`TOP-Training`: Target-Oriented Pretraining for Medical Extractive Question Answering

Anonymous ACL submission

Abstract

This work studies extractive question answering in the medical domain (Medical EQA). This problem casts two main challenges: i) domain specificity; most AI models lack the necessary knowledge; ii) extraction-based answering style, which restricts most autoregressive LLMs due to the potential hallucination concern. To handle those challenges, this work proposes `TOP-Training`, a target-oriented pretraining paradigm that stands out among all domain adaptation techniques with two desirable features: i) `TOP-Training` moves one step further than popular domain-oriented fine-tuning since it not only moves closer to the target domain, but also familiarizes itself with the target dateset, ii) it does not assume the existence of large set of unlabeled instances towards the target domain. Specifically, for a target medical EQA dataset, we extract its entities and leverage large language models (LLMs) to generate synthetic texts containing those entities; pretraining on this synthetic text data is shown better performance on the target medical EQA benchmarks. Overall, our contributions are threefold: i) `TOP-Training`, a new pretraining technique to effectively adapt LLMs to better solve a target problem; ii) `TOP-Training` has a wide application scope because it doesn't require the target problem to have a large set of unlabeled data; iii) Our experiments highlight the limitations of autoregressive LLMs, emphasizing `TOP-Training` as a means to unlock the true potential of bidirectional LLMs.

1 Introduction

EQA involves identifying a span of tokens in a passage to answer a given question. While language models such as BERT (Devlin et al., 2019), T5 (Raffel et al., 2020), and GPT-3 (Brown et al., 2020) have made remarkable leaps in this task, they suffer drastically when applied to domain-specific datasets, especially medical (Moradi et al., 2021).

We believe that the performance discrepancy is linked to the definition of a *domain*, i.e., the loose NLP equivalency of $domain = genre$ or *thematic content*, which is quite restrictive, as argued by Plank (2016). Theoretically, a model trained on a specific theme should excel in tasks related to that subject matter. However, not all domain-specific models are equal as illustrated by the difference in performance of BioBERT (Lee et al., 2020) and PubMedBERT (Gu et al., 2021), even though both were trained on PubMed data (Tables 1,2). We suggest redefining $domain = [genre + dataset]$, and emphasizing the importance of tailoring the training data to not just the theme, but also on the target task dataset. This approach acknowledges that a *one-domain-model-to-rule-them-all* is not universally applicable, and that learning should also focus on concepts relevant to specific tasks.

Recent progress in generative FMs (Ye et al., 2023; Achiam et al., 2023) gives us reasons to believe that a solution to the closed-domain problem has been reached, particularly with ChatGPT, which has demonstrated strong performance, e.g., on the USMLE (Kung et al., 2023) exam. However, our expectations might be myopic as we (authors) demonstrate that decoder-only FMs struggle (Tables 1, 2) with specialized language, such as COVID-QA (Möller et al., 2020) and RadQA (Soni et al., 2022). Moreover, their autoregressive architecture is not well-suited for EQA as they are designed to *synthesize* new text rather than *extract* spans from given text (§ 5.1, 2.1). For sequences exceeding the model's maximum limit, they need to be divided into overlapping segments (pre-processing). Though this issue applies to both bi-directional and generative models, the former is more suitable due to its inherent capabilities (2.1). Furthermore, these models handle EQA as autoregressive language modelling rather than start/end token prediction, which can lead to either mismatch in generated/true answer tokens (§2.1) or even *hal-*

lucinated text (Ji et al., 2023).

To overcome these limitations, we propose distilling the knowledge from causal FMs into smaller MLM-style models, which are better suited for EQA. Our approach uses a causal FM to *generate synthetic corpora* tailored to a specific application and fine-tuning open-domain MLM models on this corpus. We adapt open-domain models instead of further training domain-centric ones (ex. Bio/SciBERT) (we did train the best performing biomedical model on our dataset, but saw no noticeable gains) to show the resource benefits of our pipeline. That is, without relying on gigabytes of domain-specific pre-training data, we can achieve respectable gains by honing our models on modest-sized corpora. Our results demonstrate the efficacy and the running time improvements with respect to existing domain-specific models.

We generate corpora instead of using existing text because: (a) it provides the *flexibility* to create content in a certain style, (b) some corpora can be *unavailable* for privacy reasons, ex. clinical diagnostic reports, or "internal" versions of corpora such as that used by Gururangan et al. (2020), and (c) our tests can be used to determine if the content produced by such FMs is *factually grounded* and able to teach student models specific writing styles.

In summary, our contributions are,

- Proposing a pipeline for generating customized pre-training data for closed (in our case medical) domains.
- Demonstrating the effectiveness of synthetic data for achieving sizable gains with reduced memory footprint.
- Showing the benefits of creative prompting and dataset awareness.
- Benchmarking numerous pre-trained biomedical FMs on COVID-QA and RadQA and contrasting our models with them by showing their superiority.

2 Related Work

In this section, we discuss prior work on decoder-based FMs for QA, highlight their limitations and describe recent knowledge distillation pipelines.

2.1 Autoregressive modeling for QA

On release, GPT-4 (Achiam et al., 2023) became *the* benchmark FM on almost all canonical NLP tasks. Remarkably, Nori et al. (2023) proved that just by careful prompting, GPT-4 can achieve SOTA on various medical QA datasets. Initiatives such as Meditron-70B (Chen et al., 2023) became the next best thing to parallel GPT-4's medical expertise. On closer inspection however, we see that the datasets on which these two models are applied are all multiple-choice style. Thus, their capability of handling EQA datasets, especially in the medical domain remains unexplored (cf. App. C).

Attempting to autoregressive FMs, Xu et al. (2021), through indirect supervision, train BART (Lewis et al., 2020) for span extraction by considering cross-attention weights as start/end token probabilities to align the generated and true spans. Noting that generating tokens is perhaps not the best strategy, Mallick et al. (2023) reframe the problem as generating *numeric indices* indicating either token or sentence-level spans. Innovative as these solutions are, we see two issues, i) marginal performance gains, and ii) only Mallick et al. (2023) use medical EQA dataset (Xu et al. (2021) focus only on open-domain) that too without much success - both points as compared to their encoder-based SOTA indicating that they are ineffective.

Luo et al. (2022) show that on average *encoders outperform decoders* in short-span detection with the added advantage of being better at *out-of-domain generalization* whereas decoder models can be comparably used in *long-context* EQA. Additionally, they show that the encoder alone from Seq2Seq models such as T5 follow a similar trend. While they do make the case for autoregressive FMs, later studies (Liu et al., 2023) demonstrate how newer instruction-tuned FMs such as MPT-30B-Instruct (Team et al., 2023) are *sensitive to the location of the gold span* Although they do not focus on encoders-alone, they do show how Seq2Seq models are relatively robust to such positional changes due to their encoders.

2.2 Knowledge Distillation

In the area of knowledge distillation, West et al. (2022) demonstrate how GPT3 can be utilized to create high-quality knowledge graphs via prompting. He et al. (2022) show how a GPT model can be used as a "teacher" to distil knowledge into a "student". Similarly, Peris et al. (2022) use unlabelled task-relevant data and trained multilingual students with varying proportions of general/task-specific data and report the most gains using "only the downstream task's unlabelled data".

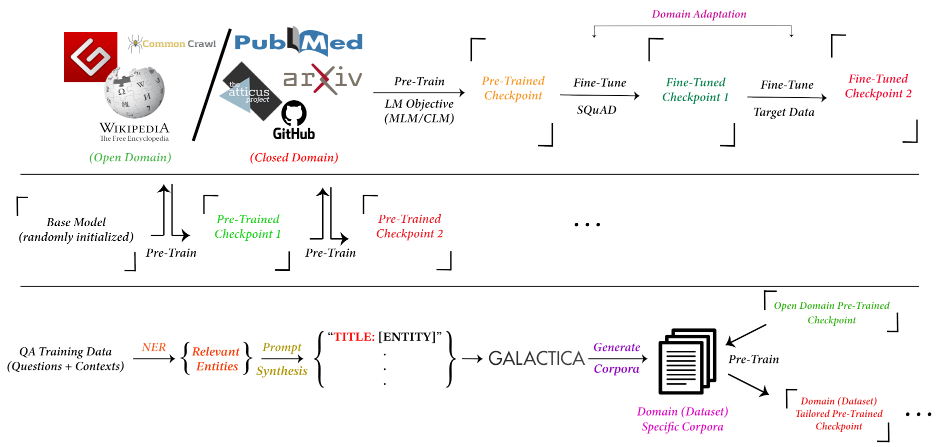


Figure 1: Pre-Training Pathways: From scratch (top); Extended (middle); **Targeted** (bottom; ours) | Note: We only show fine-tuning on EQA as it is the task of interest | The prompt handle is written in CAPITAL for emphasis.

Gururangan et al. (2020) introduced DAPT (Domain-Adaptive Pretraining) and TAPT (Task-Adaptive Pretraining), which share similarities with our work. DAPT involves extended pretraining on domain-specific corpora without labels, while TAPT focuses on pretraining on the unlabelled training set of the downstream task. Although they demonstrate the effectiveness of TAPT compared to DAPT, closed-domain datasets like COVID-QA typically lack a separate unlabelled training set and may not even have train/dev/test splits. Further, DAPT considers knowledge beyond what is necessary to the task data, whereas our approach confines training to only required concepts.

While the above techniques have achieved much success, they typically rely on high quantities of unlabelled corpora to yield useful results, thus raising the question: *What happens when we do not have enough "relevant" domain data, either in style or volume*? Inspired by the works of Gunasekar et al. (2023) and Zhou et al. (2023), who hearken the use of smaller yet *better quality* corpora, we introduce the notion of `TOP-Training`, which focuses on a specific subset of the domain, tailor-made for the ultimate downstream dataset.

3 Formulation of medical EQA

Each labeled data point consists of three elements:

- CONTEXT ($\mathcal{C}$): A piece of text in the medical domain that introduces the necessary information about a topic;
- QUESTION ($\mathcal{Q}$): A question sentence acquiring the information from CONTEXT. *It can be answerable or unanswerable*.
- ANSWER ($\mathcal{A}$): A consecutive span in CONTEXT that acts as the answer of QUESTION.

Typical datasets for medical EQA are split into `train`, `test` and optional `dev` sets.

Two main challenges for medical EQA problems: i) domain specificity; most AI models lack the necessary knowledge; ii) extraction-based answering style, which restricts most autoregressive LLMs due to the potential hallucination concern. The two challenges motivate us to propose `TOP-Training` accompanied with bi-directional LLMs for this particular problem.

4 `TOP-Training`

For the target closed-domain medical QA problem, our `TOP-Training`, depicted in Figure 1, first extract entities from the target dataset, then leverage an existing LLM to generate entity-related CONTEXT (Section 4.1), which later acts as the data for further tuning a bi-directional LLM for the target problem (Section 4.2).

4.1 LLM generates target-oriented CONTEXT

Entity collection from the target problem. In this work, we use entities in the target dataset as the connection between it with the newly generated synthetic data. To extract entities from the target EQA dataset,

First, we combine all the QUESTIONS and CONTEXTS from the `train` split of the medical EQA dataset. Next, we extract entities through Named Entity Recognition (NER) using spaCy[1]. This step identifies roughly 47k and 11k entities in COVID-QA and RadQA, respectively.

---

[1] With "`en_core_sci_sm`".

Synthetic CONTEXT generation. Next, we create prompts for the identified entities to generate CONTEXTS mimicking the target datasets. This required studying the characteristics of the target datasets such as the text genre (full research articles in COVID-QA & radiology reports in RadQA), lengths, and relevant keywords.

Galactica, a decoder-based LLM pre-trained on a collection of text encompassing research articles, knowledge bases, code and even LaTeX markup, is used to generate the synthetic data[2] with the following prompts.

- **Prompt for COVID-QA.** Since COVID-QA comprises research papers, based on this characteristic, we develop the following prompt for Galactica to generate pseudo research articles based on retrieved entities:

Title: [Entity]

where `Title` is the prompt handle/keyword and `entity` is the entity identified in the above stage.

- **Prompt for RadQA.** RadQA's CONTEXTS are redacted radiology reports without any consistent format (Hartung et al., 2020). The *Findings* and *Impressions* sections are the most vital in a patient's report (akin to the experiment and results section in a research paper). Inspired, we propose the following prompt:

Patient has Entity. FINDINGS AND IMPRESSION

It is worth noting that Galactica had not been trained on radiology reports. Through this prompt, we synthesize *pseudo-reports* bypassing any privacy concerns. To maintain size parity between the two target datasets, five CONTEXTS are generated for each entity identified in RadQA, yielding around 55k (11k*5) total CONTEXTS, and one CONTEXT for each identified entity in COVID-QA, resulting a set of size 47K.

4.2 System pre-training on synthetic data

After generating CONTEXTS for each target data, we perform `TOP-Training`i.e., extended pre-training of BERT/RoBERTa on our generated corpus. In addition, `TOP-Training`is followed by two rounds of fine-tuning:

- The system will be first fine-tuned on the SQUAD dataset to learn what EQA is in a generic domain;
- Subsequently, it is further fine-tuned on the $train$ of each target data (either COVID-QA or RadQA), to solve the $test$.

5 Experiments

We focus on two datasets: COVID-QA, comprising 2,019 answerable QA pairs (no train/dev/test splits) sourced from CORD-19 (Wang et al., 2020), and RadQA, consisting of 6,148 QA pairs from radiology reports, with a train/dev/test split of 4,878/656/614. We experiment in two areas: baselines and `TOP-Training`, and provide mean and standard deviation scores over three random seeds for each.

5.1 Baselines & influencing factors

We consider 13 encoder models to each dataset taking into account which model made most sense to apply to a dataset. On COVID-QA, we applied models from checkpoints fine-tuned on SQuAD v1 while RadQA, containing unanswerable questions, was tackled with those fine-tuned on SQuAD v2 (Rajpurkar et al., 2018). For consistency, we utilized the *cased*, *base* version of each architecture when available. We use five-fold cross-validation to fine tune the models and report the results in Table 1. Results of models applied to the prescribed splits (RadQA) are presented in Table 2. The metrics used are exact match (EM), binary measure of whether the prediction and gold-standard spans are identical and F1, the harmonic mean of the number of shared words in the two spans with respect to the number of words in the prediction (precision) and with respect to the number of words in the gold-standard span (recall).

We selected Galactica-1.3B for consistency with our corpus generation experiments, MedLLaMA(13 B) as a strong open-source medical checkpoint, and MedAlpaca(13 B) as a medical QA-specific LLaMA checkpoint. We measure the ability of the three decoder models to generate answers without fine-tuning, considering that decoders do not extract spans, but generate answers for comparison to gold-standard spans.

Following Yue et al. (2021), each sample is formatted as `Question:<question_text> Context:<part_of_context> Answer:`. Due to the large size of COVID-QA contexts, they were segmented as they exceeded the maximum

---

[2] Other generative models such as BLOOM (Scao et al., 2022) and PubMedGPT performed worse in our experiments.

sequence length of each model (2,048 tokens). We report overall EM/F1 on each dataset and average best EM/F1 (parenthesis in Table 1) from each Q+C+A chunk for COVID-QA (N/A for RadQA since the context size was much smaller than the models' maximum input length).

Corpus Size. We investigated the impact of synthetic dataset size on downstream performance in COVID-QA. We examined the effects of generating one and 10 contexts per entity.

Context Length. The average context length for COVID-QA is 6k tokens, and Galactica has a maximum context size of 2k, resulting in a misalignment between the synthetic corpus and the target dataset. We cannot increase the context size of Galactica. Training it from scratch with architectural changes is infeasible for us. Thus, we explore the impact of sequence length in the synthetic corpus by limiting the records to only 1k tokens. We cannot determine if *longer* sequences are *beneficial*; we can evaluate if *shorter* ones are *detrimental*.

Prompting Style. We explore the use of two different prompts when encouraging Galactica to generate *pseudo* radiology reports - "`Patient has [entity]. FINDINGS AND IMPRESSION`" (*fancy prompt*), and simply "`[entity]`" (*normal prompt*).

Human-Generated Contexts. We establish a *Wikipedia* baseline alongside our domain-specific models to assess the influence of content and text structure during domain adaptation. Additionally, Micallef et al. (2022) have shown that mBERTu Wiki, pre-trained on Maltese Wikipedia data, surpassed the performance of mBERT, thereby proving to be a competitive baseline. For each entity, we query Wikipedia and retrieve the complete page associated with the top search result. The number of entities available for this baseline is much smaller than that in our approach since most of the entities do not exist in Wikipedia due to either being extremely esoteric, e.g., `pulmonary parenchymal infiltrate` or improperly formed, e.g., `Bao &`.

6 Results & Discussion

We discuss the results of existing baselines (Tables 1 & 2) and `TOP-Training` (Tables 3 and 4). The last 3 rows of Tables 1 and 2 provide zero-shot performance of our chosen decoder models on RadQA and COVID-QA respectively. We do not perform multiple trials here as the extremely poor performance would not benefit from additional runs. Granted they were not fine-tuned on our datasets, their size, pre-training data coverage and reported zero-shot performance on related datasets, should have allowed them to at least perform on par or better than open-domain BERT/RoBERTa. Overall, we see that MedAlpaca seems to be the "best" among the three for RadQA and only marginally poorer in terms of F1 for COVID-QA probably since MedAlpaca is an instruction-tuned version of MedLLaMA. On COVID-QA, none of the models generated text in line with the gold standard (~0 EM) and only showed positive F1.

6.1 COVID-QA

We now discuss the results benchmarking trials on COVID-QA contrasting the methods.

6.1.1 Baselines

Our experiments demonstrate that a one-size-fits-all approach does not work always for domain adaptation. BioBERT and PubMedBERT were trained on similar corpora and yet score in the same range, indicating no clear winner. PubMedBERT that is trained from scratch using a custom vocabulary covering a range of medical jargon performs better.

SciBERT (+CORD-19) checkpoint, trained on CORD-19 articles, performs worse than regular SciBERT, suggesting potential issues in training choices or noisiness in the data. Notably, LUKE, trained solely on Wikipedia data, emerges as the best baseline model, possibly due to its entity-recognition pre-training objective, which aids in identifying relevant entities for QA tasks (Van Aken et al., 2019) and highlighting the need for entity representations in closed-domains. Models capable of handling longer context i.e. Longformer, BigBird and XLNet do not show marked improvements. XLNet, degrades completely on COVID-QA, potentially due to token permutation hindering its reasoning across large and conceptually dense contexts.

6.1.2 `TOP-Training`

Wiki Baseline: Fine-Tuning BERT on Wikipedia yields marginal improvement of 0.9% EM and 1.2% F1; RoBERTa shows a 3.7% increase in EM and a 1.3% increase in F1. Our Wikipedia corpus, while small, contains relevant information per-

Table 1: Baseline Bio Models (COVID-QA). [1](Peng et al., 2019); [2](Yuan et al., 2022); [3](Beltagy et al., 2020); [4](Zaheer et al., 2020); [5](Yamada et al., 2020); Blue = best/red = worst scores overall; **bold = best decoder**

Model	**Pre-Training Corpus**	**Corpus Size**	**EM**	**F1**
BioBERT	PubMed	4.5B words	37.62 ± 0.18	65.73 ± 0.35
SciBERT	Semantic Scholar	3.2B words	37.52 ± 0.23	65.58 ± 0.18
SciBERT(+CORD-19)	Semantic Scholar + CORD-19	3.2B words + 20GB	35.61 ± 0.30	63.60 ± 0.59
PubMedBERT	PubMed	3.1B words / 21GB	39.87 ± 0.74	68.47 ± 0.13
BlueBERT [1]	PubMed + MIMIC	4.5B words	27.35 ± 0.30	52.18 ± 0.40
CODER [2]	Unified Medical Language System	NA	39.33 ± 0.47	67.01 ± 0.30
Longformer [3]	Books + Wiki + RealNews + Stories	6.5B tokens	37.79 ± 0.39	66.58 ± 0.22
BigBird [4]	Books + CC-News + Stories + Wiki	Ĩ60GB (same as RoBERTa)	32.79 ± 0.13	60.06 ± 0.41
LUKE [5]	Wikipedia	3.5B words	41.01 ± 0.30	68.23 ± 0.19
XLNET	BooksCorpus + Wikipedia + Giga5 + ClueWeb 2012-B + Common Crawl	32.89B words	2.45 ± 0.08	8.64 ± 0.19
Galactica	c.f. section 4	106B tokens	0 (0)	5.01 (11.11)
MedLLaMA	Medical Corpora	NA	0 (0)	**5.81 (12.79)**
MedAlpaca	Medical Meadow	NA	**0.03 (0.2)**	5.21 (12.73)

Table 2: Baseline Bio Models (RadQA). H(F1): HasAns_F1, H(EM): HasAns_EM; Blue = best/red = worst scores overall; **bold = best decoder**; [1](Yan et al., 2022); [2](Alsentzer et al., 2019); [3](Gururangan et al., 2020)

Model	**Corpus**	**Corpus Size**	**Dev**				**Test**			
			EM	**F1**	**H(EM)**	**H(F1)**	**EM**	**F1**	**H(EM)**	**H(F1)**
BioBERT	PubMed	4.5B words	26.42 ± 0.49	44.26 ± 0.09	40.79 ± 0.76	68.31 ± 0.14	49.95 ± 1.08	63.32 ± 0.40	45.65 ± 1.21	63.50 ± 0.57
SciBERT	Semantic Scholar	3.2B words	27.03 ± 0.32	44.40 ± 0.06	41.65 ± 0.62	68.45 ± 0.22	53.04 ± 0.38	67.17 ± 0.73	48.62 ± 0.70	67.49 ± 1.02
PubMedBERT	PubMed	3.1B words / 21GB	31.45 ± 0.17	47.89 ± 0.46	48.40 ± 0.27	73.77 ± 0.62	54.07 ± 0.71	68.76 ± 0.22	49.49 ± 0.87	69.09 ± 0.80
BlueBERT	PubMed + MIMIC	4.5B words	30.08 ± 1.33	47.14 ± 0.81	46.12 ± 2.01	73.44 ± 2.78	54.99 ± 1.91	68.11 ± 1.41	48.55 ± 1.66	66.06 ± 1.40
CODER	UMLS	N/A†	40.50 ± 1.31	57.32 ± 1.74	47.37 ± 1.70	73.34 ± 1.29	53.74 ± 0.71	68.36 ± 0.36	49.86 ± 0.50	69.36 ± 1.00
LUKE	Wikipedia	3.5 billion words	27.44 ± 0.70	44.77 ± 0.40	42.35 ± 1.08	69.10 ± 0.62	50.92 ± 1.26	64.47 ± 1.75	46.16 ± 0.25	64.25 ± 1.28
RadBERT[1]	Radiology reports	2.6 GB	30.34 ± 1.50	48.00 ± 1.43	45.73 ± 0.68	73.00 ± 0.73	54.40 ± 2.84	67.34 ± 1.74	51.52 ± 0.87	68.80 ± 0.80
ClinicalBERT[2]	MIMIC	0.5B words / 3.7GB	27.18 ± 1.89	44.69 ± 0.54	41.88 ± 2.86	68.90 ± 0.71	50.27 ± 1.63	63.40 ± 1.52	46.89 ± 0.13	64.41 ± 0.16
BioMed-RoBERTa[3]	S2ORC	7.55B tokens / 47GB	27.44 ± 1.10	45.44 ± 0.64	42.35 ± 1.70	70.14 ± 0.99	52.82 ± 0.57	66.52 ± 0.32	48.62 ± 0.33	66.91 ± 0.82
Galactica	c.f. section 4	106B tokens	1.37	8.5	1.37	8.5	0.49	10.23	0.49	10.23
MedLLaMA	Medical Corpora	NA	0.3	10.63	0.3	10.63	0.16	12.14	0.16	12.14
MedAlpaca	Medical Meadow	NA	**1.68**	**15.18**	**1.68**	**15.18**	**1.3**	**16.95**	**1.3**	**16.95**

taining to COVID literature, which in turn aids in answering related questions.

47k corpus: With `TOP-Training`, BERT achieves a 4.01% increase in EM and a 3.5% increase in F1, while RoBERTa shows a 6.7% increase in EM and a 2.5% increase in F1, **setting a new SOTA on COVID-QA**. RoBERTa even outperforms the previous SOTA model (LUKE) by 1.2% in EM and 1.3% in F1, despite using a training corpus significantly smaller (67.4 MB/0.032B words) than LUKE's 3.5B-word corpus (0.9% of the size). Moreover, any variation of our approach, ablation or otherwise, improves performance for both models over the Wikipedia baseline. These results indicate that our models benefit from additional training on corpora aligned with entity information from the downstream dataset.

470k [10x] corpus: Training with a 10x corpus (10 contexts per entity) led to the most improvements for BERT with EM increasing by 8.2% and F1 by 6.4%. This is consistent with Liu et al. (2019) who argue that BERT was *significantly undertrained*. Though this improvement does not achieve RoBERTa performance, it demonstrates the scalability of our approach. RoBERTa improves too, but, not as much as when using the base 47k corpus. Undertrained BERT is more *malleable* to learning new concepts whereas RoBERTa seems to have hit its *ceiling for learning* in this domain.

Filtration - 50k corpus: Surprisingly, on removing ill-formed entities, Table 3 (Row 5) the performance declined with respect to the best BERT (10x corpus - row 4) and best RoBERTa (base 47k corpus - row 3) model. Our regex-based filtering rules mistakenly (as they cannot distinguish between true/false patterns) removed entities relevant to research articles such as author names or URLs, leading to the decline in performance.

Reduced context length - 47k (at most 1k context tokens) corpus: due to limitations in Galactica's token generation i.e. at most 2048 tokens (last row of Table 3). Both models perform worse on both metrics as Galactica is unable to generate content matching the style of COVID-QA's con-

Table 3: `TOP-Training`(COVID-QA). Time#: to generate corpus; ♣: entity filter; Gal = Galactica; max_length = Context Max Length. Blue = best/red = worst scores overall; **bold = best BERT/RoBERTa setup.**

Train Dataset	Time#	Corpus Size	Model	EM	F1
NA [Vanilla Fine-Tuning]	NA	NA	BERT	33.62 ± 0.59	60.01 ± 0.36
			RoBERTa	38.89 ± 0.52	67.44 ± 0.47
Wikipedia	≈2.5 hrs	139.6 MB	BERT	33.95 ± 0.13	60.76 ± 0.78
			RoBERTa	40.33 ± 0.60	68.30 ± 0.54
Gal(47k)	≈ 6.5 hrs	67.4 MB	BERT	34.97 ± 0.18	62.11 ± 0.32
			RoBERTa	**41.51** ± **0.48**	**69.10** ± **0.27**
Gal(470k) [10x]	≈ 2.5 days	558.2 MB	BERT	**36.39** ± **0.27**	**63.84** ± **1.16**
			RoBERTa	41.31 ± 0.22	68.84 ± 0.28
Gal(25k*2 = 50k)♣	≈ 6.5 hrs	64.0 MB	BERT	35.03 ± 0.38	62.14 ± 0.48
			RoBERTa	41.36 ± 0.35	69.00 ± 0.52
Gal(47k) [max_length = 1k]	≈ 2.5 hrs	44.8 MB	BERT	34.90 ± 0.14	62.02 ± 0.95
			RoBERTa	41.57 ± 0.33	68.98 ± 0.31

texts (research papers), underscoring the importance of domain-aware writing styles for adaptation pipelines.

6.2 RadQA

Results on RadQA are presented for both its validation and test splits. We consider various combinations of contexts (prompts) and entity filtration. Higher test scores on average are observed as compared to validation scores, which we attribute to fewer unanswerable questions in the test set (154 vs. 231) and slightly shorter contexts (73.82 vs. 78.1 tokens). We found no information leakage. We report scores for both, but mainly focus our analysis on the validation set.

6.2.1 Baselines

On the RadQA benchmark dev set, CODER, a PubMEDBERT checkpoint, has the best EM & F1 but suffers slightly v/s PubMedBERT on only answerable questions presumably because CODER learned clinical embeddings from the UMLS knowledge graph with radiology terms.

Surprisingly, PubMed/Blue-BERT perform similarly on both the dev and test sets. Theoretically, BlueBERT should have performed better being pretrained on MIMIC clinical notes. RadBERT, which is a superior RoBERTa architecture, and specifically trained on radiology reports did not perform well overall. Although it marginal improved over PubMed/Blue-BERT, it comes at the cost of a fraction of the training data. This again indicates the importance of proper domain alignment i.e. *what* data the models are trained on.

Unfortunately, LUKE performed poorly as compared to Bio/Sci-BERT, showing little, v/s both on dev, to no gain, v/s SciBERT on test, during evaluation. The impact of writing styles in the training corpora is clearly evident in the performance gap between Clinical/Rad-BERT. While the former was trained on more clinical data, it was not the *right* type of data i.e. radiology reports, leading RadBERT to outperform it on both splits.

6.2.2 `TOP-Training`

Wiki Baseline: Training on the Wikipedia corpus, BERT shows an increase in EM only (2.8%) on the dev set, but an overall improvement in all measures on the test set compared to vanilla fine-tuning. RoBERTa, on the other hand, showed overall improvement on the dev set while test scores suffered compared to the the vanilla baseline.

Normal Prompting: Training on the unfiltered corpus with normal prompts (row 3) improved BERT and RoBERTa 3.7% EM, and 1.4% F1 for RoBERTa, over the vanilla baseline, which is greater than that of our Wiki baseline. However, **when the filter is applied (row 4)** BERT shows $\sim$7% EM & 0.5% F1 increase while RoBERTa shows $\tilde{6}$% EM & $\tilde{3}$% F1 increase over regular fine-tuning. These scores indicate the necessity of using even a simple filter for selecting a subset of training corpora.

Best BERT setting: When both filtered entities and the corpus from the fancy prompt are used (row 6), improvements over basic fine-tuning (8.1% EM, 0.4% F1 on dev) and the Wikipedia baseline (5.2% EM, 1.1%F1 on dev) occur. Note: a) the benefit of studying RadQA and designing targeted prompts, and b) that BERT reaches these scores with a modest 34.3MB corpus, which is much smaller than the benchmarked models.

Best RoBERTa setting: RoBERTa demonstrates improvements across different combinations of filtration methods, prompt styles and using the Wikipedia corpus. However, the improvements are inconsistent with respect to a specific approach. Excluding the combined corpora settings, we see it achieve the best performance, on validation, using the corpus obtained from the filtered entities

Table 4: `TOP-Training` Results (RadQA). H(F1): HasAns_F1, H(EM): HasAns_EM; *: [Vanilla Fine-Tuning]. [†: normal, ‡: fancy] prompt, ♣: entity filter. Blue = best/red = worst overall; **bold = best BERT/RoBERTa setup**.

Train dataset / Corpus Size	**Time#**	**Model**	**Dev**				**Test**			
			EM	**F1**	**H(EM)**	**H(F1)**	**EM**	**F1**	**H(EM)**	**H(F1)**
NA*	NA	BERT	23.83 ± 0.49	42.91 ± 0.46	36.79 ± 0.76	66.23 ± 0.70	46.20 ± 1.96	59.42 ± 1.10	40.65 ± 1.15	58.30 ± 0.74
		RoBERTa	26.12 ± 0.69	43.83 ± 0.44	40.31 ± 1.06	67.65 ± 0.68	51.68 ± 0.73	64.94 ± 0.62	46.45 ± 0.66	64.14 ± 0.42
Wikipedia/18.4 MB	≈30 mins	BERT	24.49 ± 0.38	42.62 ± 0.08	37.80 ± 0.59	65.79 ± 0.13	47.40 ± 1.85	60.11 ± 1.59	41.95 ± 1.73	58.92 ± 1.56
		RoBERTa	27.19 ± 0.32	44.49 ± 0.31	41.96 ± 0.49	68.68 ± 0.47	50.54 ± 0.62	63.43 ± 0.75	45.72 ± 0.50	62.92 ± 0.67
Galactica(≈55k) †/ 81.6 MB	≈11 hrs	BERT	24.70 ± 0.46	42.88 ± 0.30	38.12 ± 0.71	66.19 ± 0.46	46.74 ± 1.84	59.69 ± 0.62	41.60 ± 2.31	58.88 ± 1.64
		RoBERTa	27.09 ± 0.70	44.42 ± 0.55	41.81 ± 1.09	68.56 ± 0.86	51.25 ± 0.41	64.41 ± 0.95	46.45 ± 0.66	64.01 ± 0.71
Galactica(≈55k) †♣/ 80.3 MB	≈11 hrs	BERT	25.51 ± 0.38	43.11 ± 0.23	40.38 ± 2.31	64.76 ± 2.75	46.36 ± 1.79	59.41 ± 1.67	46.59 ± 8.06	57.83 ± 1.81
		RoBERTa	27.69 ± 0.23	45.16 ± 0.35	42.74 ± 0.36	69.70 ± 0.54	51.14 ± 0.49	64.29 ± 0.38	46.31 ± 1.36	63.85 ± 0.76
Galactica(≈55k) ‡/ 38.1 MB	≈11 hrs	BERT	25.10 ± 0.32	42.78 ± 0.55	38.74 ± 0.49	66.03 ± 0.84	46.85 ± 1.74	59.54 ± 1.19	41.66 ± 0.45	58.60 ± 0.37
		RoBERTa	27.64 ± 0.49	44.99 ± 0.11	42.67 ± 0.76	69.45 ± 0.16	52.39 ± 0.80	65.57 ± 0.93	47.76 ± 1.85	65.34 ± 1.97
Galactica(≈55k) ‡♣/ 34.3 MB	≈11 hrs	BERT	**25.76 ± 0.66**	**43.10 ± 0.27**	**39.68 ± 1.11**	**66.44 ± 0.55**	46.52 ± 1.20	58.98 ± 1.09	40.44 ± 0.95	57.06 ± 0.94
		RoBERTa	27.08 ± 0.46	44.67 ± 0.23	41.81 ± 0.72	68.95 ± 0.35	51.30 ± 1.17	64.14 ± 0.41	47.39 ± 1.64	64.53 ± 0.85
Galactica(≈100k) †‡/ 120.8 MB	≈22 hrs	BERT	25.10 ± 0.54	42.74 ± 0.06	38.74 ± 0.83	65.97 ± 0.09	47.07 ± 1.23	60.04 ± 1.19	42.46 ± 0.13	59.78 ± 0.41
		RoBERTa	27.49 ± 0.49	44.82 ± 0.38	42.43 ± 0.75	69.17 ± 0.58	51.79 ± 1.98	64.99 ± 1.80	47.03 ± 1.45	64.45 ± 1.21
Galactica(≈100k) † ‡ ♣/ 115.6 MB	≈22 hrs	BERT	24.75 ± 0.23	42.96 ± 0.10	38.20 ± 0.36	66.31 ± 0.16	47.50 ± 1.06	60.38 ± 0.48	41.96 ± 1.31	59.14 ± 1.97
		RoBERTa	**27.85 ± 0.09**	**45.16 ± 0.11**	**42.98 ± 0.14**	**69.71 ± 0.18**	51.79 ± 0.59	64.77 ± 0.40	46.88 ± 1.28	64.21 ± 0.75

and normal prompting (row 4) and on test, using filtered entities and fancy prompting (row 6).

RoBERTa v/s Benchmarks: In row 4, RoBERTa, outperforms Bio/Sci-BERT and LUKE on all metrics along with long-context models BigBird & Longformer. Interestingly, this RoBERTa even beats out BioClinicalBERT which was trained using much higher quality clinical notes scoring 1.9% EM & 1.1% F1 more than it.

Combined Prompting Styles: Merge the contexts from both prompt styles (rows 7 and 8) for the filtered and the unfiltered entities separately. BERT shows better performance in row 7 i.e. using the unfiltered corpus (1.7% EM increase over the filtered variant and roughly same F1) and RoBERTa in row 8 i.e. using the filtered corpus (1.3% EM and 0.8%F1 over the unfiltered variant). Overall, this version of RoBERTa resulted in our best overall model suggesting that incorporating a mixture of prompt styles will create more diverse corpora, enhancing domain alignment.

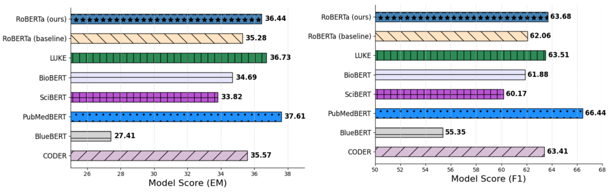


Figure 2: Information Leakage Validation Trials (Left - EM | Right - F1): RoBERTa (ours) was trained on a subset of the 47k corpus with entities only from the 80% train set. All of the models were fine-tuned in the usual manner i.e. SQuAD→COVID-QA (80% train set) and evaluated on the 20% test set.

6.3 Investigating Information Leakage

Given that the synthetic corpus generated for COVID-QA in §5 contains entities identified in the *entire* COVID-QA dataset - not from the *train* split within each *fold* - we explore if the performance gains from `TOP-Training` are a result of information leak. To this end, we construct a roughly 80%/20% train/test split (1,676/343 records), ensuring no context overlap, and apply a suite of models to this new split. When applying our `TOP-Training`, a synthetic corpus is generated *only* from entities identified in the train split.

RoBERTa subjected to `TOP-Training` still yields strong performance in this restricted scenario, only surpassed by PubMedBERT (and marginally by LUKE in EM) (Fig. 2) demonstrating that the improved performance on COVID-QA cannot be attributed to information leak from the test set. Although the scores are lower than those in Table 3, the relative scores produced by each model leads to a similar conclusion that `TOP-Training` yields optimal results.

7 Conclusion

In this study, we introduced `TOP-Training`, an innovative pretraining technique designed to enhance the alignment of a LLM with rare-domain target problems. The distinctive feature of `TOP-Training` lies in its automated synthesis of target-oriented data with the assistance of a LLM. Our experiments on medical EQA demonstrate the effectiveness of `TOP-Training`, shedding light on the limitations of widely used autoregressive LLMs.

Limitations

We identify two limitations of our work. First, we use a number of GPUs to generate our corpus. While we were fortunate to have access to powerful computing clusters, this could form a bottleneck when being deployed on low-end hardware. However, with cloud services being made more and more affordable, we feel that this point can only be a deal-breaker in severely budget-constrained settings. And second, in this study, we have only shown how to generate corpora for the biomedical domain. For an even wider applicability, we need to study generation techniques for other closed domains such as Finance, Law, Aviation, etc.

Ethics Statement

Our work relied on publicly available datasets. Ethical ramifications here are limited. To use RadQA, we had to acquire certifications to access it. Even though it is deidentified, we store the data in a protected fashion and have attempted not to quote even proxy identifiers that can be used to decode a person's identity.

Indirect hazards may be that our improved Q&A methodology can be used to ask questions that can be used to design bio-attacks on humans but such abusive uses are there with all technology and we strongly believe that the benefits world-wide of an improved closed-domain Q&A system is much bigger than the risks involved due to this technology.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Helena Balabin, Charles Tapley Hoyt, Colin Birkenbihl, Benjamin M Gyori, John Bachman, Alpha Tom Kodamullil, Paul G Plöger, Martin Hofmann-Apitius, and Daniel Domingo-Fernández. 2022. Stonkgs: a sophisticated transformer trained on biomedical text and knowledge graphs. *Bioinformatics*, 38(6):1648–1656.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

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.

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*.

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.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Michael P Hartung, Ian C Bickle, Frank Gaillard, and Jeffrey P Kanne. 2020. How to create a great radiology report. *RadioGraphics*, 40(6):1658–1670.

Xuanli He, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. 2022. Generate, annotate, and learn: NLP with synthetic text. *Transactions of the Association for Computational Linguistics*, 10:826–842.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. 2023. Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models. *PLoS digital health*, 2(2):e0000198.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

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

Man Luo, Kazuma Hashimoto, Semih Yavuz, Zhiwei Liu, Chitta Baral, and Yingbo Zhou. 2022. Choose your QA model wisely: A systematic study of generative and extractive readers for question answering. In *Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge*, pages 7–22, Dublin, Ireland and Online. Association for Computational Linguistics.

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

Kurt Micallef, Albert Gatt, Marc Tanti, Lonneke van der Plas, and Claudia Borg. 2022. Pre-training data quality and quantity for a low-resource language: New corpus and BERT models for Maltese. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 90–101, Hybrid. Association for Computational Linguistics.

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.

Milad Moradi, Kathrin Blagec, Florian Haberl, and Matthias Samwald. 2021. Gpt-3 models are poor few-shot learners in the biomedical domain. *arXiv preprint arXiv:2109.02555*.

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, et al. 2023. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy. Association for Computational Linguistics.

Charith Peris, Lizhen Tan, Thomas Gueudre, Turan Gojayev, Pan Wei, and Gokmen Oz. 2022. Knowledge distillation transfer sets and their impact on downstream NLU tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 128–137, Abu Dhabi, UAE. Association for Computational Linguistics.

Barbara Plank. 2016. What to do about non-standard (or non-canonical) language in NLP. In *Proceedings of the 13th Conference on Natural Language Processing, KONVENS 2016, Bochum, Germany, September 19-21, 2016*, volume 16 of *Bochumer Linguistische Arbeitsberichte*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Sarvesh Soni, Meghana Gudala, Atieh Pajouhi, and Kirk Roberts. 2022. RadQA: A question answering dataset to improve comprehension of radiology reports. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6250–6259, Marseille, France. European Language Resources Association.

MosaicML NLP Team et al. 2023. Introducing mpt-30b: Raising the bar for open-source foundation models.

Betty Van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. 2019. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1823–1832.

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.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625.

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*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

An Yan, Julian McAuley, Xing Lu, Jiang Du, Eric Y Chang, Amilcare Gentili, and Chun-Nan Hsu. 2022. Radbert: Adapting transformer-based language models to radiology. *Radiology: Artificial Intelligence*, 4(4):e210258.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

Zheng Yuan, Zhengyun Zhao, Haixia Sun, Jiao Li, Fei Wang, and Sheng Yu. 2022. Coder: Knowledge-infused cross-lingual medical term embedding for term normalization. *Journal of biomedical informatics*, 126:103983.

Zhenrui Yue, Bernhard Kratzwald, and Stefan Feuerriegel. 2021. Contrastive domain adaptation for question answering using limited text corpora. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9575–9593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

A Token Filtering

We performed entity filtering as a common ablation technique for both datasets. We used regular expressions to remove entities with special characters such as *, !, etc., as well as specific text patterns like `https*` and `baby`. We implemented a length-based filter, retaining only entities longer than a certain number of characters. Additionally, for COVID-QA, we applied a second round of filtration using TF-IDF, considering the questions + context as the corpus and retaining the top 25k entities with the highest IDF scores. However, as this approach did not lead to substantial gains, we decided not to use it for RadQA. Due to the large number of possible combinations, we did not extensively explore these settings in our experiments.

B Synthetic Corpora Samples

We provide samples from our generated corpus. In Figure 3 we show two positive results for our COVID-QA directed corpus. The top one is in the style of a research paper while the bottom one, though shorter in length, details useful information on the required entity. We show negative examples of the same in Figure 4. The top one completely degrades into noise while the bottom one although coherent is talking about an unrelated topic.

For RadQA, we show similar examples. In Figure 5 we show positive (top) and negative (bottom) samples from normal prompting. Note that since programming languages were a part of Galactica's training corpora, it sometimes generates the same during prompting. Finally, Figure 6 shows postive (top) and negative (bottom) examples of fancy prompting. While the positive sample shows a strong correlation with an actual radiology report, the bottom one completely breaks down.

C ChatGPT trials

We were curious to see if ChatGPT (with GPT3.5) would be able to answer a few questions from COVID-QA. Only the questions were provided to it without the context since they were much longer than its input capacity. We did not test RadQA since its questions were heavily linked to the patient report on hand and were not as *general* as the former's. Figure 7 shows a few examples from our trials. As we can see, it seems to answer the questions correctly. However, as mentioned before, these answers need validation from a professional in the field. Also, we see that sometimes it **cannot answer** questions on topics not found in its training data, a problem not faced by our encoder models which rely only on the provided context.

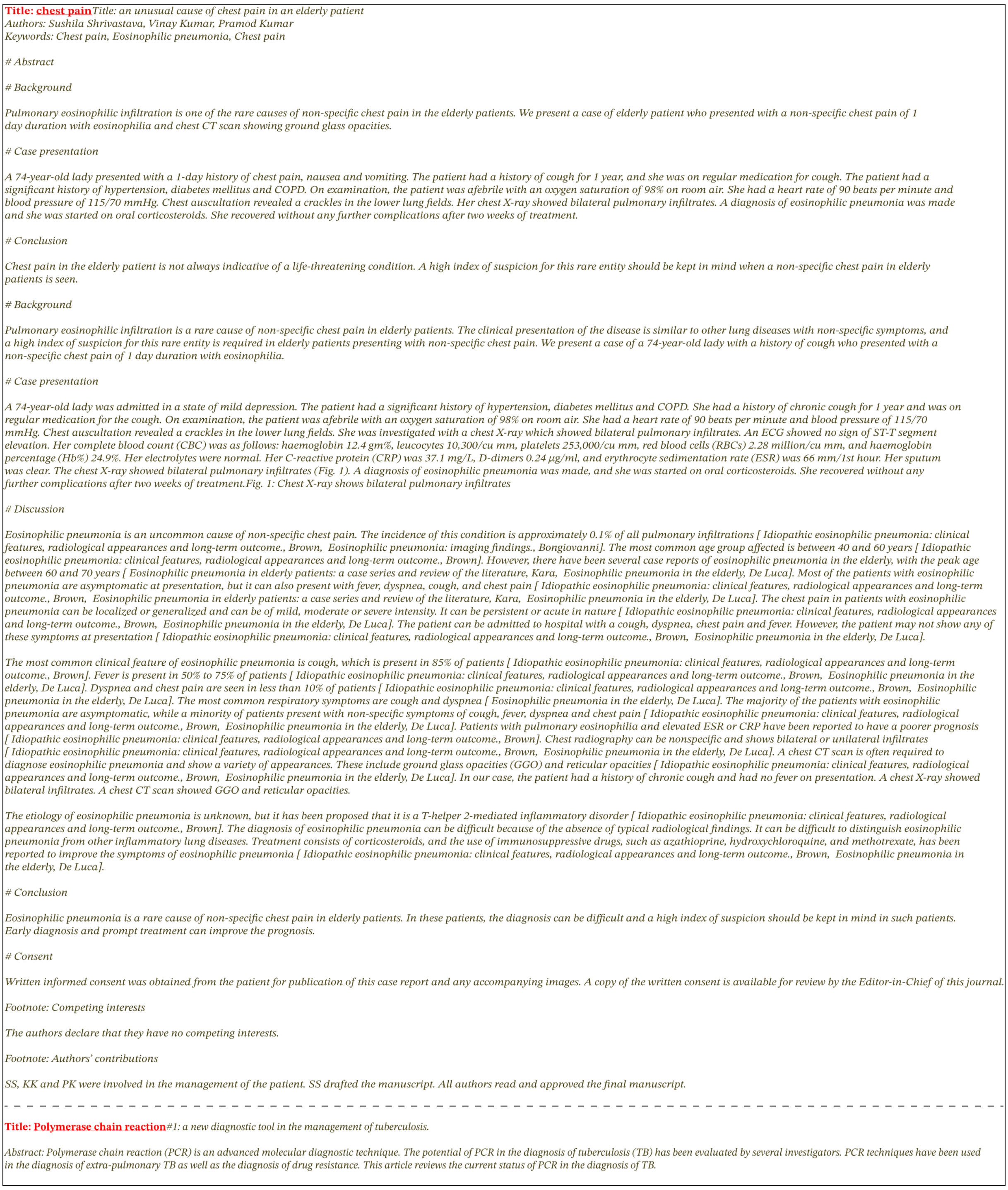
D Hardware Details

To run our corpus generation code, we used a total of 8 40GB NVIDIA A100 or 8 32GB NVIDIA Tesla V100 cards, depending on our institutes job scheduler and availability. Although the former is a superior card, we preferred the latter since it gives relatively quick turnaround and is a general purpose GPU unlike the A100 which is required for even heavier compute. In fact we were requested by our system administrator to use the V100 over the A100, if it was possible, to avoid blocking off even more compute-heavy experiments, which we did not have an issue obeying. That being said, if the A100 cluster was free, we did take advantage of it to run our code. For pre-training and fine-tuning BERT/RoBERTa, we used only 1 A100 80GB GPU as RoBERTa was unable to fit on the 40/32 GB cards owing to the large batch size.

E Prompts Tried

As mentioned before, we knew the prompt to be used for COVID-QA since it was a feature of Galactica. However, we had to experiment with a few prompt styles for RadQA before landing on our final one. We describe the prompts we tried below for the entity `small-bowel injury` and the reasons for not using them. The text in `courier` is the prompt and in *italics* is the generated text.

- `IMPRESSION: small-bowel injury` *(SBI) was defined as injury to the small bowel wall, or transmural disruption of the bowel wall, in the absence of penetrating abdominal trauma (The management of small-bowel injuries in the emergency room., Brown). Patients were excluded if the SBIs occurred after admission to the emergency department or were complicated by perforation. ...* (truncated)
    - Issue - Did not include the `FINDINGS` section.
- `FINDINGS: small-bowel injury` *was observed in 11.1% and 21.9% of patients, respectively. Small-bowel injury associated with liver injury occurred in 10.1%. The mortality rate was significantly higher in the small-bowel injury group than in the non-small-bowel injury group (11.6% vs. 2.1%, P = 0.005). This study shows that small-bowel injury is associated with a high mortality rate, and that this complication is an important cause of morbidity and mortality in patients with severe acute pancreatitis.*
    - Issue - Did not include the `IMPRESSIONS` section.
- `Patient has small-bowel injury. FINAL REPORT` *This patient was treated with conservative methods. The small-bowel injury was treated surgically with resection of the bowel, primary anastomosis, and end-to-end anastomosis between the ends of the resected bowel. She did not have recurrence of intestinal obstruction after 3 years.*
    - Issue - Did not include either `IMPRESSIONS` or `FINDINGS` section.
- `John Doe has small-bowel injury. IMPRESSIONS` *: This*



Title: chest pain Title: an unusual cause of chest pain in an elderly patient
Authors: Sushila Shrivastava, Vinay Kumar, Pramod Kumar
Keywords: Chest pain, Eosinophilic pneumonia, Chest pain

Abstract

Background

Pulmonary eosinophilic infiltration is one of the rare causes of non-specific chest pain in the elderly patients. We present a case of elderly patient who presented with a non-specific chest pain of 1 day duration with eosinophilia and chest CT scan showing ground glass opacities.

Case presentation

A 74-year-old lady presented with a 1-day history of chest pain, nausea and vomiting. The patient had a history of cough for 1 year, and she was on regular medication for cough. The patient had a significant history of hypertension, diabetes mellitus and COPD. On examination, the patient was afebrile with an oxygen saturation of 98% on room air. She had a heart rate of 90 beats per minute and blood pressure of 115/70 mmHg. Chest auscultation revealed a crackles in the lower lung fields. Her chest X-ray showed bilateral pulmonary infiltrates. A diagnosis of eosinophilic pneumonia was made and she was started on oral corticosteroids. She recovered without any further complications after two weeks of treatment.

Conclusion

Chest pain in the elderly patient is not always indicative of a life-threatening condition. A high index of suspicion for this rare entity should be kept in mind when a non-specific chest pain in elderly patients is seen.

Background

Pulmonary eosinophilic infiltration is a rare cause of non-specific chest pain in elderly patients. The clinical presentation of the disease is similar to other lung diseases with non-specific symptoms, and a high index of suspicion for this rare entity is required in elderly patients presenting with non-specific chest pain. We present a case of a 74-year-old lady with a history of cough who presented with a non-specific chest pain of 1 day duration with eosinophilia.

Case presentation

A 74-year-old lady was admitted in a state of mild depression. The patient had a significant history of hypertension, diabetes mellitus and COPD. She had a history of chronic cough for 1 year and was on regular medication for the cough. On examination, the patient was afebrile with an oxygen saturation of 98% on room air. She had a heart rate of 90 beats per minute and blood pressure of 115/70 mmHg. Chest auscultation revealed a crackles in the lower lung fields. She was investigated with a chest X-ray which showed bilateral pulmonary infiltrates. An ECG showed no sign of ST-T segment elevation. Her complete blood count (CBC) was as follows: haemoglobin 12.4 gm%, leucocytes 10,300/cu mm, platelets 253,000/cu mm, red blood cells (RBCs) 2.28 million/cu mm, and haemoglobin percentage (Hb%) 24.9%. Her electrolytes were normal. Her C-reactive protein (CRP) was 37.1 mg/L, D-dimers 0.24 µg/ml, and erythrocyte sedimentation rate (ESR) was 66 mm/1st hour. Her sputum was clear. The chest X-ray showed bilateral pulmonary infiltrates (Fig. 1). A diagnosis of eosinophilic pneumonia was made, and she was started on oral corticosteroids. She recovered without any further complications after two weeks of treatment.Fig. 1: Chest X-ray shows bilateral pulmonary infiltrates

Discussion

Eosinophilic pneumonia is an uncommon cause of non-specific chest pain. The incidence of this condition is approximately 0.1% of all pulmonary infiltrations [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown, Eosinophilic pneumonia: imaging findings., Bongiovanni]. The most common age group affected is between 40 and 60 years [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown]. However, there have been several case reports of eosinophilic pneumonia in the elderly, with the peak age between 60 and 70 years [Eosinophilic pneumonia in elderly patients: a case series and review of the literature, Kara, Eosinophilic pneumonia in the elderly, De Luca]. Most of the patients with eosinophilic pneumonia are asymptomatic at presentation, but it can also present with fever, dyspnea, cough, and chest pain [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown, Eosinophilic pneumonia in elderly patients: a case series and review of the literature, Kara, Eosinophilic pneumonia in the elderly, De Luca]. The chest pain in patients with eosinophilic pneumonia can be localized or generalized and can be of mild, moderate or severe intensity. It can be persistent or acute in nature [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown, Eosinophilic pneumonia in the elderly, De Luca]. The patient can be admitted to hospital with a cough, dyspnea, chest pain and fever. However, the patient may not show any of these symptoms at presentation [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown, Eosinophilic pneumonia in the elderly, De Luca].

The most common clinical feature of eosinophilic pneumonia is cough, which is present in 85% of patients [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown]. Fever is present in 50% to 75% of patients [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown, Eosinophilic pneumonia in the elderly, De Luca]. Dyspnea and chest pain are seen in less than 10% of patients [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown, Eosinophilic pneumonia in the elderly, De Luca]. The most common respiratory symptoms are cough and dyspnea [Eosinophilic pneumonia in the elderly, De Luca]. The majority of the patients with eosinophilic pneumonia are asymptomatic, while a minority of patients present with non-specific symptoms of cough, fever, dyspnea and chest pain [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown, Eosinophilic pneumonia in the elderly, De Luca]. Patients with pulmonary eosinophilia and elevated ESR or CRP have been reported to have a poorer prognosis [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown]. Chest radiography can be nonspecific and shows bilateral or unilateral infiltrates [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown, Eosinophilic pneumonia in the elderly, De Luca]. A chest CT scan is often required to diagnose eosinophilic pneumonia and show a variety of appearances. These include ground glass opacities (GGO) and reticular opacities [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown, Eosinophilic pneumonia in the elderly, De Luca]. In our case, the patient had a history of chronic cough and had no fever on presentation. A chest X-ray showed bilateral infiltrates. A chest CT scan showed GGO and reticular opacities.

The etiology of eosinophilic pneumonia is unknown, but it has been proposed that it is a T-helper 2-mediated inflammatory disorder [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown]. The diagnosis of eosinophilic pneumonia can be difficult because of the absence of typical radiological findings. It can be difficult to distinguish eosinophilic pneumonia from other inflammatory lung diseases. Treatment consists of corticosteroids, and the use of immunosuppressive drugs, such as azathioprine, hydroxychloroquine, and methotrexate, has been reported to improve the symptoms of eosinophilic pneumonia [Idiopathic eosinophilic pneumonia: clinical features, radiological appearances and long-term outcome., Brown, Eosinophilic pneumonia in the elderly, De Luca].

Conclusion

Eosinophilic pneumonia is a rare cause of non-specific chest pain in elderly patients. In these patients, the diagnosis can be difficult and a high index of suspicion should be kept in mind in such patients. Early diagnosis and prompt treatment can improve the prognosis.

Consent

Written informed consent was obtained from the patient for publication of this case report and any accompanying images. A copy of the written consent is available for review by the Editor-in-Chief of this journal.

Footnote: Competing interests

The authors declare that they have no competing interests.

Footnote: Authors' contributions

SS, KK and PK were involved in the management of the patient. SS drafted the manuscript. All authors read and approved the final manuscript.

Title: Polymerase chain reaction#1: a new diagnostic tool in the management of tuberculosis.

Abstract: Polymerase chain reaction (PCR) is an advanced molecular diagnostic technique. The potential of PCR in the diagnosis of tuberculosis (TB) has been evaluated by several investigators. PCR techniques have been used in the diagnosis of extra-pulmonary TB as well as the diagnosis of drug resistance. This article reviews the current status of PCR in the diagnosis of TB.

Figure 3: **COVID-QA positive** examples (underline = entity | red = prompt) | olive = generation

```
Title: virion perimetrical#1032; #1034; #1036; #1038; #1040; #1042; #1044; #1046; #1048; #1050; #1052; #1054; #1056; #1058; #1060; #1062; #1064; #1066; #1068; #1070; #1072; #1074; #1076; #1078; #1080; #1082; #1084; #1086; #1088; #1090; #1092; #1094; #1096; #1098; #1100;
#1102; #1104; #1106; #1108; #1110; #1112; #1114; #1116; #1118; #1120; #1122; #1124; #1126; #1128; #1130; #1132; #1134; #1136; #1138; #1140; #1142; #1144; #1146; #1148; #1150; #1152; #1154; #1156; #1158; #1160; #1162; #1164; #1166; #1168; #1170; #1172; #1174; #1176; #1178;
#1180; #1182; #1184; #1186; #1188; #1190; #1192; #1194; #1196; #1198; #1200; #1202; #1204; #1206; #1208; #1210; #1212; #1214; #1216; #1218; #1220; #1222; #1224; #1226; #1228; #1230; #1232; #1234; #1236; #1238; #1240; #1242; #1244; #1246; #1248; #1250; #1252; #1254; #1256;
#1258; #1260; #1262; #1264; #1266; #1268; #1270; #1272; #1274; #1276; #1278; #1280; #1282; #1284; #1286; #1288; #1290; #1292; #1294; #1296; #1298; #1300; #1302; #1304; #1306; #1308; #1310; #1312; #1314; #1316; #1318; #1320; #1322; #1324; #1326; #1328; #1330; #1332; #1334;
#1336; #1338; #1340; #1342; #1344; #1346; #1348; #1350; #1352; #1354; #1356; #1358; #1360; #1362; #1364; #1366; #1368; #1370; #1372; #1374; #1376; #1378; #1380; #1382; #1384; #1386; #1388; #1390; #1392; #1394; #1396; #1398; #1400; #1402; #1404; #1406; #1408; #1410; #1412;
#1414; #1416; #1418; #1420; #1422; #1424; #1426; #1428; #1430; #1432; #1434; #1436; #1438; #1440; #1442; #1444; #1446; #1448; #1450; #1452; #1454; #1456; #1458; #1460; #1462; #1464; #1466; #1468; #1470; #1472; #1474; #1476; #1478; #1480; #1482; #1484; #1486; #1488; #1490;
#1492; #1494; #1496; #1498; #1500; #1502; #1504; #1506; #1508; #1510; #1512; #1514; #1516; #1518; #1520; #1522; #1524; #1526; #1528; #1530; #1532; #1534; #1536; #1538; #1540; #1542; #1544; #1546; #1548; #1550; #1552; #1554; #1556; #1558; #1560; #1562; #1564; #1566; #1568;
#1570; #1572; #1574; #1576; #1578; #1580; #1582; #1584; #1586; #1588; #1590; #1592; #1594; #1596; #1598; #1600; #1602; #1604; #1606; #1608; #1610; #1612; #16
```

- -

```
Title: ADATitle: Automatic Data Augmentation for Text Classification
Authors: Koushik Ghosh, Kishore Kumar Chaudhuri, Siddharth Jha, Abhishek Kumar

# Abstract

A data set is small in size and is not representative of the entire population. So it is important to train a model that can perform well on unseen data, and this can be achieved by using various types of data augmentation techniques. Data augmentation techniques are applied to the data to create synthetic examples. However, most of the existing approaches of data augmentation are heuristic and do not make use of all the available information. In this paper, we propose a general framework to automatically generate the synthetic examples for a given dataset. The proposed framework is based on the observation that the model needs to be trained on the augmented data set for obtaining the better generalization performance. We show that using the proposed framework one can achieve up to \(2.4\%\) higher accuracy on a real-world data set with an average of \(6.4\%\) more data augmentation.

# I Introduction

An important step in supervised machine learning is the creation of a labelled data set for training a classifier. In this step, a large set of examples is obtained from an unlabeled data set and manually annotated with a label [ Data Augmentation for Text Classification: A Survey, Zhang]. A labelled data set can be very expensive and time consuming to create. One way to reduce the cost is to obtain the annotated data set using data augmentation techniques.

Data augmentation techniques are commonly used in image classification to create synthetic examples that are very similar to the original ones [ A survey on Image Data Augmentation for Deep Learning, Shorten,  mixup: Beyond Empirical Risk Minimization, Zhang,  mixup-v0: Improved Mixup for Data Augmentation, Kumar]. These techniques have also been extended to other data types such as text, audio, video etc. [ Data Augmentation for Text Classification: A Survey, Zhang]. The key idea is to generate synthetic examples which have similar characteristics as the original examples.

Data augmentation techniques are often heuristic in nature. They are designed to be intuitive and easy to use, but they do not make use of the available information and hence they often perform sub-optimally. In this paper, we propose a general framework to automatically generate the synthetic examples for a given dataset. The proposed framework is based on the observation that the model needs to be trained on the augmented data set for obtaining the better generalization performance. We show that using the proposed framework one can achieve up to \(2.4\%\) higher accuracy on a real-world data set with an average of \(6.4\%\) more data augmentation.

The main contributions of this work are as follows:

1. We propose a general framework to automatically generate the synthetic examples for a given dataset. The proposed framework is based on the observation that the model needs to be trained on the augmented data set for obtaining the better generalization performance.
2. We show that using the proposed framework one can achieve up to \(2.4\%\) higher accuracy on a real-world data set with an average of \(6.4\%\) more data augmentation.

# Ii Related work

# Ii-a Data Augmentation for Image Classification

There have been many proposed data augmentation techniques for image classification. For example, Cutout [ Improved Regularization of Convolutional Neural Networks with Cutout, Devries] and Cutmix [ CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features, Yun] techniques are proposed to cut out a portion of the image and paste it on the other part of the image in an random way. Data augmentation techniques proposed for text classification are also quite diverse. Examples of such techniques include: random erasing [ Random Erasing Data Augmentation, Zhong], semantic augmentation [ Semantic Data Augmentation for Deep Learning, Tao], random re-ordering of words [ Text Augmentation for Learning Natural Language Inference Models, Zhang], random word deletion [ Data Augmentation for Low-Resource Neural Machine Translation, Fadaee], etc. A detailed survey on image data augmentation techniques can be found in [ A survey on Image Data Augmentation for Deep Learning, Shorten].

# Ii-B Data Augmentation for Text Classification

Data augmentation techniques are very common in text classification tasks. The techniques proposed for text classification can be broadly divided into two categories: generative data augmentation and heuristic data augmentation. Generative data augmentation methods generate synthetic examples by training a generator model. For example, text completion [ A Hybrid Method for Text Classification with Generative Adversarial Network and Deep Learning, Jahan] and word dropout [ Data Augmentation for Low-Resource Neural Machine Translation, Fadaee] generate synthetic examples by training a generator model. Heuristic data augmentation techniques generate synthetic examples by performing some simple operations on the original examples. For example, Mixup [ mixup: Beyond Empirical Risk Minimization, Zhang], Cutmix [ CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features, Yun] and Cutout [ Improved Regularization of Convolutional Neural Networks with Cutout, Devries] use randomly selected examples from the original examples and add them together.

Generative data augmentation techniques have been extended to other data types such as images, audio, video etc. [ Data Augmentation for Text Classification: A Survey, Zhang]. However, the heuristic data augmentation techniques are used in the majority of the text classification tasks. Examples of heuristic data augmentation techniques for text classification include: the random swapping of two words [ Text Augmentation for Learning Natural Language Inference Models, Zhang], the random insertion of a random word at random position in a sentence [ Text Augmentation for Learning Natural Language Inference Models, Zhang], the random word deletion [ Data Augmentation for Low-Resource Neural Machine Translation, Fadaee], etc.

# Iii Problem Formulation

We consider a given data set \(X=\{x_{1},x_{2},\ldots,x_{N}\}\) where \(x_{i}\in\mathbb{R}^{d}\) is the \(i^{th}\) example with \(d\) dimension. The objective of supervised learning is to train a model \(f_{\theta}\) for a given classification task. The model \(f_{\theta}\) takes a set of input examples \(X\) and outputs a prediction of the class for each input example. We denote the set of labels for the data set as \(Y=\{y_{1},y_{2},\ldots,y_{N}\}\) where \(y_{i}\) is the label for the \(i^{th}\) example. The training set is composed of a set of pairs \(S=\{(x_{i},y_{i})\}_{i=1}^{N}\), where \((x_{i},y_{i})\) is the pair for the \(i^{th}\) example. The training objective is to find the model parameters \(\theta^{*}\) that minimizes the training error.

The data augmentation technique, \(T_{aug}\), is used to generate a set of augmented examples, \(X^{\prime}=\{x_{1}^{\prime},x_{2}^{\prime},\ldots,x_{N}^{\prime}\}\), from the original data set, \(X\). The generated data set \(X^{\prime}\) has similar characteristics as the original set \(X\).

The training objective is to find the model parameters \(\theta^{*}\) that minimizes the training error, \(e_{train}(\theta)\), using the augmented set \(X^{\prime}\). We denote the model trained using the augmented set \(X^{\prime}\) as \(f_{\theta}^{\prime}\).

# Iv Proposed Framework

Fig. 1: Framework of the proposed framework

We propose a framework for automatic data augmentation which is shown in Fig. 1. The proposed framework is based on the following observations: 1) the model needs to be trained on the augmented data set for obtaining the better generalization performance, and 2) using the proposed framework one can achieve up to \(2.4\%\) higher accuracy on a real-world data set with an average of \(6.4\%\) more data augmentation.

We use the concept of domain adaptation in our framework. This idea is based on the observation that a model needs to be trained on the augmented data set for obtaining the better generalization performance. Let \(\mathcal{L}_{train}\) be the training loss function, \(\mathcal{D}\) be the domain, \(T_{aug}\) be the data augmentation technique, \(X^{\prime}\) be the set of
```

Figure 4: **COVID-QA negative** examples (underline = entity | red = prompt) | olive = generation

fine-needle aspiration *of cervical lymph nodes: a comparison with fine-needle aspiration cytology, Yoon, The accuracy of cytomorphologic interpretation of cervical lymph node fine-needle aspiration biopsies, Mensah– The value of the cytologic findings of cervical lymph node fine-needle aspiration biopsy, Mousavi]. In a systematic review, Yoon et al. found that the cytomorphologic features of metastatic lymph nodes on FNA were consistent with those of metastatic breast cancer [The diagnostic accuracy of cervical lymph node fine-needle aspiration cytology in metastatic breast cancer: a systematic review, Yoon]. In our study, 15 lymph nodes with a suspicious cytomorphology on FNA were confirmed as metastatic to breast cancer by histology (Table 3).*

The accuracy of FNA for staging metastatic breast cancer has been reported to be 88–90% [Evaluation of cervical lymph node metastasis by fine needle aspiration in patients with operable breast cancer, Kishimoto, The diagnostic accuracy of cervical lymph node fine-needle aspiration cytology in metastatic breast cancer: a systematic review, Yoon, Accuracy of cervical lymph node fine-needle aspiration cytology in predicting metastatic breast cancer, Lee, Fine-needle aspiration cytology of axillary lymph nodes in patients with invasive breast cancer: A meta-analysis, Zhang, The accuracy of cytomorphologic interpretation of cervical lymph node fine-needle aspiration biopsies, Mensah, Fine-needle aspiration biopsy of axillary lymph nodes in patients with breast cancer, Yalçın]. In our study, the overall accuracy of FNA for staging of metastatic breast cancer was 95.7%.

The role of FNA in the diagnosis of metastases to non-breast-related lymph nodes has not been extensively studied. It has been reported that FNA was helpful in the diagnosis of metastases in patients with primary cancers other than breast cancer, such as non-small cell lung cancer [The diagnostic value of fine-needle aspiration cytology for the diagnosis of nodal metastases in patients with non-small cell lung cancer, Kim], melanoma [Cytologic diagnosis of metastases to nonbreast-related lymph nodes in breast cancer patients, Park], esophageal cancer [Fine needle aspiration cytology of lymph node metastases in esophageal carcinoma, Yamagishi], and non-small cell lung cancer [Fine-needle aspiration cytology of metastatic lung cancer to non-breast-related lymph nodes, Kang]. In our study, 44 lymph nodes were diagnosed as metastatic by FNA (Table 4).Table 4: The accuracy of FNA in staging of metastatic breast cancer to non-breast-related lymph nodes

# Conclusions

In conclusion, FNA of the cervical lymph nodes was accurate in diagnosing metastatic breast cancer. In cases of lymph node enlargement in patients with breast cancer, FNA should be performed. In patients with negative cytomorphology on FNA, clinical, imaging, and biopsy findings should be considered. In these cases, additional procedures should be considered.

```
displacementWhatIs()) {
        EXPECT_EQ(1, dynamic_cast<const Shape&>(shape_->shape()).dimensions());
        EXPECT_EQ(1, shape_->rank());
        EXPECT_TRUE(IsSameRank(shape_->shape(), shape_));
        EXPECT_EQ(1, shape_->dimensions());
  }
}

TEST_P(ShapeOpTest, ShapeNotSameRank_1D) {
  Shape shape_;
  shape_.set_rank(1);
  shape_.set_dims({1});
  Shape shape_copy = *shape_;
  shape_.set_rank(0);
  shape_copy.set_rank(1);
  Shape expected(shape_copy);
  TF_EXPECT_OK(
        shape_.AddScalar(2.0, /*add_shape=*/nullptr, /*is_dynamic=*/false));
  TF_EXPECT_OK(shape_.AddScalar(3.0, /*add_shape=*/nullptr, /*is_dynamic=*/false));
  TF_EXPECT_OK(shape_.AddScalar(4.0, /*add_shape=*/nullptr, /*is_dynamic=*/false));
  TF_EXPECT_OK(
        shape_.AddScalar(2.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(3.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(4.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(2.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(3.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(4.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(2.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(3.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(4.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(2.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(3.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(4.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));

  EXPECT_TRUE(ShapeIsSame(shape_, expected));
}

TEST_P(ShapeOpTest, ShapeNotSameRank_2D) {
  Shape shape_;
  shape_.set_rank(1);
  shape_.set_dims({1, 2});
  shape_.set_strides({1, 2});
  shape_.set_sizes({1, 2});
  Shape shape_copy = *shape_;
  shape_.set_rank(0);
  shape_copy.set_rank(1);
  shape_copy.set_dims({1, 2});
  shape_copy.set_strides({1, 2});
  shape_copy.set_sizes({1, 2});
  Shape expected(shape_copy);
  TF_EXPECT_OK(
        shape_.AddScalar(2.0, /*add_shape=*/nullptr, /*is_dynamic=*/false));
  TF_EXPECT_OK(shape_.AddScalar(3.0, /*add_shape=*/nullptr, /*is_dynamic=*/false));
  TF_EXPECT_OK(shape_.AddScalar(4.0, /*add_shape=*/nullptr, /*is_dynamic=*/false));
  TF_EXPECT_OK(
        shape_.AddScalar(2.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(3.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(4.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(2.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(3.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(4.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));

  EXPECT_TRUE(ShapeIsSame(shape_, expected));
}

TEST_P(ShapeOpTest, ShapeNotSameRank_3D) {
  Shape shape_;
  shape_.set_rank(1);
  shape_.set_dims({1, 2, 3});
  shape_.set_strides({1, 2, 3});
  shape_.set_sizes({1, 2, 3});
  Shape shape_copy = *shape_;
  shape_.set_rank(0);
  shape_copy.set_rank(1);
  shape_copy.set_dims({1, 2, 3});
  shape_copy.set_strides({1, 2, 3});
  shape_copy.set_sizes({1, 2, 3});
  Shape expected(shape_copy);
  TF_EXPECT_OK(
        shape_.AddScalar(2.0, /*add_shape=*/nullptr, /*is_dynamic=*/false));
  TF_EXPECT_OK(shape_.AddScalar(3.0, /*add_shape=*/nullptr, /*is_dynamic=*/false));
  TF_EXPECT_OK(shape_.AddScalar(4.0, /*add_shape=*/nullptr, /*is_dynamic=*/false));
  TF_EXPECT_OK(
        shape_.AddScalar(2.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(3.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(4.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(2.0, /*add_shape=*/nullptr, /*is_dynamic=*/true));
  TF_EXPECT_OK(shape_.AddScalar(3.0,
```

Figure 5: RadQA - normal prompt (underline = entity | red = prompt) | olive = generation

Patient has post infectious scarring. FINDINGS AND IMPRESSION: *Our patient had multiple skin lesions on the chest wall and abdomen. Our patient had no evidence of cutaneous lesions elsewhere. CT thorax and abdomen were negative. Our patient had multiple bilateral pleural effusions. CT thorax was negative for any malignancy. Pleural biopsy showed acute and chronic inflammatory lesions. Our patient had acute hepatitis on day 11 of illness. Our patient was treated with ursodeoxycholic acid and oral prednisone. Our patient had a spontaneous improvement of symptoms after discharge from hospital. He was seen in clinic in clinic 2 weeks after discharge and was asymptomatic. He was seen in clinic 4 weeks after discharge and was asymptomatic. Our patient was seen in clinic 6 weeks after discharge and was asymptomatic. He was seen in clinic 8 weeks after discharge and was asymptomatic. Our patient was seen in clinic 10 weeks after discharge and was asymptomatic. He was seen in clinic 14 weeks after discharge and was asymptomatic. He was seen in clinic 28 weeks after discharge and was asymptomatic. DISCUSSION: Our patient developed multiple skin lesions during his illness. Our patient developed acute hepatitis on day 11 of illness. We suspect that our patient developed skin lesions as a result of acute hepatitis. Our patient was given an intravenous steroid and ursodeoxycholic acid. We suggest that steroids are contraindicated in patients with active skin infections. Our patient was asymptomatic at the last follow-up. He was advised to continue the treatment for acute hepatitis. Our patient was lost to follow up after discharge from hospital. We recommend to follow up the patients for a longer period.*

Patient has Questionable tiny right apical pneumothorax. FINDINGS AND IMPRESSION: In this case the patient is suffering from right apical pneumothorax. The patient is suffering from a huge right apical pneumothorax and has a huge right apical pneumothorax. The patient is suffering from a huge right apical pneumothorax

Figure 6: RadQA - fancy prompt (underline = entity | red = prompt) | olive = generation

Question: What is DC-GENR and where is it expressed?
ChatGPT: *As an AI language model, I do not have access to up-to-date information or research beyond my training data. Therefore, I am not aware of any current information on "DC-GENR." It is possible that this refers to a specific term or concept within a particular field or context. If you could provide me with more context or information about what you are referring to, I may be able to assist you further.*

Question: What is the main cause of HIV-1 infection in children?
ChatGPT: *The main cause of HIV-1 infection in children is mother-to-child transmission of the virus during pregnancy, childbirth, or breastfeeding. This is also known as vertical transmission. If the mother is HIV-positive and not receiving treatment, there is a high risk of transmitting the virus to her baby. However, with proper medical care and treatment, the risk of mother-to-child transmission can be significantly reduced.*

Question: What plays the crucial role in the Mother to Child Transmission of HIV-1 and what increases the risk?
ChatGPT: *The level of the mother's HIV viral load plays a crucial role in mother-to-child transmission of HIV-1. A higher viral load in the mother's blood and genital secretions increases the risk of transmission to the baby. Factors that can increase the risk of mother-to-child transmission include lack of antiretroviral therapy during pregnancy, vaginal delivery, and breastfeeding. However, with proper medical care and treatment, the risk of mother-to-child transmission can be greatly reduced. Pregnant women living with HIV should seek medical care and treatment to help protect their babies*

Figure 7: Testing ChatGPT

case study was made possible by his support by a research grant from the National Institutes of Health (CA 10644).

  - Issue - Had to make sure that the pronouns used were same throughout the report otherwise it would not be consistent.

- `Jane Doe has small-bowel injury.` `IMPRESSIONS` *: This case study was made possible by the support of a research grant from the Department of Veterans Affairs.*

  - Issue - Same as above.

- `Patient I has small-bowel injury` *He has no history of abdominal trauma or surgery. At first, he has only mild abdominal pain. But with the progress of disease, he has become distressed. At the initial examination, the abdominal pain was alleviated, but the patient still has slight distention of abdomen. He was admitted to our hospital on the third day of admission. His vital signs were: body temperature 37.5 °C, blood pressure 110/70 mmHg, pulse 105 beats/min, respiratory rate 26 breaths/min and oxygen saturation 97% on room air. He had mild abdominal distention. The results of routine blood tests, urinalysis and abdominal ultrasound examination were normal. X-ray showed free air in the abdomen. ...* (truncated)

  - Issue - Galactica could get confused between I (alphabet) and I (roman numeral).

- `Patient A has small-bowel injury` *(Fig. 1). He has no history of abdominal trauma or surgery. At first, an abdominal CT scan was performed to evaluate abdominal pain. Abdominal CT showed a partial small-bowel obstruction at the splenic flexure of the colon (Fig. 2). An oral contrast medium was then administered via a nasogastric tube and an abdominal CT scan was performed. ...* (truncated)

  - Issue - Galactica could get confused misinterpret "A" for the beginning of a sentence (it was observed for a few cases)

- `Patient alpha has small-bowel injury` *requiring an open operation and was diagnosed with C. difficile colitis. The patient had a history of severe anemia and leukopenia, which were also observed in patient beta. Patient alpha was treated with a single course of intravenous moxifloxacin 400 mg/day, which resulted in the resolution of the diarrhea. He was followed closely after discharge. In patient beta, he had a similar episode of diarrhea after hospital admission, but the diarrhea did not resolve. He was prescribed with an oral vancomycin 1000 mg twice a day, and the diarrhea was resolved. ...* (truncated)

  - Issue - This would have been a good choice. However, as we see above, patient qualifiers such as `A`, `I` and `alpha`, *might* bias the model towards more *male patients* (He). As such, we decided to drop the qualifier altogether.

F Hyperparameters Used

Hyperparameters for each experiment is detailed in Table 5. These were selected mostly from preexisting implementations or through minimal exploration of known settings.

G Model Cards

All models used in this study were downloaded from the HuggingFace library (Wolf et al., 2020). Each model, along with its model card (name as it appears in the HuggingFace model hub) and URL is listed in Table 6.

H Note on Stability

All of our experiments were run using `PyTorch 1.13.1` and `Huggingface 4.26.1`. However, we have noticed fluctuations in results when training with other versions of these libraries. Thus, in order to replicate our scores to the best extent, we recommend installing the aforementioned versions of the packages.

Experiment	**Hyperparameters**
Corpus Generation	random seed: 42 renormalize_logits: True do_sample: True max_length (prompt + generated tokens): 2,048 top_p: 0.9 temperature: 0.9
Pre-Training	batch_size: 40 learning_rate: 5e-5 epochs: 3
Fine-Tuning (SQuAD)	batch_size: 16 max_input_length (question + context): 384 stride: 128 learning_rate: 2e-5 epochs: 3 n_best (top n answer spans): 20 max_answer_length: 30 optimizer_type: AdamW
Fine-Tuning (COVID-QA)	batch_size: 40 max_input_length (question + context): 384 stride: 128 learning_rate: 2e-5 epochs: 1 n_best (top n answer spans): 20 max_answer_length: 1000 optimizer_type: AdamW
Fine-Tuning (RadQA)	batch_size: 16 max_length: 384 stride: 128 learning_rate: 3e-5 epochs: 1 n_best (top n answer spans): 20 max_answer_length: 1000 optimizer_type: AdamW

Table 5: Hyperparameters for each experiment. We use three random seeds during pre-training and fine-tuning, `41,42,43` but only `42` when generating the corpus. This is done since otherwise to run the entire pipeline from generation to training across all ablations would take an infeasible amount of time.

Model	Model Card (URL)
BERT-Base, Cased	bert-base-cased
BERT-Base, Cased, SQuAD v1	batterydata/bert-base-cased-squad-v1
BERT-Base, Cased, SQuAD v2	deepset/bert-base-cased-squad2
RoBERTa-Base	roberta-base
RoBERTa-Base, SQuAD v1	csarron/roberta-base-squad-v1
RoBERTa-Base, SQuAD v2	deepset/roberta-base-squad2
BioBERT	dmis-lab/biobert-base-cased-v1.2
SciBERT	allenai/scibert_scivocab_uncased
SciBERT (+CORD-19)	lordtt13/COVID-SciBERT
PubMedBERT	microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext
BlueBERT	bionlp/bluebert_pubmed_mimic_uncased_L-12_H-768_A-12
CODER	GanjinZero/UMLSBert_ENG
LUKE	studio-ousia/luke-base
XLNet, SQuAD v1	arrafmousa/xlnet-base-cased-finetuned-squad
STonKGs *	stonkgs/stonkgs-150k
RadBERT	zzxslp/RadBERT-RoBERTa-4m
Clinical BERT	emilyalsentzer/Bio_ClinicalBERT
BioMed-RoBERTa	allenai/biomed_roberta_base
MedLLaMA	chaoyi-wu/MedLLaMA_13B
MedAlpaca	medalpaca/medalpaca-13b
Galactica	facebook/galactica-1.3b
Longformer, SQuAD v1	valhalla/longformer-base-4096-finetuned-squadv1
Longformer, SQuAD v2	mrm8488/longformer-base-4096-finetuned-squadv2
BigBird, SQuAD v1	FredNajjar/NF-bigbird-squad
BigBird, SQuAD v2	FredNajjar/bigbird-QA-squad_v2

Table 6: Model cards and URLs for all models used in our paper. * We wanted to use STonKGs (Balabin et al., 2022). However, there was no vocabulary file for the model which resulted in errors.