# Known Unknowns and Unknown Unknowns: Teaching QA models to recognize when they can't answer a question

Bogac Kerem Goksel

## Problem Description

The general idea of my project is exploring how question answering models today handle questions that are not answerable given the information context but are still relevant to the context as in the following example:

**Context:**
The costume collection in the V&A Museum in London consists of 2,500 period pieces from the Victorian and Edwardian era London theaters.

**Question:**
How many pieces does the jewelry collection in the V&A Museum in London have?

**Expected Answer:**
I can't answer this question

**Most Models Today:**
2,500

There has been recent interest in this problem, and there's recent work on generating such questions using both human writers and automated data augmentation methods. [1,2]

In this project I work on automated generation methods and evaluate their effectiveness in training models that are then evaluated on the human-generated dataset.

## Datasets Used

I use the Stanford Question Answering Dataset (SQuAD) as the baseline dataset of passages, related questions and answers. SQuAD only consists of factual questions that are paired with relevant Wikipedia paragraphs that contain the answer to them.

[1] has released SQuADRun, a human-made dataset of question-context pairs that are similar in structure to SQuAD but where the questions are not answerable given the context even though they share a topic.

## Generating More Unanswerables

[2] experiments with re-pairing questions and contexts from the original SQuAD dataset such that the questions and contexts share a topic but the exact answer to the question is not included in the paragraph using the following method:

- Compute tf-idf scores using one-hot bag-of-word features from paragraphs and questions
- Use cosine distance between these feature vectors to rank the similarity of each question-paragraph pair
- For each question, pick the closest paragraph that doesn't contain the original answer string in it

I experiment with this method, as well as combining TF-IDF features and average word vector representations before ranking. I also experiment with a more rule based approach where I just remove the sentences containing the original answer string from the original context and keep the rest as an unanswerable context.

## Model Used for Evaluation

I mainly use the model from [2] as its architecture already outputs the probability that it thinks the question is unanswerable. The architecture consists of the following layers:

- Embedding
- Bidirectional Attention
- Self-Attention
- Answer Prediction
- No-answer Probability Prediction

As such, the model always predicts an answer but also outputs its 'confidence' in that answer, which means the threshold for outputting 'no answer' can be empirically chosen over the validation data.

I've experimented with a boosting-like approach where consecutive models are trained using a subset of the generated unanswerable questions that were the most difficult for the previous models, hoping later models can learn more fine grained features.

## Results

| Question Set Used to Train Model | SQuADRun F1 | Original Eval F1 | Ratio of Unanswerables with probability 1.0 |
|---|---|---|---|
| 1) TF-IDF 1:1 | 55.736 | 75.099 | 20.048 |
| 2) TF-IDF 2:1 | 56.899 | 74.167 | 22.944 |
| 3) TF-IDF 7:1 | 56.282 | 72.529 | 23.944 |
| 4) TF-IDF 1:1 (hardest 10% of TF-IDF 10:1) | 56.731 | 73.811 | 23.013 |
| 5) Ensemble of 1 & 4 | 56.731 | 73.441 | 24.496 |
| 6) Originals with answer sentences removed | 50.923 | 11.985 | 93.003 |

## Discussion

The results show that generating unanswerable questions is a very difficult problem. The main issues are:

- Difficulty of generation for humans
- Majority of TF-IDF generated questions are irrelevant
- Good portion of the harder TF-IDF generated questions are somewhat answerable, but with a different answer
- Training with more/harder unanswerable questions biases models too much towards counting too many questions as unanswerable
- The domain of unanswerable questions is really big and questions generated with different methods have different distributions (that are themselves different from the human-generated distribution)
- Training the models on only relevant paragraphs conditions the model on the answerability of the question, models that work well under this conditioning don't necessarily work well without this conditioning.

## Future Work

Future work will involve exploration of more rule-based methods for questions generation and a semi-supervised approach where more rule-based transforms are applied without the constraint that the new question is not answerable but that the answer cannot have stayed the same. Also experimenting with different model architectures.