Investigating Methods of Using Context to Augment pre-trained Language Models for Question Answering

Stanford CS224N Custom Project

Sanjay Nagaraj
Department of Computer Science
Stanford University
sanjay25@stanford.edu

Josh Sanyal
Department of Computer Science
Stanford University
jsanyal@stanford.edu

Rohan Davidi
Department of Computer Science
Stanford University
rohand25@stanford.edu

Abstract

Given the limitations of pre-trained language models for open-domain question answering, research has shown improved performance using information retrieval of relevant context. While much work has focused on how to retrieve the most relevant information, we focus on how best to incorporate the retrieved contextual information by exploring three frameworks: 1) Input Augmentation – concatenating the question with the context, 2) Embedded Context Injection – injecting context embeddings between the model encoder and decoder, and 3) Decoder Token Upsampling – increasing the sampling probability of tokens in the context. We evaluate and compare these frameworks across various context types by assessing the accuracy of generated answers and logical consistency between generated answers and provided context. We find that input augmentation and decoder token upsampling outperform our baseline model without context. However, these methods deteriorate with longer contexts whereas embedded context injection is quite robust across context types, indicating potential for a model that combines these frameworks.

1 Key Information to include

- Mentor: Siyan Li
- External Collaborators (if you have any): N/A
- Sharing project: No

2 Introduction

Despite containing significant amounts of knowledge and performing quite well on sophisticated knowledge-based tasks, pre-trained language models can still confidently produce inconsistent and inaccurate answers (Jiang et al., 2021). Recently, this can be seen by Google's Bard model making a factual error about the James Webb Space Telescope in its first demo. Furthermore, given the abundance of existing documents and facts, it seems unlikely that pre-trained language models with a fixed number of weights can store and retrieve these facts (Guu et al., 2020).

Previous studies have focused on developing information retrieval methods to improve the performance of pre-trained language models on question answering (QA). Many of these retrieval methods

focus on retrieving and ranking the most relevant information but understanding the best way to utilize this information remains an important question.

In this work, we experiment with different approaches to incorporate contextual information to improve the accuracy of pre-trained language models for QA and ensure that generated answers are consistent with the given context. Our methods involve 1) incorporating retrieved context as input alongside the question, 2) performing embedded context injection between the model encoder and decoder, and 3) modifying the sampled probability distribution $P(\theta)$ to increase the likelihood of sampling tokens in the context. We evaluate these methods on three types of contexts (phrase, sentence, paragraph(s)) across two metrics, an F1 score to measure the accuracy of generated outputs compared to ground truth answers and a natural language inference (NLI) model-based evaluation of the logical consistency between answers and the provided context.

3 Related Work

Several studies have explored different methods of retrieving or generating contextual information to augment pre-trained language models for question answering. Petroni et al. (2020) compares context retrieved from an oracle (always contains related true information), an off-the-shelf information-retrieval system, and an autoregressive language model, finding that the context from the oracle and information-retrieval system help outperform the standard QA model. Guu et al. (2020) train a latent knowledge retriever to retrieve context from a corpus that outperforms other context retrieval methods by a significant margin. In both papers, the retrieved context is appended to the question before being passed as input to the QA model.

While the above research clearly demonstrates the potential of incorporating relevant context into QA models and investigates how best to retrieve that context, research on how best to use retrieved context remains comparatively under-investigated. While concatenating questions and context before passing input to the models is a popular approach, the max token length of models like BERT (512 tokens) limits the amount of context you can utilize (Devlin et al., 2018). Furthermore, while these papers show that retrieved context can help achieve better F1 scores, it is unclear how effectively context is being used in generating answers and if there is room for improvement in this regard.

As such, we focus on comparing various methods of incorporating context into pre-trained QA models across three different context types and two metrics. In addition to measuring F1 scores to measure the accuracy of our answers, we also leverage NLI models to measure the consistency between generated answers and provided context. This serves as a proxy to understand how well the model is leveraging the given context and was loosely inspired by Mitchell et al. (2022) which uses NLI to maximize the consistency of several generated answers on a batch of questions.

4 Approach

4.1 Baseline

Our baseline model is an off-the-shelf T5-small pre-trained on the training set of the Natural Questions (NQ) dataset to perform closed-book question answering (Raffel et al., 2020; Kwiatkowski et al., 2019). We evaluate the performance of this model using F1-score for comparison against our context incorporation methods. We will discuss these metrics more in-depth in the evaluation method section.

4.2 Experimental Frameworks

4.2.1 Input Augmentation

Our first method to incorporate contextual information is input augmentation, which fuses the question and the retrieved context together before passing it into the encoder of the T5-small baseline model. Mathematically, this can be represented as creating vector $x' \in \mathbb{R}^{m+n}$ by concatenating the tokenized context vector $c \in \mathbb{R}^n$ and the original tokenized input vector $x \in \mathbb{R}^m$. Figure 1 visually depicts this setup.

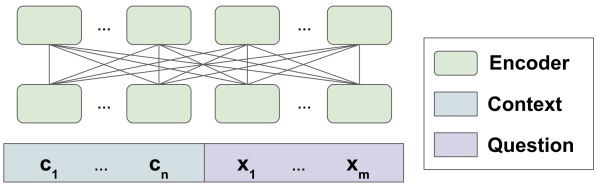

Figure 1: Input Augmentation Method

We evaluate this method using zero-shot learning, where this concatenated representation is used at test time with no fine-tuning. We also train and evaluated a fine-tuned model using these concatenated representations.

4.2.2 Embedded Context Injection

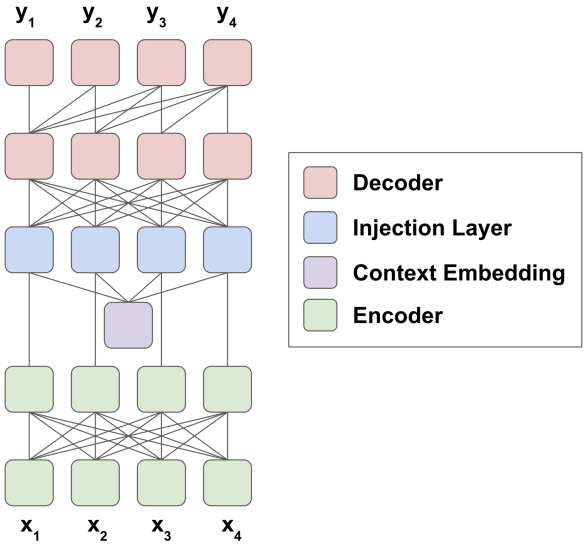

Figure 2: Simplified architecture of modified T5 model with context injection

To incorporate context at a deeper layer in model, we examine dense context injection between the encoder and decoder blocks. First, we vectorize our context using two different embeddings, word-averaged GLoVe embeddings and sentence-averaged SentenceBERT embeddings. For reference, we use off-the-shelf 300-dimension GloVe embeddings that were pre-trained on Wikipedia 2014 + Gigaword 5 and 768-dimension SentenceBERT embeddings from the multi-qa-mpnet-base-dot-v1 model, a finetuned Sentence Transformer model from HuggingFace that had the highest performance on semantic search (Reimers and Gurevych, 2019; Pennington et al., 2014).

Next, we pass our context embeddings along with our final hidden state from the T5-small encoder to our injection layer as seen in Figure 2. The injection layer consists of individual blocks which concatenate each vector in the final hidden state $x \in \mathbb{R}^d$ with the context embedding $c \in \mathbb{R}^n$ and feed this concatenated vector into an MLP neural network which outputs an updated hidden state vector $x' \in \mathbb{R}^d$ to feed into the decoder. We experiment with different numbers of layers for this neural network, ranging from 1 to 7 fully-connected layers with each layer, except the last, followed by a ReLU non-linearity. Additionally, each linear layer received an input of dimension $d + n * \frac{N-l+1}{N}$ and an output of dimension $d + n * \frac{N-l}{N}$, where l represents the linear layer number, and N represents the total number of layers in the given MLP architecture. Using the GLoVe embeddings, we test with 1, 3, and 5 layers in our injection block. Since the SentenceBERT embeddings are richer representations of the context, we tested with 3 and 7 layers.

4.2.3 Decoder Token Upsampling

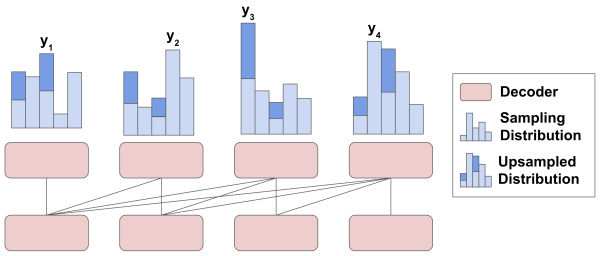

Figure 3: Simplified depiction of the token upsampling framework

In our final approach, we modify the decoding process of our model based on the given context as shown in Figure 3. Specifically, before the generation of each new token, we modify the baseline model decoder's sampling distribution by multiplying the probability of tokens by 10 if included in the generated context. Mathematically, at time step t, this can be represented as taking our original distribution $P(y_t|\{y_{<t}\})$ and creating a new upsampled distribution $P'(y_t|\{y_{<t}\})$ where $P'(y_t = w|\{y_{<t}\}) = P(y_t = w|\{y_{<t}\})$ for all $w \notin C$ and $P'(y_t = w|\{y_{<t}\}) = 10 \times P(y_t = w|\{y_{<t}\})$ for all $w \in C$ where C is the context and V is the entire vocabulary. $P'(y_t|\{y_{<t}\})$ is then scaled by a multiplicative factor to sum to 1. Finally, we sample our token from this upsampled token distribution and proceed to the next token, following the same process, until we are done. The intuition behind this method is that if our answer is included in the context, then increasing the probability that those tokens get sampled should help produce better answers.

5 Experiments

5.1 Data

We use the Natural Questions dataset which consists of real queries to the Google search engine (Kwiatkowski et al., 2019). For each question, an annotator is given a related Wikipedia page and annotates a long answer and a short answer to this query if present. For the open-domain question-answering setup, we use the annotated short answer as the answer to the query and set our retrieved context to either the short answer (short context), long answer (long context), or the sentence containing the short answer (encapsulated context). There are 307,373 training examples that we use to fine-tune our pre-trained T5 model and 7,830 development examples of which the first 5,000 examples are for testing and the remaining examples for validation. We emulate the preprocessing from Mitchell et al. (2022) by removing all questions without annotated answers such as those with only True/False outputs.

5.2 Evaluation method

Our quantitative evaluation criteria is two-fold. We first evaluate the F1 score of the model outputs compared to the annotated short answers, the standard metric for measuring QA accuracy. The F1 score is calculated as the harmonic mean of the precision and recall scaled by 100. Precision is calculated as the proportion of common tokens in the output and gold standard answer out of the output token count (precision) while recall is calculated as the proportion of common tokens in the output and gold standard answer as out of the answer token count (recall).

In addition, we use an NLI system, RoBERTa-Large NLI model, pre-trained on a combination of well-known NLI datasets: SNLI, MNLI, FEVER-NLI, ANLI (R1, R2, R3) to evaluate if our model outputs are logically aligned with the injected contextual information. We do this by computing entailment probability (ranging from 0 to 1) and reporting the NLI entailment score as this probability times 100 for each output-context pair (Liu et al., 2019). The F1 scores will be computed on all models while NLI alignment scores will only be computed for models with context use as it will be leveraged to determine how the stage and context type impacts alignment of output with the provided information. Furthermore, particular experimental variables, namely the number of layers

and embedding type in the injection models, will be investigated for impact on output-context NLI alignment.

Additionally, we perform a qualitative analysis by determining examples that have differing F1 and NLI scores across the experimental models. These examples are then investigated to determine the impact of provided context, the possibility of context information "shocking" and deteriorating the model, and any other qualitatively determinable effects across test data points.

5.3 Experimental details

All experiments were performed using the pretrained T5-small baseline model and evaluated on the test set across the three different context types (short, encapsulated, and long). We define short context as the short-form gold answer, encapsulated context as the sentence in which the short-form answer is contained, and long context as one or more paragraphs from the associated Wikipedia article containing the answer.

For the fine-tuned input augmentation model, we train for 5 epochs with all layers unfrozen. For each embedded context injection model, we freeze every layer except for the last two blocks of the encoder, the injection block, and the first two blocks of the decoder and fine-tune for 3 epochs. More generally, all fine-tuned models are trained with a batch size of 16, the AdamW optimizer, and perform standard gradient clipping at the end of each batch. We utilize standard cross-entropy loss and teacher forcing during fine-tuning. To set our hyper-parameters such as learning and set of unfrozen layers, we perform cross-validation on the validation set.

5.4 Results

In terms of F1 score accuracy, we see from Table 1 that for short and encapsulated contexts, input augmentation techniques (Finetuned, Zero Shot) and upsampling outperform the T5 Baseline. Particularly, on short contexts the finetuned input augmentation model outperforms the T5 Baseline by over 27 F1 points but falls short of the T5 Baseline on encapsulated contexts. The Zero Shot model, though, surpasses the T5 baseline on short and encapsulated contexts by over 8 points and the upsampling method exceeds the baseline by 6 points with short context performance and marginally surpasses it with encapsulated context.

We see amongst the experimental methods (non-baseline), for long contexts, the GloVe 3-layer injection model performs with the highest accuracy, but does not outperform the T5 baseline.

	Short		**Encapsulated**		**Long**	
Model	F1	NLI	F1	NLI	F1	NLI
T5 Baseline(Raffel et al., 2020)	24.17		24.17		**24.17**	
Zero Shot Input Augmentation	34.96	39.10	**32.19**	**44.50**	17.64	**43.49**
Finetuned Input Augmentation	**51.69**	**53.14**	19.94	32.39	6.74	28.98
GloVe 1-Layer Injection	19.70	21.62	3.89	13.11	1.42	16.90
GloVe 3-Layer Injection	8.82	17.02	22.28	27.68	22.36	32.97
GloVe 5-Layer Injection	21.82	22.68	22.15	27.56	22.22	32.40
BERT 3-Layer Injection	21.76	22.39	19.91	26.48	20.05	34.94
BERT 7-Layer Injection	20.09	22.00	20.03	26.24	19.74	30.75
Upsampled	30.81	30.94	24.19	31.23	12.54	34.94

Table 1: F1 scores and NLI Alignment scores across Input Augmentation, Injection, and Upsampling model techniques on short, encapsulated, and long contexts in comparison to the T5 Baseline model. Note, the T5 Baseline does not utilize contextual information and, therefore, does not have multiple F1 scores or any NLI alignment scores by context.

We also observe in Table 1, the increase of layers from 1 to 5 in the injection block of the GloVe injection model leads to signficant improvement in performance as the GloVe 5-Layer injection model produces higher F1 scores for all 3 contexts than the GloVe 1-Layer injection model. However, the BERT injection models display nearly the same accuracy performance despite being of 3-layers and 7-layers. Further, as seen in Figure 5, the best-performing BERT and GloVe models produce similar F1 scores across contexts despite the different embeddings used for injection.

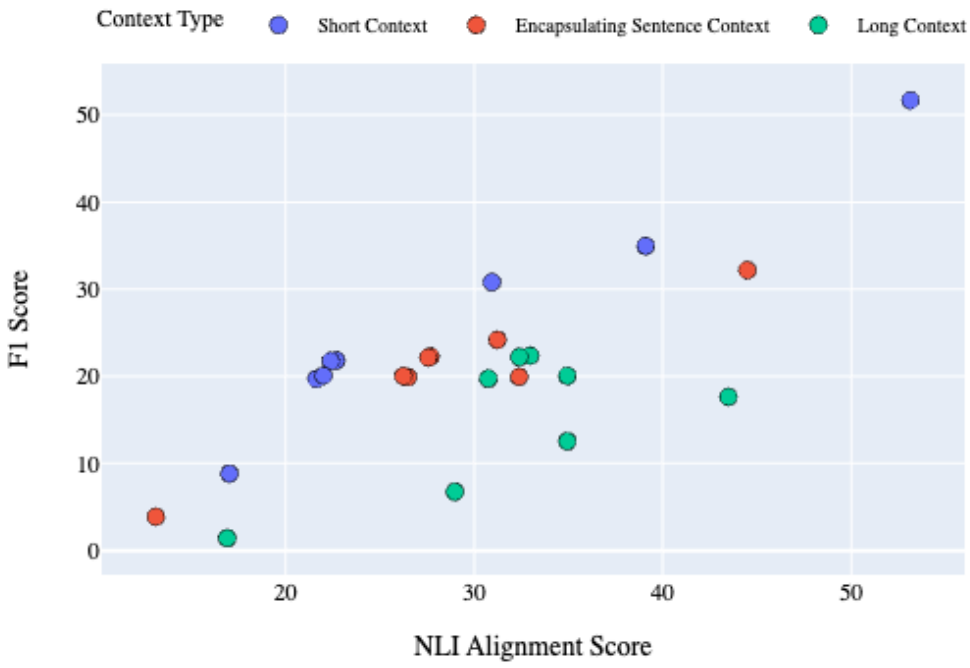

Figure 4: NLI Alignment score and F1 score scatter plot demonstrating correlation relationship.

From Figure 4, we also note a relatively strong correlation between NLI alignment score and F1 Score. This trend is consistent for short, encapsulated, and long contexts. This is corroborated in Table 1, as the highest F1 accuracy models for short and encapsulated contexts also produce the highest NLI alignment scores.

In Figure 5, we find, despite performing far better than all other experimental models, the finetuned Input Augmentation model's performance deteriorates significantly with longer contexts. Conversely, GloVe 3-layer, GloVe 5-layer, BERT 3-layer, and BERT 7-layer Injection models all produce more robustness for longer contexts. Particularly, the GloVe 5-layer and BERT injection models output consistent accuracy across all context sizes. The variability in performance are moderate in zero shot input augmentation and upsampling models, as seen below, as they show less volatility than the finetuned or GloVe 1-layer models but more than the larger injection models.

6 Analysis

6.1 Evaluation Breakdown

Our evaluations show that the incorporation of context can improve generated answers, as measured by F1 scores and an NLI evaluation system. The input augmentation and token upsampling framework had the best performance, while context injection shows promise if optimized further. The length of context also had a large impact on the performance of the input augmentation and upsampling methods, with models utilizing short and encapsulated context having much higher F1 scores than those using long-form context. This is because the input augmentation method can lead to the question embedding being drowned out by the long context, and the token upsampling method can upsample noisy words from the long-form context at key token indices.

Our findings also revealed a positive correlation between F1 and NLI entailment scores, highlighting that contextual incorporation methods improve answer self-consistency as long as they produce accurate outputs. Given this finding, we believe combining ideas from our context injection methodology and input augmentation framework could help improve accuracy and self-consistency.

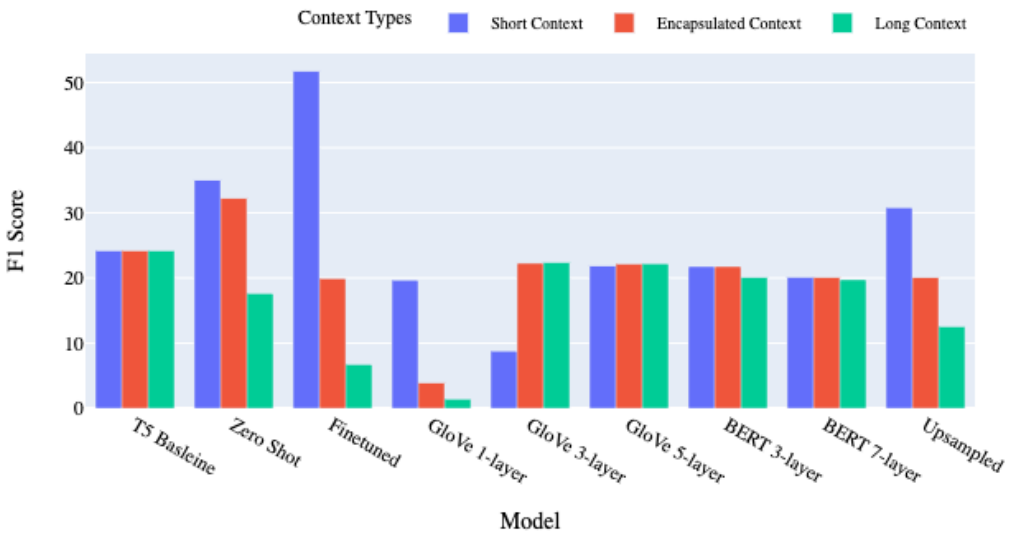

Figure 5: F1 score distribution across models and contexts.

Focusing on our injection technique, we found that using an injection block with more layers, particularly for GloVe, leads to improved consistency and performance. This is likely due to the fact that a single layer between the encoder and decoder of the model can "shock" the model with the sudden change introduced by concatenation, while a deeper neural network allows for more intricate fusing of the context embedding and the encoder output state. Although, BERT embeddings tended to produce greater robustness and performance for short-contexts than GloVe embeddings do, we surprisingly found the embedding type makes less of an impact when comparing the best-performing injection models: GloVe 5-layer and BERT 7-layer.

Another finding was that our injection framework was robust to context type. This can be seen when examining the F1 and NLI performance across the GLoVe and BERT injections with three or more layers. Thus, this method shows promise for incorporating longer contexts, which was shown to be an issue for the other methods.

6.2 Qualitative Analysis

Across our experiments, we identified four interesting classes of examples to analyze. Table 2 in the Appendix further describes these four common themes and provides concrete examples for each across all of our experimental approaches. These four classes included gold standard examples that were modified from incorrect to correct by the injection alongside 3 error classes that encompassed the majority of errors seen. The three error classes are graded from slightly incorrect to majorly incorrect to gibberish.

An example of a "general context understood, but incorrect answer" error is if given the question "Who is recognized as the founder of Islam?", the model outputs "Ali" (the successor to Muhammad) instead of Muhammad. Most errors, especially for the upsampling and augmentation methods, fell into this category. This makes sense given that the T5-small baseline model is pre-trained on a large corpus of text and has a good understanding of the question answering task, but a short piece of context may not be enough to correct model outputs.

Under the "shocked and lost answer" class we see examples like the following, where the question asks "Who is written in the book of life" and the generated answer is "Life". A smaller portion of the errors fell into this category, with the majority of these errors coming when long form context was injected with a shallow, linear injection block. This makes sense since the modified T5 is not able to effectively incorporate the new context which corrupts the embedding that is passed into the decoder.

Lastly, a very small number of errors fell into the "shocked and gibberish" category. Depending on the model setup, these errors either took the form of consistently producing a single character or

word/phrase as output. For example, for the fine-tuned input augmentation method, when prompted with the question "Who made the most free throws in NBA history", the generated output is "aside". It is hard to mitigate this class of errors with our input augmentation method given the long form context we are providing to it. However, we can address this issue in our injection framework by creating a more complex representation of our injection and can address this in the token upsampling representation by not multiplying the probabilities of filler tokens that may match the context. Overall, this analysis highlights areas in which context can improve answer generation and areas in which the answers still fall short.

7 Conclusion

This project investigates various methods of incorporating context to pretrained models for question answering. In doing so, we found input augmentation (Zero-Shot and Finetuned) and upsampling methods are able to outperform the baseline model. However, we find, in comparison to the baseline, these models deteriorate with context length. Relative to these input augmentation techniques, we find BERT Injection Models and GloVe Injection Models with 3 or more layers in their injection produce more robust and consistent performance with longer contexts. We also show that adding layers to the injection block generally improves performance. Therefore, there is potential for a model that uses these two in conjunction with a context-length threshold in order to produce a stable model that provides potential improvements in short-context performance.

With respect to NLI scores, we found there is a relatively strong positive correlation between NLI alignment scores and F1 scores. This suggests that techniques that produce outputs that better align with the contexts at shorter and longer context lengths could produce better performance. Thus, a model that takes in contexts from dataset, Wiki, Google, etc. could frame itself to use the baseline or an experimental model and continue to regenerate until an NLI alignment level is met to generate potentially high accuracy outputs.

As discussed, using combinations of these methods to create a best-performing ensemble model could be a fruitful future investigation. Due to limitations in compute units and resources, we were unable to test such combination ensembles. Further, under less time and compute restraints, we would aim to make use of other forms of context such as unrelated, adversarial, and generated contexts which could lead to additional insights.

Although these methods are applicable to various transformer models, the experiments conducted were limited to the T5 model. Additionally, contexts were derived from a curated dataset as opposed to in conjunction with a generalizable retrieval system, making it harder to generalize these results to contexts from real information retrieval systems.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*. JMLR.org.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How Can We Know When Language Models Know? On the Calibration of Language Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*.

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*.

Eric Mitchell, Joseph J. Noh, Siyan Li, William S. Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher D. Manning. 2022. Enhancing self-consistency and performance of pre-trained language models with nli. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

A Appendix (optional)

Figure 6: NLI Alignment score distribution by example for long contexts across models.

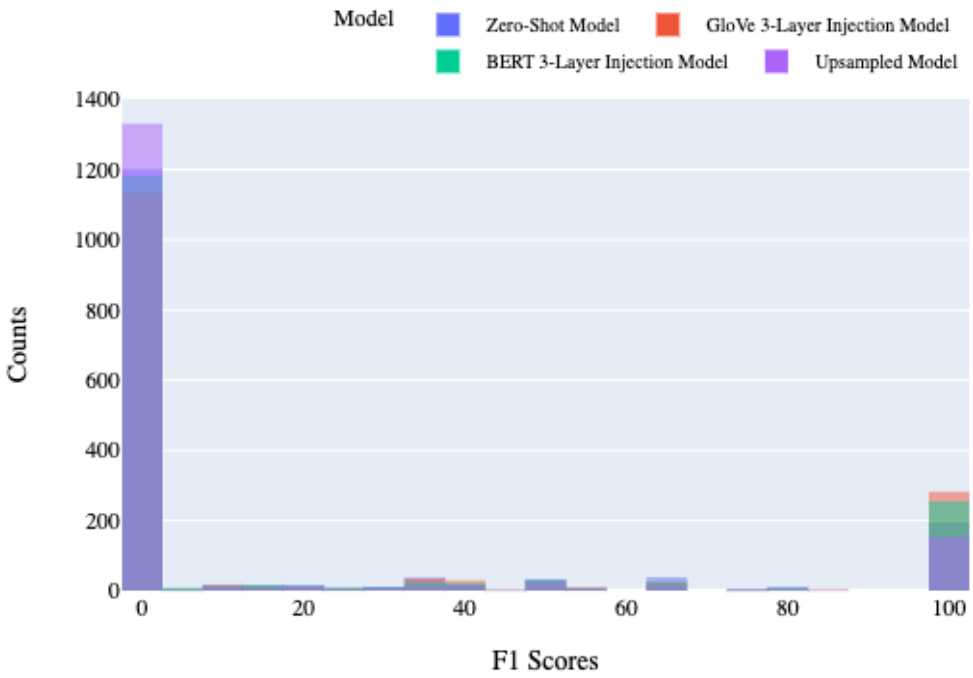

Figure 7: F1 score distribution by example for long contexts across models.

	General Context understood, but incorrect answer	**Modified into right answer from wrong answer**	**Shocked and lost answer**	**Shocked and gibberish**
Input augmentation (Zero Shot)	**Q:** Who is recognized as the founder of Islam? **Generated Answer**: Ali **Correct Answer**: Muhammad	**Q:** Who did the dominican republic gain independence from? **Generated Answer**: Haiti **Baseline Answer**: The French	**Q:** How many times has the Saints won the super bowl? **Generated Answer**: Super Bowl XLIV	**Q:** When did how you remind me come out? **Generated Answer**: "
Input augmentation (Fine-Tuned)	**Q:** What states do not allow daylight savings time? **Generated Answer**: Idaho **Correct Answer**: Arizona	**Q:** Where does the last name Waters come from? **Generated Answer**: Wales **Baseline Answer:** Old French	**Q:** What is the job of the whip in congress? **Generated Answer:** Forward	**Q:** Who made the most free throws in nba history? **Generated Answer:** aside
GLOVE Injection	**Q:** When is the last time philadelphia won the superbowl? **Generated Answer:** 2017 **Correct Answer:** Super Bowl LII	**Q:** Who performed the halftime show at super bowl 51? **Generated Answer:** Lady Gaga **Baseline Answer:** Justin Timberlake	**Q:** Who is written in the book of life? **Generated Answer:** Life	**Q:** What is the capital of georgia? **Generated Answer:** avlenie avetinovi aveti
BERT Injection	**Q:** Who won the 2017 Women's Wimbledon Final? **Generated Answer:** Roger Federer **Correct Answer:** Garbiñe Muguruza"	**Q:** When does the second half of vikings season 5 air? **Generated Answer:** 2018 **Baseline Answer:** December 9, 2017	**Q:** How much is a 72 oz steak at the big texan? **Generated Answer:** 1,389 sq ft	**Q:** Who has the most gold medals in the winter olympics? **Generated Answer:** Himself
Token Upsampling	**Q:** What is the current population of bora bora? **Generated Answer:** over 60,000 people **Correct Answer:** 10,605	**Q:** Which is the world's largest company in terms of revenue? **Generated Answer:** Walmart **Baseline Answer:** The United States Rubber Company	**Q:** Which is the ring finger for male in india? **Generated Answer:** ring	**Q:** What song is played while raising the american flag? **Generated Answer:** ' ' ' ' '

Table 2: Selected outputs from context incorporation methods when provided long form context