Temporal Generalization in Language Models

CLMM, Spring 2026

Our Topic

Mind the Gap: Assessing Temporal Generalization in Neural Language Models

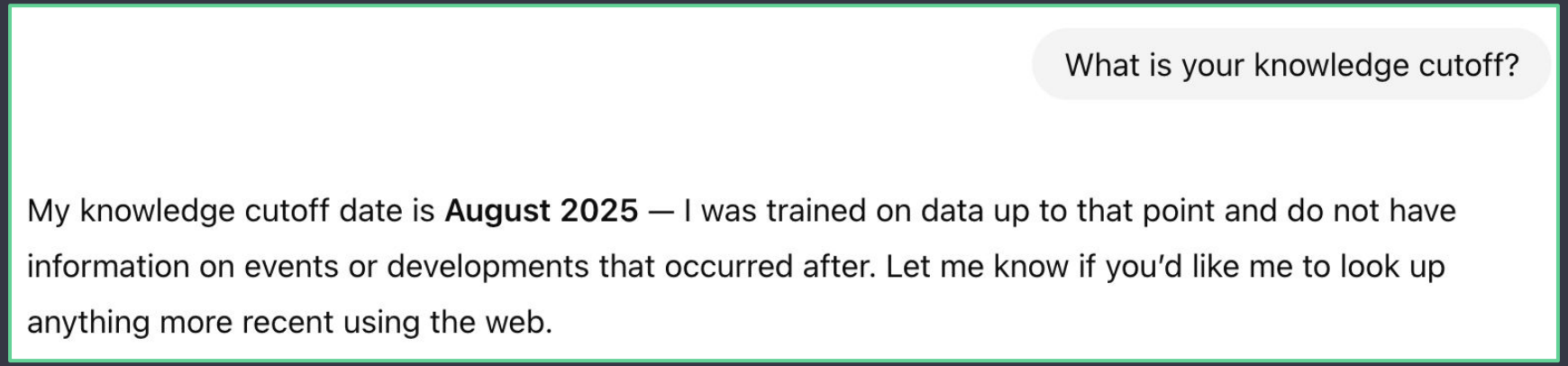
Angeliki Lazaridou[*♡△♠] **Adhiguna Kuncoro**[⋆♡△] **Elena Gribovskaya**[⋆♡△]
Devang Agrawal[◇♡] **Adam Liška**[◇♡] **Tayfun Terzi**[◇] **Mai Gimenez**[◇]
Cyprien de Masson d'Autume[◇] **Tomas Kocisky**[♡] **Sebastian Ruder**[♡]
Dani Yogatama[♣] **Kris Cao**[♣] **Susannah Young**[♣] **Phil Blunsom**[♣♠]
DeepMind, London, UK
{angeliki,akuncoro,egribovskaya}@deepmind.com

Published: October 26, 2021 **Link:** https://arxiv.org/abs/2102.01951

Knowledge Cutoffs in LLMs

World is **dynamic**: Our language is evolving, non-stationary, and open-ended.

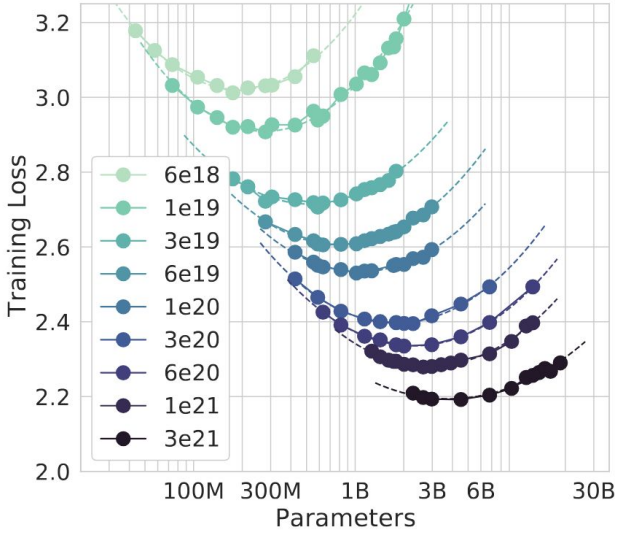
LM paradigm is **static**: Models are trained and evaluated on data from overlapping time periods.



(1) How does this affect my output? **(2)** How can modelers address this?

Motivation

- In recent years, we've gotten very good at building language models
  - More/better data, bigger/better models
  - Focus on perplexity (usually on val split) ➜ tide that lifts other boats



src: https://arxiv.org/pdf/2203.15556

Motivation

- In recent years, we've gotten very good at building language models
  - More/better data, bigger/better models
  - Focus on perplexity (usually on val split) ➔ tide that lifts other boats
  - Also, final models benchmarked on a range of static benchmark tasks
    - E.g., Qwen 3:

- **General Tasks**: MMLU (Hendrycks et al., 2021a) (5-shot), MMLU-Pro (Wang et al., 2024) (5-shot, CoT), MMLU-redux (Gema et al., 2024) (5-shot), BBH (Suzgun et al., 2023) (3-shot, CoT), SuperGPQA (Du et al., 2025)(5-shot, CoT).
- **Math & STEM Tasks**: GPQA (Rein et al., 2023) (5-shot, CoT), GSM8K (Cobbe et al., 2021) (4-shot, CoT), MATH (Hendrycks et al., 2021b) (4-shot, CoT).
- **Coding Tasks**: EvalPlus (Liu et al., 2023a) (0-shot) (Average of HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), Humaneval+, MBPP+) (Liu et al., 2023a), MultiPL-E (Cassano et al., 2023) (0-shot) (Python, C++, JAVA, PHP, TypeScript, C#, Bash, JavaScript), MBPP-3shot (Austin et al., 2021), CRUX-O of CRUXEval (1-shot) (Gu et al., 2024).
- **Multilingual Tasks**: MGSM (Shi et al., 2023) (8-shot, CoT), MMMLU (OpenAI, 2024) (5-shot), INCLUDE (Romanou et al., 2024) (5-shot).

Risks of Static Benchmarking Approach

(1) Does not assess a language model's ability to generalize to future data from beyond their training period (i.e., temporal generalization).

Temporal generalization is crucial to perform well on realistic use cases of language models in the real world.

- Flagging fake news about recent events
- Forecasting stock prices from the latest news articles
- Writing code with updated libraries
- Answering knowledge-intensive questions like "How many people have been infected by COVID-19?", whose answers have evolved with time.

Risks of Static Benchmarking Approach

(1) Does not assess a language model's ability to generalize to future data from beyond their training period (i.e., temporal generalization).

(2) Temporal overlap increases the risk of "test data contamination".

Shown repeatedly in research (and on twitter).

Research Questions

1. To what extent does the current static language modelling practice overestimate performance (i.e., perplexity), compared to the more realistic setup that evaluates LMs on future utterances?
2. How does this temporal degradation manifest in different QA tasks?
3. What is the remedy?
    a. Keeping LMs up-to-date by retraining with new data is expensive in compute and carbon costs
    b. Does size solve this?
    c. “Dynamic evaluation”

Experiment Setup

Goal: Measure how well LMs perform when evaluated on future utterances from beyond their training period

Datasets

- Sources

Dataset	Domain	Time period	#Words per Doc (Average)	Training Size (in GB)	Prop. of CONTROL's Training Data from the Test Period
WMT	News	2007 - 2019	551	22.65	6.3%
CUSTOMNEWS	News	1969 - 2019	491	395.59	34.8%
ARXIV	Scientific text	1986 - 2019	172	0.72	14.5%

- Evaluation period is last 2 years (2018-2019)
- 1K test documents per month
- Control setup ➜ add documents from 2018-2019 into training mix
  - Same dataset size
  - Allows for measures of relative perplexity

Model

Transformer XL

- 287M parameters
- Sequence length 1024
- Custom tokenization

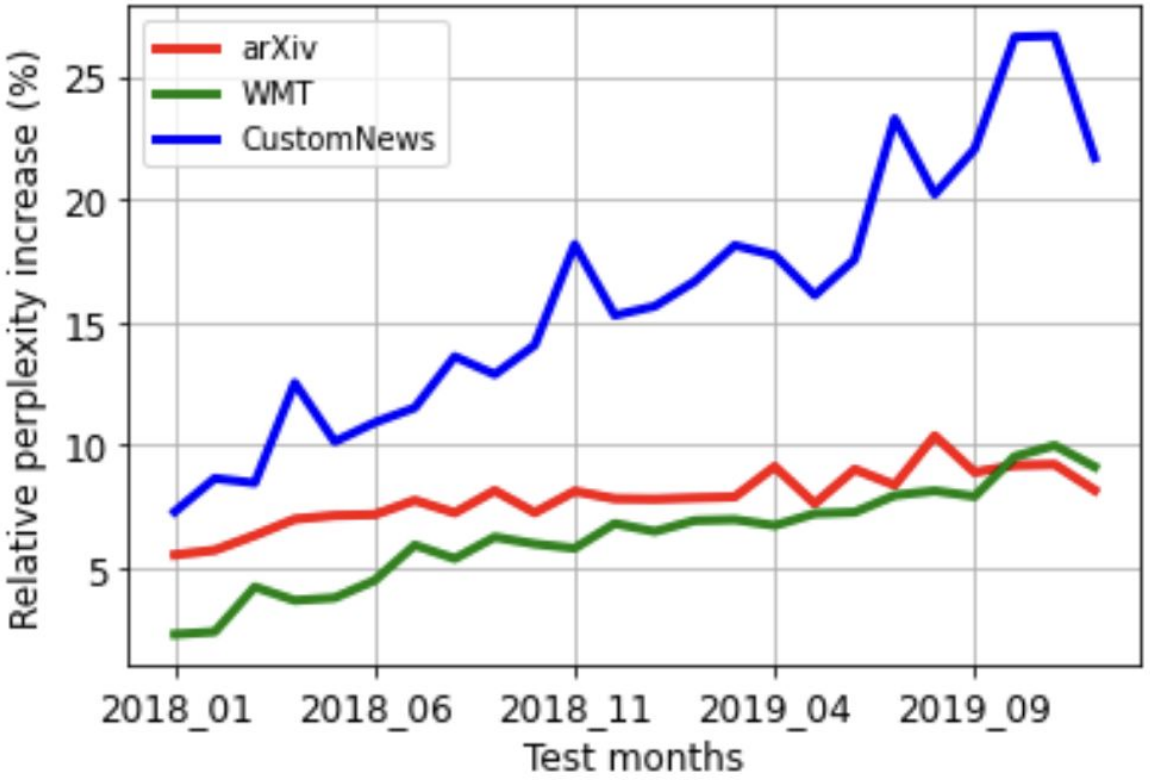
Results

To what extent does the static setup overestimate model performance, compared to evaluating LMs on future utterances?

Setup	WMT	CUSTOM NEWS	ARXIV
CONTROL	21.11	18.38	21.38
TIME-STRATIFIED	22.45	21.33	23.07
Δ, absolute	+1.34	+2.95	+1.69
Δ, relative (%)	6.34	16.04	7.90

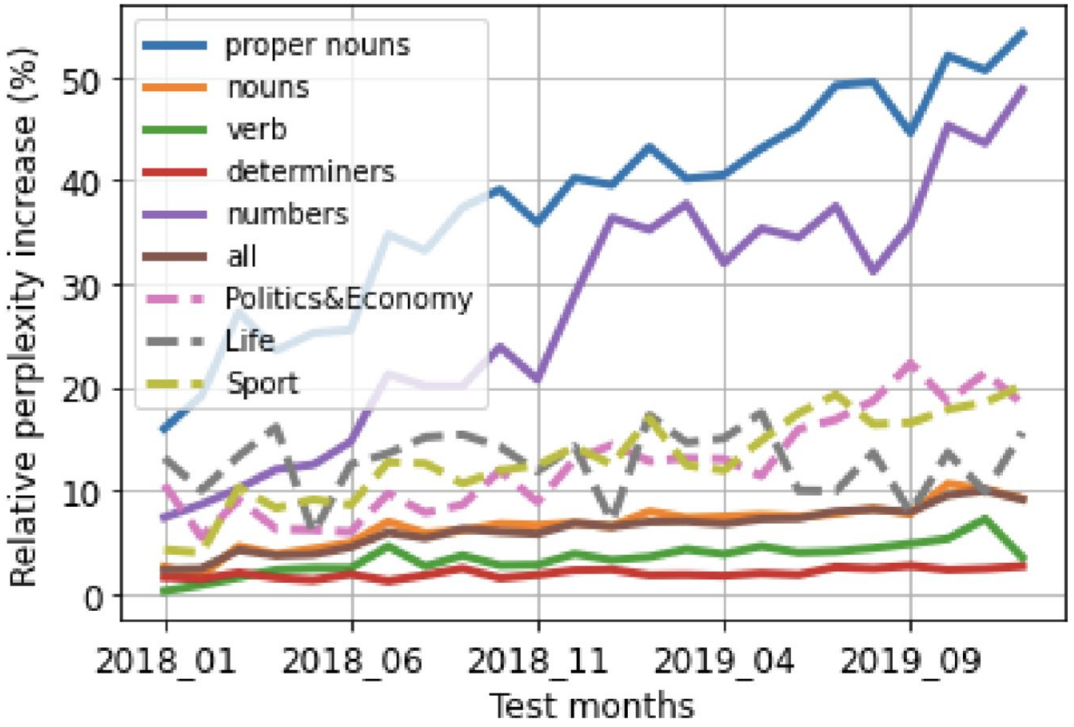
Results

Do LMs perform increasingly worse when predicting utterances further away from their training period?



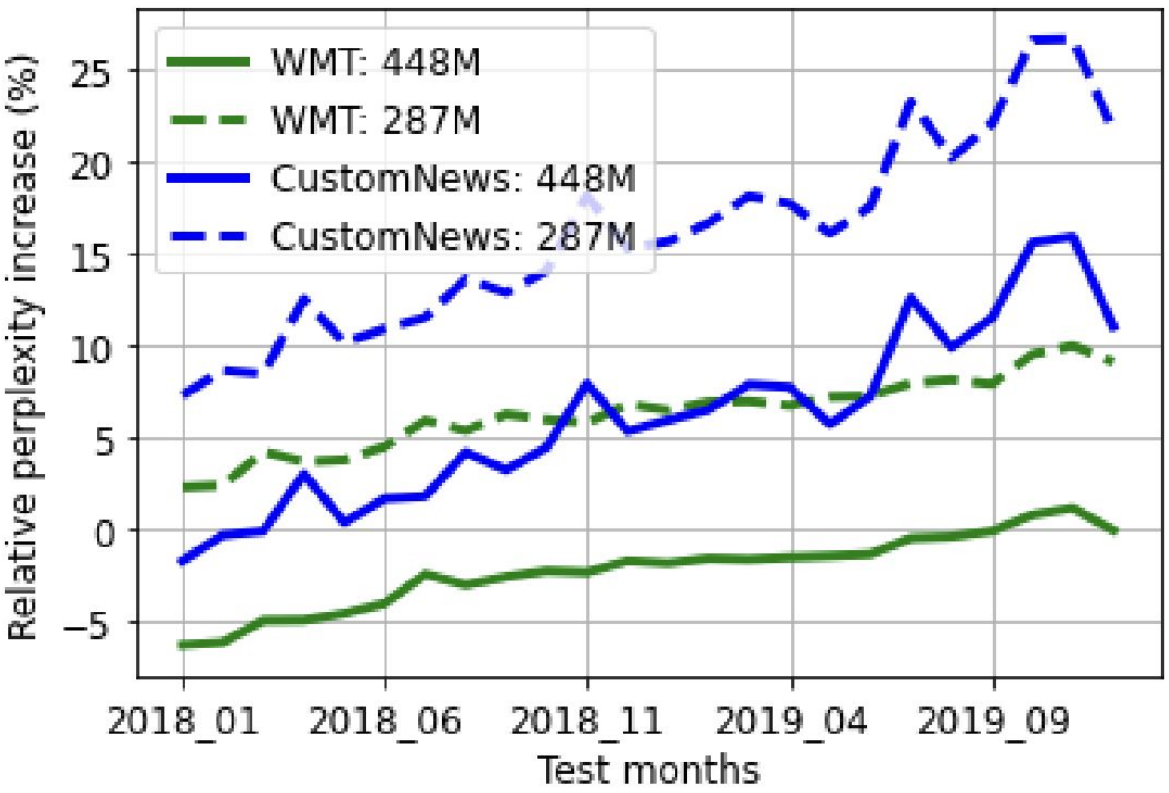
Results

Where does the degradation occur?



Results

Can big models save the day?



Other Findings...

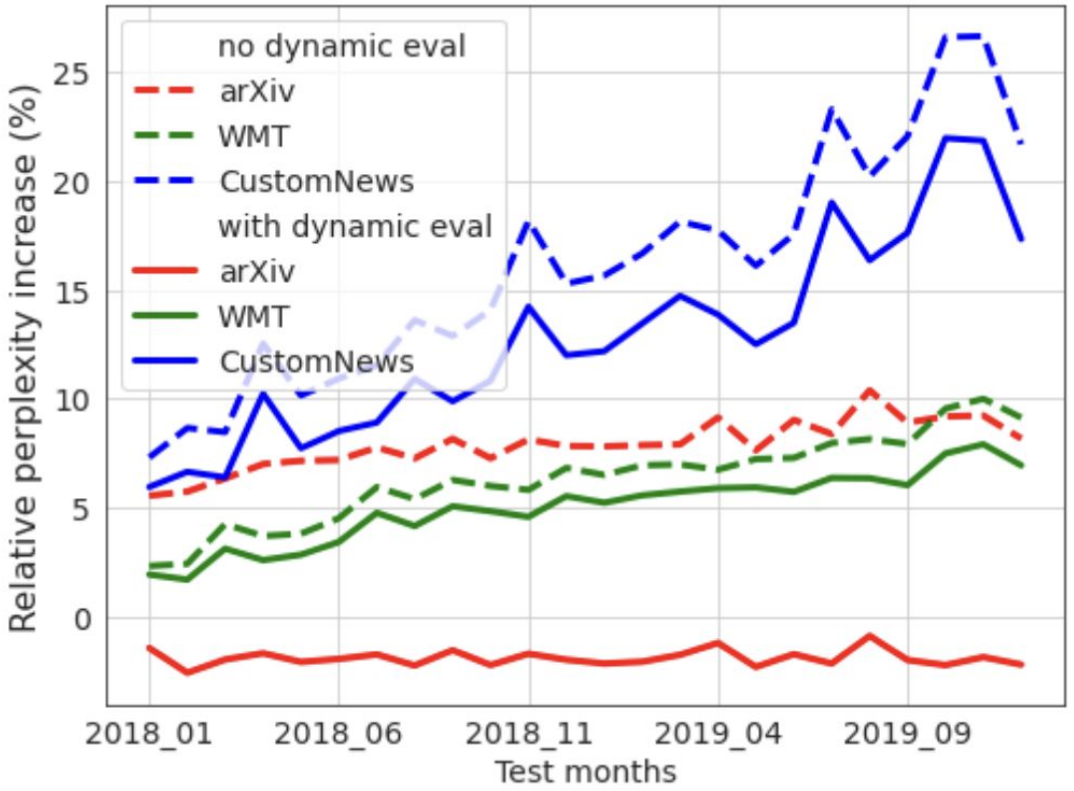
- Closed-book QA about the future is hard (duh)
- Less of a problem for reading comprehension
  - duh
  - Does not absolve RAG

Results

Dynamic evaluation: take gradient steps as new data comes in

Improves perplexity, but:

- Still positive slope
- Catastrophic forgetting



Takeaways

- We should evaluate LMs on their generalization ability to future data, which:
  - circumvents test data contamination.
  - rewards models that generalize beyond the surface patterns of their training data.
  - better reflects how large LMs are used in practical systems.
- LMs deployed far outside of their training period perform substantially worse on downstream tasks that require up-to-date factual knowledge.
  - More tasks, benchmarks, metrics to know how well LMs integrate new information.
- Beyond scaling, need development of adaptive language models that can remain up-to-date with respect to our open-ended and non-stationary world.

Research Project Ideas

- Building on this paper
  - Model (backbone): Link
  - Datasets: Link
- New benchmarks
- New CL approaches:
  - Retrieval
  - Neural memory
  - Etc.
- Scaling laws for temporal generalization
- Bias, security, privacy, etc.