Know what you do not know:

improving the detection of unanswerable questions in SQuAD2.0

the final project (due Feb 21)

THE PROJECT GOAL

- improve the ability of LLMs to handle unanswerable questions
- the task: content-grounded question answering

- context: <short passage>
- question: <question on the passage>
- answer: <model answer (grounded in the passage)>

CONTENT-GROUNDED QA – EXAMPLES

Context: Many types of Turing machines are used to define complexity classes, such as deterministic Turing machines, probabilistic Turing machines, non-deterministic Turing machines, quantum Turing machines, symmetric Turing machines and alternating Turing machines. They are all equally powerful in principle, but when resources (such as time or space) are bounded, some of these may be more powerful than others.

Question: What are two factors that directly effect how powerful a Turing machine can be?

Correct answer: time or space

Possible (incorrect) answer: resources

Possible (incorrect) answer: NO ANSWER

…

CONTENT-GROUNDED QA – EXAMPLES

Context: Many types of Turing machines are used to define complexity classes, such as deterministic Turing machines, probabilistic Turing machines, non-deterministic Turing machines, quantum Turing machines, symmetric Turing machines and alternating Turing machines. They are all equally powerful in principle, but when resources (such as time or space) are bounded, some of these may be more powerful than others.

Question: What machines are not equally powerful in principle?

Correct answer: NO ANSWER

Possible (incorrect) answer: asymmetric Turing machines

Possible (incorrect) answer: …

USEFUL LINKS

- a relevant paper on the dataset (read through it)
- the dataset page on huggingface (additional useful details)
  - SQuAD2.0 – Stanford Question Answering Dataset
- the LLM we are using in this project on huggingface
  - meta-llama/Llama-3.2-3B-Instruct
  - can run on your laptop

PROJECT STRUCTURE – data

- dev-v2.0.json – the full dev SQuAD2.0 file
- squad2.0-dev-1000.csv – its 1000-sized subset in the csv format
- squad2.0-dev-1000-sample-results.csv – sample results file
  - your output on a sample will be printed into this file

PROJECT STRUCTURE – code

- utils: evaluate_results.py (utility function for evaluation)
- utils: query_model.py (example code for querying the model)
  - make sure this code runs smoothly for you without any additional definitions, e.g. no environment variables
- config.json – project configuration files
- main.py – the main file

PROJECT STRUCTURE – instructions

- make sure you can run query_model.py smoothly
- go through the code in evaluate_results.py and understand it
- run main.py and make sure it reports some sample stats
  - look up this metric on the web and understand its components

- your task is to achieve the highest evaluation results possible
  - increasing significantly non-answerable accuracy
  - but not compromising answerable accuracy too much
  - with the main focus on “exact” and “f1”

PROJECT STRUCTURE – instructions

- during your work you should rely on the entire set of 1000 examples
- the project will be tested on multiple smaller samples (e.g., 50)

- you cannot make use of another LLM for this task (asking the same question)
- make sure a small sample results in a reasonable runtime
- do not change the main() – it will be run as-is

- do not hesitate to think out-of-the-box, come up with creative solutions

PROJECT STRUCTURE – submission

- (1) main.py with implementation of the **squad_qa()** function
- (2) your best attempt at solving the entire **squad2.0-dev-1000.csv**
  - **squad2.0-dev-1000-results.csv** with the “final answer” column
- (3) a 3-4 pages pdf document with the full report of your work
  - your directions for solution – things that worked and those that did not
  - report of your final accuracy on the attached 1000 dev sample
  - error analysis – your insights based on inspecting the results
    - what is easy? what seems to be difficult?
  - expected runtime on a 50-sized sample

GOOD LUCK!

- you can use libraries we did not mention at class
  - when at doubt – ask me
- we do not put strict runtime restrictions, but faster solutions will be scored higher

- enjoy your journey :-) that's the type of challenges contemporary NLP researchers face

- start early! do not leave the project to the last moment