Exploring Language Model Generalization in Low-Resource Extractive QA

Saptarshi Sengupta[1], Wenpeng Yin[1], Preslav Nakov[2], Shreya Ghosh[3], Suhang Wang[1]
[1]Pennsylvania State University, USA
[2]Mohamed bin Zayed University of Artificial Intelligence, UAE
[3]Indian Institute of Technology (IIT) Bhubaneswar, India
{sks6765, wenpeng, szw494}@psu.edu
preslav.nakov@mbzuai.ac.ae, shreya@iitbbs.ac.in

Abstract

In this paper, we investigate Extractive Question Answering (EQA) with Large Language Models (LLMs) under *domain drift*, i.e., can LLMs generalize to domains that require specific knowledge such as medicine and law in a zero-shot fashion without additional in-domain training? To this end, we devise a series of experiments to explain the performance gap empirically. Our findings suggest that: (a) LLMs struggle with dataset demands of closed domains such as retrieving long answer spans; (b) Certain LLMs, despite showing strong overall performance, display weaknesses in meeting basic requirements as discriminating between domain-specific senses of words which we link to pre-processing decisions; (c) Scaling model parameters is not always effective for cross-domain generalization; and (d) Closed-domain datasets are quantitatively much different than open-domain EQA datasets and current LLMs struggle to deal with them. Our findings point out important directions for improving existing LLMs.

1 Introduction

For all their success in general-domain tasks, LLM performance in critical (or closed) reasoning domains such as medicine (Ullah et al., 2024; Nazi and Peng, 2024) and law (Lai et al., 2023) has been shown to be lacking, even on traditional tasks such as Natural Language Inference (Wang et al., 2024). This is the first focus of our paper, i.e., examining the reasons for the poor performance of language models in closed domains.

Our examination focuses on Extractive Question Answering (EQA) (§2), i.e., the task of retrieving a contiguous span of tokens from a passage of text to answer a query based on it. In closed domains, response quality is crucial. Unfortunately, as generative models are prone to hallucination (Huang et al., 2023) or sensitive to the location of the answer span (Liu et al., 2024), they cannot be reliably used (yet) in such domains (Magesh et al., 2024; Pal et al., 2023). As such, extractive retrieval offers better trust in the model response. This is because a model does not need to create new information, but rather locate gold annotated text spans. This is the second focus of this paper, i.e., studying EQA in closed domains.

Self-supervised pre-training on in-domain data is generally utilized as the strategy for garnering domain expertise. However, for esoteric subjects, large-scale training data is seldom available. For example, the corpus curated by Bhattacharjee et al. (2024) discussing among others, *astrophysics* literature, consists of only 66B tokens, a small fraction of the 2T token corpus used by Llama 2 (Touvron et al., 2023b), a general domain model. As such, it is not always possible to perform in-domain pre-training. However, after pre-training in the general domain, a model can be trained to work well for related EQA (c.f. Fig. 1). This leads us to the final focus of this paper, i.e., without additional in-domain fine-tuning, we investigate the extent to which language models can generalize (zero-shot) for closed-domain EQA.

Overall, our main contributions are, (i) We motivate the importance of EQA and the challenges associated with cross-domain generalization by highlighting the poor performance of current models, (ii) Through various experiments, we offer insights into the limitations of current EQA models and complexities of closed-domain datasets that need to be addressed for adaptation across domains, (iii) Finally, we provide recommendations on model usage for EQA in particular, which can be leveraged for other tasks as well.

2 Problem Formulation

EQA has three components (Liu et al., 2019a) I) Context (C): The passage on which the question is based and from which the answer must be drawn;

Proceedings of the 31st International Conference on Computational Linguistics, pages 7106–7126
January 19–24, 2025.

II) Question (Q): The query based on the context; III) Answer (A): The span of context tokens which answers the question. Formally, EQA is defined as,

Definition 1. *Given n tokens, t, as context $C = \{t_1, \dots t_n\}$ and question Q, EQA aims to extract a continuous subsequence of k tokens from the context as the answer A, i.e., $A = \{t_i, \dots t_{i+k}\}$ where $1 \leq i \leq (i+k) \leq n$. In other words, the aim is to learn the function $EQA: f(C, Q) \rightarrow A$*

Figure 1 explains our problem statement. Usually, models are trained on EQA datasets that align with their pre-training data, generally web or open-domain corpora such as Wikipedia. This leads to strong performance when the test set is from the same domain. However, if the domain of the test set is misaligned with the training data, performance degrades sharply. We aim to study why this decline takes place. We test models that are trained on an ID EQA dataset viz., SQuAD (Rajpurkar et al., 2016) and test them on four OOD datasets, DuoRC (Saha et al., 2018), CUAD (Hendrycks et al., 2021), COVID-QA (Möller et al., 2020) and TechQA (Castelli et al., 2020) without further training (zero-shot) and explain the performance gap.

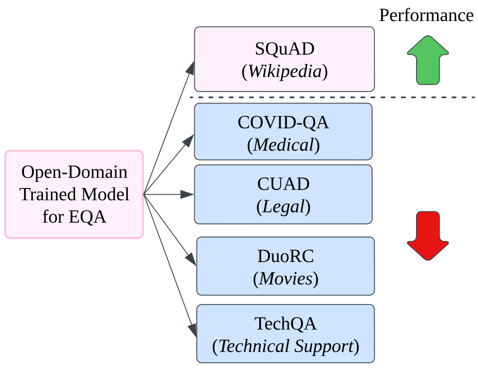

Figure 1: We attempt to explain the performance drop when a model is trained using in-domain (ID) datasets (SQuAD; pink) and tested on ID data (SQuAD) v/s OOD (out-of-domain) data (blue).

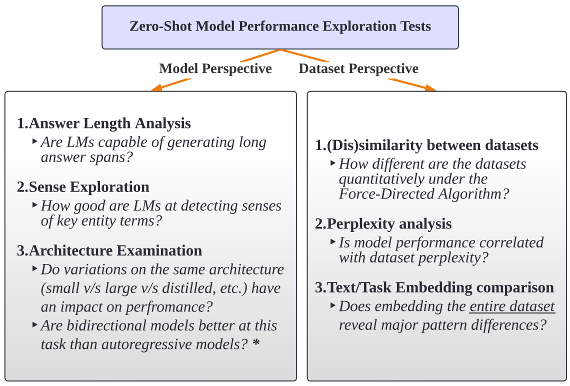

Figure 2: Proposed Experiments. *We provided a detailed analysis of causal LLMs in Appendix F and discuss why they are suboptimal for EQA.

2.1 Evaluating EQA models

Metrics used for evaluating an EQA model are EM (Exact Match) and F1. EM looks for a verbatim match between the predicted and gold answer and, is thus a 0/1 measure. F1 calculates the harmonic mean of the prediction precision (count of shared words between the prediction and gold span/count of words in the prediction) and recall (count of common words/count of words in the gold span).

EM and F1 have a very low tolerance for error due to relying on token overlap and are thus, strict measures. Despite that, prior work has primarily utilized them for reporting scores. This is because creating better, and more nuanced metrics is a non-trivial task. While there have been attempts to this end such as BERTScore (Zhang et al., 2020) and TigerScore (Jiang et al., 2024), these have not yet been widely adopted for EQA. Furthermore, there are active studies (Farea and Emmert-Streib, 2024) investigating the impact of EM/F1, which shows the preference of studies to favour a simpler metric over complex measures.

Assuming we have a better measure, isolating the correct portion of the model's generation (as the answer span) is another challenge. As shown in Figure 10, LLMs produce answers in a variety of formats. Automatically identifying where the correct (answer) sentence lies is again a non-trivial task. As a consequence of this, many works (Labrak et al., 2024; Han et al., 2023; Chen et al., 2023) focus on multiple-choice QA as the generated text is easier to parse.

3 Experiments and Results

We classify our experiments (Figure 2) as, **Model Perspective**, i.e., looking at limitations in the model themselves and **Dataset Perspective**, i.e., examining the complexities of the OOD datasets.

Each experiment, under model-perspective[1] is structured to answer the I) *Hypothesis* (in blue) - What is the main idea being investigated? II) *Motivation* - What is the background/reason for performing this test? III) *Experiment Setting* - How do we test the hypothesis? IV) *Findings* - What are the results of experiments? V) *Key Takeaways* - What

[1]We unify the discussion of the dataset-perspective experiments as they collectively describe a common story.

are the main lessons learned from the experiment?

3.1 Models and Datasets

We test various architectures for EQA, categorized as (i) Non-transformer based, BiDAF (Seo et al., 2017) and QANet (Yu et al., 2018); (i) Transformer-based further categorized as encoder-based, including BERT (and its variants), RoBERTa (Liu et al., 2019b), and decoder-based, including Falcon (Almazrouei et al., 2023), Platypus (Lee et al., 2023), Gemma (Team et al., 2024), Mistral (Jiang et al., 2023) *inter alia*.

Overall, we use five datasets in this study covering general knowledge (SQuAD), COVID-related medical literature (COVID-QA), legal documentation (CUAD), pop-culture and movies (DuoRC) and technical customer support (TechQA). Additional details on the models and datasets are provided in Appendix A.

3.2 Model Perspective

Under this category of experiments, we look at different aspects of a model to determine potential architectural limitations that lead to their poor performance in closed-domain EQA.

3.2.1 Predicted Answer Length Analysis

Motivation We hypothesize that *current EQA models are weak in generating long answer spans matching the distribution of the gold data answer spans*. Closed-domain datasets have longer questions, contexts and answers than SQuAD (c.f. 8). Thus, to answer their questions, a model needs to produce longer spans of text typically not required for simple factoid-based questions found in SQuAD. This leads to our hypothesis that *model performance is impacted due to the inability to produce long answer spans*. In other words, we test whether EQA models overfit the average gold answer length in the training data.

Experiment Setting We examine if architectures specifically trained for EQA still suffer from the same-length generalization drawback. We test if the issue persists for both non-transformer (BiDAF, QANet) and Transformer-based Masked Language Models (MLM) (BERT, RoBERTa). Causal LMs (CLM) are not used here as during inference, we can control generation till the window limit, giving them the flexibility to produce shorter or longer spans.

To test our hypothesis, we determine the *average number of characters* in the predicted answer spans

Domain	Model	Avg. Val. #chars			EM	F1
		True	Predicted	Δ		
SQuAD (Open/General)	BiDAF	18.73	25.31	6.58	65.73	75.98
	QANet		23.74	5.01	26.3	36.81
	BERT		18.18	-0.55	80.95	88.25
	RoBERTa		18.03	-0.7	82.73	90.04
COVID-QA (BioMedical)	BiDAF	93.42	986.73	893.31	17.43	38.3
	QANet		460.81	367.39	0.99	5.76
	BERT		28.81	-64.61	22.39	42.11
	RoBERTa		25.81	-67.61	21.89	40.2
CUAD (Law)	BiDAF	120.07	5261.19	5141.12	5.06	16.81
	QANet		277.32	157.25	0.8	7.01
	BERT		33.55	-86.52	7.72	15
	RoBERTa		19.55	-100.52	4.02	7.7
DuoRC (Movie Plots)	BiDAF	14.27	66.23	51.96	43.99	56.53
	QANet		155.32	141.05	19.03	27.56
	BERT		14.72	0.45	55.59	69.25
	RoBERTa		14.03	-0.24	60.6	74.43
TechQA (Technical QA)	BiDAF	156.79	4302.93	4146.14	0.625	14.56
	QANet		387.2	230.41	0	7.65
	BERT		18.42	-138.37	0	9.19
	RoBERTa		26.89	-129.9	0.625	5.94

Table 1: Zero-Shot Performance in Different Domains. Δ = Average (Predicted - Gold) Answer Span Length

and calculate the difference between it and the average gold span for the given datasets[2]. We use characters instead of tokens to have a consistent scheme across models, as each model uses their own tokenization. A non-negative difference indicates that a model produces spans matching or exceeding the expected gold length and vice versa.

Findings From Table 1, we see how well BERT and RoBERTa approximate the average gold answer length of SQuAD. However, on the OOD datasets, both consistently produce shorter spans leading to negative Δ. BERT breaks the 30-length mark only once (for CUAD) while RoBERTa can barely go beyond 25 characters. Interestingly, we see that BERT produces longer spans for all of the datasets except TechQA. We know that RoBERTa is trained on a much larger corpus and for a longer number of epochs than BERT. Producing consistently smaller spans indicates overfitting on the training corpus. Building on this analysis, we see that apart from SQuAD and DuoRC, BERT performs much better than RoBERTa on COVID-QA, CUAD and TechQA. Although BERT is also trained on Wikipedia, it is considered *undertrained* w.r.t RoBERTa and hence shows inferior performance on those two datasets but better performance in more complex domains.

For the non-transformer models, the clear winner among the two is BiDAF which produces much longer spans than QANet. BiDAF, despite being much smaller than RoBERTa, outperforms it on CUAD and TechQA and comes close to it on COVID-QA (in terms of F1). This shows that

[2] We also look at the distribution of lengths in Appendix B.

larger models do not necessarily equate to better cross-domain generalization.

BiDAF generates much longer answer spans compared to other models. These longer spans cover more context, which helps improve the F1 score as it accounts for overlapping tokens. In domains requiring detailed answers, such as law (CUAD) and technical customer support (TechQA), longer spans are beneficial because complex questions benefit from more thorough responses.

Key Takeaways Here, we observe: (i) Smaller models like BiDAF display competitive performance in zero-shot EQA over larger and more capable models. Thus, *it is not guaranteed that scale equates with domain requirements such as generating longer predictions*; and (ii) When using Transformers, *we should start with BERT as it displays a better tendency to learn new domains*.

3.2.2 Examining Polysemy of Domain Terms

Motivation We hypothesize that *LLMs are weak in detecting senses of relevant domain terms*. Polysemy is a linguistic phenomenon to describe words that take on multiple meanings or *senses*, e.g., *bank* can mean either a financial institution or the portion of land beside a river. For closed-domains datasets, we reason that words with multiple meanings will, on average, show only their *expected* usage for the domain. E.g. we would expect that a term such as *Party* would only take on the *group* meaning rather than the *occasion* meaning for a legal QA dataset (CUAD). Additionally, considering the small number of samples in closed-domain datasets, as opposed to their open-domain counterparts, it would be difficult to expect a dataset on technical customer support (TechQA) to have many/any instances of the *coffee* sense for *Java*. Taking this into account, *we test whether LLMs can discriminate between ID and OOD senses of polysemous domain terms*.

Experiment Setting To test this hypothesis, we create a small dataset of polysemous domain terms that appear in the vocabulary of various contextualized (MLM/CLM) models and the respective datasets along with their associated contexts. We start by tokenizing the contexts of the training split for each dataset and filter out tokens absent from the model's vocabulary, stop words, numbers and punctuation. Sorting the filtered list by frequency, we randomly select five polysemous terms relevant to the domain and retrieve their contexts.

As expected, the datasets usually show only a single sense of a word, and as such we had to rely on external resources to obtain contexts for the other senses. To this end, we manually scrape a well-known website for reliable word definitions and usages, i.e. `vocabulary.com`. In total, we had ten contexts per sense of a given word.

We run each context through the frozen models and extract contextualized embeddings for the polysemous words in our dataset. Using these embeddings, we compute the average cosine similarity between the target word from the same and different sense groups. The overall *logic here is that intra/same-group similarity is expected to be higher, while inter/different-group would be lower*. If not, this can indicate that the models are incapable, to an extent, of discriminating between domain and non-domain senses of words, which in turn contributes to their poor generalization.

Findings Firstly, from Table 2 we see an interesting connection between RoBERTa and Falcon. While both are different styles of models, each reports consistently high scores across all words and senses. Intra/inter-sense similarity scores for Falcon never falls below 0.97 for any dataset. This is concerning as it indicates that it fails to recognize differences in word usage across domains. While RoBERTa also shows higher similarity scores, it discriminates senses to a better extent (lower inter-sense scores).

We question if there is a common link between the two to explain the high similarity scores and find that both models rely on the same tokenization scheme, i.e., byte-level BPE (Byte-Pair Encoding) an algorithm which treats individual bytes as tokens (Sennrich et al., 2016). On the other hand, Platypus, Mistral, and Gemma use better SentencePiece BPE (Kudo and Richardson, 2018) which does not assume that words are space-separated.

The impact of tokenization on performance is a non-trivial issue particularly when dealing with out-of-vocabulary words as shown in Soler et al. (2024) which extends to OOD senses. In fact, as shown by Bostrom and Durrett (2020), straight BPE schemes are inferior to Unigram tokenization (Kudo, 2018) which in turn is not used in isolation, but coupled with SentencePiece to form the basis for tokenizers for newer models. Additionally, it is shown by Bostrom and Durrett (2020) that the latter mode of tokenization leads to better performance for QA tasks. All of this explains why

Dataset	Word	Cell						Expression			Study			Treatment			Host		
	Model	*bio (s1)*	*room (s2)*	*phone (s3)*	*(s1, s2)*	*(s1, s3)*	*(s2, s3)*	*bio (s1)*	*non-verbal (s2)*	*(s1, s2)*	*experiment (s1)*	*room (s2)*	*(s1, s2)*	*medical care (s1)*	*behavior (s2)*	*(s1, s2)*	*organism (s1)*	*organizer (s2)*	*(s1, s2)*
COVID-QA	BERT	0.75	0.73	0.6	0.37	0.4	0.38	0.83	0.65	0.37	0.8	0.66	0.41	0.73	0.59	0.39	0.81	0.61	0.42
	RoBERTa	0.96	0.95	0.91	0.89	0.87	0.89	0.97	0.93	0.87	0.96	0.96	0.89	0.95	0.92	0.87	0.96	0.94	0.88
	BioBERT	0.81	0.83	0.85	0.71	0.74	0.77	0.88	0.82	0.74	0.85	0.88	0.76	0.84	0.92	0.78	0.87	0.84	0.76
	SciBERT	0.73	0.77	0.76	0.57	0.62	0.66	0.8	0.76	0.59	0.77	0.82	0.65	0.76	0.85	0.64	0.77	0.81	0.61
	SenseBERT	0.88	0.89	0.82	0.75	0.73	0.73	0.92	0.86	0.75	0.9	0.89	0.79	0.87	0.88	0.76	0.92	0.86	0.76
	Falcon	0.98	0.98	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.97	0.98	0.98	0.97	0.99	0.97	0.97
	Platypus	0.55	0.53	0.67	0.32	0.31	0.46	0.56	0.5	0.33	0.63	0.48	0.4	0.6	0.54	0.42	0.58	0.41	0.23
	Gemma	0.7	0.86	0.86	0.66	0.71	0.83	0.87	0.86	0.83	0.92	0.77	0.79	0.85	0.93	0.83	0.82	0.88	0.77
	Mistral	0.43	0.53	0.52	0.2	0.25	0.42	0.54	0.53	0.22	0.53	0.4	0.25	0.45	0.51	0.27	0.5	0.45	0.14

Dataset	Word	Party			Agreement			Company			Product			Notice		
	Model	*group (s1)*	*occasion (s2)*	*(s1, s2)*	*contract (s1)*	*understanding (s2)*	*(s1, s2)*	*organization (s1)*	*companionship (s2)*	*(s1, s2)*	*goods (s1)*	*consequence (s2)*	*(s1, s2)*	*announcement (s1)*	*observation (s2)*	*(s1, s2)*
CUAD	BERT	0.69	0.67	0.34	0.83	0.56	0.52	0.73	0.58	0.36	0.68	0.67	0.41	0.72	0.67	0.38
	RoBERTa	0.9	0.95	0.84	0.98	0.91	0.84	0.95	0.91	0.82	0.91	0.94	0.82	0.91	0.93	0.84
	FinBERT	0.75	0.65	0.37	0.84	0.6	0.52	0.79	0.47	0.39	0.59	0.71	0.42	0.75	0.7	0.4
	LegalBERT	0.66	0.78	0.56	0.83	0.71	0.62	0.84	0.73	0.69	0.78	0.79	0.64	0.78	0.8	0.64
	SenseBERT	0.86	0.8	0.7	0.95	0.82	0.83	0.9	0.73	0.71	0.8	0.9	0.75	0.89	0.9	0.77
	Falcon	0.98	0.98	0.97	0.98	0.98	0.98	0.97	0.98	0.97	0.98	0.97	0.97	0.97	0.99	0.97
	Platypus	0.57	0.62	0.4	0.67	0.63	0.56	0.54	0.58	0.17	0.47	0.57	0.36	0.7	0.61	0.38
	Gemma	0.83	0.94	0.84	0.88	0.96	0.9	0.84	0.96	0.85	0.64	0.85	0.72	0.92	0.9	0.89
	Mistral	0.41	0.55	0.17	0.52	0.55	0.35	0.41	0.56	0.26	0.34	0.53	0.24	0.56	0.62	0.2

Dataset	Word	Server			Application						Following			Java			Windows		
	Model	*host (s1)*	*waiter (s2)*	*(s1, s2)*	*program (s1)*	*request (s2)*	*use (s3)*	*(s1, s2)*	*(s1, s3)*	*(s2, s3)*	*reference (s1)*	*pursue (s2)*	*(s1, s2)*	*software (s1)*	*coffee (s2)*	*(s1, s2)*	*software (s1)*	*framework (s2)*	*(s1, s2)*
TechQA	BERT	0.78	0.8	0.48	0.71	0.72	0.53	0.4	0.35	0.41	0.61	0.58	0.31	0.84	0.67	0.48	0.77	0.71	0.33
	RoBERTa	0.94	0.95	0.89	0.93	0.95	0.92	0.87	0.86	0.89	0.91	0.92	0.84	0.94	0.91	0.84	0.93	0.94	0.79
	SciBERT	0.79	0.82	0.71	0.75	0.76	0.74	0.62	0.59	0.66	0.67	0.69	0.55	0.79	0.65	0.59	0.71	0.85	0.62
	SenseBERT	0.88	0.89	0.75	0.86	0.85	0.81	0.74	0.71	0.73	0.72	0.81	0.52	0.9	0.69	0.53	0.83	0.92	0.76
	Falcon	0.98	0.98	0.97	0.97	0.98	0.98	0.97	0.97	0.98	0.99	0.98	0.97	0.99	0.98	0.98	0.99	0.98	0.98
	Platypus	0.63	0.4	0.32	0.52	0.55	0.46	0.16	0.16	0.45	0.6	0.45	0.22	0.77	0.35	0.17	0.73	0.42	0.16
	Gemma	0.79	0.71	0.72	0.61	0.85	0.91	0.61	0.63	0.86	0.55	0.83	0.59	0.8	0.83	0.79	0.93	0.78	0.8
	Mistral	0.5	0.41	0.26	0.44	0.5	0.43	0.25	0.19	0.35	0.66	0.58	0.32	0.59	0.34	0.28	0.57	0.45	0.23

Dataset	Word	Film			Shoot			Hand			Couple			Past		
	Model	*movie (s1)*	*verb (s2)*	*(s1, s2)*	*bullet (s1)*	*record (s2)*	*(s1, s2)*	*anatomy (s1)*	*situation (s2)*	*(s1, s2)*	*pair (s1)*	*few (s2)*	*(s1, s2)*	*time (s1)*	*pass (s2)*	*(s1, s2)*
DuoRC	BERT	0.73	0.65	0.5	0.64	0.59	0.5	0.52	0.5	0.41	0.57	0.56	0.32	0.6	0.48	0.36
	RoBERTa	0.97	0.93	0.9	0.95	0.93	0.92	0.92	0.85	0.85	0.94	0.94	0.87	0.93	0.89	0.86
	SenseBERT	0.91	0.84	0.76	0.81	0.72	0.68	0.7	0.63	0.59	0.84	0.87	0.73	0.73	0.65	0.51
	Falcon	0.98	0.99	0.97	0.98	0.97	0.97	0.98	0.98	0.98	0.98	0.97	0.97	0.99	0.99	0.98
	Platypus	0.59	0.53	0.36	0.45	0.33	0.36	0.61	0.67	0.6	0.69	0.53	0.41	0.5	0.37	0.32
	Gemma	0.87	0.68	0.73	0.79	0.76	0.75	0.96	0.97	0.96	0.79	0.85	0.75	0.97	0.76	0.73
	Mistral	0.69	0.55	0.4	0.48	0.41	0.38	0.4	0.53	0.34	0.64	0.55	0.4	0.45	0.45	0.28

Table 2: Semantic Similarity Results. s(*i*) indicates sense number and intra-group sense similarity. s(*i,j*) indicates inter-group sense similarity. ↑ s(*i*) + ↓ s(*i,j*) indicates a stronger model.

Platypus, Gemma and Mistral show greater semantic awareness than Falcon and RoBERTa and by extension why Platypus, on average, outperforms Falcon across all datasets (c.f. Table 9). Finally, while encoders show higher similarity scores than decoders it does not imply that they are worse at sense discrimination. BERT consistently shows high intra-sense and low inter-sense similarity indicating a greater degree of sense separation.

Key Takeaways The main findings here are: (i) Tokenization matters - Models employing *Sentence-Piece + Unigram tokenization demonstrate overall better performance and sense discrimination*; (ii) Despite their overall poor performance (c.f. Appendix F), *CLMs exhibit a strong degree of sense awareness* in closed domains; (iii) *BERT should be used as a starting point for linguistically challenging datasets*, as it shows strong sense discrimination in closed domains occasionally beating SenseBERT, specifically trained to be better at such tasks.

3.2.3 Architecture Examination

Motivation As discussed previously (sec. 1), scaling LLMs does not always lead to SOTA performance on every task. Accordingly, we test whether *scaling LMs is a definitive solution for zero-shot generalization or not*. We also look at how preprocessing decisions impact generalization.

Experiment Setting Here, we study how variations in the model architecture impact EQA performance. This includes analysis across varying attention heads, hidden dimensions and number of layers. Also, we look at how tokenization decisions including word masking and text normalization contribute to the performance drop.

As it is computationally intractable to analyze every LLM, we perform a deep dive on one particular model BERT to glean an understanding of the performance drop. Additionally, we use BERT due to the availability of a variety of pre-trained configurations as released by Turc et al. (2019).

For each pre-trained variant[3] with a given combination of layers (L), attention heads/hidden layer size ($A|H$), we first fine-tune them on SQuAD (Fig. 1) and then zero-shot test on the OOD datasets to gauge the impact of model size on performance.

We also investigate how pre-processing decisions impact performance. Under this bracket, we test whether *whole-word-masking* (WWM), i.e., masking ALL of the tokens associated with a word makes any difference over *word-piece-masking* (WPM), i.e., masking some or none of the tokens of a word randomly. It can be reasonably expected that closed-domain datasets will contain information on entities typically not observed in the general domain. Thus, we also test whether *case preserving* models perform better than *case-independent* ones since it has been established that models exploit casing to identify entities (Das and Paik, 2022).

[3] See Appendix C.1 for each configuration tested.

	Dataset (EM ǀ F1)				
Model	SQuAD	COVID-QA	DuoRC	TechQA	CUAD
$\text{BERT}_{\text{Base}}$[L=12 ǀ A=12 ǀ H=768]	81.30 ǀ 88.58	20.8 ǀ 39.3	54.16 ǀ 66.82	2.93 ǀ 6.58	2.44 ǀ 5.26
$\text{BERT}_{\text{Large}}$[L=24 ǀ A=16 ǀ H=1024]	84.03 ǀ 91.1	22.29 ǀ 40.26	53.99 ǀ 67.52	2.58 ǀ 7.08	2.34 ǀ 4.84

Table 3: Impact of using Cased Models with WPM.

Model	Dataset (EMǀF1)				
$\text{BERT}_{\text{Large}}$	SQuAD	COVID-QA	DuoRC	TechQA	CUAD
Uncased	86.7 ǀ 92.8	22.39 ǀ 38.61	58.2 ǀ 71.47	2.59 ǀ 8.97	2.2 ǀ 4.1
Cased	86.7 ǀ 92.9	21.25 ǀ 37.74	57.29 ǀ 70.49	3.23 ǀ 7.6	0.43 ǀ 1.66

Table 4: Impact of text normalization with WWM.

Findings A: (Word-Piece Masking) (WPM) Table 5 provide the results of our zero-shot EQA trials for BERT models trained with word-piece masking and using uncased text. As can be seen, BERT follows a consistent trend for ID SQuAD with performance improving across all axes of model size, i.e., A, H and L. Additionally, it benefits from training on cased-text as both $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$ report the best scores on SQuAD in Table 3.

Curiously, we find that results from DuoRC follow a very similar trend with SQuAD. We explain this by the fact that as DuoRC samples are drawn from Wikipedia, they align more with BERT's training data which in turn allows performance to scale across each axis of model size. Unfortunately, a similar trend is not observed for the other closed-domain datasets, i.e., scaling A, H and L does not always lead to improvements when the datasets differ widely as highlighted in Table 5. We reason that this behaviour is caused by ID fine-tuning strongly aligning the base model with the domain to the extent that any architectural modifications do not yield appreciable gains on OOD datasets.

L	A ǀ H	EM ǀ F1 (SQuAD)	EM ǀ F1 (OOD)
8	12 ǀ 768	80.68 ǀ 88.38	21.3 ǀ 38.02
10		81.33 ǀ 88.66	19.47 ǀ 35.45
12	8 ǀ 512	79.74 ǀ 87.41	1.61 ǀ 6.86
	12 ǀ 768	80.9 ǀ 88.2	1.61 ǀ 6.36
12	12 ǀ 768	80.9 ǀ 88.2	2.46 ǀ 4.63
24	16 ǀ 1024	83.49 ǀ 90.6	0.78 ǀ 3.56

Table 5: Impact of scaling L for COVID-QA (top), $A|H$ for TechQA (middle), both for CUAD (bottom); SQuAD scores are in the third column. See Figure 8 for scores from all configurations.

Findings B: Whole Word Masking (WWM) There are only two models to examine under this masking strategy, i.e., $\text{BERT}_{\text{LARGE}}$ trained with and without cased text. Comparing scores from Table 3 and 4 we see that cased-text in combination with whole-word masking leads to improved scores for SQuAD, DuoRC and TechQA. This makes sense as closed domains discuss various entities and have a processing scheme recognizing that is beneficial.

Overall, we find that WWM tends to outperform WPM. Such an observation was also made by Joshi et al. (2020) who found that span (in our case whole words) prediction as opposed to individual tokens is a more challenging task and leads to stronger models. Finally, we see that the uncased variants of this scheme display the best performance overall. We reason that this is because the models are more sensitive to the choice of masking than text normalization, e.g., irrespective of capitalization, terms such as *new york* will convey the same information.

Key Takeaways The key insights here are: (i) Although Bi-directional models as BERT are more suitable for EQA, *it is not guaranteed to see improvements in closed-domains simply by increasing model scale*; (ii) *WWM models should be preferred over WPM models for cross-domain EQA*; (iii) If a WWM variant is unavailable, consider using *uncased models as they tend to display better performance across domains*; (iv) *When dealing with long-context datasets, consider using Bi-directional models over CLMs (c.f. Appendix F)* as they do not face similar issues as the latter.

3.3 Dataset Perspective

While architecture and training decisions impact cross-domain performance, they cannot be solely accountable for the ID-OOD performance discrepancy. As evidenced by the closed-domain datasets used in this study (§Appendix A), the number of samples, along with their average answer/context length, provide initial clues for the disparity as ID models are unaccustomed to such instances. Therefore, we compare OOD datasets with their ID counterpart to see exactly how different they are. We do this through established *quantitative* measures that capture insights from the entire dataset by giving global feedback rather than a per-sample basis qualitative examination.

3.3.1 Impact of Dataset Similarity on Transferability

Here, we quantify the disparity between ID SQuAD and OOD datasets using two techniques viz., Force-Directed Algorithm (FDA) (Fruchterman and Reingold, 1991) and dataset embeddings (Vu et al., 2020). Through these measures, we gauge how different OOD datasets are w.r.t SQuAD which aids us in understanding the performance drop better.

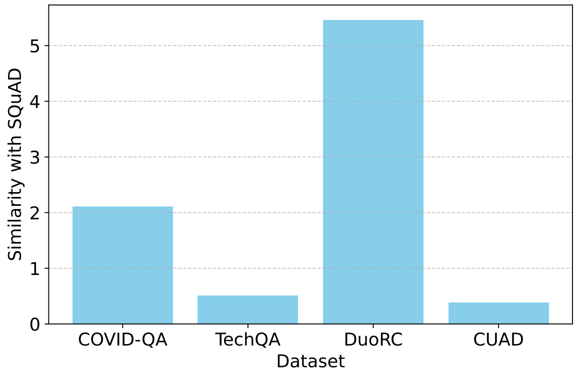

Figure 3: FDA Plot. Each bar represents FDA similarity between SQuAD and the corresponding OOD dataset.

FDA Su et al. (2019) and Talmor and Berant (2019) study Multi-Task QA and use FDA, a graph construction method, to determine dataset relatedness. According to it, the similarity of a dataset with an OOD one is given as $\frac{2P_{ij}}{P_j}$, where D_i is a dataset, P_j is the F1 when training and evaluating on D_j and P_{ij} is the F1 when training on D_i and testing on D_j. We visualize dataset similarity in Figure 3.

Analysis Figure 3 ranks the OOD datasets in order of similarity with SQuAD as,

DuoRC > COVID-QA > TechQA > CUAD

This makes sense seeing as DuoRC is sampled from Wikipedia, i.e., the same as SQuAD and deals with overall "simpler" topics (movie plots) as compared to the other datasets. Dataset characteristics such as longer sample lengths and complex subject matter explain the relative ranking of TechQA and CUAD while COVID-QA strikes a middle ground between the two. This ranking is further reinforced by the overall performance (F1) of the models, i.e.,

DuoRC > COVID-QA > TechQA > CUAD (BERT)

DuoRC > COVID-QA > CUAD > TechQA (RoBERTa)

Although there are slight deviations in rank, the overall trend places DuoRC and COVID-QA at the higher end of the similarity spectrum to SQuAD and TechQA/CUAD at the lower end, explained by the models' poor performance.

`TEXT`/`TASK` Embedding Vu et al. (2020) propose two embedding methods to capture *task (dataset) semantics*, i.e., `TEXT` and `TASK` embedding (c.f Appendix G for details). `TEXT` embedding captures semantics about the entire dataset while `TASK` embeddings determine the correlation between different tasks. If the domains/tasks of the two datasets are similar, their `TEXT`/`TASK` embeddings will be similar.

We investigate how well each embedding captures dataset semantics. For each dataset, we learn `TASK` and `TEXT` embeddings and compute cosine similarity using them between SQuAD and its OOD counterparts. The idea here is to quantify how much ID and OOD datasets differ with the hypothesis being that the two will have, on average, low cosine similarity scores (greater dissimilarity). Following Vu et al. (2020) we use uncased $\text{BERT}_{\text{BASE}}$ to extract `TEXT` and `TASK` embeddings.

Following Turc et al. (2019), we also establish a non-dense embedding baseline by representing each dataset pair as frequency vectors of the top 100 common unigrams and computing Spearman correlation between them.

Embedding Type	Target Dataset			
	COVID-QA	TechQA	DuoRC	CUAD
Common-Term Frequencies (Sparse)	-0.23	-0.27	-0.67	-0.5
`TEXT` Embedding (Dense)	0.9	0.82	0.92	0.86
`TASK` Embedding (Dense)	0.77	0.64	0.86	0.63

Table 6: Cosine similarity and Spearman Correlation scores. Each entry indicates the corresponding category score between SQuAD and each OOD dataset. Higher scores indicate greater relatedness.

	Target Dataset			
Layer	COVID-QA	TechQA	DuoRC	CUAD
1	0.9	0.72	0.97	0.74
2	0.92	0.75	0.97	0.73
3	0.9	0.71	0.96	0.78
4	0.92	0.69	0.97	0.74
5	0.91	0.7	0.96	0.73
6	0.88	0.74	0.95	0.76
7	0.88	0.69	0.94	0.72
8	0.89	0.7	0.91	0.72
9	0.83	0.64	0.92	0.68
10	0.85	0.7	0.94	0.68
11	0.37	0.69	0.84	0.22
12	8.16E-12	1.88E-09	3.91E-11	7.61E-11
Avg.	0.77	0.64	0.86	0.63

Table 7: Layerwise `TASK` Embedding similarity against SQuAD. We observe that domain divergence takes place mostly in the last 2 layers.

Analysis Spearman correlation scores from Table 6 show a completely different order than FDA as,

DuoRC > CUAD > TechQA > COVID-QA

This is due to count-based vectors failing to capture deeper dataset semantics. For example, CUAD, while drastically different from SQuAD in subject matter, use wording typically found in open-domain documents. As such, relying on unigrams alone is bound to pick up on these characteristics reflected in the overall ranking as above.

`TEXT` and `TASK` embeddings reveal a similar pattern as FDA as,

DuoRC > *COVID-QA* > *CUAD* > *TechQA* (`TEXT`)
DuoRC > *COVID-QA* > *TechQA* > *CUAD* (`TASK`)

Although there exists a slight difference in order, the overall sequence indicates a strong degree of agreement with BERT and RoBERTa's F1 scores and in turn provides further explanation for the performance discrepancy.

`TASK` embeddings produce layer-by-layer representations. This allows us to investigate fine-grained changes during learning. Table 7 shows the similarity scores between each layer's `TASK` embedding for each OOD dataset w.r.t SQuAD. Similarity scores indicate that the models learn EQA well till layer 10 across each domain. However, *in the last layers is where domain divergence manifests*. In other words, we reason that in the last layers, the signal from the domain/dataset becomes so strong as to overpower what the model learned overall about the task which in turn leads to their observed poor performance.

3.3.2 Model Perplexity v/s Performance

While typically used for evaluating language models, *perplexity* (PPL) can be extended to evaluate any dataset by converting the samples into a unified representation akin to any unstructured training corpora (§Appendix I). We convert each QA dataset into a corpus by combining all the training contexts and questions into a list of unlabeled samples. Using this converted dataset, we compute a model's PPL on it and correlate it with its performance. The logic here is straightforward; *higher perplexity should correspond to lower performance*. As text in closed domains is qualitatively more complex, it is expected that a model will face difficulty in reasoning over them, leading to higher PPL.

We choose BERT and RoBERTa and two autoregressive LLMs (Platypus and Falcon) for this test. To keep the comparison fair between the two classes of models, we use only the answerable questions from DuoRC, TechQA and CUAD.

Analysis As explained before, our hypothesis is that *model perplexity on a dataset is inversely proportional to its performance*. In other words, higher the PPL. lower is its performance. This makes sense since a model's performance is linked with its ability to comprehend the text. Trend lines for all four models (Figure 4) affirm our hypothesis. For example, BERT and RoBERTa report the highest PPL. and corresponding lowest scores for TechQA. A similar observation holds for Falcon and Platypus for CUAD. Although there exist outliers, overall, we see that for datasets with lower PPL. each model shows strong performance. This observation is displayed sharply by the LLMs, as they are overall much better at language modelling, as we can see from the location of SQuAD and DuoRC in the plots for Falcon and Platypus.

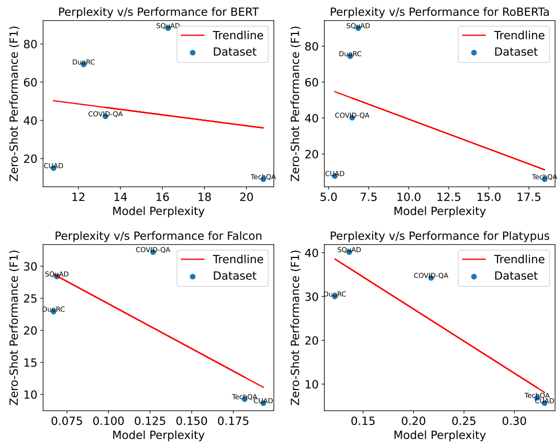

Figure 4: Scatter plot with trend line between model perplexity and performance (F1). Pearson correlation between F1 and PPL. (clockwise from top left) for BERT: -0.17, RoBERTa: -0.48, Falcon: -0.77, Platypus: -0.9.

4 Related Work

Generalization in LLMs. LLMs performance on unseen domains (Ramponi and Plank, 2020) remains an active area of study. Recently, Yang et al. (2024) and Leng and Xiong (2024) examined the effects of fine-tuning on generalization. The issue here is that they examine generalization *after* training, which is not always possible due to data scarcity. Mai et al. (2024) study LLM generalization on a synthetic domain of "gibberish" language. Although novel, their finding's impact on real-world domains remains unclear.

QA Analysis Recent studies by Pezeshkpour and Hruschka (2024) and Khatun and Brown (2024) examine the limitations of LLMs for Multiple Choice QA. Although they do not consider EQA, they provide interesting insights such as LLMs being sensitive to the location of answer choices, etc. The work done by Kamalloo et al. (2023) is in a similar direction as ours, focusing on the limitations of existing metrics for evaluating extractive or generative QA systems. We find that the closest study to ours is by Miller et al. (2020) who create new test sets to determine if models trained on SQuAD overfit to it. However, with a reported maximum drop in F1 of 17.4 their datasets are far less challenging to stress test LLMs. Also, they overlook architecture issues to explain the performance gap. **Related Experiments.** Varis and Bojar (2021) examine how Transformers struggle with length generalization if they are trained solely on samples of a given length, potentially indicating overfitting. However, they do not examine other neural architectures as us (c.f. 3.2.1) thus limiting the scope of their claims. Yenicelik et al. (2020) study how BERT organizes polysemous words in embedding space. However, i) they neglect OOD senses and, ii) while their findings explain how BERT views contexts, they do not provide any actionable insights to using models in such settings as ours.

5 Conclusion

In this paper, we examine why LMs perform poorly on zero-shot EQA in closed-domains. We consider reasons from both dataset and model perspective. Our findings reveal inadequacies in the current generation of models that need to be addressed to realize true domain generalization. Additionally, we also examine the complexities of OOD datasets which a model needs to be made aware of apriori before they can learn to generalize.

Limitations

For our polysemy tests, a point of concern might be the number of samples in our dataset. While we would have liked to use more instances, we are limited by the size of the domain dataset and consequently the number of samples we can collect for each polysemous term. That said, we believe that our findings are overall still valid as we prioritize sample quality over quantity. Additionally, for the model scale test, while it would be ideal to test even more models, as explained before, it is intractable to test all configurations for every possible model. As such, we decided to thoroughly examine one particular architecture.

Ethics Statement

As our study does not deal with sensitive information or involve multiple GPUs for training, we believe that the ethical implications of our work are limited, if any.

Acknowledgment

This material is based upon work supported by, or in part by, the Army Research Office (ARO) under grant number W911NF-21-10198 and Cisco Faculty Research Award.

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2vec: Task embedding for meta-learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. 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 3615–3620.

Bishwaranjan Bhattacharjee, Aashka Trivedi, Masayasu Muraoka, Muthukumaran Ramasubramanian, Takuma Udagawa, Iksha Gurung, Rong Zhang, Bharath Dandala, Rahul Ramachandran, Manil Maskey, et al. 2024. Indus: Effective and efficient language models for scientific applications. *arXiv preprint arXiv:2405.10725*.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Michael McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej

Sakrajda, Avi Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. 2020. The TechQA dataset. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1278, Online. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. 2023. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024. Adapting large language models via reading comprehension. *Preprint*, arXiv:2309.09530.

Sudeshna Das and Jiaul Paik. 2022. Resilience of named entity recognition models under adversarial attack. In *Proceedings of the First Workshop on Dynamic Adversarial Data Collection*, pages 1–6.

Tobias Deußer, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa. 2023. Informed named entity recognition decoding for generative language models. *arXiv preprint arXiv:2308.07791*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Amer Farea and Frank Emmert-Streib. 2024. Experimental design of extractive question-answering systems: Influence of error scores and answer length. *Journal of Artificial Intelligence Research*, 80:87–125.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

Thomas MJ Fruchterman and Edward M Reingold. 1991. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164.

Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressem. 2023. Medalpaca–an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *NeurIPS*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. 2024. TIGERScore: Towards building explainable metric for all text generation tasks. *Transactions on Machine Learning Research*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77.

Daniel Jurafsky and James H Martin. 2019. Speech and language processing 3rd edition draft.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

Aisha Khatun and Daniel G Brown. 2024. A study on large language models' limitations in multiple-choice question answering. *arXiv preprint arXiv:2401.07955*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *Preprint*, arXiv:2402.10373.

Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S Yu. 2023. Large language models in law: A survey. *arXiv preprint arXiv:2312.03718*.

Ariel N Lee, Cole J Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms. *arXiv preprint arXiv:2308.07317*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Yongqi Leng and Deyi Xiong. 2024. Towards understanding multi-task learning (generalization) of llms via detecting and exploring task-specific neurons. *arXiv preprint arXiv:2407.06488*.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. Sensebert: Driving some sense into bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.

Zekun Li, Wenxuan Zhou, Yao-Yi Chiang, and Muhao Chen. 2023. Geolm: Empowering language models for geospatially grounded language understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5227–5240.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Shanshan Liu, Xin Zhang, Sheng Zhang, Hui Wang, and Weiming Zhang. 2019a. Neural machine reading comprehension: Methods and trends. *Applied Sciences*, 9(18):3698.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Alexander Ly, Maarten Marsman, Josine Verhagen, Raoul PPP Grasman, and Eric-Jan Wagenmakers. 2017. A tutorial on fisher information. *Journal of Mathematical Psychology*, 80:40–55.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. 2024. Hallucination-free? assessing the reliability of leading ai legal research tools. *arXiv preprint arXiv:2405.20362*.

Huu Tan Mai, Cuong Xuan Chu, and Heiko Paulheim. 2024. Do llms really adapt to domains? an ontology learning perspective. *arXiv preprint arXiv:2407.19998*.

Prabir Mallick, Tapas Nayak, and Indrajit Bhattacharya. 2023. Adapting pre-trained generative models for extractive question answering. *arXiv preprint arXiv:2311.02961*.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. 2020. The effect of natural distribution shift on question answering models. In *International conference on machine learning*, pages 6905–6916. PMLR.

Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. 2020. COVID-QA: A question answering dataset for COVID-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Zabir Al Nazi and Wei Peng. 2024. Large language models in healthcare and medical domain: A review. *Informatics*, 11(3).

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2023. Med-HALT: Medical domain hallucination test for large language models. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 314–334, Singapore. 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. *arXiv preprint arXiv:2306.01116*.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693, Melbourne, Australia. Association for Computational Linguistics.

Mourad Sarrouti, Carson Tao, and Yoann Mamy Randriamihaja. 2022. Comparing encoder-only and encoder-decoder transformers for relation extraction from biomedical texts: An empirical study on ten benchmark datasets. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 376–382.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

Aina Garí Soler, Matthieu Labeau, and Chloé Clavel. 2024. The impact of word splitting on the semantic content of contextualized word representations. *arXiv preprint arXiv:2402.14616*.

Dan Su, Yan Xu, Genta Indra Winata, Peng Xu, Hyeondey Kim, Zihan Liu, and Pascale Fung. 2019. Generalizing question answering system with pre-trained language model fine-tuning. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 203–211, Hong Kong, China. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy. Association for Computational Linguistics.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Ehsan Ullah, Anil Parwani, Mirza Mansoor Baig, and Rajendra Singh. 2024. Challenges and barriers of using large language models (llm) such as chatgpt for diagnostic medicine with a focus on digital pathology–a recent scoping review. *Diagnostic pathology*, 19(1):43.

Dusan Varis and Ondřej Bojar. 2021. Sequence length is a domain: Length-based overfitting in transformer models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8246–8257, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926, Online. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Nancy Xin Ru Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The COVID-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Yuxia Wang, Minghan Wang, and Preslav Nakov. 2024. Rethinking STS and NLI in large language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 965–982, St. Julian's, Malta. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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.

Peng Xu, Davis Liang, Zhiheng Huang, and Bing Xiang. 2021. Attention-guided generative models for extractive question answering. *arXiv preprint arXiv:2110.06393*.

Haoran Yang, Yumeng Zhang, Jiaqi Xu, Hongyuan Lu, Pheng-Ann Heng, and Wai Lam. 2024. Unveiling the generalization power of fine-tuned large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 884–899, Mexico City, Mexico. Association for Computational Linguistics.

David Yenicelik, Florian Schmidt, and Yannic Kilcher. 2020. How does BERT capture semantics? a closer look at polysemous words. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 156–162, Online. Association for Computational Linguistics.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023. Controlled text generation with natural language instructions. In *International Conference on Machine Learning*, pages 42602–42613. PMLR.

A Datasets and Models

Here, we describe all the resources used in this paper, i.e. the models as well as the datasets.

A.1 Datasets/Domains Studied

For this study, we utilize five datasets covering a diverse set of domains, as described below. Statistics of the dataset are summarized in Table 8.

- SQuAD (Stanford Question Answering Dataset), generally regarded as a benchmark for EQA, is a high-quality open-domain dataset consisting of contexts from Wikipedia and crowdsourced questions-answer pairs based on them.

- DuoRC (Saha et al., 2018) is a dataset based on movie plots based on text from Wikipedia and IMDB. In terms of domain closeness, DuoRC is the closest to SQuAD as it includes data from Wikipedia. However, the challenge introduced by DuoRC is its requirement for deeper content understanding since the question and answer are based on different versions of a plot (Wikipedia v/s IMDB) ensuring a lower lexical overlap between the two.

- CUAD (Hendrycks et al., 2021) represents the legal domain. Having the longest context length, CUAD is a collection of commercial contracts for legal document understanding.

- COVID-QA (Möller et al., 2020) was developed to enable question answering for COVID-related queries. It is a collection of answerable questions only based on research articles sourced from the CORD-19 dataset (Wang et al., 2020).

- TechQA (Castelli et al., 2020) is a dataset developed by IBM for question answering in the technical customer support domain. Its subject is not only much different than SQuAD, but the overall language, style and number of samples make this a very challenging dataset.

A.2 Models Tested

We test various neural architectures categorized as Non-Transformer and Transformer-based.

Non-Transformer: We examine two models based on either recurrent or convolution networks.

- BiDAF (Bi-Directional Attention Flow) (Seo et al., 2017) is a hierarchical recurrent model (LSTM) that captures fine-grained context and question semantics through word/character embeddings and left-to-right/right-to-left attention.
- QANet (Yu et al., 2018) uses unidirectional attention and enables parallelism/scale through several layers of convolution.

Transformers-Based (Vaswani et al., 2017): We look at popular models built using the encoder or decoder part of the transformer.

- **Encoders**: Architectures that fall in this category perform Masked Language Modelling (MLM). This means that during training, the model has access to the entire input sequence, at each step of processing, and is tasked with predicting randomly replaced (*masked*) tokens in the input. In this paper, we explore the following MLM style models,
  - BERT (Devlin et al., 2019) is one of the first language models built upon the Transformer-encoder, which showed strong performance across a range of tasks on the GLUE benchmark (Wang et al., 2018). Various *domain-specific* checkpoints of BERT are also tested in this paper to evaluate the impact of further ID pre-training. For the biomedical and technical domain, these include, BioBERT (Lee et al., 2020) (medical domain) and SciBERT (both domains) (Beltagy et al., 2019); for the legal domain, these include, FinBERT (Araci, 2019) and LegalBERT (Chalkidis et al., 2020).
  - SenseBERT (Levine et al., 2020) is pretrained to predict word senses derived from the English WordNet (Fellbaum, 1998). We use this model during our semantic similarity trials (c.f. 3.2.2) to determine the impact of this training objective on closed-domain sense discrimination.
  - RoBERTa (Liu et al., 2019b) optimizes BERT by removing its *Next Sentence Prediction* objective, adding dynamic masking and training over a larger corpus of data for more number of epochs.

- **Decoders** (c.f. 9): Models that perform causal language modelling (CLM) are termed as decoder-based or autoregressive language models. Such models predict future word(s) based on the preceding context. Here, the attention head allows the model to look only at prior tokens (unidirectional), unlike MLM models.
  - We use four of the latest CLMs for our experiments, i.e., Falcon (Almazrouei et al., 2023), Platypus (Lee et al., 2023), Gemma (Team et al., 2024) and Mistral (Jiang et al., 2023). Each model uses various advancements in LLM technology such as Grouped-Query Attention (Ainslie et al., 2023), Low-Rank Adaptation (Hu et al., 2022), etc. The most important factor for these models, however, is their training data which undergoes meticulous filtration to ensure high quality, such as the RefinedWeb corpus (Penedo et al., 2023) for Falcon. As above, we use various domain-specific checkpoints, as applicable. For the medical domain, this includes, MedAlpaca (Han et al., 2023) and BioMistral (Labrak et al., 2024); AdaptLLM (Cheng et al., 2024) for the legal domain and Phi-2 [4] for the technical domain.

B Categorical Answer Length Analysis

Additionally, we look at how the predicted answer length distributions align with the gold span distribution for two datasets, SQuAD and TechQA for BiDAF and RoBERTa. We chose SQuAD as it is the main dataset on which the models are trained, TechQA as it has the longest average gold span, RoBERTa as it is a better model overall than BERT and BiDAF as observed to be the better of the two non-transformer models. We plot histograms for this test, which are shown in Figure 5. The x-axis shows the length ranges of the gold spans for either dataset, and the y-axis shows how many answers

[4] https://huggingface.co/microsoft/phi-2

Dataset	Train				Validation				Test			
	Average Question Length	Average Context Length	Average Answer Length	Number of Records	Average Question Length	Average Context Length	Average Answer Length	Number of Records	Average Question Length	Average Context Length	Average Answer Length	Number of Records
SQuAD	59.57	754.36	20.15	87,599	60.01	778.98	18.73	10,570	-	-	-	-
DuoRC	40.14	3,801.48	14.32	60,721 (26,633)	39.97	3,837.77	14.38	12,961 (5,780)	39.88	3,763.95	14.27	12,559 (3603)
COVID-QA	58.54	32,082.28	93.42	2,019	-	-	-	-	-	-	-	-
CUAD	254.96	64,684.51	131.48	22,450	-	-	-	-	260	46,848.19	120.07	4,182
TechQA	270.88	51,452.02	269.85	600	286.77	92,629.53	156.79	310 (9)	-	-	-	-

Table 8: Dataset Statistics. Apart from number of records, all entries are text lengths in terms of average number of characters. '-' indicates unavailable dataset split. Numbers in parentheses for TechQA and DuoRC indicate the number of samples for which the answer span is not present **exactly** in the context for reasons such as inconsistent spaces, capitalization, word inflection (form) variation, etc.

fall within each range bucket. We use the same x-axis for all plots for a given dataset to determine the number of predictions falling within the corresponding gold range buckets.

Looking at the fine-grained answer length distribution in Figure 5 we get a better understanding of why there is poor generalization in MLM style models. For SQuAD, we see that both RoBERTa and BiDAF approximately mimic the gold span answer length distribution. RoBERTa of course performs better than BiDAF owing to its superior architecture and being trained on much more aligned data. However, for TechQA, RoBERTa does not show the same distribution. It is completely left-skewed, with all the predictions falling under 115 characters. On the other hand, BiDAF, despite also being left-skewed, shows a more spread-out distribution. Two of its answers fall in the 918-1033 character range, the same as that in the gold distribution. While RoBERTa cannot break the 30-character length mark for TechQA, BiDAF manages an average of 4k characters [5] (Table 1).

C Benefits of encoder models (BERT)

Despite falling out of favour [6] instead of newer autoregressive models, BERT and its variants have the following advantages,

1. *Shorter training times*, e.x. Pre-training GPT-1 (Radford et al., 2018) took 1 month across 8 GPUs [7] v/s $\text{BERT}_{\text{BASE}}$ took 4 days on 16 TPU's (Devlin et al., 2019).
2. *Smaller model size* e.g. $\text{BERT}_{\text{BASE}}$ has 110M parameters (Devlin et al., 2019) v/s GPT-1 has 117M parameters (Radford et al., 2018).
3. *Being more suitable for information extraction tasks* such as Named Entity Recognition (Deußer et al., 2023) (although not always as shown in (Sarrouti et al., 2022) for biomedical relation extraction, where encoder-decoder models can occasionally top encoder-only models) and span detection (EQA) (Xu et al., 2021; Mallick et al., 2023).
4. *We still find instances of bidirectional language modelling being used in innovative ways*, such as (Li et al., 2023) who propose an encoder-only model to link text modality with geospatial content.

C.1 BERT configurations tested

Figures 6 and 7 provide an overview of all BERT variations that were tested. Figure 8 provides the range of scores for all configurations of uncased BERT models with word-piece masking.

D Testing ChatGPT

We were curious to see how well ChatGPT with either GPT-4 or GPT-3.5 was able to perform zero-shot EQA. We select a random sample from a biomedical dataset, BioASQ (Tsatsaronis et al., 2015) as it has shorter contexts, to test ChatGPT. In Figure 9, for the given question, against the true answer of *zfPanx1 was identified on the surface of horizontal cell dendrites invaginating deeply into the cone pedicle near the glutamate release sites of the cones, providing in vivo evidence for hemichannel formation at that location.*, GPT-4 almost identifies the correct span while introducing minimal new text (period instead of comma). GPT-3.5 introduces/patches together even more text with the answer, *The protein Pannexin1 (zfPanx1) is located on the surface of horizontal cell dendrites*

[5]Longer spans cannot be shown in the plot since they exceed the gold limit (only 278 out of 310 samples are shown in the plot).

[6]https://www.deepset.ai/blog/the-definitive-guide-to-bertmodels

[7]https://openai.com/research/language-unsupervised

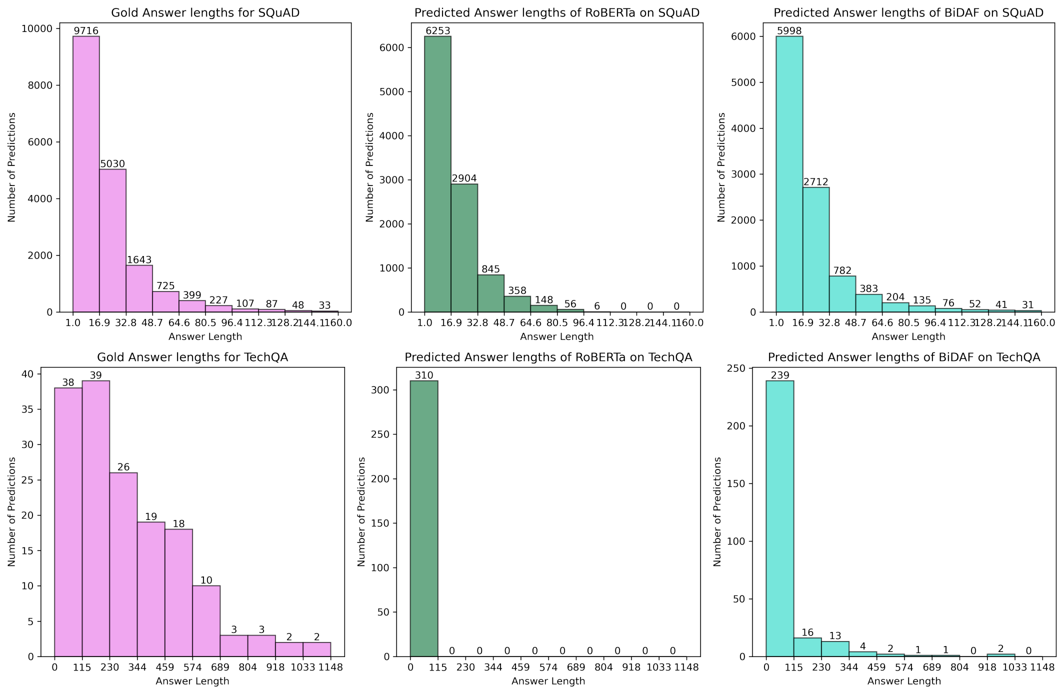

Figure 5: Answer length distribution for BiDAF and RoBERTa on SQuAD (top) and TechQA (bottom).

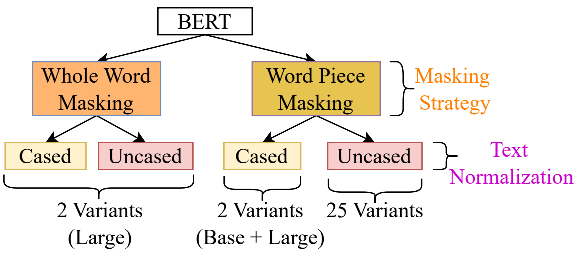

Figure 6: Classification of all BERT models tested for zero-shot EQA based on pre-processing choices. Overall, this gives us 25 (Fig. 7)+2+2=29 variations to test.

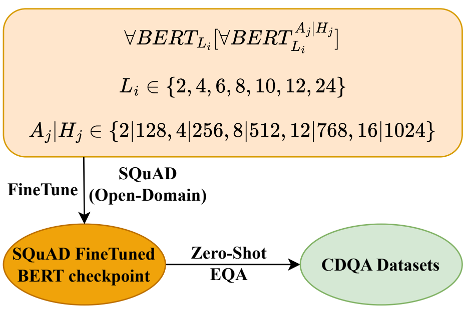

Figure 7: Testing various configurations of BERT to see how they impact zero-shot EQA performance. Note, $A|H = 16|1024$ is only available for $L = 24$.

invaginating deeply into the cone pedicle near the glutamate release sites of the cones in the zebrafish retina. Changing the prompts did not seem to improve performance. Although this is a single example, it goes to show that even the most capable LLMs struggle with span extraction due to their generative nature and tendency to hallucinate.

E Instruction Templates and autoregressive LLM testing setup

We use the following prompt template across all models, as described in Han et al. (2023).

```
Context:  {context text}
Question:  {question text}
Answer:  <generated text>
```

The models were prompted in this manner for the following reasons,

- Initially, we attempted to format the samples using the instruction tags [8] for each corresponding model. However, we found that for certain models such as `Gemma` (Team et al., 2024), even when using the appropriate tags/prompt template, they produce answers in inconsistent formats[9].

[8] `https://huggingface.co/docs/transformers/main/en/chat_templating`
[9] Figure 10

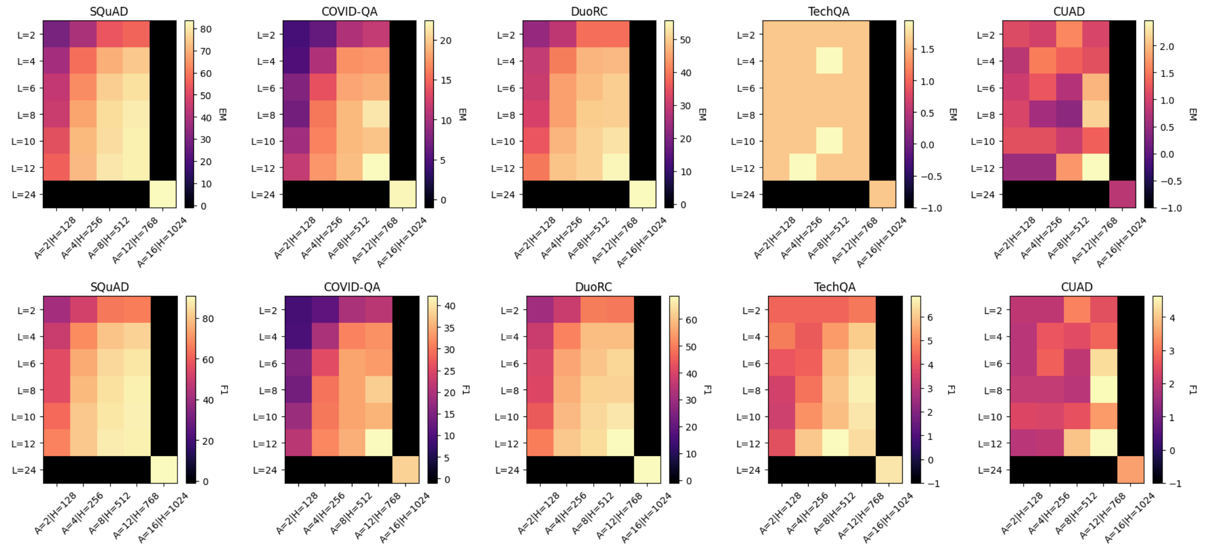

Figure 8: Impact of scaling number of layers (L), attention heads (A) and layer dimension size (H) on EQA generalization for uncased BERT with word-piece masking. EM scores (top) | F1 scores (bottom).

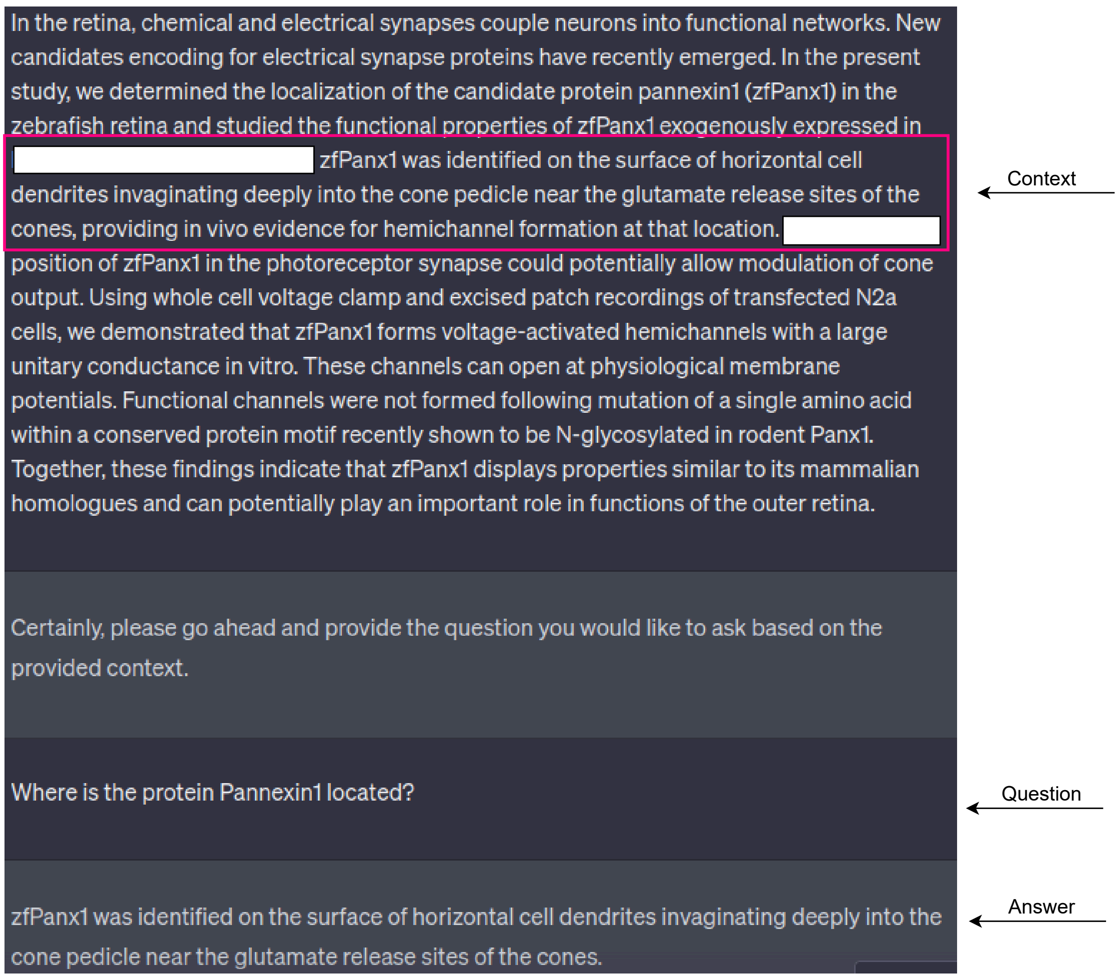

Figure 9: Testing ChatGPT (with GPT-4) on a "simple" Biomedical EQA question.

- It allows us to hold each tested model to a common standard of evaluation, and also provides us with the ease to extract the generated answer automatically by clipping the prompt at a common point (`Answer:`).

In our experiments, the autoregressive models were given the following advantages,

1. Retaining answerable questions only - By the very nature of causal training, autoregressive models generate text based on the input prompt. Thus, if the provided context does not contain the answer to the question, they will be penalized for producing an incorrect answer. Thus, we only consider answerable samples for a fair evaluation.

2. Reduced context length - As it has been established that LLMs struggle with processing long contexts (Liu et al., 2024; Li et al., 2024), we truncate the samples from the long context datasets COVID-QA, TechQA and CUAD such that each context chunk is guaranteed to contain the answer while being smaller than the models' maximum input window. SQuAD and DuoRC do not need to be truncated as they have shorter contexts.

F Limitations of autoregressive LLMs for EQA

The true power of zero-shot learning came into the picture with the release of GPT-3 (Brown et al., 2020) a massive 175B autoregressive (or causal) model (trained to predict the next word conditioned only on preceding words) capable of remarkable zero-shot and few-shot learning, i.e., supplied with zero/a few test samples it can perform the target task directly without need of further training. The success of GPT-3 propelled NLP into the LLM era (Zhao et al., 2023) where models are essentially used off-the-shelf for a range of real-world tasks simply by explaining the problem in natural language, a process called *prompting* (Liu et al., 2023). Harnessing the power of this new feature, we test the generalization capabilities of several state-of-the-art (SOTA) causal models by benchmarking them on our datasets in zero-shot fashion, i.e. directly asking them to answer the question based on the context without further training.

In addition to gauging their raw performance *we test two hypotheses* to explain their behaviour,

Model	Dataset	EM	F1	Predictions in Full Context
MedAlpaca	COVID-QA	**6.79**	**39.59**	384
Falcon		4.06	32.2	117
Platypus		5.25	34.25	380
Gemma		2.77	18.04	**687**
BioMistral		3.41	32.94	203
Falcon	SQuAD	13.29	28.4	931
Platypus		**23.6**	**40.14**	3098
Gemma		10.78	17.45	**7364**
Mistral		1.2	22.25	136
AdaptLLM	CUAD	0	7.38	13
Falcon		0	8.63	3
Platypus		0	5.64	194
Gemma		0	1.45	**798**
Mistral		0	**11.6**	1
Falcon	TechQA	0	**9.26**	0
Platypus		0	6.86	0
Gemma		**0.66**	5.11	**23**
Mistral		0	8.76	0
phi-2		0	7.6	2
Falcon	DuoRC	**11.29**	22.92	682
Platypus		10.1	**30.01**	1707
Gemma		6	10.61	**8339**
Mistral		4.4	29.97	517

Table 9: Zero-Shot Decoder Evaluation on all five datasets. Contexts of COVID-QA, TechQA and DuoRC are truncated such that each context chunk always contains the answer. Blue indicates ID models while **bold** is the best performing model.

the first of which is linked to their core **operating objective**. Causal models treat EQA as a standard *language modelling problem* (e.q. (1) (Radford et al., 2018)) where the model (with parameters θ) predicts the future word (u_i) given the preceding context ($u_{<i}$) by maximizing the log-probability of the generated sequence.

$$\sum_i \log P(u_i|u_{<i};\theta) \tag{1}$$

Contrary to this, bidirectional models, are trained by adding a linear layer on top of the base model and predicting the start and end tokens of the answer span by minimizing the loss (Jurafsky and Martin, 2019) as shown in e.q. (2) where P_{start_i} is the probability of the ith context token being the start token and similar for the end token [10].

$$-\log P_{start_i} - \log P_{end_i} \tag{2}$$

Contrasting the language modelling with the start/end token prediction objective, we consider two hypotheses to potentially explain shortcomings in the former for EQA. First, with bidirectional models, we can constrict them to use only context tokens for answer prediction since the linear layer is trained to process only those tokens which means the answer span will *always* come from the

[10] For details on how the probabilities are calculated, we refer the reader to chapter 14 of (Jurafsky and Martin, 2019)

```
<bos><start_of_turn>user
Select the most appropriate context span which answers the given question without any explanation.
Context: Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ bee-YON-say) (born September 4, 1981) is an American singer, songwriter, record producer and actress.
Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer
of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their
hiatus saw the release of Beyoncé's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and
featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".

Question: When did Beyonce start becoming popular?
<end_of_turn>
<start_of_turn>model
The answer is: in the late 1990s as lead singer of R&B girl-group Destiny's Child.
<end_of_turn>
<eos>
------------------------------------------------------------------------------------------------------------------------------------------------------------
<bos><start_of_turn>user
Select the most appropriate context span which answers the given question without any explanation.
Context: Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ bee-YON-say) (born September 4, 1981) is an American singer, songwriter, record producer and actress.
Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer
of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their
hiatus saw the release of Beyoncé's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and
featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".

Question: What areas did Beyonce compete in when she was growing up?
<end_of_turn>
<start_of_turn>model
The most appropriate context span for this question is:

**Context:** Beyoncé Giselle Knowles-Carter (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in
Houston, Texas, she **performed in various singing and dancing competitions** as a child, and rose to fame in the late 1990s as lead singer of R&B girl-
group Destiny's Child.

This context span answers the question of "what areas did Beyonce compete in when she was growing up" by mentioning "singing and dancing competitions."
<end_of_turn>
<eos>
------------------------------------------------------------------------------------------------------------------------------------------------------------
<bos><start_of_turn>user
Select the most appropriate context span which answers the given question without any explanation.
Context: Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ bee-YON-say) (born September 4, 1981) is an American singer, songwriter, record producer and actress.
Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer
of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their
hiatus saw the release of Beyoncé's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and
featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".

Question: When did Beyonce leave Destiny's Child and become a solo singer?
<end_of_turn>
<start_of_turn>model
The text states that Beyoncé left Destiny's Child and became a solo singer in the late 1990s. Therefore, the most appropriate context span is "the late
1990s."
<end_of_turn>
<eos>
```

Figure 10: Testing chat template for Gemma. As can be seen, for the recommended template, the model is inconsistent in producing its answer.

context (Jurafsky and Martin, 2019). However, with decoder-based models, there is no such constraint, which means that they are free to predict the most likely word(s) from their vocabulary conditioned on the starting text ($Q + C$). Although there have been attempts to remedy this via libraries such as `lm-format-enforcer`[11] and through additional training on instruction-data (Zhou et al., 2023), there is no such requirement *baked* into the makeup of these models. This leads us to test *how many times the generated answer is present in the context*. If the model generates new text rather than using only context tokens, the EM (Exact Match) will decrease considerably, which in turn will lead to its overall poor performance.

The second relates to how the models **process the input**. Bidirectional models are trained to distinguish between question and context (or any two separate sequences) by using a special $[SEP]$ token. $Q + C$ is then processed as $[CLS][Q^n_{i=1}][SEP][C^m_{j=1}]$ where $[CLS]$ is a special token used for classification tasks and Q_i/C_j are question/context tokens respectively. However, causal models view the entire input as a single sequence without any special separator in between. This leads us to question whether they can *identify the question and context correctly*, which should be considered a basic ability. If they are not able to do so, it could be another explanation for their poor performance. We test this idea by *simply asking the models to repeat the context and questions verbatim* by prompting them as,

```
Write the context and question
exactly.
Context: {context text}
Question: {question text}
```

We also test whether the models are sensitive to the location of either component by reversing the order of the context and question. It should be mentioned here that we provide an instruction for this task and use each model's prompt template as they were seen to perform better than in their absence as opposed to the zero-shot setting described before. We explain this behaviour by observing that copying text is qualitatively easier for the model than extraction based on a condition (question). As such,

[11] https://github.com/noamgat/lm-format-enforcer

		Normal Order			Reverse Order		
Model	Dataset	CIC	CIQ	CIB	CIC	CIQ	CIB
Falcon	SQuAD	0.26	0.31	0.2	0.12	0.18	0.09
Platypus		0.03	0.01	0.01	0.17	0.13	0.11
Gemma		0.22	0.82	0.2	0.01	0.75	0.01
Mistral		0	0	0	0	0.14	0
Avg.		**0.13**	**0.29**	**0.1**	**0.08**	**0.3**	**0.05**
Falcon	DuoRC	0.07	0.07	0.04	0.08	0.65	0.05
Platypus		0.02	0	0	0	0.08	0
Gemma		0	0.59	0	0	0.4	0
Mistral		0	0	0	0	0.1	0
Avg.		0.02	0.17	0.01	0.02	0.31	0.01
MedAlpaca	COVID-QA	0	0.14	0	0	0.15	0
Falcon		0.05	0.28	0.02	0	0.26	0
Platypus		0	0.02	0	0	0.01	0
Gemma		0	0.66	0	0	0.48	0
BioMistral		0	0	0	0	0.04	0
Avg.		0.01	0.22	0	0	0.19	0

Table 10: Testing instruction following capabilities of each decoder model. Normal Order = Context followed by Question | Reverse Order = Question followed by Context. CIC/CIQ/CIB = Correctly Identified Context/Question/Both. Each score represents the fraction out of 100 randomly selected samples for which the model positively identifies/repeats each component.

they can follow the instructions much better here. Overall, this will test not only their instruction-following abilities but also reveal a potential flaw in their design, i.e., the inability to identify what portion of the input corresponds to which segment.

For these experiments, we use only SQuAD, DuoRC and the truncated samples from COVID-QA. This is because repeating the examples from CUAD and TechQA would exceed the model's maximum context window, even if they are truncated. We randomly sample 100 examples from the selected datasets to run our trials and report the fraction of samples that were correctly identified.

We first analyze their cross-domain performance in Table 9 to investigate the first hypothesis i.e. whether their answers are an exact match with the associated context, since the requirement for EQA is that the answer span must match the context verbatim. Despite having to process less context, poor EM and prediction hit rate (number of times the answer matches the context exactly) indicate that this is a major bottleneck for these models. As discussed previously, this is unsurprising since generation is unconstrained, i.e., conditioned on the seed text, the models are free to predict the next word based on the most probable token in its vocabulary. Overall, we see how serious this issue is since the models reporting the highest hit rates for the more challenging COVID-QA, CUAD and TechQA datasets, could barely break the 40% mark (percentage of predictions in full context/total number of samples).

Although it is natural to expect that at least the ID models would display the best performance, we find that for COVID-QA, CUAD and TechQA, only MedAlpaca (for COVID-QA) performs the best. For TechQA, this makes sense since phi-2 is a much smaller model (2.7B params) than the others. In the case of CUAD, although AdaptLLM was trained on legal knowledge, it uses LLaMA-1 (Touvron et al., 2023a) as the backbone whereas the best-performing model, Mistral, is a much stronger model capable of outperforming the more powerful LLaMA-2 (Touvron et al., 2023b).

Results from the context and question identification trials are presented in Table 10. First, model-wise, `Gemma` displays the most impressive instruction-following abilities as, on average, it reports the most identified samples in either configuration and across domains. This also aligns with the fact that it reported the most number of exact answer predictions for each dataset (c.f. Table 9). Second, from the results, it is evident that the location of each component plays an important part, i.e., Normal order or, `Context` followed by the `Question`, appears to be the preferred way of formatting samples for EQA. Finally, as expected, each model recognizes the most number of samples for SQuAD displaying again a weakness in generalizing to OOD datasets. Surprisingly, `MedAlpaca` and `BioMistral` while being biomedical models are outperformed by `Gemma`, perhaps owing to its superior instruction tuning. Overall, the takeaways are,

1. Although nowhere near good enough for EM, **LLMs display better performance than Bi-directional models for extremely challenging OOD datasets such as CUAD and TechQA in terms of F1**. Thus, if the dataset can be constrained to only answerable questions, LLMs in zero-shot *could* potentially be a good option.

2. **LLMs are sensitive to the location of the context and question in the prompt**. Thus, care should be taken when formatting the samples as it can impact cross-domain performance.

G TEXT/TASK Embedding Background

TEXT Embedding Each sample is processed by the frozen base model, i.e., without any additional training, and the average of the *pooled* representation from each input sequence stands as the datasets' TEXT embedding. This vector is used to compare datasets across different domains. Here, the specific task is not important, i.e., as long as two datasets belong to the same domain, their TEXT embeddings will be similar.

TASK Embedding We provide an intuitive understanding of TASK embeddings and direct interested readers to Achille et al. (2019) and Vu et al. (2020) for a deeper understanding of related concepts. First, TASK embeddings view the entire model as a real-valued vector with the total number of dimensions equal to the number of model parameters. During training, each dimension of this vector reflects how much a parameter changes or, is affected during backpropagation. In other words, each dimension tracks the gradient of the loss function w.r.t. each parameter. However, for extremely large models, the TASK embedding can become unmanageably high dimensional. Thus, to compress the feature space, they employ the Fisher information matrix (Ly et al., 2017) to retain the top-N parameters which have the most impact on model performance and by extension the task itself. For TASK embeddings, the dataset semantics are secondary to the actual task itself, i.e., as long as the two tasks are similar, their corresponding embeddings will be similar even if they deal with different domains.

H Note on Force-Directed Algorithm

FDA is used to build graphs. However, since we only have a single focal point (SQuAD) and all other datasets are evaluated w.r.t it, it does not make sense to have a graph with just four outgoing edges from a single node. Thus, we use a bar chart for clarity. Also, it should be noted that we kept the values as is without normalization too $[0, 1]$, for better visualization.

I Dataset Perplexity Background

When training a language model, PPL is used to gauge how well it understands unseen corpora. Perplexity can be understood from various points. Intuitively, it indicates how *perplexed* or *confused* the model is by the test data. In other words, given a sequence of tokens, PPL measures how likely (probable) the model believes it is grammatically and semantically. Lower PPL on a corpus indicates a well-trained model.

Typically, PPL has been used to describe the performance of causal language models (predicting future tokens given the preceding context). However, the definition can just as easily be extended to MLM (BERT) style models[12]. Formally, perplexity of a sequence of tokens $X = (x_0 \dots x_t)$ is given in eq. (3). Accordingly, it computes the average log probability over the entire sequence of tokens. This value is negated to shift the score to a positive scale and exponentiated for better readability of very small values.

$$PPL(X) = \exp\left\{-\frac{1}{t}\sum_{i=0}^{t}\log p_\theta(x_i|x_{<i})\right\} \quad (3)$$

J Software

All of our code and datasets are available at https://github.com/saptarshi059/generalization-hypothesis.

K Why aren't Causal LMs evaluated in section 3.2.1?

This point was raised by a reviewer during the review process. We provide our clarification as follows,

This is a fair question. We can certainly evaluate them. However, the focus was on models that were already trained to predict a certain answer length. Thus, we investigate whether they overfit it or are capable of generalizing to longer spans. As the generation length of autoregressive LLMs can be controlled, we felt their inclusion here was against the point. That said, we do test their other aspects in later sections (App. E/F).

[12]The HuggingFace library (Wolf et al., 2020) computes the PPL of both model types in the same way: https://huggingface.co/docs/transformers/en/perplexity, https://huggingface.co/learn/nlp-course/en/chapter7/3#fine-tuning-distilbert-with-accelerate