Genre Classification in Rust

Marko Ložajić, Anna Soboleva

Description

This crate contains a single-label genre classifier that was trained to distinguish between five common literary genres, using TF-IDF for feature extraction and the Naive Bayes method for classification.

1. Corpus

Our corpus contains 472 texts spanning five genres: Detective, Erotica, Horror, Romance and Science Fiction. The train/test split follows an approximately 80/20 percent distribution. Texts were collected from Project Gutenberg and Archive of our Own in .html and .txt formats.

2. Preprocessing

The texts are read and tokenized with the option to remove stopwords (as "stopwords" a set of nltk stopwords is used), and the words are stored together with their counts in a matrix.

3. Vector representations

The documents are represented as TF-IDF vectors, where each element (column) corresponds to one word in the vocabulary, resulting in a large, sparse term-document matrix. We attempted to mitigate by this using singular value decomposition (SVD) to reduce the matrix size, where we tried three different implementations (la, nalgebra, ndarray) that unfortunately either proved too computationally demanding or yielded subpar results, so we ended up sticking to the TF-IDF representations for the classification.

4. Classifier

For the classification task we use rusty-machine's Naive Bayes implementation, based on whose predictions we calculate macro-averaged precision, recall and F1-score. The pretrained model yields a modest F1-score of around 41 percent, some improvements for which are suggested in section 6. "Possible improvements"

5. User experience

The program has two main modes of operation - the "regular" mode, where data is loaded from data and labels of the user's choice (defaults provided), and the "autopilot" mode, loading pretrained TF-IDF vectors (trained with the default data and stopwords removed).

6. Possible improvements

- Divide longer texts to ensure all data is of similar length (data augmentation);
- Increase corpus size and number of possible genres;
- Include class probabilities to help detect borderline cases;
- Try out different classification methods (e.g. Logistic Regression. Neural Networks);
- Currently works only on Mac and Linux - make Windows-friendly as well;