CS 388: Natural Language Processing Spring 2016

Project Report — Original Language Detection

Students: M. Denend, J. Hoffmann, L.Prakash *Prof. Ray Mooney*

1 Project description

(a) Description

The goal of this project is to be able to find the original language a text was written in, from the translated English version. We believe that even with completely bilingual translators, languages have a specific structure that will influence the translator in the way he creates the translation. We hope to recover this structure during our task.

(b) Methodology

From Project Gutenberg, we built a dataset of books written in different languages, and translated into English. We then cut these books into slices of 400 sentences, and compute features on these slices. After feature extraction, we feed these features into a classifier and test it on a completely different set of books.

2 Building the dataset

(a) Pitfalls

Since we built it ourselves, our dataset is rather small. We therefore were extremely careful not to overfit on books particularities that are not specific to the language, such as topic, style, author, geographic places, time period. We believe the way we've constructed the dataset satisfies these criteria.

(b) Description of the dataset

When building the dataset, we used books written from 1700 to 1920 to control the time period as best as we can. We made sure that we never used the same author nor translator (?) in train and test, and we tried varying the topics between train and test as much as we could once the previous criteria were met. When using Lexical Features, we only remembered the most common 1000 English words to control overfitting on topics as well. The dataset we used is further described on this table:

Language	Title	Author	Pub.Date
American	The Adventures of Huckleberry Finn	Mark Twain	1884
American	The Invisible Man	Orson Wells	1897
American	The Scarlet Letter	Nathaniel Hawthorne	1850
American	Moby Dick	Herman Melville	1851
American	Uncle Toms Cabin	Harriet Beecher Stowe	1852
American	The Narrative of A. G. Pym of Nantucket	Edgar Allen Poe	1899
English	Great Expectations	Charles Dickens	1861
English	The Picture of Dorian Gray	Oscar Wilde	1891
English	Jude the Obscure	Thomas Hardy	1895
English	Treasure Island	R.L Stevenson	1883
English	Middlemarch	George Eliot (M. Evans)	1874
English	Frankenstein	Mary Wollstonecraft Shelley	1818
French	Around the world in 80 days	Jules Verne	1873
French	Father Goriot	Balzac	1835
French	Les Miserables	Victor Hugo	1862
French	Madame Bovary	Gustave Flaubert	1857
French	The Man in the Iron Mask	Alexandre Dumas	1637
French	Candide	Voltaire	1759
French	The Red and the Black	Stendahl	1830
Russian	Fathers and Children	Ivan Turgenev	1862
Russian	The Man who was afraid	Maxim Gorky	1899
Russian	War and Peace	Leo Tolstoy	1869
Russian	The Brothers Karamazov	Fyodor Dostoyevsky	1880
Russian	A hero of Our Time	M. Y. Lermontov	1840
Russian	The Death of the Gods	Dmitri Mérejkowski	1895
Spanish	Don Quixote	Miguel de Cervantes	1605
Spanish	Dona Perfecta	Benito Pérez Galdós	1876
Spanish	The Visions of Quevedo	Francisco de Quevedo	1626
Spanish	Tragic Sense of Life	Miguel de Unamuno	1912
Spanish	The Life of Lazarillo of Tormes	Lazarillo of Tormes	1554
Spanish	The Four Horsemen of the Apocalypse	Vicente Blasco Ibanez	1916
Spanish	Pepita Ximenez	Juan Valera	1874
Spanish	The Fourth Estate	Armando Palacio Valdés	1901
German	The Devil's Elixir	E. T. A. Hoffmann	1815
German	Debit and Credit	Gustave Freytag	1855
German	Old Fritz and the New Era	Luise Mühlbach	1868
German	Venus in Furs	Leopold V. Sacher-Masoch	1870
German	The Sorrows of Young Werther	Johann Wolfgang von Goethe	1774
German	The Banished: A Swabian Historical Tale	Wilhelm Hauff	1839

3 Features

(a) General Description

When computing the features, we cut the books in slices of n sentences. With $n = 1$, we get a lot of training examples, but finding out the original language from only 1 sentence seems unreasonable (which was proven by experiment). With the whole book as a sample, we didn't get enough training instances. We chose $n = 400$, as it provided good balance between number of train samples, and informativeness of the samples.

In training, we take 8 slices of each book, so that we have the same size of training for each language. In testing, we take all the slices, except the incomplete last one.

(b) Lexical Feature

Three different kind of lexical features were used:

1000 most common English words After overfitting on words like "Moscow" or "Madame" when using unigrams, we decided to only use the 1000 most common words in English to avoid overfitting on topics, local idioms, or geographic places.

POS unigram We only used POS unigrams, since we believed using bigram or trigram would just add redundancy with the Parse features.

Etymology For each word used, we computed the etymology of the word from parsing Wiktionary [2], and added all possible etymology as a feature. Rational being that if you're translating from a latin language, you may be more likely to use a latin word when choosing between synonyms. We only included the etymologies of nouns, verb, adjective and adverb, since we believed preposition, pronouns and others wouldn't be significant and would only drown the relevant data.

(c) Parse Features

Inspired by [3], we ran the Stanford parser on our slices, and computed the parse trees. By running a BFS, we then compute the generation rules and count them. We don't take into account the leaves of the trees to avoid redundancy with lexical features (and possible topic overfitting).

(d) Homemade Features

We also computed all the features of [1], which include length of sentences, length of words, ratio of pronouns, nouns, complex sentences etc. [MATT ADD WHATEVER YOU WANT].

4 Classification task

(a) Features Extraction

Following [3], we were ready to use Information gain for feature extraction. However, it turned out that we didn't have that many features and could run all the algorithms on our dataset. Using Information Gain or Anova didn't help the results, so we finally removed this step.

(b) Comparisons of different algorithm

We compared the results of a Logistic Regression, a Random Forest, a SVM with gaussian kernel and XGBoost with trees. Surprisingly enough, even if XGBoost is known to perform extremely well for classification tasks, the Logistic Regression always outperformed all the other classifiers, whatever number of features we picked.

5 Experiments

(a) Description of different experiments

We first decided to reproduce the experimental conditions of [1], to show that they're non representative of the task.

We then experimented with the different kind of features we computed, to pick the best possible mix.

(b) Results

Experiment	Accuracy	Best classifier
1000 Most common words	96.67 %	LR

Figure 1: Testing and training on the same books (different parts)

As expected, using the same books for training and testing gives excellent results even when using only the 1000 most common English words. Since we used a lot of books in common with [1], we're wondering how they got such poor results.

Experiment	Accuracy
Unigrams	61.80 %
1000 Most common words	62.92 %
1000 Most common words + etymology	61.80 %
1000 Most common words + POS	46.06 %
Parse features only	50.56 %
Homemade only	33.70 %
1000 most common + Parse	
1000 most common + Homemade	
Parse + Homemade	51.69 %
1000 most common + Parse + Homemade	64.04 %
1000 most common + Parse + Homemade + etymology	65.17
1000 most common + Parse + etymology	65.17

Figure 2: Lexical Features

We get better results when using only the 1000 most common English words, as opposed to all the words (Unigrams). This was a very positive result, since it suggests that we did built our dataset in a robust way. Indeed, using unigrams completely overfitted on topic, as illustrated by the following example: we used Moby Dick in our training, and therefore *whale* is the third most relevant feature for American books. Contrary to intuition, POS unigram dramatically reduced the accuracy. It's interesting to notice that etymology features decrease the accuracy when considering only lexical features, but increased it when considering everything.

When using Homemade only, we classify everything as Spanish or Russian, which is even worse than what the already poor results led to believe. As following results suggest, Homemade features are not informative, and they never appear in the top 100 most relevant features for a language.

6 Conclusion

We have successfully built a scientifically robust experiment for Original Language Detection, which had never been done before. We built a consistent dataset, computed Lexical, Parse and Homemade features, chose the most suitable classification algorithm, and reached 64% accuracy on our testing set. However, our testing set is made of slices of 400 sentences. For a given book, if we compute those slices and take the majority vote for the whole book (the book is assigned the class that got the most slices), we reach a 100 % accuracy on our testing set!!! Obviously, the dataset is not big enough for this experiment to be conclusive, and we wish we had included more languages. Nevertheless, the results are extremely promising!

References

[1] Gerard Lynch and Carl Vogel. Towards the automatic detection of the source language of a literary translation. 2012.

[2] Christian M. Meyer and Iryna Gurevych. Wiktionary: A new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography. In Sylviane Granger and Magali Paquot, editors, *Electronic Lexicography*, chapter 13, pages 259–291. Oxford: Oxford University Press, November 2012.

[3] Sze-Meng Jojo Wong and Mark Dras. Exploiting parse structures for native language identification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1600–1610. Association for Computational Linguistics, 2011.