

# Research Problem Statement

Dean Earl Wright

May 3, 2005

## **Abstract**

The number of network accessible documents increases hourly. These documents may or may not be in the reader's native language. Determining the language of the document is required to effectively apply many search and information retrieval techniques. Several methods exist for determining the language of an electronic document. This research will examine the effectiveness of integrating those techniques into existing document processing systems.

A pair of text classification systems (Ad Hoc and Statistical) will be extended to include language identification as part of their processing. The accuracy of the language identification will be measured as well as the additional processing time required.

## **1 Introduction**

One need only look at the amount of material available on the web over the past few years to see the growth in the number of documents available electronically. Add to that e-mail, instant messaging, and news groups and there is a staggering amount of electronic text available. This material can be written in any one of a dozen major languages or hundreds of lesser ones. People can generally recognize the languages that they normally come into

contact with but, with the click of the mouse, you can retrieve almost anything from almost anywhere in almost any language.

Determining the language of an electronic document has been an area of active research for many years. Numerous methods have been proposed based on linguistic, statistical, and vector space analysis of the unknown document. All of the methods work. That is, they do identify the language of the electronic text fairly accurately. The level of accuracy varies but is almost always above 90% and generally above 98% once the input text size is sufficient. What differentiates the methods is their computational requirements, their ability to provide an estimate of the accuracy of their identification, the ability to handle noise in the input (misspelled words or the inclusion of passages from multiple languages, the amount of training text required, and the ability to scale in order to handle many languages.

Language determination is often the first step in a text processing system that may involve machine translation, semantic understanding, categorization, routing or storage for information retrieval. Knowing the language of the text allows the correct dictionaries, sentence parsers, profiles, distribution lists, and stop-word list to be used. Incorrectly identifying the language would result in garbled translations, faulty or no information analysis, and poor precision and recall in searching.

While existing language identification methods can produce reasonable results, they often do so at a large computational cost (in terms of both space and time). Many methods require large lists of words and/or n-grams with associated frequency counts for each language. Others require matrices whose size is dependent on the number of unique words and the number of documents in the reference language set. Calculations on large lists and matrices make these methods expensive to use.

To reduce the expense of language determination, the language determination method should be integrated into the larger text processing system and make use of the data structures and calculations performed. When integrated at this level, language determination

can be done without causing a negative impact on the system.

To demonstrate the integration of the language determination function within a text processing system, I propose to add language determination to a text categorization system so that the system is capable of identifying several (at least a dozen) European Union languages (encoded using Latin-1) with at least 98% accuracy and not require more than 5% additional processing as measured by CPU usage.

## 2 Background

Imagine that your job is to sort incoming mail at an embassy or consulate. The incoming mail is to be sorted into four piles:

**visa requests** The most numerous type containing requests for tourist, student, and work visas.

**citizen matters** Requests for replacement passports and other assistance to the traveling citizen.

**cultural exchange** Requests and offers for the sharing of music, art, and stage productions.

**other** Any items not covered above including military and intelligence matters.

Most of the mail is addressed to the ambassador so the address is of little help. Each letter must be opened and the contents (or at least a portion of them) read to determine the proper pile.

This kind of problem is one of Text Classification where you must choose the most likely category for each document. While the category is usually a subject area, categories can also be anything that divides the documents into subsets. Reading level, author's sex, suitability for employment can all be used as categories. Identifying the language of

a document is also a Text Classification problem where the category you are trying to determine is the document's language.

## **2.1 Text Classification Methods**

Many different techniques can be used for Text Classification. They can be divided into three categories representing both a general chronological order and that of increasing complexity.

### **2.1.1 Ad Hoc Text Classification Methods**

If you classify the embassy's mail by looking for a key word such as visa to select mail for the visa request category, you are using an Ad Hoc classification method. Ad Hoc classification systems require knowledge of what should and should not be in each category. This is usually obtained from the people currently performing the classification when automating an existing manual process. The expert knowledge is then embodied within the system as rules or searches. While the general techniques of ad hoc classification (e.g. searching for key words or phrases) can be used in different systems, each system will require an expert to craft the specific rules.

### **2.1.2 Statistical Text Classification Methods**

By examining a number of documents that belong to each category, a statistical model of the perfect document for that category can be made. Any one of several of many different items can be used for the statistical measures. Often it is short groups of characters, called n-grams. When faced with the task of putting a document into a category, the statistical measures of the new document are taken and then compared to the statistics of each of the categories. The category whose statistics are the closest match to the new document is the category selected.

### **2.1.3 Latent Semantic Analysis for Text Classification**

One statistic that can be used is the number of times each word appears in the document. This is done for a number of documents. A processing and memory intensive procedure is then performed to obtain the Single Value Decomposition of the words by document matrix mapping each document in an  $n$ -dimensional space. This information is used to cluster the documents into groups. Each cluster represents a different category. To categorize a new document, its words are processed to map it into the  $n$ -dimensional space and then determine the cluster to which it is closest.

## **2.2 Language Identification Methods**

As mentioned above, Language Identification, is a form of text classification. Thus, any of the techniques of text classification can be applied to language identification.

### **2.2.1 Ad Hoc Language Identification Methods**

Ad Hoc or Linguistic methods use some aspect of the language to identify it. These approaches are usually dictionary-based with a set of words from each language. As large dictionaries take longer to process, the dictionaries are usually lists of the short, common words in the language such as pronouns, articles, and prepositions. (In text retrieval systems these lists would be the "stop" words lists used to prevent indexing on words with little semantic meaning.) When trying to identify the language of an unknown document, each list is consulted and the document is declared to be of the language with the most matching words.

Norman Ingle[4] created a method using only one and two letter words. While designed for use by librarians it is easily automated. One starts by assuming that the unknown document could be of any language. Then the one and two letter words are examined to see which languages use that word. Languages that don't use the word are eliminated. The process continues until all the one and two letters words have been examined or until all

but a single language is identified.

### **2.2.2 Statistical Language Identification Methods**

As with statistical test categorization, sample documents from each language are subjected to a statistical analysis based on some characteristic of the text. This is often n-gram character analysis or counts of short words. Gregory Grefenstette[3] compared the use of small words and tri-grams. Tri-grams worked better on short sentences but as the length of the text grew larger, they both performed about the same.

To determine the language of a document, the same statistical procedure is done to the new document and the document's statistics are compared to those derived from the language training documents to determine the closest match.

### **2.2.3 Vector Language Identification Methods**

Laura Mather[5] provided a vector space model using single value decomposition. All of the words in the unknown document are added as another column of an  $m$  by  $n$  where  $m$  is the number of unique words and  $n$  is the number of documents. An iterative algorithm partitions the matrix so that the unknown document will be clustered with other documents of the same language. The matrix processing requirements are alleviated somewhat but using the power method instead of full SVD as only the parts of SVD results are used. Experiments using subsets of fifteen languages achieved only 92% correct identification. There was the common problem of the closeness of Portuguese, Spanish and Italian, but problems were also caused by classifying the tab character as a word in several languages and having a small number of samples of some languages.

## **3 Technical Approach**

### **3.1 Selection of Compatible Text Classification and Language Identification Methods**

In order to integrate a language identification capability into a text classifying system without a significant increase in processing, the two methods need to share the same data structures and do as much of the processing as possible in common. The obvious starting place is to select methods that are both either ad hoc, statistical, or vector based.

I propose to investigate two such matched pairings. Julian Yochum[6] describes a fast text searching technique used to implement an ad hoc text classification. The fast text searching will be used with stop word lists from various languages to do ad hoc language identification. The same fast search technique was used to implement a statistical text classification method[7] which which can be paired with a statistical language identification technique.

### **3.2 Ad Hoc Text Classification with Language Identification**

The fast text search method will be implemented in Python (or Java - the final choice has not been made). This will include programs to determine trigraph frequencies from a training set of documents, optimize searches based on trigram frequencies, and classify a document by executing the optimized searches. Additionally, classification searches and execution scripts will be created.

The document classification program will be extended to run additional searches based on stop word lists for languages. A program to create classification searches from stop word lists will be created as well as additional execution scripts.

### **3.3 Statistical Text Classification with Language Identification**

Given the fast text searching program generated for the testing of the ad hoc methods, classification using statistical measures will require little additional coding. Again, a text classification program will be created and then augmented to include language identification.

A program will be developed to create classification searches from the trigram frequencies of the training documents. This program will be run against category training documents to obtain the category matching statistics and against the language training documents to obtain the language matching statistics. Execution scripts will be created to automate the running of the programs.

## **4 Methodology**

### **4.1 Obtaining Test Documents**

Documents from many languages will be needed to test the combined text classification and language identification system. Many documents are available on the internet as well as from other sources (e.g. USENET).

#### **4.1.1 CDROM**

A couple of papers reported using the European Corpus Initiative Multilingual Corpus 1 CDROM from the Association for Computing Linguistics. Using this corpus of documents has the advantages of having had other language identification researchers already examine it to find any weaknesses and it will provide a basis of comparison with their results.

#### **4.1.2 Criteria for Usable Languages**

The ECIMC1 CDROM contains samples of 23 languages but not all can be used. Several of the languages have only one or two documents and some are not in the Latin-1

encoding. This leaves about 15 languages which will be enough to validate the processing.

### **4.1.3 Extraction and Verification of Documents**

For each usable language on the ECIMC1 CDROM, the documents will be examined to determine suitability for the testing (e.g. Dictionaries and multi-language parallel texts will be discarded). The remaining documents will be broken into fifty-line (approximately one page) files. These files will be processed using Ingle's method to validate the target language.

### **4.1.4 Training and Experimental Documents**

The statistical text classification and language identification methods require training documents. One third of the files of each language (randomly chosen) will be reserved as the training set for that language. The remaining documents will be randomly split into two test sets.

All three sets of document files (the training set and the two test sets) will be processed to obtain the trigram frequency counts. A Friedman two-way analysis of variance by ranks will be done to ensure that any differences in the trigram frequencies between the three sets is due only to randomness. Except for this verification, the trigram frequency numbers from the two test sets will not be used.

## **4.2 Creation of Test Software**

As mentioned above, several pieces of software will be created to test language identification within a text classification system. The programs will be created in an object-oriented manner to facilitate reuse among the components and to keep the actual coding to a minimum.

### **4.2.1 Trigraph Extraction and Frequency Counting**

The fast text search algorithm uses only a reduced character set of the 26 uppercase letters, the numbers, and a blank. This gives a 37 character set to which 3 additional (unused) characters are added giving 40 characters total. When counting trigrams from a training set, the 40x40x40 matrix is initialized to all zeroes. Then, for all document in the training set, the characters in a document are mapped to the reduced character set. As the trigrams are extracted from the reduced character set document, the appropriate entry in the matrix is incremented. After processing all documents in the training set, the counts in the matrix are scaled and written to a file.

A table with the mapping of the characters of the Latin-1 set to the reduced character set will be created. This mapping will be encapsulated in a reusable object. The operations on the trigram matrix will also be done in a reusable object. Using these objects, a program to create the frequency file will be created.

### **4.2.2 Search Optimization**

The Text Classification routine needs to have the searches ordered by the least frequent trigram in order to process efficiently. A program will be created to read in a file containing a search (search terms connected with ANDs and Ors) and order the internal search testing order based on a trigram frequency table. The optimized search is a table of trigrams to check and action to be taken based on success or failure. The table is output to a file for input into the text classification routine.

### **4.2.3 Text Classification Routine**

The text classification routine takes a number of optimized searches and unknown documents and, depending on the results of the search, assigns one or more categories to each document. The classifier loads all of the searches and then processes each document in turn. For each document, the trigram searching tables are built. Then each search is ex-

ecuted against the tables and is evaluated as true or false. The name of each search that evaluated to true is associated with the document as its category.

The character set mapping and trigram extraction objects from the trigram frequency counters are reused. The search evaluator uses the optimized search action tables and the document's trigram searching tables to evaluate each search. The results for each document are written to a file showing which searches succeeded. The same text classifier routine is used for both ad hoc and statistical categorization. The difference is how the searches are constructed.

#### **4.2.4 Language Identification Routine**

This is the same as the text classification routine, but two sets of optimized searches are used: one for text classification and a second set for language identification. The same text classifier routine is used for both ad hoc and statistical language identification. The difference is, again, how the searches are constructed.

#### **4.2.5 Ad Hoc Text Classification Searches**

In order to establish the processing baseline for the text classification task, the text classifier will need a set of text classification searches. These searches will divide the English language documents from the ECIMC1 CDROM into a number of categories. The exact categories will be determined after a review of the documents. Several dozen categories will be selected and a search created for each category.

#### **4.2.6 Ad Hoc Language Identification Searches**

Ad hoc language identification will be done by matching a document against a list of the common language words for each language. Stop word lists are available for several languages. A program will be created to take a list of words (one per line) and create a search from all the words.

### **4.2.7 Statistical Text Classification Searches**

In order to do statistical text classification, a set of searches will be programatically generated from the trigram frequencies of the documents in each category. A program will be developed that takes a trigram frequency table and produces searches for the top  $n$  searches where  $n$  can be set as needed.

### **4.2.8 Statistical Language Identification Searches**

The process for creating these searches is the same as for creating the statistical text classification searches. The same program will be used. The only differences is that the trigram frequencies of the language training sets will be used as input to the search generator.

### **4.2.9 Test Scripts**

All of the work to set up to run a test, the running of the test, and the evaluation of the results will be done under the control of a test script. This will allow rerunning of the tests as needed and prevent that most unstable of elements (the researcher) from making simple mistakes that invalidate the test results.

## **4.3 Experiments**

### **4.3.1 Ad Hoc**

The training set documents are processed to obtain the trigram frequencies. These trigram frequencies will be used to optimize the text categorization and language identification searches.

Test sets one and two are will be processed by the Ad Hoc Text Classification System with the running time and the document categories recorded. Test sets one and two will then be processed again using the Ad Hoc Text Classification System augmented with the ad hoc language identification searches with the running time, the document categories, and

the identified language recorded. The tests will be repeated ten times to account for any variability in processing times for a total of forty experimental runs for ad hoc processing.

### **4.3.2 Statistical**

The training set documents, divided into groups, will be processed to obtain the trigram frequencies. These trigram frequencies will be used to create the category and language identification searches. The searches will be optimized using the overall trigram frequencies obtained in the ad hoc processing

Test sets one and two will be processed by the Text Classification using the statistically generated text classification profiles with the running time and the document categories recorded. Test sets one and two will then be processed again using both the text classification and language identification searches with the running time, the document categories, and the identified language recorded. The tests will be repeated ten times to account for any variability in processing times for a total of forty experimental runs for statistical processing.

## **4.4 Analysis of Experiments**

A grand total of eighty experiments will be run: forty each for ad hoc and statistical approaches. For all of the experiments, the processing time and the text classification results were recorded. Half of the experiments also have the language identified. All of this data will be examined to evaluate effectiveness of including language identification with text classification.

### **4.4.1 Analysis of Processing Times**

The processing times for the set one and set two test data ought to be the same. For each of the four sets of data a paired *t*-test will be used to validate that the only difference in these times is due to random factors. Next, the times between runs with and without language

identification will be compared to see how much additional processing was required for language identification. A *t*-test will be used to determine if the additional processing time is significant.

#### **4.4.2 Analysis of Text Classification**

While the the robustness of the text classification technique was not a variable in this experiment, it is necessary to ensure that the text classification results were not altered by the addition of the language identification process. Any discrepancy between the text classification with and without the language identification component represents an unacceptable condition.

#### **4.4.3 Analysis of Language Identification**

The language results would be examined to see which documents were misidentified. Each of these documents will be examined to see if they are anomalies which should be discarded or genuine misidentifications. Anomalies may be removed from the test sets and the experiments rerun.

For both ad hoc and statistical language identification methods, an overall percent correct, and percent correct by language will be calculated. For both methods, a confusion matrix will be produced, showing with which languages the misidentified documents were confused.

## **5 Results**

Nothing yet, but watch this space. Paper number one would be an evaluation of the language files on the ECIMC1 CDROM using Ingle's technique.

## 6 Related Work

Most of the papers on text classification or language identification talk about one or the other but not both. Of those that discuss both, they are described as separate processes.

Cavnar and Trenkle[1] discuss using  $n$ -grams for classifying one group of USENET documents and then language identification on a different set of documents. Marc Damashek[2] clustered  $n$ -grams for languages and wire service articles but with different scoring algorithms.

## 7 Future Work

Latent Semantic Analysis provides a powerful concept based search mechanism but at an expensive processing price. How much of that processing can be reused for language identification?

## 8 Conclusions

Multiple methods are available for performing Language Identification. Most of these methods share algorithms and data structures with Text Classification methods. Carefully pairing the techniques will allow obtaining language identification at the same time as text classification with little additional processing. Systems that process large numbers of documents or have strict processing time requirements will benefit most by combining the two activities.

## References

- [1] Willian B. Cavnar and John M. Trenkle. N-gram-based text categorization. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, pages 161–169, 1994.
- [2] Marc Damashek. Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267(5199):843–848, 1995.
- [3] Gregory Grefenstette. Comparing two language identification schemes. In *3rd International Conference on Statistical Analysis of Textual Data*, pages 130–137, 1995.
- [4] Normal C. Ingle. A language identification table. *The Incorporated Linguist*, 15(4):98–101, 1976.
- [5] Laura A. Mather. A linear algebra approach to language identification. In *PODDP '98: Proceedings of the 4th International Workshop on Principles of Digital Document Processing*, pages 92–103. Springer-Verlag, 1998.
- [6] Julian A. Yochum. A high-speed text scanning algorithm utilizing least frequent tri-graphs. In *IEEE International Symposium On New Directions In Computing*, 1985.
- [7] Julian A. Yochum. Research in automatic profile creation and relevance ranking with LMDS. In *Overview of the Third Text Retrieval Conference (TREC-3)*. NIST Special Publication 500-225, 1995.