ROC Story Cloze Shared Task

Introduction:

The project for this class will be to design, build, and evaluate a system for solving a Story Cloze task, a commonsense reasoning task. Enabling AI systems with commonsense knowledge allows them to handle many ambiguous scenarios better, as well as properly track not only logical relationships being generated from the text, but also temporal relationships, and more. We will be focusing on this problem through tackling the Story Cloze task created by Mostafazadeh et al [1].

Cloze tasks are used to determine whether a human or a system is able to understand a language by deleting a random word from a sentence and asking the test-taker to fill in the blank [2]. The Story Cloze task represents the same idea but focuses on short stories and finding the correct ending to that story from two given endings. Consider the example below from the authors of this data set.

Context	Right Ending	Wrong Ending
Jim got his first credit card in college. He didn’t have a job so he bought everything on his card. After he graduated he amounted a $10,000 debt. Jim realized that he was foolish to spend so much money.	Jim decided to devise a plan for repayment.	Jim decided to open another credit card.

Our context here is a short story of 4 sentences and we are given a correct ending and an incorrect ending, and it is our task to figure out which of the two sentences is the correct one. This task is still an open problem and so it is not expected that you achieve perfect results. Furthermore, this task should be treated as a research project, which means finding papers that attempt to address this problem and considering them as your team decides on what needs to be done.

Your team will need to create a plan on how you wish to address this problem. Much like in Homework 1’s task 2, this plan should be grounded in research. You are welcome to reproduce a strategy described by a paper or come up with your own, basing it off of the research you have read, but be sure to cite your decisions appropriately. As a starting point, read the corpus paper [1] and the shared task paper [3] that are already supplied to you to see what strategies have already been done.

Data Format and management:

The data will be given to you in a csv file with the following columns

- Input Story ID: an ID used to represent and label the story
- Input Sentence 1-4: the context sentences that represent the start of the story
- RandomFifthSentenceQuiz1-2: the ending sentences
- AnswerRightEnding: a number of which sentence is the correct ending

You will be given a subset of the corpus to use on Courseweb. Since the data was not in the format described above, we transformed the data that Mostafazadeh et al [1] so that it is easier to use. We also shuffled some of the data around and the files supplied to you are described below.

- ROC-Story-Cloze-Data.csv – the modified training data following the above format containing 45,496 four sentence stories with a missing ending.

- ROC-Story-Cloze-Val.csv – this is a shuffled validation set that should be used if your approach needs hyper-parameter tuning
- ROC-Story-Cloze-Test-Release.csv – this is a shuffled test set where I have removed the correct label, this is given to you so that your model can produce predictions for us to verify. We describe this process a little later.
- evalScript.py – the evaluation script that computes accuracy given two files, has an optional argument "-v or --verbose" that will print out which labels did not match.
- sampleGold.txt – a sample prediction file with each gold label on a new line to show the formatting needed for the evaluation script
- samplePrediction.txt – a sample prediction file with each predicted label on a new line to show the formatting needed for the evaluation script

Even though we have given you data splits, it is up to your team to determine how you want to use that data to create a model that addresses this task. For example, you can do cross-validation on the training data to help select the best model using the validation to determine the value of your metric. You could also split the training data even further into a training and test set for internal verification. You should appropriately select your data management strategy based on the method you decide for handling this task.

The data you are getting is not exactly the same data as we have modified and transformed the data. The test data supplied is what we will use to evaluate your model and is given to you just so you can see and process it without potentially facing any parsing issues. At no point should you use the test data for any part of the learning process. After you submit the code, we will release the test data with true labels so that you can evaluate your model and place those results in the report and the presentation.

We are also supplying an evaluation script and two sample input files for this evaluation script. When you make predictions on your test data please output the results to a text file that follows the same format as the samplePrediction and sampleGold files, which has each line containing a single prediction. This will simplify the evaluation process when the TA verifies your models, described more below. This evaluation script is the same one that the TA will be using to verify your results, so make sure that your predictions on the test set is saved to a file. This evaluation script produces the accuracy between the two inputted files.

Time-line and responsibilities:

For this project you will need to be submit a couple of things as you progress.

1) On Thursday, November 7th a one page description of how you plan on tackling this problem

This needs to be a clear description of your approach giving a basic explanation of what the approach is as well as a description of your experiment. This paper should include citations to papers motivating your approach, using some citation style. Be sure to consider how you will compare your attempt and show that it does perform well. Finally, be sure to detail how you plan to manage the data that was given to you. This will be used as a checkup to make sure that all of the groups are on track. Also, you are not required to stick to the plan you describe in this checkup.

2) On Sunday, December 8th the code that will perform this task

In order to verify your results, the TA will be running your models on the test set and evaluating them against the true labels using the provided evaluation script. Therefore, please make sure that you save your model in a usable format and save your model's predictions into a text file that matches the format used in the sample files given to you. This also means you should test your preprocessing against the test data and ensure there are no parsing issues. Also, provide a description of how to use your code and any dependencies so that we can reproduce your results, including the training process. Finally, part of your grade will be based on how your performance ranks against the rest of the students.

3) On Tuesday, December 10th a report of your results.

Your report should both describe your system (the architecture, components, etc.) and contain a discussion evaluating how well your model performed from the data you were given and the final evaluation (using the provided programs to compute performance). Your report should also discuss the experiment that was done to produce your results. Once again this report should cite papers that motivated your approach. This report should be structured much like a paper one would submit to an NLP conference. Papers should be NO LONGER THAN 4 pages (excluding references) using a LaTeX or Word Template found on ACL's website. http://www.acl2019.org/EN/call-for-papers.xhtml (it is about half way through the page)

4) On Thursday, December 12th from 1:00pm to 3:50pm, presentations showcasing what your attempt was at addressing this task, and what your results are.

We have allocated 2.5 hours for presentation with a 20 min break in between.
This will be 7-8 minutes long for each presentation with 3 minutes for questions from the class.
This might be updated depending on .

Project grade Breakdown:

5%: Submitted the one page checkup paper.
55%: Submitted the code, a description of how to use it, and how your performance compares with the rest of the groups.
25%: The report
15%: The presentation

Tips:

- One advice we can give for how much information to include or exclude in your papers is to consider your classmates as your audience. Include enough information such that any of your classmates outside of your group can understand what you are trying to do and if they come back with questions, you should consider addressing them in the paper.
- One way to make the final Report writing simpler is to extend off of the checkup paper that you submitted earlier.

References:

[1] N. Mostafazadeh *et al.*, "A corpus and evaluation framework for deeper understanding of commonsense stories," *CoRR*, vol. abs/1604.01696, 2016.

[2] W. L. Taylor, "'Cloze procedure': A new tool for measuring readability," *Journalism Bulletin*, vol. 30, no. 4, pp. 415–433, 1953.

[3] N. Mostafazadeh, M. Roth, A. Louis, N. Chambers, and J. Allen, "LSDSem 2017 shared task: the story cloze test," in *Proceedings of the 2nd workshop on linking models of lexical, sentential and discourse-level semantics*, Valencia, Spain, 2017, pp. 46–51.