Comparing methods for language identification

Muntsa Padró and Lluís Padró
TALP Research Center
Universitat Politècnica de Catalunya
Jordi Girona Salgado 1-3
08034 Barcelona, Spain
{mpadro,padro}@lsi.upc.es

Resumen: En este artículo se comparan tres sistemas estadísticos de identificación de idioma. Se presenta también un estudio detallado de la influencia de algunos factores importantes sobre la precisión de los sistemas. Estos factores son: la medida del corpus de entrenamiento, la cantidad de texto que se quiere clasificar y las lenguas entre las cuales el sistema es capaz de distinguir (se estudiará tanto el número de lenguas cómo cuáles son esas lenguas).
Palabras clave: identificación de idioma, sistemas estadísticos, multilinguismo, modelos de Markov visibles, vectores de frecuencias de trigramas, categorización de textos basada en n-gramas

Abstract: In this work three different statistical language identification methods are compared, and a detailed study of the influence on those systems of some basic parameters is performed. The analyzed parameters are the size of the train set, the amount of text that we want to classify and the languages the system is able to distinguish (it will be studied not only the influence of the number of languages but also the influence of wich are the considered languages).
Keywords: language identification, statistical systems, multilinguality, Visible Markov Models, Trigram Fequency Vectors, n-gram Based Text Categorization

1 Introduction

Language identification is one of the most basic steps to be taken in many systems that involve NLP. Tasks as Sumarization, Question Answering, Translation, etc. need to know the language of a given text in order to process it. Nowadays, with the increasing use of Internet, it is becoming more usual to have the texts to be processed written in different languages. So despite of the fact that language identification is an easy and very studied task, it could be still necessary to study the differences between some systems and the influence of some factors in the system precision. This is even more crucial in bilingual or multilingual societies in which NLP related applications (news/information providers, Q&A, IR, etc.) may want to offer their services to each customer in a different language.

There are many approaches to this task, most of them using some linguistic information such as diacritics and special characters (Newman, 1987), characteristic letter sequences (Dunning, 1994), etc. There are also statistical methods to perform language identification. Most approaches use low order n-gram models, such as those described in (Hayes, 1993) and (Churcher et al., 1994), and the systems studied in this work. Other statistical systems determine the most common words of a language (Johnson, 1993), though this may easily fail when the amount of text to classify is not big enough.

The goal of this paper is to make a comparison of three statistical systems for language identification and to study the behaviour of these systems under different conditions. The systems will be trained and tested with different amounts of text and the influence of the number (and which) languages the system can distinguish will be studied.

In section 2 the used methods are introduced. Section 3 describes the performed experiments and the obtained results. Section 4 states some conclusions and further work.

2 Compared Methods

We studied three statistical methods for language identification. All of them are trained and tested with the same data. These methods are based on: Markov Models, comparison of Trigram Frequency Vectors, and n-gram text categorisation. Now we present a brief introduction to these

three methods.

Figure 1 shows the general architecture of the three systems. All of them are statistical methods and work with a predetermined set of languages. When we train the system for one language, it stores the information in a certain way. Each system will have a statistical modelization of each language it has been trained for. When a text has to be classified, the system compares the unknown text with each of the language models, computes some kind of distance or similarity measure, and chooses the closest language as the correct one. The three presented systems differ on how a language is modelled and in the used similarity measure.

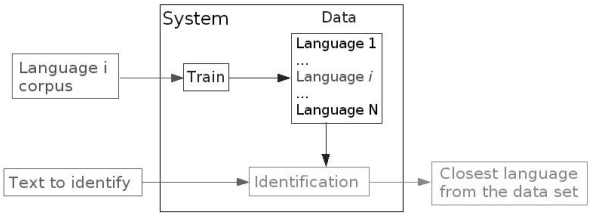

Figure 1: General architecture of the systems

2.1 Markov Models

Hidden Markov Models (HMM) are commonly used in spoken language identification (Zissman and Singer, 1994; Lamel and Gauvain, 1994) but they are also used for written language (Xafopoulos et al., 2004; Ueda and Nakagawa, 1990). Nevertheless, this task can be performed with visible Markov Models (MM) which is the first of the three systems compared in this work.

For each language that the system must know about, a model is trained from a text corpora, and stored for later comparison with unidentified text. In these models each state s_i represents a character trigram $c_{1i}c_{2i}c_{3i}$. Thus, the parameters of the MM are the transition probability and the initial probability:

$a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$: transition probability from the state i to the state j. As from one state $c_1c_2c_3$ it is only possible to go to another state $c_2c_3c_4$ the probability transition is, in fact, the 4-gram probability $P(c_4|c_1c_2c_3)$.

$\pi_i = P(q_1 = s_i)$: probability of starting a sequence in state i.

These probabilities are estimated via MLE, i.e. computing the relative frequency of each transition or initial state in the training data:

$$a_{ij} = \frac{\#(s_i \rightarrow s_j)}{\#s_i}$$

$$\pi_i = \frac{\#s_i(t=0)}{\#sentences}$$

Very simple smoothing is performed, counting a fixed small number of occurrences for trigrams not appearing in the training set.

To classify a new text, the system computes the sequence probability using each language model it has been trained for.

$$P(q_1, ..., q_T) = \pi_{q_1} \prod_{t=1}^{T-1} a_{q_t q_{t+1}}$$

Then the system chooses the language that gives the largest probability.

2.2 Trigram Frequency Vectors

The trigram frequency vectors technique (Damashek, 1995) consists in comparing a vector of trigram frequencies for the text to classify with the vectors of known language, and select the closest one.

A trigram t_i is formed by three consecutive characters of the text we are analysing. Then $t_i = c_{1i}c_{2i}c_{3i}$.

A vector of trigram frequency is a vector in a N-dimensional space, where N is the number of possible trigrams,

$$\vec{v} = (v_1, v_2, ..., v_N)$$

and v_i is the occurrence frequency of the trigram t_i. To train the system we compute the relative frequency of each trigram that occurs in the train set for a determined language. With these frequencies we build the vector for this language ($\vec{l^j}$).

When we want to determine the language of a piece of text we build the vector for this text ($\vec{w}$), computing the occurrence frequencies of each trigram in the text, and then we compare this vector with the vectors of each language ($\vec{l^j}$). This comparison is made computing the normalised dot product of the two vectors:

$$\vec{w} \cdot \vec{l^j} = \frac{\sum_{i=1}^{N} w_i l_i^j}{|\vec{w}||\vec{l^j}|}$$

This product is a factor that gives an idea of how similar are the two vectors. The closest to 1 is this factor, the more similar are the vectors. So we choose the language that gives the maximum dot product, what means that this language is the most similar to the unclassified text.

2.3 n-Gram Based Text Categorisation

This technique is a text categorisation method, presented in (Cavnar and Trenkle, 1994), that can be applied to language identification, where each category is a language. The implementation of this technique is named TextCat and it is available in the web[1].

The system is based on comparing n-gram frequency profiles. A n-gram frequency profile is a list of the occurring n-grams sorted in decreasing frequency order. For each language we want to train the system, we create its n-gram profile using all the n-grams for all values of n from 1 to 5.

When we want to classify a piece of text we build the n-gram frequency profile for this text and compare it with each language profile we have computed when training the system. This comparison is made computing a distance measure between the profiles, which consists in counting how different is the position $rank(t_i, text)$ of n-gram t_i in the unclassified text profile with respect to the position $rank(t_i, l_j)$ of the same n-gram in the language j profile . The distance between the two profiles is computed summing all the distances for each trigram.

$$D_j = \sum_{i=1}^{N} |rank(t_i, text) - rank(t_i, l_j)|$$

where N is the number of trigrams.

The system computes the distance from the profile of the unclassified text to each profile of the known languages and chooses the language that gives the smallest distance.

3 Experiments and Results

The performed experiments are focused on studying the influence of the training set size, the amount of text to classify and the number of languages among which the system can choose, in order to determine the influence they have on the system performance, with a special interest in the application of language identification systems to multilingual NLP applications.

Corpora for six different languages have been used in the experiments. Those languages are Catalan, Spanish, English, Italian, German and Dutch. The corpora are formed by a set of daily newspaper news. For each of these corpus a random partition containing about 30,000 words was selected to be used as the test set. The rest of the corpus was used to randomly extract training samples of different sizes.

The performed experiments involve training each system for all languages using a train set ranging from 2,500 to 250,000 words, and evaluate their performance over the test data. The test is done giving the system an amount of unclassified text ranging from 5 to 1000 characters. The process is repeated for all possible combinations of languages, from two to six languages.

3.1 Influence of the training size

First of all we present the influence of the training set size. Here we show the evolution of the precision when systems are trained with different sized corpus. Figures 2 and 3 show the average precision when the systems are distinguishing six and two languages respectively. The intermediate cases have a similar behaviour.

In order to see the general behaviour of the systems, we have extracted 10,518 complete sentences in the 6 languages. The length of these sentences ranges from 2 to 310 characters. Here it is represented the average of each system precisions when identifying the language of these sentences.

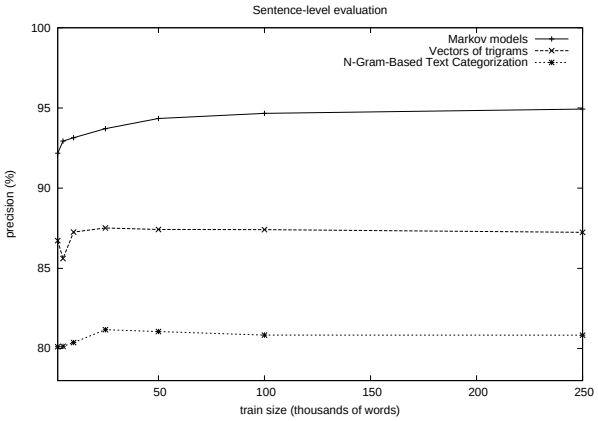

Figure 2: Precision of the systems distinguishing 6 languages

We can see that when the systems are trained with more than 50 kwords the precision does not rise significantly. So the training size of the corpus is only a significant factor for very small amount of training data. In order to study the influence of the other factors more clearly, the corpus train size is fixed to the maximum size (250 kwords) to minimise the impact of this parameter.

[1] http://odur.let.rug.nl/~vannoord/TextCat/

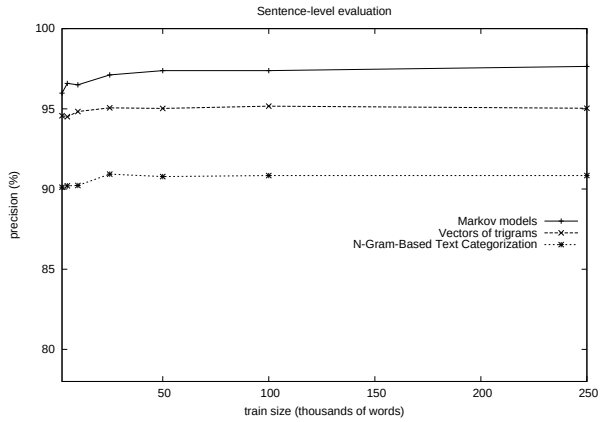

Figure 3: Precision of the systems distinguishing 2 languages (average)

3.2 Influence of the test size and the number of distinguished languages

More important factors for the precision of the system are the amount of text the system has to classify and the number of distinguished languages. In Figure 4 we show the results for the Markov Model system when modifying the set of languages the system can identify or the size of the unclassified text.

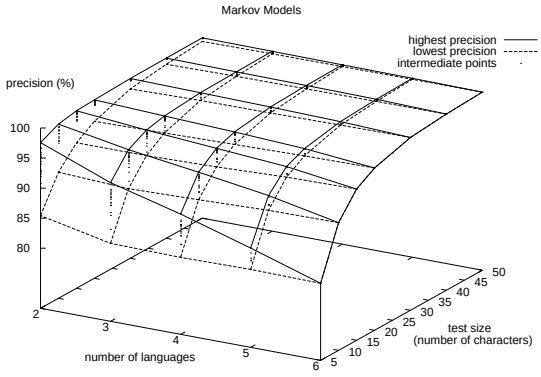

Figure 4: Precision of the system that uses Markov Models trained with 250 kwords

It can be seen how the performance of the system is much lower for small data samples (5-20 characters), but it quickly raises to almost 100% when the text to classify is larger than 20 characters.

It is also remarkable that, although the average precision is higher when distinguishing only two languages, this case presents a variability of 1.5% (or more, for small input texts) between the best and worst pair of languages. This indicates that the difficulty of the task greatly varies depending on how similar are the involved languages.

3.3 More about the influence of the distinguished languages

As discussed above, the number of languages among which the system can choose and which are those languages are factors with a large influence on the precision of the system.

It may be expected that when distinguishing languages belonging to the same philogenetic family (e.g. romance languages, Germanic languages) the precision of the systems will be lower. If these languages also share sociocultural environment, the task will be even harder. For instance, Catalan has a larger lexical similarity with Italian (87%) than with Spanish (85%)[2], but a large *content* similarity with the latter: the same topics, names, locations, etc. will tend to appear in texts, largely contributing to confuse statistical systems such as those evaluated here.

In figure 5 the precision for different combinations of two languages is plotted. In order to ease the study of the figure, the legend is ordered from highest to lowest precision.

It is clear that when distinguishing similar languages the precision falls with respect to the precision when the system has to discern among two very different languages. On the one hand the most difficult languages to distinguish are Catalan and Spanish, followed by Catalan and Italian, Dutch and German, etc. On the other hand the system achieves a high precision when distinguishing any romance language from German, or from Dutch, that are more different languages.

Studying the distribution of the errors provides evidence consistent with these facts: When classifying text with all 6 language as possible choices, the system tends to confuse the more similar languages.

Table 1 shows this error distribution for the Markov Model system. For each correct language there are represented (in %) the frequency of mistaking this language with each of the other five languages.

In this table it can be observed that the most frequent errors are confusing Catalan and Spanish. Dutch, German and English are often confused too, but not as frequently as the first two. As in the two languages case this may be explained by the similarity of those groups of languages.

In fact, it can be seen that while the 6 possible confusions among romance languages (Catalan, Spanish and Italian) represent 51.8% of the er-

[2] According to http://www.ethnologue.com

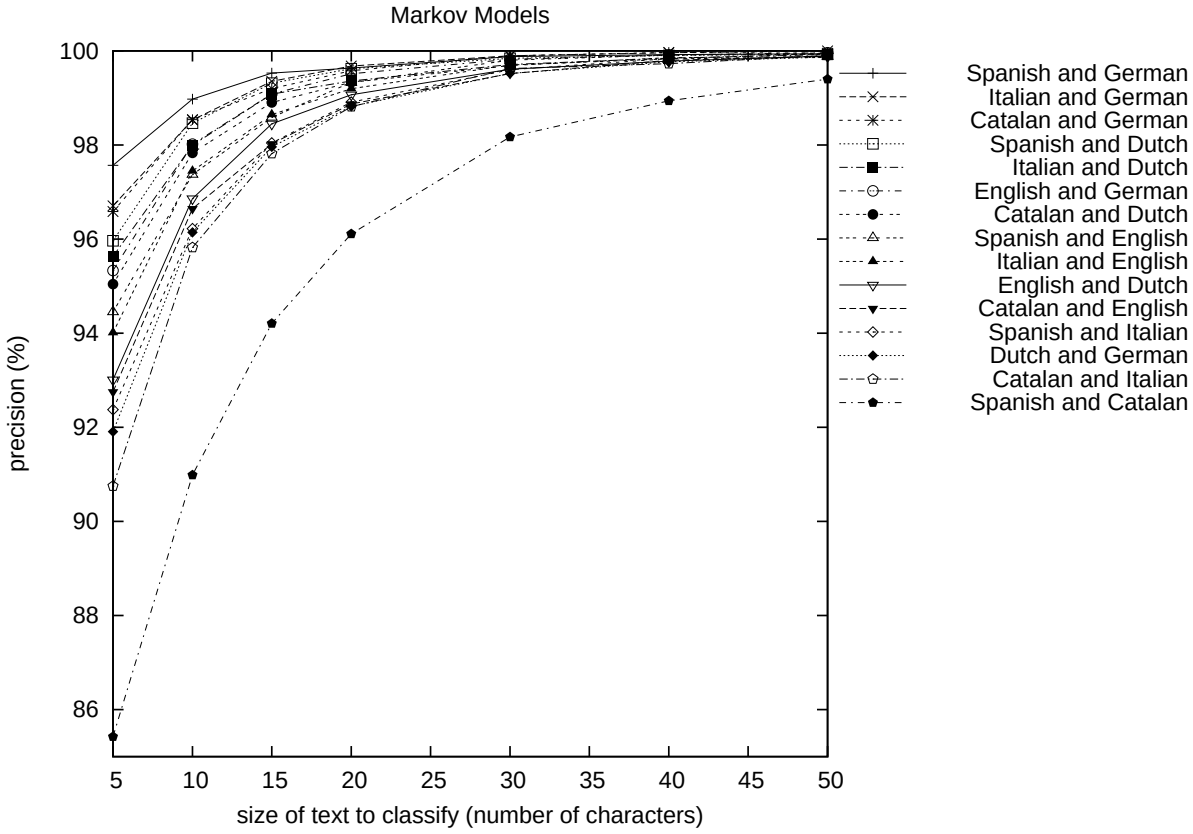

Figure 5: Precision for different combinations of two languages

Correct Language	Language determined by the system						Total
	Catalan	Spanish	Italian	English	German	Dutch	
Catalan		**14.2**	1.7	4.5	0.2	0.9	21.5
Spanish	**25.1**		3.8	3.8	0.5	1.4	34.6
Italian	3.6	3.4		1.8	0.7	2.5	12.0
English	2.9	2.3	1.6		0.8	4.7	12.3
German	0.9	0.2	0.5	1.8		6.8	10.2
Dutch		0.2	1.6	3.8	3.8		9.4
Total	32.5	20.3	9.2	15.7	6.0	16.3	100

Table 1: Error distribution for the MM system when classifying texts of 30 characters.

rors, the 6 possible confusions among Germanic languages (English, German and Dutch) produce the 21.7% of them. The other 18 possible confusions between groups give the rest of them (26.5%).

3.4 Comparison of the three methods

Figure 6 presents a comparison of the obtained precision with the three used methods when identifying different amounts of text. There are represented the results when distinguishing among two and six languages. For the case of two languages, the average precision is plotted.

In order to show the evolution of the precision for small texts, the figure presents only the precision when recognising texts from 5 to 150 characters. For larger input texts, precisions are very high for all systems, though TextCat is somewhat behind the others. Nevertheless, according to its authors (Cavnar and Trenkle, 1994) TextCat is highly tolerant to textual errors so it is possible that this system would perform better than the others when dealing with noisy texts. This is beyond the scope of this paper, but a further work to be performed is to study and compare the behaviour of all systems under such situation.

This figure shows clearly that the precision largely improves with longer input texts. In addition, differences between systems for small input texts are very large. The system that uses

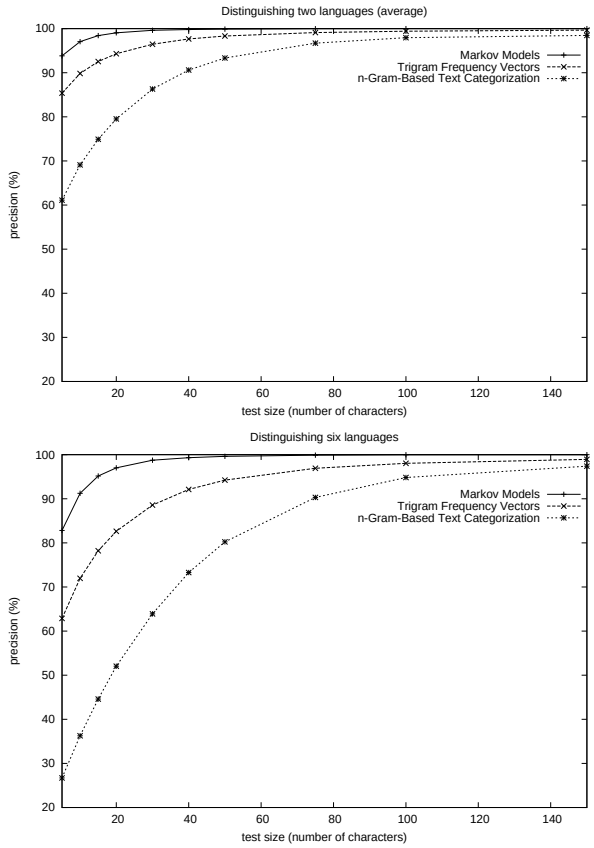

Figure 6: Precision of the three systems

Markov Models achieves always the highest precision, with a very significant difference when classifying small texts. Furthermore, this system is the fastest, performing the classification about 4 times faster than the other two.

When using n-Gram-Based Text Categorisation, the precision is very low for the smallest texts but rises as the amount of text to classify becomes larger, becoming comparable to the precision of the other systems.

Another important observation is the difference of the obtained precision when distinguishing two or six languages: As it can be expected, when the system has to choose among six languages the precision falls, especially for small texts, while when distinguishing among two languages the average precision is higher.

4 Conclusions and Further Work

The influence of some important factors in a language recognising system has been studied. It has been shown that the influence of the train set size is not important when this size is bigger than approximately 50 kwords.

The other studied factors have been proved to be more significant. The amount of text to classify is crucial, but it is not necessary to have very long texts to achieve a good precision. For texts over 500 characters, all the systems get a precision higher than 95% (99% if we exclude TextCat), and for texts of 5000 characters the precision is higher than 99% with all systems, reaching 100% in many cases, specially with the MM system.

Otherwise, it is important to highlight that for small texts there is a big difference (almost 60 points in the worst case) among the precisions obtained by the three systems in the same situation, being the Markov Model system the one with the highest precision, probably because its global sequence probability optimisation captures language features (length of the words, frequent words or stems) that can not be dealt with by the other systems.

In fact, the system that uses Markov Models performs better or equal than the others in all situations. This proves that it is important to take into account not only the appearing frequency of the n-grams, but also some sequence information. In addition to that, as it has been said before, this system is faster than the others, so it seems to be the best choice for a statistical language recogniser.

The influence of the number of languages the systems can identify is a very relevant factor to take into account. The more languages the system has to recognise, the less precision it will have. Furthermore, it is clearly decisive which languages we want to discern. While if the language identification system has to be applied in a multilingual environment involving similar languages the precision of the system is expected to fall, if the languages to distinguish have different origins the task will achieve a high precision.

Although the reported conclusions may seem obvious, they constitute an empirical validation of what intuition suggests about the influence of the evaluated parameters. Furthermore, the results of this study may be useful to derive some trusty indicators of the confidence for language recognisers output, and to establish the ranges and conditions where each system performs best.

Some further work that could be realized is adapting the systems to recognise multilingual texts and to detect intra–document language changes. It can be very useful when dealing with systems applied to mail or news processing where there are usually insertions of languages different from the main one (replying a mail in another language, quoting someone...). This phenomenon happens very often with languages that share the same social environment, specially in

multilingual regions.

Furthermore, it could be interesting to study the behaviour of the systems when the text to classify comes from a noisy source and contains some contextual errors. This errors could arise from an OCR system or from the unsupervised typewriting of an e-mail, among others.

In addition, the presented experiments have been performed under a closed–world assumption (i.e. all possible languages for input text are known to the system). This may be enough for applications restricted to a bilingual or multilingual environment, but when moving to an unrestricted domain (e.g. Internet) the possibility that the input text is written in a language unknown to the system should be considered.

Finally, since the tested systems tend to fail when distinguishing similar languages (e.g. Spanish and Catalan), further research could be done to solve these cases, maybe in the line including in the system the ability to deal with some specific morphological features.

References

B. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, USA. UNLV Publications/Reprographics.

Gavin Churcher, Judith Hayes, Stephen Johnson, and Clive Souter. 1994. Bigraph and trigraph models for language identification and character recognition. In *Proceedings of 1994 AISB Workshop on Computational Linguistics for Speech and Handwriting Recognition*, University of Leeds, UK.

Marc Damashek. 1995. Gauging similarity with n-grams: Language independent categorization of text. *Science*, 267(5199):843–848.

Ted Dunning. 1994. Statistical identification of language. Technical report mccs 94-273, New Mexico State University.

Judith Hayes. 1993. Language recognition using two-and three-letter clusters. Technical report, school of computer studies, University of Leeds.

Stephen Johnson. 1993. Solving the problem of language recognition. Technical report, school of computer studies, University of Leeds.

L. Lamel and J. Gauvain. 1994. Language identification using phone-based acoustic likelihoods. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing [ICA94]*.

Patricia Newman. 1987. Foreign language identification: First step in the translation process. In *Proceedings of the 28th Annual Conference of the American Translators Association*, pages 509–516.

Yoshio Ueda and Seiichi Nakagawa. 1990. Prediction for phoneme/syllable/word-category and identification of language using hmm. In *Proceedings of the 1990 International Conference on Spoken Language Processing*, Kobe, Japan.

A. Xafopoulos, C. Kotropoulos, G. Almpanidis, and I. Pitas. 2004. Language identification in web documents using discrete hmms. *PR*, 37(3):583–594, March.

Marc A. Zissman and Elliot Singer. 1994. Automatic language identification of telephone speech messages using phoneme recognition and n-gram modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing [ICA94]*, Kobe, Japan.