XCS224U: Assignments Description

Homework and Bake-off 1 Notebook

Word similarity and relatedness datasets have long been used to evaluate distributed representations. This notebook provides code for conducting such analyses with a new word relatedness datasets. It consists of word pairs, each with an associated human-annotated relatedness score.

The evaluation metric for each dataset is the Spearman correlation coefficient between the annotated scores and your distances, as is standard in the literature.

This homework asks you to write code that uses the count matrices in data/vsmdata to create and evaluate some baseline models. The final question asks you to create your own original system for this task, using any data you wish. This accounts for 9 of the 10 points for this assignment.

For the associated bake-off, we will distribute a new dataset, and you will evaluate your original system (no additional training or tuning allowed!) on that datasets and submit your predictions. Systems that enter will receive the additional homework point, and systems that achieve the top score will receive an additional 0.5 points.

Assignment and Bake-off 2 Notebook

This homework and associated bakeoff are devoted to supervised sentiment analysis using the ternary (positive/negative/neutral) version of the Stanford Sentiment Treebank (SST-3) as well as a new dev/test dataset drawn from restaurant reviews. Our goal in introducing the new dataset is to push you to create a system that performs well in both the movie and restaurant domains.

The homework questions ask you to implement some baseline system, and the bakeoff challenge is to define a system that does well at both the SST-3 test set and the new restaurant test set. Both are ternary tasks, and our central bakeoff score is the mean of the macro-FI scores for the two datasets. This assigns equal weight to all classes and datasets regardless of size.

The SST-3 test set will be used for the bakeoff evaluation. This dataset is already publicly distributed, so we are counting on people not to cheat by developing their models on the test set. You must do all your development without using the test set at all, and then evaluate exactly once on the test set and turn in the results, with no further system tuning or additional runs. Much of the scientific integrity of our field depends on people adhering to this honor code.

XCS224U: Assignments Description

Assignment and Bake-off 3 Notebook

The goal of this homework is to explore few-shot (or, prompt-based) learning in the context of open-domain question answering. This is an exciting area that brings together a number of recent task ideas and modeling innovations.

Our core task is open-domain question answering (OpenQA). In this task, all that is given by the dataset is a question text, and the task is to answer that question. By contrast, in modern QA tasks, the dataset provides a text and a gold passage, with a guarantee that the answer will be a substring of the passage.

OpenQA is substantially harder than standard QA. The usual strategy is to use a retriever to find passages in a large collection of texts and train a reader to find answers in those passages. This means we have no guarantee that the retrieved passage will contain the answer we need. If we don't retrieve a passage containing the answer, our reader has no hope of succeeding. Although this is challenging, it is much more realistic and widely applicable than standard QA. After all, with the right retriever, an OpenQA system could be deployed over the entire Web.

The task posed by this homework is harder even than OpenQA. We are calling this task few-shot OpenQA. The defining feature of this task is that the reader is simply a general purpose autoregressive language model. It accepts string inputs (prompts) and produces text in response. It is not trained to answer questions per se, and nothing about its structure ensures that it will respond with a substring of the prompt corresponding to anything like an answer.