# Question Answering Infused Pre-training of General-Purpose Contextualized Representations

Anonymous ACL submission

## Abstract

We propose a pre-training objective based on question answering (QA) for learning general-purpose contextual representations, motivated by the intuition that the representation of a phrase in a passage should encode all questions that the phrase can answer in context. To this end, we train a bi-encoder QA model, which independently encodes passages and questions, to match the predictions of a more accurate cross-encoder model on 80 million synthesized QA pairs. By encoding QA-relevant information, the bi-encoder's token-level representations are useful for non-QA downstream tasks without extensive (or in some cases, any) fine-tuning. We show large improvements over both RoBERTa-large and previous state-of-the-art results on zero-shot and few-shot paraphrase detection on four datasets, few-shot named entity recognition on two datasets, and zero-shot sentiment analysis on three datasets.

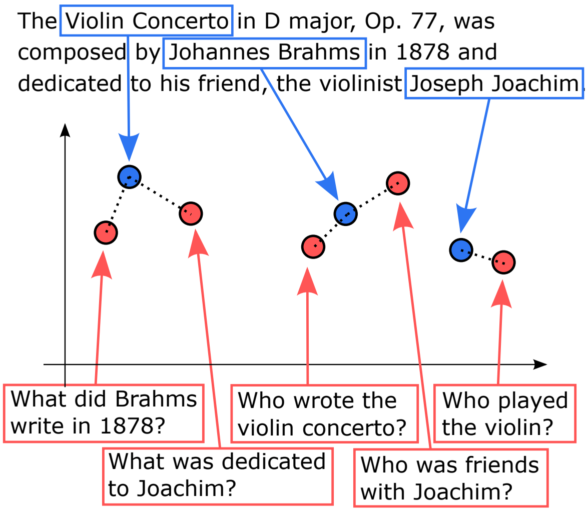
The Violin Concerto in D major, Op. 77, was composed by Johannes Brahms in 1878 and dedicated to his friend, the violinist Joseph Joachim.
What did Brahms write in 1878?
What was dedicated to Joachim?
Who wrote the violin concerto?
Who was friends with Joachim?
Who played the violin?

Figure 1: An overview of Question Answering Infused Pre-training. Our model independently creates vector representations (middle) for phrases in a passage (top) and for synthesized questions (bottom). Our objective encourages the vector for each phrase to have high similarity with the vectors for all questions it answers.

## 1 Introduction

Although masked language models build contextualized word representations, they are pre-trained with losses that minimize distance to *uncontextualized* word embeddings (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019). In this paper, we introduce Question Answering Infused Pre-training (QuIP), a new pre-training loss based on question answering (QA) that depends much more directly on context, and learns improved token-level representations for a range of zero- and few-shot tasks.

Our intuition for QuIP is that the contextualized representation for a phrase in a passage should contain enough information to identify all the questions that the phrase could answer in context. For example, in Figure 1, the representation for *Johannes Brahms* should be similar to the representation of all questions it can answer, such as "*Who wrote the violin concerto?*" We anticipate that optimizing passage representations for QA should benefit many downstream tasks, as question-answer pairs have been used as broad-coverage meaning representations (He et al., 2015; Michael et al., 2018), and a wide range of NLP tasks can be cast as QA problems (Levy et al., 2017; McCann et al., 2018; Gardner et al., 2019), For instance, our learned representations should encode whether a phrase answers a question like "*Why was the movie considered good?*", which corresponds to identifying rationales for sentiment analysis.

We train QuIP with a bi-encoder extractive QA objective. The bi-encoder model independently encodes passages and questions such that the representation of each phrase in a passage is similar to the representation of reading comprehension questions answered by that phrase. To train this model, we use a question generation model to synthesize 80 million QA examples, then train the bi-encoder to match the predictions of a cross-encoder QA model, which processes the passage and question together, on these examples. Bi-encoder QA has been used before for efficient open-domain QA via phrase retrieval (Seo et al., 2018, 2019; Lee et al., 2020, 2021), but its lower accuracy compared to cross-encoder QA has previously been viewed as a drawback. We instead view the relative weakness of bi-encoder QA as an opportunity to improve contextual representations via knowledge distillation, as self-training can be effective when the student model must solve a harder problem than the teacher (Xie et al., 2020). In particular, since the bi-encoder does not know the question when encoding the passage, it must produce a single passage representation that simultaneously encodes the answers to *all* possible questions. In contrast, while cross-encoder QA models are more accurate, they depend on a specific question when encoding a passage; thus, they are less suited to downstream use cases that require contextualized representations of passages in isolation.

We show that QUIP token-level representations are useful in a variety of zero-shot and few-shot learning settings, both because the representations directly encode useful contextual information, and because we can often reduce downstream tasks to QA. For few-shot paraphrase detection, QUIP with BERTScore-based features (Zhang et al., 2020) outperforms prior work by 9 F1 points across four datasets. For few-shot named entity recognition (NER), QUIP combined with an initialization scheme that uses question embeddings improves over RoBERTa-large by 14 F1 across two datasets. Finally, for zero-shot sentiment analysis, QUIP with question prompts improves over RoBERTa-large with MLM-style prompts by 5 accuracy points across three datasets, and extracts interpretable rationales as a side effect. Through ablations, we show that using real questions, a strong teacher model, and the bi-encoder architecture are all crucial to the success of QUIP. We will release code to reproduce all results upon publication.

## 2 QA Infused Pre-training

QA Infused Pre-training (QUIP) involves pre-training contextual representations with a bi-encoder extractive QA objective. In this section, we introduce some notation (§2.1), then describe our QUIP pipeline, which consists of three steps: question generation (§2.2), cross-encoder teacher re-labeling (§2.3), and bi-encoder training (§2.4).

### 2.1 Notation

All models operate on sequences of tokens $x = [x_1, \dots, x_L]$ of length $L$, where $x_1$ is the special beginning-of-sequence token. We learn an encoder $r$ that maps inputs $x$ to outputs $r(x) = [r(x)_1, \dots, r(x)_L]$ where each $r(x)_i \in \mathbb{R}^d$ for some fixed dimension $d$. We call $r(x)_i$ the contextual representation of the $i$-th token in $x$.

In extractive question answering, a model is given a context passage $c$ and question $q$, and must output a span of $c$ that answers the question. Typically, models independently predict probability distributions $p(a_{\text{start}} \mid c, q)$ and $p(a_{\text{end}} \mid c, q)$ over the answer start index $a_{\text{start}}$ and end index $a_{\text{end}}$.

### 2.2 Question Generation

**Question generation model.** We train a BART-large model (Lewis et al., 2020) to generate question-answer pairs given context passages. The model receives the passage as context and must generate the answer text, then a special separator token, then the question; this approach is simpler than prior approaches that use separate models for answer and question generation (Lewis and Fan, 2019; Alberti et al., 2019; Puri et al., 2020), and works well in practice.

**Training data.** We train on data from the MRQA 2019 Shared Task (Fisch et al., 2019), which includes six datasets: HotpotQA (Yang et al., 2018), NaturalQuestions (Kwiatkowski et al., 2019), NewsQA (Trischler et al., 2017), SearchQA (Dunn et al., 2017), SQuAD (Rajpurkar et al., 2016), and TriviaQA (Joshi et al., 2017). These datasets cover many of the text sources commonly used for pre-training (Liu et al., 2019; Lewis et al., 2020), namely Wikipedia (HotpotQA, NaturalQuestions, SQuAD), News articles (NewsQA), and general web text (SearchQA, TriviaQA).

**Generating questions.** We run our question generation model over a large set of passages to generate a large dataset of question-answer pairs. We decode using nucleus sampling (Holtzman et al., 2020) with $p = 0.6$, which was chosen by manual inspection to balance diversity with quality of generated questions. We do not filter questions in any way. While we observed some flaws related to question quality (questions were not always well-formed) and diversity (for some passages, the same or very similar questions were asked multiple times), this approach nonetheless yielded good downstream results. Attempts to mitigate these issues, such as using a two-stage beam search to ensure that questions for the same passage have different answers, did not noticeably change our downstream results (see §4.8). We obtain passages from the same training corpus as RoBERTa (Liu et al., 2019), which uses four sub-domains: BOOKCORPUS plus Wikipedia, CC-NEWS, OPENWEBTEXT, and STORIES. For each domain, we sample 2 million passages and generate 10 questions per passage, for a total of 80 million questions.[1]

### 2.3 Teacher Re-labeling

The answers generated by our BART model are not always accurate, nor are they always spans in the context passage. To improve the training signal, we relabel examples with a teacher model, as is common in knowledge distillation (Hinton et al., 2015). We use a standard cross-encoder RoBERTa-large model trained on the MRQA training data as our teacher. The model takes in the concatenation of the context passage $c$ and question $q$ and predicts $a_{\text{start}}$ and $a_{\text{end}}$ with two independent 2-layer multi-layer perceptron (MLP) heads. Let $T_{\text{start}}(c, q)$ and $T_{\text{end}}(c, q)$ denote the teacher's predicted probability distribution over $a_{\text{start}}$ and $a_{\text{end}}$, respectively.

### 2.4 Bi-encoder Training

Finally, we train a bi-encoder model to match the cross-encoder predictions on the generated questions. This objective encourages the contextual representation for a token to have high similarity (in inner product space) with the representation of every question that is answered by that token.

**Model.** The bi-encoder model with parameters $\theta$ consists of of three components: an encoder $r$ and two question embedding heads $h_{\text{start}}$ and $h_{\text{end}}$ that map $\mathbb{R}^d \to \mathbb{R}^d$. These heads will only be applied to beginning-of-sequence (i.e., `CLS`) representations; as shorthand, define $f_{\text{start}}(x) = h_{\text{start}}(r(x)_1)$ and likewise for $f_{\text{end}}$. Given a context passage $c$ and question $q$, the model predicts

$$p_\theta(a_{\text{start}} = i \mid c, q) \propto e^{r(c)_i^\top f_{\text{start}}(q)} \quad (1)$$

$$p_\theta(a_{\text{end}} = i \mid c, q) \propto e^{r(c)_i^\top f_{\text{end}}(q)} \quad (2)$$

In other words, the model independently encodes the passage and question with $r$, applies the start and end heads to the `CLS` token embedding for $q$, then predicts the answer start (end) index with a softmax over the dot product between the passage representation at that index and the output of the start (end) head. We initialize $r$ to be the pre-trained RoBERTa-large model (Liu et al., 2019), which uses $d = 1024$. $h_{\text{start}}$ and $h_{\text{end}}$ are randomly-initialized 2-layer MLPs with hidden dimension 1024, matching the default initialization of classification heads in RoBERTa.

**Training.** For an input consisting of context $c$ of length $L$ and question $q$, we train $\theta$ to minimize the KL-divergence between the student and teacher predictions, which is equivalent to the objective

$$-\sum_{i=1}^{L} T_{\text{start}}(c, q)_i \log p_\theta(a_{\text{start}} = i \mid c, q) + T_{\text{end}}(c, q)_i \log p_\theta(a_{\text{end}} = i \mid c, q) \quad (3)$$

up to constants that do not depend on $\theta$. We train for two epochs on the 80 million generated questions, which takes roughly 56 hours on 8 V100 GPUs, or roughly 19 GPU-days.[2] For details about training, see Appendix A.1.

## 3 Downstream Tasks

We evaluate QUIP on zero-shot paraphrase ranking, few-shot paraphrase classification, few-shot NER, and zero-shot sentiment analysis. We leverage the improved token-level representations afforded by QUIP, and in many cases also directly use QUIP's question-answering abilities.

### 3.1 Paraphrase Ranking

We first evaluate QUIP token-level representations by measuring their usefulness for zero-shot paraphrase ranking. In this task, systems must rank sentence pairs that are paraphrases above pairs that are non-paraphrases, without any task-specific training data. We compute similarity scores using the $F_{\text{BERT}}$ variant of BERTScore (Zhang et al., 2020), which measures cosine similarities between the representation of each token in one sentence and its most similar token in the other sentence. Given sentences $x_1$ and $x_2$ of lengths $L_1$ and $L_2$, define

$$B(x_1, x_2) = \frac{1}{L_1} \sum_{i=1}^{L_1} \max_{1 \le j \le L_2} \frac{r(x_1)_i^\top r(x_2)_j}{\|r(x_1)_i\| \|r(x_2)_j\|}.$$

[1] We estimate that using the entire corpus with these settings would generate around 900 million questions. We leave investigation of further scaling to future work.

[2] For comparison, pre-training RoBERTa-large from scratch took roughly 5000 GPU-days (Liu et al., 2019)

The $F_{\text{BERT}}$ BERTScore is defined as the harmonic mean of $B(x_1, x_2)$ and $B(x_2, x_1)$. Zhang et al. (2020) showed that BERTScore with RoBERTa is useful for both natural language generation evaluation and paraphrase ranking. Since BERTScore uses token-level representations, we hypothesize that it should pair well with QUIP. As in Zhang et al. (2020), we use representations from the layer of the network that maximizes Pearson correlation between BERTScore and human judgments on the WMT16 metrics shared task (Bojar et al., 2016).

### 3.2 Paraphrase Classification

We next use either frozen or fine-tuned QUIP representations for few-shot paraphrase classification, rather than ranking. Through these experiments, we can compare QUIP with existing work on few-shot paraphrase classification.

**Frozen model.** We train a logistic regression model that uses BERTScore with frozen representations as features. For a given pair of sentences, we extract eight features, corresponding to BERTScore computed with the final eight layers (i.e., layers 17-24) of the network. These layers encompass the optimal layers for both RoBERTa-large and QUIP (see §4.4). Freezing the encoder is often useful in practice, particularly for large models, as the same model can be reused for many tasks (Brown et al., 2020; Du et al., 2020).

**Fine-tuning.** For fine-tuning, we use the same computation graph and logistic loss function, but now backpropagate through the parameters of our encoder. For details, see Appendix A.3.

### 3.3 Named Entity Recognition

We also use QUIP for few-shot[3] named entity recognition, which we frame as a BIO tagging task. Since questions in QA often ask for entities of a specific type, we expect QUIP representations to contain rich entity type information. We add a linear layer that takes in token-level representations and predicts the tag for each token, and backpropagate log loss through the entire network. By default, the output layer is initialized randomly.

As a refinement, we propose using question prompts to initialize this model. The output layer is parameterized by a $T \times d$ matrix $M$, where $T$ is the number of distinct BIO tags. The log-probability of predicting the $j$-th tag for token $i$ is proportional to the dot product between the representation for token $i$ and the $j$-th row of $M$; this resembles how the bi-encoder predicts answers. Thus, we initialize each row of $M$ with the start head embedding of a question related to that row's corresponding entity tag. For instance, we initialize the parameters for the `B-location` and `I-location` tags with the embedding for "*What is a location ?*" We normalize the question embeddings to have unit L2 norm. This style of initialization is uniquely enabled by our bi-encoder QA model; it would be unclear how to use a language model or a cross-encoder QA model similarly.

### 3.4 Zero-shot Sentiment Analysis

Finally, we use QUIP for zero-shot binary sentiment analysis. We reduce sentiment analysis to QA by writing a pair of questions that ask for a *reason* why an item is good or bad (e.g., "*Why is this movie [good/bad]?*"). We predict the label whose corresponding question has higher similarity with the QUIP representation of some token in the input. This prompting strategy has the additional benefit of extracting rationales, namely the span that the QUIP model predicts as the answer to the question. While we focus on sentiment analysis, extractive rationales have been used for a wide range of NLP tasks (DeYoung et al., 2020), suggesting that this method could be applied more broadly.

More formally, let $x$ be an input sentence and $(q_0, q_1)$ be a pair of questions (i.e., a prompt). For label $y \in \{0, 1\}$, we compute a score for $y$ as

$$S(x, y) = \max_i r(x)_i^\top f_{\text{start}}(q_y) + \max_i r(x)_i^\top f_{\text{end}}(q_y). \quad (4)$$

This formula is a straightforward way to measure the extent to which *some* span in $x$ looks like the answer to the question $q_y$, based on the model's pre-trained ability to perform QA. We predict whichever $y$ has the higher value of $S(x, y) - C_y$, where $C_y$ is a calibration constant that offsets the model's bias towards answering $q_0$ or $q_1$. Our inclusion of $C_y$ is inspired by Zhao et al. (2021), who recommend calibrating zero-shot and few-shot models with a baseline derived from content-free inputs to account for biases towards a particular label. To choose $C_y$, we obtain a list $W$ of the ten most frequent English words, all of which convey no sentiment, and define $C_y$ as the mean over

[3] Yang and Katiyar (2020) study few-shot NER assuming data from other NER datasets is available; we assume no such data is available, matching Huang et al. (2020).

$w \in W$ of $S(w, y)$, i.e., the score when using $w$ as the input sentence (see Appendix A.5).

## 4 Experiments

### 4.1 Experimental details

**Datasets.** For paraphrasing, we use four datasets: QQP (Iyer et al., 2017), MRPC (Dolan and Brockett, 2005), PAWS-Wiki, and PAWS-QQP (Zhang et al., 2019). The PAWS datasets were designed to be challenging for bag-of-words models, and thus test whether our representations are truly contextual or mostly lexical. For QQP and MRPC, we use the few-shot splits from Gao et al. (2021) that include 16 examples per class; for the PAWS datasets, we create new few-shot splits in the same manner. We report results on the development sets of QQP and MRPC (as test labels were not available), the test set of PAWS-Wiki, and the "dev-and-test" set of PAWS-QQP. For NER, we use two datasets: CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) and WNUT-17 (Derczynski et al., 2017). We use the few-shot splits from Huang et al. (2020) that include 5 examples per entity type. All few-shot experiments report an average over five random splits and seeds, following both Gao et al. (2021) and Huang et al. (2020). For sentiment analysis, we use two movie review datasets, SST-2 (Socher et al., 2013) and Movie Reviews (MR; Pang and Lee, 2005), as well as the Customer Reviews (CR) dataset (Hu and Liu, 2004). We evaluate on the SST-2 development set and the MR and CR test sets made by Gao et al. (2021).

**Hyperparameter and prompt selection.** Due to the nature of zero-shot and few-shot experiments, we minimize the extent to which we tune hyperparameters, relying on existing defaults and previously published hyperparameters. For few-shot paraphrase classification, NER, and sentiment analysis, we developed our final method only using QQP, CoNLL, and SST-2, respectively, and directly applied it to the other datasets with no further tuning. We did measure zero-shot paraphrase ranking accuracy on all datasets during development of QUIP. For more details, see Appendix A.4.

For NER, we used the first question prompts we wrote for both CoNLL and WNUT, which all follow the same format, "*Who/What is a/an [entity type] ?*" (see Appendix A.8 for all prompts). For sentiment analysis, we wrote six prompts (shown in Appendix A.10) and report mean accuracy over these prompts, to avoid pitfalls associated with prompt tuning (Perez et al., 2021). We use the same prompts for SST-2 and MR; for CR, the only change we make is replacing occurrences of the word "*movie*" with "*product*" to reflect the change in domain between these datasets.

### 4.2 Baselines and Ablations

To confirm the importance of all three stages of our pre-training pipeline, we compare with a number of baselines and ablations.

**No question generation.** We train the bi-encoder model directly on the MRQA training data ("Bi-encoder + MRQA"). We also include the cross-encoder teacher model trained on MRQA as a baseline ("Cross-encoder + MRQA"). These settings mirror standard intermediate task training (Phang et al., 2018; Pruksachatkun et al., 2020).

**No teacher.** We train the bi-encoder using the answer generated by the question generation model ("QUIP, no teacher"). If the generated answer is not a span in the passage, we consider the question unanswerable and treat the span containing the `CLS` token as the answer, as in Devlin et al. (2019).

**Cross-encoder self-training.** To test whether the bottleneck imposed by the bi-encoder architecture is crucial for QUIP, we also train a cross-encoder model on our generated data ("QUIP, cross-encoder student"). Since this student model has the same architecture as the teacher model, we train it to match the teacher's argmax predictions, a standard self-training objective (Lee, 2013). We train for a comparable number of GPU-hours as QUIP (see Appendix A.2 for details).

**Unsupervised QA.** We test whether QUIP requires real QA data, or if a rough approximation suffices. We thus train a bi-encoder on 80 million pseudo-questions generated by applying noise to sentences ("Bi-encoder + UnsupervisedQA"), as in Lewis et al. (2019).

### 4.3 Bi-encoder Question Answering

While not our main focus, we first check that QUIP improves bi-encoder QA accuracy, as shown in Table 1. QUIP improves over Lee et al. (2021) by 5.4 F1 on the SQuAD development set. It also surpasses the reported human accuracy of 91.2 F1 on the SQuAD test set, as well as the best cross-encoder BERT-large single model from Devlin et al. (2019). QUIP greatly improves over baselines that

| Model | EM | F1 |
|---|---|---|
| Lee et al. (2021) | 78.3 | 86.3 |
| Bi-encoder + UnsupervisedQA | 17.4 | 24.9 |
| Bi-encoder + MRQA | 70.7 | 79.4 |
| QuIP, no teacher | 75.3 | 84.7 |
| QuIP | **85.2** | **91.7** |
| BERT-large cross-encoder | 84.2 | 91.1 |
| Cross-encoder + MRQA | 88.8 | 94.7 |
| QuIP, cross-encoder student | **89.5** | **94.8** |

Table 1: EM and F1 scores on the SQuAD development set for bi-encoder (top) and cross-encoder (bottom) models. QuIP outperforms the other bi-encoder model baselines, and even a cross-encoder BERT-large model. The RoBERTa cross-encoder models are better at QA, but will underperform QuIP on non-QA tasks.

directly train on MRQA data or do not use the teacher model. As expected, our cross-encoder models are more accurate at QA; we will show that their representations are less useful for non-QA tasks than the QuIP bi-encoder. Appendix A.6 shows results on all MRQA development datasets.

### 4.4 Zero-shot Paraphrase Ranking

We validate our approach and study the effects of various ablations on zero-shot paraphrase ranking. The first half of Table 2 shows WMT development set Pearson correlations averaged across six to-English datasets, as in Zhang et al. (2020), along with the best layer for each model. QuIP reaches its optimal score at a later layer (20) than RoBERTa-large (17), which may suggest that the QuIP training objective is more closely aligned with learning better representations than MLM.

The rest of Table 2 shows zero-shot paraphrase ranking results using BERTScore. QuIP improves substantially over RoBERTa on all four datasets, with an average improvement of .076 AUROC. The improvement is greatest on the PAWS datasets; since these datasets cannot be solved by lexical features alone, QuIP representations must be much more contextualized than RoBERTa representations. Training on Unsupervised QA data degrades performance compared to RoBERTa, showing that QuIP does not merely make word representations encode local context in a simple way. Training the bi-encoder directly on the MRQA dataset or without the teacher improves on average over RoBERTa, but QuIP greatly outperforms both baselines. The cross-encoder models also lag behind QuIP at paraphrase ranking, despite their higher QA accuracy. Thus, we conclude that having real questions, accurate answer supervision, and a bi-encoder student model are all crucial to the success of QuIP.

### 4.5 Paraphrase Classification

Table 3 shows few-shot paraphrase classification results. As we studied ablations in the previous section, we focus on the comparison between QuIP and non-QuIP baselines. First, we use RoBERTa-large embeddings in place of QuIP in our method. Second, we compare with LM-BFF (Gao et al., 2021), which pairs RoBERTa-large with MLM-style prompts. We use LM-BFF with manually written prompts and demonstrations, which was their best method on QQP by 2.1 F1 and was 0.3 F1 worse than their best method on MRPC. QuIP used as a frozen encoder is competitive with LM-BFF on QQP and outperforms it by 6.1 F1 on MRPC, 11.2 F1 on PAWS-Wiki, and 12.1 F1 on PAWS-QQP. Fine-tuning QuIP gives additional improvements on three of the four datasets, and outperforms fine-tuning RoBERTa by an average of 6.9 F1.

### 4.6 Named Entity Recognition

Table 4 shows few-shot NER results on the CoNLL and WNUT datasets. QuIP improves over RoBERTa-large by 11 F1 on CoNLL and 2.9 F1 on WNUT when used with a randomly initialized output layer. We see a further improvement of 4 F1 on CoNLL and 7.4 F1 on WNUT when using question embeddings to initialize the output layer. Using the cross-encoder trained directly on QA data is roughly as good as QuIP when using randomly initialized output layers, but it is incompatible with question embedding initialization.

### 4.7 Sentiment Analysis

Table 5 shows zero-shot accuracy on our three sentiment analysis datasets. We compare with zero-shot results for LM-BFF (Gao et al., 2021)[4] and reported zero-shot results from Zhao et al. (2021) using GPT-3 with Contextual Calibration (CC) on SST-2. QuIP using an average prompt outperforms zero-shot LM-BFF by 5.4 points, averaged across the three datasets. Choosing the best prompt on SST-2 and using that for all datasets improves results not only on SST-2 but also MR, and maintains average accuracy on CR. Using the cross-encoder student QA model with the same prompts leads to worse performance: we hypothesize that the bi-encoder bottleneck encourages QuIP to make each

[4] We tried applying our calibration strategy to LM-BFF as well, but found that it did not improve accuracy.

| Model | WMT $r$ | WMT Best Layer | QQP | MRPC | PAWS-Wiki | PAWS-QQP |
|---|---|---|---|---|---|---|
| RoBERTa-large | .739 | 17 | .763 | .831 | .698 | .690 |
| Cross-encoder + MRQA | .744 | 16 | .767 | .840 | .742 | .731 |
| QuIP, cross-encoder student | .753 | 16 | .769 | .847 | .751 | .706 |
| Bi-encoder + UnsupervisedQA | .654 | 11 | .747 | .801 | .649 | .580 |
| Bi-encoder + MRQA | .749 | 15 | .771 | .807 | .747 | .725 |
| QuIP, no teacher | .726 | 19 | .767 | .831 | .780 | .709 |
| QuIP | **.764** | 20 | **.809** | **.849** | **.830** | **.796** |

Table 2: Pearson correlation on WMT development data, best layer chosen based on WMT results, and AUROC on zero-shot paraphrase ranking using BERTScore. QuIP outperforms all baselines on all datasets.

| Model | Fine-tuned? | QQP | MRPC | PAWS-Wiki | PAWS-QQP |
|---|---|---|---|---|---|
| LM-BFF (reported) | Fine-tuned | $69.8_{0.8}$ | $77.8_{0.9}$ | - | - |
| LM-BFF (rerun) | Fine-tuned | $67.1_{0.9}$ | $76.5_{1.5}$ | $60.7_{0.7}$ | $50.1_{2.8}$ |
| RoBERTa-large | Frozen | $64.4_{0.4}$ | $80.6_{0.7}$ | $62.3_{0.9}$ | $50.6_{0.4}$ |
| QuIP | Frozen | $68.9_{0.2}$ | $82.6_{0.4}$ | $71.9_{0.5}$ | $\mathbf{63.0_{1.2}}$ |
| RoBERTa-large | Fine-tuned | $64.9_{0.7}$ | $84.4_{0.3}$ | $65.7_{0.3}$ | $50.9_{0.8}$ |
| QuIP | Fine-tuned | $\mathbf{71.0_{0.3}}$ | $\mathbf{86.6_{0.4}}$ | $\mathbf{75.1_{0.2}}$ | $60.9_{1.0}$ |

Table 3: F1 scores on few-shot paraphrase classification, averaged across five training splits (standard errors in subscripts). QuIP outperforms prior work (LM-BFF; Gao et al., 2021) as well as our own RoBERTa baselines.

| Model | CoNLL | WNUT |
|---|---|---|
| Huang et al. (2020) | 65.4 | 37.6 |
| **Standard init.** | | |
| RoBERTa-large | $59.0_{2.4}$ | $39.3_{0.6}$ |
| Cross-encoder + MRQA | $68.9_{3.3}$ | $43.0_{0.9}$ |
| QuIP, cross-encoder student | $63.4_{3.3}$ | $39.4_{1.7}$ |
| Bi-encoder + UnsupervisedQA | $58.2_{2.6}$ | $26.0_{1.0}$ |
| Bi-encoder + MRQA | $66.4_{3.3}$ | $42.2_{0.4}$ |
| QuIP, no teacher | $67.7_{1.9}$ | $40.7_{1.4}$ |
| QuIP | $70.0_{2.4}$ | $42.2_{0.5}$ |
| **Question prompt init.** | | |
| Bi-encoder + UnsupervisedQA | $62.7_{3.3}$ | $30.4_{0.8}$ |
| Bi-encoder + MRQA | $72.0_{2.8}$ | $44.0_{1.3}$ |
| QuIP, no teacher | $71.4_{3.0}$ | $47.8_{1.1}$ |
| QuIP | $\mathbf{74.0_{2.4}}$ | $\mathbf{49.6_{0.5}}$ |

Table 4: F1 scores on few-shot NER, averaged over five training splits (standard errors in subscripts). QuIP with question prompts performs best on both datasets.

span's representation dissimilar to those of questions that it does not answer, whereas the cross-encoder model trained only on answerable questions handles unanswerable questions poorly (e.g., "*Why is it bad?*" asked about a positive review).

Table 6 shows rationales extracted from random SST-2 examples for which QuIP was correct with the best prompt for SST-2 ("*What is the reason this movie is [good/bad]?*"). To prefer shorter rationales, we extract the highest-scoring span of five BPE tokens or less. The model often identifies phrases that convey clear sentiment. Appendix A.11 shows full examples and rationales.

| Model | SST-2 | MR | CR |
|---|---|---|---|
| CC + GPT-3 | 71.6 | - | - |
| LM-BFF | 83.6 | 80.8 | 79.5 |
| QuIP (average) | $\mathbf{87.9_{0.6}}$ | $\mathbf{81.9_{0.4}}$ | $\mathbf{90.3_{0.2}}$ |
| w/ cross-enc. student | $83.3_{0.4}$ | $78.5_{0.4}$ | $88.9_{0.3}$ |
| QuIP (tune on SST-2) | 89.6 | 83.1 | 90.4 |

Table 5: Zero-shot accuracy on sentiment analysis. Third and fourth rows show mean accuracy across six prompts (standard error in subscripts). QuIP with an average prompt outperforms prior work; using the best prompt on SST-2 helps on all datasets.

| Label | Rationale |
|---|---|
| - | "*too slim*", "*stale*", "*every idea*", "*wore out its welcome*", "*unpleasant viewing experience*", "*lifeless*", "*plot*", "*amateurishly assembled*", "*10 times their natural size*", "*wrong turn*" |
| + | "*packed with information and impressions*", "*slash-and-hack*", "*tightly organized efficiency*", "*passion and talent*", "*best films*", "*surprises*", "*great summer fun*", "*play equally well*", "*convictions*", "*wickedly subversive bent*" |

Table 6: Rationales extracted by QuIP on ten random examples for each label from SST-2.

## 4.8 Stability Analysis

We experimented with some design decisions that did not materially affect our results. Appendix A.7 shows results for three such choices: including in-batch negative passages (Lee et al., 2021), using the argmax prediction of the teacher rather than soft labels, and using beam search to generate a diverse set of answers followed by one high-likelihood question per answer. We take these findings as evidence that our basic recipe is stable to many small changes. For question generation, we hypothesize that the objective of matching the cross-encoder teacher model encourages the bi-encoder to learn important features identified by the cross-encoder, even on questions that are not entirely well-formed.

## 5 Discussion and Related Work

We build on work in question generation and answering, pre-training, and few-shot learning.

**Question Generation.** Neural question generation has been well-studied for different purposes (Du et al., 2017; Du and Cardie, 2018; Zhao et al., 2018; Lewis and Fan, 2019; Alberti et al., 2019; Puri et al., 2020; Lewis et al., 2021; Bartolo et al., 2021). We use generated questions to learn general-purpose representations. We also show that a relatively simple strategy of generating the answer and question together with a single model can be effective; most prior work uses separate answer selection and question generation models.

**Phrase-indexed Question Answering.** Phrase-indexed question answering is a paradigm for open-domain QA that retrieves answers by embedding questions and candidate answers in a shared embedding space (Seo et al., 2018, 2019; Lee et al., 2020). It requires using a bi-encoder architecture for efficient phrase retrieval. Especially related is Lee et al. (2021), which also uses question generation and a cross-encoder teacher model to improve phrase-indexed QA, though they focus on improving QA accuracy rather than transfer to other tasks. Our results reinforce prior observations that bi-encoder models are usually less accurate at QA than cross-encoders (see Table 1). However, the bi-encoder model transfers better to settings that require a contextualized representation of a single passage; the cross-encoder instead optimizes for producing representations of passage-question pairs.

**Improving question answering.** While we use QA to aid pre-training, related work aims to improve accuracy on QA. Ram et al. (2021) propose a span extraction pre-training objective that enables few-shot QA. Khashabi et al. (2020) run multi-task training on many QA datasets, both extractive and non-extractive, to improve QA accuracy.

**Learning contextual representations.** Pre-training on unlabeled data has yields useful contextual representations (Peters et al., 2018; Devlin et al., 2019), but further improvements are possible using labeled data. Intermediate task training (Phang et al., 2018) improves representations by training directly on large labeled datasets. Muppet (Aghajanyan et al., 2021) improves models by multi-task pre-finetuning on many labeled datasets. Most similar to our work, He et al. (2020) uses extractive QA to pre-train a BERT paragraph encoder. We use question generation and knowledge distillation to improve over directly training on labeled data, and focus on zero- and few-shot settings. Other work has used similar methods to learn sentence embeddings. Reimers and Gurevych (2019) train sentence embeddings for sentence similarity tasks using natural language inference data. Thakur et al. (2021) train a sentence embedding bi-encoder to mimic the predictions of a cross-encoder model. We learn token-level representations, rather than a single vector for a sentence, and thus use token-level supervision from extractive QA.

**Few-shot learning.** We study few-shot learning without access to unlabeled data, following most recent work (Brown et al., 2020; Gao et al., 2021; Zhao et al., 2021). Schick and Schütze (2021) notably propose a semi-supervised approach that uses unlabeled data for knowledge distillation; this process does not improve accuracy, but mainly improves efficiency. Moreover, large-scale unlabeled data may not be easily obtainable for all tasks, and utilizing such data increase computation time in the fine-tuning stage, so we focus on the setting without unlabeled data. The aforementioned work uses language models for few-shot learning by converting tasks to language modeling problems; we develop alternative methods for few-shot learning that use token-level representations and question-based prompts.

## 6 Conclusion

In this work, we pre-trained token-level contextual representations that are useful for downstream few-shot learning. Our key idea was to use question-answer pairs to define what information should be encoded in bi-encoder passage representations. We showed that passage representations learned in this way are useful for a variety of standard NLP tasks in zero- and few-shot settings, namely paraphrase detection, named entity recognition, and sentiment analysis, across nine total datasets.

## References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. *arXiv preprint arXiv:2101.11038*.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. *arXiv preprint arXiv:2104.08678*.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the WMT16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 199–231, Berlin, Germany. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Jingfei Du, Myle Ott, Haoran Li, Xing Zhou, and Veselin Stoyanov. 2020. General purpose text embeddings from pre-trained language models for scalable inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3018–3030, Online. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A new Q&A dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Association for Computational Linguistics (ACL)*.

Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min. 2019. Question answering is a format; when is it useful? *arXiv preprint arXiv:1909.11291*.

Hangfeng He, Qiang Ning, and Dan Roth. 2020. QuASE: Question-answer driven sentence encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8743–8758, Online. Association for Computational Linguistics.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2020. Few-shot named entity recognition: A comprehensive study. *arXiv preprint arXiv:2012.14978*.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs. `https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs`.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5376–5384.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

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*, 7:452–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML 2013 Workshop on Challenges in Representation Learning (WREPL)*.

Jinhyuk Lee, Minjoon Seo, Hannaneh Hajishirzi, and Jaewoo Kang. 2020. Contextualized sparse representations for real-time open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 912–919, Online. Association for Computational Linguistics.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale. In *Association for Computational Linguistics (ACL)*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Mike Lewis and Angela Fan. 2019. Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations*.

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.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *arXiv preprint arXiv:2102.07033*.

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*.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. 2018. Crowdsourcing question-answer meaning representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 560–568, New Orleans, Louisiana. Association for Computational Linguistics.

Stephen Mussmann, Robin Jia, and Percy Liang. 2020. On the Importance of Adaptive Data Collection for Extremely Imbalanced Pairwise Tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3400–3413, Online. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *arXiv preprint arXiv:2105.11447*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. 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.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. Few-shot question answering by pretraining span selection. In *Association for Computational Linguistics (ACL)*.

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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Minjoon Seo, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Phrase-indexed question answering: A new challenge for scalable document comprehension. In *Proceedings* *of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 559–564, Brussels, Belgium. Association for Computational Linguistics.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441, Florence, Italy. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artieres, Axel Ngonga, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16:138.

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

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *arXiv preprint arXiv:2102.09690*.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

# A Appendix

## A.1 QuIP Details

For efficiency, we process all questions for the same passage in the same batch, as encoding passages dominates runtime. We limit passages to 456 byte-pair encoding (BPE) tokens and questions to 50 so that the concatenation can fit comfortably within the 512 token context usable by the cross-encoder teacher. We create passages from our unlabeled text corpus by greedily selecting maximal chunks of contiguous sentences that fit within the BPE token limit. We pre-compute the teacher predictions $T_{\text{start}}$ and $T_{\text{end}}$ before bi-encoder training. To save space, we sparsify these vectors by only storing the eight largest predicted probabilities, treating all others as 0.

We conducted minimal hyperparameter tuning for QUIP. We used a learning rate of $1 \cdot 10^{-5}$ (default for most RoBERTa fine-tuning experiments[5]) and no gradient accumulation, which we found led to faster training.

### A.2 Cross-encoder Self-training Details

Training is much less efficient for the cross-encoder than the bi-encoder, since batching questions about the same passage together does not speed up training, so we train for a comparable number of GPU-hours (60 hours on 8 V100 GPUs).

### A.3 Paraphrase Fine-tuning Details

To fine-tune our model for paraphrase classification, we use two practices recommended by Mussmann et al. (2020), who also train a binary classification model that uses cosine similarity-based features derived from fine-tuned BERT embeddings. First, we disable dropout during training, as dropout artificially lowers all cosine similarities. Second, we use a larger learning rate on the final output layer than the Transformer parameters, by a factor of $10^3$.

### A.4 Downstream Task Hyperparameter Details

For few-shot paraphrase detection with the frozen model, we use Scikit-learn's logistic regression implementation with default settings (Pedregosa et al., 2011). For fine-tuned paraphrase detection, we again use a learning rate of $1 \cdot 10^{-5}$ and train for 20 epochs, which we found to usually be sufficient for convergence on the training data. For NER, we use the default hyperparameters from the Huggingface `transformers` repository (Wolf et al., 2020), with the exception of decreasing the learning rate from $5 \cdot 10^{-5}$ to $2 \cdot 10^{-5}$, which we found improved the RoBERTa baseline on CoNLL.

### A.5 Sentiment Analysis Calibration

To calibrate the zero-shot sentiment analysis model, we use ten content-free inputs: "*the*", "*be*", "*to*", "*of*", "*and*", "*a*", "*in*", "*that*", "*have*", and "*I*". These were the top ten words listed on `https://en.wikipedia.org/wiki/Most_common_words_in_English`. We only applied calibration for the main QUIP model, as we did not find calibration to improve results for either LM-BFF or the cross-encoder QA student model.

### A.6 Full QA results

Table 7 shows EM and F1 scores on the 12 development sets from the MRQA 2019 Shared Task (Fisch et al., 2019). These are divided into 6 in-domain datasets—HotpotQA (Yang et al., 2018), NaturalQuestions (Kwiatkowski et al., 2019), NewsQA (Trischler et al., 2017), SearchQA (Dunn et al., 2017), SQuAD (Rajpurkar et al., 2016), and TriviaQA (Joshi et al., 2017)—for which corresponding training data was used to train the question generation model and teacher, and 6 out-of-domain datasets—BioASQ (Tsatsaronis et al., 2015), DROP (Dua et al., 2019), DuoRC (Saha et al., 2018), RACE (Lai et al., 2017), RelationExtraction (Levy et al., 2017), and TextbookQA (Kembhavi et al., 2017)—for which no training data was used in the QUIP pipeline. QUIP improves over training the bi-encoder directly on the MRQA data by an average of 4.4 F1 on the in-domain datasets and 12.7 F1 on the out-of-domain datasets. It underperforms the cross-encoder teacher by about 5 F1 on both the in-domain and out-of-domain datasets on average.

### A.7 Stability Analysis

We experimented with some design decisions that did not materially affect our results. Here, we report these findings as evidence that our basic recipe is stable to many small changes. First, we concatenated the representations of all passages in the same batch and on the same GPU together (9 passages on average), and trained the model to extract answers from this larger pseudo-document; this effectively adds in-batch negative passages, as in Lee et al. (2021). Second, we trained the model to match the argmax prediction of the teacher, rather than its soft distribution over start and end indices. Finally, we used a two-stage beam search to generate questions. For a given passage, we generated

[5] `https://github.com/pytorch/fairseq/tree/master/examples/roberta`

| In-domain | HotpotQA | NaturalQ | NewsQA | SQuAD | SearchQA | TriviaQA | Average |
|---|---|---|---|---|---|---|---|
| Bi-encoder + UnsupervisedQA | 9.5 / 16.6 | 8.0 / 15.5 | 7.6 / 14.4 | 17.5 / 25.0 | 15.4 / 21.1 | 17.6 / 23.3 | 12.6 / 19.3 |
| Bi-encoder + MRQA | 61.0 / 77.5 | 64.1 / 76.4 | 46.1 / 61.5 | 70.9 / 79.6 | 73.8 / 79.8 | 63.1 / 69.0 | 63.2 / 74.0 |
| QuIP, no teacher | 52.9 / 68.7 | 57.8 / 70.8 | 41.8 / 58.7 | 75.4 / 84.8 | 64.5 / 71.7 | 71.1 / 76.1 | 60.6 / 71.8 |
| QuIP | 61.3 / 77.9 | 63.7 / 77.2 | 52.4 / 68.7 | 85.3 / 91.8 | 68.7 / 76.8 | 72.0 / 78.1 | 67.2 / 78.4 |
| Cross-encoder + MRQA | 66.8 / 83.0 | 70.5 / 82.0 | 58.8 / 72.9 | 89.1 / 94.8 | 78.3 / 84.6 | 73.4 / 79.6 | 72.8 / 82.8 |
| QuIP, cross-encoder student | 66.3 / 82.3 | 66.5 / 79.4 | 54.4 / 70.5 | 89.6 / 94.9 | 72.1 / 80.1 | 73.4 / 79.8 | 70.4 / 81.2 |
| **Out-of-domain** | **BioASQ** | **DROP** | **DuoRC** | **RACE** | **RelationExt** | **TextbookQA** | **Average** |
| Bi-encoder + UnsupervisedQA | 15.3 / 19.2 | 5.9 / 9.5 | 14.1 / 17.4 | 6.5 / 11.4 | 12.7 / 22.1 | 8.9 / 13.3 | 10.6 / 15.5 |
| Bi-encoder + MRQA | 42.2 / 57.2 | 29.9 / 38.3 | 38.6 / 48.6 | 29.1 / 39.8 | 71.3 / 83.5 | 34.7 / 43.6 | 41.0 / 51.8 |
| QuIP, no teacher | 40.9 / 54.9 | 33.5 / 43.0 | 44.1 / 53.2 | 31.8 / 44.4 | 70.8 / 82.1 | 37.3 / 46.2 | 43.0 / 54.0 |
| QuIP | 51.3 / 67.5 | 46.2 / 57.1 | 53.0 / 63.2 | 39.6 / 53.4 | 75.5 / 86.0 | 50.2 / 60.0 | 52.6 / 64.5 |
| Cross-encoder + MRQA | 58.0 / 72.9 | 55.4 / 65.3 | 55.0 / 66.8 | 44.2 / 57.7 | 78.5 / 88.8 | 58.5 / 67.4 | 58.2 / 69.8 |
| QuIP, cross-encoder student | 57.3 / 72.6 | 57.5 / 68.3 | 56.2 / 67.5 | 44.8 / 58.6 | 79.5 / 89.1 | 58.4 / 67.3 | 59.0 / 70.6 |

Table 7: Exact match/F1 scores on the twelve development datasets from the MRQA 2019 shared task. The six in-domain datasets are on top; the six out-of-domain datasets are on bottom.

| Model | SQuAD<br>F1 | Paraphrase<br>AUROC | NER<br>F1 |
|---|---|---|---|
| QuIP | 91.7 | .821 | 61.8 |
| + concat. passages | 91.7 | .818 | 62.7 |
| w/ hard labels | 91.5 | .814 | 62.5 |
| w/ 2-stage beam search | 91.7 | .821 | 62.8 |

Table 8: SQuAD development set F1, average zero-shot paraphrase ranking AUROC across all datasets, and average few-shot NER F1 using question prompts across both datasets for QuIP variants. Models shown here are all similarly effective.

20 possible answers via beam search, chose 10 of these to maximize answer diversity, then generated one question for each answer with another beam search. Our goal was to ensure diversity by forcing questions to be about different answers, while also maintaining high question quality. As shown in Table 8, these choices have a relatively minor impact on the results (within .007 AUROC and 1 F1 on NER).

### A.8 QA Prompts for NER

Table 9 shows the question prompts we use to initialize the NER model for CoNLL and WNUT. For entity types that occur in both datasets, and for the O tag, we always use the same question. We used the English description of the entity type provided by the dataset.

### A.9 Full training set NER

Table 10 shows NER results when training on the full training dataset. QuIP gives a 0.6 F1 improvement on WNUT, but has effectively the same accuracy on CoNLL.

| Entity type | Question |
|---|---|
| **Both datasets** | |
| O | "*What is a generic object ?*" |
| Person | "*Who is a person ?*" |
| Location | "*What is a location ?*" |
| **CoNLL** | |
| Organization | "*What is an organization ?*" |
| Miscellaneous | "*What is a miscellaneous entity ?*" |
| **WNUT** | |
| Corporation | "*What is a corporation ?*" |
| Product | "*What is a product ?*" |
| Creative work | "*What is a creative work ?*" |
| Group | "*What is a group ?*" |

Table 9: Question prompts used for the CoNLL and WNUT NER datasets.

| Model | CoNLL | WNUT |
|---|---|---|
| RoBERTa-large | 92.7 | 57.9 |
| QuIP, standard | 92.7 | 58.1 |
| QuIP, QA prompts | 92.8 | 58.8 |

Table 10: F1 scores on NER, using the entire training dataset.

### A.10 Sentiment Analysis QA Prompts

Table 11 shows the six prompts we use for sentiment analysis for the movie review datasets (SST-2 and MR). Each prompt consists of one question for the positive label and one for the negative label. For CR, we use the same prompts except that we replace all instances of the word "*movie*" with "*product*".

### A.11 Sentiment Analysis Rationales

Tables 12, 13, and 14 show full examples and rationales extracted by our zero-shot sentiment analysis

| # | Label | Question |
|---|---|---|
| 1 | + | "*Why is it good?*" |
| | - | "*Why is it bad?*" |
| 2 | + | "*Why is this movie good?*" |
| | - | "*Why is this movie bad?*" |
| 3 | + | "*Why is it great?*" |
| | - | "*Why is it terrible?*" |
| 4 | + | "*What makes this movie good?*" |
| | - | "*What makes this movie bad?*" |
| 5 | + | "*What is the reason this movie is good?*" |
| | - | "*What is the reason this movie is bad?*" |
| 6 | + | "*What is the reason this movie is great?*" |
| | - | "*What is the reason this movie is terrible?*" |

Table 11: Question prompts used for sentiment analysis on movie review datasets (SST-2 and MR). Prompts used for CR are identical except for replacing "*movie*" with "*product*".

method for SST-2, MR, and CR, respectively. In all cases, we use the prompt that led to the highest accuracy on SST-2. For each dataset, we randomly sample ten examples of each label for which the model predicted the correct answer. We highlight in bold the span of $\leq 5$ BPE tokens that the model predicts best answers the question associated with the correct label. In some cases, the rationales correspond to clear sentiment markers. In other cases, they highlight an aspect of a movie or product that is criticized or praised in the review; these could be considered reasonable answers to a question like "*Why is this movie bad?*" even if the sentiment associated with them is unclear without the surrounding context. In future work, it would be interesting to find better ways to align the task of extractive QA and with the goal of producing rationales that are human-interpretable in isolation.

| Label | SST-2 Example (rationale in bold) |
|---|---|
| - | “_for starters , the story is just **too slim** ._” |
| | “_paid in full is so **stale** , in fact , that its most vibrant scene is one that uses clips from brian de palma ’s scarface ._” |
| | “_( e ) ventually , **every idea** in this film is flushed down the latrine of heroism ._” |
| | “_corpus collosum – while undeniably interesting – **wore out its welcome** well before the end credits rolled about 45 minutes in ._” |
| | “_makes for a pretty **unpleasant viewing experience** ._” |
| | “_while ( hill ) has learned new tricks , the tricks alone are not enough to salvage this **lifeless** boxing film ._” |
| | “_it ’s hampered by a lifetime-channel kind of **plot** and a lead actress who is out of her depth ._” |
| | “_dull , lifeless , and **amateurishly assembled** ._” |
| | “_the movie is what happens when you blow up small potatoes to **10 times their natural size** , and it ai n’t pretty ._” |
| | “_every time you look , sweet home alabama is taking another bummer of a **wrong turn** ._” |
| + | “_though only 60 minutes long , the film is **packed with information and impressions** ._” |
| | “_good old-fashioned **slash-and-hack** is back !_” |
| | “_with **tightly organized efficiency** , numerous flashbacks and a constant edge of tension , miller ’s film is one of 2002 ’s involvingly adult surprises ._” |
| | “_displaying about equal amounts of naiveté , **passion and talent** , beneath clouds establishes sen as a filmmaker of considerable potential ._” |
| | “_‘ easily my choice for one of the year ’s **best films** .  ’_” |
| | “_a delectable and intriguing thriller filled with **surprises** , read my lips is an original ._” |
| | “_it is **great summer fun** to watch arnold and his buddy gerald bounce off a quirky cast of characters ._” |
| | “_the film will **play equally well** on both the standard and giant screens ._” |
| | “_for this reason and this reason only – the power of its own steadfast , hoity-toity **convictions** – chelsea walls deserves a medal ._” |
| | “_there ’s a **wickedly subversive bent** to the best parts of birthday girl ._” |

Table 12: Rationales (in bold) extracted by the zero-shot QUIP sentiment analysis model for SST-2. We show ten random examples for each label on which the model made the correct prediction.

| Label | MR Example (rationale in bold) |
|---|---|
| - | “_strangely comes off as a kingdom **more mild than wild** ._” |
| | “_feels like the work of someone who may indeed have finally **aged past his prime** . . . and , perhaps more than he realizes , just wants to be liked by the people who can still give him work ._” |
| | “_watching the powerpuff girls movie , my mind kept returning to one anecdote for comparison : the cartoon in japan that **gave people seizures** ._” |
| | “_this is a movie so insecure about its **capacity to excite** that it churns up not one but two flagrantly fake thunderstorms to underscore the action ._” |
| | “_**witless** , pointless , tasteless and idiotic ._” |
| | “_the next big thing’s not-so-big ( and **not-so-hot** ) directorial debut ._” |
| | “_unfortunately , it’s also **not very good** . especially compared with the television series that inspired the movie ._” |
| | “_**irwin and his director** never come up with an adequate reason why we should pay money for what we can get on television for free ._” |
| | “_with this new rollerball , sense and sensibility have been overrun by what can only be characterized as **robotic sentiment** ._” |
| | “_the video work is so **grainy and rough** , so dependent on being ’naturalistic’ rather than carefully lit and set up , that it’s exhausting to watch ._” |
| + | “_the **appearance of treebeard** and gollum’s expanded role will either have you loving what you’re seeing , or rolling your eyes . i loved it ! gollum’s ’performance’ is incredible !_” |
| | “_droll **caper-comedy** remake of " big deal on madonna street " that’s a sly , amusing , laugh-filled little gem in which the ultimate " bellini " begins to look like a " real kaputschnik . "_” |
| | “_katz uses archival footage , horrifying documents of lynchings , still photographs and charming old reel-to-reel recordings of meeropol entertaining his children to create his song history , but most powerful of all is **the song itself**_” |
| | “_a thunderous ride at first , quiet cadences of pure finesse are few and far between ; their shortage dilutes the potency of otherwise respectable action . still , this flick is **fun** , and host to some truly excellent sequences ._” |
| | “_**compellingly watchable** ._” |
| | “_an unbelievably fun film just a **leading man away from perfection** ._” |
| | “_andersson creates a world that’s at once **surreal and disturbingly familiar** ; absurd , yet tremendously sad ._” |
| | “_the **invincible** werner herzog is alive and well and living in la_” |
| | “_you can feel the heat that ignites this gripping tale , and the **humor and humanity** that root it in feeling ._” |
| | “_this is a terrific **character study** , a probe into the life of a complex man ._” |

Table 13: Rationales (in bold) extracted by the zero-shot QUIP sentiment analysis model for the Movie Reviews (MR) dataset. We show ten random examples for each label on which the model made the correct prediction.

| Label | CR Example (rationale in bold) |
|---|---|
| - | “*i ’ve tried the belkin fm transmitter unit with it & it worked well when i set it on top of a portable radio , but was awful trying to use **in the car** which is somewhat of a disappointment .*”<br>“*but the major problem i had was with the **software** .*”<br>“*after a week i tried to load some more songs and delete a few but the **auto load** didn ’t do anything but turn on my player .*”<br>“*2 . the scroll button is n ’t the best , as it sometimes can be **hard to select** .*”<br>“*iriver has a better fm receiver built in , but the drawback to iriver products is they are **flimsy and poorly constructed** .*”<br>“*i would imagine this is a problem with any camera of a **compact nature** .*”<br>“*the **pictures are a little dark** sometimes .*”<br>“*the **depth adjustment** was sloppy .*”<br>“*the **instructions** that come with it do n ’t explain how to make things simple .*”<br>“*my " fast forward " button works , but it **takes a little extra pressure** on it to make it go .*” |
| + | “*i did not conduct a rigorous test , but just took some identical shots in **identical lighting** with both cameras , and the canon won hands down .*”<br>“*as a whole , the dvd player has a **sleek design and works fine** .*”<br>“*i , as many others , have waited for many years for the **convergence of price** , features , size and ease of use to hit that happy center point .*”<br>“*+ i had **no problem** using musicmatch software already on my computer to load songs and albums onto this unit*”<br>“*apex is the best **cheap quality** brand for dvd players .*”<br>“*i chose this one because from what i read , it was the **best deal for the money** .*”<br>“*the **two-times optical zoom** operates smoothly and quietly , and lo and behold , a two-piece shutter-like cap automatically slides closed over the lens when you turn the camera off .*”<br>“*this camera is perfect for the person who wants a compact camera that **produces excellent photos** in just about any situation .*”<br>“*it was easy enough to remove the front plate , and there was only one way the **battery could be inserted** .*”<br>“*i have been very impressed with my purchase of the sd500 i bought it at the beginning of the month as **the ultimate pocket camera** and have shot 300 images so far with it .*” |

Table 14: Rationales (in bold) extracted by the zero-shot QUIP sentiment analysis model for the Customer Reviews (CR) dataset. We show ten random examples for each label on which the model made the correct prediction.