Applying Monte Carlo Techniques to Language Identification

Arjen Poutsma

SmartHaven, Amsterdam

Abstract

Two major stages stages in language identification systems can be identified: the language modeling stage, where the distinctive features of languages are determined and stored in models, and the classification stage, in which the model of the (partial) input document is compared to the reference language models. The language model most similar to the input document represents the language of the document. We describe the best-known modeling and classification techniques known in literature, and identify one disadvantage in them: the need to create a model of the entire document, even though the language can be identified with a small number of features. To avoid this, we introduce a new language identification technique that is based on Monte Carlo sampling. We show that, by determining the language of a large enough number of random features, we can determine the document language to be the language which result most often from these features. Whether the amount of samples is sufficiently large can be determined by calculating the standard error of the samples. Finally, we discuss some pilot experiments where we compare this new technique with others.

1 Introduction

In multi-lingual environments, the need for automatic language identification often arises. Identifying the language of a piece of text is a prerequisite for subsequent processing, such as indexing, categorization, and keyword extraction. For instance, morphologically-based stemming, which is highly language dependent, has proved important in improving information retrieval. Likewise, any system that filters out stopwords must identify the language to pick the correct stopwords list.

Language identification systems have been reported to have nearly perfect performance (Cavnar and Trenkle 1994, Dunning 1994, Grefenstette 1995, Sibun and Reynar 1996). As such, it is one of the most successful types of text classification (van Rijsbergen 1979).

This paper consists of five sections. In the next section, we identify two major stages in language identifiers: the modeling stage, where the distinctive features of a language or a document are determined, and the classification stage, where the document's features are compared to those of all available languages. We describe all major modeling and classification techniques, with their respective (dis)advantages. To resolve one specific disadvantage, we introduce a new classification technique, based on Monte Carlo sampling, in section 3. In the following section, we describe and discuss the experiments we conducted with the various modeling and classification techniques, and finally, in section 5, we come to the conclusion of this paper.

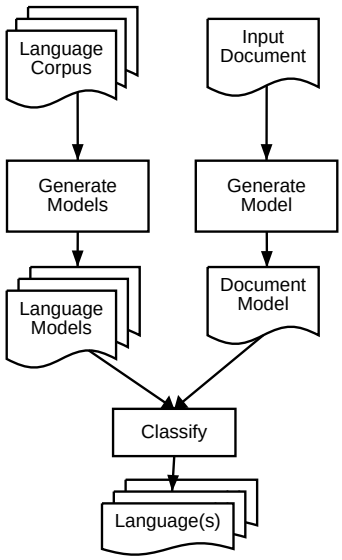

Figure 1: The major stages of language identification systems.

2 Previous Work

Language identifiers consist of two major stages. These stages are graphically depicted in figure 1.

At the top of this figure, we see the *modeling stage*. During this stage, the language-specific features of a text are learned and stored in a model. First, as can be seen on the upper left-hand side in this figure, the distinctive features for each language in a multi-lingual corpus are determined and stored in a *language model*. Later, seen on the upper right-hand side, the features of a specific text are determined and stored in a *document model*. The exact modeling method depends on the modeling technique used (see below).

At the bottom of this figure, the *classification stage* is shown. During this stage, the document model is compared to the language models. The language model which is most similar to the document model is then selected, and represents the language of the document. The actual comparison method depends on the classification technique used (see section 2.2).

We will discuss the techniques used at these stages in separate sections.

2.1 Modeling techniques

In the modeling stage, language models are created from a corpus of training documents, and document models are created from an input document. The two main modeling techniques focus on common words and character N-grams respectively.

Dan	Dut	Eng	Fre	Ger	Ita	Nor	Por	Spa	Swe
i	de	the	de	der	di	.	de	de	och
af	van	and	la	die	e	og	a	la	i
og	het	to	le	und	il	det	que	que	att
at	een	of		den	che	,	o	el	som
§	en	a	et	in	la	han	e	en	en
til	in	in	des	von	a	i	do	y	r
for	dat	was	les	.	in	er	da	a	p
en	is	his	du	zu	per	”	no	los	det
om	te	that	”	dem	del	p	um	del	av
der	op	I	en	,	un	til	em	se	fr
er	voor	he	un	fr		at	para	por	med
U	met	as	que	mit	non	som	com	las	den
ikke	die	had	a	das	i	var	se	con	till
eller	De	with	qui	des	si	jeg		un	har
som	zijn	it	dans	ist	le	med	os	para	de

Table 1: Most frequent common words per language.

We will discuss these techniques separately.

2.1.1 Common Words Technique

Common words such as determiners, conjunctions and prepositions seem good clues for guessing a language (Johnson 1993). From a corpus of documents in a certain language, the most frequent words are determined and placed in a model for this language. Each word has a significance score based on the frequency of that particular word in the corpus. By dividing this frequency by the total frequency of all words, we give each word a probability, and we can see the language model as a probability distribution.

Table 1 shows the most frequent common words for 10 European languages, as obtained in experiments described by Grefenstette (1995).

The disadvantage of this technique is that, though common words occur enough in larger texts, they might not occur in a shorter input text.

2.1.2 N-Gram Technique

The second modeling technique is based on character N-grams (Cavnar and Trenkle 1994, Grefenstette 1995).[1] Similarly to the common words technique, this technique assembles a language model from a corpus of documents in a particular language; the difference being that the model consists of character N-grams instead of complete words. Likewise, each N-gram has a frequency score.

[1] An N-gram is an n-character slice of a longer string.

Dan	Dut	Eng	Fre	Ger	Ita	Nor	Por	Spa	Swe
er␣	en␣	␣th	␣de	en␣	␣di	et␣	␣de	␣de	en␣
en␣	de␣	he␣	es␣	er␣	to␣	␣.␣	de␣	de␣	␣.␣
for	␣de	the	de␣	␣de	␣de	en␣	os␣	os␣	er␣
et␣	et␣	␣,␣	ent	der	di␣	er␣	do␣	␣la	et␣
ing	an␣	nd␣	nt␣	ie␣	␣co	␣de	que	el␣	tt␣
␣fo	n␣d	ed␣	␣le	ich	la␣	␣ha	␣qu	la␣	␣de
␣af	␣he	␣an	e␣d	sch	re␣	an␣	␣co	que	ar␣
␣de	er␣	and	le␣	ein	ion	de␣	as␣	as␣	␣,␣
nde	␣va	␣.␣	ion	che	ent	␣,␣	ent	ue␣	fr␣
els	van	␣to	s␣d	die	e␣d	det	␣o␣	␣qu	om␣
lse	een	ing	e␣l	ch␣	le␣	ar␣	ue␣	␣co	␣oc
ret	ver	to␣	␣la	den	o␣d	␣og	␣a␣	␣en	ch␣
␣sa	aar	ng␣	la␣	nd␣	ne␣	og␣	o␣d	en␣	␣de
der	␣ee	er␣	re␣	␣di	no␣	te␣	␣se	ent	och
␣i␣	het	␣of	on␣	ung	␣in	han	␣o␣	es␣	an␣

Table 2: Most frequent trigrams per language.

To make the distinction between inner-word N-grams and N-grams at the beginning and end of each word, the N-grams are padded with blanks (␣). For instance, the word *TEXT* would result in the following N-grams:

bi-grams: ␣T, TE, EX, XT, T␣
tri-grams: ␣TE, TEX, EXT, XT␣
quad-grams: ␣TEX, TEXT, EXT␣

As can be seen from this list, the disadvantage that holds for the common words technique does not hold for the N-gram technique, since even a single word contains multiple N-grams.[2] Additionally, character N-grams have proved to perform very well when dealing with noisy input (Suen 1979).

Table 2 shows the most frequent trigrams for five European languages, as obtained in experiments described by Grefenstette (1995).

The disadvantage of the N-gram modeling technique is that N-grams are not very distinctive. In table 2, for instance, multiple trigrams occur in more than one column. Common words, on the other hand, are much more distinctive, as can be seen in table 1.

2.2 Classification techniques

When language and document models have been generated, the document model is compared with the reference language models. The result of this stage is one or more languages; this also serves as the output of the language identifier as a whole.

[2] In general, a padded string of length k contains $k - N + 3$ N-grams.

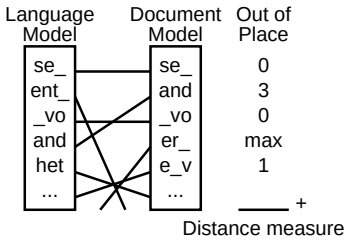

Figure 2: Calculating the out-of-place measurement between two profiles.

2.2.1 Rank order statistics

This technique, as described by Cavnar and Trenkle (1994), determines how far out of place an N-gram is in the language models from its place in a document model. Since this technique operates on the relative indices of features (words or N-grams), these need to be sorted in order of frequency.

How this technique is applied to N-gram features is shown in figure 2. For each N-gram in a document model, its counterpart in a language model is located, and then calculated how far out of place it is. If an N-gram is not in the language model, it takes a maximum out-of-place value, which is equal to the amount of N-grams in the model. The sum of all of the out-of-place values for all N-grams is the distance measure for the document from the language. The language model with the smallest distance from the document represents the language of the document.

2.2.2 Mutual Information statistics

Sibun and Reynar (1996) report the use of a well-known information theoretic measure for language identification: mutual information statistics.[3] The mutual information between two probability distributions reflects the amount of additional information necessary to encode the second distribution using an optimal code generated for the first distribution. It is a useful measure of similarity between probability distributions. Mutual information ranges from 0 to ∞, with the minimum generated when the two distributions are identical. The equation for the mutual information statistics is:

$$MI(p||q) = \sum_{x \in \chi} p(x) \log \frac{p(x)}{q(x)}$$

Applying this technique to language identification is straightforward: we simply use the above equation to calculate the distance between the language model generated for a document with the model generated for each language, and pick the language with the lowest distance. Thus, we can rewrite the previous equation

[3] Also known as relative entropy, or Kullback Leibler distance.

as follows, where L is a language and D is a document.

$$MI(D||L) = \sum_{f \in D} P(f|D) \log \frac{P(f|D)}{P(f|L)}$$

The language model with the lowest mutual information statistic constitutes the language of the document.

3 Monte Carlo Classification

In the previous section, we have presented the standard techniques for modeling and classifying input within a language identification system. In this section, we will present a new classification method, which is based on Monte Carlo sampling.

We will first discuss the motivation for this technique. Next, we give some background statistics, and finally, we present the Monte Carlo technique.

3.1 Motivation

A major drawback of the classification techniques described in section 2.2 is the necessity to create a complete document model. In order to produce this model, we need to determine the features for the entire document. This seems a waste of effort, since—as Grefenstette (1995) shows— the language of a document can be determined with 93% accuracy with as little data as the trigrams of three words.

This issue can be resolved by taking the features of a large enough subset of the document (Grefenstette 1995, Dunning 1994); or by limiting the amount of N-grams (Cavnar and Trenkle 1994). However, by taking such a subset, there is a risk that the subset of the document model does not contain enough language-characteristic features.

Thus, instead of creating a complete, static model of (a fixed-size subsection of) the document, we create a dynamic model of the document: we increase it's size until it is sufficiently characteristic.

3.2 Method

This subsection describes the statistical method used in the Monte Carlo language identification technique. Basically, any statistics-based language identifier seeks the most probable language given a certain document. In other words, it seeks to maximize $P(L|D)$, where L is a language, and D is a document. Using Bayes' law, we rewrite this as equation 1, below. Since the denominator is the same for all languages, and since all languages are equally probable, we need only maximize $P(D|L)$. Thus, we can rewrite 1 as 2:

$$\begin{aligned} \max P(L|D) &= \max \frac{P(L) \cdot P(D|L)}{P(D)} && (1) \\ &\approx \max P(D|L) && (2) \end{aligned}$$

Thus, by calculating the maximum probability of a document given a language, we calculate the most probable language given that document. Since both document and languages are represented by a model, and since models are probability distributions for features (see 2.1.1), this is equal to:

$$\max P(L|D) = \max \sum_{f \in D} P(f|L)$$

So, by iteratively determining the language from a large number of random features from a document, we can determine the language of this document to be the language which results most often from these random features. The most probable language can be estimated as accurately as desired by making the number of samples sufficiently large. According to the Law of Large Numbers, the language that was selected most often converges to the most probable language. Methods that estimate the probability of an event by taking random samples are known as Monte Carlo methods (Meyer 1956).

We can determine whether the amount of samples is sufficiently large by calculating the standard error σ of the samples. The standard error is defined as the squared root of the variance of the underlying probability distribution, divided by the number of samples:

$$\sigma = \sqrt{\frac{\text{Var}(X)}{N}}$$

Since, language identification uses a Bernoulli distribution[4], the variance is defined as:

$$\text{Var}(L|D) = P(L|D)(1 - P(L|D))$$

Thus, the standard error for language identification is:

$$\sigma = \sqrt{\frac{P(L|D) \cdot (1 - P(L|D))}{N}}$$

Standard values for standard errors include 0.01 or 0.05.

We can expect N-grams to be equally distributed across the entire document, meaning that every N-gram is as likely to appear at the start of the document, as it is to appear in the middle or at the end. Thus, *we can simply sample N-grams from the beginning of the document.*

4 Experiments

To test the performance of the new classification technique, and to compare it with others, we conducted several experiments. In these experiments, we compared all combinations of modeling and classification techniques. In other words, we compared the following language identifiers:

[4] Meaning that—when identifying the language of a given document—there are two possible outcomes: the document is written in that language or not.

- Rank order statistics identifier with common words,
- Rank order statistics identifier with N-grams,
- Mutual information statistics identifier with common words,
- Mutual information statistics identifier with N-grams,
- Monte Carlo identifier with common words,
- Monte Carlo identifier with N-grams.

4.1 Test Environment

The performance of the language identifications method was evaluated using the corpus available on the ECI CD-ROM.[5] This CD-ROM contains text in various languages; we only used the first million characters of text in each of the following languages: Danish, Dutch, English, French, German, Italian, Norwegian, Portuguese, Spanish, and Swedish.

We used a blind testing method, dividing the corpus into a 90% training set, and a 10% test set. We carried out ten experiments, each using a different split of training and test set. Depending on the modeling technique being tested, the training set was converted into words or N-grams, enriched with their model frequencies. We filtered out all but 400 of the most frequent features. The test set served as input that was classified using the language models generated from the training set. In each of the ten experiments, we varied the amount of input data from 10 characters to 500 characters. The output from the language identifier was compared with the language of the test input, and the performance of each method was calculated as being the number of correct identifications divided by the total amount of identifications.

Additionally, we measured the elapsed time (in milliseconds) it took to run the ten experiments. These measurements were obtained while doing the experiments on a 700 MHz Pentium III computer, running the Linux operating system. The whole experiment setting was implemented in the Java programming language.

4.2 Results

The results we obtained for the performance scores are displayed in figure 3. On the horizontal axis, the amount of input is specified; on the vertical axis, the performance (i.e. the number of correct identifications divided by the total amount of identifications). There are a few interesting things to note in this figure. First, the Rank Order classification method has the best overall performance; the Monte Carlo method scores slightly less. The Mutual Information method scores worst, especially when combined with the common words modeling technique. Since this technique is heavily dependent on a probability distribution, and because such

[5] For more information, see `http://www.elsnet.org/eci.html`.

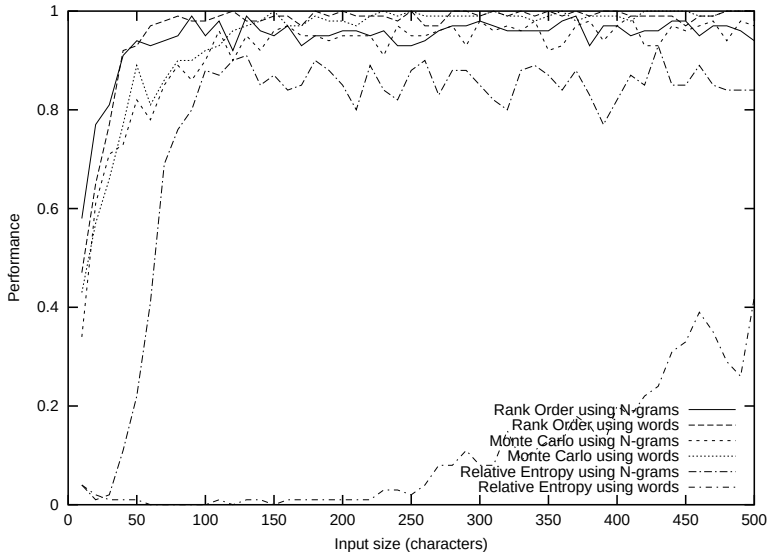

Figure 3: Performance score for six Language Identification methods.

a distribution can only be generated with larger amounts of data, this behavior can be expected.

Secondly, both with the Rank Order and Monte Carlo techniques, the N-gram modeling method performs best with small amounts of input (i.e. less than 100 characters of input). This can be expected, since any text contains more N-grams than words, and thus it is more likely that language discriminating features are present. However, the Common Words technique performs slightly better with larger amounts of input (i.e. more than 350 characters of input). Since common words are more distinctive than N-grams, this type of behavior can also be expected.

Figure 4 shows the amount of time necessary to complete the 10 experiments, on a logarithmic scale. The most important item in this figure is the large amount of time required for the Rank Order method, especially compared to the other two classification method. As a comparison: when using N-grams as features and with an input of 200 characters, the Rank Order method took approximately 2 minutes per 10 experiments, as compared 1.4 seconds with the Monte Carlo method or 800 milliseconds with the Mutual Information method.

Another thing that can be seen in this figure is the fact that the N-gram modeling technique takes longer than the common words technique. Since the creation of N-grams is computationally more expensive than the creation of words, this can be expected.

Generally, we can say that the N-gram modeling method is somewhat slower, but performs best with little input, and that the Rank Order classification technique

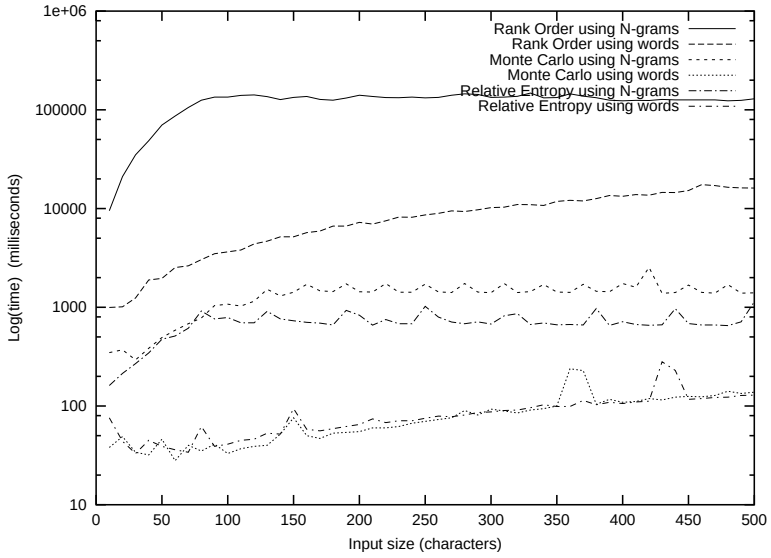

Figure 4: Time required for three Language Identification methods.

scores best, but is very slow. The Monte Carlo sampling method scores slightly less, and is much faster. The Mutual Information technique scores worst, especially when used with common words.

5 Conclusion

We have identified two major stages in language identification: the modeling stage and the classification stage. In the modeling stage, the most frequent words or character N-grams are identified and stored in a model. In the classification stage, the model of the input document is compared to the reference language models: the most similar language model represents the language of the document. We discussed the best-known modeling and classification techniques.

We introduced a novel classification method, which is based on Monte Carlo sampling. The advantage of this new technique is that it samples features from the document, and stops sampling when the language has been identified. We performed several experiments, comparing the new technique to those known in literature, and show that the Monte Carlo technique performs somewhat less than the best-performing classification technique, but is much faster.

References

Cavnar, W. B. and Trenkle, J. M.(1994), N-Gram-Based Text Categorization, *Proceedings of the Third Annual Conference on Document Analysis and Infor-*

mation Retrieval (SDAIR), Las Vegas, pp. 161–175.

Dunning, T.(1994), Statistical identification of language, *Technical Report MCCS 94–273*, New Mexico State University.

Grefenstette, G.(1995), Comparing two Language Identification Schemes, *JADT 1995, 3rd International conference on Statistical Analysis of Textual Data*, Rome.

Johnson, S.(1993), Solving the problem of language recognition, *Technical report*, School of Computer Studies, University of Leeds.

Meyer, H. (ed.)(1956), *Symposium on Monte Carlo Methods*, Wiley, New York.

Sibun, P. and Reynar, J.(1996), Language identification: Examining the issues, *Proceedings of the Fifth Annual Conference on Document Analysis and Information Retrieval (SDAIR)*, Las Vegas, pp. 125–135.

Suen, C. Y.(1979), N-Gram Statistics for Natural Language Understanding and Text Processing, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1**(2), 164–172.

van Rijsbergen, C. J.(1979), *Information Retrieval*, 2nd edn, Butterworths, London.