MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.806/6.864 Advanced Natural Language Processing
Fall 2017

Project Instructions

1 Question Retrieval

This part is for 50 points.
In this project, you are required to implement a Question Answering (QA) system. QA is a very challenging task. Imagine you post a question on StackOverflow or Quora. We humans could easily translate our knowledge to an answer to your question in natural language. But it is very hard for computers to answer your question like us. First, it requires the system to understand what the question is about. Moreover, the system must have enough knowledge about this question. Lastly, it has to express the answer in natural language. This involves natural language generation, a very hard task in general.

For the ease of evaluation, we consider a narrower setting where the system is provided with a large corpus of question-answers pairs posted by human. To answer a question, it only needs to *retrieve* relevant answers from that corpus. This scenario is common on StackOverflow or StackExchange, where users often give a link to another thread as the answer to a question.

To be more specific, if question A is similar to question B, then the system could use the answers of B as the answers of A. Therefore, the key component of the system is a function $f(\cdot, \cdot)$ that measures the similarity between two questions. We call this task *Question Retrieval*.

1.1 Setup

In question retrieval, you will be given a set of questions $Q = \{q_1, \cdots, q_n\}$. In addition, you are given a training set of similar question pairs $\{(q, q')\}$. The goal of the model is: Given a new question q, retrieve all relevant questions from the set Q.

Your model will be trained and evaluated on Stack Exchange AskUbuntu dataset. This dataset contains about 165K unique questions, each consisting of a title and a body, and a set of user-marked similar question pairs. The dataset could be downloaded from [link]. Please read the descriptions of the data format and instructions.

1.2 Milestones

1. Read the paper Semi-supervised Question Retrieval with Gated Convolutions. Make sure you understand what are baselines (Section 5), evaluation metrics (Sec-

tion 6) and pre-training (Section 4.2).

2. Train CNN and LSTM models as described in the paper, without pre-training. Report MAP, MRR, P@1 and P@5 of your best models on the test set. For fair comparison, any of your models must have less than 450K parameters. Under this budget, try to optimize hyper-parameters like hidden state size, pooling operations, etc.

2 Transfer Learning

This part is for 50 points.

2.1 Goal

Having set benchmarks for in-domain performance, in this part of the project, we will look at a slightly modified formulation of the question similarity problem. While, the performances of models are fine when we have several tags of similar questions, we want to explore the case when we don't have such tags. Note that this is a very common scenario. Since, for several online forums, we don't have such tagged pairs during training.

One possible solution to tackling such a problem is to use a corpus which has such positive pairs in abundance and then use this data for the target domain.

Through this section of the project, we want you to understand the limitations of the model that you have developed in terms of its adaptability to other domains.

We want you to learn about adversarial domain adaptation methods and explore the performance of such techniques on the task.

You are encouraged to invent new ideas for the problem of transfer learning. This part is open-ended and invites you to explore your creativity.

2.2 Setup

We will transfer to the Android dataset [https://android.stackexchange.com/]. This is another forum from the StackExchange family.

The dataset for evaluation is present at this link [https://github.com/jiangfeng1124/Android]. The dataset consists of dev and test question pairs, separated in files for positive and negative pairs.

There is a corpus.tsv file, which has been gzipped. This contains, the title and body for each question, as in the askubuntu dataset.

2.3 Milestones

1. Compile results on the Android dataset using the following baselines

(a) Unsupervised methods used in the first part (Cosine Similarity etc)

(b) Direct transfer using models trained on the AskUbuntu dataset and then evaluated on the Android dataset, without doing any domain adaptation

2. Understand and explore adversarial domain adaptation techniques. You can refer to https://arxiv.org/pdf/1409.7495.pdf for information on how domain adaptation can be implemented. You can also refer to https://arxiv.org/pdf/1701.00188.pdf to understand how domain adaptation can be applied in a real world task.

3. Expand on domain adaptation techniques or create new ways to consider domain adaptation.

2.4 Evaluation

You will use the Area Under Curve (AUC) metric for the domain adaptation model. The area under the curve (AUC) can be interpreted as the probability that, given a randomly selected positive example and a randomly selected negative example, the positive example is assigned a higher score by the classification model than the negative example. Note that because of possibly large number of false negative pairs, we will not be using the traditional information retrieval evaluation metrics. In order to observe greater improvement in accuracy on the AUC metric, you should also look at the AUC score when the false positive ratio is less than a fraction of 0.05 AUC(0.05).