Proceedings of Student/Faculty Research Day, CSIS, Pace University, May 7th, 2004

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

Bashir Ahmed, Sung-Hyuk Cha, and Charles Tappert

## Abstract

*This paper describes the preliminary results of an efficient language classifier using an ad-hoc Cumulative Frequency Addition of N-grams. The new classification technique is simpler than the conventional Naïve Bayesian classification method, but it performs similarly in speed overall and better in accuracy on short input strings. The classifier is also 5-10 times faster than N-gram based rank-order statistical classifiers. Language classification using N-gram based rank-order statistics has been shown to be highly accurate and insensitive to typographical errors, and, as a result, this method has been extensively researched and documented in the language processing literature. However, classification using rank-order statistics is slower than other methods due to the inherent requirement of frequency counting and sorting of N-grams in the test document profile. Accuracy and speed of classification are crucial for a classier to be useful in a high volume categorization environment. Thus, it is important to investigate the performance of the N-gram based classification methods. In particular, if it is possible to eliminate the counting and sorting operations in the rank-order statistics methods, classification speed could be increased substantially. The classifier described here accomplishes that goal by using a new Cumulative Frequency Addition method.*

*Keywords: language identification, cumulative frequency addition, Naïve Bayesian classification, rank order statistics*

## Introduction

Most Natural Language Processing (NLP) problems were once viewed as extremely difficult. In recent years, however, many of these problems have become more tractable due largely to the rapid growth in computing power and the advancements made in language processing algorithms. Research effort in NLP in both academic and commercial settings has increased greatly due to globalization and the need to communicate in an international corporation. The end result of all the effort has resulted in many commercial speech recognition, speech translation, and text-to-speech systems [1]. This trend is creating awareness of language processing (Speech recognition, speech translation and text-to-speech) as a viable tool, and language identification is the prerequisite of any such system. Corporate America is constantly looking to increase its bottom line by finding ways to increase productivity, and many corporations are trying to leverage speech recognition and text-to-speech technology for customer service. For example, automatic detection of a shift from one language to another in written text can play an important role in the quality of a text-to-speech (TTS) system. The accurate tagging of the words with correct language is critical for the other steps in TTS synthesis to complete successfully [9]. Thus, when a TTS system is turned on in English mode and it encounters a German word, if the system can detect that the next word is German, it can then automatically switch to German lexica and pronounce the word correctly in German. Authors in [3, 4] reported on such polyglot systems and they used heavy linguistic knowledge to resolve dual lingual text to separate English words from German text. This could increase the user acceptance of TTS systems and make commercial application more appealing. To accomplish this goal, we need to have a generic language classifier in front of the TTS module that can detect the language from short character strings.

The field of linguistic research has existed for a long time. As a result, we find many successful commercial grade language identifiers in the market today. However, not all identifiers are implemented the same way. There are two major approaches to text analysis – using linguistic models and statistical models. Sproat [8] reported on a text-analysis system based on weighted Finite-State Transducers where transducers were constructed using a lexical toolkit that allowed declarative description of lexicons, morphological rules, and phonological rules etc. While the linguistic models are realistic models, they are complex and require language specific processing rules. Statistical models are generic models and, using machine learning, they utilize different features from training samples to categorize text. Several methods of feature extraction have been used for language classification, including unique

letter combinations, short word method [6, 7], N-gram method, and ASCII vector of character sequences. Among the most reported classifiers are Bayesian Decision Rules [10], Rank Order Statistics [11], K-Nearest Neighbor, and Vector Space Model. For a language classifier to be useful, especially in applications like shifting from one language to another, it must be fast. This paper describes preliminary research on such a classifier.

## Methodology

Canvar and Trenkle [11] reported 99.8% correct classification rate on Usenet newsgroup articles written in different languages using rank-order statistics on N-gram profiles. They reported that their system was relatively insensitive to the length of the string to be classified, but the shortest strings that they reported classifying was 300 Bytes (characters). They classified documents by calculating the distances of a test document's N-gram profile from all the training language N-gram profiles and then taking the language corresponding to the minimum distance. In order to perform the distance measurement they had to sort the N-grams in both the training and test profiles. Table 1 illustrates how they performed their calculation.

| | Language profile | Test document profile | Out of place |
|---|---|---|---|
| Most Frequent | TH | TH | 0 |
| | ER | ING | 3 |
| | ON | ON | 0 |
| | LE | ER | 2 |
| | ING | AND | 1 |
| Least Frequent | AND | ED | No-match = max distance |
| | | | |
| Test Document Distance from Language = Sum of out-of-place values | | | |

Table 1. Distance calculation using rank-order statistics [11].

Their method is simple and works well in identifying the language from text string of 300 bytes or more. The method is also insensitive to typographical errors. However, the authors did not provide any information on the speed of classification rate, only mentioning that their training N-gram profiles were small. In this paper, we focus on the speed of classification rate using N-gram based classification as the underlying method. The two main problems with their classification scheme are the requirements to count the frequency of each N-gram in the test document and to sort the N-grams to perform distance measurement. It is well known that executing a database query with a sort operation is a resource intensive and time consuming operation, and that elimination of sorting from computing algorithms will enhance performance. Here we introduce a new classifier using a similar N-gram profile, but without requiring a sort operation on either the training or testing N-gram profiles. Instead of rank-order statistics, we use Cumulative Frequency Addition (CFA) to classify test documents into different categories. For purposes of comparison, using the same training dataset, we performed classification using the rank-order statistical method and the Naïve Bayesian classification method.

## Collection of Text Samples and Creation of N-gram Profiles

Text samples were collected in 12 languages. For example, we collected test samples from Danish, English, French, Italian and Spanish online newspapers using a semi-automatic program written in VBA in Microsoft Access. The sample collection program runs by opening Internet Explorer to a given URL, such as a newspaper website, extracting text strings from the active page using Microsoft's dynamic HTML features (MSHTML object library), and storing the content in a text file. To keep track of which file came from which website, the program kept an URL/File mapping table in the database with the URL location and the name of the file created from that URL.

We created a training database containing N-grams from 240 sample files from 12 Latin-character-based languages, 20 files in each language. The twelve languages were English, Spanish, Italian, Danish, Polish, Swedish, Portuguese, German, French, Romanian, Dutch and Tagalog. The training sample sizes ranged from 65K to 105K as shown in Table 2. We collected 2, 3, 4, 5, 6 and 7 grams from these language samples and stored them with their counts of occurrence in a database table. After collecting the N-grams, we deleted those that occurred only once, greatly reducing the number of N-grams.

Many authors reported preprocessing of training data [2, 4, 5, 6, 10, 11], such as getting rid of punctuation marks such as commas, numbers and special characters, before collecting N-gram statistics. Unlike those authors, however, no preprocessing was performed on these training data. Rather, the N-grams were obtained by simply reading each line of text and segmenting the string into different size N-grams, moving forward character by character. Some authors also reported replacing space character with "_" (underscore), but we left space as a valid character. By not preprocessing we had many N-grams in the training database that contained only numbers such as "11", "22", "011" etc. These N-grams were deleted since they are not much help in differentiating among languages.

| Language | Number of training files | Size of all training files | Total N-grams | N-gram Sizes |
|---|---|---|---|---|
| Danish | 20 | 88K | 41,485 | 2 to 7 |
| Dutch | 20 | 67K | 30,276 | 2 to 7 |
| English | 20 | 81K | 36,633 | 2 to 7 |
| French | 20 | 92K | 42,108 | 2 to 7 |
| German | 20 | 80K | 35,524 | 2 to 7 |
| Italian | 20 | 65K | 29,878 | 2 to 7 |
| Polish | 20 | 104K | 41,116 | 2 to 7 |
| Portuguese | 20 | 67K | 33,574 | 2 to 7 |
| Romanian | 20 | 105K | 40,625 | 2 to 7 |
| Spanish | 20 | 65K | 32,983 | 2 to 7 |
| Swedish | 20 | 89K | 38,591 | 2 to 7 |
| Tagalog | 20 | 65K | 29,316 | 2 to 7 |
| **Totals** | **240** | **968K** | **432,109** | |

Table 2. Training sample size and the corresponding N-gram statistics.

## N-gram Frequency Calculation

After eliminating the unnecessary N-grams, we calculated the total N-gram counts for each language as well as the overall N-gram count for the entire training set. We calculated two frequencies for each N-gram – the internal frequency and the overall frequency – as follows:

$$F_I(i, j) = C(i, j) / \sum_i C(i, j)$$
$$F_O(i, j) = C(i, j) / \sum_{i,j} C(i, j)$$

$F_I(i, j)$ = Internal frequency of a N-gram i in language j
$F_O(i, j)$ = Overall frequency of a N-gram i in language j
$C(i, j)$ = Count of the $i^{th}$ N-gram in the $j^{th}$ language
$\sum_i C(i, j)$ = Sum of the counts of all the N-grams in language j
$\sum_{i,j} C(i, j)$ = Sum of the counts of all the N-grams in all the languages

Table 3 shows a sample calculation using a hypothetical Ngram "xyz".

| Language | N-gram | N-gram Count | Total N-grams in this Language | Total N-grams in all Languages | Internal Frequency | Overall Frequency |
|---|---|---|---|---|---|---|
| English | xyz | 10 | 100 | 1000 | 10/100 | 10/1000 |
| Danish | xyz | 15 | 150 | 1000 | 15/150 | 15/1000 |
| French | xyz | 10 | 200 | 1000 | 10/200 | 10/1000 |
| Italian | xyz | 15 | 300 | 1000 | 15/300 | 15/1000 |
| Tagalog | xyz | 3 | 40 | 1000 | 3/40 | 3/1000 |
| German | xyz | 13 | 60 | 1000 | 13/60 | 13/1000 |
| Spanish | xyz | 28 | 150 | 1000 | 28/150 | 28/1000 |

Table 3. A sample of how internal and overall frequencies are calculated.

The internal and overall frequencies of each N-gram were normalized by dividing each value by the highest frequency of the entire training database and then adding 1 to each value. Thus, the final value of each N-gram frequency was normalized to a value between 1 and 2. Table 4 shows sample data from our training database with internal and overall frequencies before normalization.

## N-gram Rank Ordering

In the training set, we rank ordered the N-grams in two different ways. First, the internal rank ordering was done 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. Second, the overall rank ordering was done for the entire training set by sorting the N-grams in descending order of language occurrence, ranking them from 1 to 12 since there are 12 languages in the training database. Note that in the internal rank ordering scheme, an N-gram could be ranked from 1 to the highest ranked position in that language training set. Table 4 shows sample data on these two rank ordering schemes. Canvar and Trenkle previously described internal rank ordering [11].

| N-gram | N-gram Count | Language | Internal Rank Order | Overall Rank Order | Internal Frequency | Overall Frequency |
|---|---|---|---|---|---|---|
| minist | 75 | Portuguese | 186 | 1 | 1.75E-04 | 1.33E-05 |
| minist | 61 | Italian | 192 | 2 | 1.45E-04 | 1.08E-05 |
| minist | 49 | Danish | 220 | 3 | 8.78E-05 | 8.70E-06 |

Table 4. Sample data with calculated internal and overall rank orders, and internal and overall frequencies.

The data in Table 4 shows that the internal rank ordering of the N-gram "Minist" was ranked 186th in Portuguese, 192 in Italian, and 220 in Danish. However, this same N-gram when sorted for the entire training set, ranked 1 for Portuguese (count 75), 2 in Italian (count 61), and 3 in Danish (count 49).

## Testing Procedures

## Classification by Rank-Order Statistics

Each test string was tokenized using N-grams of sizes 2, 3, 4, 5, 6 and 7. Just as in the training set, no preprocessing of the string was performed. To classify the string using the rank-order statistical method, while tokenizing, we kept count of each N-gram and incremented the counter if it occurred multiple times. Since we were using a Microsoft Access database, we did this by inserting each N-gram as a record in a test table and updated the N-gram count field of the record on each additional occurrence. We used a Microsoft Access database because we were familiar with it and because the N-grams in both the training and test data were persistent in the tables so we could visually examine each N-gram and check the resulting N-gram list that participated in the classification. After tokenizing and computing N-grams counts, we sorted the N-grams and created the rank ordered lists.

Once we had the training N-grams and the test N-grams ranked with rank order ids, by issuing a simple SQL and joining the test N-grams and the Training N-grams table, we came up with a candidate N-grams list and used these to perform the distant measurement. A test N-gram that did not have a match in the training database for any language was given a default maximum distance of 1000. Canvar and Trenkle used similar maximum distance for unmatched Ngrams, but did not specify what the maximum distance was. Table 5 shows the list of candidate N-grams that would have been used to classify the string "Bon Jour."

| N-gram | Lang | Test Rank ID | Overall Rank ID | Internal Rank ID | Distance from Overall rank ID | Distance from Internal rank ID |
|---|---|---|---|---|---|---|
| Our | Danish | 1 | 4 | 229 | 3 | 228 |
| Bon | Danish | 5 | 4 | 240 | 1 | 235 |
| ……. More Records | | | | | | |
| Jo | French | 6 | 3 | 232 | 3 | 226 |
| Our | French | 1 | 1 | 34 | 0 | 33 |
| n Jo | French | 6 | 3 | 259 | 3 | 253 |
| ……. More French Records | | | | | | |
| Jo | Tagalog | 6 | 7 | 190 | 1 | 184 |
| Bon | Tagalog | 5 | 2 | 189 | 3 | 184 |
| **Distance from Overall rank id=Absolute value of Overall rank Id - Test rank id<br>**Distance to Internal Rank Id = Absolute value of Test rank Id – Internal rank Id<br>*** For Ngrams with no match, a default max distance of 1000 was used. | | | | | | |

Table 5. Candidate N-grams from the string "Bon Jour" with their rank-order statistics.

The rank ordered distances were then summed and sorted from lowest to highest, and the language with the lowest number was selected as the language category. Table 6 shows the result for 'Bon jour."

| Final Results | | |
|---|---|---|
| Language | Cumulative Distance Using Overall rank order | Cumulative Distance Using Internal Rank Order |
| French | 9044 | 11251 |
| English | 12018 | 13589 |

Table 6. Classification of the string "Bon Jour" as French using the rank-order statistics method.

## Classification using Cumulative Frequency Addition

Similar to the rank-order statistics method, each test string was tokenized using N-grams of sizes 2, 3, 4, 5, 6 and 7. Again, as with the training set, no preprocessing of the string was performed. To classify the string using the cumulative frequency addition method, we simply tokenized the string and built an N-gram list. We did not have to keep count of occurrence for each N-gram and did not have to sort the N-grams. After tokenizing the string, the resulting N-gram list may have contained multiple repetitions of the same N-gram. The following simple SQL statement operates on the training and test N-grams to provide the N-grams participating in the classification (All_Training_N-grams is the name of the database table where the training N-gram profiles are stored).

**Select** N-gram, language, internal_frequency, overall_frequency from All_Training_N-grams where N-gram in ("Test N-gram list")

Again, any test N-gram that did not have a match in the training database for any language was dropped from the calculation. For the string "Bon Jour," Table 7 shows a list of candidate N-grams with their internal and overall frequencies, and Table 8 shows the final result.

| N-gram | Lang | Internal N-gram Frequency | Overall N-gram Frequency |
|---|---|---|---|
| our | Danish | 5.56E-05 | 5.51E-06 |
| Bon | Danish | 1.62E-05 | 1.60E-06 |
| n Jo | Danish | 1.62E-05 | 1.60E-06 |
| …. Some records deleted from here | | | |
| Jour | French | 4.77E-05 | 4.79E-06 |
| on | Tagalog | 1.30E-03 | 8.82E-05 |

Table 7. Candidate N-grams from the string "Bon Jour" with their internal and overall frequency statistics.

| Final Results | | |
|---|---|---|
| Lang | Cumulative sum of Internal N-gram Frequencies | Cumulative sum of Overall N-gram Frequencies |
| French | 3.97E-03 | 3.99E-04 |
| English | 1.60E-03 | 1.36E-04 |

Table 8. Classification of the string the string "Bon Jour" as French using the cumulative frequency sum.

## Classification using Naïve Bayesian Classifier

The same set of candidate N-grams from above was used for the NBC method. In this case, instead of addition, we multiplied the normalized frequencies of all candidate N-grams from each language of the training set. The language that produced the highest number was 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. Table 9 shows the number of files tested in each language with the % accuracy obtained in all three methods. Figure 1 shows the results in from 50, 100, and 150 byte strings using the three classification methods. For files of 150 Bytes (characters) in length, the rank-order statistics and cumulative frequency addition methods attained an accuracy of 100%, while the Naïve Bayesian classifier attained 99.7%. The cumulative frequency addition and the Naïve Bayesian methods were of comparable speed and 3-10 times faster than the rank-order statistics method (Table 10).

| Length of String Tested | Percent Correct Rank Order Statistics (All languages) | Percent Correct Cumulative Frequency Sum (All languages) | Percent Correct Naïve Bayesian Classifier (All languages) |
|---|---|---|---|
| 50 | 96.56 | 97.59 | 88.66 |
| 100 | 98.97 | 98.97 | 96.90 |
| 150 | 100.00 | 100.00 | 99.70 |

Table 9. Summary of Results.

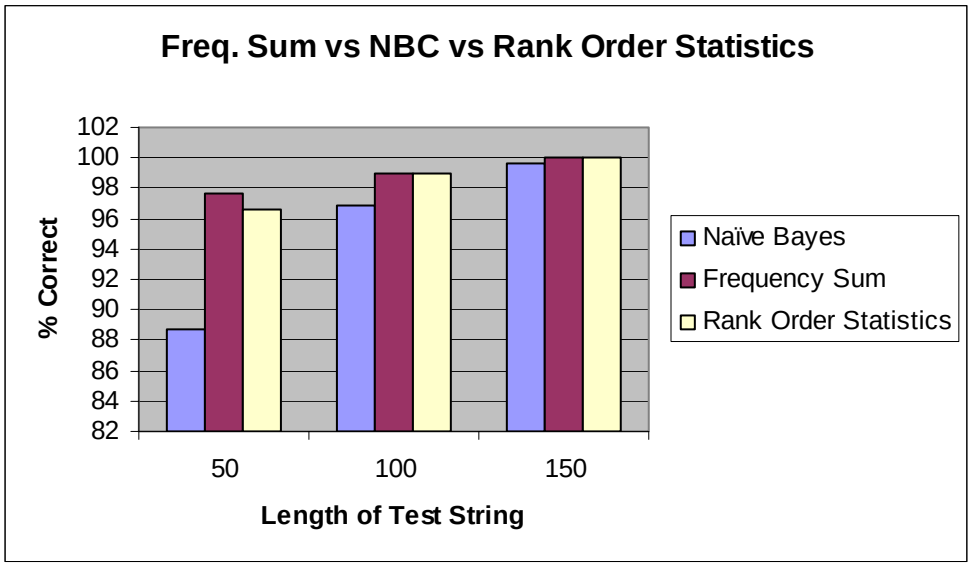


Figure 1. Percent accuracy of classification of NBC, CFA, and rank-order statistics.

| Language | String Length | Cumulative Frequency Addition (seconds) | Rank-Order Statistics (seconds) | Ratio |
|---|---|---|---|---|
| French | 236 | 0.46 | 4.6 | 10.00 |
| French | 206 | 0.48 | 4.1 | 8.54 |
| French | 160 | 0.44 | 3.2 | 7.27 |
| French | 134 | 0.37 | 2.8 | 7.57 |
| French | 101 | 0.37 | 2.1 | 5.68 |
| French | 75 | 0.32 | 1.8 | 5.62 |
| French | 51 | 0.32 | 1.2 | 3.75 |

Table 10. Speed of classification for cumulative frequency addition versus rank-order statistics.

## Conclusion

The strength of the cumulative frequency addition and the Naïve Bayesian classification methods are the speed of classification while the strength of the rank-order statistics method is its accuracy of classification. Furthermore, the accuracy of the CFA method described here is comparable to that of the rank-order statistics method in classifying short strings. The speed of classification of the new approach can be particularly useful when classifying mixed lingual texts where the major language is to be identified from long test strings and the minor languages from short strings. The rank-order statistics method is not appropriate for short strings because rank ordering and sorting is slow and requires long strings. Although the Naïve Bayesian classification method is extremely fast, it did not have good accuracy with small strings where it performed poorly compared to the other methods. The reason the CFA methods works well on the language classification problem might be attributable to linguistic properties of this problem. NBC assumes that the probabilities of unique N-grams are independent of other N-grams, and this is not a good assumption, especially when dealing with language. For example, the occurrence of “th” increases the probability that the word may be “the” or ‘that.” Thus, assuming that “th” is independent of “that” or “the” is not valid. This may explain why frequency addition performs better than NBC with short strings. To more verify these preliminary findings, more testing is needed on larger data sets, and this work is in progress.

## References


[1] A Burdick, The Mathematics of . . . Artificial Speech: Two centuries of tinkering finally produce a sweet-talking machine, *DISCOVER* Vol. 24 No. 01, January 2003

[2] B. Pfister and H. Romsdorfer, Mixed Lingual Text Analysis for Polyglot TTS Synthesis, *Proceedings of Eurospeech 2003, Geneva, Switzerland, September 1-4, 2003*

[3] B. Pfister, E. Wehrli et al. *Lexical and Syntactic Analysis of Mixed-Lingual Sentences for Text-to-Speech*. Final Report of SNSF Project No 21-59396.99. Institut TIK, ETH Zurich, November 2002.

[4] C. Traber, B. Pfister, et al. From Multilingual to Polyglot Speech Synthesis. In *Proceedings of the Eurospeech*, pages 835–838, September 1999.

[5] E. Giguet, Multilingual Sentence Categorization According to Language, *Proceedings of the European Chapter for Computational Linguistics SIGDAT workshop*, Dublin, March 1995.

[6] E. Giguet, Categorization According to Language: A Step Toward Combining Linguistic Knowledge and Statistics Learning, *Proceedings of the International workshop for Parsing Technologies*, Prague-Karlovy Vary, Czech Republic, September 20-24, 1995.

[7] E. Giguet, The Stakes of Multilinguality: Multilingual Text Tokenization in Natural Language Diagnosis, *Proceedings of the 4rth International Conference on Artificial Inteligence Workshop*, Cairns, Australia, August 27, 1996.

[8] R. Sproat. Multilingual Text Analysis for Text-To-Speech Synthesis. In *Proceedings of the ICSLP'96*, Philadelphia, October 1996.

[9] Thiery Dutoit, High Quality Text-to-Speech Synthesis: An Overview, *Journal of Electrical & Electronics Engineering*, Australia: Special Issues on Speech Recognition and Synthesis, Vol. 17, No 1, Pages 25-37

[10] Ted Dunning, Statistical Identification of Languages, Computing Research Laboratory, New Mexico State University, March 10, 1994.

[11] W. B. Canvar and J. M. Trenkle, N-gram based Text Categorization, *Symposium on Document Analysis and Information Retrieval*, Pages 161-176, University of Nevada, Las Vegas, 1994.