contradiction: Like other crustaceans, lobsters have a hard exoskeleton which they must shed in order to maintain their size, in a process called ecdysis (moulting). 3. Change: "shed" to "keep". Additional changes: "in order to grow" to "in order to strengthen". Contradiction: Like other crustaceans, lobsters have a hard exoskeleton which they must keep in order to strengthen, in a process called ecdysis (moulting). ## Context: The ridge offered a natural avenue of approach to the airfield, commanded the surrounding area and was almost undefended. Edson and Thomas tried to persuade Vandegrift to move forces to defend the ridge, but Vandegrift refused, believing that the Japanese were more likely to attack along the coast. Completion: Finally, Thomas convinced Vandegrift that the ridge was a good location for Edson’s Raiders to rest from their actions of the preceding month. 1. Change: "rest" to "keep up". Contradiction: Finally, Thomas convinced Vandegrift that the ridge was a good location for Edson’s Raiders to keep up with their actions of the preceding month. 2. Change: "convinced Vandegrift" to "made Vandegrift doubt". Contradiction: Finally, Thomas made Vandegrift doubt that the ridge was a good location for Edson’s Raiders to rest from their actions of the preceding month. 3. Change: "rest" to "continue". Contradiction: Finally, Thomas convinced Vandegrift that the ridge was a good location for Edson’s Raiders to continue their actions of the preceding month. ## Context: According to a report titled Wolves in Sheep’s Clothing, which documents the increase in potentially violent, profane, and sexual content in children’s programming, the Parents Television Council, a watchdog media group, and fans believed the SpongeBob SquarePants episode" Sailor Mouth "was an implicit attempt to promote and satirize use of profanity among children. Completion: The episode originally aired during the 2001 – 02 television season, ironically the season in which the PTC named SpongeBob SquarePants among the best programs on cable television, but the report cited a repeat broadcast of the episode from 2005 to prove its point that it promoted use of profanity among children. 1. Change: "prove" to "refute". Additional changes: "best" to "most profane". Contradiction: The episode originally aired during the 2001 – 02 television season, ironically the season in which the PTC named SpongeBob SquarePants among the most profane programs on cable television, but the report cited a repeat broadcast of the episode from 2005 to refute its point that it promoted use of profanity among children. 2. Change: "originally aired" to "pulled off". Contradiction: The episode was pulled off from the 2001 – 02 television season, ironically the season in which the PTC named SpongeBob SquarePants among the best programs on cable television, but the report cited a repeat broadcast of the episode from 2005 to prove its point that it promoted use of profanity among children. ## Context: {context} Completion: {completion}

Table 10: Prompt for predicate-errors generation (the rest of the prompt is in table 11)

Type	Prompt
Predicate	Context: By Part II of the series, Shikamaru is capable of utilizing multiple shadow-based techniques at once and can lift his shadow from the ground in order to interact with physical objects; for instance, he can pierce enemies with the shadow tendrils or use them to throw weapons. Shikamaru approaches the exams with a sense of apathy; when he battles the Sunagakure ninja Temari, he defeats her but forfeits his match to her, due to his chakra being low. Completion: Despite this loss, he is the only ninja among his peers to be promoted to the rank of Chunin, as the overseers of the exams were impressed by the insight and intelligence he demonstrated against Temari. 1. Change: "promoted" to "demoted". Additional changes: "Despite" to "Due", "as" to "although". Contradiction: Due to this loss, he is the only ninja among his peers to be demoted to the rank of Chunin, although the overseers of the exams were impressed by the insight and intelligence he demonstrated against Temari. 2. Change: "were impressed" to "underappreciated". Additional changes: "as" to "although". Contradiction: Despite this loss, he is the only ninja among his peers to be promoted to the rank of Chunin, although the overseers of the exams underappreciated the insight and intelligence he demonstrated against Temari. 3. Change: "demonstrated" to "failed to demonstrate". Additional changes: "as" to "although", "impressed" to "disappointed". Contradiction: Despite this loss, he is the only ninja among his peers to be promoted to the rank of Chunin, although the overseers of the exams were disappointed by the insight and intelligence he failed to demonstrate against Temari. ## Context: {context} Completion: {completion}

Table 11: Prompt for predicate-errors generation (continue of the prompt in table 10)

Type	Prompt
Link	Given a sentence, write contradictory sentences by modifying a temporal link. First, identify a link between events, and then modify it. Make sure the contradictions are grammatically correct and fluent. If no such link exists, answer "NA". ## Sentence: Prior to filming, a week was spent reinforcing the roof of the liquor store to ensure it would not collapse if it were to be intruded by a group of fans. 1. Change: "prior to" to "after". Contradiction: After filming, a week was spent reinforcing the roof of the liquor store to ensure it would not collapse if it were to be intruded by a group of fans. ## Sentence: Lewis McAllister, a businessman in Tuscaloosa, Alabama, was the first Republican to serve in the Mississippi House of Representatives since Reconstruction, 1962-1968; he resided in Meridian prior to 1971. 1. Change: "prior to" to "after". Contradiction: Lewis McAllister, a businessman in Tuscaloosa, Alabama, was the first Republican to serve in the Mississippi House of Representatives since Reconstruction, 1962-1968; he resided in Meridian after 1971. 2. Change: "since" to "before" Contradiction: Lewis McAllister, a businessman in Tuscaloosa, Alabama, was the first Republican to serve in the Mississippi House of Representatives before Reconstruction, 1962-1968; he resided in Meridian prior to 1971. ## Sentence: The decline of the railroad industry caused significant job losses, resulting in a population decline as workers left for other areas. 1. Change: "caused" to "caused by". Contradiction: The decline of the railroad industry, caused by significant job losses, resulting a population decline as workers left for other areas. 2. Change: "resulting" to "was the result of". Contradiction: The decline of the railroad industry caused significant job losses, was the result of a population decline, as workers left for other areas. ## Sentence: {completion}

Table 12: Prompt for link-errors generation