Improving Logical Consistency in Pre-Trained Language Models using Natural Language Inference

Ananth Agarwal, Cameron Tew, Anthony Tzen

Problem

Current state-of-the-art pre-trained language models (PTLMs) contain rich and vast amounts of world knowledge, demonstrating an ability to extrapolate information from contextual texts and to accurately answer questions. However, the latent factual understanding captured by PTLMs can be irrational and incohesive, causing PTLMs to be prone to generating logically inconsistent statements.

Macaw, a PTLM built on T5, outputs the following inconsistent result:

- Q: Is a puppy a vertebrate? A: Yes
- Q: Is a vertebrate a crustacean? A: No
- Q: Is a puppy a crustacean? A: Yes

We aim to improve accuracy and logical consistency of PTLMs using natural language inference (NLI) and a heuristic function to revise contradictory PTLM answers within a batch of input questions.

Dataset

We are using the BeliefBank dataset curated by Kassner et al. to tune and evaluate our model. The dataset contains the following:

- **Constraint graph**: Directed graph derived from the ConceptNet semantic knowledge graph. Nodes are modeled as statements of the form (subject, relation, object), and edges capture directional implications between nodes.

- **Test silver facts**: 12,636 facts harvested from the constraint graph consisting of 95 animal and plant entities. Silver facts are represented as (subject, relation, object), where the subject and object are paired with a truth value.

- **Development silver facts**: Facts used to tune model hyperparameters.

We sample facts for each entity to create one batch of facts per entity. Dev batch size is 50, and test batch size is 100.

Metrics

\[
F_1 = \frac{TP}{TP + 0.5(TP + FN) + PN}
\]

\[
\text{Consistency} = \frac{1 - ||c \cap s_y \cap s_3||}{||c \cap s_y||}
\]

The denominator of consistency is the number of constraints with a true premise \(s_y\) contained in the batch. The numerator is the number of these constraints that are violated (where \(s_y \rightarrow s_3\) is false). Thus, consistency is defined as the complement of the fraction of all violated constraints.

Methodology

- For each question \(q_y\), a PTLM predicts an answer \(s_y\), with an associated prediction probability, \(P_{PTLM}(s_y)\).
- Each question-answer \((q_y, s_y)\) is translated into a statement \(y\). The prediction prob. of \(s_3\) would be the same: \(P_{PTLM}(s_3) = P_{PTLM}(s_y)\).
- For each ordered pair of statements \((s_y, s_3)\) where \(s_y\) is the hypothesis and \(s_3\) is the premise, a NLI model returns an entailment probability and a contradiction probability. We use various ways to estimate how the probability of \(s_3\) relates to the probability of \(s_y\):

  - Single Contrast Scenario
    - \(P_{d}(s_3) = P(s_3)\)
    - \(P_{d}(s_3) = P(s_y \land s_3)\)
    - \(P_{d}(s_3) = P(s_3)\)
    - \(P_{d}(s_3) = P(s_y \land \neg s_3)\)
    - \(P_{d}(s_3) = P(s_3)\)
    - \(P_{d}(s_3) = P(s_y \land s_3)\)

  - Maximum
    - \(P_{d}(s_3) = \max(P(s_y \land s_3), P(s_3))\)
    - \(P_{d}(s_3) = \frac{P(s_y \land s_3)}{P(s_y)}\)
    - \(P_{d}(s_3) = \frac{P(s_y \land s_3)}{P(s_y)}\)

  - Weighted Average
    - \(P_{d}(s_3) = \frac{1}{\sum_{i} P(s_y \land s_3)}\)
    - \(P_{d}(s_3) = \frac{1}{\sum_{i} P(s_y \land s_3)}\)
    - \(P_{d}(s_3) = \frac{1}{\sum_{i} P(s_y \land s_3)}\)

- To compute a final confidence score for \(s_3\), we balance the NLI estimates and the PTLM estimate of \(P(s_3)\):

  \[
  \text{score}(s_3) = \frac{0.5 \cdot P_{d}(s_3) + 0.5 \cdot (1 - P_{d}(s_3))}{1 - P_{d}(s_3)}
  \]

- To correct the original PTLM predictions, the statement with the lowest score is inverted (“flipped”) if it is under a minimum score. This is iteratively repeated, with the scores being updated after each flip.

Results and Analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>Min. Score</th>
<th>Max Flips</th>
<th>(\lambda)</th>
<th>F1</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.787</td>
<td>0.826</td>
</tr>
<tr>
<td>Max</td>
<td>0.573</td>
<td>9</td>
<td>0.422</td>
<td>0.807</td>
<td>0.836</td>
</tr>
<tr>
<td>Average</td>
<td>0.543</td>
<td>6</td>
<td>0.519</td>
<td>0.812</td>
<td>0.846</td>
</tr>
<tr>
<td>Weighted Avg.</td>
<td>0.367</td>
<td>7</td>
<td>0.832</td>
<td>0.833</td>
<td>0.858</td>
</tr>
</tbody>
</table>

Table 1. Approach vs. Baseline Performance

- Baseline scores are taken from the PTLM’s raw output
- Score increases after flipping incorrect statements
- The weighted average produces higher variance during scoring, which may allow for easier identification of statements to flip
- High \(\lambda\) for Weighted Average indicates NLI score is a good signal
- Lower \(\lambda\) for Max suggests the max is noisy, so the model learns to weigh PTLM prediction probability higher
- Q: Is a puppy a crustacean? PTLM: Yes Our model: No

Conclusion

Combining NLI output and the PTLM’s confidence in its original predictions through a heuristic function to identify and revise contradictory statements improves both F1 score and logical consistency without needing hand-written constraints.