Generating Online Dating Profiles

Team Information

Name: Pushpak Raj Gautam

PID: A13708485

Problem Definition

The goal is to write a dating profile for a person based on their attributes (which may or may not include age, sex, location, height, income, language, et cetera). A tool as such can help dating apps provide a default description for new users when they join. The generated profile might work as someone's final version or as a starting point to improve upon.

A trivial solution to this problem would be to provide a basic template for each user, customized on some parameters. However, this won't look personalized and won't be interesting enough to garner interest. Going through a huge list of profiles, a model should be able to pick up things that are important for a profile but at the same time keep it random enough to give a personalized feel.

Related Paper Summary

Paper

Name: *Generative Concatenative Nets Jointly Learn to Write and Classify Reviews*

Authors: *Zachary C. Lipton, Sharad Vikram, Julian McAuley*

Publication Venue: https://arxiv.org/abs/1511.03683 (Could not find the venue but it's published)

Contributions

The paper is the first attempt at generating personalized beer reviews based on some beer attributes. It is also the first work to use character-level RNNs instead of word-level ones to produce relevant text. Based on their evaluation, their generated reviews took care of misspellings, slang, large vocabulary and negation.
Evaluation uses perplexity as a measure. Both the average test set perplexity and median test set perplexity are reported and compared against an unsupervised LSTM language model. Their model always outperforms this baseline.
Here's the dataset they used. It has been accumulated from *BeerAdvocate.com* and contains over 1.5 million reviews - https://data.world/socialmediadata/beeradvocate

Critical Analysis

The authors mention that classification becomes slow if the number of classes increase. For example, to identify authors of a review, the model must run through the network 1000 times for each of the roughly 1000 authors. This can be made more efficient. Apart from this, the assumption, the paper makes, is that everyone has a certain style of writing. This seems fair. A simpler solution would be to use something like an n-gram model and somehow personalize it.

Note

The dataset I'll use is different and is located here - https://github.com/rudeboybert/JSE_OkCupid. It has over 65000 profiles. There's also a paper associated with this dataset, but it mostly provides statistics. Since the dataset size is smaller than the one used in the above-mentioned case, I would start with word-level classification with the RNN to remove the requirement of learning spellings.