Enhancing Self-Consistency and Performance of Pre-Trained Language Models through Natural Language Inference

Abstract

While large pre-trained language models are powerful, their predictions often lack logical consistency across test inputs. For example, a state-of-the-art Macaw question-answering (QA) model answers Yes to Is a sparrow a bird? and Does a bird have feet? but answers No to Does a sparrow have feet?. To address this failure mode, we propose a framework, Consistency Correction through Relation Detection, or ConCoRD, for boosting the consistency and accuracy of pre-trained NLP models using pre-trained natural language inference (NLI) models without fine-tuning or re-training. Given a batch of test inputs, ConCoRD samples several candidate outputs for each input and instantiates a factor graph that accounts for both the model's belief about the likelihood of each answer choice in isolation and the NLI model's beliefs about pair-wise answer choice compatibility. We show that a weighted MaxSAT solver can efficiently compute high-quality answer choices under this factor graph, improving over the raw model's predictions. Our experiments demonstrate that ConCoRD consistently boosts accuracy and consistency of off-the-shelf closed-book QA and VQA models using off-the-shelf NLI models, notably increasing accuracy of LXMERT on ConVQA by 5

1 Introduction

Reliable and trustworthy AI systems should demonstrate internal self-consistency, in the sense that their predictions across inputs should imply logically compatible beliefs about the world. However, even powerful large language models are known to lack self-consistency. For example, a question-answering (QA) model that answers the question Is a sparrow a bird? and Does a bird have feet? with Yes is implicitly expressing the belief that A sparrow is a bird and A bird has feet. If the same model answers the question Does a sparrow have feet? with No, the model expresses the logically incompatible belief A sparrow does not have feet. In such cases, ascertaining the model's ˘201ctrue˘201d belief is difficult, making interpreting and validating its behavior correspondingly challenging.

Prior work has improved model self-consistency by training with specialized loss functions or data augmentation, or alternatively re-ranking model predictions based on their mutual self-consistency using pre-written logical constraints, such as ˘201call mammals have fur˘201d. However, the first class of methods requires expensive fine-tuning which might be impractical for many practitioners for very large pre-trained models, and re-ranking methods require an explicit collection of the logical relations of interest, making scaling a challenge. Still, re-ranking-based approaches have the benefit of not requiring fine-tuning, and we hypothesize that their scalability limitations may be addressed by estimating logical relationships between model predictions on the fly. Specifically, we hypothesize that existing pre-trained natural language inference (NLI) models can estimate logical relationships between an arbitrary pair of model predictions well enough to provide an effective, scalable substitute for explicit collection of such constraints. Leveraging these estimated constraints, we can construct a factor graph representing a probability distribution over model outputs that incorporates both the original model's confidence scores and the NLI model's beliefs about logical relationships.

Our primary contribution is Consistency Correction through Relation Detection, or ConCoRD, a framework to improve the consistency and performance of a pre-trained base language model without fine-tuning by using more confident and better attested model predictions to override less confident model beliefs. To enable propagation of model beliefs, we estimate pair-wise logical relationships between model predictions using a pre-trained NLI model. Using these pair-wise relationships, we define an undirected graphical model representing a distribution over responses accounting for both the base model's beliefs and the NLI model's estimates of answer compatibility. We efficiently find the approximate mode of this distribution among the base model's top answer choices for each input as the solution of a MaxSAT problem, which consistently produces more accurate and self-consistent predictions than using the raw model predictions. We find that ConCoRD produces an 8.1

2 Related Work

Prior work for maintaining consistency in the question-answering space often involves additional training to improve performance. Some work generates questions from unlabeled texts, then filters them to ensure roundtrip consistency; pre-training on this synthetic set improves performance on SQuAD 2.0 and Natural Questions. Other work augments QA-pairs with their logically symmetric and transitive counterparts through linguistic approaches to enhance cross-dataset QA performance. ConCoRD differs significantly from these question-answering-specific approaches because no fine-tuning of the base model is needed and the methodology is not specific to question-answering.

Similarly to ConCoRD, other work re-rank model predictions by solving an optimization problem defined by a combination of the base model confidence scores and pair-wise constraints representing the logical compatibility of different model predictions stored in a persistent memory, which they call BeliefBank. The key distinguishing property of ConCoRD is the fact that pair-wise constraints between model predictions are dynamically estimated by a pre-trained NLI model, rather than drawn from a fixed, pre-collected set of constraints. Dynamically estimating the constraints has a variety of benefits, eliminating the need for manually collecting the logical constraints of interest, automating the process of determining whether a particular constraint applies to a particular pair of predictions, and likely inheriting improvements in Natural language inference (NLI) models over time.

NLI has long been used to maintain logical consistency in generated dialogue utterances, radiology report domain entities, and summarization. Perhaps most similarly, other work uses NLI to estimate constraints between factual statements produced by GPT-3. These prior approaches support our intuition for using NLI models to improve logical consistency among batches of answers. While the authors explore applications of this framework to multi-step reasoning for True/False questions or statements, our work focuses on applying this methodology to more general settings, such as VQA, open-ended QA, and model editing.

3 Consistency Correction through Relation Detection

ConCoRD contains three key components, the base model, a relation model (typically a pre-trained NLI model), and an inference procedure that combines the predictions of the two models into a more accurate and self-consistent set of beliefs. Importantly, both the base model and relation model are pre-trained, off-the-shelf models; ConCoRD does not update any weights or require training data for either model, using only a small validation set for hyperparameter tuning. We next explain the function of each of these components when executing ConCoRD.

3.1 Base Model

The core function of the base model in ConCoRD is generating a set of candidate outputs for a given input, which are ultimately re-ranked by the inference process (Sec. 3.3). Given a batch of N model queries $Q = \{q_i\}$, the first step of ConCoRD is to generate a set of J candidate outputs for each query $\hat{A}_i = \{\hat{a}_{i1}, ..., \hat{a}_{iJ}\}$, along with their corresponding likelihoods $p_\theta(\hat{a}_{ij}|q_i)$. Note that the candidate outputs need not be an IID sample from the base model; for example, we might use beam search with a diversity bonus to produce a more diverse set of candidates. Each pair of query and candidate output forms a model belief $b_{ij} = (q_i, \hat{a}_{ij})$; the output of the base model is the complete set of model beliefs $B = \{b_{ij}\}$ and their corresponding normalized probabilities p_{ij}. The base models in our experiments are pre-trained question-answering models based on T5-large and pre-trained visual question-answering models such as LXMERT and ViLT.

3.2 Relation Model

The relation model $p_\theta(: | x_i, x')$ estimates the most likely logical relationship between an ordered pair of natural language utterances from the choices $\{none, fwd-entail, contradict, equivalence\}$. In addition to the model beliefs B, we define optional context statements $c_{ijk} = C(b_{ij})$, K relevant statements that may be retrieved, generated, or manually written for each model belief. The ability to incorporate context statements enables ConCoRD to modulate model behavior independently for each input in the test batch, rather than reasoning transductively about pairs of test inputs. Inputs to the relation model are either pairs of two model beliefs $(b_{ij}, b_{i'j'})$ or pairs of one model belief and one context statement (b_{ij}, c_{ijk}). We define the most likely inter-belief relation as $r_{ij,i'j'} = argmax_r p_\theta(r | b_{ij}, b_{i'j'})$, and similarly for belief-context relations $r_{ij,k} = argmax_r p_\theta(r | b_{ij}, c_{ijk})$. The output of the relation model is the set of most-likely relations $R = \{r_{ij,i'j'}\} \cup \{r_{ij,k}\}$ and their associated probabilities, which we denote as $p_\phi^{ij,i'j'}$ and $p_\phi^{ij,k}$. Our experiments use various pre-trained NLI models based on RoBERTa and ALBERT as the relation model.

Question-answer to statement conversion. While concatenating query q_i and candidate output $\hat{a}_{ij}$ to produce inputs to the relation model is perhaps the simplest approach to estimating soft constraints, we use a statement conversion model to provide inputs to the relation model that are closer to its training distribution. Instead of defining the belief $b_{ij} = (q_i, \hat{a}_{ij})$ as concatenation of q_i and $\hat{a}_{ij}$, we define b_{ij} to be the statement $f_\phi(q_i, \hat{a}_{ij})$, where f_ϕ is the conversion model. We fine-tune a small T5 model on a combination of data from and BeliefBank to produce a model that maps a (question, answer) pair into a natural language statement.

3.3 Inference

ConCoRD's inference procedure maps the set of beliefs B and pair-wise relations R into a choice of the most likely belief for each question. To define the inference problem, we first define a binary decision variable z_{ij} representing the estimated truth value of model belief b_{ij}. A value of 1 for node z_{ij} in the maximum likelihood configuration means that $\hat{a}_{ij}$ is returned for query q_i; the problem includes a constraint that exactly one candidate answer is true for each query. The factor graph includes the set of variables $Z = \{z_{ij}\}_{i,j=1,1}^{N,J}$ and various factors (functions mapping a subset of Z to a non-negative scalar) derived from the base model and relation model's beliefs and the hard constraint of returning only one answer per question. Factors are defined such that more desirable configurations of z_{ij} yield a larger product of the individual factors. First, unary factors $\phi_{ij}(z_{ij})$ encode the base model's beliefs about the likelihood of specific answers, and are defined as:

$$\phi_{ij}(z_{ij}) = \{ p_{ij} \, if z_{ij} = 11 - p_{ij} otherwise \tag{1}$$

where $p_{ij} = p_\theta(\hat{a}_{ij} | q_i)$; in other words, the factor takes the odds ratio if the corresponding statement variable z_{ij} is assigned a truth value of 1; otherwise, the factor takes value 1. In order to encode the hard constraint that exactly one output should be returned for each query, we include a J-ary factor $\phi_i(Z_i)$ for each group of nodes $Z_i = \{z_{ij}\}_{j=1}^{J}$, which is equal to 1 for configurations where exactly one of the nodes takes a value of 1, and 0 for all other configurations.

Binary factors $\psi_{ij,i'j'}(z_{ij}, z_{i'j'})$ and optionally $\psi_{ijk}(z_{ij}, c_{ijk})$ encode compatibility between pairs of model beliefs (or model belief-context pairs):

$$\psi_{ij,i'j'}(z_{ij}, z_{i'j'}) = \{ 1 \; if r_{ij,i'j'}(z_{ij}, z_{i'j'}) p_\phi^{ij,i'j'} otherwise \tag{2}$$

where we define the relation function $r_{ij,i'j'}$ to evaluate to true if its arguments satisfy the underlying relation, and false otherwise; $\psi_{ijk}(z_{ij}, c_{ijk})$ is defined similarly to $\psi_{ij,i'j'}(z_{ij}, z_{i'j'})$. The inference problem amounts to finding $argmax_Z \Phi(Z)$, where

$$\Phi(Z) = \prod_i \phi_i \prod_{ij} \phi_{ij} \prod_{ij,i'j'} \psi_{ij,i'j'} \prod_{ijk} \psi_{ijk} \tag{3}$$

An approximate solution to this inference problem can be efficiently found for most problems with a MaxSAT solver such as RC2. We omit arguments to the factors for conciseness.

Entailment correction. Consider a belief b, a set of its entailed statements $S = \{s_i\}$, unary factors $\phi(z_b)$ and $\{\phi(z_{s_i})\}$, and binary factors $\Psi = \{\psi(z_b, z_{s_i})\}_i$. Recall that an entailment relation $r_{ij,i'j'}(z_{ij}, z_{i'j'})$ is satisfied (and the binary factor is maximized) if either $z_b = 0$ or all $z_{s_i} = 1$. Consequently, as the cardinality of $\{z_s | z_{s_i} = 0\}$ increases, the more likely it is that $z_b = 0$ will maximize the product of all binary factors $\prod_i \psi(z_b, z_{s_i})$. This is true even if most entailed statements are true, ie., $|\{z_s | z_{s_i} = 1\}| > |\{z_s | z_{s_i} = 0\}|$. If most of the statements entailed by a belief are true, assigning the belief to be false due to a small number of (potentially spuriously) false entailed statements may be undesirable. To mitigate this outcome, we experiment with an additional type of factor in which configurations satisfying entailments with both $z_b = 1$ and $z_{s_i} = 1$ are 'rewarded' more than other configurations satisfying the entailment:

$$\Psi_{b,s_i}(z_b, z_{s_i}) = \{ 1 \; if z_b, z_{s_i} = 11 - p_\phi^{b,s_i} if z_b, z_{s_i} = 0\sqrt{1 - p_\phi^{b,s_i}} otherwise \tag{4}$$

Applying entailment correction consistently improves ConCoRD's performance.

3.4 Hyperparameters of ConCoRD

We introduce two key hyperparameters to ConCoRD. Because we do not know a priori the relative reliability of the base model and relation model, we introduce the hyperparameter $\delta \in [0, 1]$, corresponding to a trade-off between the predictions of the base model and relation model. A value of $\delta = 1$ corresponds to simply taking the raw predictions of the base model, while $\delta = 0$ corresponds to optimizing purely for answers that are self-consistent according to the relation model, without considering the base model's beliefs. The unary factors in the factor graph become $\phi_i(z_i) = (\phi_{ij}(z_{ij}))^\delta$ and $\psi_{ij,i'j'}(z_{ij}, z_{i'j'}) = (\psi_{ij,i'j'}(z_{ij}, z_{i'j'}))^{1-\delta}$ (and similarly for ψ_{ijk}). In addition to δ, we introduce a threshold λ for relation model confidence to filter out low-confidence relation estimates. That is, we discard a relation $r_{ij,i'j'}$ or $r_{ij,k}$ if $p_\phi^{ij,i'j'} < \lambda$ or $p_\phi^{ij,k} < \lambda$, respectively. In practice, we find that the optimal δ and λ vary across problems, perhaps due to the varying complexity of the model belief and context statements (and therefore the reliability of the relation model's predictions). Therefore, we use the hyperopt library for automated hyperparameter optimization, using the Tree Parzen Estimator (TPE) algorithm to tune δ and λ jointly. We use the optimal hyperparameters found on the validation data for each problem to compute test performance.

4 Experiments

Our experiments are broadly designed to answer the high-level question: can ConCoRD leverage the relational knowledge in pre-trained NLI models to produce more accurate, self-consistent system behavior, without additional data or fine-tuning? Further, we investigate ConCoRD's applicability to performing test-time model editing, or injection of new information, and ConCoRD's sensitivity to the choice of hyperparameters and types of relations detected.

4.1 Internal Consistency in Closed-Book Question-Answering

Protocol. To evaluate the accuracy and consistency of a set B of beliefs, we synthesize a gold standard for those beliefs and the inferred relations R. Following this prior work, we assume the following is given:

- A set of entities $s_m \in S$
- A set of unary predicates $P_n \in P$
- A collection of 2̆01cfacts2̆01d $(P_n(s_m))_i$, whose binary truth value is known
- A directed graph of gold-standard constraints $G(P, E)$, whose edges $(P_i, P_j) \in E$ represent first-order logical formulae

From these, we construct simple yes/no questions using natural language templates. For example, for fact $P_n(s_m)$, if entity s_m represents a lion and predicate P_n represents an ability to drink liquids, the template-generated gold question answer pair (q_i, a_i) is Q: Is it true that a lion is able to drink liquids?; A: Yes.

We evaluate ConCoRD by sampling candidate answers from the top-2 output sizes of a multi-angle question answering model, given a multiple choice angle with choices Yes and No. The questions and retrieved answers $(q_i, \hat{a}_i)$ form a set of beliefs B_{sm} for each entity. Since these are closed-book questions, no context statements are supplied; because they are yes/no questions, only one candidate answer is obtained, i.e., $J = 1$. Question-answer to statement conversion is applied to all questions with a default answer of Yes regardless of the answer $\hat{a}_i$, in order to provide the relation model with positive natural language assertions from which to infer sets of relations R_{sm}; where the base model answers $\hat{a}_i$ are No we replace node z_i in the factor graph with its complement. Configurations Z_{sm} are found for each $s_m \in S$ which maximize Equation 2 given B_{sm}, R_{sm} and together form a global solution Z.

Datasets. We use a database with 12,636 facts (2̌01csilver facts2̌01d), each indicating whether one of 601 predicates relates to one of 85 entities, as well as 4,060 confidence-weighted first-order constraints manually gathered from ConceptNet, forming a constraint graph G. Additionally, they provide 1,072 distinct 2̌01ccalibration facts2̌01d, each relating one of 7 entities to one of 334 predicates.

We tune β and λ using a validation set of questions generated from the calibration facts, and evaluate test time performance with questions generated from silver facts.

Metrics. We measure accuracy using binary F1 between elements z_i of the configuration Z maximizing $\phi(Z)$ (as in Equation 2), and the truth value of facts $(P_n(s_m))_i$. We use F1 for evaluation because gold answers are highly biased towards true No answers.

We compute consistency within batches of questions using the complement of of conditional constraint violation metric τ, defined here as the proportion of relevant gold constraints in G which are violated; a constraint $\forall(P_i(x) \to P_j(x))$ is relevant iff, for some entity s, there is some belief $b_i \in B, s_m$ from fact $(P_i(s_m))_i$ such that $z_i = 1$, and there is some belief $b_j \in B_{sm}$ that corresponds to fact $(P_j(s_m))_j$; the constraint is violated when $z_j = 0$.

Comparisons. ConCoRD is evaluated against a naive baseline where only base model answers $\hat{a}_i$ and probabilities are considered. A second baseline (G.C.) performs the inference described in Sec. 3.3, replacing the inferred relations R with the gold constraints from constraint graph G, rather than those estimated by the relation model.

Results. Results are shown in Table 1. ConCoRD provides an absolute improvement of over 8% in F1 and consistency for Macaw-Large and 7% for Macaw-3B compared to the baseline. Notably, the margin of superiority of the Macaw-3B base model is mostly preserved after applying ConCoRD, suggesting that ConCoRD may provide a significant benefit even for very large models. A surprising result is that ConCoRD shows marked improvements in F1 over the gold constraint baseline, suggesting that the detection and filtering of relations ConCoRD provides may, in this setting, be an improvement over rigid adherence to the logical connections specified a priori.

Table 1: F1 and consistency (1 - τ) for two sizes of Macaw QA models, comparing ConCoRD to a naive QA baseline (Base) and ConCoRD with gold constraints (G.C.). ConCoRD significantly improves both F1 and consistency for both models.

2*Model	Base		ConCoRD		G.C	
	F1	Con.	F1	Con	F1	Con
Mac-Lg	0.831	0.835	0.914	0.920	0.862	0.934
Mac-3B	0.855	0.871	0.931	0.947	0.905	0.936

4.2 Internal Consistency in VQA

Protocol. The Visual Question Answering (VQA) task involves a language model generating answers to questions that are directly associated with images. VQA tests for robustness and generalizability of ConCoRD as it introduces an additional layer of difficulty; the task moves away from purely text-based tasks while expanding the answer space to the vocabulary of the LM being used. The questions from the ConVQA dataset and its associated images from the Visual Genome dataset provide an apt setting to assess ConCoRD, as the relatedness of questions for each image provide ample opportunity for model self-inconsistency.

The ConVQA dataset consists of a set of images each associated with a group of related questions about the image, such as What color is the horse? and Is the horse brown? for a picture of a brown horse in a stable. We evaluate ConCoRD with two VQA models, LXMERT and ViLT. For each group of questions $Q_n = \{q_{ni}\}_i$, we sample the top-2 candidate outputs $\{\hat{a}_{ni1}, \hat{a}_{ni2}\}$ for each question, and use a pre-trained NLI model to infer the most likely pair-wise relations R between outputs from different questions. We use the RC2 MaxSAT Solver to estimate the configuration that maximizes Equation 2.

Metrics. We report accuracy as the proportion of questions answered correctly across all groups. We infer consistency using a metric previously used in the literature for the ConVQA dataset called 2̆01cperfect consistency2̆01d. For all groups of related questions, a group is perfectly consistent if all its questions are answered correctly. Perfect consistency then reports the proportion of question groups that were perfectly consistent. While this is not a perfect measure of consistency as it excludes cases in which incorrect answers are consistent with each other, it still serves as a meaningful proxy since the dataset was designed such that any incorrect answer in a question group implies the presence of inconsistency.

Datasets. We divide the ConVQA dataset into a 2̆01cclean2̆01d (i.e. human verified and filtered) test set and a non-test set (train + val + test as defined by previous work). From the non-test set, we sample 10,000 random images equivalent to 123,746 questions to be used as our validation set for tuning our two hyperparameters. We use the clean test set 2̆013 725 images and 6,751 questions 2̆013 to report our final results.

Comparisons. ConCoRD is compared with a naive baseline and a top-2 oracle upper bound. The naive baseline is the answer with the highest VQA model probability. Top-2 oracle upper bound selects the correct answer if present within the top-2 predictions of the VQA model. Top-2 is appropriate given our use of the top-2 candidate outputs to generate inferences with NLI models.

Results. The final results for ConCoRD, baseline, and oracle upper bound are shown in Table 2. ConCoRD increases the accuracy of LXMERT and ViLT by 5% and 2% respectively, and the consistency of LXMERT and ViLT by 4.9% and 5.9% respectively.

Table 2: ConVQA accuracy (Acc.) and perfect consistency (P.C.) of LXMERT and ViLT VQA models with and without ConCoRD. ConCoRD significantly improves accuracy and consistency of both models. Oracle performance is top-2 performance, as ConCoRD attempts to select the best of the top 2 answer choices of the base model.

2*Model	Base		ConCoRD		Oracle	
	Acc.	P.C.	Acc.	P.C.	Acc.	P.C.
LXM	0.656	0.360	0.706	0.409	0.824	0.572
ViLT	0.784	0.489	0.804	0.548	0.882	0.690

4.3 Test-Time Information Injection

Protocol. We perform an additional experiment to evaluate ConCoRD's ability to integrate external factual information into its inference process, rather than only using other predictions in the test batch. Such an ability enables editing a model's behavior at test time, without re-training, as new information becomes available. We use the Natural Questions (NQ) dataset, rather than BeliefBank, to provide more challenging inputs to the relation model. Given a question from NQ, a sentence from the ground truth context document containing information about the answer is retrieved and provided as an additional input to ConCoRD; we constrain the node representing this context variable in the factor graph to be true. Constraints are predicted between each answer choice and the context statement. As in the other experimental settings, hyperparameters are tuned on the validation set and applied on the test set.

Metrics. Model performance is evaluated using the SQuAD F1 score for overlapping tokens, following the same answer normalization protocols, including lower-casing and removing punctuation.

Datasets. The NQ development set consists of 7830 open-book question-answer pairs, with both long and short gold annotations in their context passages. Since the NQ test set is not available, we create a test and validation set from the NQ validation questions as follows: we take the first 5000 questions to form our test set, and the rest to be our val set, which we use for hyperparameter tuning. Then each set is filtered such that only the answerable questions remain. 2̆01cAnswerable2̆01d is defined as having a 2̆01cshort answers̆pan defined in the annotations. This filtering process gives 2713 test entries and 1576 val entries.

Comparisons. ConCoRD is compared with a naive baseline and an oracle upper bound. All of these approaches operate on the fixed set of QA model answers for a specific QA model (one of T5-Sm-NQ, T5-Lg-NQ, and T5-3B-NQ), specifically the set of top-4 answers for each question. The naive baseline selects the answer with the highest QA model probability, $argmax_{\hat{a}_{ij}} p_\theta(\hat{a}_{ij}|q_i)$. The oracle upper bound approach selects the answer that has the best score with the gold short answer span, $argmax_{\hat{a}_{ij}} F1(\hat{a}_{ij}, a_{ij})$.

Results. The results on the test set using the naive baseline, ConCoRD, and oracle upper-bound are reported in Table 4. ConCoRD always outperforms the naive approach, demonstrating that the framework is useful even when each query input is processed independently (i.e., non-transductively). However, despite providing a relative gain of as high as 8.7% over the naive baseline, there is still a gap between ConCoRD and the oracle. This gap may be attributable to the complexity of the NQ questions and context information compared with the statements in prior experimental settings. Other work demonstrates a significant gain in calibration performance from training on MultiNLI to training on a combination of MultiNLI and their NLI corpus adapted from NQ, perhaps hinting that crucial knowledge present in Natural Questions is not covered in MultiNLI, partially explaining the gap between ConCoRD and oracle F1 performance. Overall, these results suggest that ConCoRD can reason between context statements and model beliefs in addition to pairs of model beliefs, improving performance even with the increased complexity of the data.

Table 3: Using ConCoRD to inject contextual information into a model's decisions at test time. Injecting gold Natural Questions contexts consistently improves performance over the base model without requiring fine-tuning.

2*Model	F1		
	Base	ConCoRD	Oracle
T5-Sm-NQ	0.207	0.225	0.281
T5-Lg-NQ	0.314	0.328	0.393
T5-3B-NQ	0.332	0.351	0.423

4.4 Ablating Relation Types

Given that we consider two types of relations in our experiments, contradiction and entailment, it is natural to wonder the relative contribution of these to ConCoRD's performance improvement; Table 5 shows the results of this ablation. We re-run ConCoRD with either entailment or contradiction relations removed, re-tuning the hyperparameters for both of the new settings (contradiction-only or entailment-only). We find that the relative contribution of contradiction and entailment relations varies significantly across models even within the same task, but using both relation types always performs approximately as well or better than using just one, suggesting that both types of detected relations from the NLI model carry useful information. However, we observe in several cases, such as ViLT and the T5 models, that the entailment and contradiction relations may encode somewhat redundant information, as the performance when including either type of constraint alone nearly matches that of using both types.

5 Conclusion

This paper presents a novel method, ConCoRD, for enhancing the self-consistency and performance of pre-trained language models without requiring fine-tuning. ConCoRD leverages pre-trained NLI models to estimate logical relationships between model predictions and uses a MaxSAT solver to enforce consistency. The experimental results demonstrate that ConCoRD improves over off-the-shelf

Table 4: Ablating the relation types considered in ConCoRD2̌019s inference procedure. The Only cont. and Only ent. are the results of applying ConCoRD with all entailment or con- tradiction relations removed, respectively. The ConCoRD column is a reproduction of the results from Sections 4.1-4.3, for convenience. Value shown is F1 score for BeliefBank (BB) and Natural Questions (NQ) and accuracy for ConVQA (CVQA). Note that hyperparameters 0̆3b2 and 0̆3bb are re-tuned on the respective validation set for each setting.

Table 5: Comparing ConCoRD2̌019s performance for various NLI models on BB (BeliefBank), ConVQA, and NQ. Performance is measured as F1 score between predicted and gold text for BB and NQ, exact match accuracy for ConVQA. We use Macaw 3B for BB results, LXMERT for VQA results and T5-3B for NQ results. The best NLI model(s) in each column are bolded; the best NLI model varies across problems.

NLI Model	Data	F1/Accuracy	
		BB	ConVQA
Alb-XXL	ANLI	0.892	0.689
RoB-Lg	ANLI	0.931	0.706
RoB-Lg	MNLI	0.918	0.706

performance in a variety of settings and that it is relatively robust to the choice of hyperparameters. The paper also discusses potential future directions, such as integrating ConCoRD with other methods and exploring its applications beyond natural language processing.

Table 6: The QA statement conversion model outputs declarative statements from question-answer pairs. Out of the four validation examples presented, three are correct. The Red, bolded portion of the output of the second example indicates how it differs from the Teal, bolded corresponding portion of the gold statement.

Dataset	Input	Output
SQuAD	Who established Yale’s residen- tial college system? Edward S. Harkness	Edward S. Harkn
SQuAD	How did Kuhn view the his- tory of science? competing paradigms or conceptual sys- tems	Kuhn viewed the
BeliefBank	Is it true that a poodle is a river? No	A poodle is not a
BeliefBank	Is a pigeon a living thing? Yes	A pigeon is a liv

Table 7: Comparison of ConCoRD test performance vs. base- line with and without entailment correction (E.C.) across base+relation models for closed-book question answering (Macaw) and VQA (LXMERT, ViLT) experiments (F1 for closed-book QA, exact-match accuracy for VQA), showing that the entailment correction improves performance for most con01gurations.

	F1/Accuracy	
Mac-Lg+Rob/ANLI 0.831	0.914	0.909
Mac-3B+Rob/ANLI 0.855	0.931	0.886
LXMERT+Rob/MNLI 0.656	0.706	0.701
LXMERT+Rob/ANLI 0.656	0.706	0.693
ViLT+Rob/MNLI 0.784	0.804	0.810
ViLT+Rob/ANLI 0.784	0.814	0.807

Table 8: The numbers of good and bad flips in each of the experiments performed. We define flips as choosing a different candidate from the naive baseline for the multiple choice experiments, and a binary truth value flip for BeliefBank. "Good" flips are flips that improve performance, and "bad" flips are those that are detrimental to performance.

Experiment	Model	Good Flips	Bad Flips
BeliefBank	Macaw-3B	723	277
VQA	LXMERT	576	238
NQ	T5-3B-NQ	168	69

Table 9: Editing a model's behavior by adding new information to the context. The underlined generation is the answer with the highest QA model confidence. The bolded generation is what ConCoRD selects after NLI inference. Teal, bolded generations indicate that ConCoRD selects a generation with higher token overlap F1, while red, bolded generations indicate that ConCoRD selects a worse generation.

!

Model	**Input & Gold Answer**	**Generations**	**Added Context**
T5-Sm-NQ	Q: Who was the declaration of independence written for? A: the Second Continental Congress	Second Continental Congress; the United States; the British Crown; Great Britain	The United States Declaration of Independence is the statement adopted by the Second Continental Congress meeting at the Pennsylvania State House (Independence Hall) in Philadelphia on July 4, 1776, which announced that the thirteen American colonies, then at war with the Kingdom of Great Britain, regarded themselves as thirteen independent sovereign states, no longer under British rule.
T5-Sm-NQ	Q: What is the scientific name for the calf muscle? A: gastrocnemius muscle	The serratus f muscle; muscle; gastrocnemius; The serratus calfi; The serratus muscle	Along with the soleus muscle, the gastrocnemius forms half of the calf muscle.
T5-3B-NQ	Q: Who is the actor that plays Dr. Sean Murphy? A: Freddie Highmore	Freddie Highmore; Daryl "Chill" Mitchell; Dylan Christopher Minnette; Javier Muoz	The series stars Freddie Highmore as Shaun Murphy, a young surgical resident with autism and savant syndrome at San Jose St. Bonaventure Hospital. Freddie Highmore as Shaun Murphy: A surgical resident with autism and savant syndrome.
T5-3B-NQ	Q: Who is the founder of the Ubuntu project? A: Mark Richard Shuttleworth	Linus Torvalds; Mark Shuttleworth; Richard St. John Hopper; Richard St John Redmond	Mark Richard Shuttleworth (born 18 September 1973) is a South African entrepreneur who is the founder and CEO of Canonical Ltd., the company behind the development of the Linux-based Ubuntu operating system.

Table 10: Validation performance on the BeliefBank cal- ibration facts. Both models achieve best validation per- formance with the RoBERTa-Large ANLI model.

Model	F1	0̆3b2	0̆3bb	E.C.
Macaw-Large	0.919	0.753	0.855	True
Macaw-3B	0.94	0.804	0.873	True

Table 11: Validation performance on VQA. Both models achieve best validation performance with the RoBERTa-Large MNLI model.

VQA	Acc.	0̆3b2	0̆3bb	E.C
LXMERT	0.691	0.208	0.805	True
ViLT	0.787	0.395	0.772	True

Table 12: Validation performance on NQ. All models achieve best validation performance with the ALBERT ANLI model.

Model	F1	ŏ3b2	ŏ3bb	E.C.
T5-Small	0.227	0.112	0.540	True
T5-Large	0.331	0.081	0.413	False
T5-3B	0.353	0.072	0.477	True