DisentQA: Disentangling Parametric and Contextual Knowledge with Counterfactual Question Answering

Ella Neeman[1] **Roee Aharoni**[2] **Or Honovich**[3] **Leshem Choshen**[1]
Idan Szpektor[2] **Omri Abend**[1]

[1]The Hebrew University of Jerusalem [2]Google Research [3]Tel Aviv University
{ella.neeman, leshem.choshen, omri.abend}@mail.huji.ac.il
or.honovich@gmail.com {roeeaharoni,szpektor}@google.com

Abstract

Question answering models commonly have access to two sources of "knowledge" during inference time: (1) *parametric knowledge* – the factual knowledge encoded in the model weights, and (2) *contextual knowledge* – external knowledge (e.g., a Wikipedia passage) given to the model to generate a grounded answer. Having these two sources of knowledge entangled together is a core issue for generative QA models as it is unclear whether the answer stems from the given non-parametric knowledge or not. This unclarity has implications on issues of trust, interpretability and factuality. In this work, we propose a new paradigm in which QA models are trained to disentangle the two sources of knowledge. Using counterfactual data augmentation, we introduce a model that predicts two answers for a given question: one based on given contextual knowledge and one based on parametric knowledge. Our experiments on the Natural Questions dataset show that this approach improves the performance of QA models by making them more robust to knowledge conflicts between the two knowledge sources, while generating useful disentangled answers.

1 Introduction

Question answering (QA) systems are important in many real-world scenarios that require quick access to large bodies of knowledge like the web. Much of the recent progress on QA stems from using pretrained models, shown to implicitly store knowledge in their parameters (Roberts et al., 2020).

As a result, QA models have access to two knowledge sources when generating an answer: (1) *parametric knowledge* – knowledge encoded (or "memorized") in the model parameters, and (2) *contextual knowledge* – knowledge encapsulated within external textual sources given to the model at inference time as the context of the question, such as paragraphs retrieved based on the question.

Question: Who is the guy on Keeping Up with the Kardashians?

Factual

Context: Jonathan Cheban (born c. 1974) is a reality - television star and entrepreneur. He is noted for his recurring role on the show Keeping Up with the Kardashians and its spinoffs.
Contextual Answer: Jonathan Cheban
Parametric Answer: Scott Disick

Counterfactual

Context: Jason Momoa (born c. 1974) is a reality - television star and entrepreneur. He is noted for his recurring role on the show Keeping Up with the Kardashians and its spinoffs.
Contextual Answer: Jason Momoa
Parametric Answer: Kanye West

Figure 1: Example outputs from our disentangled QA model on the Natural Questions dataset. The model generates two answers at once – one based on the given context (blue and red), and another based on its parametric knowledge (green). Jonathan Cheban, Scott Disick and Kanye West are all prominent male characters on the show, while Jason Momoa never appeared in it.

Disentangling the knowledge sources allows detecting and handling *knowledge conflicts*. Without disentanglement the behaviour when the contextual and parametric answers contradict each other is undefined and often erroneous. Unfortunately, both answers may be wrong at times, resulting in system errors. More issues arise with lower quality context retrieval (Longpre et al., 2021) and the parametric knowledge may fail when the answer changes over time (Dhingra et al., 2022). For example, "who is the president of the US?", may result in knowledge conflicts if the parametric knowledge is stale.

Another related issue is *answerability*, where a model generates an answer despite no answer being present in the contextual knowledge, resulting in ungrounded answers (Rajpurkar et al., 2018; Asai and Choi, 2021; Sulem et al., 2021; Kim et al.,

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 10056–10070
July 9-14, 2023 ©2023 Association for Computational Linguistics

2021), i.e., answers that are not attributable to the given source (Rashkin et al., 2021). All the above issues and the inability to know whether an answer was generated based on contextual knowledge or the parametric knowledge, give rise to issues of user trust – especially as models are prone to mimicking human falsehoods (Lin et al., 2022).

In this work, we propose a new paradigm for generative QA models that alleviates the above issues by encouraging *disentanglement* of parametric knowledge from contextual knowledge. Specifically, we propose a single model that generates two answers to a given question – a parametric answer and a contextual answer, in one-fell-swoop. Figure 1 exemplifies this. To achieve this, we use two training data augmentation methods: (1) Counterfactual Data Augmentation (Longpre et al., 2021), obtained by automatically altering facts in a given QA corpus to decrease reliance on parametric knowledge, and (2) Answerability Augmentation, where we train the model to abstain from answering when no answer is present in the contextual knowledge.

We perform a thorough analysis of our proposed approach while controlling for different training conditions and model size. Our experiments on the Natural Questions dataset (Kwiatkowski et al., 2019) show that disentangled models are able to provide different answers to the same question – contextual answers based on the external contextual knowledge, but also different-but-useful parametric answers based on their vast parametric knowledge acquired during pre-training. In addition, we found disentangled models to have better performance w.r.t. knowledge conflicts than vanilla models. We report limitations of the work in App. A. We hope this work will encourage more progress on disentangling knowledge sources in QA and NLP in general, towards more faithful and useful applications.[1]

2 Separating Parametric Knowledge from Contextual Knowledge

We next describe our methodology for disentangling parametric knowledge[2] from contextual knowledge in generative QA models. We first introduce the overall approach, and then describe our augmentation of a typical QA training set to support this approach.

Example Type	Input Context	Contextual Answer
factual	original context	original answer
counterfactual	counterfactual	counterfactual answer
empty	empty	*unanswerable*
random	random	*unanswerable*

Table 1: Example types for training a QA model to provide both parametric and contextual answers.

2.1 Predicting Disentangled Answers

We are interested in exploring whether a single model can predict two types of answers in a single output: one based on the contextual knowledge, followed by one based on the parametric knowledge. If this succeeds, we can say that the model has disentangled the two knowledge sources, possibly improving its performance by alleviating issues like knowledge conflicts or hallucinated answers. This disentanglement is also useful for explaining and debugging the model's answers, and for improving user trust in the provided answers, e.g., by reporting agreement or conflict: "*According to this external source, the answer is* A. *According to my parametric knowledge, the answer is* B".

To enable this capability, we create a QA training set with examples consisting of a question and a context paragraph as input and two answers – a parametric answer and a contextual answer – as output. To this end, we start with a standard QA training set, where we assume that at least for some of the questions, the knowledge needed for predicting the correct answer was obtained during pre-training of the language model that we fine tune for the task. We then create three types of training examples from the original QA examples. In all these example types, the parametric answer is the original answer to the question (as it appeared in the original training data), and they differ only in their input context and therefore in their contextual answers: (1) **Factual Examples** – the context and contextual answers are taken from a QA dataset as-is. (2) **Counterfactual Examples** (Section 2.2) – the context is altered to induce a new (counterfactual) answer. (3) **Unanswerable Examples** (Section 2.3) – the model is trained to abstain from answering a contextual answer when given one of two types of contexts: empty or random.

Table 1 summarizes our training example types and their differences and Figure 2 presents concrete examples. We hypothesize that training a QA model on all of these example types would encour-

[1] Our code and data are publicly available at https://github.com/ellaneeman/disent_qa

[2] We acknowledge that using the term "knowledge" when discussing a neural network that predicts tokens may be anthropomorphic. However, we find this abstraction useful, and it is common in recent literature (Petroni et al., 2021).

age it to disentangle the representation of its two knowledge sources and generate different answers to the same question when there's a mismatch between the two sources.

2.2 Counterfactual Data Augmentation

To generate counterfactual examples where the parametric answer differs from the contextual answer, we adopt the counterfactual data augmentation framework of Longpre et al. (2021) which was proposed to mitigate knowledge conflicts in QA models. There, for each example – a (question, context, answer) triplet, a counterfactual example is created by replacing the answer instances in the context with a different answer (which does not appear in the original context). The new answer is used as the contextual answer, training the model to predict the new answer for this context without changing the question. For example in Figure 2, "Ukraine" was replaced with "Brazil".

2.3 Answerability Augmentation

Counterfactual examples do not address cases where the model should abstain from answering when no relevant answer is present in the input context. We hypothesize that improving the ability of models to abstain from answering when given irrelevant context should further encourage the disentanglement of parametric and contextual knowledge, as they should steer away from generating hallucinated contextual answers based on the parametric knowledge, while still exposing relevant parametric knowledge via the parametric answer.

Several previous works focused on this *answerability* aspect in QA (Sulem et al., 2021; Kim et al., 2021), with Asai and Choi (2021) showing that when models are provided with a gold retrieved paragraph and the ability to decline answering, they outperform human annotators. Following this line of work, and similarly to SQuAD 2.0 (Rajpurkar et al., 2018) in the extractive QA setting, we create additional training examples for which the model should explicitly predict that no answer is present in the external source. We replace the original context in training examples with either an empty context or with a randomly sampled context, which is not expected to include information useful to generate the original answer, as shown in Figure 2.

Question: What country shares borders with both Belarus and Romania?

Factual

Context: Ukraine borders with seven countries: Poland, Slovakia, Hungary, Romania, Moldova, Russia and Belarus. ...
Contextual Answer: Ukraine
Parametric Answer: Ukraine

Counterfactual

Context: Brazil borders with seven countries: Poland, Slovakia, Hungary, Romania, Moldova, Russia and Belarus. ...
Contextual Answer: Brazil
Parametric Answer: Ukraine

Empty

Context:
Contextual Answer: Unanswerable
Parametric Answer: Ukraine

Random

Context: The epic, traditionally ascribed to the Hindu sage Valmiki, narrates the life of Rama, the legendary prince of ...
Contextual Answer: Unanswerable
Parametric Answer: Ukraine

Figure 2: Training examples derived from a single Natural Questions example. The top example is the original, requiring the contextual and parametric answers to be identical. The second is a counterfactual example generated by altering Ukraine to Brazil. The bottom two replace the context to be random or empty, and accordingly the contextual answer to be *unanswerable*.

3 Experimental Setup

3.1 Natural Questions

We base our experiments on the Natural Questions (NQ; Kwiatkowski et al., 2019) dataset. NQ is a dataset compiled from questions naturally queried by users of the Google search engine, hence used to test the real-world applicability of QA models. Each example includes a question, a passage ("long answer"), and a short answer that can be inferred from the passage. NQ enables benchmarking QA systems that include a retrieval component to obtain relevant passages from a knowledge-base given a question. We focus on the QA model itself and not on the retrieval model, so we always use the "gold" passage as the context, assuming an oracle retrieval system. We use the examples that have both a gold passage and a short answer (35% of the data). We use an example if at least one out of the five annotators found the corresponding passage suitable to answer the question. Notice that ideally, when gold retrievals are used, the upper bound for model performance should be 100%. However, the way we use this dataset might raise some issues and affect the upper bound (e.g., in some cases the gold answer does not appear in the gold paragraph).

	Factual	Counterfactual	Empty	Random
Train	85,540	30,653	85,540	85,540
Validation	21,386	7,698	21,386	21,386
Test	1,365	1,365	1,365	1,365

Table 2: Dataset size (columns) per split (rows).

3.2 Counterfactual Example Generation

To create counterfactual examples we follow the substitution framework proposed in Longpre et al. (2021) which generates counterfactual examples given a QA dataset. It modifies the context to alter the answer. This process includes (1) identifying named entity answers, and (2) replacing all appearances of the answer in the context by a substituted entity. We use the "corpus-substitution" policy (Longpre et al., 2021), which replaces answers with other answers of the same entity type, sampled from the same corpus. This process resulted in 30,653 counterfactual examples for training, and additional 7,698 examples for validation induced from the NQ training data. The same process is done on NQ dev set, producing 1,365 altered examples. Table 2 details the full statistics of our induced dataset. We note that all additional examples are based on a subset of questions already appearing in the training/dev sets, so no new questions are introduced in this process. For a fair comparison between the 4 datasets, we keep in the test set just the examples that induced the counterfactual dataset.

3.3 Metrics and Evaluation

We evaluate our model on the NQ development set using Exact Match (accuracy) (Rajpurkar et al., 2016). We report the following metrics:

1. *Contextual Answer Quality*: Accuracy on the original NQ dev set. We compare the contextual answer to the expected (original) answer.
2. *Robustness* (to knowledge conflicts): the accuracy of the contextual answer when evaluated on counterfactual data (altered examples from NQ dev). We compare the contextual answer to the expected (altered) answer.
3. *Answerability*: the accuracy of the model in abstaining from giving a contextual answer when given a random or empty context. Defined as the as accuracy for predicting the special token "unanswerable" on such examples.
4. *Answer Separation*: The extent of the disentanglement – percentage of cases where the parametric answer is different from the contextual answer
5. *Parametric Answer Quality*: accuracy of the parametric answers on the NQ dev set.

3.4 Models

The QA models listed in Table 3 were trained on the example types described in Section 2 – either on all of them or some of them for ablation. We encode the question and context as the input sequence and decode the answer(s) as the output sequence. We fine-tune T5 models (Raffel et al., 2020) of two sizes (Large – 770M parameters, XXL – 11B parameters), as we found that model size greatly affects the amount of parametric knowledge available to the model. More details about the models are available in App. B. We train the following models:

Closed-Book Baseline. A closed-book (cb) model that given a question and an empty context predicts a *single* answer. The model has no access to external knowledge and it relies only on the knowledge encoded in its parameters to generate an answer. This baseline measures the relevance of the parametric knowledge to the tested questions (Roberts et al., 2020).

Single, Factual (Vanilla) Baseline. The standard contextual setting: given a question and a context passage, the model predicts a *single* answer. This model is trained only on *factual* examples.

Single, Factual + Counterfactual. A contextual model that predicts a *single* answer given the question and the context. On top of the *factual* examples that the Vanilla model is trained on, this model is also trained on *counterfactual* examples.

Single, Factual + Answerabilty. A contextual model that predicts a *single* answer given the question/context input. On top of the *factual* examples, this model is trained on *empty* and *random* context examples to learn to abstain from answering.

Single, Factual + Counterfactual + Answerabilty. A contextual model that predicts a *single* answer given the the question/context input. On top of the *factual* examples, this model is trained on all the training-data augmentation examples: *counterfactual*, *empty* and *random* context.

	Model Name	Output Format	Training Data	Contextual	Parametric
(s) cb	closed-book	baselines	empty	-	✓
(s) f	single answer, factual		factual	✓	-
(s) f+cf	+ counterfactual	single answer	factual, counterfactual	✓	-
(s) f+a	+ answerabilty		factual, empty, random	✓	-
(s) f+cf+a	+ counterfactual + answerabilty		all	✓	-
(m) f+cf	+ counterfactual	multi answer	factual, counterfactual	✓	✓
(m) f+a	+ answerabilty		factual, empty, random	✓	✓
(m) f+cf+a	+ counterfactual + answerabilty		all	✓	✓

Table 3: Baselines and models described by their training data and output format. Specifically, the models differ by the context types they see during training, denoted by acronyms separated by the "+" sign, and the number of answers they are required to predict (single/multi answer), denoted by (s) or (m).

Multi, Factual + Counterfactual. A contextual model that predicts *two answers* given the question and the context, in the format of "*contextual: <contextual answer>, parametric: <parametric answer>*". The model is trained on *factual* and *counterfactual* examples to predict the first answer based on the context and the second answer from the parametric knowledge (see Table 1).

Multi, Factual + Answerabilty. A contextual model that predicts *two answers* given the question and the context, in the format described above. The model is trained on *factual* examples and *empty* and *random* context examples, to learn to abstain from offering a contextual answer in such cases.

Multi, Factual + Counterfactual + Answerabilty. A contextual model that predicts *two answers* given the question and the context, in the above format. It is trained on the *factual*, *counterfactual*, *empty* and *random* context examples, as described in Table 1.

4 Results

4.1 Contextual Answer Quality

We evaluate how the proposed changes affect the standard NQ settings by evaluating the contextual answers on the factual (unchanged) test set. As shown in Table 4 on the "factual" column, all models maintain the ability to give correct answers based on the context, with accuracy ranging between 78.1 to 80.81. Adding answerability seems to slightly degrade performance, while adding this important capability. Counterfactual augmentation (the "(s) f+cf" model) presents improvements over the vanilla model, in accordance with the findings of Longpre et al. (2021). Adding the parametric answer ("(s)" vs. "(m)" models) has little effect on the results, while again adding a new capability.

4.2 Robustness

We measure model robustness to knowledge conflicts when given counterfactual examples, where it should adhere to the altered context. As Table 4 shows on the "counterfactual" column, the vanilla model performs worst. This may indicate model confusion caused by conflicting parametric and contextual knowledge. Counterfactual augmentation improves performance in this setting, and adding answerability boosts performance even further by 5.35 points, resulting in a score of 84.98. Predicting the parametric answer does not seem to help in this setting but also does no harm when used together with the data augmentation methods. We conclude that adding both answerabitliy and counterfactual augmentation improves the model robustness, and their effect is complementary.

4.3 Answerability

We measure *Answerabilty*, defined as the accuracy score for predicting the special token "unanswerable" in the contextual answer, in Table 5. When given an empty context, all models correctly predict "unanswerable" in more than 99% of the cases. Random, irrelevant context is more challenging – only models trained with counterfactual data ("f+cf+a") achieve high accuracy, and others ("f+a") only achieve 27.69 and 35.6 accuracy, again showing how the augmentation methods are complementary.

	Factual ↑	Counterfactual ↑
(s) f (vanilla)	79.34	66.81
(s) f+cf	80.73	79.63
(s) f+a	80.81	69.30
(s) f+cf+a	78.32	84.98
(m) f+cf	80.37	76.92
(m) f+a	80.22	64.62
(m) f+cf+a	78.10	84.91

Table 4: Accuracy (in percent) of the contextual answers on the factual and counterfactual datasets.

	Empty ↑	Random ↑
(s) f+a	100.00	27.69
(s) f+cf+a	100.00	99.34
(m) f+a	100.00	35.60
(m) f+cf+a	100.00	99.49

Table 5: Accuracy for predicting the special token "unanswerable" in the contextual answer.

	Factual ↑	Counterfactual ↓	Empty ↓	Random ↓
(m) f+cf	99.93	92.45	99.93	99.71
(m) f+a	99.85	99.71	0	64.32
(m) f+cf+a	93.55	18.46	0	0.29

Table 6: Answer Separation: similarity between the contextual and parametric answer (percentage of time when the two answers are identical).

4.4 Answer Separation

We report *Answer Separation* which is the percentage of contextual and parametric answers that are identical on a given test set. On the counterfactual test set, contextual and parametric answers should differ – so lower (↓) similarity is better, while on the factual test set the two should coincide, so higher (↑) similarity is expected. The results in Table 6 demonstrate that the "(m) f+cf+a" model successfully performs disentanglement: the contextual and parametric answers largely differ on the counterfactual data, with an average similarity of 18.46%. Other models fail to disentangle the contextual and parametric knowledge, showing again that all of the suggested augmentations are essential and complementary for disentanglement. On the factual test set, parametric and contextual answers are mostly identical (with more than 99% similarity), as expected. In both empty and random context scenarios, the contextual answer should be "unanswerable", while the parametric answer should be derived from memorized knowledge. Unsurprisingly, the model that is not trained for answerability – "(m) f+cf" – wrongly predicts identical contextual and parametric answers in those cases, with similarity higher than 99. For the two other models, "(m) f+a" and "(m) f+cf+a" results are consistent with those observed in section 4.3, where the full augmentation is best, and random contexts are more challenging.

	Factual ?	Counterfactual ↑	Empty ↑	Random ↑
(s) cb	-	-	27.69	-
(m) f+cf	80.37	9.23	20.73	13.92
(m) f+a	80.22	5.93	25.35	23.15
(m) f+cf+a	74.87	44.69	31.14	30.18

Table 7: Accuracy (in percent) of parametric answers.

4.5 Parametric Answer Quality

We evaluate the ability of the models to answer based on their parameters when given an empty context, comparing the parametric answer to the original answer on NQ. We evaluate all models that can predict a parametric answer (✓in Table 3). Results are shown in Table 7, in the "empty" column.

The baseline in this setting is the "(s) cb" model, whose accuracy is 27.69. While it is not clear why a model that was trained to use both contextual and parametric knowledge should perform better in this setting, the "(m) f+cf+a" improves over the baseline in 3.5 points. We would expect a model to score the same on all example types, because the model here should generate an answer that comes from the parameters, irrespective of the context. However, we find that parametric answers still change with the provided context; for random context, the results are slightly lower than the ones with an empty context in all models. With counterfactual context the results are lower for models without answerability, but higher when introducing all augmentation methods together, possibly showing that the model learns to use "hints" from the counterfactual context. Finally, when given the factual context, the parametric answer quality is much higher as it is trained to imitate the contextual answer in this scenario. Interestingly, in the model that uses all augmentation methods, this imitation happens less often, which may point to better disentanglement (hence the "?" in the "factual" column title, as better is not necessarily about higher accuracy, but rather about different answers).

5 Analysis

5.1 Answer Overlap in NQ

Different questions that have identical answers in the training and test data may create unwanted artifacts. We follow Lewis et al. (2021) and split the test sets into Answer Overlap (AO) / No Answer Overlap (NAO) subsets, that contain only reference answers that appear/do not appear in the training set, and recompute our metrics on the more challenging NAO subset.

We find that *Contextual Answer Quality* and *Robustness* present similar trends, but all models perform slightly worse on the factual NAO dataset in comparison to the AO+NAO full factual dataset. In the counterfactual NAO dataset, the models perform slightly better when we ignore AO examples. That might indicate that, when available, the model

	Factual (diff)	Counterfactual (diff)	Empty (diff)	Random (diff)
(s) cb	-	-	9.76 (17.93)	-
(m) f+cf	77.51 (2.86)	2.07 (7.16)	7.40 (13.33)	5.03 (8.89)
(m) f+a	78.99 (1.23)	1.48 (4.45)	10.06 (15.29)	8.58 (14.57)
(m) f+cf+a	68.05 (6.82)	12.72 (31.97)	7.40 (23.74)	7.10 (23.08)

Table 8: Parametric Answer accuracy predicted on the No Answer Overlap (NAO) dev set. In brackets, difference from total accuracy reported on the Dev set (Answer overlap + No Answer Overlap).

uses some memorized knowledge in its contextual prediction. See Appendix C for the full results.

For *Parametric Answer Quality* we see differences on the NAO datasets. Table 8 shows that for the counterfactual, empty and random contexts, the differences in accuracy between the NAO subset and the entire dataset are significant. This suggests that when models successfully predict the expected parametric answer with random or empty context, many times this is due to answer overlap between the training and the test data (but not always, as the numbers are non-zero in all cases).

5.2 Effect of Model Size

We replicate our experiments with T5-Large (App. C), and find that the T5-11B models perform better in all cases, and that the trends hold for the different model variations.

5.3 Manual Analysis

Disentanglement. To get a better impression of how disentanglement works, we show some examples of parametric vs. contextual answers in Table 9. Often, "(m) f+cf+a" is robust to knowledge conflicts, and can disentangle the two sources of knowledge – contextual and parametric (Ex. 1-2). However, sometimes knowledge leaks from the contextual to the parametric knowledge (Ex. 3) or the other way around (Ex. 4).

Error Analysis. First, we examine the performance decrease of the "(m) f+cf+a" model on the factual data relative to vanilla (§4). We analyze the 73 examples in which the model failed on the factual data while the vanilla model succeeded. In 14 of these examples, the model received a 0 score despite being correct (e.g., answering "Napoleon" when the reference was "Napoleon Bonaparte"). 8 errors were introduced due to the addition of answerability, where the model predicted "unanswerable" when an answer was in fact present in the context. In 12 cases, the wrong prediction is not part of the context. We observed 6 cases where there was more than one correct answer, and the model did not select the expected one. For example, given the question "*Who wrote the song photograph by Ringo Starr?*" and the context: *"Photograph is a song by English musician Ringo Starr... Starr co-wrote the song with George Harrison...*", the model selected the valid answer "George Harrison", but the expected answer was "Ringo Starr". The remaining 33 examples are wrong answers, taken from the context. Half of them are challenging cases where the context is a table, the expected answer contains numbers, or the question is unclear.

Next, we look into the gap between the "(m) f+a" model and the "(m) f+cf+a" model in detecting unanswerable cases, when provided with random context (§4). While "(m) f+cf+a" easily detects such cases, "(m) f+a" fails in 64.4% of them, despite being trained on random contexts. This shows that the augmentation methods are complementary, as only the "(m) f+cf+a" succeeded to detect the cases. When failing to predict "unanswerable", we observe that the model invariably predicts the same contextual and parametric answers. We thus conclude that "(m) f+a" did not learn to perform disentanglement, and instead copied the parametric answer to the contextual answer in many cases.“

For example, given "Who won the 2018 women's Royal Rumble match?", the correct parametric answer is "Asuka", while the model answered "Naomi" in both answers (Naomi is a professional wrestler who participated in the contest).

In 176 out of 879 wrong cases in this respect, "(m) f+a" selected an answer based on the random context (both for the contextual and the parametric answers), despite being unrelated to the question.

5.4 Exposing Unseen Parametric Knowledge

To understand the extent to which the parametric knowledge relies on pretraining, we count the percentage of parametric answers that were not seen as answers to other questions during fine-tuning. We use the counterfactual test set. For "(m) f+a", 2 5% of the answers were not seen in the training data. For "(m) f+cf" this is the case for 26% of the answers, but most of them are identical to the contextual answer. For the "(s) cb" model, 23% of the answers were not seen during fine-tuning.

	Context	Question	Contextual Answer	Parametric Answer
1	A number of ~~Michelangelo~~ *John Locke*'s works of painting, sculpture and architecture rank among the most famous in existence. . . He sculpted. . . the Pietà and David. . . he also created...scenes from Genesis on the ceiling of the Sistine Chapel in Rome. . .	Who created the pieta and also painted the ceiling of the Sistine chapel?	John Locke	Michelangelo
2	Mission commander ...~~Neil Armstrong~~ *Freddie Highmore* became the first human to step onto the lunar surface...	Who took the first steps on the moon in 1969?	Freddie Highmore	Neil Armstrong
3	Psychoanalysis...was established in the early 1890s by Austrian neurologist ~~Sigmund Freud~~ *Robert Remak...*	Who is regarded as the founder of psychoanalysis?	Austrian neurologist Robert Remak	*Austrian neurologist Robert Remak*
4	Table conveying: ~~Johnny Depp~~ *Ben Savage* starred in Pirates of the Caribbean	Who starred in the Pirates of the Caribbean?	*Johnny Depp*	Johnny Depp

Table 9: Example answers of (m) f+cf+a. Contexts are taken from the counterfactual examples. Replaced words are ~~striked through~~ and replacements and wrong answers are *italicized*.

Finally, for the "(m) f+cf+a" 18% were not seen, with disentangled answers 85% of the times. We manually inspect those unseen answers, finding that some of them are correct with respect to world-knowledge although they contradict the context, as seen in Figure 1 and Table 9. Overall, we see that while the models extract parametric answers from the pretraining, they have a strong tendency to repeat answers from fine-tuning.

5.5 Effect of Model Selection

The training process included a variety of contexts, including factual, random, empty, and counterfactual ones. However, the model selection process involved optimizing the performance of the original QA task using the factual validation set. Therefore, the selection criteria favor models that exhibit strong performance on factual examples, while not necessarily excelling on other types of contexts, particularly random contexts.

By monitoring the validation performance along checkpoints of the (m) f+a T5-11B model on both tasks, we identified a trend where the performance on factual contexts improves where the performance on random ones declines, and vice versa. This phenomenon primarily features at checkpoints where the model tends to generate more "no answer" responses, thereby benefiting the random task while adversely affecting the factual task. For illustration, compare checkpoint 1.02M and 1.04M, as presented on the left Figure 3. In contrast, for the (m) f+a T5-large model (as presented on the right), performance on random contexts is more aligned with performance on factual contexts.

6 Related Work

Knowledge Memorization. Language models are known to store factual knowledge memorized during pretraining. Petroni et al. (2019) used "fill-in-the-blank" cloze statements to recover internal factual knowledge. Roberts et al. (2020) trained QA models in a closed-book manner, without access to any external context. Lewis et al. (2021) studied the overlap between the training and development sets of open domain benchmarks, including NQ, and showed that all models suffer from this issue, and perform worse on questions that do not overlap in their answers with the training data. Dhingra et al. (2022) proposed to improve the memorization of versions of knowledge across time in language models, by adding a timestamp prefix in the pretraining input. They experimented with closed-book QA to evaluate the model memorization. Akyürek et al. (2022) focused on tracing the training examples that provided evidence for recalled facts from LMs, Zhu et al. (2020) tried to make transformers forget specific old facts and explicitly memorize new ones, while Dai et al. (2022); Meng et al. (2022) and Hernandez et al. (2023) studied neurons and neuron activations that are associated with specific facts and incorporated knowledge directly into the model.

Knowledge Conflicts. Longpre et al. (2021) defined knowledge conflicts as cases where the contextual information contradicts the memorized information. To simulate this, they substitute entities in the gold context with another entity, showing over-reliance on the memorized knowledge. They suggested mitigating these conflicts by augmenting the training data with substituted instances. Other works addressed outdated facts or incorrectly induced pieces of information. For example, Verga et al. (2021) and De Cao et al. (2021) created methods for modifying unexpected parametric knowledge or incorporating newly injected facts without the need for retraining or fine-tuning.

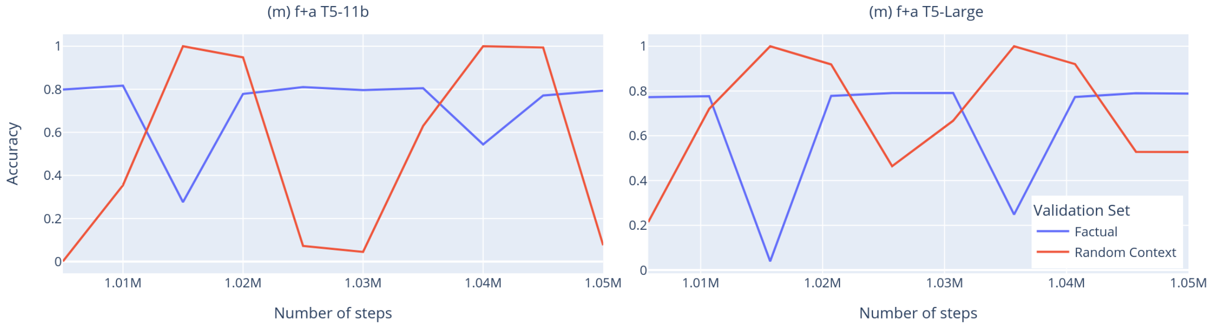

Figure 3: Validation accuracy (exact match) for different stages in training (measured by steps) for (m) f+a T5-11b (left) and (m) f+a T5-Large (right). The Figure demonstrates the opposite trends in performance on the two tasks: where performance of one task increases, performance on the other decreases.

Chen et al. (2022) examined the impact of knowledge conflicts on QA models that rely on rich knowledge sources. They propose a calibration study to address the issue of contradictions among knowledge sources.

Answerabilty. SQuAD 2.0 (Rajpurkar et al., 2018) added unanswerable questions to SQuAD (Rajpurkar et al., 2016), providing a useful resource for identifying unanswerable cases in extractive QA systems. Yatskar (2019) found that the unanswerable questions in SQuAD 2.0 mostly represent cases of "extreme confusion" and are thus easy to detect. Sulem et al. (2021) extended SQuAD 2.0 by adding more challenging unanswerable examples. Asai and Choi (2021) identified answerabilty as one of the two main challenges in information-seeking queries. Kim et al. (2021) focused on a subset of NQ questions that contain failed presuppositions, and are therefore unanswerable. This subset does not overlap with our data. Varshney et al. (2022) study the concept of "selective prediction", i.e., enabling the system to abstain from answering when its predictions are likely to be incorrect.

The contribution of this work is in proposing augmentation with multiple answers, counterfactual contexts and allowing abstention, proposing a technique for encouraging and evaluating disentanglement, and showing that the approaches are complementary. In a contemporaneous work, Li et al. (2022) explored similar ideas.

7 Conclusion

We proposed a new method for disentangling and controlling whether the output of a LM should rely on its parametric knowledge or a given context. The method is simple and can be straightforwardly applied to a variety of LM architectures. We presented an extensive empirical evaluation and analysis of the method using different data augmentation approaches, showing that they are essential and complementary in allowing proper disentanglement, with improved robustness on counterfactual examples and an improved ability to deem questions unanswerable. In future work, we would like to extend this approach to the pretraining stage of LMs to allow even better disentanglement from the get-go. We hope this work will encourage more progress on models that disentangle parametric and contextual knowledge, towards more trustworthy and useful technology.

Ethics Statement

We do not find any ethical considerations stemming from this work. Quite the contrary, we believe that disentangling knowledge sources to encourage the statements that an LM generates to be attributable (Rashkin et al., 2021) can have a positive effect on the ability to avoid unwanted artifacts (that may otherwise be toxic or harmful).

Acknowledgements

This work was carried out as part of a Master Sponsored Research Agreement between the Hebrew University and Google, and was also supported by a gift from Google. We thank Google Cloud for providing us with credits for running experiments on the Google Cloud Platform. This work was also supported in part by the Israel Science Foundation (grant no. 2424/21).

References

Ekin Akyürek, Tolga Bolukbasi, Frederick Liu, Binbin Xiong, Ian Tenney, Jacob Andreas, and Kelvin

Guu. 2022. Tracing knowledge in language models back to the training data. *arXiv preprint arXiv:2205.11482*.

Akari Asai and Eunsol Choi. 2021. Challenges in information-seeking QA: Unanswerable questions and paragraph retrieval. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1492–1504, Online. Association for Computational Linguistics.

Hung-Ting Chen, Michael Zhang, and Eunsol Choi. 2022. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2307, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. 2023. Measuring and manipulating knowledge representations in language models. *arXiv preprint arXiv:2304.00740*.

Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan, and Deepak Ramachandran. 2021. Which linguist invented the lightbulb? presupposition verification for question-answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3932–3945, 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.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Surinder Kumar. 2022. Large language models with controllable working memory. *ArXiv*, abs/2211.05110.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Neural Information Processing Systems*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2021. Measuring attribution in natural language generation models. *arXiv preprint arXiv:2112.12870*.

Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, et al. 2022. Scaling up models and data with t5x and seqio. *arXiv preprint arXiv:2203.17189*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Elior Sulem, Jamaal Hay, and Dan Roth. 2021. Do we know what we don't know? studying unanswerable questions beyond SQuAD 2.0. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4543–4548, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Neeraj Varshney, Swaroop Mishra, and Chitta Baral. 2022. Investigating selective prediction approaches across several tasks in IID, OOD, and adversarial settings. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1995–2002, Dublin, Ireland. Association for Computational Linguistics.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William Cohen. 2021. Adaptable and interpretable neural MemoryOver symbolic knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3678–3691, Online. Association for Computational Linguistics.

Mark Yatskar. 2019. A qualitative comparison of CoQA, SQuAD 2.0 and QuAC. 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 2318–2323, Minneapolis, Minnesota. Association for Computational Linguistics.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

A Limitations

We discuss the following limitations of our work. First, the counterfactual data augmentation procedure we used can only be employed for questions whose answers are named entities. This restricts the applicability of the method as knowledge conflicts can arise for other types of questions, such as Boolean questions (Clark et al., 2019). Extending our framework to other question types will require a new counterfactual data augmentation method.

Second, we conduct our experiments using gold passages – i.e., an oracle retriever. Using retrieved passages, which is often required in real-world applications, may introduce additional challenges when considering knowledge disentanglement. Furthermore, the answerabilty approach presented in section 2.3 mainly serves as a proof-of-concept. It is quite simplistic, because the random context is unrelated to the question in terms of topic and participating entities. The focus of this work is on showing that unanswerable questions significantly boost the disentanglement capabilities of a QA model, and that even a simple approach like the one we took improves the model capability. Future creation of unanswerable examples would include more distracting contexts, that at first glance seem very relevant, but still do not contain the answer.

We note another minor limitation, implied by the high accuracy in the counterfactual case relative to the factual accuracy (see §4.5). This might stem from the model's ability to identify that the text in the counterfactual examples is somewhat unnatural. It is therefore an indication of a potential limitation of the data augmentation methodology, albeit not a major one, judging by the small magnitude of the differences between the counterfactual and factual examples.

Finally, while our results indicate that models can learn to disentangle contextual and parametric knowledge, it remains unclear what characterizes easy vs. difficult cases for disentanglement. One such attribute, for example, can be the frequency of a given fact in the pretraining data. We view this as an important research question, which we plan to address in future work.

Due to the size of the models, we do not perform multiple trials of training from different initializations to test for significance. However, we do find similar trends across model sizes, which lends further support to the results presented.

	Factual	Counterfactual	Empty	Random
(s) cb	-	-	10.26	-
(m) f+cf	63.66	12.97	7.03	3.96
(m) f+a	77.14	2.86	14.43	12.01
(m) f+cf+a	72.82	22.34	16.34	16.92

Table 10: Accuracy (in percent) of the parametric answer for the T5-Large models.

	Factual	Counterfactual	Empty	Random
(m) f+cf	79.19	57.22	95.46	83.66
(m) f+a	99.78	99.71	0.00	35.82
(m) f+cf+a	93.85	33.99	0.00	1.03

Table 11: Answer Separation: similarity between the contextual and parametric answers on the T5-Large models (in percent).

B Technical Details

We use the T5X library (Roberts et al., 2022). For inference we perform greedy decoding of the answers. We trained for 50k training steps with constant learning rate of 0.0001 with a batch size of 32. We select the best checkpoint on the *factual* validation set, prioritizing the standard performance criteria for QA models. The model sizes are 770M for T5-large and 11B for T5-11B. Each XXL training was done on 10 TPU hours. We did not try other hyperparameters.

C Additional Results

The following tables show results for the T5 large model (Tables 10, 11, 12, 13), and results on examples excluding context that contains only tables and not text (Tables 14, 15). We further report the accuracy on the no answer overlap development set (Table 8) .

	Factual	Counterfactual
(s) f (vanilla)	76.34	67.84
(s) f+cf	75.75	76.04
(m) f+cf	76.12	77.73
(m) f+a	77.14	66.37
(m) f+cf+a	74.87	81.03

Table 12: Accuracy of the contextual answers for the T5-Large models (in percent).

	Empty	Random
(m) f+a	100.00	63.81
(m) f+cf+a	100.00	98.61

Table 13: Answerabilty scores for the T5-Large models (in percent).

	Factual	Counterfactual
(s) f (vanilla)	86.79	79.23
(s) f+cf	88.10	91.43
(s) f+cf+a	87.50	95.77
(m) f+cf	87.70	89.82
(m) f+a	87.30	79.03
(m) f+cf+a	86.19	96.37

Table 14: Accuracy for contextual answer on the test set without tabular contexts (73% of the data did not include tables)

	Factual	Counterfactual	Empty	Random
(s) cb	-	-	25.40	-
(m) f+cf	87.70	6.65	17.34	13.91
(m) f+a	87.30	0.71	22.78	23.89
(m) f+cf+a	81.96	44.86	28.53	30.95

Table 15: Accuracy for parametric answer on the test set without tabular contexts (73% of the data did not include tables)

	Factual	Counterfactual	Empty	Random
(s) cb (T5-11B)	68.35	18.68	27.69	25.20
(s) cb (T5-Large)	61.83	6.667	10.26	9.963

Table 16: Accuracy (in percent) for the closed book baseline, that was not trained to answer questions using a context, as opposed to the other models

	Factual ↑ (diff ↓)	Counterfactual ↑ (diff ↓)
(s) f (vanilla)	78.11 (1.23)	69.82 (-3.01)
(s) f+cf	79.88 (0.85)	82.25 (-2.62)
(s) f+cf+a	76.63 (1.69)	86.98 (-2.00)
(m) f+cf	77.51 (2.86)	79.59 (-2.67)
(m) f+a	78.99 (1.23)	70.12 (-5.5)
(m) f+cf+a	74.85 (3.25)	87.28 (-2.37)

Table 17: Contextual Answer accuracy predicted on the No Answer Overlap (NAO) Dev set. In brackets, difference from total accuracy reported on the Dev set (Answer overlap + No Answer Overlap).

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations section in Appendix A

☑ A2. Did you discuss any potential risks of your work?
Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, section 1

☒ A4. Have you used AI writing assistants when working on this paper?
-

B ☑ Did you use or create scientific artifacts?

2.2, 3.1

☑ B1. Did you cite the creators of artifacts you used?
2.2, 3.1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We do not share any artifacts so the license is irrelevant. (We did train a model, so we created the artifact, we just don't put it anywhere for future use and will probably delete it after the paper is done)

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We use standard data approved by our legal department. We follow the lisence of the framework and data we use in the paper but since it's very standard we didn't see a reason to discuss this in the paper.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We use a standard benchmark in the NLP community, Natural Questions. Other than that we don't collect data ourselves.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We do not share any new artifacts.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3 (Experimental Setup), Table 2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 3, Appendix B

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix B

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix B. We didn't perform hyperparameter search. Model selection is discussed in 5.5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3, 4 and Appendix C

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix B

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.