CS446 Class Project: Determining Language of Origin in Hand-Translated Texts from the 19th Century

Sean Wilner (swilner2) and Ben Seefeldt (seefldt2)

October 21, 2013

Abstract

We will use one-vs.-all multiclass classification to determine the original language of texts translated into English. We will use a dataset gathered from the public domain ebook source, Project Gutenberg.[1] By calcuating a variety of features over our dataset using the Standford CoreNLP toolset[2] we hope to isolate features that will allow us to classify texts based simply on their English translation.

Task description

We would like to classify previously translated texts based on their original language. In addition to classifying based on language, we will classify based on larger language families. These classification tasks will be completed over a variety of features pulled from the source text focusing on length of sentences, frequency of grammatical constructs, and ratios between different pieces of speech.

This cuts the project into two discrete tasks. First is the collection of data and the distillation of features from the data. Second is the multi-class classification over these given features.

Background

Natural Language Processing is an extremely large field, with many different applications. One particular subfield of NLP is machine translation. In a way, the work we are hoping to do is a return to traditional forms of human translation.

While there has been significant work in the field of authorship attribution, [1][2][3] we are unaware of any projects which are focused on the problem of original language. The existing work can provide a good grounding for the types of features available, and what types of approaches give the best classification.

Data and evaluation

The data we will use for our project will be taken from Project Gutenberg, a collection of public domain texts. We hope to draw entirely from fiction works of prose written in the 19th century. By sampling from a relatively narrow slice of literature, we hope to remove any potential classification based on modern language or significantly differing literary styles. We hope to use each novel-length piece of text for multiple samples, using chapters as our individual sample.

We will evaluate our system using the standard five-fold validation along with the t-test significance testing covered in class. The lowest possibly bar for the desired accuracy would be achieving any increase

[1] http://www.gutenberg.org/

[2] http://nlp.stanford.edu/software/corenlp.shtml

over what would be expected by randomly guessing. In terms of usable results, we believe a meaningful result would be found in the ability to observe which grammatical constructs have a stronger impact on classification, so as to make general conclusions about the typical literary style of translations done from this language.

Approach

We will tackle the task in two major steps. First we will acquire a corpus of data to work from in the form of ebooks of chosen literary works with known translations to English which fall into our time period. These works will be further segmented into approximately chapter length sections which will be used in the initial data processing.

Informed by the selected articles found in the bibliography we will choose a series of features which should capture the overall pacing, rhythm, and feel of a text. Using the Standford CoreNLP toolkit, we will generate a final data set which contains each chapter section represented as a series of features, as well as the given label. Note that this label is not singular, we will attempt classification on both specific language and wider language family.

While we plan on using the one-v.-all method for multiclass classification, the specifics of our implementation as well as other possible methods which may increase classification accuracy will be chosen as time permits, and as indicated by limited trial runs.

Schedule

10-25 Book selection complete

11-02 Feature selection and creation complete

11-09 Detailed algorithm/test plan complete

11-16 Implementation functional

11-23 Test runs complete

11-30 Scheduled make-up week

12-07 Write-up complete

References

[1] Joachim Diederich, Jörg Kindermann, Edda Leopold, and Gerhard Paass. Authorship attribution with support vector machines. *Applied intelligence*, 19(1-2):109–123, 2003.

[2] Patrick Juola. Authorship attribution. *Foundations and Trends in information Retrieval*, 1(3):233–334, 2006.

[3] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology*, 60(1):9–26, 2009.