CSE156 Final Report: ProtoQA

Ishan Vaish
University of California, San Diego
ivaish@ucsd.edu

Mehak Kapur
University of California, San Diego
mekapur@ucsd.edu

Adam ELDabet
University of California, San Diego
aeldabet@ucsd.edu

Carson Rae
University of California, San Diego
carae@ucsd.edu

Vincent McCloskey
University of California, San Diego
vmccloskey@ucsd.edu

Abstract

ProtoQA presents a unique challenge to natural language processing (NLP) by requiring models to generate multiple reasonable answers to questions that test commonsense reasoning. This dataset, inspired by the game show *Family Feud*, emphasizes cultural contexts and social norms. We chose this task due to its compelling bridge between machine learning and human reasoning. Our study surveys current ProtoQA methodologies, highlights their limitations, and proposes techniques to enhance performance. By improving models' commonsense reasoning in everyday scenarios, we aim to develop AI systems that interact naturally with humans. Our key contributions include:

- A two-stage approach integrating external commonsense knowledge (ConceptNet, ATOMIC) for better candidate generation and re-ranking.
- An improved ranking mechanism prioritizing brevity, concreteness, and typicality.
- Experimental evidence that knowledge integration substantially improves performance, especially on culture- or commonsense-heavy questions.

1 Introduction

ProtoQA presents a unique challenge to NLP: models must propose multiple plausible answers to questions that require commonsense reasoning. The dataset, inspired by *Family Feud*, requires culturally and socially informed reasoning. Our goal is to evaluate current ProtoQA methods and design improvements. Initial experiments show that many models struggle with tasks demanding cultural knowledge or social understanding, motivating a more knowledge-rich framework. By refining these models' ability to handle common, everyday scenarios, we help pave the way for AI systems that can engage users more naturally in real-world situations. Our hypothesis is that integrating structured commonsense knowledge into transformer-based models will significantly improve the diversity and accuracy of answers to ProtoQA questions.

2 Background

Boratko et al. (2020) introduced ProtoQA in 2020 with questions designed to evaluate *prototypical* commonsense reasoning. Unlike QA tasks emphasizing strictly factual content, ProtoQA requires

Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023.

a *set* of possible answers, graded by plausibility. Earlier work leveraged ensemble approaches, knowledge-graph traversal, and fine-tuning large pre-trained models. Despite some successes, these strategies struggle to produce *diverse yet relevant* answers aligning with human intuition. Metrics such as precision, recall, and mean average precision (MAP) measure performance, emphasizing alignment to *human frequency* data. Our approach builds on these methods by explicitly dividing answer generation from ranking, bringing more structure and interpretability. Our post-processing also tackles issues like repetition and generic or low-quality outputs.

3 Method

Our approach to tackling the ProtoQA task combines transformer-based language models with structured commonsense knowledge from ConceptNet, followed by a specialized answer ranking pipeline. We implemented and tested multiple strategies across two primary phases: **generation** and **re-ranking**.

3.1 Generation Phase

We employed several language models—T5, BART, and GPT-2—fine-tuned on the ProtoQA dataset to generate candidate answers. Based on our preliminary results (which showed low semantic accuracy using models alone), we enhanced the pipeline by integrating external knowledge.

Key Generative Setups:

1. **T5 + ConceptNet**:
   - Used T5-base fine-tuned on 1000 ProtoQA samples.
   - Retrieved up to 50 related concepts per question from ConceptNet.
   - Combined generated answers with ConceptNet concepts for a broader candidate pool.
2. **Clean Beam T5**:
   - Implemented beam search decoding (`num_beams=5`, early stopping, and length penalty).
   - Improved data cleaning and prompt construction significantly reduced noise.
3. **Fast T5**:
   - Minimal training with only 1 epoch and `num_beams=3`.
   - This lightweight setup surprisingly outperformed more complex pipelines, highlighting the impact of decoding strategies.
4. **Multi-Model Ensemble**:
   - Generated answers using T5, BART, and GPT-2 simultaneously.
   - Augmented outputs with ConceptNet concepts to form a diverse and comprehensive answer set.

3.2 Ranking Phase

Since raw generations (even when knowledge-enhanced) often contained irrelevant, duplicate, or generic answers, we developed a custom ranking pipeline based on multiple criteria:

- **Lexical similarity** (e.g., via TF-IDF).
- **Semantic similarity** (e.g., embedding-based cosine similarity).
- **Commonsense alignment** (e.g., typicality, brevity, and concreteness).

A weighted composite scoring function was used to prioritize more prototypical and human-aligned responses. Final outputs were pruned using a semantic similarity threshold to remove redundancy.

3.3 Post-Processing

We addressed issues such as hallucination, repetition, and low-quality answers through the following strategies:

- Sentence-level pruning based on semantic similarity.
- Filtering out generic phrases (e.g., "<pad> food") using regex and blacklist rules.
- Grammar correction and capitalization fixes on final answers.

4 Experiments

We designed controlled experiments to validate each enhancement component in our pipeline. The process included:

1. **Baseline transformer performance:** Evaluate T5, BART, GPT-2 without external knowledge.
2. **Knowledge integration:** Inject commonsense information from ConceptNet (and optionally ATOMIC).
3. **Answer ranking:** Re-score candidates using lexical, semantic, and *typicality* metrics.

Below, we detail the models, datasets, baselines, and implementation code.

4.1 Model

We experimented with the following transformer-based architectures:

- **T5** (*Text-To-Text Transfer Transformer*): All tasks reformulated in a text-to-text format, well-suited for structured outputs.
- **BART** (*Bidirectional and Auto-Regressive Transformer*): A denoising autoencoder robust to partial inputs.
- **GPT-2** (*Generative Pretrained Transformer-2*): Excels at fluent text generation but can ramble without constraints.

All models were fine-tuned on ProtoQA to generate multiple plausible responses. We tested two approaches:

1. Direct fine-tuning of each transformer.
2. Knowledge-augmented generation (ConceptNet + model) + ranking.

Implementation 1 – T5 + ConceptNet + TF-IDF Ranking (Baseline Knowledge Integration)

- **Description:** Combine T5-generated answers with ConceptNet candidates, then rank via TF-IDF to the question.
- **Setup:**
  - T5-base fine-tuned on 1000 samples.
  - Query ConceptNet for up to 50 related concepts per question.
  - TF-IDF ranking; measure Precision@5 using fuzzy matching.
- **Result:** Average Precision@5 = 0.01
- **Observation:** Predictions often noisy (e.g., "<pad> food"), ConceptNet gave coverage but was misaligned with gold answers, and TF-IDF alone was insufficient.

Implementation 2 – Clean T5 with Beam Search Decoding

- **Description:** Use beam search and improved data preprocessing.

- **Setup:**
  - T5-base, 3 epochs, learning rate $5e{-}5$.
  - Beam search $(\text{num\_beams}{=}5)$, early stopping, length penalty.
- **Result:** Average Precision@5 = 0.05
- **Observation:** Less noise (e.g., "tiger", "pancakes", "lawyer"), minimal external knowledge usage, coverage still limited.

Implementation 3 – Fast T5 (Lightweight, Efficient Baseline)

- **Description:** Minimal setup with just 1 epoch, beam search $(\text{num\_beams}{=}3)$, refined prompt.
- **Result:** Average Precision@5 = 0.08
- **Observation:** Outperforms more complex T5+ConceptNet; underscores the impact of *clean data* and *decoding choices* over naive knowledge integration.

Implementation 4 – Multi-Model Baseline + ConceptNet + Composite Ranking + Post-Processing

- **Description:** Combine T5, BART, and GPT-2 outputs with ConceptNet expansions, rank with a weighted composite metric (*brevity, concreteness, typicality*), then prune redundancies.
- **Pipeline:**
  1. Generate from T5, BART, GPT-2.
  2. Retrieve related concepts from ConceptNet.
  3. Rank all with a composite scoring function.
  4. Remove duplicates (semantic similarity threshold).
- **Observations:**
  - GPT-2 often echoed or hallucinated the question.
  - BART repeated input with blanks or offered little new info.
  - T5 sometimes broke grammar if not fine-tuned.
  - ConceptNet gave diverse entities (*"airplane," "canary," "calendar"*), plus some irrelevant ones (*"every person," "imaginary friend for grown ups"*).
  - Composite ranking helped push plausible answers to the top (e.g., "pancakes," "duck," "taxi," "baggage").
- **Representative Examples:**
  - Q: *Name a bird with a color in its name* $\rightarrow$ top outputs: "blue," "yellow," "crow," "robin," "canary".
  - Q: *Name something at the airport* $\rightarrow$ extracted: "carry-on bag," "check-in counter," "taxi," "terminal".
- **Result:** No numeric Precision@5 computed, but qualitatively more *diverse* and *human-like* than all previous baselines.
- **Contribution:** This introduces a new multi-model + knowledge ensemble, composite ranking metric, and post-processing approach.

4.2 Datasets

We used the **ProtoQA** dataset:

- Format: JSONL with normalized questions and grouped gold answers.
- For experiments: 1000 training examples, 100 evaluation examples.

4.3 Baselines

Baselines included:

- Pretrained (zero-shot) T5, BART, GPT-2.
- Fine-tuned T5 (without external knowledge).
- ConceptNet-only generation + naive ranking.
- TF-IDF vs. composite scoring ablations.

These help isolate the effect of knowledge integration, answer ranking, and post-processing.

4.4 Code

Code is provided in two shared Colab notebooks:

- `https://colab.research.google.com/drive/1_a_9jKxqETs0jxuUYxHXDBHqlBPDO9rh`
- `https://colab.research.google.com/drive/1giXHkBMClBpItUG2JEMkaMYcLIq_eJ-f`

Both notebooks illustrate data preprocessing, model fine-tuning, knowledge integration, and ranking steps.

5 Results

We report **Precision@5** for three numeric experiments, plus qualitative evaluation of the multi-model approach.

Table 1: Final Results: Precision@5 on Numeric Experiments

Implementation	Knowledge	Ranking	Avg P@5
T5 + ConceptNet + TF-IDF (Impl. 1)	ConceptNet + T5	TF-IDF	0.01
T5 Clean Beam (Impl. 2)	None	None	0.05
Fast T5 (Impl. 3)	None	None	0.08
Multi-Model + ConceptNet (Impl. 4)	T5, BART, GPT-2 + ConceptNet	Composite + Postproc	Qualitative

Implementation 1 (T5 + ConceptNet + TF-IDF).

- **P@5 = 0.01.**
- Naive ranking did not adequately filter noisy or off-topic candidates.

Implementation 2 (T5 Clean Beam).

- **P@5 = 0.05.**
- Beam search decoding and better prompts produced more coherent outputs, yet lacked external knowledge coverage.

Implementation 3 (Fast T5).

- **P@5 = 0.08.**
- A "fast" approach with minimal training but well-tuned decoding outperformed Implementation 1, highlighting the hazards of *unstructured* knowledge injection.

Implementation 4 (Multi-Model + ConceptNet).

- **No numeric P@5** reported.
- Qualitative inspection shows far *greater* diversity and *commonsense alignment*.

- Redundancy removal and composite ranking overcame many issues from earlier pipelines.

Surprising Findings.

1. A 1-epoch T5 with careful beam search (Impl. 3) beat T5+ConceptNet under naive TF-IDF (Impl. 1).
2. Proper decoding hyperparameters often improved performance more than small architecture changes.
3. Multi-model ensembles + knowledge appear best, albeit not fully quantified by P@5 alone.

Comparison to Original ProtoQA Benchmarks. We did not replicate official leaderboard MAP or nDCG, but our partial metrics confirm the value of knowledge integration + ranking vs. raw transformer baselines.

Error Analysis

To better understand the limitations of our system, we manually reviewed incorrect or low-ranked outputs. Common error types included:

- **Redundancy**: Answers like "airplane" and "plane" were treated as separate.
- **Semantic drift**: Some answers were topically adjacent but not directly relevant.
- **Commonsense gaps**: Abstract or culturally specific questions failed to retrieve relevant ConceptNet concepts.

These suggest that future models should combine semantic similarity pruning with more nuanced knowledge retrieval.

6 Conclusion

Our experiments reveal that **pure language models** (T5, BART, GPT-2) are insufficient for robust commonsense QA, frequently echoing the query or proposing irrelevant text. By adding external knowledge from ConceptNet (and optionally ATOMIC), we significantly broadened candidate answers and produced more natural, commonsense-aligned responses. Further, **two-stage pipelines** that first generate candidate answers and then *rank* them show distinct performance benefits, filtering out hallucinations and duplicates.

Nonetheless, these methods still lag behind human-level commonsense, particularly for abstract, culturally nuanced, or open-ended questions. Future directions include:

- Larger transformer backbones (e.g., T5-XXL).
- More refined knowledge sources (ATOMIC, WordNet) with category-specific expansions.
- Advanced ranking (contrastive learning, semantic embeddings).

Through enhanced commonsense representation and multi-stage pipelines, we move closer to AI systems capable of genuine, culturally aware QA.

References

[1] M. Boratko, et al. ProtoQA: A Question Answering Dataset for Prototypical Commonsense Reasoning. *arXiv preprint arXiv:2005.00771*, 2020. `https://arxiv.org/abs/2005.00771`.