A Comprehensive Evaluation of Question Answering Systems: Introducing the UpstreamQA Dataset

Fernando Almansa Moreno De Barreda
Supervised by: Prof. Janet B. Pierrehumbert

UpstreamQA Dataset and Models:
https://feralmansa01.github.io/UpstreamQA/

1 Introduction and Motivation

Recent Natural Language Processing (NLP) advancements have led to the release of powerful models like ChatGPT. Given their extensive potential impact on society, it is crucial to ensure that their Question Answering (QA) capabilities provide reliable and accurate responses to a wide range of questions and so they do not mislead users.

Project Goals:

The primary goal of this thesis is to present a new dataset designed to train and evaluate QA systems on these three crucial properties simultaneously:

- handling unanswerable questions
- robustness to paraphrasing
- adaptability to evolving information

What is Question Answering?

Question Answering (QA) is a subfield of NLP that aims at developing systems capable of providing answers to questions posed in natural language.

Shortfalls of Current QA Datasets and Introduction of UpstreamQA:

- SQuAD 2.0 [1] is a popular QA dataset, it includes unanswerable questions. However, it cannot train or test a system's adaptability to evolving information.
- The StreamingQA [2] dataset evaluates how QA systems adapt to evolving information but it does not include unanswerable questions.
- To overcome these limitations, we introduce UpstreamQA, the first dataset designed to train and evaluate QA systems on all three of the crucial capabilities outlined above.

2 UpstreamQA

UpstreamQA builds upon StreamingQA [2] by adding 30,000 unanswerable and 21,000 paraphrased questions across 10 and 7 distinct categories respectively. Questions are created through strategic substitutions; four example question categories presented below:

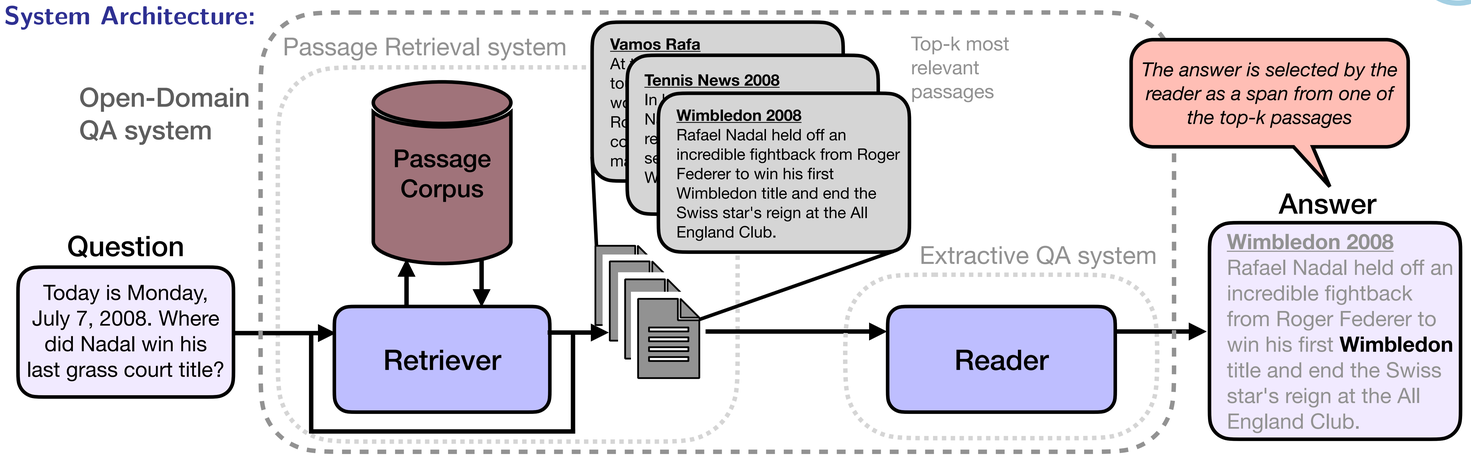
Like in StreamingQA [2], in order to to measure a system's adaptability to evolving information, questions and passages in UpstreamQA are temporally grounded.

Examples of Question Categories from UpstreamQA:

		Creation Strategy	Example Question	Gold Label
Temporal	Relative future date	**`U-RF`**: Swap the year in the original question with a relative mention of a corresponding future year.	'Today is Thursday, May 28, 2020. In 2020 which pharmaceutical company is working with Oxford?'	answerable
			'Today is Thursday, May 28, 2020. Four years after 2019 which pharmaceutical company is working with Oxford?'	unanswerable
Entity	Unknown entity swap	**`U-US`**: Swap a named entity in the original question with a named entity that does not appear in the entire corpus.	'Today is Saturday, February 4, 2012. Which Man Utd player was injured before the game against Arsenal?'	answerable
			'Today is Saturday, February 4, 2012. Which Man Utd player was injured before the game against Striped Southerners?'	unanswerable
Temporal	Relative suitable date	**`P-RS`**: Swap the four-digit year in the original question with a relative reference to the four-digit year.	'Today is Saturday, February 8, 2020. In 2020 what is the name of the NBA logo?'	answerable
			'Today is Saturday, February 8, 2020. Four years before 2024 what is the name of the NBA logo?'	paraphrased
Entity	Entity nickname swap	**`P-NS`**: Swap the entity in the original question with its entity nickname.	'Today is Saturday, September 7, 2013. How many times did Van Persie make appearances for Manchester United?'	answerable
			'Today is Saturday, September 7, 2013. How many times did Van Persie make appearances for the Red Devils?'	paraphrased

3 Methods

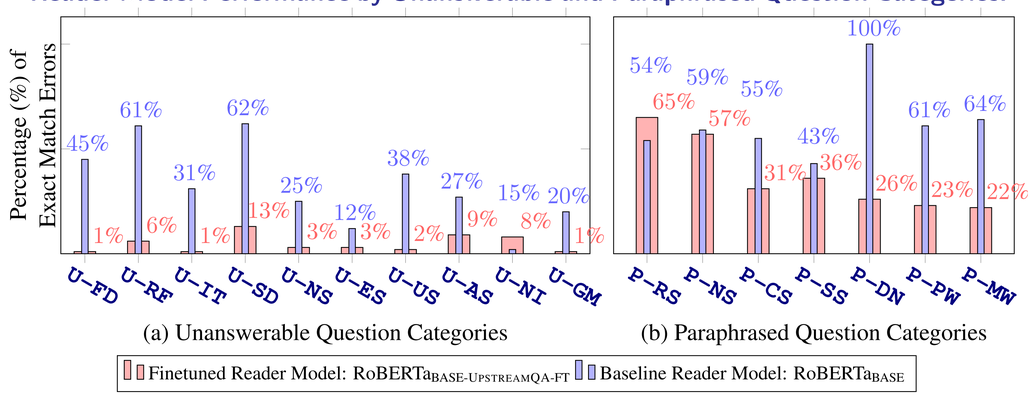
System Architecture:



- We use an Open-Domain QA system consisting of two stages. First, given an input question, the retriever returns the top-k most relevant passages from a large corpus of passages; Then, the reader finds the answer as a span from one of these returned passages.
- We compare baseline RoBERTa [3] readers to RoBERTa readers finetuned on UpstreamQA, and we also evaluate both sparse (BM25) and dense (DPR-FT [4]) retrievers.

4 Results and Contributions

Reader Model Performance by Unanswerable and Paraphrased Question Categories:



Open-Domain QA System Performance by Question Type:

Retriever-Reader	Answerable Questions: Computer Generated EM	Computer Generated F1	Human Written EM	Human Written F1	Paraphrased EM	Paraphrased F1	Unanswerable EM	Unanswerable F1	Overall EM	Overall F1
BM25$_{k=5}$	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
RoBERTa$_{\text{BASE-UPSTREAMQA-FT}}$	54.0	60.2	29.1	40.0	43.4	49.6	86.9	86.9	57.1	62.3
RoBERTa$_{\text{LARGE-UPSTREAMQA-FT}}$	**57.3**	**63.5**	**33.7**	**45.5**	**46.4**	**52.4**	**90.6**	**90.6**	**60.5**	**65.9**
DPR-FT$_{k=5}$	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
RoBERTa$_{\text{BASE-UPSTREAMQA-FT}}$	34.8	40.3	18.3	27.4	29.3	34.8	85.0	85.0	43.3	48.0
RoBERTa$_{\text{LARGE-UPSTREAMQA-FT}}$	36.9	42.7	21.3	30.9	31.3	37.0	87.5	87.5	45.6	50.5

Contributions:

- We introduce UpstreamQA: a novel dataset for QA systems.
- We show that finetuning reader models on UpstreamQA significantly improves performance on unanswerable and paraphrased questions.
- We find sparse retrievers to be the most effective on UpstreamQA.
- We make all our models and the UpstreamQA dataset accessible through our website (access via QR code).

Future Work:

We plan to evaluate top chatbots, including ChatGPT, on UpstreamQA, with the aim of identifying their strengths and weaknesses in the three crucial QA properties this project outlines.

References:

[1] SQuAD 2.0 [Rajpurkar *et al.*, 2018]
[2] StreamingQA [A. Liska *et al.*, 2022]
[3] DPR [Karpukhin *et al.*, 2020]
[4] RoBERTa [Liu *et al.*, 2019]