Project 4 – CS4740/5740 Introduction to NLP
Fall 2018
Machine Reading Comprehension

Proposal (Optional): due by Tuesday, 11/27 11:59pm
Final version: due by Tuesday, 12/04 11:59pm

1 Overview

Goal for this project: To gain a bit of experience with the task of machine reading comprehension (MRC). To gain experience in working with standard off-the-shelf NLP components and/or machine learning or neural network components.

You are to apply NLP knowledge to implement a MRC system that will work on a real world challenge task: the **S**tanford **Qu**estion **A**nswering **D**ataset (SQuAD) v2. (See [1] if you are interested in the details of the full task.) We will only be doing part of the full MRC task. In particular, input to the system is a passage with a couple of questions. For each question, the training data contains either an **Answer** or a **Plausible Answer**. The latter is provided for questions that are not answerable given the accompanying text passage. The full MRC task requires providing the answer for answerable questions and identifying the remainder as unanswerable. For this assignment, we will only be attempting the second task: for each question, the output of your system should be a binary value indicating whether the question is answerable (a value of 1) or not answerable (a value of 0). No human intervention is allowed in deriving answers. Below is an example (from the training data):

Article: *Endangered Species Act*
Paragraph: *" . . . Other legislation followed, including* the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society — the species were relatively rare — and little opposition was raised."
Question 1: *"When was the Bald Eagle Protection Act passed?"*
Answer: *1940*
Question 2: *"Which laws faced significant opposition?"*
Plausible Answer: *later laws*
Question 3: *"What was the name of the 1937 treaty?"*
Plausible Answer: *Bald Eagle Protection Act*

In the Figure, red indicates plausible answers, which are actually incorrect. Relevant keywords are shown in blue.

2 Data

We provide you with the following files:

- **training.json** The training set.
- **development.json** The development set.
- **testing.json** The test set.
- **sample.json** A sample prediction file for the development set.
- **evaluate.py** The evaluation script. To run it, execute

  ```
  python evaluate.py <path_to_data> <path_to_predictions>
  ```

  For example, you can evaluate the sample prediction file with the following command:

  ```
  python evaluate.py development.json sample.json
  ```

Please note that the training, development, and test sets for the project are **different** from the official ones at Stanford, so you need to download the datasets from CMS.

3 Project Phases

3.1 Proposal (Optional)

For this part, you need to:

- Design and implement a baseline system for the task. (At this point in the semester, you should know what a baseline system is . . .) There are many options. Implement something very simple! (For the final version of the project you will probably want to implement additional (better) baselines to which you can compare your final system.)
- Run, evaluate and analyze the behavior and performance of the baseline system on the training and development sets.
- Develop a plan for the final system.

In the proposal, you should provide a short description of the baseline system and a justification for why you chose it. Include performance results on the training and development sets as well as an analysis of the results. Finally, describe your plans for implementing your final system.

3.2 Final System

The goal for this part is to:

- Implement the final system that distinguishes answerable vs. unanswerable questions in a MRC context.
- Run, evaluate and analyze the behavior and performance of your final system on the development and test sets. Predictions on the test set will be submitted to **Kaggle**. Details regarding the Kaggle competition will be posted on Piazza.

4 What to Submit

4.1 Proposal (Optional)

- Code for at least one baseline system. Should include README file explaining how to run code. Submit to CMS.
- Write-up satisfying the requirements in section 3.1. Submit to Gradescope.

4.2 Final System

- Source code with adequate comments and executables. Submit to CMS.
- Prediction file produced for the test set. Submit to CMS.
- Report. Submit to Gradescope. The report should contain:
  - Names and netid of all group members.
  - Kaggle team name.

- Description of your system including the motivation behind its design. We want to be able to understand what your system does, how it does it, and why you designed it the way that you did.
- Description of the baselines you compare your system to including the motivation behind their design. We want to be able to understand what the baseline(s) do, how they work, and why you selected them.
- Performance of your system vs. your baseline comparison system(s) on the development and test sets. The latter will be provided via the Kaggle competition.
- Analysis of the results. What worked? What didn't work? Examples are useful here. In addition, you can conduct "ablation" studies to see the effect of various aspects/components of the system on performance. E.g., obtain results after removing each component in turn; or after removing features (if you use a machine learning solution).

5 Grading Rubric (Tentative)

Guidelines: Getting extremely high performance is not required in this assignment. We expect that **the system and report reflect a serious effort at trying to build a good system for this MRC task using knowledge about NLP that you have gained this semester**.

5.1 Proposal (0 pts)

We will provide quick feedback (1) on whether the planned final system looks too difficult/too easy to implement given the time available and the number of students on your team; (2) if the baseline approach or its evaluation appears flawed.

5.2 Final System (100 pts)

5.2.1 Basic functionality (10 pts)

- (5 pts) **Baseline system** If baseline system can generate the answer file in the correct format.
- (5 pts) **Final system** If final system can generate the answer file in the correct format.

5.2.2 Report (60 pts)

- (3 pts) Names, netids, Kaggle team name.
- (15 pts) **Baseline system(s)** description, evaluation and analysis of results as described above.

- (40 pts) Description, evaluation and analysis of results of the **final system** as outlined above. Make clear which components of the system you built yourself vs. those you downloaded from elsewhere or got from another student in the course.

- (2 pts) **Individual member contribution**. Include a section describing how each member contributed to the project.

5.2.3 Design and implementation quality (30 pts)

- **Poor (∼7 pts)** The design and implementation is not much more advanced in terms of NLP/ML techniques than the baseline.

- **Average (∼14 pts)** The design and implementation is acceptable in terms of the NLP/ML techniques employed.

- **Good (∼21 pts)** The design and implementation show a better than average amount of work for this project in terms of the NLP/ML techniques employed.

- **Excellent (∼30 pts)** The submission contains more ideas and/or work than most submissions for this project and goes the extra mile.

References

[1] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.