Final Project Proposal: Disentangled Representation Learning

COSI 137B
Prof. Ben Wellner
Team member: Yonglin Wang, Xiaoyu Lu

Project Overview

In this project, we aim to examine a technique for text formality extraction, namely, disentangled representation learning. Specifically, we will be applying the tools and procedures described in a paper on disentangled representation learning for style transfer (John et al., 2018). We train an autoencoder with two discriminator classifiers to produce disentangled latent spaces, one of which is the content vector space and the other style vector space.

Related work

We will be largely borrowing the ideas and framework described in (John et al., 2018). However, there are some key differences. First, we will be using a formality dataset from Grammarly, distinct from the Amazon and Yelp review datasets described in the paper. Second, we may only train a part of the model architecture described in the paper (esp. Experiment I in the paper).

However, if this is too complicated, we might turn to a simpler architecture (partial code).

Experiment Plan

Although the original paper was intended for style transfer, in our experiments, we will only focus on the procedure that will allow us to generate disentangled style and content vectors:

1. Preprocess our data for later transformation, using code similar to what is described in here, which happens to be intended for the Yahoo answers dataset we have.
2. Use the procedure and model similar to as described in this code to train classifiers on different latent spaces (as described in Experiment I by John et al. (2018)) and get disentangled style and content embeddings for each input sentence.
3. Plot the style and content embeddings as described in this code to see:
    a. If the styles of two classes are clearly separated as expected
    b. If the content embeddings are all close together and inseparable as expected

Packages and Resources

The model may need to train on GPU, to which we do have access. For a preliminary estimation, we expect to use TensorFlow, matplotlib, and pandas as the primary tools. We will try to avoid using packages requiring *sudo* priviledges if possible.

We will also need to a close reading of the model design in John et al., 2018 to determine which part of their proposed architecture is related to our experiment plan.

If we inevitably need to implement a model from scratch, it may take us time to understand how the definition of a loss function translates to code implementation.