Using Natural Language Inference to Correct Question-Answering Inconsistencies

Kevin Yang, Patrick Liu, Ian Ng

Department of Computer Science, Stanford University

Introduction

In this project, we aim to improve language model consistency in binary question-answering tasks by identifying and flipping contradictory statements via natural language inference and a variety of correction algorithms. To do so, we apply a pretrained Question Answering (QA) model to judge whether or not statements about a chosen entity are true. We then apply a Natural Language Inference (NLI) model to the produced statements in pairwise fashion, determining the probability of QA judgments being contradictory, entailed, or consistent with one another.

Due to the QA model's incomplete knowledge base or its inability to interpret the given statements, the QA model generates contradictory judgements $\approx 20\%$ of the time. Our goal is therefore to use the outputs and confidences of the QA model, as well as the contradiction probabilities output by the NLI model, to determine which judgments should be negated to minimize contradictions and thus improve consistency. We implement several different families of algorithms to determine which judgments to flip.

Background: Data, Sampling and Models

- **Entities, facts, and constraints** are drawn from the BeliefBank memory bank. The original BeliefBank provides 85 entities, 12,525 facts, and 2,225 constraints of the form "statement A implies statement B". We augment these constraints according to the following logical expression:

$$A \to B \land B \to C \Rightarrow A \to C$$

  Repeatedly applying this chain logic, we can augment over the course of about 10 iterations to 15,277 constraints.
- The **dev set** contains 60 entities and corresponding facts/constraints drawn from Beliefbank. The **test set** contains the other 25 entities.
- **Statements** are sentences of the form "[*entity*] [*connector*] [*descriptor*]".
- The **QA Model** takes in statements, converted to the form of a binary (yes-no) question. It outputs its answer to the question, as well as its confidence in the provided answer.
- The **NLI model** takes in a list of statements of the above form. For each pair of statements (a, b), it provides a triple (x, y, z) of probabilities, where x is the probability that a implies, or entails, b; y is the probability that a and b are unrelated, and z is the probability that a and b contradict.

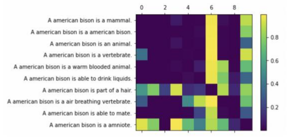


Figure 1. NLI contradiction matrix representing pairwise contradiction probabilities between 10 example sentences.

Pipeline and Methods

For our experiments, we follow the following steps:

1. For 10,000 batches, select an entity (e.g. "computer") with a list of ten associated facts per batch (e.g. "A computer is a machine.").
2. Convert each fact to a yes/no question and query a pre-trained Macaw QA model on each question. Convert the QA model's outputs to assertions.
3. Use a RoBERTa-based NLI model, pretrained on determining consistencies between statements, to calculate a 10x10 **matrix of contradiction scores** representing how likely any pair of assertions is to be entailing or contradictory.
4. Using the values in the matrix and the QA confidence values, identify and correct incorrect statements (described in detail in Experiments).

Experiments

To detect contradictions, we apply four classes of correction algorithms:

1. **Constraint satisfiability problems:** In this approach, we construct a boolean MaxSAT (maximum satisfiability) problem by creating a weighted graph of constraints, based on the NLI and QA model outputs. We have two forms of soft constraints:
   - Unary constraints, with the constraint's weight being the statement's confidence from the QA model's output.
   - Binary constraints, with the constraint's weight generated from the confidence statement pairs are contradictory, based on the NLI model.

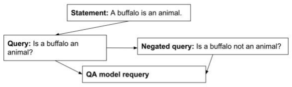
   We then utilize the RC2 MaxSAT solver to find the optimal configuration that maximizes constraint weights.
2. **Contradiction-based probabilistic estimation (C1-C8):** In this family of approaches, we examine the NLI and QA outputs corresponding to each individual statement to determine the probability of a given statement being a contradiction based on its agreement with other statements.
   Example:

$$\text{C1}(a) = \frac{1}{\|S\| - 1} \sum_{b \in S \setminus a} N(a, b) Q(b)$$

3. **Entailment-based probabilistic estimation (C9-C10):** In this family of approaches, we utilize the NLI's *entailment probability*- the probability, predicted by the NLI model, that statement a entails statement b (note that this is a directional relation.)
4. **Requerying approach (C11-C12):** In this approach, we first utilize a contradiction-based probabilistic estimation. Once we have determined the targets for flipping using such a method, we then add an additional layer of filtering, or "requerying" the QA model; that is, reformulating the statement as various forms of a question and querying the QA model once again on the reformulated question. Intuition: The QA model's response and confidence is not necessarily robust to wording changes in a query given similar semantic meaning.

In addition to our correction methods, we perform several ablations, across:

- **QA model**: Macaw (Our standard QA model), UnifiedQA T5-small (performs worse than Macaw), Aristo Roberta V7 Model (performs better than Macaw)
- **Correction algorithm hyperparameters**: For algorithms with hard thresholds for entailment/contradiction or flipping constraints, we ablated across several threshold values.



Statement: A buffalo is an animal.

Query: Is a buffalo an animal?

Negated query: Is a buffalo not an animal?

QA model requery

Figure 2. Pipeline for C12.

Analysis

We can analyze with respect to each of the different classes of algorithms (MaxSAT, C1-C8, C9-10, C11-12). We note the following observations:

- Contradiction-based probabilistic estimation methods tend to perform approximately the same. The exception is C8; analyzing the number of flips reveals that C8 flips far more than necessary. This suggests that the algorithm for C8, which simply flips the lower confidence statement out of the pair, doesn't incorporate enough information to make an informed decision.
- C1-C5 perform the best out of the contradiction-based probabilistic estimation methods. This may be attributable to the fact that they incorporate all pairs involving a given statement as well as utilizing confidence weights, thereby using the most information.
- Requerying approach C11 (incorporating context into requerying) performs worse than the baseline, despite using more information than its counterpart C12.
- C12 has a simple approach of requerying the QA with the negation of the original question. Its simplicity and effectiveness is likely due to PTLMs' poor handling of negation.

Results

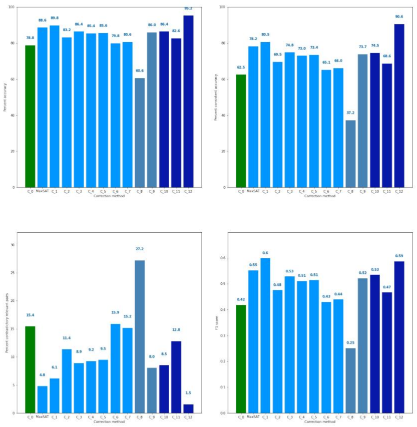


Figure 3. Comparison graphs for RoBERTa run, across all correction algorithms: from left to right, top to bottom: accuracy, consistent accuracy, relevant contradiction, and F1 score.

Conclusions

- Especially for poorly-performing QA models, using a correction algorithm with confidence-weighted contradiction scores can far outperform the QA model by itself.
- By requerying the QA model with both positive and negative statements, accuracy can be boosted even higher than vanilla probabilistic estimation methods.
- While our methods yield higher improvements over baseline for worse QA models, the relative performance of our methods is around the same.
- Future approaches to further improve consistency:
  - Explicit graph-based approaches
  - Potentially RL with prediction as state and correction as actions.
  - Experiments with high contradiction density batches, and with paraphrasing

References

Fabio Petroni, Tim Rocktäschel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases? CoRR, abs/1909.01066, 2019.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. Beliefbank: Adding memory to a pre-trained language model for a systematic notion of belief. CoRR, abs/2109.14723, 2021.

Oyvind Tafjord and Peter Clark. General-purpose question-answering with macaw. CoRR, abs/2109.02593, 2021

CS 224N Custom Final Project