# MemoryBank: A Flexible Framework for Systemic Beliefs

Stanford CS224N Custom Project

**Siyan Li, Julie Wang, Gabe Mudel**
Department of Computer Science
Stanford University
`{siyanli, juliesw, gmudel}@stanford.edu`

## Abstract

While large language models have achieved success in the Question-Answering domain, one significant limitation is their inconsistencies across answers, such as providing contradictory answers to questions referring to the same fact. The BeliefBank [1] framework tackles this issue by augmenting a QA model with external storage for previous questions and answers (*beliefs*) and by periodically performing constraint solving on these beliefs. This approach results in increasing accuracy and high consistency on the question answering task, but requires hand-constructed constraints. We improve the extensibility of BeliefBank by introducing MemoryBank, which replaces the explicit constraints with a Natural Language Inference (NLI) model containing implicit constraints encoded in its weights. We find that while MemoryBank provides marginal improvement in Question-Answering consistency and accuracy, ultimately the chosen NLI model is not powerful enough to replace the constraint graph in BeliefBank.

## 1 Key Information to include

- Mentor: Eric Mitchell
- External Collaborators (if you have any): None
- Sharing project: No

## 2 Introduction

In their vanilla forms, today's question-answering (QA) models operate on each input question independently. This means that there is no explicit mechanism to enforce logical consistencies between answers. An ideal model would be able to reason over prior statements and beliefs to ensure that they are providing consistent responses. For example, consider the following sequence of questions and answers:

1. Q. Is a poodle a dog? A. Yes
2. Q. Does a dog have a tail? A. Yes
3. Q. Does a poodle have a tail? A. ___

Given previous answers (1) and (2), a reasonably intelligent agent should answer "Yes" to (3). However, previous authors have discovered that models regularly fail to behave consistently in this manner [1]. As such, an active research area has been to encourage QA models to have a consistent set of "beliefs."

Prior work has fine-tuned a QA model with augmented data and objectives to incentivize logical consistency in predictions, which results in marginal improvements over a baseline QA model [2][3].

A more successful framework named BeliefBank augments a QA model with a memory component that stores previous questions and answers (*beliefs*) and leverages them to answer further queries [4]. This framework also retroactively revises the model's beliefs using a weights maximum satisfiability solver to maximize consistency across beliefs. This approach successfully increases model prediction accuracy over time without any additional training while also maintaining high consistency across batches. However, in order to run the solver, BeliefBank requires hand-constructed constraints regarding all possible relations between entities, rendering this approach unscalable.

To improve the scalability and robustness of the BeliefBank framework, we introduce MemoryBank, which eliminates the use of a constraint graph during inference time. We present the following contributions:

1. We demonstrate that a Natural Language Inference model can be used to enforce consistency over a Question Answering model's answers.
2. We provide a local, greedy constraint resolution mechanism that flip beliefs based on NLI model predictions and Question Answering confidence scores that performs significantly faster than a MaxSAT solver.

This work is significant because it provides a step towards making enforcement of consistent model beliefs scalable, allowing models to consolidate their beliefs over time without direct human intervention.

## 3 Related Work

Existing literature [2] [3] explores approaches to improve QA consistency using data augmentation and auxiliary losses. The augmented data are either randomly sampled from the dataset then altered using consistent symbolic logic [2], or instances from a separate dataset [3]. Both approaches employ training losses that discourages inconsistent predictions. Our approach, like BeliefBank, does not require any additional training, and therefore is potentially more generalizable than these approaches.

The BeliefBank framework [4] adds a weighted Max-SAT solver as a reasoning component to the QA pipeline. In doing so, it seeks to minimize the weight of violated "constraints" – implications about a set of entities (in this case, animals) represented as a graph – which the authors sourced from ConceptNet [5]. In addition, the framework selects “relevant” previous beliefs based on constraint graph connections and uses them as context for the QA model. While this approach is performant, it lacks scalability – it is not feasible to store hard-coded entailment relationships for all entities in the world, primarily due to the human oversight that's required to do this. Our system builds on this approach by formulating these constraints with a Natural Language Inference (NLI) model.

## 4 Approach

We introduce a modified BeliefBank called MemoryBank (Figure 1). We implemented MemoryBank from scratch except for the large pretrained language models. Our code can be found here. Below is a description of each component.

### 4.1 Question Answering Model

We use a pretrained Macaw-large model [6], consistent with the BeliefBank paper. We do not fine-tune this model.

### 4.2 MemoryBank

The MemoryBank stores all *existing beliefs*, which come from prior answers from the QA model. The MemoryBank contains an indexer from the FAISS library [7] that is used to store semantically similar sentences. Each sentence is stored in the indexer as a sentence transformer embedding, specifically a SentenceBERT [8]) embedding.

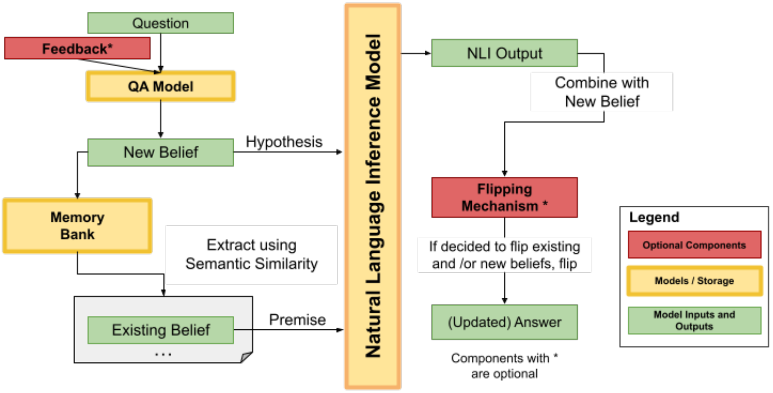

Figure 1: Overview of the MemoryBank pipeline. The red components with asterisks are optional to the framework.

### 4.3 Natural Language Inference model

This is one of the key components of our system. Given a premise $p$ and hypothesis $p$, the NLI model determines if $p$ and $q$ form an entailment, contradiction, or are not related (neutral). Output from this model replaces the handmade constraint graph in [4]. We evaluate the RoBERTa model from [9] and RoBERTa-Large-MNLI model from [10].

### 4.4 Feedback Generation

As in [4], we employ two types of feedback generation mechanisms. **On-topic** feedback is generated by randomly retrieving previous beliefs about the same entity. **Relevant** feedback is created by using sentence transformer embeddings to find the most semantically similar existing beliefs to the new belief. We select three random on-topic beliefs and the top three most semantically similar beliefs for each approach, respectively. Three beliefs are empirically found to yield the best results [4], and allows us to make fair comparisons with the results of Kassner et al.

### 4.5 Belief Flipping Mechanism

The flipping mechanism is the other key component of our system. We refer to inverting a true/false belief as "flipping". We implement and evaluate both a weighted MaxSAT-based approach and a custom heuristic based approach. The Max-SAT approach attempts to solve a boolean satisfiability problem between the incoming hypothesis and all existing beliefs. The heuristic-based approach is a greedy local algorithm that flips new beliefs and existing beliefs to maximize consistency.

#### 4.5.1 Weighted MaxSAT-based Flip Decision

Following BeliefBank, we experiment with using a weighted MaxSAT solver [11] to determine which beliefs are inconsistent. The MaxSAT solver attempts to maximize the satisfied weights when solving a series of weighted constraints containing weighted variables. In BeliefBank, the constraints are manually created with pre-determined weights (either 0.7 or 0.3), and the variables are beliefs weighted by their corresponding QA model confidence. In our formulation, we use NLI inferences instead of manual constraints while keeping the belief weighting the same. The NLI inferences are weighted by the corresponding NLI probabilities. For example, if NLI determines that "It has a dog" $\rightarrow$ "It has a nose" is an entailment with probability 0.7, we then introduce this as a constraint with weight 0.7 to the maxSAT solver.

The MaxSAT solver attempts to solve the cumulative list of NLI inference constraints and beliefs so far. As opposed to heuristic-based flipping, which is conducted at the end of every batch, Weighted MaxSAT-based flipping is called every 10% of the data. This is to minimize the runtime of the maxSAT solver, and is consistent with how BeliefBank runs its solver [4].

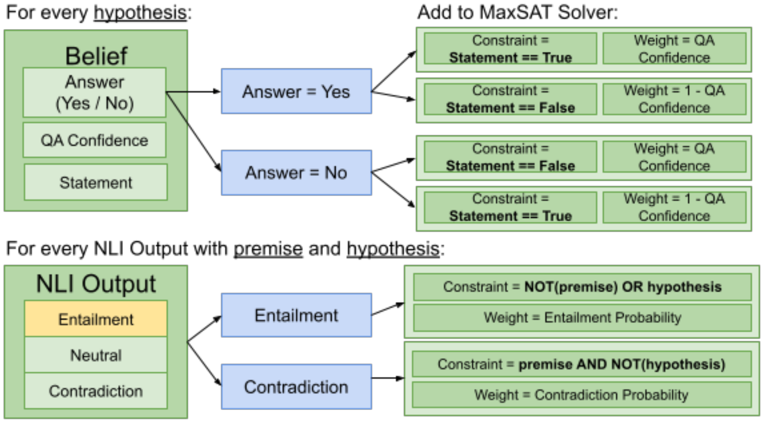
For every hypothesis:
Belief
Answer (Yes / No)
QA Confidence
Statement
Answer = Yes
Answer = No
Add to MaxSAT Solver:
Constraint = Statement == True
Weight = QA Confidence
Constraint = Statement == False
Weight = 1 - QA Confidence
Constraint = Statement == False
Weight = QA Confidence
Constraint = Statement == True
Weight = 1 - QA Confidence
For every NLI Output with premise and hypothesis:
NLI Output
Entailment
Neutral
Contradiction
Entailment
Contradiction
Constraint = NOT(premise) OR hypothesis
Weight = Entailment Probability
Constraint = premise AND NOT(hypothesis)
Weight = Contradiction Probability

Figure 2: Explanation of how we add Max SAT constraints to the solver.

### 4.5.2 Heuristic-based Flipping

Given the new belief $q$ (current answer from the QA model), set of premises $P = \{p_1, ..., p_n\}$, (on-topic or relevant feedback), and NLI output probabilities $S = \{(e_1, n_1, c_1), ..., (e_n, n_n, c_n)\}$[1] between all $p_i \in P$ and $q$, MemoryBank decides heuristically whether the new belief is being contradicted if the number of contradicting premises is greater than the number of entailing premises. That is:

$$|\{s_i \in S|\max(s_i) = c_i\}| > |\{s_i \in S|\max(s_i) = e_i\}|$$

We use this heuristic because each NLI model output can be interpreted as a sort of "vote" with respect to consistency between $P$ and $q$ – if there are more votes suggesting that $P$ and $q$ are not logically consistent, then it follows that the system should revise its beliefs. This method is also more computationally efficient than the Max-SAT approach – its runtime scales in proportion to $|P|$, whereas the Max-SAT approach is markedly slower.[2]

If the contradiction vote is larger than the entailment vote, the NLI model is signaling that $q$ is heavily contradicted by $P$. We further divide this into two cases:

**(1)** $q$ is a correct belief, and some number of premises that contradict $q$ are incorrect beliefs. In this case, MemoryBank flips premises $p_i \in P$ that form a contradiction with $q$, as determined by the NLI model. This case is executed when contradicting premises are generally less confident than the hypothesis $q$.

**(2)** $q$ is an incorrect belief, and some number of premises that entail $q$ are incorrect beliefs as well. In this case, MemoryBank flips premises $p_i \in P$ that form an entailment with $q$, as well as $q$ itself. This case is executed when the *majority* of premises are both more confident than $q$ and are in contradiction with $q$.

To decide which premises $p_i$ to flip in cases (1) and (2), MemoryBank compares the difference between the QA confidence scores of $p_i$ and $q$; if the premise confidence is more than a threshold amount $\gamma$ below the hypothesis confidence, the premise is flipped. That is, we flip if $score_{p_i} + \gamma < score_q$.

To flip a belief, MemoryBank inverts the answer corresponding to the belief of interest and update the corresponding QA confidence to a default value (which is a hyper-parameter). More detailed explanations of the hyperparameters used in this process are included in the Experimental Details section.

[1] $e_i, n_i, c_i$ denote entailment, neutral, and contradiction probabilities as predicted by the NLI model. Note $e_i + n_i + c_i = 1$.

[2] [11] doesn't provide an exact runtime bound for the Max-SAT solver, but the time complexity of iterative flipping of entries in our MemoryBank is at least proportional to its size, $|M|$. After as little as one batch of questions, $|P| << |M|$.

**Algorithm 1** Heuristic-based Flipping

```
Given: New belief (hypothesis) nb, List of relevant previous beliefs (premises) prems, NLI predictions with premises and new belief nli_probs

if mostly_contradicting(nli_probs) then
                                        ▷ Decide whether most relevant beliefs contradict nb
    prems_contract ← Previous beliefs that contradict nb
    prems_entail ← Previous beliefs that entail nb
    for_votes ← Number of contradictors with lower QA Confidence than nb
    against_votes ← Number of contradictors with higher QA Confidence than nb
    if more_for_than_against(for_votes, against_votes) then
        optionally_flip(prems_contract)
    else
        Flip nb                          ▷ Flip the new belief, assign new confidence value.
        optionally_flip(prems_entail)
    end if
end if
Return: nb

Note: All functions make use of hyperparameters that were tuned in our experiments.
```

## 5 Experiments

### 5.1 Dataset

We use BeliefBank's [4] silver facts as the dataset of questions to answer. These facts consist of 12,363 true/false statements about 85 plants and animals derived from ConceptNet's relations [5]. They are provided as triplets containing (entity, relation, answer). Additionally, we use the constraint graph provided by BeliefBank [4] in evaluating the performance of the model. These constraints describe which relations entail others, for example "IsA, dog" → "HasA,nose" means that if an entity is a dog, then it has a nose. Similarly, "IsA,art" → !"IsA,company" means if an entity is art, then it is not a company.

We wrote our own template-based approach for translating constraint graph tuples and question triplets into natural language questions and declarative statements. For example, the silver fact ("poodle", "IsA,dog", "yes") can be formulated as both a question-answer pair: ("Is a poodle a dog?", "yes"), and declarative statement: "A poodle is a dog."

We randomly split the silver facts dataset into 20% validation and 80% test. There is no training dataset because we never modify any model weights. We use the validation set for fine-tuning of the hyperparameters and the test dataset for final evaluation.

### 5.2 Evaluation Method

At the end of every 10% of the data, the beliefs in the memory bank are evaluated on their F1 score and consistency. Note that *all* existing beliefs are used in calculating these metrics, not just the current batch's beliefs. This is because our goal is to improve model consistency over time, so all previous facts from the model should be considered. We implement the complement of Li et al. (2019)'s conditional constraint violation metric: the number of violated constraints divided by the total number of applicable constraints.

$$\text{consistency} = 1 - \tau$$
$$\tau = |\{c_i \mid \neg(s_i.l_i \rightarrow s_j.l_j)\}| \;/\; |\{c_i \mid s_i.l_i\}|$$

Where $c_i$ is a constraint consisting of sentences $s_i, s_j$ and their corresponding truth values $l_i, l_j$.

| Model | Flipping | Feedback | NLI Model |
|---|---|---|---|
| Baseline | None | None | `ynie/roberta-large-snli_mnli` |
| Heuristic Flipping | Heuristic | None | `ynie/roberta-large-snli_mnli` |
| Heuristic Flipping + Relevant Feedback | Heuristic | Relevant | `ynie/roberta-large-snli_mnli` |
| Heuristic Flipping + On Topic Feedback | Heuristic | On Topic | `ynie/roberta-large-snli_mnli` |
| SAT Flipping | Max SAT | None | `ynie/roberta-large-snli_mnli` |
| SAT Flipping + Relevant Feedback | Max SAT | Relevant | `ynie/roberta-large-snli_mnli` |
| SAT Flipping + On Topic Feedback | Max SAT | On topic | `ynie/roberta-large-snli_mnli` |
| Heuristic Flipping with roberta model | Heuristic | None | `roberta-large-mnli` |

Table 1: Descriptions of the model configurations which were evaluated.

### 5.3 Experimental details

Like Kassner et al, we evaluate the system dynamically. Batches of questions are posed to the question answering model, and a growing MemoryBank holds the edited beliefs. The following model configurations were tested:

The following hyperparameters were tuned:

- `sentence_similarity_threshold` - Threshold used to retrieve similar sentences from the index.
- `default_flipped_confidence` - When flipping a belief, the initial confidence on that flipped belief.
- `flip_premise_threshold` - If the hypothesis confidence + `flip_premise_threshold` is greater than the premise confidence, flip the premise.

### 5.4 Results

From the results displayed in Figure 3, we can see that on the test set, our best model is able to outperform the baseline by roughly 6% on accuracy, and 5% on consistency. However, none of our models perform as well as BeliefBank. Notably, we do not see the increasing model accuracy over epochs that the BeliefBank is able to achieve. We explore why this is in the Analysis section.

## 6 Analysis

We draw the following conclusions based on our experiments:

- **NLI models do not have sufficient world knowledge to label constraints as entailments**. A perfect NLI model would assign "entailment" with probability 1 to every constraint. However, since an NLI model has imperfect knowledge of the world, it assigns many constraints to "neutral" rather than "entailment". In Figure 4, we visualize a histogram of the weights assigned to "entailment", "neutral" and "contradiction". We see that many of the constraints are labelled "neutral" with high confidence! Furthermore, this effect is present in another popular NLI model, `roberta-large-mnli`. These figures illustrate the primary reason that our system is not performing at the level of that given in [4] – it seems that generally, state-of-the-art NLI models do not have the inference capabilities necessary to reliably evaluate the correctness of boolean statements as they were presented in our dataset. We suspect this could be the case in part because of a significant distribution shift between the training data used to train the NLI models and the data given in [4].
- **NLI models are sensitive to the phrasing of questions** We test how the models perform on the logical inverse of constraints, namely, $!(p \to q) \equiv p \wedge \neg q$. Surprisingly, we find that the `roberta-large-mnli` model starts to label most statements as contradictions! This includes statements which should be completely neutral to each other, like "An owl can fly" and "A human is not a company" (labelled as a contradiction with probability 0.876). We believe that this is because `roberta-large-mnli` is sensitive to the word "not". This effect is less pronounced in the `ynie-roberta` model, which has been trained on a more

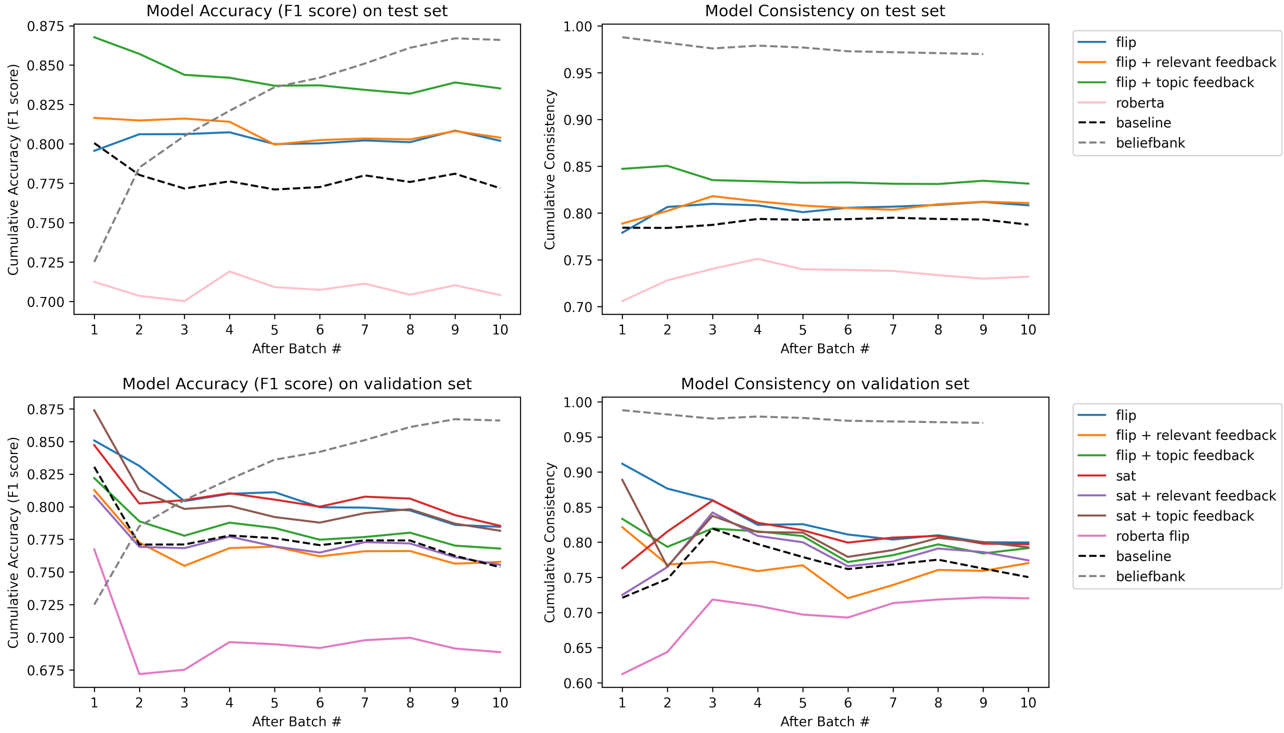

Figure 3: MemoryBank accuracies and consistencies on the validation set and test set. Note that MaxSAT-based approaches are not included in the test set plots. Additionally, we include accuracies and consistencies reported in the original BeliefBank paper for comparison, but the batch size and sampling differ from the other data.

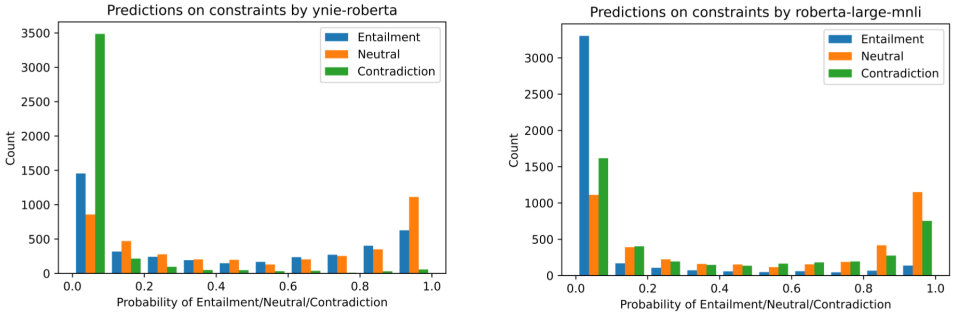

Figure 4: Histograms of probabilities for two pretrained NLI Models on original constraints

difficult question answering task [9]. This histogram of weights is included in the Appendix, Figure 5.

- **On-topic feedback is more beneficial than relevant feedback**. We hypothesize this is a result of how MemoryBank retrieves relevant feedback. BeliefBank utilizes the constraint graph to extract the most relevant potential clashes to new beliefs to use as context for the QA model, while we use semantic similarity from SentenceBERT embeddings. It should be noted that semantic similarity does not equate logical relevance. For example, a belief deemed relevant by MemoryBank to "a hound can mother its young" is "a hound is not larger than cat". As a result, feeding in semantically relevant content may not provide as much logical context to the QA model compared to BeliefBank. Furthermore, when there aren't many entries in the MemoryBank, there may be no relevant feedback to return. On the other hand, as long as at least three questions about the same entity have been answered, on-topic feedback will generate additional context for the question at hand.

- **Our Max SAT-based flipping decision mechanism does not outperform BeliefBank.** In BeliefBank, the weighted Max SAT solver attempts to maintain consistency over a fixed set of constraints over all previous batches. For MemoryBank, the list of constraints is ever-growing as we add new NLI inferences into the constraints maintained by the Max SAT solver. Since we have demonstrated that our NLI model does not successfully recognize all relationships present in the original constraint graph, the added NLI inference constraints would not be comparable to the original constraints. A potential future step would involve filtering incoming NLI inference constraints to only include strong logical correlations. This would also improve the runtime of the MaxSAT solver.
- **MemoryBank mistakes are mostly false positives.** We find that both the baseline and the flip only models have a higher false positive rate (about 0.15) than false negative rate (around 0.08). This is because the false positives are usually statements about unusual entities, or are nonsensical questions, for example, "Is a peony an Africa?" and "Is an oryx a body of water?". These are unlikely to have occurred in either the Question Answering model's nor the NLI model's training.
- **Unlike BeliefBank, we do not observe increasing accuracy over batches.** There are likely batching and sampling discrepancies between our work and BeliefBank, which would lead to different results. More importantly, we have shown that our NLI model is unable to capture all constraints as entailments, leading to a noisy signal from the NLI model when determining whether to flip beliefs.

## 7 Conclusion

We have shown that question-answering models can improve their consistency and accuracy across answers by using an external memory to store prior beliefs and an NLI model and constraint solver to resolve inconsistencies between beliefs. However, because an NLI model does not have perfect world knowledge, it does not lead to the same accuracy and consistency improvements as an approach which uses a hand crafted constraint graph. Our experiments with various NLI models, constraint resolution mechanisms, and different feedback mechanisms have shown this to the be the case. While our improvements are marginal, we believe our approach to be more generalizable than BeliefBank. Improvements in the NLI model would directly lead to better question-answering consistency, whereas a manually created constraint graph is fundamentally limited in scope.

Future work would include improving the computational performance of the Max SAT solver based approach by performing hyperparameter tuning so that there are comparable results to the heuristic based approach. We would also evaluate MemoryBank on different datasets, especially those which are not just true/false statements. This would allow us to evaluate the flexibility of our approach, as the NLI model can produce more nuanced classifications than just true false statements (unlike the constraint graph in BeliefBank).

## References

[1] Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184, Florence, Italy, July 2019. Association for Computational Linguistics.

[2] Akari Asai and Hannaneh Hajishirzi. Logic-guided data augmentation and regularization for consistent question answering. *CoRR*, abs/2004.10157, 2020.

[3] Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. A logic-driven framework for consistency of neural models. *CoRR*, abs/1909.00126, 2019.

[4] Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. Beliefbank: Adding memory to a pre-trained language model for a systematic notion of belief. *arXiv preprint arXiv:2109.14723*, 2021.

[5] Joshua Chin Robyn Speer and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. *AAAI*, 2017.

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

[7] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[8] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[9] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[11] Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer, 2008.

# A Appendix

## A.1 Using NLI models to infer the original constraints

Here we include the figures generated when examining NLI models' sensitivity to phrasing.

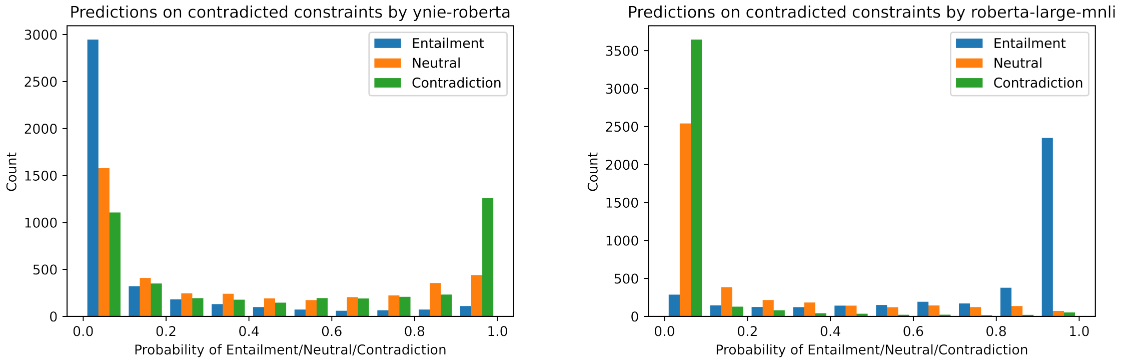

Figure 5: Histograms of probabilities for two pretrained NLI Models on inverse constraints

### A.2 Data examples

| | |
|---|---|
| A black race car starts up in front of a crowd of people. | A calf can die no more than once. |
| A man inspects the uniform of a figure in some East Asian country. | A puppy can eat cat food but probably shouldn’t. |
| A happy woman in a fairy costume holds an umbrella. | An orchid is a flower. |
| A soccer game with multiple males playing. | A peony is not a car. |
| A black race car starts up in front of a crowd of people. | A cat cannot sleep on a windowsill. |

Table 2: Selected examples from NLI training data (left) and BeliefBank (right).