Author Prediction for Poetry

Katrin Schmidt
Immatriculation-no
@ims.uni-stuttgart.de

Carlotta Quensel
3546286
@ims.uni-stuttgart.de

Abstract

Same structure as the whole paper, but in short

1 Introduction

Short motivation and explanation of relevancy of your task, research questions/hypothesis

- what is author classification
- why poetry
- research question
  - Goal: find features inherent to poetry
    - is our goal possible
    - which features are good
  - Problem: style features might depend on medium (e.g. Limmerick) more than on the author
- Motivation: ??

2 Method

Description of your method (e.g. perceptron) without talking about the specific task too much. Explain features used (with or without being task specific), but do not judge them.

- Maximum Entropy classifier (explain why not other approaches)
  - short program description (training, classification)
- Features: MaxEnt/Bag-of-Words(/poetry specific?), learnFeatures (PMI)
- subsection with data/corpus creation from Poetry Foundation
  - which information is included
  - preprocessing steps (tokenizing)
  - statistics (number of poems/poets/poems per poet with graph)

3 Experiments

3.1 Experimental Design

For all model configurations, our hyperparameters of the number of authors, the maximum training iterations, the accuracy threshold and the number of features per author were left untouched. The training stopped when the accuracy improvement fell below 0.001 or after 100 iterations (which was never reached during training).

With at least 30 poems per author and 1569 datapoints, the split between test and training data was pseudo-randomized 75 to 25 to ensure sufficient coverage of each author in training and evaluation. This way the least prolific author (Edmund Spenser) had 25 poems for training and eight for the evaluation. The poems alloted for training were also used to compute the pointwise mutual information because of the aforementioned (2) data sparsity. This was done despite the risk of overfitting since a second division of the data would leave us with ten to fifteen poems per author for both the feature extraction and training, which is not sufficient for either method (PMI or Maximum Entropy training).

The features themselves consisted of individual tokens, the number of verses and stanzas as well as the rhyme scheme for each poem. The word features were obtained by converting the poem into a bag-of-words vector and retrieving the value (0 or 1) for a specific word in the vocabulary. This vocabulary was built from the tokenized training data as the classifier will only learn weights for features that are seen during training. A simple classifier with only the tokens was trained as a baseline for the poetry specific features.

The rhyme scheme was obtained from the first four lines of the poems. While there might be rhymes that span more than four lines (***abcda***), this is highly unlikely without a repetition of the first rhyme or another rhyme pair in the lines in between 1 and 5. The scheme was constructed by consecu-

	Baseline	Full	Verse	Stanza	Rhyme
Accuracy					
Precision					
Recall					
micro F_1					
macro F_1					

Table 1: Evaluation of the different model configurations on the unseen test data and the model's accuracy on the training data for comparison.

tively taking the last word of the first unmatched line and checking all other unmatched lines for rhymes with `pronouncing.rhymes(word)` (Parrish, 2015). This lead to a four letter string for each poem of the form:

$$\text{“}a\{a,b\}\{a,b,c\},\{a,b,c,d\}\text{”}$$

The number of verses per poem were sorted in x-line steps after looking at the distribution in the training data. The steps were converted into bins of at least and at most x number of verses. Similarly, the number of blank lines in a poem was used to determine the number of stanzas and sorted into steps of x, y or z stanzas.

For our first experiments, the classifier was initialized with the 30 most informative features per author, which for 30 authors resulted in 900 features whose weights were trained. We compared the baseline of just words to a classifier with all features ("full"), combinations of words and only one of the advanced features (from here on referred to by the name of that feature, i.e. stanza model, rhyme model, verse model) and a model with all features ecxept for the words. After comparing these models with fixed hyperparameters, we changed the number of learned features and the size of the author set to observe the parameters' effect.

3.2 Results

For the training, the models rarely performed more than ten optimization steps before the accuracy stopped changing. Tracking the aggregated loss as well as the accuracy showed us that the loss was still high when the accuracy stopped improving. here graph? Training of the baseline model terminated with an accuracy 50% on the training data, which dropped to a micro f_1 score of .115 with the unseen test data. This pattern was also present in the other model configurations as shown in table 1.

3.3 Error Analysis

Given the configurations in the Results section, what are frequent sources of errors

- specifics and numbers about errors?
- overprediction of alphabetically first author
- many authors not predicted (uneven data distribution or bad features)
- feature weights converge similarly (no real weighting)

4 Summary & Conclusion

Explain and summarize your results on a more abstract level. What is good, what is not so good. What are the main contributions in your experiments?

5 Future Work

What did you have in mind what else your would have liked to experiment with? Other ideas?

- other models (e.g. Neural Net)
- other features (Topics from Poetry Foundation website)
- feature interdependencies/more data analysis
- genre interaction with author classification (multitask learning?)

A Contributions

Who implemented what? Who participated in the design of which components? Who wrote which part of the review?

B Declaration of Originality

we hereby certify that this report has been composed by us and is based on our own work, unless stated otherwise. No other person's work has been used without due acknowledgement our own contributions are listed under A. All references and verbatim extracts have been quoted, and all sources of information, including graphs and data sets, have been specifically acknowledged.