Black-box language model explanation by context length probing

Anonymous ACL submission

Abstract

The increasingly widespread adoption of large Transformer language models has highlighted the need for improving their explainability. We present *context length probing*, a novel explanation technique for causal language models, based on tracking the predictions of a model as a function of the length of available context, and allowing to assign *differential importance scores* to different contexts. The technique is model-agnostic and does not rely on access to model internals beyond computing token-level probabilities. We apply context length probing to large pre-trained language models and offer some initial analyses and insights, including the potential for studying long-range dependencies. The source code[1] and an interactive demo[2] of the method are available.

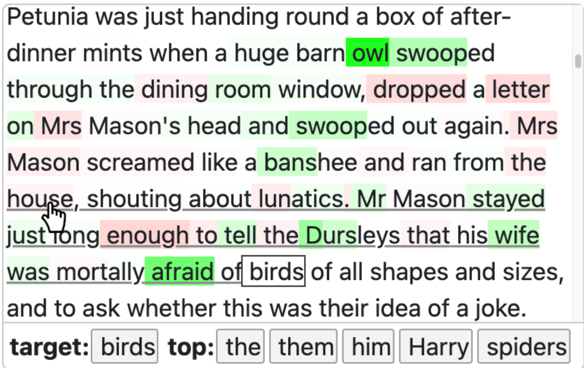


Figure 1: A screenshot of an interactive demo[2] of the proposed method. After selecting a target token (here "**birds**"), the preceding tokens are highlighted according to their (normalized) *differential importance scores* (green = positive, red = negative), obtained using our method. The user can also explore the top predictions for contexts of different lengths (here the context "house, shouting about lunatics [...] mortally afraid of").

1 Introduction

Large language models, typically based on the Transformer architecture (Vaswani et al., 2017), have recently seen increasingly widespread adoption, yet understanding their behaviour remains a difficult challenge and an active research topic.

A popular way to dissect Transformers is by visualizing their attention weights (e.g. Vig, 2019; Hoover et al., 2020). However, it has been argued that this does not provide reliable explanations and can be misleading (Jain and Wallace, 2019; Serrano and Smith, 2019). A more recent line of work (Elhage et al., 2021; Olsson et al., 2022) explores "mechanistic explanations", based on reverse-engineering the computations performed by Transformers. These techniques are tied to concrete architectures, which are often "toy" versions of those used in real-world applications, e.g. attention-only Transformers in Elhage et al..

Other options include general-purpose methods like neuron/activation interpretation (e.g. Geva et al., 2021; Goh et al., 2021; Dai et al., 2022), saliency maps (e.g. Ancona et al., 2019; Fong and Vedaldi, 2017) and influence functions (Koh and Liang, 2017). These require access to internal representations and/or the ability to backpropagate gradients, and have some caveats of their own (Kindermans et al., 2019; Kokhlikyan et al., 2021).

In this work, we propose a simple explanation technique for causal LMs, *context length probing*, based on tracking the predictions of the model as a function of the number of tokens available as context. Our proposal has the following advantages:

- It is conceptually simple, as it provides an answer to a natural question: *How does the length of available context impact the prediction?*
- It can be applied to a pre-trained model without retraining or fine-tuning and without training any auxiliary models.
- It does not require access to model weights, internal representations or gradients.
- It is model-agnostic: it can be applied to any causal LM, including attentionless architectures

[1] https://anonymous.4open.science/r/context-probing-DBEB

[2] https://context-probing.netlify.app/

like RNN (Mikolov et al., 2010) and CNN (Dauphin et al., 2017). The only requirement for the model is to accept arbitrary input segments (i.e. not be limited to document prefixes).

One application of context length probing that we explore here lies in assigning what we call *differential importance scores* to contexts of different lengths. This can be seen as complementary to other techniques like attention or saliency map visualization, and has a potential to be applied to studying long-range dependencies.

2 Method

2.1 Context length probing

A causal LM estimates the conditional probability distribution of a token given its left-hand context in a document:

$$p(x_n \mid x_1, \ldots, x_{n-1}). \tag{1}$$

We are interested here in computing the probabilities conditioned on a *reduced* context of length $c \in \{1, \ldots, n-1\}$:

$$p(x_n \mid x_{n-c}, \ldots, x_{n-1}), \tag{2}$$

so that we may then study the behavior of this distribution as a function of c.

An apparent obstacle in doing so is that applying the model to an arbitrary subsequence $x_{n-c}, \ldots, x_{n-1}$, instead of the full document $x_1, \ldots, x_N$, may lead to inaccurate estimates of the probabilities in Eq. (2). However, we note that large LMs are not usually trained on entire documents. Instead, the training data is pre-processed by shuffling all the documents, concatenating them (with a special token as a separator), and splitting the resulting sequence into *chunks* of a fixed length (usually 1024 or 2048 tokens) with no particular relation to the document length. Thus, the models are effectively trained to accept sequences of tokens starting at arbitrary positions in a document and it therefore appears correct to employ them as such to compute estimates of Eq. (2).[3]

It now remains to be detailed how to efficiently evaluate the above probabilities for all positions n and context lengths c. Specifically, for a given document $x_1, \ldots, x_N$ and some maximum context length $c_{\max}$, we are interested in an $(N-1) \times c_{\max} \times |\mathcal{V}|$ tensor $\boldsymbol{P}$, where $\mathcal{V} = \{w_1, \ldots, w_{|\mathcal{V}|}\}$ is the vocabulary, such that:

$$\boldsymbol{P}_{n,c,i} = p(x_{n+1} = w_i \mid x_{n-c+1}, \ldots, x_n), \tag{3}$$

with $\boldsymbol{P}_{n,c,*} = \boldsymbol{P}_{n,n-1,*}$ for $n \leq c$. Observe that by running the model on any segment $x_m, \ldots, x_n$, we obtain all the values $\boldsymbol{P}_{m+c-1,c,*}$ for $c \in \{1, \ldots, n-m+1\}$. Therefore, we can fill in the tensor $\boldsymbol{P}$ by applying the model along a sliding window of size $c_{\max}$, i.e. running it on N (overlapping) segments of length at most $c_{\max}$. See Appendix A for an illustration and additional remarks.

2.2 Metrics

Having obtained the tensor $\boldsymbol{P}$ as we have just described, we use it to study how the predictions evolve as the context length is increased from 1 to $c_{\max}$. Specifically, our goal is to define a suitable metric that we can compute from $\boldsymbol{P}_{n,c,*}$ and follow it as a function of c (for a specific n or on average).

One possibility would be to use the negative log-likelihood (NLL) loss values:

$$-\log p(x_{n+1} \mid x_{n-c+1}, \ldots, x_n). \tag{4}$$

However, this may not be a particularly suitable metric for explainability purposes, as it depends on the *ground truth* x_{n+1}, which is only one of many plausible continuations. For this reason, we propose to instead measure the Kullback-Leibler (KL) divergence to the maximum-context predictions,

$$\begin{aligned} \mathcal{D}_{n,c} &= D_{\mathrm{KL}}[\boldsymbol{P}_{n,c_{\max},*} \parallel \boldsymbol{P}_{n,c,*}] \\ &= \sum_{i=1}^{|\mathcal{V}|} \boldsymbol{P}_{n,c_{\max},i} \log \frac{\boldsymbol{P}_{n,c_{\max},i}}{\boldsymbol{P}_{n,c,i}}, \end{aligned} \tag{5}$$

which expresses the amount of information lost from the maximum-context prediction by limiting the context length to c.

2.3 Differential importance scores

We are also interested in studying how individual *increments* in context length affect the predictions. We propose to quantify this as the change in the KL divergence metric (5) when a new token is introduced into the context. Specifically, for a pair of tokens x_{n+1} (the *target token*) and x_m (the *context token*), we define a *differential importance score* (Δ-score for short)

$$\Delta\mathcal{D}_{n,m} = \mathcal{D}_{n,n-m-1} - \mathcal{D}_{n,n-m}. \tag{6}$$

[3]For models trained on data that is pre-processed differently, (re)training or fine-tuning with data augmentation such as random shifts may be needed in order to apply our method.

name	#param	#layer	#head	d_{model}	max len
gpt2	117 M	12	12	768	1024
gpt2-xl	1.5 B	48	25	1600	1024
gpt-j-6B	6.1 B	28	16	4096	2048

Table 1: Hyperparameters of the 3 models used.

We may visualize these scores as a way to explain the LM predictions, much like is often done with attention weights, with two important caveats. First, a high $\Delta\mathcal{D}_{n,m}$ should not be interpreted as meaning that x_m in isolation is important for predicting x_{n+1}, but rather that it is salient given the context that follows it (i.e. it *disambiguates* the following context). Second, unlike attention weights, our scores need not sum up to one, and can be negative: a token may *introduce ambiguity*. The proposed representation is hence more conceptually similar to a saliency map than to an attention map.

3 Results

We apply the proposed technique to publicly available pre-trained large Transformer language models, namely GPT-J (Wang and Komatsuzaki, 2021) and two GPT-2 (Radford et al., 2019) variants – see Table 1 for an overview. We use the validation set of the English LinES treebank[4] from Universal Dependencies (UD; Nivre et al., 2020), containing 8 documents with a total length of 20 672 tokens[5] and covering fiction, an online manual, and Europarl data. We set $c_{\text{max}} = 1023$. We use the 🤗 Transformers library[6] (Wolf et al., 2020) to load the pre-trained models and run inference. Further technical details are included in Appendix B.

3.1 LM loss by context length

Fig. 2 shows the cross entropy losses (NLL means) across the whole validation dataset as a function of context length c. Larger models perform better as expected, but increased context improves performance in all cases.

We display in Fig. 3 the same information (loss by context length) broken down by part-of-speech (POS) tags, for GPT-J only. For most POS tags, the behavior is similar to what we observed in Fig. 2 and the loss appears to stabilize around context lengths 16–64. However, we see a distinct behaviour for proper nouns (PROPN), which are the hardest-to-predict category for short contexts, but whose loss improves steadily with increasing c, surpassing that of regular nouns (NOUN) at $c = 162$.

[4] https://universaldependencies.org/treebanks/en_lines/index.html

[5] After concatenating all sentences and applying the GPT-2 tokenizer, which is used by both GPT-2 and GPT-J.

[6] https://github.com/huggingface/transformers

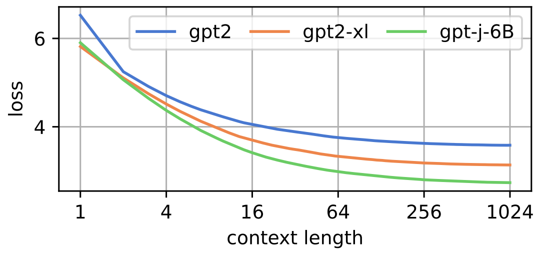


Figure 2: Mean LM losses by context length.

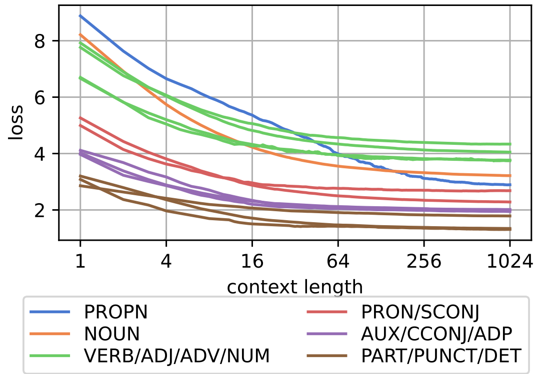


Figure 3: Mean GPT-J loss by context length and part-of-speech (POS) tag of the target token. Only POS tags with at least 100 occurrences in the dataset are included. The tags are grouped (arbitrarily) for clarity.

3.2 Per-token losses by context length

We have also examined token-level losses, as well as the KL divergence metric (see Section 2.2); an example plot is shown in Fig. 4 and more are found in Appendix C.1. In general, we observe that the values tend to change gradually with c; large differences are sparse, especially for large c, and can often be attributed to important pieces of information appearing in the context (e.g. “owl” and “swoop” in the context of “birds” in Fig. 4). This justifies our use of these differences as importance scores.

3.3 Differential importance scores

To facilitate the exploration of Δ-scores from Section 2.3, we have created an interactive web demo,[2] which allows visualizing the scores for any of the 3 models on the validation set as shown in Fig. 1.

We display in Fig. 5 the magnitudes of the Δ-scores – normalized for each position to sum up

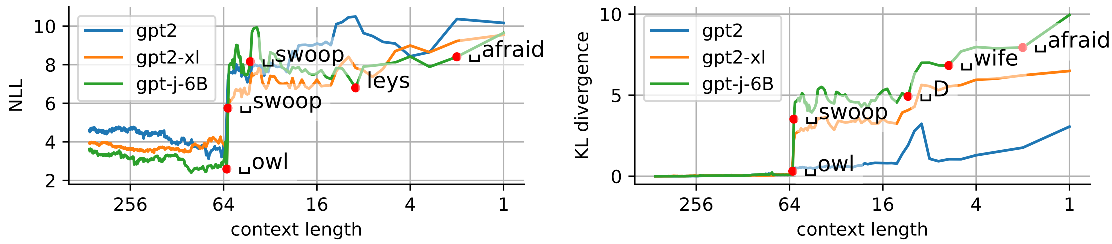


Figure 4: NLL (left) and KL divergence (right) as a function of context length for a selected example: "[…] mortally afraid of **birds**" (same as in Fig. 1). The x axis is reversed for visual correspondence with the left-hand context. The 5 context tokens causing the largest drops in each metric for GPT-J are marked by red dots.

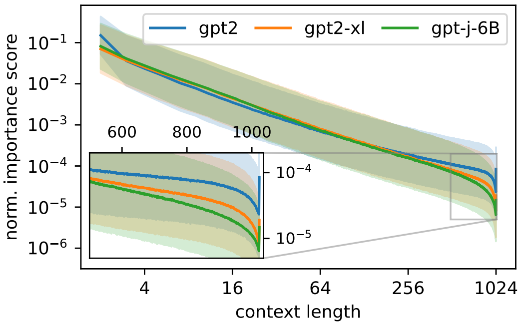


Figure 5: Normalized Δ-score log-magnitude (mean and std. dev.) by context length and by model. Only positions $n \geq 1024$ are included.

to 1 across all context lengths – as a function of context length. The plot suggests a power-law-like inverse relationship, i.e. increasing context length proportionally reduces the average Δ-score magnitude. We interpret this as far-away tokens being less likely to carry information not already covered by shorter contexts. Long contexts (see inset in Fig. 5) bear less importance for larger models than for smaller ones, perhaps because the additional capacity allows relying more on shorter contexts.

See Appendix C.2 for further analysis.

4 Limitations

Experiments. We acknowledge the limited scope of our experiments, including only 8 (closed-domain) documents, 3 models and a single language. This is largely due to the limited availability of suitable large LMs and their high computational cost. Still, we believe that our experiments are valuable as a case study that already clearly showcases some interesting features of our methodology.

Choice of metrics. The proposed methodology allows investigating how any given metric is impacted by context, yet our study is limited to NLL loss and the proposed KL divergence metric (the latter for defining importance scores). These may not be optimal for every purpose, and other choices should be explored depending on the application. For example, to study sequences *generated* (sampled) from a LM, one might want to define importance scores using a metric that does depend on the generated token, e.g. its NLL loss or its ranking among all candidates. (Indeed, our web demo also supports Δ-scores defined using NLL loss values.)

5 Conclusion and future directions

We have presented *context length probing*, a novel causal LM explanation technique based on tracking the predictions of the LM as a function of context length, and enabling the assignment of *differential importance scores* (*Δ-scores*). While it has some advantages over existing techniques, it answers different questions, and should thus be thought of as complementary rather than a substitute.

A particularly interesting feature of our Δ-scores is their apparent potential for discovering *long-range dependencies* (LRDs) (as they are expected to highlight information not already covered by shorter contexts, unlike e.g. attention maps).

Remarkably, our analysis suggests a power-law-like inverse relationship between context length and importance score, seemingly questioning the importance of LRDs in language modeling. While LRDs clearly appear crucial for applications like long-form text generation, their importance may not be strongly reflected by current LM performance metrics. We thus believe that there is an opportunity for a specialized benchmark of LRD modeling capabilities of different models. This should elucidate questions like to what extent improvements in LM performance are due to better LRD modeling, how LRDs are handled by various Transformer variants (e.g. Kitaev et al., 2020; Katharopoulos et al., 2020; Choromanski et al., 2021), or what their importance is for different tasks.

References

Marco Ancona, Cengiz Oztireli, and Markus Gross. 2019. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 272–281. PMLR.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. 2021. Rethinking attention with Performers. In *9th International Conference on Learning Representations (ICLR 2021)*. OpenReview.net.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 933–941. PMLR.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for Transformer circuits. *Transformer Circuits Thread*.

Ruth C. Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3449–3457. IEEE Computer Society.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. 2021. Multimodal neurons in artificial neural networks. *Distill*.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196, Online. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are RNNs: Fast autoregressive Transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2019. The (un)reliability of saliency methods. In Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller, editors, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*, pages 267–280. Springer.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient Transformer. In *8th International Conference on Learning Representations (ICLR 2020)*. OpenReview.net.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR.

Narine Kokhlikyan, Vivek Miglani, Bilal Alsallakh, Miguel Martin, and Orion Reblitz-Richardson. 2021. Investigating sanity checks for saliency maps with image and text classification. *arXiv preprint arXiv:2106.07475*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048. ISCA.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*.

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

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 billion parameter autoregressive language model.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. 🤗 Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

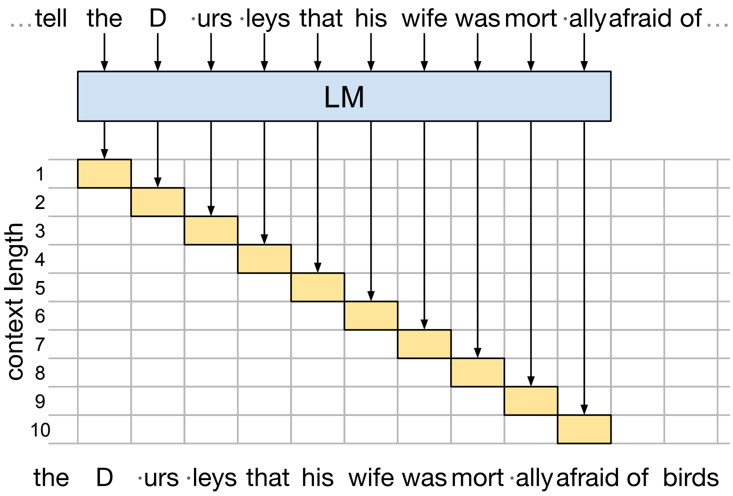


Figure 6: A step of context length probing with $c_{\max} = 10$. The input tokens are shown at the top, the target tokens at the bottom. The effective context length for each target token is equal to its offset from the beginning of the segment, e.g. the context for predicting "␣D" is "␣the" ($c = 1$), the context for "urs" is "␣the␣D" ($c = 2$), etc.

A Context length probing

Fig. 6 illustrates a step of context length probing. We wish to obtain the tensor $\boldsymbol{P}$ from Eq. (3), understood as a table where each cell contains the prediction (next-token logits) for a given position in the text and a given context length. By running our LM on a segment of the text, we get predictions such that for the n-th token in the segment, the effective context length is equal to n, which corresponds to a diagonal in the table. We can thus fill in the whole table by running the LM on all segments of length $c_{\max}$ (plus trailing segments of lengths $c_{\max} - 1, \ldots, 1$).

Notice that this process is somewhat similar to (naïvely) running the LM in generation mode, except that at each step, the leading token is removed, preventing the use of caching to speed up the computation.

In practice, it is not necessary to explicitly construct the tensor $\boldsymbol{P}$. Indeed, we find it more efficient to instead store the raw logits obtained by running the model on all the segments, then do the necessary index arithmetics when computing the metrics.

B Technical details

Data. The LinES treebank is licensed under Creative Commons BY-NC-SA 4.0. We concatenated all tokens from each of the documents from the treebank, then re-tokenized them using the GPT-2 tokenizer. We mapped the original (UD) POS tags to the GPT-tokenized dataset in such a way that every GPT token is assigned the POS tag of the first UD token it overlaps with.

Models. We used the models `EleutherAI/gpt-j-6B` (Apache 2.0 license), and `gpt2-xl` and `gpt2` (MIT license), all from `huggingface.co`.

Computation. We parallelized the inference over 500 jobs on a compute cluster,[7] each running on 8 CPU cores with at least 8 GB of RAM per core, with a batch size of 16. Each job took about 10–20 min for GPT-2 and 30–60 min for GPT-J. Additionally, computing the metrics from the logits (which take up 2 TB of disk space in `float16`) took between 2 and 4 h per model on a single machine with 32 CPU cores. The total computing time was 318 core-days, including debugging runs and runs repeated due to flaws.

[7] [anonymized], the cluster computing infrastructure of [anonymized]

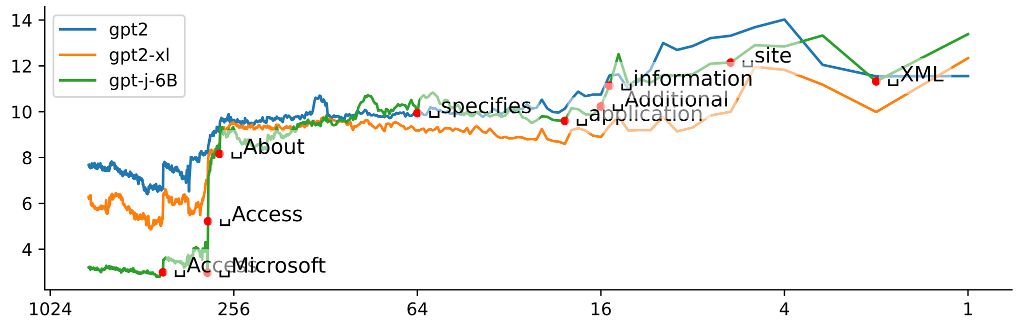
C Additional plots

C.1 Token-wise metrics as a function of context length

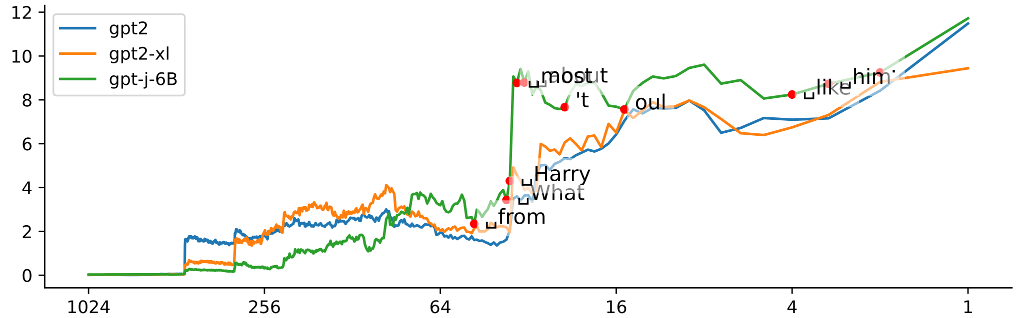
Figs. 7 and 8 show NLL and KL divergence (5), respectively, as a function of context length, for selected target tokens (proper nouns) from the validation set.

C.2 Importance scores

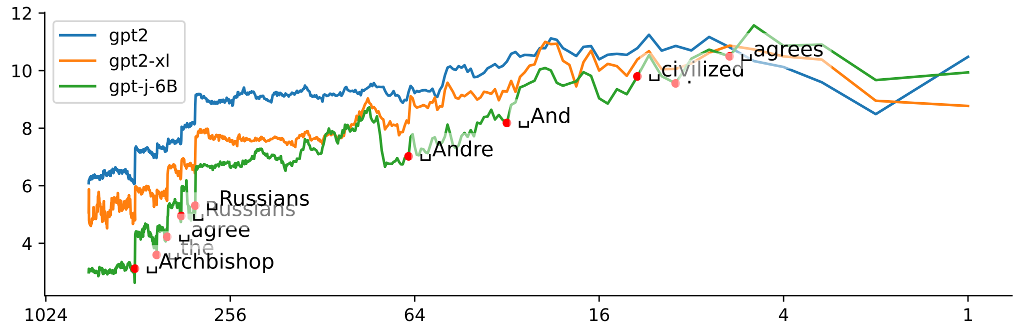
In Fig. 9, we display the mean importance score received by each POS category, by model. We can see that proper nouns (PROPN) are substantially more informative than other categories, but less so for the smallest model. This could mean e.g. that larger models are better at memorizing named entities from training data and using them to identify the topic of the document.



(a) . . . and attribute means (and thus how the data between them will look in a browser), XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the application that reads it. Additional information about XML can be found on the web site. About importing XML data **Access**

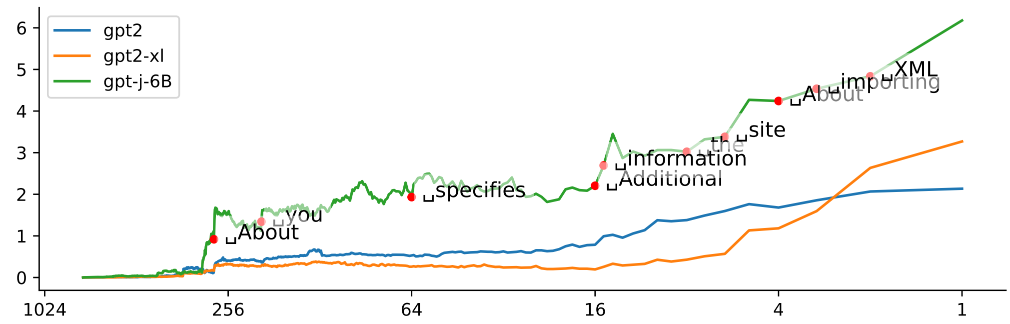


(b) . . . he felt things were getting too quiet, and small explosions from Fred and George's bedroom were considered perfectly normal. What Harry found most unusual about life at Ron's, however, wasn't the talking mirror or the clanking ghoul: it was the fact that everybody there seemed to like him. Mrs **Weasley**

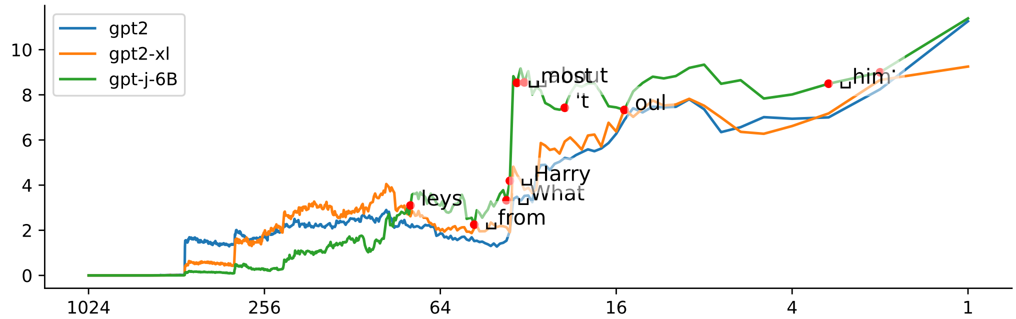


(c) . . . is a great difference between Napoleon the Emperor and Napoleon the private person. There are raisons d'etat and there are private crimes. And the talk goes on. What is still being perpetuated in all civilized discussion is the ritual of civilized discussion itself. Tatu agrees with the Archbishop about the **Russians**

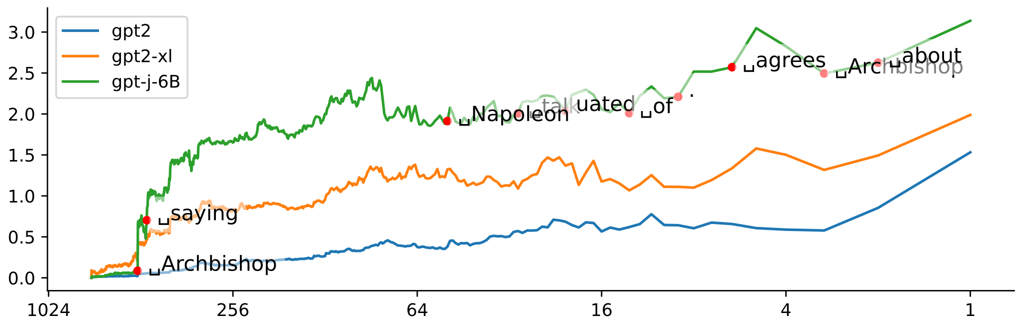
Figure 7: NLL losses (y axis) for 3 selected target tokens as a function of context length (x axis). Below each plot, the target token is displayed in bold, along with a context of 60 tokens. The x axis is reversed to correspond visually to left-hand context. The red dots show the 10 tokens that cause the largest drops in cross entropy when added to the context.



(a) . . . and attribute means (and thus how the data between them will look in a browser), XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the application that reads it. Additional information about XML can be found on the web site. About importing XML data **Access**



(b) . . . he felt things were getting too quiet, and small explosions from Fred and George's bedroom were considered perfectly normal. What Harry found most unusual about life at Ron's, however, wasn't the talking mirror or the clanking ghoul: it was the fact that everybody there seemed to like him. Mrs **Weasley**



(c) . . . is a great difference between Napoleon the Emperor and Napoleon the private person. There are raisons d'etat and there are private crimes. And the talk goes on. What is still being perpetuated in all civilized discussion is the ritual of civilized discussion itself. Tatu agrees with the Archbishop about the **Russians**

Figure 8: KL divergences (y axis) from Eq. (5) for 3 selected target tokens as a function of context length (x axis). Below each plot, the target token is displayed in bold, along with a context of 60 tokens. The x axis is reversed to correspond visually to left-hand context. The red dots show the 10 tokens that cause the largest drops in the metric when added to the context.

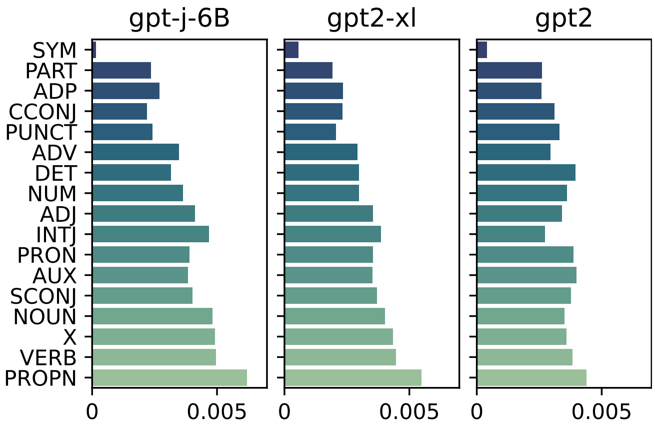


Figure 9: Mean differential importance score by POS tag of the context token and by model.