Margin Triggered Reranking for Extractive Question Answering on SQuAD v1.1

Agarwal Ishan, Agarwal Samriddh, Goel Arnav, Kakkar Nihirra, Sikka Arshin
CS4248 Group G02, School of Computing, National University of Singapore

Abstract

We study extractive question answering on SQuAD v1.1 using a RoBERTa base model and a light reranking layer. First, we fine-tune RoBERTa on SQuAD and obtain a strong baseline with 84.28 exact match and 90.93 F1 on the development set. We then analyze the distribution of the model's top span candidates and observe that the gold answer lies in the top five spans for about 95% of questions, which reveals clear headroom for better ranking. Based on this observation, we introduce a margin triggered reranking layer that operates on only the top two spans and uses a bi encoder to rescore candidates. Reranking is activated only when the score margin between the best two spans is small. Our final system improves the baseline to 84.40 exact match and 91.04 F1 while modifying only about 0.5% of predictions. This shows that simple local reranking can yield consistent gains over a strong encoder only baseline with very low computational overhead.

1 Introduction

Extractive question answering (QA) models take a question and a context passage and predict a contiguous answer span. Pre-trained transformer encoders such as BERT (Devlin et al.2019) and RoBERTa (Liu et al.2019) achieve strong performance on benchmarks like SQuAD (Rajpurkar et al.2016). However, these models typically produce a full distribution over candidate spans and only the top one is used at inference time.

In this project, we study how to exploit this rich candidate distribution. We focus on SQuAD v1.1 and fine-tune RoBERTa base as a strong baseline. We then ask a simple question. When the model is wrong, is the gold span often already present among its high scoring candidates, and if so, can a targeted reranking layer recover some of these errors

Our analysis confirms that the gold answer appears in the top five spans for about 95% of development questions, and an oracle that always picks the best span among the top five would achieve more than 95% exact match. This suggests that many errors are due to misordering of a small set of plausible spans rather than complete failure to generate the correct span.

To exploit this, we propose a margin triggered reranking layer. Instead of reranking every example, we only rerank examples where the score margin between the top two candidates is small, which we interpret as a sign of uncertainty. We use a sentence level bi encoder to compute semantic similarity between the question and each candidate span, and combine these scores with the baseline span scores through a simple interpolation. On SQuAD v1.1, this approach yields a modest but consistent improvement over our RoBERTa baseline while changing very few predictions.

2 Related Work

Extractive QA was popularized by SQuAD (Rajpurkar et al.2016), where models predict answer spans inside a given context. Modern systems typically fine-tune pre-trained transformers such as BERT (Devlin et al.2019) and RoBERTa (Liu et al.2019) with a span prediction head over start and end positions.

Reranking is widely used in retrieval and QA. Cross encoder rerankers process concatenated query and candidate text and are accurate but expensive (Nogueira and Cho2019), while bi encoder models such as Sentence BERT (Reimers and Gurevych2019) encode them separately and use cosine similarity for efficiency. Our work follows the bi encoder line and applies a very light span level reranker that is only triggered when the base model scores are ambiguous.

3 Dataset and Preprocessing

We use SQuAD v1.1 (Rajpurkar et al.2016), which contains 87,599 training questions and 10,570 development questions. Each example consists of a Wikipedia paragraph, a natural language question, and one or more annotated answer spans inside the paragraph.

We follow the standard preprocessing setup from the Hugging Face SQuAD example. Each question context pair is tokenized with the RoBERTa tokenizer. To handle long contexts, we use a maximum sequence length of 384 tokens and a document stride of 128 tokens. For each context, we create overlapping windows that contain the question followed by a slice of the context. Windows are padded to the maximum length.

For each window, we map the character level answer span to token indices. If the answer lies fully inside the current window, we set the start and end positions to the corresponding token indices. If the answer falls outside the window, we assign both start and end positions to the special classification token and do not expect this window to contain the gold span. This yields a supervised dataset of tokenized inputs with aligned start and end indices for training.

4 Baseline Model

4.1 Architecture

We use RoBERTa base (Liu et al.2019) as our baseline encoder. Given a tokenized sequence of length n, the encoder produces contextual representations $h_i \in \mathbb{R}^d$ for each token i. On top of these representations, we add a standard span prediction head as implemented in the Hugging Face transformers library.

The head consists of a linear layer that maps each h_i to two logits, one for the start index and one for the end index

$$\ell_i^{(s)} = w_s^\top h_i + b_s \quad (1)$$

$$\ell_i^{(e)} = w_e^\top h_i + b_e \quad (2)$$

where $w_s, w_e \in \mathbb{R}^d$ and $b_s, b_e \in \mathbb{R}$ are learnable parameters. During training, we apply cross entropy loss to the start logits and end logits and minimize the sum of these two losses.

4.2 Training Procedure

We fine-tune RoBERTa using the Hugging Face `Trainer` API. Unless otherwise noted, we use the hyperparameters in Table 1.

Setting	Value
Base model	`roberta base`
Optimizer	AdamW
Learning rate	3×10^{-5}
Weight decay	0.01
Batch size per device	8
Grad. accumulation	1, 2, or 4
Max sequence length	384
Document stride	128
Epochs	1 to 3

Table 1: Main hyperparameters for RoBERTa fine-tuning.

We first ran a small debug model on CPU on a subset of the data to validate preprocessing and the training loop. We then fine tuned on a GPU server and evaluated on the SQuAD development set using the official evaluation script. Our fine tuning code is adapted from the public Hugging Face SQuAD example, and we use the SQuAD evaluation script unchanged. All code for extracting top-k candidates, computing candidate statistics, and implementing margin triggered reranking is written by our group, and is available at `https://github.com/arshinsikka/CS4248_G02_QA`.

A three epoch run shows the training loss decreasing from about 1.80 to about 0.49, while the evaluation loss reaches its minimum around the first or second epoch and then slightly deteriorates. This behavior suggests that long training runs risk overfitting on SQuAD, which is consistent with prior work. Based on these observations, we focused our main experiments on one and two epochs with different gradient accumulation settings.

4.3 Baseline Results

Following the official SQuAD evaluation protocol, we report exact match (EM) and token level F1 on the development set.

We experimented with three main configurations.

- Small CPU run on a subset of the data. This model achieved 70.44 EM and 79.54 F1 and was used only as a sanity check.
- Full GPU run, 1 epoch, gradient accumulation 1. This model achieved 82.89 EM and 90.01 F1.

- Full GPU run, 2 epochs, gradient accumulation 2. This model achieved 84.28 EM and 90.93 F1.

We also trained a variant with 2 epochs and gradient accumulation 4, which obtained 83.95 EM and 90.68 F1. The best overall baseline uses 2 epochs and gradient accumulation 2.

5 Candidate Analysis

The baseline model produces a probability distribution over all possible start and end positions, which can be converted into a ranked list of span candidates. To gauge the potential benefit of reranking, we extracted the top k spans for each question with $k \in \{2, 3, 5\}$ and computed oracle scores.

For each k, we checked whether the gold span appears among the top k candidates and computed the best possible EM if we were allowed to pick the correct candidate whenever it is present. With $k = 5$, the gold span appears at ranks 1 to 5 in 8,908, 661, 253, 150, and 81 examples respectively, and is missing from the top five in 517 cases.

Thus the gold span is in the top five candidates for about 95.1% of questions. The oracle exact match improves from 84.28 for top one to 95.11 for top five. The corresponding oracle F1 increases from 90.93 to 97.08. These numbers indicate significant headroom for a reranker that can identify the correct span among a small set of candidates.

We also analyzed the score margins between the top candidates. Let c_1 and c_2 be the top two candidates for a question with scores s_1 and s_2. Define the margin $m_{1,2} = s_1 - s_2$. When the gold span is at rank 1, the median margin is about 0.60. When the gold span is at rank 2, the median margin drops to about 0.11. This suggests that small margins correlate with misranked questions where the correct span is present but not chosen.

6 Margin Triggered Reranking

6.1 Reranking Rule

Inspired by the margin analysis, we introduce a margin triggered reranking rule. For each question, let c_1 and c_2 be the top two spans according to the baseline scores, and let $m_{1,2}$ be their score margin. We choose a margin threshold τ and apply the following decision.

- If $m_{1,2} \geq \tau$, keep c_1 as the final prediction.
- If $m_{1,2} < \tau$, invoke a reranker to rescore c_1 and c_2 and pick the better one.

In other words, we only rerank questions where the baseline has low confidence in its top choice as measured by the margin between the top two scores.

6.2 Bi Encoder Reranker

We experimented with both cross encoder and bi encoder rerankers. Cross encoders process the concatenated question and candidate text through a transformer and then output a relevance score. In our experiments, cross encoders were more expensive and did not consistently outperform the baseline, even when combined with margin triggers.

Our final reranker uses a bi encoder based on the public sentence transformer model `all MiniLM L6 v2` (Reimers and Gurevych2019). The bi encoder encodes the question and each candidate span separately into embeddings e_q and e_{c_i}. We then compute a cosine similarity score

$$s_{\text{rerank}}(c_i) = \cos(e_q, e_{c_i}) \tag{3}$$

for each candidate c_i. These scores are combined with the baseline scores through a simple interpolation

$$s_{\text{final}}(c_i) = \alpha s_{\text{base}}(c_i) + (1 - \alpha) s_{\text{rerank}}(c_i) \tag{4}$$

where $\alpha \in [0, 1]$ controls the relative importance of the baseline and the reranker. When reranking is triggered, we select the candidate with the highest s_{final} among the top two spans.

We conducted a grid search over $\alpha \in \{0.3, 0.4, 0.5, 0.6\}$ and margin thresholds $\tau \in \{0.05, 0.10, 0.15, 0.20, 0.25\}$. We also experimented with top three and top five settings, where the reranker considers more candidates, but these settings consistently underperformed top two.

6.3 Reranking Results

Our best configuration uses top two spans, $\alpha = 0.5$, and margin threshold between 0.05 and 0.25. All thresholds in this range produced the same dev set scores. The results are summarized in Table 2.

The margin triggered reranker improves exact match by 0.12 points and F1 by 0.11 points over the RoBERTa baseline. Although the absolute gain is small, the reranker is extremely conservative. Out of 10,570 development questions,

System	EM	F1	Changed
Baseline RoBERTa	84.28	90.93	0
Rerank (top2, $\alpha = 0.5$)	84.40	91.04	55

Table 2: Development results on SQuAD v1.1. "Changed" counts how many predictions differ from the baseline.

only 55 predictions are changed. Among these changes, 14 examples become correct after reranking, 1 example becomes incorrect, and the remaining examples are incorrect both before and after reranking.

For comparison, global reranking without a margin trigger, especially with low α that relies heavily on the reranker scores, consistently decreases both exact match and F1. Reranking over top three or top five candidates also hurts performance, presumably because it brings in more noisy spans. Overall, we find that a light margin triggered reranker on top two spans is the most effective and robust setting.

7 Novelty and Contributions

Our work combines several simple components into a focused study of span reranking for extractive QA.

- We provide a detailed analysis of the baseline span distribution on SQuAD v1.1, including top k coverage statistics and score margin distributions between candidates.
- We propose margin triggered reranking that uses the baseline model's own score margin to decide when to rerank instead of reranking every example.
- We show that a lightweight bi encoder reranker, combined with the baseline scores through interpolation, can recover errors in ambiguous cases while modifying fewer than 1% of predictions.
- We demonstrate that reranking only the top two candidates is more robust than reranking larger candidate sets, which tend to introduce noisy spans.

Taken together, these contributions show that careful use of a reranking layer on top of a strong encoder only model can provide small but reliable gains with minimal additional complexity.

8 Discussion and Limitations

Our candidate analysis in Section 5 shows that the gold span is already present in the top five candidates for about 95% of questions, and that an oracle that always picks the best span among the top five would reach 95.1 EM and 97.1 F1. However, our margin triggered reranker only changes 55 out of 10,570 predictions, so most of this headroom remains unused.

The 517 questions where the gold answer is not in the top five candidates constitute a hard set that our reranker cannot fix, since it never generates new spans. Some of these errors are likely due to context truncation or fundamental model errors rather than ranking. Finally, the small absolute gain of our reranker suggests that more powerful approaches might need to operate at the passage selection or span generation level in order to make larger improvements.

9 Conclusion and Future Work

We studied extractive question answering on SQuAD v1.1 with a strong RoBERTa baseline and a simple margin triggered reranking layer. The gold span is usually in the top five candidates, and misrankings often occur when the score margin between the top candidates is small. A bi encoder reranker applied only to the top two spans in low margin cases yields small but consistent improvements in EM and F1 while changing fewer than 1% of predictions. This suggests that conservative local reranking can reliably fix a small subset of ambiguous errors at almost no additional cost, although larger gains will likely require improving the underlying span extractor or passage selection.

There are several directions for future work. First, instead of choosing the margin threshold and interpolation weight by grid search, they could be learned directly on held out data, for example through a calibration loss. Second, our current reranker only looks at the top two spans. A more expressive approach would train a small classifier that takes pairs of candidate spans from the top five list and predicts which span is more likely to be correct, then use these pairwise preferences to build a better ranking over all five candidates. Finally, future work can extend our approach to more challenging settings, such as SQuAD v2.0 with unanswerable questions or multi paragraph contexts, where both passage selection and span selection need to be improved.

References

[Devlin et al.2019] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

[Liu et al.2019] 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. In *arXiv preprint arXiv:1907.11692*.

[Rajpurkar et al.2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000 plus questions for machine comprehension of text. In *Proceedings of EMNLP*.

[Reimers and Gurevych2019] Nils Reimers and Iryna Gurevych. 2019. Sentence BERT: Sentence embeddings using Siamese BERT networks. In *Proceedings of EMNLP IJCNLP*.

[Nogueira and Cho2019] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re ranking with BERT. In *arXiv preprint arXiv:1901.04085*.