

EECS595: Natural Language Processing

Fall 2020

Final Project Description

(Released September 30, 2020)

1. Introduction

Recent years have seen an increasing amount of work that attempts to address commonsense reasoning for language understanding. Many benchmarks have been developed for this purpose. Here is a survey paper on the recent progress in this area:

[Recent Advances in Natural Language Inference: A Survey of Benchmarks, Resources, and Approaches](#). S. Storks, Q. Gao, and J.Y. Chai. arXiv: 1904.01172.

Now, it's time to get your hands on some of the benchmarks driving this progress.

In this final project, you will form a team of three (3) to work on the project. You will need to get my approval if you choose to form a team of 2 or 4 members. A sign-up sheet for team forming is set up in the course Google Drive, under the Final_Project_Common subdirectory.

<https://drive.google.com/drive/folders/0ACOkKkIdRJVmUk9PVA>

You will need to use your <username>@umich.edu Google account to access the sheet. The deadline for forming the team is **October 9**.

[Sign up here!](#)

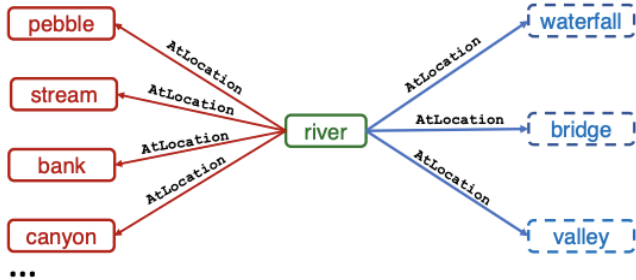
2. Detailed Tasks

Each team will work on three benchmarks: [CommonsenseQA](#), **Conversational Entailment**, and **Everyday Actions in Text (EAT)**. These datasets are chosen to provide a variety of challenges:

1. They contain different magnitudes of data, which may motivate different types of approaches to the problems.
2. They've been publicized to different degrees, varying the number of existing approaches and thus progress on the benchmarks so far.
3. They are formulated in different ways, and focus on various domains of commonsense reasoning for language understanding.

Note that EAT is a dataset from the SLED (Situating Language and Embodied Dialogue) group curated by Shane Storks. The dataset has not been publicly released yet. **Please do not share the dataset with people outside of this class.**

2.1. [CommonsenseQA](#)

Task Definition
Given a natural language question (based on relations in ConceptNet), choose the correct answer out of 5 natural language choices (5-way text classification).
Example
<p>a) Sample ConceptNet for specific subgraphs</p>  <p>b) Crowd source corresponding natural language questions and two additional distractors</p> <p>Where on a river can you hold a cup upright to catch water on a sunny day?</p> <p>✓ waterfall, ✗ bridge, ✗ valley, ✗ pebble, ✗ mountain</p>
Evaluation Metrics
Accuracy (% correct) on validation set (as test data is withheld for the leaderboard). This result can be self-reported.
Relevant Papers
CommonsenseQA: A Question Answering Dataset Targeting Commonsense Knowledge . A. Talmor, J. Herzig, N. Lourie, and J. Berant. Conference on Empirical Methods in Natural Language Processing (EMNLP), Minneapolis, MN, June, 2019.
Where to Find Data
Use HuggingFace's nlp package in Python to access the dataset. Check the link for more information, including a demo on how to use the package in Colab.

2.2. Conversation Entailment

<i>Task Definition</i>
Given a short natural language dialogue and a hypothesis sentence, infer whether or not the hypothesis can be inferred from the dialog (binary text classification).
<i>Example</i>
<p>Dialogue Segment:</p> <p>A: And where about were you born? B: Up in Person Country.</p> <p>Hypothesis:</p> <p>(1) B was born in Person Country. (2) B lives in Person Country.</p>
<i>Evaluation Metrics</i>
Accuracy on the test set (withheld by us). You can split the training data as needed for cross-validation. <i>Look out for more information on how your code will interface with our test set evaluation.</i>
<i>Relevant Papers</i>
<p>What do We Know about Conversation Participants: Experiments on Conversational Entailment. C. Zhang and J. Y. Chai. SIGDIAL 2009 Conference, London, UK, September, 2009.</p> <p>An Investigation of Semantic Representation in Conversation Entailment. C. Zhang and J. Y. Chai. Conference on Empirical Methods in Natural Language Processing (EMNLP), MIT, MA, October, 2010.</p>
<i>Where to Find Data</i>
The training data will be available on the course Google Drive under <i>Conversational_Entailment</i> . Please make a copy for your local Google Drive.

2.3. EAT (Everyday Actions in Text).

<i>Task Definition</i>
Given a short natural language story, determine (1) whether the story is physically plausible, and (2) if implausible, which sentence is the breakpoint, i.e., the sentence after which the story stops making sense? Both sub-tasks will be formulated as classification problems (more information below).
<i>Example</i>
<ol style="list-style-type: none">1. John opened the cabinet.2. John took out a pan.3. John put a potato in the pan.4. John mashed the potato thoroughly.5. John cut the cooked potato in half. <p>Plausibility: <i>implausible</i> Breakpoint: 5</p>
<i>Evaluation Metrics</i>
<p>Accuracy for Sub-task 1, and macro-precision, recall, and F-measure for Sub-task 2 on the test set (withheld by us). You can split the training data as needed for cross-validation.</p> <p>For Sub-task 1, these metrics will simply be applied to the binary <i>plausible/implausible</i> predictions. For Sub-task 2, these metrics will be applied to the <i>breakpoint</i> predictions of each story. For a story predicted to be plausible, you should predict a label of -1, while for a story predicted to be implausible, you should predict a label between 2 and the number of sentences in a story as the breaking point can occur anywhere in the story except for the first one.</p> <p><i>Look out for more information on how your code will need to interface with our test set evaluation.</i></p>
<i>Where to Find Data</i>
The training data will be available on the course Google Drive under <i>EAT</i> . Please make a copy for your local Google Drive.

3. Programming Environment

We will be using [Google Colab](#) for the project, as it provides quick and free access to GPUs for faster experimentation. Using Colab is required, and we will need a .ipynb file for each of the models you create.

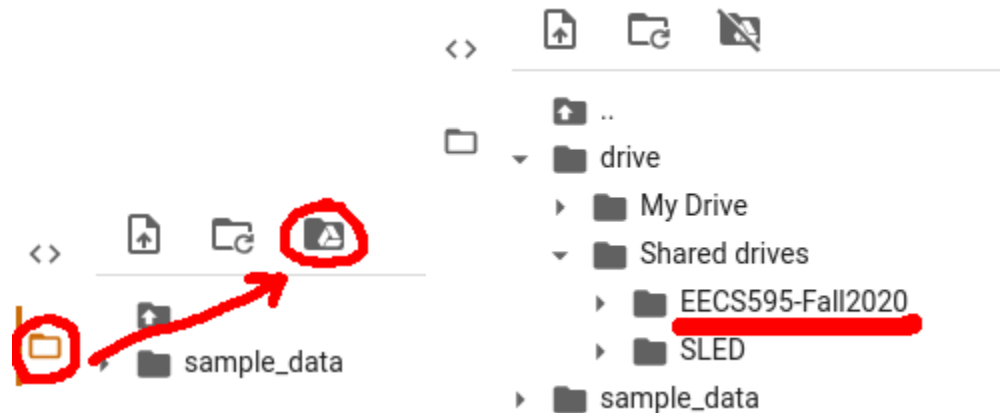
3.1 Opening Colab Files from UMich Google Drive

You may find it convenient to be able to open Python notebook files in Colab from Google Drive. However, this option may not be available by default on your UMich Google account. Follow the following steps to enable it:

1. Go to Colab at [this link](#).
2. Check the account displayed in the top right corner to verify that you are logged in on your UMich Google account. If not, cancel out of the pop-up menu and use the menu to switch to your UMich Google account.
3. If not already open, open the “Welcome to Colaboratory” notebook in the pop-up menu.
4. On the top left in the menu bar, select File > Save a Copy in Drive. This will install the Colab plugin to your Google Drive, and create a “Colab Notebooks” folder in your Google Drive.
5. Now, you can open any Python notebook file in Colab by right-clicking on the file in Google Drive.

3.2 Opening Google Drive Files from Colab

You will need to mount your Google Drive to Colab in order to access external files (e.g., your local copies of the datasets) from Colab. Use the “Google Drive” icon in Colab to get access to files in your Google Drive and the shared course drive:



3.3 Colab Tutorials

In the course Google Drive, you will find an example notebook with some basic examples for handling the datasets.

If you're new to Colab, here are some other tutorials:

- [Intro to Colab tutorial in Medium](#)
- [PyTorch and Colab tutorial from Justin Johnson's EECS 598 course](#)
 - Only follow steps 1-3; read through and run the cells in the notebook
 - Requires you to edit code in external .py files; don't do this on the project
- [Tutorial for fine-tuning BERT on an NLP dataset in Colab](#)

3.4 Models

As a team, you are required to implement one model in this environment for each task. These models could be a direct re-implementation of previously published models, or models designed by yourselves (need to make this clear in your final report). This does not mean the three models will have to be completely different. In fact, they may (and very likely) share some components.

BERT and other existing state-of-the-art NLP models are easily installable through Colab. See the last tutorial above for more information. A document further detailing how to use BERT can be found at:

<https://github.com/huggingface/pytorch-pretrained-BERT>

4. Evaluation

Please see descriptions above for the evaluation metrics that should be used for each benchmark, and the tasks and formulations they will be applied in. For more objective evaluation, we are withholding the test data on the Conversational Entailment and EAT benchmarks (training data should be split up for cross-validation as needed). As such, we will need to be able to consistently evaluate your models' Colab notebooks on the test data. *More information will be released on this issue soon.*

For each task, you should experiment with different model configurations and different hyper-parameter tuning and report your results. It's up to you to decide what goes into the final paper to show some results that are interesting and worth talking about.

The best performing system for each of these tasks will receive a prize at the end of the semester.

5. Presentation

Each team will present to the class their methods and results. Each team has 5 minutes. Since you are all working on the same problem, the presentation will only consist of methods and results. The presentation will take place the last week of the class: **December 2 and December 4 during class.**

6. Submission

You will need to submit the following items:

- Progress report
- Python code for all models and a README file explaining how the code works.
- Presentation slides
- Final report

All files should be submitted through Canvas. Everyone needs to submit a copy even though the same copy would be shared among team members.

6.1. Progress Report

A report that summarizes your progress on the project: what models (e.g., from what papers) you intend to apply to these problems; work division among the team; where you are in terms of implementation and getting the results. Limit it to two pages ACL style (see below).

Due date: November 4, 11:59pm EST

6.2. Model Code

The Python notebook files from Colab for each of your models must be submitted for evaluation on the withheld test data. *Look out for more information on how your code will need to interface with our test set evaluation.*

Python code may be submitted until the due date for the final report. However, in order to participate in the class competition, you must submit your code for the Conversational Entailment and EAT datasets a bit earlier. Competition participation will not affect your grade.

Due date for competition participation: November 29, 11:59pm EST

Final due date: December 16, 11:59pm EST

6.3. Presentation Slides

You need to submit your presentation slides.

Due date for presentation slides: December 4, 11:59pm EST

6.4. Final Report

You need to jointly write a report to comprehensively summarize your work. The report should include the following sections:

1. *Introduction:* The particular problems you are addressing (this section would be quite similar for all groups).

2. *Computational Models*: Clearly describe the models and implementations (including any hyper-parameters).
3. *Experimental Results*: Report the results you have for each of the tasks. Depending on the task, this will vary (see individual benchmark descriptions in Section 2).
4. *Discussion*: Summarize the insights you have gained through this exercise. You can do an error analysis here.

Due date: December 16, 11:59pm EST

All reports will need to use ACL format, which can be found:

<http://acl2020.org/downloads/acl2020-templates.zip>

Your final report should have at least 6 pages in ACL style.