Inference in the Time of GPT*

Mark Steedman *with* Nick McKenna, Tianyi Li, Liang Cheng,
Javad Hosseini, Liane Guillou, and others

* and apologies to G. Garcia Marquez, *Love in the Time of Cholera*

June 2nd, 2023

School of informatics

Which jobs will be changed by LLMs?

- Jobs where Precision (False Positives) matters less than Recall will either be eliminated, or switch to checking LLM positives, increasing productivity.
  - Web Search and Product Recommendation
  - Surveillance
  - Detecting tumours in radiographs
  - Drug molecular prediction
- Tasks where Precision matters will be less affected:
  - Medical advice
  - Legal argument
  - Multi-document Summarization
  - Scientific writing

The Problem of Open Domain QA

- There are too many ways of asking and answering the same question:
- You want to know Who played against Manchester United? The text says:
  - Arsenal beat Manchester United.
  - Manchester United's defeat by Arsenal.
  - Arsenal obliterated Manchester United.
  - etc.
- So if you just build a knowledge graph based on relations found in text (a "Semantic Network"), you won't be able to interrogate it effectively.
- Googling it doesn't always get you what you need:
- "Miles Davis records without Fender-Rhodes piano?" gets you pages about MD with Fender-Rhodes.

Open Domain QA Needs Inference

- Webber, Gardent, & Bos, 2002 give more QA examples, including
  - Query expansion to entailing alternatives;
  - Eliminating spurious answers;
  - Eliminating redundant alternative answers;
  - Detecting equivalence to FAQs;
  - Generating explanatory answers.
- Fan, Gardent, Braud, & Bordes, 2019 Multi-Document summarization:
  - "General Relativity is a theory of Albert Einstein. Einstein developed this theory."
- These are all tasks where precision matters!

The Problem of Inference

- The problem arises from the lack of a usable NL semantics supporting common-sense inference, such as that $\langle team \rangle defeat \langle team \rangle$ entails $\langle team \rangle play\, against \langle team \rangle$, $\langle recording \rangle without \langle musical\ instrument \rangle$ entails $\langle recording \rangle \wedge \neg with \langle musical\ instrument \rangle$, and $\langle theory \rangle of \langle person \rangle$ entails $\langle person \rangle develop \langle theory \rangle$.

- Two solutions:

  1. Use of a pretrained LM, such as BERT or GPT-3, as a latent entailment model, with or without Supervised fine-tuning using an NLI dataset, "Train-of-thought" prompting, "In-context learning", etc.;
  2. Unsupervised induction of an entailment graph from text, using some form of the Distributional Inclusion Assumption (Geffet and Dagan, 2005).

§1. LMs as Latent Entailment Models

- Schmitt and Schütze (2021b,a) claim that fine-tuning BERT/RoBERTa LM using NLI training datasets makes it learn entailment, as assessed on NLI test-sets.
- They embedded entailment pairs in text-like patterns, such as "P, and so Q".
- However, evaluating supervised text inference is an open problem: NLI datasets are:
  - Riddled with artefacts that ML can learn as a proxy;
  - Dominated by paraphrase and selection-bias; and
  - Fail to include false inverses of directional entailments.
- When these artefacts are properly controlled for, Li *et al.* (2022a) fail to support Schmitt and Schütze's claims.
- LLMs seem to model mere non-directional associative similarity.

Very Large LMs as Latent Entailment Models

- Some of our current work investigates Very Large Language Models such as GPT-3 as entailment models (McKenna *et al.*, 2023).

VLLMs appear to memorize the training data, and to organize the memory according to similarity of textual context.

The larger they are, the more literally this is the case (Zhang *et al.*, 2021; Tirumala *et al.*, 2022)

- They excel at tasks where the memorized text actually contains something similar to the question (particularly with respect to nouns and named-entities).

We don't know what GPT has been trained with (Fu *et al.*, 2022).

We believe it may even have been trained on our test data.

VLLMs as Entailment Models

- We embed entailment test pairs in MNLI-like Schmitt and Schütze multiple-choice patterns: eg.:

  "If Google bought YouTube, then Google owns YouTube.
  A) Entailment
  B) Neutral
  C) Contradiction
  Answer:"

- When we test Zero-shot with these patterns, GPT-3 does quite poorly:

Pattern GPT-3.5	Precision	Recall	F1
With Named Entities:	53.4	79.7	64.0
With Entity Types:	53.1	52.9	54.0
With Untyped ABC:	53.03	44.0	48.1
All-positive baseline	50.0	100.0	**66.7**

VLLMs as Entailment Models

- Two-shot "Train-of-Thought" prompt training with a pair of such examples as prefix augmented with an explanation for the decision ("owning is a consequence of buying") prefixed to each MNLI-syle text item adapted from Levy Holt Directional Subset got F1 of **74.3** with full named entities.
- It was still quite bad at rejecting non-entailing inverses.
- Performance again degrades with substitution of type or untyped identifiers for original NEs
- —consistent with the idea that VLLMs memorize the training data, organizing it by similarity of association.

What VLLMs are Really Doing

- Consistent with the idea that VLLMs work by memorizing the training data, performance on NLI datasets is dominated by two biases, which are also characteristic of NLI datasets consisting of premise-hypothesis pairs $P \models H$:
  1. **Veracity (V)**: If H or something like it is likely to have been actually attested in the pretraining data, the model is likely to predict entailment.
  2. **Relative Frequency (F)**: If the entity pair and/or the predicate H is significantly more frequent than P the model is likely to predict entailment.
- Performance degrades for test items that are adversarial to these biases:
- This effect is seen across all language models, and seems to be inherent in word distributions in text and the algorithms for building embedding spaces.
- Our few-shot regime of two entailing and two non-entailing prompts seems enough for the models to pick up this signal as a proxy for entailment.

Bias-Consistency of Test Items

		Levy/Holt					
Model	**Task**	V_C	V_A	*diff.*	F_C	F_A	*diff.*
LLaMA-65B	I	65.5	8.1	**-57.4**	41.4	34.3	**-7.1**
GPT-3.5	I	85.0	10.8	**-74.2**	52.6	41.0	**-11.6**
PaLM-540B	I	79.1	31.5	**-47.6**	62.3	51.7	**-10.6**
LLaMA-65B	I_{TA}	52.1	34.4	**-17.7**	53.1	37.7	**-15.4**
GPT-3.5	I_{TA}	67.1	18.8	**-48.3**	52.2	36.2	**-16.0**
PaLM-540B	I_{TA}	58.1	46.6	**-11.5**	58.2	44.8	**-13.4**

- McKenna *et al.*, 2023: Table 7: LLM performance on subsets where V/F is Consistent/Adversarial to gold labels, measured with AUC norm (0% = random chance performance). Decrease from V_C/F_C toV_A/F_A subsets are presented in the diff. columns. I_{TA} is with type-argument substitution. F here is predicate frequency. (See Poster, Edinburgh Huawei Lab meeting.)

Interim Conclusion

- Large Language Models cannot safely be used on their own, as Latent Entailment models, for NLP tasks where Precision matters.

§2. Combining VLLMs with Entailment Graphs

- Build an unsupervised natural language Knowledge Graph (KG) from large amounts of multiply-authored text by Open Relation Extraction (ORE) of subject-relation-object triples by machine-reading different articles about the same events grounded in the same named-entity tuples.
- Map the KG onto a learned directed Entailment graph (EG), capturing such observations as that if one entity of type team *beat* another entity of that type in one document it's likely that the same two entities will *play against* each other in another.

Entailment Graphs are an efficient representation for knowledge and inference as what Carnap (1952) called Meaning Postulates, what Wittgenstein (1953) seems to have meant by "Meaning as Use", and what Fodor (1975) thought of as content-word semantics.

Entailment Graphs for QA

- EGs can be used for bridging inference from statements in text or Knowledge Graphs to the question in QA.
- They are capable of high precision.


The weakness of EGs is sparsity, arising from Zipf's Law, lowering recall.


Can we use LLMs to compensate for sparsity in EG?

Entailment Graphs

- We have built EGs for English and Chinese, using a variety of methods: (Hosseini *et al.*, 2018, 2019, 2021; Li *et al.*, 2022b).

- Our methods scale: (20M sentences $\Rightarrow$ >200M sentences).

Some Statistics on Unsupervised KG/EG

- Knowledge Graphs built on NewsSpike and NewsCrawl (Hosseini *et al.*, 2021)
  - Newspike is 0.5M multiply-sourced news articles over 2 months, 20M sentences; NewsCrawl is 5.4M articles sourced over 9 years, 256M sentences
  - NewsSpike KG has 326K typed relations, NewsCrawl, 1.05M
  - NewsSpike 29M relation triple tokens (before cutoff); NewsCrawl 729M.
  - NewsSpike 8.5M triple tokens (after cutoff); NewsCrawl 35m.
  - NewsSpike 3.9M triple types (after cutoff); NewsCrawl 13.4m
- We have built working typed global entailment graphs:
  - NewsSpike EG has 346 local typed subgraphs, NewsCrawl, 691
  - NewsSpike 23 subgraphs >1K nodes; NewsCrawl, 161
  - NewsSpike 7 subgraphs >10K nodes; NewsCrawl, 21

Statistics on Chinese KG/EG

- Chinese Knowledge Graphs built on WebHose and CLUE (Li *et al.*, 2021)
  - Webhose is 0.3M multiply-sourced news articles over 1 month, 19M sentences; CLUE is 2.4M articles sourced over 1 year, 193M sentences
  - WebHose KG has 363K typed relations, CLUE, 127M
  - WebHose 35M relation triple tokens (before cutoff); CLUE 792M.
  - WebHose 8.6M triple tokens (after cutoff); CLUE 18.5M.
  - WebHose 1.4M triple types (after cutoff); CLUE 276K
- We have built Chinese working typed global entailment graphs:
  - WebHose EG has 942 local typed subgraphs, CLUE, 384
  - WebHose 149 subgraphs >1K nodes; CLUE, 38
  - WebHose 26 subgraphs >10K nodes; CLUE, 4

Open Domain QA with Entailment Graphs

- Current work (Cheng *et al.*, 2023) uses the Newspike-based English Entailment Graph to augment a Knowledge Graph built from the entire Wikipedia corpus, and performs strongly zero-shot in comparison to LMs including GPT on LAMA-Probe QA datasets.

Dataset		Single Model					Augmented KG	KG with LM backoff		Augmented KG with backoff	
	Rel	Freq	RE	KG	BERT	GPT-3	KG+EG	KG+BERT	KG+GPT	KG+EG+BERT	KG+EG+GPT
Google-RE	PoR	4.6	13.8	19.9	16.1	30.3	27.7	27.7	34.4	30.7	37.0
	DoB	1.9	1.9	7.7	1.0	2.0	8.5	9.8	11.3	9.9	11.3
	PoD	6.8	7.2	14.6	14.0	24.7	26.0	25.9	27.6	29.6	29.7
	Total	4.4	7.6	14.0	10.5	19.0	20.7	20.2	24.3	23.5	**26.0**
TREx	Total	22.0	33.8	29.2	31.5	59.1	35.1	35.4	65.0	64.7	**79.3**

Table 2: Main results on cloze-style QA under zero-shot settings. This table performs the F-1 scores on BERT-large, GPT3, parsed-KG and its augmented versions across the set of evaluation corpora.

- In terms of F1-scores, it looks as though the combination of GPT and KG+EG does better than either alone.

Precision and Recall Analysis

- However, the F-score conceals the fact that the increased recall obtained from the LLM comes at the expense of massive loss of precision:

	Models	Precision@1	Recall	F-score
Single Model	KG	**58.8**	8.5	14.9
	BERT	10.5	10.5	10.5
	GPT-3	19.0	19.0	19.0
Augmented Models	KG+BG	**41.7**	17.0	24.1
	KG+BERT	20.2	20.2	20.2
	KG+GPT	24.3	24.3	24.3
Augmented Models with Backoff	KG+BG+BERT	23.5	23.5	23.5
	KG+BG+GPT	**26.0**	26.0	26.0

Table 3: Precision and Recall for averaged Google RE task.

- Backing-off to LLMs is unsafe for tasks requiring high precision.

§3. Smoothing Entailment Graphs with LLMs

The Problem for the directional Entailment Graph is Zipfian Sparsity of Machine-Reading.

- Can we Smooth Entailment Graphs with non-sparse but non-directional LMs without compromising the directional precision of EG?

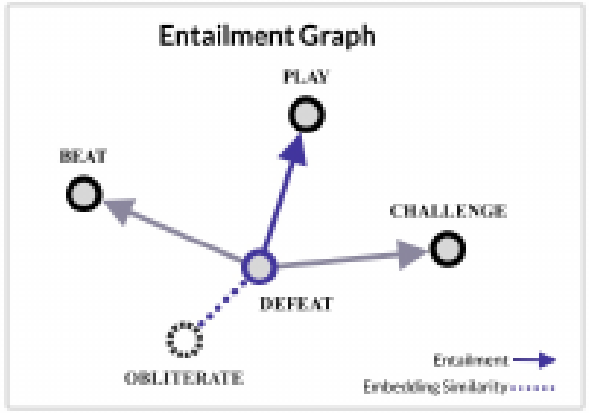
The Idea

- If the P(remise)/Antecedent and/or H(ypothesis)/Consequent are missing from the EG through sparsity, EG loses.
- If we can find P' and/or H' that are in the graph, then:
  - if $P \models P'$ and/or $H' \models H$, and
  - $P' \models H'$ is in the graph, then by transitivity of entailment:
  - $P \models H$, else:
  - $P \not\models H$.
- The idea (McKenna and Steedman, 2022): Iff P and/or H are not in the graph, use LMs to find P' and/or H' that ARE in it.

This technique is orthogonal to earlier backoff to LM for QA.

Smoothing Entailment Graphs with LMs

Step 1: LM embeds all EG predicates.



Question: "Did Arsenal play Man United?"

Text: "Arsenal **obliterated** Man United on Saturday at Emirates Stadium."

Step 2: LM embeds the predicate missing from the EG to find the most similar one.

Step 3: EG completes the directional inference.

Answer: "Yes, Arsenal defeated Man United." ✓

Smoothing Entailment Graphs with LMs

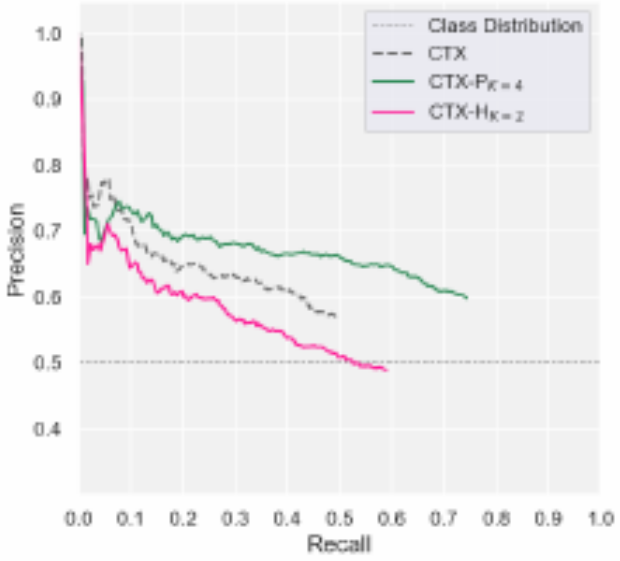
- For P and/or H that is missing in the EG find the K nearest neighbour relations P' and/or H' that are in the EG, using contextualized embedding vectors.
- Then try to establish $P/P' \models H/H'$.
- If $P/P' \models H'/H$, assume $P \models H$

Note that there is no guarantee for LM-KNN P' and/or H' that $P \models P'$ and/or $H' \models H$.

- Nevertheless, we are minimizing the impact on precision of the non-directional LM, unlike the earlier backoff technique.

Smoothing Entailment Graphs with LMs

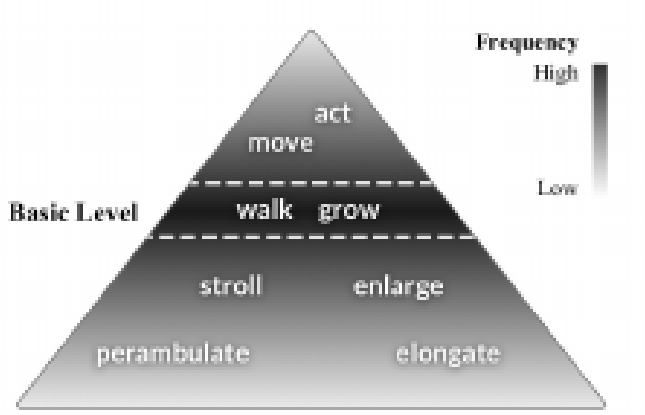
- Smoothing with an LM (RoBERTa) works for P, the antecedent:



- However, LM smoothing is deleterious for H, the consequent.
- Why is LM smoothing asymmetrical for P and H?

Why does LM Smoothing Work At All?

- There is a decrease in frequency with distance on either side of the basic level of "natural kinds" for terms on the hypernym-hyponym dimension of generality-specificity;
- There is also an increase in the number of terms with specificity:



Why is LM Smoothing Asymmetrical?

- This skewed distribution leads to a bias towards more frequent and more general predicates in generating nearest in-graph neighbours P' and/or H' for missing P and/or H using LMs.
- Since specifics are often hyponyms and related generics hypernyms, it is likely that $P \models P'$ obtained in this way.
- However, by the same reasoning, the nearest neighbours H' of H that are most likely to be in the EG are likely to be hypernyms of H, rather than hyponyms, so that it is less likely that $H' \models H$
- McKenna and Steedman (2022) show that smoothing with attested hyper-/hypo-nyms from WordNet has the predicted effect.

This bias is well-known as the source of "translationese", and is also the source of the Relative Frequency Bias (F) we saw in §1 for LLM NLI.

§4. Conclusion

- LLMs work by memorizing the pretraining data, organized by associative similarity, with a Frequency/Generalization gradient.
- The pretraining data is unlikely to include statements of entailments. (Entailments, by definition, "go without saying").
- Fine-tuning LLMs on NLI datasets just seems to pick up artefacts.
- However, you can exploit the generalization gradient of LLM neighborhoods to smooth recall in entailment graphs, without compromising EG precision. . .
- . . . supporting inference needed for NLP tasks like generation, summarization, and Open-Domain QA.

Thanks. . .

- The research was funded in part by ERC grant SEMANTAX and Huawei Edinburgh Laboratory

References

Carnap, Rudolf, 1952. "Meaning Postulates." *Philosophical Studies* 3:65–73. reprinted as Carnap, 1956:222-229.

Carnap, Rudolf (ed.), 1956. *Meaning and Necessity*. Chicago: University of Chicago Press, second edition.

Cheng, Liang, Hosseini, Javad, and Steedman, Mark, 2023. "Complementary Roles of Inference and Language Models in Open-domain QA." In *submitted*.

Fan, Angela, Gardent, Claire, Braud, Chloé, and Bordes, Antoine, 2019. "Using Local Knowledge Graph Construction to Scale Seq2Seq Models to Multi-

Document Inputs." 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)*. 4186–4196.

Fodor, Jerry, 1975. *The Language of Thought*. Cambridge, MA: Harvard.

Fu, Yao, Peng, Hao, and Khot, Tushar, 2022. "How does GPT Obtain its Ability? Tracing Emergent Abilities of Language Models to their Sources." https://yaofu.notion.site/How-does-GPT-Obtain-its-Ability-Tracing-Emergent-Abilities-of-Language-Models-to-their-Sources-b9a57ac0fcf74f30a1ab9e3e36fa1dc.

Geffet, Maayan and Dagan, Ido, 2005. "The Distributional Inclusion Hypothesis and Lexical Entailment." In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. ACL, 107–114.

Hosseini, Javad, Chambers, Nathaniel, Reddy, Siva, Ricketts-Holt, Xavier, Cohen, Shay, Johnson, Mark, and Steedman, Mark, 2018. "Learning Typed Entailment Graphs with Global Soft Constraints." *Transactions of the Association for Computational Linguistics* 6:703–718.

Hosseini, Javad, Cohen, Shay, Johnson, Mark, and Steedman, Mark, 2019. "Duality of Link Prediction and Entailment Graph Induction." In *Proceedings of the 57th Annual Conference of the Association for Computational Linguistics (long papers)*. ACL, 4736–4746.

Hosseini, Javad, Cohen, Shay, Johnson, Mark, and Steedman, Mark, 2021. "Open-Domain Contextual Link Prediction and its Complementarity with Entailment Graphs." In *Findings of the Association for Computational Lingustics: EMNLP*. 1137–1150.

Li, Tianyi, Hosseini, Javad, Weber, Sabine, and Steedman, Mark, 2022a.

"Language Models are Poor Learners of Directional Inference." In *Findings of the Conference on Empirical Methods in Natural Language Processing*. ACL, 903–921.

Li, Tianyi, Li, Sujian, and Steedman, Mark, 2021. "Semi-Automatic Construction of Text-to-SQL Dataset for Domain Transfer." In *Proceedings of the 14th International Conference on Parsing Technology*. 38–49.

Li, Tianyi, Weber, Sabine, Hosseini, Javad, Guillou, Liane, and Steedman, Mark, 2022b. "Cross-lingual Inference with a Chinese Entailment Graph." In *Findings of the Association for Computational Linguistics*. 1214–1233.

McKenna, Nick, Li, Tianyi, Cheng, Liang, Hosseini, Mohammad Javad, Johnson, Mark, and Steedman, Mark, 2023. "Sources of Hallucination by Large Language Models on Inference Tasks." *arXiv preprint arXiv:2305.14552* .

McKenna, Nick and Steedman, Mark, 2022. "Smoothing Entailment Graphs with Language Models." *arXiv preprint arXiv:2208.00318* .

Schmitt, Martin and Schütze, Hinrich, 2021a. "Continuous Entailment Patterns for Lexical Inference in Context." In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 6952–6959.

Schmitt, Martin and Schütze, Hinrich, 2021b. "Language Models for Lexical Inference in Context." In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 1267–1280.

Tirumala, Kushal, Markosyan, Aram, Zettlemoyer, Luke, and Aghajanyan, Armen, 2022. "Memorization Without Overfitting: Analyzing the Training Dynamics of Large Language Models." *Proceedings of the 36th Conference on Neural Information Processing Systems (NIPS)* .

Webber, Bonnie, Gardent, Claire, and Bos, Johan, 2002. "Position Statement: Inference in Question Answering." In *Proceedings of the International Conference on Language Resources and Evaluation*. Las Palmas: ELRA, 24–31.

Wittgenstein, Ludwig, 1953. *Philosophische Untersuchungen (Philosophical Investigations)*. Oxford: Basil Blackwell.

Zhang, Chiyuan, Bengio, Samy, Hardt, Moritz, Recht, Benjamin, and Vinyals, Oriol, 2021. "Understanding Deep Learning (Still) Requires Rethinking Generalization." *Communications of the ACM* 64:107–115.

Zhang, Congle and Weld, Daniel, 2013. "Harvesting Parallel News Streams to Generate Paraphrases of Event Relations." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Seattle: ACL, 1776–1786.