Project on
AUTOMATIC LANGUAGE DETECTION PROGRAM

Submitted by

SHUBHAM BALASAHEB PATIL

Internship at

ECKOVATION CAREERS

Course Name

PYTHON PROGRAMMING CERTIFICATION

Abstract

Language identification is the task of automatically detecting the language(s) present in a document based on the content of the document. In this work, we address the problem of detecting documents that contain text from more than one language (multilingual documents). We introduce a method that is able to detect that a document is multilingual, identify the languages present, and estimate their relative proportions. We demonstrate the effectiveness of our method over synthetic data, as well as real-world multilingual documents collected from the web.

Index

Introduction

Language identification is the task of automatically detecting the language(s) present in a document based on the content of the document. Language identification techniques commonly assume that every document is written in one of a closed set of known languages for which there is training data, and is thus formulated as the task of selecting the most likely language from the set of training languages. In this work, we remove this monolingual assumption, and address the problem of language identification in documents that may contain text from more than one language from the candidate set. We propose a method that concurrently detects that a document is multilingual, and estimates the proportion of the document that is written in each language. Detecting multilingual documents has a variety of applications. Most natural language processing techniques presuppose monolingual input data, so inclusion of data in foreign languages introduces noise, and can degrade the performance of NLP systems (Alex et al., 2007; Cook and Lui, 2012). Automatic detection of multilingual documents can be used as a pre-filtering step to improve the quality of input data. Detecting multilingual documents is also important for acquiring linguistic data from the web (Scannell, 2007; Abney and Bird, 2010), and has applications in mining bilingual texts for statistical machine translation from online resources (Resnik, 1999; Nie et al., 1999; Ling et al., 2013). There has been particular interest in extracting text resources for low-density languages from multilingual web pages containing both the low-density language and another language such as English (Yamaguchi and Tanaka-Ishii, 2012; King and Abney, 2013). King and Abney (2013, p1118) specifically mention the need for an automatic method "to examine a multilingual document, and with high accuracy, list the languages that are present in the document". We introduce a method that is able to detect multilingual documents, and simultaneously identify each language present as well as estimate the proportion of the document written in that language. We achieve this with a probabilistic mixture model, using a document representation developed for monolingual language identification (Lui and Baldwin, 2011). The model posits that each document is generated as samples from an unknown mixture of languages from the training set. We introduce a Gibbs sampler to map samples to languages for any given set of languages, and use this to select the set of languages that maximizes the posterior probability of the document.

Our method is able to learn a language identifier for multilingual documents from monolingual training data. This is an important property as there are no standard corpora of multilingual documents available, whereas corpora of monolingual documents are readily available for a reasonably large number of languages (Lui and Baldwin, 2011). We demonstrate the effectiveness of our method empirically, firstly by evaluating it on synthetic datasets drawn from Wikipedia data, and then by applying it to real-world data, showing that we are able to identify multilingual documents in targeted web crawls of minority languages (King and Abney, 2013).

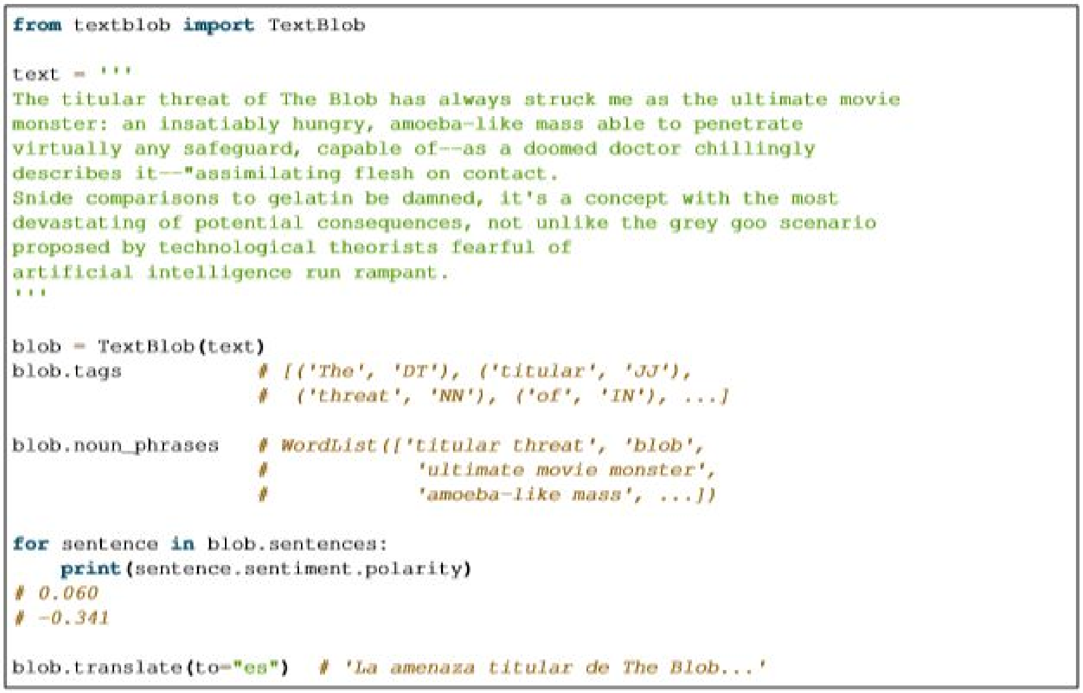
Our main contributions are: (1) we present a method for identifying multilingual documents, the languages contained therein and the relative proportion of the document in each language; (2) we show that our method outperforms state-of-the-art methods for language identification in multilingual documents; (3) we show that our method is able to estimate the proportion of the document in each language to a high degree of accuracy; and (4) we show that our method is able to identify multilingual documents in real-world data.

Background

Most language identification research focuses on language identification for monolingual documents (Hughes et al., 2006). In monolingual LangID, the task is to assign each document D a unique language Li ∈ L. Some work has reported near-perfect accuracy for language identification of large documents in a small number of languages (Cavnar and Trenkle, 1994; McNamee, 2005). However, in order to attain such accuracy, a large number of simplifying assumptions have to be made (Hughes et al., 2006; Baldwin and Lui, 2010a). In this work, we tackle the assumption that each document is monolingual, i.e. it contains text from a single language. In language identification, documents are modeled as a stream of characters (Cavnar and Trenkle, 1994; Kikui, 1996), often approximated by the corresponding stream of bytes (Kruengkrai et al., 2005; Baldwin and Lui, 2010a) for robustness over variable character encodings. In this work, we follow Baldwin and Lui (2010a) in training a single model for languages that naturally use multiple encodings (e.g. UTF8, Big5 and GB encodings for Chinese), as issues of encoding are not the focus of this research. The document representation used for language identification generally involves estimating the relative distributions of particular byte sequences, selected such that their distributions differ between languages. In some cases the relevant sequences may be externally specified, such as function words and common suffixes (Giguet, 1995) or grammatical word classes (Dueire Lins and Gonc¸alves, 2004), though they are more frequently learned from labeled data (Cavnar and Trenkle, 1994; Grefenstette, 1995; Prager, 1999a; Lui and Baldwin, 2011). Learning algorithms applied to language identification fall into two general categories: Bayesian classifiers and nearest-prototype (Rocchio-style) classifiers. Bayesian approaches include Markov processes (Dunning, 1994), naive Bayes methods (Grefenstette, 1995; Lui and Baldwin, 2011; Tiedemann and Ljubesiˇ c, 2012), and compressive mod- ´ els (Teahan, 2000). The nearest-prototype methods vary primarily in the distance measure used, including measures based on rank order statistics (Cavnar and Trenkle, 1994), information theory (Baldwin and Lui, 2010a), string kernels (Kruengkrai et al., 2005) and vector space models (Prager, 1999a; McNamee, 2005). Language identification has been applied in domains such as USENET messages (Cavnar and Trenkle, 1994), web pages (Kikui, 1996; Martins and Silva, 2005; Liu and Liang, 2008), web search queries (Ceylan and Kim, 2009; Bosca and Dini, 2010), mining the web for bilingual text (Resnik, 1999; Nie et al., 1999), building minority language corpora (Ghani et al., 2004; Scannell, 2007; Bergsma et al., 2012) as well as a large scale database of Interlinear Glossed Text (Xia et al., 2010), and the construction of a large-scale multilingual web crawl (Callan and Hoy, 2009).

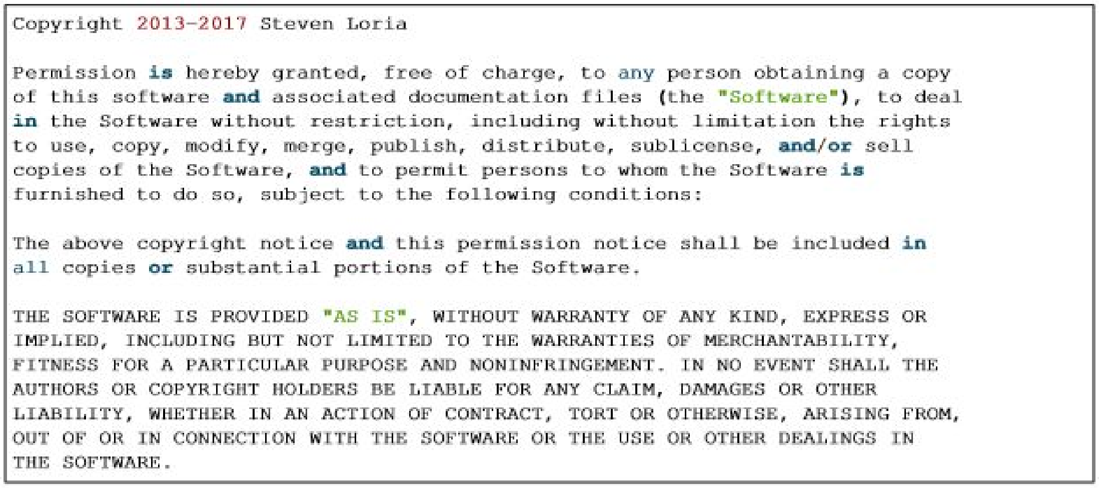
Textblob

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.



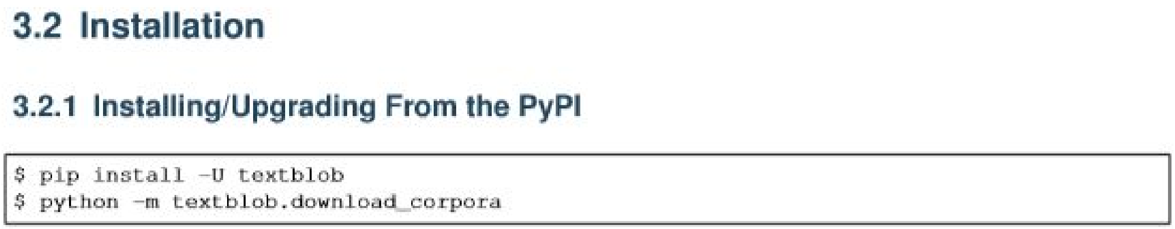
```
from textblob import TextBlob

text = '''
The titular threat of The Blob has always struck me as the ultimate movie
monster: an insatiably hungry, amoeba-like mass able to penetrate
virtually any safeguard, capable of--as a doomed doctor chillingly
describes it--"assimilating flesh on contact.
Snide comparisons to gelatin be damned, it's a concept with the most
devastating of potential consequences, not unlike the grey goo scenario
proposed by technological theorists fearful of
artificial intelligence run rampant.
'''

blob = TextBlob(text)
blob.tags           # [('The', 'DT'), ('titular', 'JJ'),
                    #  ('threat', 'NN'), ('of', 'IN'), ...]

blob.noun_phrases   # WordList(['titular threat', 'blob',
                    #            'ultimate movie monster',
                    #            'amoeba-like mass', ...])

for sentence in blob.sentences:
    print(sentence.sentiment.polarity)
# 0.060
# -0.341

blob.translate(to="es")  # 'La amenaza titular de The Blob...'
```

3.1 License



```
Copyright 2013-2017 Steven Loria

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```

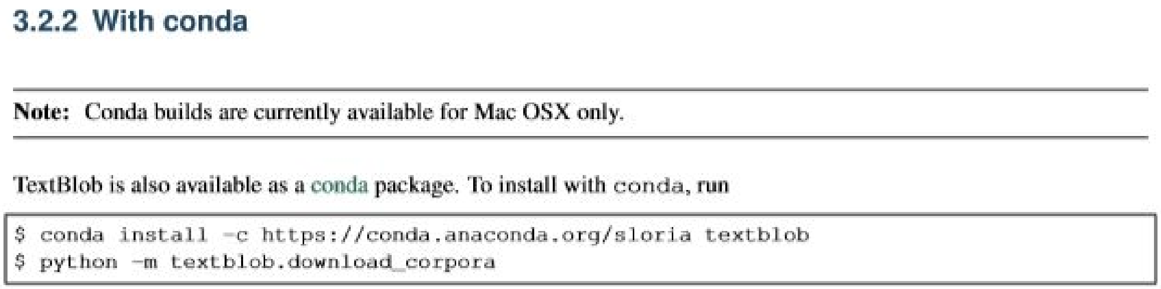
Installation



3.2 Installation

3.2.1 Installing/Upgrading From the PyPI

```
$ pip install -U textblob
$ python -m textblob.download_corpora
```



3.2.2 With conda

Note: Conda builds are currently available for Mac OSX only.

TextBlob is also available as a conda package. To install with `conda`, run

```
$ conda install -c https://conda.anaconda.org/sloria textblob
$ python -m textblob.download_corpora
```

Features

- Noun phrase extraction
- Part-of-speech tagging
- Sentiment analysis
- Classification (Naive Bayes, Decision Tree)
- Language translation and detection powered by Google Translate
- Tokenization (splitting text into words and sentences)
- Word and phrase frequencies
- Parsing
- n-grams
- Word inflection (pluralization and singularization) and lemmatization
- Spelling correction
- Add new models or languages through extensions
- WordNet integration

Methodology

Language identification for multilingual documents is a multi-label classification task, in which a document can be mapped onto any number of labels from a closed set. In the remainder of this paper, we denote the set of all languages by L. We denote a document D which contains languages Lx and Ly as D → {Lx, Ly}, where Lx, Ly ∈ L. We denote a document that does not contain a language Lx by D → {Lx}, though we generally omit all the languages not contained in the document for brevity. We denote classifier output using .; e.g. D . {La, Lb} indicates that document D has been predicted to contain text in languages La and Lb.

Feature Selection

We represent each document D as a frequency distribution over byte n-gram sequences such as those in Table 1. Each document is converted into a vector where each entry counts the number of times a particular byte n-gram is present in the document. This is analogous to a bag-of-words model, where the vocabulary of "words" is a set of byte sequences that has been selected to distinguish between languages. The exact set of features is selected from the training data using Information Gain (IG), an information-theoretic metric developed as a splitting criterion for decision trees (Quinlan, 1993). IGbased feature selection combined with a naive Bayes classifier has been shown to be particularly effective for language identification (Lui and Baldwin, 2011).

Multi-label text classification, topic modeling and our model for language identification in multilingual documents share the same fundamental representation of the latent structure of a document. Each label is modeled with a probability distribution over tokens, and each document is modeled as a probabilistic mixture of labels. As presented in Griffiths and Steyvers (2004), the probability of the i th token (wi) given a set of T labels.

The set of tokens w is the document itself, which in all cases is observed. In the case of topic modeling, the tokens are words and the labels are topics, and z is latent. Whereas topic modeling is generally unsupervised, multi-label text classification is a supervised text modeling task, where the labels are a set of predefined categories (such as RUBBER, IRON-STEEL, TRADE, etc. in the popular Reuters21578 data set (Lewis, 1997)), and the tokens are individual words in documents. z is still latent, but constrained in the training data (i.e. documents are labeled but the individual words are not). Some approaches to labeling unseen documents require that z for the training data be inferred, and methods for doing this include an application of the ExpectationMaximization (EM) algorithm (McCallum, 1999) and Labeled LDA (Ramage et al., 2009). The model that we propose for language identification in multilingual documents is similar to multilabel text classification. In the framework of Equation 1, each per-token label zi is a language and the vocabulary of tokens is not given by words but rather by specific byte sequences (Section 3.1). The key difference with multi-label text classification is that we use monolingual (i.e. mono-label) training data. Hence, z is effectively observed for the training data (since all tokens must share the same label). To infer z for unlabeled documents, we utilize a Gibbs sampler, closely related to that proposed by Griffiths and Steyvers (2004) for LDA.

Language Identification

The model described can be used to compute the most likely distribution to have generated an unlabeled document over a given set of languages for which we have monolingual training data, by letting the set of terms w be the byte n-gram sequences we selected using per-language information gain (Section 3.1), and allowing the labels z to range over the set of all languages L. Using training data, we compute φˆ (w) j (Equation 3), and then we infer P(Lj |D) for each Lj ∈ L for the unlabeled document, by running the Gibbs sampler until the samples for zi converge and then tabulating zi over the whole d and normalizing by |d|. Naively, we could identify the languages present in the document by D . {Lx if ∃(zi = Lx|D)}, but closely related languages tend to have similar frequency distributions over byte n-gram features, and hence it is likely that some tokens will be incorrectly mapped to a language that is similar to the “correct” language. We address this issue by finding the subset of languages λ from the training set L that maximizes P(λ|D) (a similar approach is taken in McCallum (1999)). Through an application of Bayes' theorem, P(λ|D) ∝ P(D|λ)·P(λ), noting that P(D) is a normalizing constant and can be dropped. We assume that P(λ) is constant (i.e. any subset of languages is equally likely, a reasonable assumption in the absence of other evidence), and hence maximize P(D|λ). For any given D = w1· · ·wn and λ, we infer P(D|λ) from the output of the Gibbs sampler.

In practice, exhaustive evaluation of the powerset of L is prohibitively expensive, and so we greedily approximate the optimal λ using Algorithm 1. In essence, we initially rank all the candidate languages by computing the most likely distribution over the full set of candidate languages. Then, for each of the top-N languages in turn, we consider whether to add it to λ. λ is initialized with Lu, a dummy language with a uniform distribution over terms (i.e. P(w|Lu) = 1 |w|). A language is added if it improves P(D|λ) by at least t. The threshold t is required to suppress the addition of spurious classes. Adding languages gives the model additional freedom to fit parameters, and so will generally increase P(D|λ). In the limit case, adding a completely irrelevant language will result in no tokens being mapped to the a language, and so the model will be no worse than without the language. The threshold t is thus used to control “how much” improvement is required before including the new language in λ.

Evaluation

We seek to evaluate the ability of each method: (1) to correctly identify the language(s) present in each test document; and (2) for multilingual documents, to estimate the relative proportion of the document written in each language. In the first instance, this is a classification problem, and the standard notions of precision (P), recall (R) and F-score (F) apply. Consistent with previous work in language identification, we report both the document level micro-average, as well as the language-level macro-average. For consistency with Baldwin and Lui (2010a), the macro-averaged F-score we report is the average of the per-class F-scores, rather than the harmonic mean of the macro-averaged precision and recall; as such, it is possible for the F-score to not fall between the precision and recall values. As is common practice, we compute the F-score for β = 1, giving equal importance to precision and recall.2 We tested the difference in performance for statistical significance using an approximate randomization procedure (Yeh, 2000) with 10000 iterations. Within each table of results 4, all differences between systems are statistically significant at a $p < 0.05$ level. To evaluate the predictions of the relative proportions of a document D written in each detected language Li , we compare the topic proportion predicted by our model to the gold-standard proportion, measured as a byte ratio as follows: gs(Li |D) = length of Li part of D in bytes length of D in bytes (7) We report the correlation between predicted and actual proportions in terms of Pearson's r coefficient. We also report the mean absolute error (MAE) over all document–language pairs.

Document segmentation by language could be accomplished by a combination of our method and the method of King and Abney (2013), which could be compared to the method of Yamaguchi and TanakaIshii (2012) in the context of constructing corpora for low-density languages using the web. Another area we have identified in this paper is the tuning of the parameters α and β in our model (currently α = 0 and β = 1), which may have some effect on the sparsity of the model. Further work is required in dealing with crossdomain effects, to allow for "off-the-shelf" language identification in multilingual documents. Previous work has shown that it is possible to generate a document representation that is robust to variation across domains (Lui and Baldwin, 2011), and we intend to investigate if these results are also applicable to language identification in multilingual documents. Another open question is the extension of the generative mixture models to "unknown" language identification (i.e. eliminating the closed-world assumption (Hughes et al., 2006)), which may be possible through the use of non-parametric mixture models such as Hierarchical Dirichlet Processes (Teh et al., 2006).

Conclusion

We have presented a system for language identification in multilingual documents using a generative mixture model inspired by supervised topic modeling algorithms, combined with a document representation based on previous research in language identification for monolingual documents. We showed that the system outperforms alternative approaches from the literature on synthetic data, as well as on real-world data from related research on linguistic corpus creation for low-density languages using the web as a resource. We also showed that our system is able to accurately estimate the proportion of the document written in each of the languages identified. We have made a full reference implementation of our system freely available,8 as well as the synthetic dataset prepared for this paper, in order to facilitate the adoption of this technology and further research in this area.

References

Steven Abney and Steven Bird. 2010. The human language project: building a universal corpus of the world's languages. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 88–97. Association for Computational Linguistics.

Beatrice Alex, Amit Dubey, and Frank Keller. 2007. Using foreign inclusion detection to improve parsing performance. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning 2007 (EMNLP-CoNLL 2007), pages 151–160, Prague.

Timothy Baldwin and Marco Lui. 2010a. Language identification: The long and the short of the matter. In Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010), pages 229–237, Los Angeles, USA.

Timothy Baldwin and Marco Lui. 2010b. Multilingual language identification: ALTW 2010 shared task dataset. In Proceedings of the Australasian Language Technology Workshop 2010 (ALTW 2010), pages 5–7, Melbourne, Australia.

Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson. 2012. Language identification for creating language-specific Twitter collections. In Proceedings the Second Workshop on Language in Social Media (LSM2012), pages 65–74, Montreal, Canada.

John M. Prager. 1999b. Linguini: Language identification for multilingual documents. Journal of Management Information Systems, 16(3):71–101.

Wang Ling, Guang Xiang, Chris Dyer, Alan Black, and Isabel Trancoso. 2013. Microblogs as parallel corpora. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 176–186, Sofia, Bulgaria, August. Association for Computational Linguistics.

W. J. Teahan. 2000. Text Classification and Segmentation Using Minimum Cross-Entropy. In Proceedings the 6th International Conference "Recherche d'Information Assistee par Ordinateur" (RIAO'00), pages 943–961, Paris, France.