**Unlimited Attempts Allowed**

Available after Oct 1, 2024 10:00amAvailable after Oct 1, 2024 10:00am

**Details**

Relation Extraction from Natural Language

The goal of this homework is to train deep learning models to determine knowledge graph relations (in this case, according to the Freebase schema) that are invoked in user utterances to a conversational system. You will be given a training set of utterances paired with a set of relations, that you can use to train models to predict the corresponding relations for a given utterance. Here is an example utterance from the dataset:

Show me movies directed by Woody Allen recently.

There are two relations that are invoked by this utterance:

1. movie.directed\_by
2. movie.initial\_release\_date

Here are the files you'll need:

[hw1\_test.csv](https://canvas.ucsc.edu/courses/76207/files/9560265?wrap=1)[Download hw1\_test.csv](https://canvas.ucsc.edu/courses/76207/files/9560265/download?download_frd=1)

[hw1\_train.csv](https://canvas.ucsc.edu/courses/76207/files/9560264?wrap=1)[Download hw1\_train.csv](https://canvas.ucsc.edu/courses/76207/files/9560264/download?download_frd=1)

[exampleSubmission.csv](https://canvas.ucsc.edu/courses/76207/files/9605161)

The goal is to use PyTorch to develop and train your own deep neural network models and output the associated set of relations when given a new utterance. You will have to try different techniques to improve upon a baseline model (which you need to define in your report). Here are some techniques you may try:

* Multilayer Perceptron (MLP)
* Different core features, such as bag-of-words, word embeddings (randomly initialized).
* Additional information derived from the text
* Different loss functions
* Different optimizers, such as stochastic gradient descent, AdaGrad, or Adam
* Preventing overfitting via dropout or weight-decay/regularization

In order to train a robust model, you will need to compare the performance of different models to identify their suitability. **Test out key hyper-parameters and find what works best**. Look at how similar problems are solved in research papers and try to implement those approaches. It may require some ingenuity in choosing features and also developing/picking the right models.

One more aspect to consider is that this homework requires you to do **multi-label classification**. You may reach a certain level of accuracy by predicting only a single class label, but to reach higher values (and potentially 100%), some examples require you to predict zero or multiple classes.

Dataset

This dataset is generated based on film schema of Freebase knowledge graph. There are three CSV files.

**hw1\_train.csv**

The file has three columns:

1. ID: the id for each row

2. UTTERANCE: the natural language text from you will extract relations

3. CORE RELATIONS: the relations invoked in the utterance.

**hw1\_test.csv**

This file is similar to hw1\_train.csv, but only contains the ID and UTTERANCE columns. Your model will predict the CORE RELATIONS.

**exampleSubmission.csv**

This file is an example of what your submission file should look like. It contains only the ID and CORE RELATIONS columns.

You may submit your predictions no more than 3 times each day before the competition deadline. You will get an opportunity to review your scores in the output logs before making a submission to the leaderboard. We advise you to start making submissions as early and as frequently as possible!

Competition

You will need to register on [KaggleLinks to an external site.](https://www.kaggle.com/t/6803dbf7c1c845d68355a983e7ae90ac" \t "_blank) (you need to use your ucsc email) to participate in the competition.

You may use the training data to train a model. It's recommended to use part of your training data as a validation set, but we've left that up to you. Using your models predictions on the test set, generate a submission file in the format of [example](https://canvas.ucsc.edu/courses/76207/files/9605161)[Submission.csv](https://canvas.ucsc.edu/courses/76207/files/9605161/download), and submit on Kaggle. **You can submit your predictions no more than 5 times every day** before the competition deadline. However, there is no limit on the total number of submissions you can make. Kaggle will automatically evaluate your performance.

The leaderboard on Kaggle displays your model's performance on the test set. The final results will be based on your highest scoring submission.

Submission

You need to submit your predicted relations on [KaggleLinks to an external site.](https://www.kaggle.com/t/6803dbf7c1c845d68355a983e7ae90ac" \t "_blank). The submission must be a CSV file.

In addition, you are also required to submit one PDF report (please name it as hw1\_report.pdf) as well as your training and inference code (packaged in a zip file named hw1.zip) on Canvas. The code package should include everything that would be required to train and use your models at run-time. Please ensure you set up your code accordingly for us to be able to replicate your training and submissions — this involves setting a random seed. You should have comments in the code to help us understand your code as well.

Your report should describe your models, approach, and hyperparameters.

Evaluation

Submissions are scored based on accuracy score. If an example utterance belongs to multiple classes, you need to predict all class labels correctly to be counted as correct, and if an example utterance belongs to no class at all, you need to predict an empty string.

Homework Report

The homework report must be a detailed summary of the approach you followed to complete your work. We highly recommend that you use a [LaTeX templateLinks to an external site.](https://www.overleaf.com/latex/templates) for your report since for your proposal and final project, you will need to prepare those using the [ACL ProceedingsLinks to an external site.](https://www.overleaf.com/latex/templates/acl-2020-proceedings-template/zsrkcwjptpcd) format. You will be required to provide the following high level sections as part of your report with additional subsections as described:

**Introduction**

* Provide a formal statement of the problem you are trying to solve — whether it is a supervised or unsupervised problem, what specific task it is.
* Describe the dataset that was provided to you — background information, descriptive statistics of the dataset, what the input and output of the dataset are. Provide examples from the dataset inline or in tables.

**Models**

* Give a description of what embedding methods have used for the models that you are training and why you pick them .
* Include a subsection for each model that you are training. Give a brief summary of how the models are implemented, trained, and the tuneable hyperparameters of the model. Additionally, provide citations to the original work that implemented these methods.

**Experiments**

* Provide a description of the data-set split, the method for selecting hyper- parameters of your final models, any approaches you used for handling data sparsity/imbalance.
* Include a subsection for each model to describe the different values for the hyper-parameters tested, any special configurations for your model such as solvers or algorithms used.
* Describe the methods you used to evaluate how good your models were and what criterion you used to select the models for generating your test set submissions.

**Results**

* Describe how well each model performed on your train, validation and test sets. Describe how the performance varied with different choice of hyperparameter values. Include the requisite tables, plots and figures to illustrate your point.
* Identify the best performing approach(es) and validate why they performed well. Try to bolster your conclusions by finding and citing work which arrive at similar results.

**References**

* Provide a bibliography for the literature that you cited. You can make use of bibtex or natbib packages to automatically generate the bibliography section.

**Appendix**

* Include an appendix for more detailed table, plots and figures, if needed.