### Corpus:

Web pages in Chinese, English, French, Japanese, Korean and Turkish (25 texts each)

### Sources :

“Language Identification from Text Using N-gram Based Cumulative Frequency Addition”

-> Language classifier using an ad-hoc Cumulative Frequency Addition of N-gram

(in comparison with 2 other methods: rank-order statistics and Naïve Bayesian classifier)

“Automatic Language Identification in Texts: A Survey”

“Language Identification of Short Text Segments with N-gram Models “

-> to complement some missing information

### State of the art:

* Collection of Text Samples and Creation of N-gram Profiles
  + test samples: Danish, English, French, Italian and Spanish online newspapers using a semi-automatic program written in VBA in Microsoft Access
  + training database: N-grams from 240 sample files from 12 Latin-character-based languages, 20 files in each language (English, Spanish, Italian, Danish, Polish, Swedish, Portuguese, German, French, Romanian, Dutch and Tagalog)
  + collect 2, 3, 4, 5, 6 and 7 grams from samples and store them with their counts of occurrence in a database table (delete those that occurs only once)
  + no preprocessing except database that contained only numbers (ex. “11”)
* N-gram Frequency Calculation
  + calculate the total N-gram counts for each language and the overall N-gram count for the entire training set

-> 2 frequencies for each N-gram (internal frequency and overall frequency)

FI (i, j) = C (i, j) / ∑i C (i, j)

FO (i, j)=C(i, j) / ∑i, j C(i, j)

FI (i, j) = Internal frequency of a N-gram i in language j  
FO (i, j) = Overall frequency of a N-gram i in language j  
C (i, j) = Count of the ith N-gram in the jth language  
∑i C (i, j) = Sum of the counts of all the N-grams in language j  
∑i, j C (i, j) = Sum of the counts of all the N-grams in all the languages

* + normalize the internal and overall frequencies of each N-gram by dividing each value by the highest frequency of the entire training database and then adding 1 to each value
* N-gram Rank Ordering
  + rank the N-grams in two different ways:
  1. internal rank ordering for each language by sorting all the N-grams within each language in descending order of frequency and ranking them from 1 to an incrementally higher number
  2. overall rank ordering for the entire training set by sorting the N-grams in descending order of language occurrence, ranking them from 1 to 12 (= 12 languages in the training database)

### Testing Procedures

* Classification by Rank-Order Statistics
  + tokenize each test string using N-grams of sizes 2, 3, 4, 5, 6 and 7
  + no preprocessing of the string
  + to classify the string using the rank-order statistical method, while tokenizing, count each N-gram and increment the counter if it occurred multiple times
  + sort the N-grams and create the rank ordered lists
  + by issuing a simple SQL and joining the test N-grams and the Training N-grams table, create a candidate N-grams list <- use these to perform the distant measurement
  + a default maximum distance of 1000 to a test N-gram without a match in the training database for any language
  + sum and sort the rank ordered distances from lowest to highest -> the language with the lowest number as the language category
* Classification using Cumulative Frequency Addition:
  + tokenize each test string using N-grams of sizes 2, 3, 4, 5, 6 and 7 and build an N-gram list
  + no preprocessing of string
  + provide the N-grams participating in the classification of both the training and test N-grams
  + delete from the calculation any test N-gram with no match in the training database for any language
* Classification using Naïve Bayesian Classifier
  + use the same set of candidate N-grams from above for the NBC method
  + instead of addition, multiply the normalized frequencies of all candidate N-grams from each language of the training set
  + the language that produced the highest number = identified as the correct one

### Results :

Pour la méthode N-gram, ça dépend des cas.

Les 3-grams avec la filtration :

|  | Précision | Rappel |
| --- | --- | --- |
| ENG | 1.0 | 1.0 |
| TRK | 0.5 | 0.14 |
| FRA | 0.5 | 0.85 |

Si on prend 2, 4, 6 grams avec filtration :

|  | Précision | Rappel |
| --- | --- | --- |
| ENG | 1.0 | 0.28 |
| TRK | 0.0 | 0.0 |
| FRA | 0.25 | 0.57 |

A partir de ces deux tableaux, il est facile de voir que le meilleur résultat est la précision et le rappel de la langue anglaise lorsqu'on prend 3-grams. Le plus mauvais résultat est la langue turc quand on fait 2,4,6 grams. C'est une différence très évidente.

Pourtant, alors qu’on fait N-gram, les résultats ne sont pas tout à fait ce qu'ils semblent être :

|  | Précision | Rappel |
| --- | --- | --- |
| ENG | 0.0 | 0 |
| TRK | 0.0 | 0.0 |
| FRA | 0.0 | 0.0 |

Par conséquent, cela indique qu'il y a certains problèmes. Ce résultat suggère que nous devons peut-être améliorer la collecte du corpus afin de nous assurer que nous pouvons au moins obtenir des résultats.

Pour la méthode Naïve Bayes, pour notre algorithme, nous avons utilisé scikit-learn qui fournit plusieurs bayes naïfs, nous avons choisi mutlnomial Naive Bayes (MultinomialNB()). Les résultats sont ci=dessous :

|  | Précision | Rappel | F-mesure |
| --- | --- | --- | --- |
| ENG | 0.89 | 1.00 | 0.94 |
| TRK | 1.00 | 1.00 | 1.00 |
| FRA | 1.00 | 0.86 | 0.92 |

Si nous faisons une comparaison, le classificateur bayésien donne de meilleurs résultats en termes de données.

Problems et Reflection :

* L'optimisation des scripts est nécessaire pour améliorer la précision de la détection. Disons que nous exécutons avec des résultats étranges, mais nous ne pouvons pas trouver une solution plus raisonnable pour le moment
* Pour le prétraitement du corpus. Nous avons suivi l'approche expérimentale des articles de N gram (les auteurs n'ont pas effectué ce processus de prétraitement), mais en pratique, nous avons constaté que le fait de ne pas effectuer cette étape aurait eu un certain impact sur les résultats ultérieurs.
* Nous manquons de temps pour réaliser chaque tâche du projet de manière plus détaillée.

Reference website :

<https://scikit-learn.org/stable/tutorial/basic/tutorial.html>

<https://www.kaggle.com/>