Datasets

Kaggle: https://www.kaggle.com/rtatman/questionanswer-dataset
SQuAD: https://rajpurkar.github.io/SQuAD-explorer/

The datasets in Kaggle has more columns other than question and answer such as ArticleTitle and difficulty from question/answer, which might be useful in later model selection stage (for example, for simpler questions, model A might have a better performance but for harder questions, model B might have a higher accuracy).

The datasets in SQuAD contain unanswerable questions, which Kaggle datasets lack. Also it provided an evaluation script and a leaderboard so we can easily know how good or bad our model is compared to other submitted models.

Methodology

Data Preprocessing

The Kaggle dataset contains more columns than the SQuAD dataset does as mentioned above. Whether to keep those columns remains to be explored. The SQuAD dataset is in .json format, thus we need to use the json library. Overall, other than common preprocessing steps such as tokenization, n-grams and so on, there won't be a lot of additional preprocessing work because both datasets are in a clean and neat format.

Machine Learning Model

We aim for accuracy and simplicity. That said, Python open source libraries would be our first choice. We could start with LSTM, and if time and technology permits, we could opt for transformers.

Evaluation Metric

The accuracy of our model can be evaluated using the evaluation script SQuAD provides. Also, it can be compared to that of other submissions.

Final Conceptualization

We plan to showcase our project using a Flask web app. In it, the users can provide a passage and ask some questions about that passage. It can be extended to be the core model of many real life applications like a customer service bot.

Visualization/Application

This project can power an online Q&A platform where users provide the passage and question and our AI gives the answer to the question. It would greatly reduce the time needed for users that know precisely what they are looking for to scan through trivial information in a news article or journal. Therefore, the front-end of this project should take uploaded files, parse them into texts, take the reader's question, and return the predicted answer. We expect to use Flask to host the web app front end.