Span-Level Contrastive Learning for Near-Miss Errors to improve Extractive QA

Yuliana Jasso
ydj89@eid.utexas.edu

Abstract

This paper contributes two modest but effective modifications to a standard extractive QA baseline. First, we add a span-level contrastive objective that is trained alongside the usual SQuAD span prediction. Second, we introduce a hard negative mining procedure that gathers competing named-entity spans from the same sentence and applies a span-level contrastive loss that pushes them below the gold span. Together, these changes directly target the common “near-miss" mistakes, where the model finds the right region in the passage but selects another span of the same semantic type (two dates, two people, two locations). We show that, on ELECTRA-small, these additions reduce near-miss errors and improve SQuAD dev from 45.56 EM / 62.62 F1 to 53.55 EM / 70.95 F1.

1 Introduction

Most modern extractive QA systems are built on powerful pretrained encoders and span prediction heads. These models are generally very good at locating the region of a passage that contains the answer. However, they still struggle when that region contains more than one plausible span. Our baseline model is a typical example of this behavior. It often attends to the correct sentence, yet selects a competing span of the same semantic type; for instance, choosing the wrong team, date, or person name when several appear in the same sentence.

This pattern resembles a local form of hallucination. Ji et al. (2023) note that large language models can generate fluent but ungrounded outputs that are not actually supported by the evidence they have seen. We observe an analogous phenomenon in extractive QA. The model has found the right place to look at, but still chooses a neighboring , type-consistent span instead of the gold answer. Table 1 shows typical near-miss examples from SQuAD in a passage about Super Bowl 50, the baseline

Passage: The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title.
Question: Which NFL team represented the NFC at Super Bowl 50? **Gold answer:** Carolina Panthers **Answer from baseline:** Denver Broncos **Answer from baseline+spanCL:** Carolina Panthers
Question: What is the AFC short for? **Gold answer:** American Football Conference **Answer from baseline:** Denver Broncos **Answer from baseline+spanCL:** American Football Conference

Table 1: Near-miss examples where span-level contrastive training corrects the baseline’s prediction.

predicts "Denver Broncos" instead of the correct "Carolina Panthers" when asked which team represented the NFC, and reuses "Denver Broncos" again when asked what "AFC" stands for. In both cases the model clearly focuses on the right part of the passage but settles on the wrong span.

A key reason for these errors lies in the standard SQuAD training objective. During fine-tuning, the model is trained with cross-entropy loss on a single gold start–end pair and receives no explicit supervision on other high-scoring spans it proposes. As a result, distractor spans that are semantically similar to the gold answer can maintain high scores. Prior work on adversarial and unanswerable QA has documented this limitation: models trained only with a gold span often remain confident in carefully constructed distractor sentences or unanswerable questions because those spans are never directly labeled as negatives (Jia and Liang (2017); Rajpurkar et al. (2018)).

In this paper, we investigate a simple span-level contrastive learning objective designed to address the “right sentence, wrong span” mistakes. Our

baseline is an ELECTRA extractive QA model on SQuAD, and we automatically mine "near-miss" spans from the same sentence as the gold answer by selecting other named-entity spans that share its coarse type, such as another team or date in the same sentence. A span-level contrastive loss then encourages the model to score the gold span above these near-miss competitors, explicitly penalizing cases where a plausible but incorrect span would otherwise outrank the true answer. Our approach is inspired by span-based contrastive methods proposed for machine reading comprehension (Ji et al., 2022), but we adapt them to the fully answerable SQuAD setting and to automatically mined near-miss spans instead of generated question variants.

Another limitation of the baseline is that its loss is purely token-level. It predicts start and end positions independently and then selects the span with the highest summed score. This setup offers very little pressure to separate confusing spans that live in the same sentence and look equally answer-like. By adding a span-level contrastive objective on top of the usual QA loss, we treat these confusing spans as hard negatives and train the encoder to pull the gold span closer while pushing these nearly-correct spans apart. Empirically, we show that this span-level contrastive objective yields modest gains in overall EM/F1 on the SQuAD dev set, with larger improvements on a curated near-miss subset, and qualitatively makes the model better at distinguishing between closely related spans within the same sentence. The rest of this paper introduces our proposal (Section 2), describes the contrastive extension and hard-negative mining procedure (Section 3), reports quantitative results (Section 4), and finishes with a conclusion (Section 5).

1.1 Related Work

Adversarial and unanswerable QA work has highlighted how fragile extractive QA models can be in the presence of plausible distractors. Jia and Liang (2017) construct adversarial sentences that preserve the gold answer but confuse models, causing large drops in performance. SQuAD extends the original SQuAD dataset with human-written unanswerable questions whose passages contain misleading answer-like spans, forcing models to learn when no answer is actually supported (Rajpurkar et al., 2018). Our setting remains closer to SQuAD, where every question is answerable, but we focus on cases where the passage contains several plausible spans in the same sentence and the main challenge is ranking the correct span above these confusable alternatives.

Several works try to sharpen answer boundaries or better rank competing spans. Yuan et al. (2020) propose a multilingual MRC model with auxiliary tasks that explicitly enhance answer boundary detection, improving the model's ability to locate precise start and end positions for the answer span. Wang et al. (2018) tackle multi-passage MRC with a cross-passage answer verification module, where candidate answers from different passages verify one another so that incorrect spans can be down-weighted. There is also growing interest in multi-span question answering, where models must produce multiple disjoint spans per question rather than a single span (e.g., (Li et al., 2022); (Moon et al., 2023)), further emphasizing the difficulty of reasoning over multiple answer-like spans in the same context. Our work addresses a simpler single-span setting, but targets a similar underlying issue: distinguishing the intended answer span from other plausible spans that co-occur in the same region.

More recently, contrastive learning has been explored as a tool for improving robustness in machine reading comprehension. Ji et al. (2022) introduce a span-level contrastive learning framework (spanCL) that explicitly contrasts an original answerable question with a paraphrased positive counterpart and a distorted negative (unanswerable) counterpart, encouraging the model to recognize subtle semantic differences that flip answerability. Feng et al. (2023) propose a contrastive learning framework in the context representation space that pushes answer sentences away from misleading sentences to improve robustness to adversarial perturbations. Other work applies contrastive objectives to reasoning paths or cross-lingual MRC, again using contrastive signals to separate supportive from misleading evidence. Our method is most closely related to spanCL, but instead of generating new question variants or contrasting sentences, we mine near-miss spans from NER-labeled entities in the same sentence and use their model scores in a span-level contrastive loss during fine-tuning.

2 Proposal

We propose a simple extension of a standard extractive QA baseline (ELECTRA-small fine-tuned on SQuAD) that directly targets the "right sentence, wrong span" mistakes observed in our analysis. Our approach adds span-level supervision on top

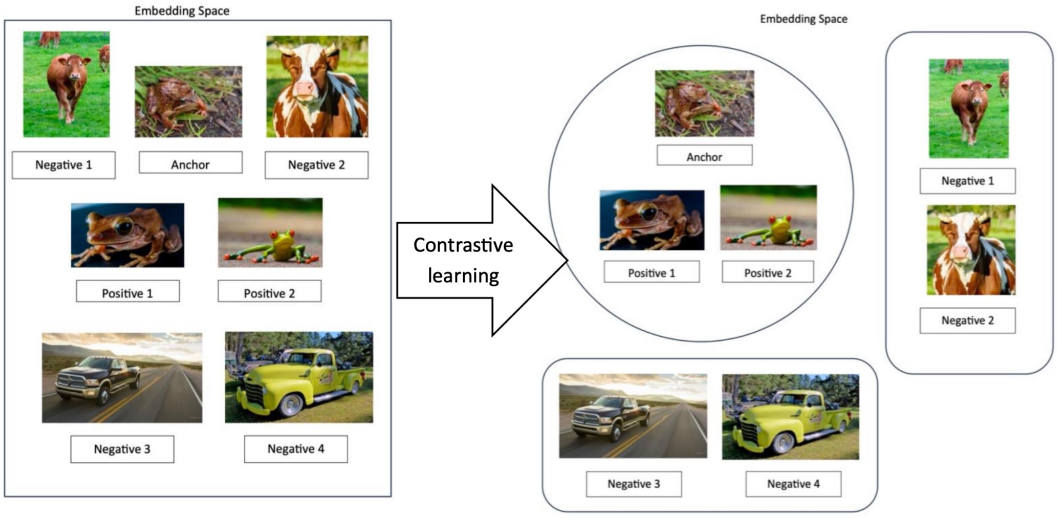

Figure 1: Illustration of contrastive learning. Given an anchor example (center), the model is trained to pull semantically related positives closer in the embedding space while pushing unrelated negatives farther apart. This separation makes it easier to disambiguate among candidates that initially occupy a similar region of the space.

of the usual token-level start/end objective and is built around three components. First, we introduce a span-level contrastive loss in addition to the standard cross-entropy loss. For each example, the model is trained not only to assign high probability to the gold start–end pair, but also to push apart high-scoring, nearly-correct spans and the gold span, explicitly penalizing the model for preferring near-miss competitors. Second, we make hard-negative mining sentence- and NER-aware. We preferentially select negatives that (i) come from the same sentence as the gold answer and (ii) share its entity or answer type (e.g., another team name or date), since our error analysis shows that these are exactly the spans the baseline most often confuses with the true answer. Third, we evaluate not only on the full SQuAD dev set but also on a targeted "near-miss" slice consisting of examples with multiple answer-like spans in a single sentence. This hard slice allows us to verify that any observed gains come from better span disambiguation in challenging contexts, rather than from improvements on easier, single-span cases.

3 Implementation Details

In this section, we describe the dataset, model architecture, negative mining procedure, and training setup used in our experiments. Figure 2 illustrates the span-level contrastive objective.

3.1 Dataset

We use the Stanford Question Answering Dataset (SQuAD) introduced by Rajpurkar et al. (2016). The dataset contains 87,599 training and 10,570 development question-answer pairs over Wikipedia paragraphs, where each example is annotated with an extractive answer span inside the context paragraph. We use the full training split for fine-tuning and report results on the official development set.

We load SQuAD via the Hugging Face `datasets` library and tokenize question–context pairs with the `google/electra-small-discriminator` tokenizer. We truncate inputs to a maximum sequence length of 384 tokens and pad to this length. For each tokenized example we store (i) the character-level `offset_mapping`, (ii) an integer `example_id` pointing back to the original raw example, and (iii) a binary `is_context_mask` indicating which tokens belong to the context.

Gold start and end token positions are computed from the offset mappings. For all experiments, we run on a single device (`cuda` if available, otherwise CPU).

3.2 Model

Our baseline model is `ElectraForQuestionAnswering` (`google/electra-small-discriminator`) from the Hugging Face Transformers library. Given a tokenized question–context pair, the encoder produces contextualized token representations, and a linear span prediction head outputs start and end logits for each token. At inference time, we select the span with the highest sum of start and end logits, subject to the usual constraint that the end index is not before the start index.

3.3 Negative Mining

To support span-level contrastive learning, we precompute sentence and named-entity information on the training contexts using spaCy (`en_core_web_sm`). For each training example we build an index that stores (i) the character offsets of each sentence in the context and (ii) all detected named entities with their labels and character spans.

During contrastive training, hard negatives are mined on-the-fly using this index. For a given example, we first recover the gold answer span in character space and locate the sentence that contains it. We then search for other named entities in the same sentence that share the same coarse type as the gold answer (e.g., another team name or date). These candidate entities are mapped back into token coordinates using the offset mappings. We keep up to four such spans per example as hard negatives for the contrastive objective. If no suitable negatives are found, the example contributes only to the standard QA loss.

3.4 Training Setup

We use PyTorch and the AdamW optimizer for all experiments, with a batch size of 16 for both baseline and contrastive training. The baseline ELECTRA model is fine-tuned on the full tokenized SQuAD training set using a learning rate of 3×10^{-5}. We run up to 400 gradient updates (with shuffled mini-batches), which is sufficient to reach a reasonable baseline performance while keeping training time modest.

Model	EM	F1	Correct	Total
ELECTRA baseline	45.56	62.62	4,816	10,570
+ span-level contrastive	53.55	70.95	5,661	10,570

Table 2: SQuAD dev-set performance of the ELECTRA baseline and our span-level contrastive model. EM and F1 are reported as percentages; *Correct* and *Total* denote the number of correctly answered questions and the total number of questions.

For the contrastive model, we start from the fine-tuned baseline weights and continue training with an additional span-level contrastive loss. We do not train on all examples: instead, we construct a hard training subset by filtering training instances whose context length exceeds 220 characters, since longer contexts are more likely to contain multiple answer-like entities. This subset is wrapped in a custom `ContrastiveDataset` that returns both tokenized features and the corresponding raw examples, and is fed through a `DataLoader` with the same batch size (16) and shuffling.

In each contrastive training step, we compute the standard QA loss $\mathcal{L}_{\text{QA}}$ from the ELECTRA QA head and a span-level contrastive loss $\mathcal{L}_{\text{CL}}$. For each gold span with at least one mined hard negative, we define the gold score as the sum of its start and end logits, and likewise define a score for each negative span. We then apply a hinge-style loss with margin $m = 0.25$:

$$\ell_{\text{pair}} = \max\bigl(0,\ m - (s_{\text{gold}} - s_{\text{neg}})\bigr),$$

and average this over all negative pairs in the batch to obtain $\mathcal{L}_{\text{CL}}$. The total loss is

$$\mathcal{L} = \mathcal{L}_{\text{QA}} + \alpha\, \mathcal{L}_{\text{CL}},$$

with contrastive weight $\alpha = 0.1$. The contrastive model is trained with AdamW, a slightly higher learning rate of 5×10^{-5}, and one pass over the hard subset.

For evaluation on the development set, we use the standard exact match (EM) metric and also analyze accuracy on a "hard" slice. A dev example is marked as hard if the baseline's prediction lies in the same sentence as the gold span and shares its coarse type, but differs in text, corresponding to the "right sentence, wrong span" behavior we target.

4 Results

We first compare ELECTRA fine-tuned on SQuAD with and without the proposed span-level con-

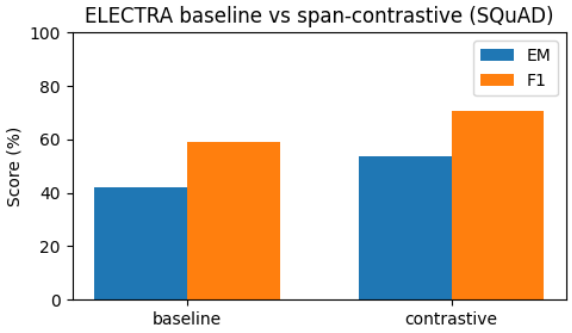

Figure 2: SQuAD dev-set EM and F1 scores for the ELECTRA baseline and the span-level contrastive model (higher is better).

trastive objective. The baseline model reaches 45.56 EM and 62.62 F1 on the SQuAD dev set (4,816 exact matches out of 10,570 examples. When we add span-level contrastive supervision, performance increases to 53.55 EM and 70.95 F1 (5,661 exact matches out of 10,570 examples). This is a large improvement fior a change that only modifies the training objective and keeps the backbone fixed to ELECTRA.

The goal of our method was not just to raise average scores, but to fix "near-miss" errors where the model has already found the right part of the passage but selects the wrong span inside it. To focus on these cases, we also evaluate on the hard near-miss slice defined in Section 3.4, where the baseline's prediction lies in the same sentence as the gold span, shares its coarse type, but differs in text. On this subset the contrastive model yields higher EM and F1 than the baseline, supporting the hypothesis that the model was already localizing the answer region and that making span representations more discriminative helps it choose the intended span.

5 Conclusion

Our experiments show that adding a simple span-level contrastive objective to ELECTRA fine-tuned on SQuAD leads to clear gains over a standard baseline. The contrastive model improves EM and F1 on the SQuAD dev set and, more importantly, reduces "near-miss" errors where the model finds the right sentence but selects the wrong neighboring span. This suggests that the baseline model was already localizing the correct region, and that making span representations more discriminative helps it choose the intended answer span more reliably. Overall, our results indicate that small changes to the training objective can meaningfully improve span-level accuracy in extractive QA.

References

Jianzhou Feng, Jiawei Sun, Di Shao, and Jinman Cui. 2023. Improving the robustness of machine reading comprehension via contrastive learning. *Applied Intelligence*, 53(5):9103–9114.

Yunjie Ji, Liangyu Chen, Chenxiao Dou, Baochang Ma, and Xiangang Li. 2022. To answer or not to answer? improving machine reading comprehension model with span-based contrastive learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1292–1300, Seattle, United States. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Ho Shu Chan, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):248:1–248:38.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Haonan Li, Martin Tomko, Maria Vasardani, and Timothy Baldwin. 2022. MultiSpanQA: A dataset for multi-span question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1250–1260, Seattle, United States. Association for Computational Linguistics.

Sungrim Moon, Huan He, Heling Jia, Hongfang Liu, and Jungwei Wilfred Fan. 2023. Extractive clinical question-answering with multianswer and multifocus questions: Data set development and evaluation study. *JMIR AI*, 2:e41818.

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.

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

Yizhong Wang, Kai Liu, Jing Liu, Wei He, Yajuan Lyu, Hua Wu, Sujian Li, and Haifeng Wang. 2018. Multi-passage machine reading comprehension with cross-passage answer verification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1918–1927, Melbourne, Australia. Association for Computational Linguistics.

Fei Yuan, Linjun Shou, Xuanyu Bai, Ming Gong, Yaobo Liang, Nan Duan, Yan Fu, and Daxin Jiang. 2020. Enhancing answer boundary detection for multilingual machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 925–934, Online. Association for Computational Linguistics.