Récapitulatif

## “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)

### Avantages:

* Comparing to the conventional Naïve Bayesian classification method…
  + simpler
  + performs similarly in speed overall
  + better in accuracy on short input strings
* Comparing to N-gram based rank-order statistical classifiers…
  + 5-10 times faster

### 2 major approaches to text analysis:

* Linguistic models… realistic models
  + based on weighted Finite-State Transducers (using a lexical toolkit)
  + require language specific processing rules
* Statistical models… generic models
  + utilize different features from training samples to categorize tex

-> Several methods of feature extraction:

unique letter combinations, short word method [6, 7], N-gram method, ASCII vector of character sequences, Bayesian Decision Rules [10], Rank Order Statistics [11], K-Nearest Neighbor, and Vector Space Model

### Methodology of Cumulative Frequency Addition:

* 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)

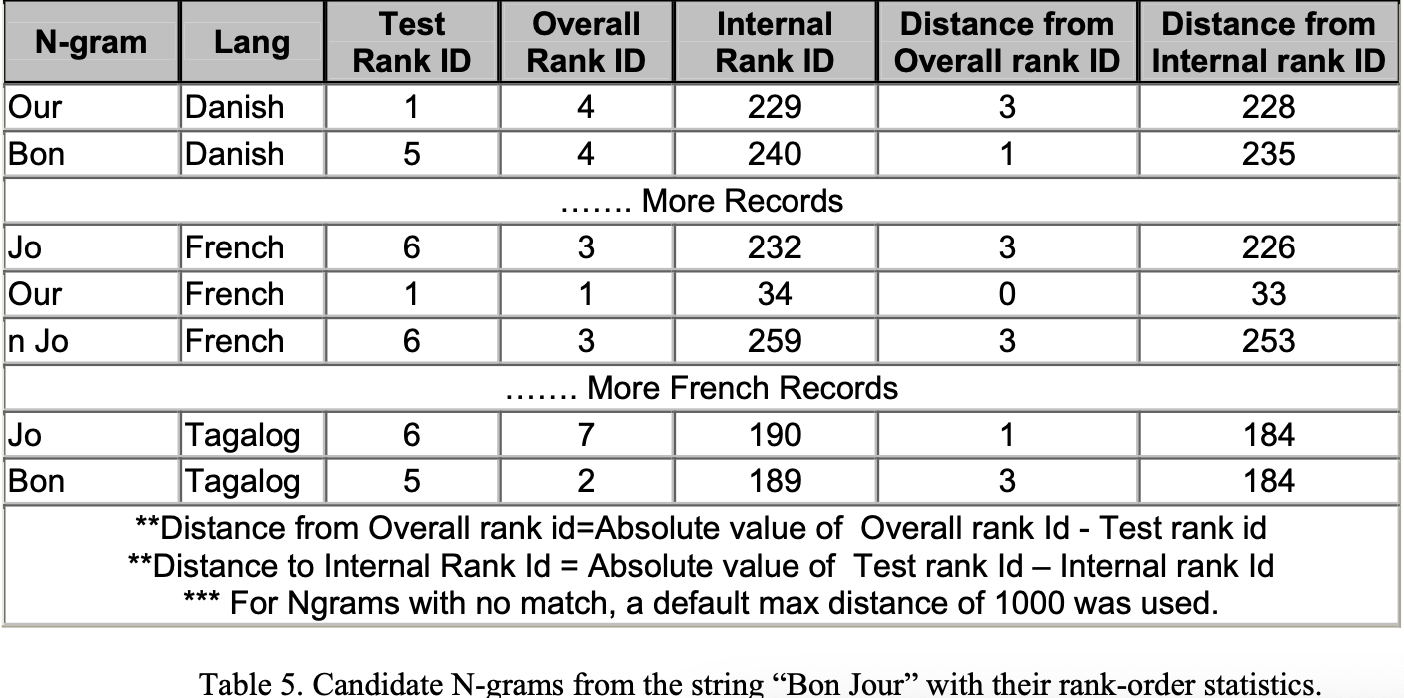
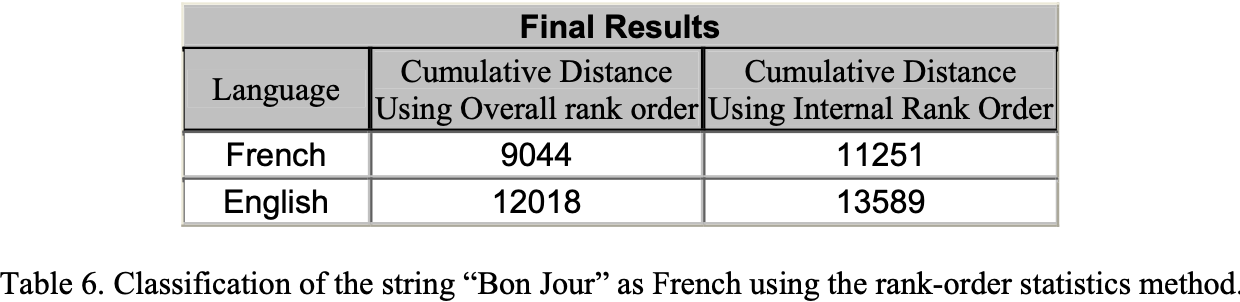
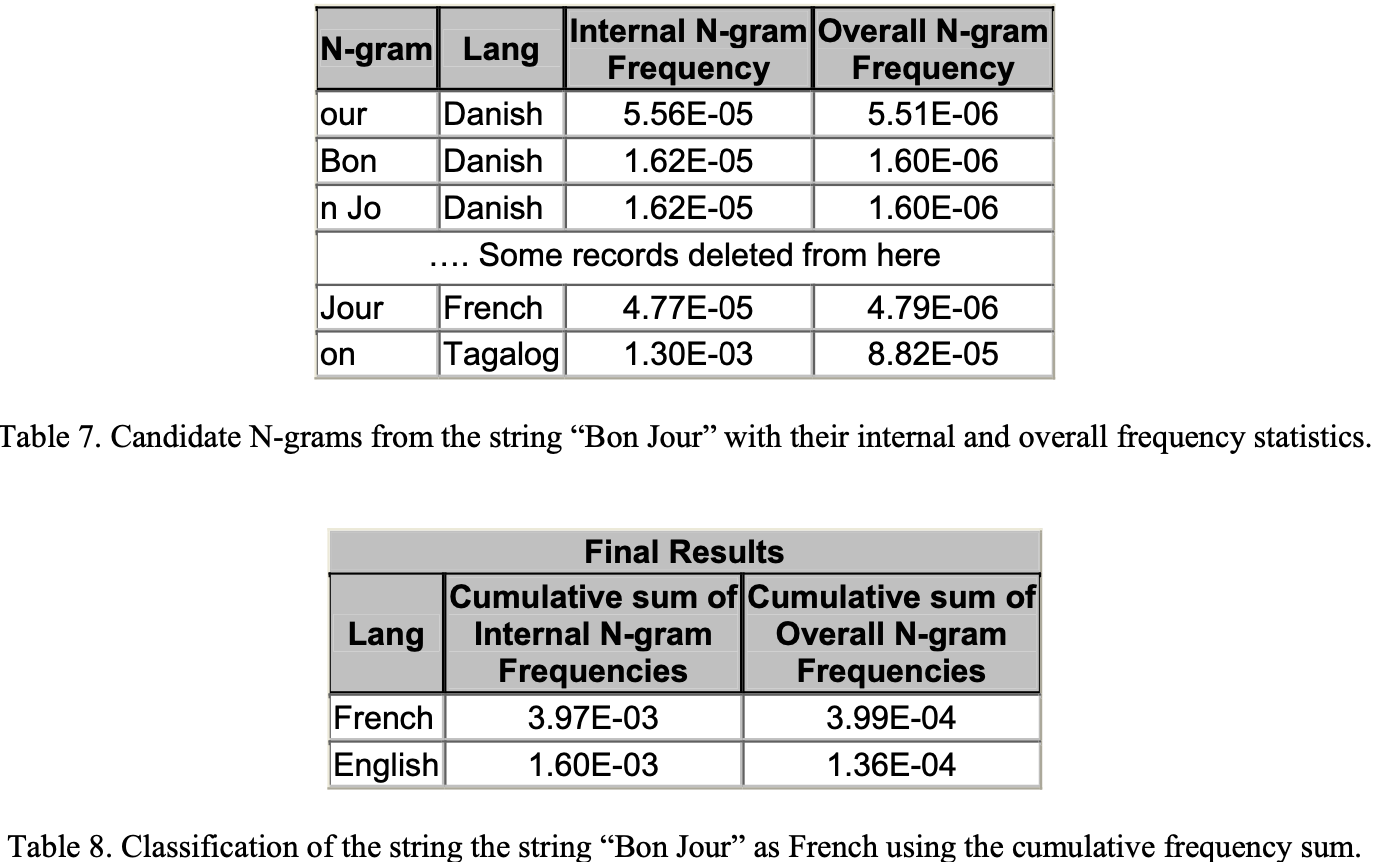
FI (i, j) = C (i, j) / ∑i C (i, j)

FO (i, j)=C(i, j) / ∑i, j C(i, j)

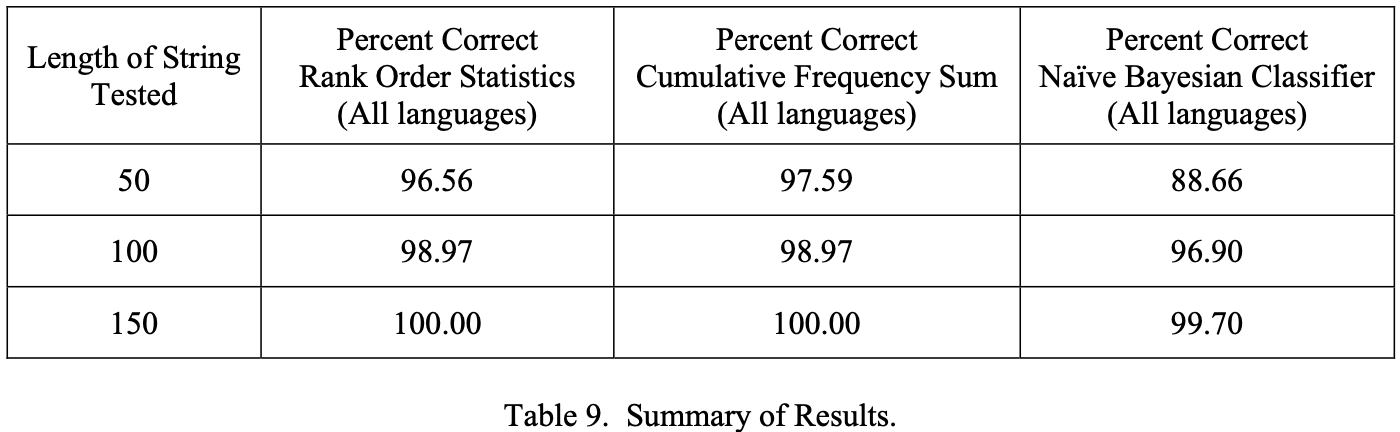
FI (i, j) = Internal frequency of a N-gram i in language j  
FO (i, j) = Overall frequency of a N-gram i in language j  
C (i, j) = Count of the ith N-gram in the jth language  
∑i C (i, j) = Sum of the counts of all the N-grams in language j  
∑i, j C (i, j) = Sum of the counts of all the N-grams in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

A total of 291 files from 5 different languages were used for testing: Danish 52, English 66, French 53, Italian 60 and Spanish 60.

### Conclusion

* The Naïve Bayesian classification methods
  + strength: speed of classification (extremely fast)
  + not have good accuracy with small strings comparing to the other methods
* The rank-order statistics method
  + strength: accuracy of classification
  + not appropriate for short strings because rank ordering and sorting is slow and requires long strings
* The strength of the cumulative frequency addition
  + strength: speed of classification
  + accuracy comparable to that of the rank-order statistics method in classifying short strings