CS-5340/6340 Project Description, Fall 2022

The NLP class project will be to design and build a question answering system. For this project, we will use stories that were collected from the Canadian Broadcasting Corporation web page for kids. The CBC published five current events stories a week for over two years that targeted elementary and middle school students. Figure 1 shows a sample story.

HEADLINE: An Arctic Struggle
DATE: June 29, 1999
STORYID: 1999-W27-2

TEXT:

A group of 50 beluga whales is fighting to stay alive in an icy trap in the Canadian Arctic near Ellesmere Island.

An unexpected freeze has left dozens of the whales trapped in a sea of ice, with one small hole as their only window for air. The open sea is 20 kilometres away.

About 20 of them have already died, despite the best efforts of wildlife officials. The reason: polar bears. With the whales swarming their only air hole, they've become easy prey for the bears.

The polar bears are hunting the belugas at will. Even in the best of conditions, belugas can only spend about 10 minutes underwater. When they come up, the bears jump onto the whales and tear chunks out of them.

Unless the ice breaks up soon, the belugas' chances of survival are grim.

Beluga whales live only in the arctic and subarctic. They live in the Arctic Ocean and its adjoining seas, including the Sea of Okhotsk, the Bering Sea, the Gulf of Alaska, the Beaufort Sea, Baffin Bay, Hudson Bay, and the Gulf of St. Lawrence.

Figure 1: A story

Two people at the MITRE Corporation created questions and an answer key for each story. (One of these people had professional experience writing questions for reading comprehension exams.) Figure 2 shows the answer key for the story above. Each question has a unique *QuestionID* which is the *StoryID* followed by a dash and question number. For example, "1999-W27-2-1" refers to question #1 pertaining to story "1999-W27-2". The questions have difficulty ratings assigned to them, as judged by the person who created the question. For this project, all of the questions will have a difficulty rating of "easy" or "moderate". Note that the QuestionID numbers may not be consecutive because questions that had a high difficulty were removed.

In some cases, the answer key allows for several acceptable answers (e.g., "Toronto, Ontario" or "Toronto"), paraphrases (e.g., "Human Immunodeficiency Virus" or "HIV"), varying amounts of information (e.g., "he died" or "he died in his sleep of natural causes"), or occasionally different interpretations of the question (e.g., "Where did the boys learn how to survive a storm?" "camping tips from a friend" or "their backyard"). When more than one answer is acceptable, the acceptable answers are separated by a vertical bar (|).

```
QuestionID: 1999-W27-2-1
Question: Where in the Canadian Arctic are the 50 beluga whales trapped?
Answer: near Ellesmere Island
Difficulty: easy

QuestionID: 1999-W27-2-2
Question: Why have the whales become trapped?
Answer: an unexpected freeze
Difficulty: moderate

QuestionID: 1999-W27-2-3
Question: Through what are the whales breathing?
Answer: one small hole in the ice | one small hole
Difficulty: moderate

QuestionID: 1999-W27-2-4
Question: Who is trying to save the whales?
Answer: wildlife officials
Difficulty: moderate

QuestionID: 1999-W27-2-5
Question: Even in the best of conditions, how long can beluga whales stay under water?
Answer: only about 10 minutes | about 10 minutes
Difficulty: moderate

QuestionID: 1999-W27-2-6
Question: How are the bears killing the whales?
Answer: the bears jump onto the whales and tear chunks out of them
Difficulty: moderate
```

Figure 2: The answer key for story 1999-W27-2

Judging answers is subjective, so you may occasionally disagree with MITRE's answers. But people will never completely agree on this kind of thing, and it is necessary to choose some set of answers for evaluation purposes, so we will stick with MITRE's judgements as the gold standard.

The Task: Your team must build a question answering (Q/A) system that can process a story and a list of questions, and produce an answer for each question. Each team will consist of 1-2 people. Each team's system **must** conform to the following input and output specifications, but other than that you can design your system however you want!

The Input

Your Q/A system should accept a single input file as a command-line argument. We should be able to run your program like this:

```
qa <inputfile>
```

The first line of the input file will be a directory path. Each subsequent line in the file will be a StoryID. Your Q/A system should then assume that for each StoryID, the directory contains a story file named StoryID.story (e.g., "1999-W02-5.story") and a question file named StoryID.questions (e.g., "1999-W02-5.questions"). Your Q/A system should produce an answer for each question in the question file, based on the corresponding story file. A sample input file is shown below.

```
/home/cs5340/project/developset/
1999-W02-5
1999-W03-5
1999-W04-5
1999-W05-4
1999-W05-5
1999-W06-5
1999-W07-5
1999-W08-1
```

Each story file will be formatted like Figure 1. Each story will include a Headline, Date, and StoryID line, followed by the text of the story.

Each question file will contain 3 lines for each question indicating the QuestionID, the question itself, and a difficulty rating. The question file will be formatted like Figure 2, except that there will not be an answer line. For example, it would look like this:

```
QuestionID: 1999-W27-2-1
Question: Where in the Canadian Arctic are the 50 beluga whales trapped?
Difficulty: easy

QuestionID: 1999-W27-2-2
Question: Why have the whales become trapped?
Difficulty: moderate
```

The Output

Your Q/A system should produce a single **Response File**, printed to standard output, which contains the answers that your system finds for all of the stories and questions in the input file. The output of your system should be formatted as follows:

```
<QuestionID>
<Answer>

<QuestionID>
<Answer>

...
```

For example, your output should look like this:

```
QuestionID: 1999-W02-5-1
Answer: Canada

QuestionID: 1999-W02-5-2
Answer: Betty Jean Aucoin

QuestionID: 1999-W02-5-3
Answer: a fitness club

QuestionID: 1999-W03-5-4
Answer: 5 feet, 2 inches

QuestionID: 1999-W03-5-5
Answer:

QuestionID: 1999-W03-5-6
Answer: 502

QuestionID: 1999-W02-5-7
Answer: $135

QuestionID: 1999-W03-5-2
Answer: Edmonton

QuestionID: 1999-W03-5-3
Answer: forward

...
```

IMPORTANT: Your response file should have a QuestionID and Answer for **every question** in **every story** specified by the input file, in **exactly the same order**. Also, be sure to print each answer on a single line. If your Q/A system can't find an answer to a question (or your system chooses not to answer a question), then **leave the answer blank**, as done for Question 1999-W03-5-5 above.

The Data Sets

You will be given three sets of data at different points in the project.

Development Set: 73 stories and answer keys

Test Set #1: 39 stories and answer keys

Test Set #2: 39 stories and answer keys

Project Phases

The project will involve three phases:

Development Phase: A **Development Set** is available on CANVAS for you to use when creating your Q/A system. You may use these stories and the answer keys in any way that you wish.

In addition, we will give you the scoring program that we will use to evaluate your Q/A system. You can use this scoring program to assess the performance of your system yourself as you experiment with different ideas. The arguments that it takes are described at the beginning of the file.

Midpoint Evaluation: There will be a midpoint evaluation of everyone's Q/A systems. Each team will submit the source code for their Q/A system and we will evaluate each system on a new data set called **Test Set #1**. The purpose of this evaluation is to make sure that every team is making progress on creating a Q/A system and to allow everyone to see how other teams are performing at the (roughly) halfway point.

Once the midpoint evaluation is over, we will release Test Set #1 so that you can improve the performance of your system on those stories.

Final Evaluation: For the final evaluation, each team will submit the source code for their Q/A system. We will run your Q/A system on a new data set called **Test Set #2**.

Evaluation Metrics

The performance of each Q/A system will be evaluated using the F-measure statistic, which combines recall and precision in a single evaluation metric. Since Q/A systems often produce answers that are partially but not fully correct, we will score each answer by computing the *Word Overlap* between the system's answer and the strings in the answer key. Given an answer string from the answer key, your system's response will be scored as:

Recall (R): the number of correct words generated by your system divided by the total number of words in the answer string.

Precision (P): the number of correct words generated by your system divided by the total number of words generated by your system.

F-measure: $\mathbf{F}(\mathbf{R}, \mathbf{P}) = \frac{2 \times P \times R}{P + R}$

This formula tries to find a good balance between recall and precision. (It is the harmonic mean of recall and precision.) *The final performance of each system will be based on its F-measure score.*

As an example, suppose your system produces the answer "Elvis is great" and the correct answer string was "Elvis Presley". Your system's answer would get a recall score of 1/2 (because it found "Elvis" but not "Presley"), a precision score of 1/3 (because 1 of the 3 words that it generated is correct), and an F-measure score of .40.

Two important things to make a note of:

- This scoring measure is not perfect! You will sometimes receive partial credit for answers that look nothing like the true answer (e.g., they both contain "two" but all other words are different). And you may sometimes get a low score for an answer that seems just fine (e.g., it contains additional words that are related to the true answer, but these extra words aren't in the answer key). Word order is also not taken into account, so if your system produces all the correct answer words, but in the wrong order, it doesn't matter – your system will get full credit! Automatically grading answers is a tricky business. This metric, while not perfect, is meant to be generous and give your system as much partial credit as possible.
- The answer key often contains multiple answer strings that are acceptable for a question. Your system will be given a score based on the answer string that mostly closely matches your system's answer.

Schedule

October 19: Fill in the Team Request Form on Canvas!

November 8: Midpoint evaluation on Test Set #1.

November 29 (by noon!): Final evaluation on Test Set #2.

December 5, 7: In-class project presentations.

December 9: Final project slides due.

Grading

Each project will be graded according to the following criteria:

- 30% of the grade will be based on the performance of your Q/A system on Test Set #1 during the midpoint evaluation.
- 65% of the grade will be based on the performance of your Q/A system on Test Set #2 during the final evaluation.
- 5% of the grade will be based on your project presentation. The 10 top-performing teams will give an in-class presentation of their Q/A system, and the remaining teams will need to prepare a set of slides that describe their Q/A system but there will not be an oral presentation. All teams will also be required to submit their presentation slides for grading.

To determine the grades for the midpoint and final evaluations, teams will be ranked based on the performance of their system relative to the other Q/A systems. The teams will then be clustered (manually) so that teams whose systems produced similar scores will get similar grades. It is fine to share ideas with other teams (but not code!), and to compare your system's performance with other teams. But if your team is doing almost exactly the same thing as many other teams, chances are your system will end up in the middle of the rankings. To distinguish yourself and stand apart in the rankings, we encourage teams to try different things!

IMPORTANT: I will ask each team to document the specific software contributions of each team member. This will allow me to adjust individual grades in case one person puts in substantially more effort than the other. If all teammates contribute (roughly) equally to the project (as I expect in most cases), then they will get the same project grade. But if one team member does substantially less software development, then they will get a lower grade. Also, please be advised that if I detect an extreme imbalance where one team member is contributing extremely little to the project (in terms of software development), then I reserve the right to split up the team after the midpoint evaluation. In that case, each person would need to work on the project independently for the final evaluation. I hope that this will not be necessary.

The final grading will be based on how well your system does relative to other team's systems, but it is not the case that the highest ranked system will get an automatic 'A' or that the lowest ranked system will get an automatic ‘E’. If every team produces a system that works well, then I will be happy to give everyone an ‘A’ on the project! If, at the other extreme, no one generates a system that works at all, then I would have to give every team a failing grade. I hope that the competitive spirit will energize everyone to work hard and produce interesting and effective Q/A systems so that I can give many teams a high grade!

External Software & Data

You may use external software packages and data resources for your project, as long as the following criteria are met:

- **You may NOT use any external software that performs question answering or a Q/A-specific subtask such as question type classification!** If we discover that your submitted system uses any external system or code that performs a Q/A task, then you will be disqualified and get a zero for the project.

- **You may NOT use any pre-trained neural language models (LMs) EXCEPT one specific model that we will indicate is allowed (to be announced soon).** The primary reason for this is to level the playing field for everyone. Some large LMs have absorbed so much language that using them in a simple way could yield substantially higher F-scores than using other methods. Giving people the freedom to use any LM that they can find could (1) make some of you feel that you *must* use large LMs to achieve a competitive score in the class, and (2) lead to an arms race where some people spend all their time searching for the best LM. There are also substantial computational issues with using large LMs on the CADE machines, and grading challenges if we allow code to be run from different platforms. So, one specific language model will be approved for use in this project if some people want to use this type of approach. But no one should feel obligated to use it. We will be ranking the systems that use the approved LM separately from those that do not, so that there will not be competition between the LM-based systems and other types of systems.

- You **MUST** fully acknowledge in your final presentation slides all of the external software and data resources that you used in your system.

- **We MUST be able to run all of your software (your own and external resources) on the linux-based CADE machines.** This means either including the software in your code submission, or installing it in your own CADE directory and giving us full permission to access it from there. Feel free to discuss options with the TAs.

- You MAY use external NLP software that performs basic NLP functionality, including tokenizers, sentence splitters, part-of-speech taggers, syntactic parsers, named entity recognizers, coreference resolvers, and general-purpose semantic dictionaries such as WordNet. **If you are uncertain about whether a specific resource is acceptable to use, please ask the instructor!**

Machine Learning

You do not need to use machine learning (ML) for this project. But you are welcome to do so if you wish. If you choose to use ML:

- You MAY use *general-purpose* external ML software packages, such as scikit-learn. For example, you can train your own logistic regression classifier or SVM classifier.
- You may NOT use any ML models that have been previously trained for question answering or a Q/A-specific subtask such as question classification.
- You may NOT re-train a system that was designed for another task using our cs5340/6340 data. For example, suppose you obtain software for Named Entity Recognition. You are permitted to use that software for Named Entity Recognition. But you are NOT allowed to try to re-purpose it for Question Answering by re-training it on Q/A data. Any ML module that you create for Q/A needs to be designed by you.
- You MAY use the Development Set as training data for both the Midpoint and Final Evaluations. You may use Test Set #1 as additional training data for the Final Evaluation.
- If you create your own ML model for question answering, you must train it yourself using ONLY the cs5340/6340 data. You may NOT use **any** additional sources of training data. If you submit an ML model that has been trained with external data, your Q/A system will be disqualified and you will get a zero for the project. The reason is to level the playing field for all teams in the class, so that everyone is using exactly the same data set to build their Q/A systems.

CAVEAT AND ENCOURAGEMENT

Building an effective question answering system is hard! Question answering is not a solved problem in NLP, so you shouldn't expect your Q/A systems to produce super-high scores! Just try to design a Q/A system as best you can to perform reasonably well on the cs5340/6340 data.

I encourage everyone to have fun with this project and experiment with lots of ideas. Creativity is appreciated! I chose this project because question answering is an important NLP application. Right now, it is also a hot research area, but hopefully in the next 10 years the technology will be good enough to start incorporating Q/A into real products. This project will give you exposure to a cutting edge research area, understanding of an important application area for NLP, the experience of building a real NLP system, and the opportunity to explore your creative side!