2nd Hellenic Conf. on AI, SETN-2002, 11-12 April 2002, Thessaloniki, Greece,
Proceedings, Companion Volume, pp. 269-280

Language identification of web documents using discrete HMMs

A. Xafopoulos, C. Kotropoulos, G. Albanidis, and I. Pitas

Dept. of Informatics, Aristotle Univ. of Thessaloniki
Box 451, Thessaloniki 54006, GREECE
Fax: +30 31 998419
{alexandr,costas,galba,pitas}@zeus.csd.auth.gr

Abstract. Automatic language identification in written text documents is an issue which deserves significant attention in the context of the ever-growing volume of web documents. This paper deals with language identification in the domain of electronic texts related to tourism. The proposed system is built on Hidden Markov Models (HMMs) that enable the modeling of character sequences. For this purpose, a parallel structure of ergodic discrete HMMs is used. During testing a previously unseen document is divided into its sentences and each of them is independently characterized in terms of the language it is written in. Experiments conducted on sentence-long documents demonstrated high identification rates.

1 Introduction

It is being argued that text-based language identification (TLI) is almost a solved problem. In this paper we deal with this task on web documents, where the aforementioned argument is contradicted. There are many causes for the arousal of difficulties in web documents. First, the web documents contain extra information for the visual representation of the web page, which may interfere with the text, especially in the case of a faulty page composition. Second, the web documents may have textual information in a form that is useful when displaying the page, but disorganized when they are considered as consolidated texts. An example is data formatted as lists. Moreover, spelling and syntax errors are more frequent in document collections from the web than in corpora constructed from books or newspapers. Another issue is that web documents are not using the same encoding even if they are in the same language due to the existence of quite a few different encodings for electronic text. Although Unicode encoding would help toward this direction, the web documents that do not follow this standard are still markedly numerous. Finally, the usage of international terms or proper names, which occur in abundance in web documents, introduces additional difficulty to the recognition process.

To cope with some of the above difficulties hidden Markov models (HMMs) are used, counting on the fact that the effect of errors, as outliers, is small due to their low frequency of occurrence. HMMs have been successfully applied in

speech-based language identification [1], [2]. Our target is to test their application to text documents, and in particular, to documents that have been extracted from HTML pages, by establishing a correspondence between language characters and integer values. The latter implies the notion of the language as a signal.

Our effort focuses on the achievement of high identification rates using a small corpus of web documents for training and testing. During training HMM models for each language are created from the training part of the corpus. The computational requirements for the training are small. During testing each of the test corpus documents is split into its sentences and identification rates are measured as a result of the identification procedure on each sentence. The size of test documents used is generally small due to the fact that the test documents are only sentence-long documents.

Five main European languages are selected: English, German, French, Spanish, and Italian. When a full HTML document is provided for recognition, a slightly different procedure than the testing one is followed. After the document is split and characterized at the sentence level, wherever many contiguous sentences of the same language are found only one language tag is assigned to all of them. In this way a multilingual document can be more correctly characterized and further processed. It is worth mentioning that by the term "sentence" we do not refer to a syntactically and grammatically correct sequence of vocabulary words but a contiguous collection of words ended by a period having taken into account some "cleaning" rules.

The outline of the paper is as follows. An overview of TLI is provided in Sect. 2. The application of discrete HMMs to TLI is presented in Sect. 3. Experimental results are reported in Sect. 4 and conclusions are drawn in Sect. 5.

2 Text-based Language Identification

Text-based language identification (TLI) is seen as a classification task. That is, given a collection of texts written in a number of known languages the objective is to determine the language an input document is written in. The decision is made upon document characteristics usually at the word or character level. Working at the character level generally seems to be a more robust approach. An overview of TLI issues can be found in [3].

Several methods for TLI have been proposed. One of them is to use a vocabulary of each language and decide upon the number or percentage of words found in each vocabulary. However, this approach has difficulties in coping with inflected words or spelling errors. Variations of this method are the use of the most frequent words in each language and the use of grammatical or function words like prepositions, determiners, pronouns and conjunctions [4].

Another extensively applied approach is the so-called character n-grams. The most frequent sequences of n characters, where n can take more than one values, are found from a training corpus and the number of occurrences of these sequences in the test document is used in the decision criterion either directly or through the formation of probability estimates [4], [5]. In [6] the ranking of

the most frequent character n-grams is used instead of their absolute frequencies. There is also the possibility of using word n-grams, where probabilities of word sequences of length n, instead of character sequences, are estimated. In this case, more training data and computational resources are required. The technique based on character n-grams appears to be the most flexible.

For the identification method several approaches exist as well. In [7] both character n-grams and words are considered as features of a vector-space based categorizer. The relative entropy also called Kullback-Leibler distance is considered in [3]. The use of Markov models for TLI is considered in [8], while the application of decision trees is studied in [9]. Potential applications of TLI to web documents are closely related to: cross-language information retrieval, electronic libraries construction, and machine translation of online texts.

3 Discrete HMMs for Text-based Language Identification

3.1 Introduction

Hidden Markov Models (HMMs) [10] are a quite old concept that is thoroughly investigated and used not only for speech recognition, but also for other applications, like biometrics, bioinformatics and network communications. There are two general kinds of HMMs, the continuous (density) models and the discrete models depending on whether the observations used are continuous or discrete random variables, respectively. In our case we are interested in discrete models because we represent characters as discrete random variables, which admit integer values. What is achieved by using an HMM is that the parameters of the model are adjusted to specific inputs, and therefore a suitable representation of this input is acquired. The input in which we are interested in the present study is written text in one of the examined languages. The representation attained by the HMMs can be thought as the probability distribution of the language characters derived from text realizations from a given language. The idea that was followed for the whole experiment is that one discrete HMM (DHMM) per language is constructed.

The training of this DHMM is done using a training portion of the corpus as one observation sequence containing integers. This sequence is iteratively presented to the DHMM an adequate number of times until a termination criterion is met. Having done this for as many DHMMs as the number of languages under examination, the language identified in the test document is the one associated to the DHMM that best represents this document.

A DHMM contains a set of N interconnected states $S = \{s_1, s_2, \ldots, s_N\}$ and M symbols $V = \{v_1, v_2, \ldots, v_M\}$. It can enter each of the states at a given time instant t by providing an observation symbol (or feature) o_t as output of the entered state q_t. The symbol o_t is generated using an output (observation) discrete probability distribution $b_l(k) = P(o_t = v_l | q_t = s_k)$, $1 \leq l \leq M$ for the entered state. Between the states the probability of transition is $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$, $1 \leq i \leq N$, $1 \leq j \leq N$. Non-connected states have $a_{ij} = 0$. The

entire model can be written as $\lambda = (A, B, \pi)$, where B is the output probability distribution matrix, A is the transition probability matrix and π denotes the initial state distribution.

In the case under study two special states are used, an entry state s_1 and an exit state s_N, in addition to the other "standard" states. These special states are non-emitting, so they do not have output probability distributions associated with them but only transition probabilities. The entry state is always the first state of the model: $q_1 = s_1$, $\pi_1 = 1$, $\pi_i = 0$, $2 \leq i \leq N$, and the exit state the last. or from the exit state. The summation of the transition probabilities which initiate at a given state equals one ($\sum_{j=1}^{N} a_{ij} = 1$, $1 \leq i \leq N$), except for the exit state, since there is no possible transition from the exit state ($a_{Nj} = 0$, $1 \leq j \leq N$). The value $j = 1$ in the summation can be disregarded, since there is no possible transition to the entry state ($a_{i1} = 0$, $1 \leq i \leq N$). We also used $a_{12} = 1$, meaning that the only transition from the entry state is toward the first standard state.

An ergodic DHMM is one that from every standard state there is the possibility of transition to every other standard state plus this state itself. This topology is tested in our experiments. Experimental evidence showed that the best case is to use only one standard state, thus having three total states per DHMM. In the literature this type of DHMM is referred to as single state DHMM neglecting the special states. The fact that using multistate DHMMs does not contribute toward better identification rates is in accordance with the results obtained when continuous density HMMs were applied for speech-based language identification in [2].

A possible abstract explanation of the meaning of states is that of considering them as groups of characters with common characteristics. For example vowels and consonants could form two different groups. The formation of these groups is not a priori known and is left to be decided at the learning phase. The latter fact agrees with the theoretical assumption of DHMMs that the states are not directly observable but hidden, as opposed to simple discrete-time Markov processes. Each observation is a probabilistic function of the state from which it is derived.

3.2 Formation of observation features

The characters considered for the formation of observation features are the 26 English alphabet letters ('a'-'z'), the 32 ISO Latin-1 characters used in western European languages (Table 1), and a symbol assigned for the end of sentence and the end of word yielding a total of $M = 59$ symbols. For the 26 English alphabet letters as well as for the 30 ISO Latin-1 characters (the 32 mentioned except 'szlig' and 'yuml') where both an uppercase and a lowercase version exists, both versions are treated the same. Furthermore, multiple white space occurrences are considered a single white space. Experimental evidence, in our case, showed that the inclusion of the remaining characters in the observations, as well as the distinction between lowercase and uppercase letters, cause the average performance to deteriorate.

Table 1. ISO Latin-1 lowercase letters used for the observation features. The left column for each letter is the name used in HTML entities and the right column is its written representation.

agrave	à	egrave	è	eth	ð	ugrave	ù
aacute	á	eacute	é	ntilde	ñ	uacute	ú
acirc	â	ecirc	ê	ograve	ò	ucirc	û
atilde	ã	euml	ë	oacute	ó	uuml	ü
auml	ä	igrave	ì	ocirc	ô	yacute	ý
aring	å	iacute	í	otilde	õ	thorn	þ
aelig	æ	icirc	î	ouml	ö	szlig	ß
ccedil	ç	iuml	ï	oslash	ø	yuml	ÿ

Several kinds of (observation) features were extracted based on these characters. The first is the mapping of characters according to their ISO Latin-1 code value. The resulting codes are hereafter called *symbol features* f_{s}. In our case the number of symbols for the symbol features is $M_{\mathrm{s}} = 59$. Another two representations consider the value of the difference between the ISO Latin-1 code values of two characters. The first representation is applied on two consecutive characters, while the second one on two characters separated by another character. The resulting codes are called *delta-zero* $f_{\Delta 0}$ and *delta-one* $f_{\Delta 1}$ features, respectively, where Δ denotes the difference, and the number on its right denotes the characters omitted between the characters whose difference is calculated. For the number of symbols for these delta features it can be verified that $M_{\Delta 0} = M_{\Delta 1} = 2M_{\mathrm{s}} - 1$. In addition, some combinations of these features were tested, for example *symbol-delta-zero* $f_{\mathrm{s}\&\Delta 0}$ and *symbol-delta-zero-delta-one* $f_{\mathrm{s}\&\Delta 0\&\Delta 1}$. The code values, when used in the combinations, were arranged in a non-overlapping, interleaved way. Non-overlapping, in the sense that the values of the constituent features $(\mathrm{s}, \Delta 0, \Delta 1)$ are mapped on successive, disjoint, integer ranges beginning from the unity. Interleaved, meaning that in a sequence of combined features, the values of the constituent features alternate successively. This arrangement seemed to achieve the best results. As an example the word "acb" is encoded as follows for the 59 characters used when having $f_{\mathrm{s}\&\Delta 0\&\Delta 1}$ as the feature:

$$f_{\mathrm{s}\&\Delta 0\&\Delta 1}([\mathrm{a\,c\,b}]) = [1\,119\,236 \,|\, 3\,120\,238 \,|\, 2\,117\,236]$$

($M_{\mathrm{s}\&\Delta 0\&\Delta 1} = 5M_{\mathrm{s}} - 2 = 293$, ranges: 1-59 f_{s}, 60-176 $f_{\Delta 0}$, 177-293 $f_{\Delta 1}$)

Experiments conducted showed that in order of increasing identification rates the features seemed to be:

$$f_{\mathrm{s}} < f_{\Delta 1} < f_{\Delta 0} < f_{\mathrm{s}\&\Delta 1} < f_{\mathrm{s}\&\Delta 0} < f_{\mathrm{s}\&\Delta 0\&\Delta 1}.$$

Also, experiments with $f_{\mathrm{s}\&\Delta 0\&\Delta 1\&\Delta 2}$ resulted in a little worse results than $f_{\mathrm{s}\&\Delta 0\&\Delta 1}$. Moreover, the order of the constituent features seemed not to change the results, while the combined features include a small time overhead.

3.3 Learning and Identification Processes

The general learning process uses a segmental K-means algorithm. Initially all symbols in each standard state are set to be equally likely, that is $b_j(k) = \frac{1}{M}$, $1 \leq j \leq M$, $2 \leq k \leq N-1$. At the initialization phase, the observation vectors $\mathbf{o}_i = (o_{i1}, o_{i2}, \ldots, o_{iT})$ are presented to the corresponding DHMM and they are segmented equally among the (standard) model states s_i, in other words the vectors assigned to a state are considered as being generated by this state. Having done this, maximum likelihood probability estimates are found for the model parameters $\lambda = (A, B, \pi)$, so that $P(\mathbf{o}_i|\lambda)$ is maximized. The *Viterbi alignment* is used next for the resegmentation of the observation vectors and the recomputation of the MAP estimates of A, the transition probability matrix, and B, the output probability distribution matrix, until convergence is achieved or an upper limit on the iteration count is reached. The transition probabilities at each iteration are estimated by using the number of times each transition is made in the Viterbi alignments and normalizing.

In our case there is only one observation vector $\mathbf{o}$ and the whole process is accordingly adjusted to this fact. The approach of considering the whole (training or test) text as one observation vector of features instead of segmenting it into more vectors is followed because the sentences to be identified are of variable size, which is in contrast to the demand that the vectors should have the same dimension. Their size depends on the position of the period and is therefore varying.

Later, there is the possibility of using a re-estimation phase during which each observation vector is assigned to every state in proportion to the probability of the model being in that state when the vector was observed. The probability of state occupation is calculated efficiently using the forward-backward algorithm. The whole process is called Baum-Welch re-estimation. Experimental evidence showed that the application of the *Baum-Welch re-estimation* leaves the results almost unmodified and was subsequently omitted.

At the identification stage the *Viterbi decoding* is used. The purpose of this algorithm is to find the best path of state transitions $\mathbf{q} = (q_1, q_2, \ldots, q_T)$ (in the maximum likelihood sense) given the observation vectors $\mathbf{o}_i$, using the model parameters λ. The log probability of a path is computed by taking the summation of the log output probabilities and the log transition probabilities. In our case the Viterbi decoding becomes trivial, since there is only one standard state to which the symbols should be assigned. For the implementation of TLI a structure of the trained DHMMs in parallel is constructed. For each DHMM there is a probability outcome computed for the best path that was found by the Viterbi algorithm. The identification result is found by choosing among the DHMMs the one that is more likely to have produced the test observation vector, that is the one with the highest probability outcome. This approach is also met in isolated word recognition using HMMs [10]. The HMM that maximizes the probability of the observation sequence when the parameters of the HMM are given is chosen and since each HMM represents one word, or in our case one language, this is declared as the recognized one. For the implementation of the learning and the

identification phases version 3 of the HTK toolkit [11] is used. The identification process is depicted in Fig. 1.

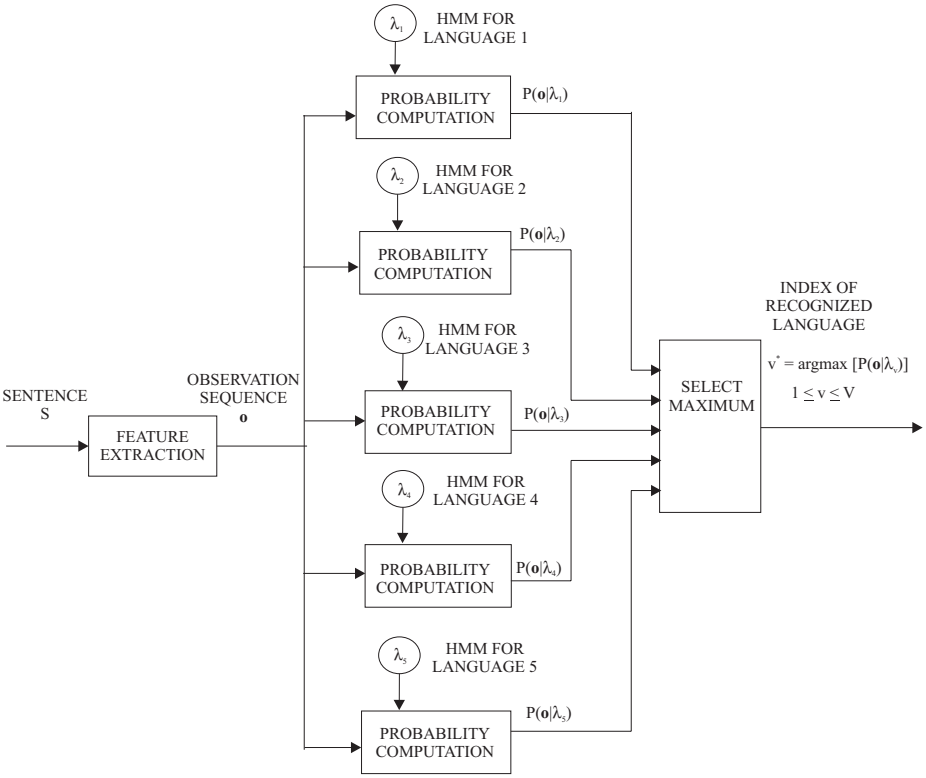

Fig. 1. Block diagram of the identification stage of a language recognizer using DHMMs.

4 Experimental Results

4.1 Experimental Setup

In order to test the performance of the suggested TLI technique 151 HTML pages p_i were manually selected and collected over the Internet, so that a small tourism corpus is created. Pages with streaming text were quested, avoiding those full of short lists, of price tables and of abbreviations. We also tried to find monolingual documents with little text in other languages, but without requiring all the text to be strictly in one language. Most documents were collected from hotel promotion sites where many hotel pages were present (e.g. `www.rosciolihotels.com`,

`www.venere.com`, `www.hotels.fr`). Moreover, we exploited the fact that some web sites provide their context in more than one language.

At its current state the corpus consists of web pages related to hotels and accommodation written in five languages: English (language code "en"), German ("de"), French ("fr"), Spanish ("es") and Italian ("it"). Not only is this corpus used for training purposes, that is the construction of statistical language models, but also for test purposes, that is the evaluation of the classification technique. About thirty documents for each language were collected. Although the size of the corpus may seem inappropriately small, experimental evidence showed that convergence to high identification ratios is attained after a certain increase in the relatively small training sizes. Small sizes, on the order of tenths or hundreds of kilobytes, are also used in similar efforts like [6].

The documents selected were annotated so that ground truth is incorporated. There are five categories L_i used for the annotation, one for each of the five languages considered. The annotation was not performed in detail, that is each of the sentences s_{pi} of a whole document p was considered to be in the same language $s_{pi} \in L_j \, \forall i$ as the one that was assigned to the document ($p \in L_j$). Nevertheless, there were actually few sentences s_{pk} written in a different language than the language of the document they were part of: $s_{pk} \notin L_j$, $p \in L_j$. This fact introduces small errors in the training and the evaluation processes.

Having annotated the collection of the whole multilingual corpus C, the documents formed five groups C_l according to their language l. For each group a certain procedure, depicted in Fig. 2, is followed to segment it into the training $C_{l(\mathrm{trn})}$ and test $C_{l(\mathrm{tst})}$ part. Firstly, a cleaning process is executed for the removal of HTML structures and other useless data in p_i, while all ISO Latin-1 representations are converted to one. The result of this are the plain documents $p_{i(\mathrm{pln})}$.

After this, the separation of $p_{i(\mathrm{pln})}$ into the documents used for training $p_{i(\mathrm{pln})(\mathrm{trn})}$ and the ones used for testing $p_{i(\mathrm{pln})(\mathrm{tst})}$ is made. A sufficient number of documents $N_{l(\mathrm{trn})} = 20$ are selected to be included in the training corpus of each language and the rest $N_{l(\mathrm{tst})} \simeq 10$ are selected for inclusion in the test corpus. The plain train and test documents are in turn split into as many parts as the "sentences" they contain forming a training $s_{li(\mathrm{trn})}$ and a testing $s_{li(\mathrm{tst})}$ sentence pool per language. Sentences of $p_{i(\mathrm{pln})}$ with less than 20 characters are omitted.

As for the "sentence" notion, it should be noted that the sentence splitting in this context is performed in a wide sense and not in a strict grammatical sense. Generally, sentences are split whenever a period is encountered except for certain cases, which include emails, URLs and acronyms, where the period does not signify the end of a sentence. Where a period is followed by a letter which in turn is followed by a period and so on, the whole string is considered as an acronym inside a sentence. To ensure that a sentence does not expand over more than one HTML page an extra period is added at the end of each HTML page when merging takes place.

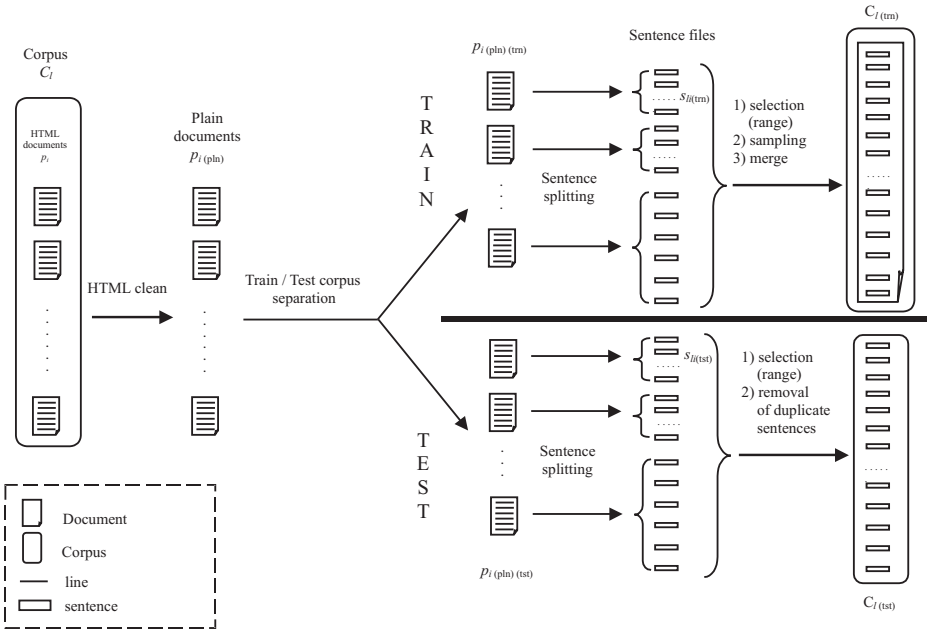

Fig. 2. Block diagram of the segmentation of the corpus into the training and test part.

The training corpus $C_{(\mathrm{trn})}$ was decided to be five texts $C_{l(\mathrm{trn})}$, one for each language l. Each text is created as follows. First, from the textually ordered sentences of the training sentence pool for the corresponding language l, $s_{li(\mathrm{trn})}$, those that lie in a specific range of lengths in "clean text" characters are selected. The range used in our case is $20 - 100000$ so that practically all sentences over 20 "clean text" characters are included. Afterward, these sentences are sampled taking one out of r_{samp} sentences. We experimented with $r_{\mathrm{samp}} = 1$ (all samples) and $r_{\mathrm{samp}} = 10$, so that a comparison for the training corpus size effect on the identification rates is enabled. The resulting sentences are merged into $C_{l(\mathrm{trn})}$. Finally, the sizes of the each of $C_{l(\mathrm{trn})}$ are made equal to the minimum of the five by omitting the last extra characters in order to provide the same amount of training material for each language.

By "clean text" characters we mean the ones that are used in the construction of the features, that is the 59 symbols referred in Sect. 3, except the ones that are included in emails, URLs, acronyms and a small stoplist (containing currency acronyms). An algorithm for the detection of URLs, emails and acronyms is used so that these do not interfere with the linguistic content of the documents. The stoplist contains some frequently used tourism and currency related words that are found in the corpus documents irrespective of the language. Finally, if 75% or more of the words in a sentence are "only-first-capital" ones, that is, begin with a capital letter and do not contain other capital letters, the latter words are

not taken into account for the feature extraction. This fact compensates for the inclusion of sentences in a list form, where many proper names are enumerated.

The fact that the training corpus may include some text that is not characteristic of the language structure, or the fact that a small part of it may even be in a different language, since the pages were not manually modified, may be the source of wrong classifications. One reason for this is that this way the training corpus can be updated more easily. A manual intervention is usually onerous and it may also be subjective.

On the other hand, the test corpus $C_{(\mathrm{tst})}$ consists of five groups $C_{l(\mathrm{tst})}$ of separately considered sentences s_{li}, one per language l. From the textually ordered sentences of the test sentence pool for language l, $s_{li(\mathrm{tst})}$, only those that lie in a specific range of lengths in "clean text" characters are selected. Sentences with the same "clean text" content are detected and taken into account only once. For each sentence-long "document" $s_{li(\mathrm{tst})}$ of $C_{(\mathrm{tst})}$ an identification result is extracted.

Using $C_{(\mathrm{trn})}$ the observation vectors are created and 5 DHMMs are trained, one per language as described in Sect. 3. Subsequently, $C_{(\mathrm{tst})}$ is used to implement the evaluation procedure. For each sentence of the 5 languages the observation vectors are extracted and they are presented to the structure of DHMMs in parallel, where the decision for the language is made. It is worth noting that only "clean text" characters are included in the features.

4.2 Performance Evaluation

Our experiments were focused on the evaluation of the effect of various parameters on the identification rates and also the comparison of our technique with a standard technique using variable character n-grams, as described in [6]. The same preprocessing steps were followed for both techniques as described earlier in this section. Table 2 and Table 3 were created using $f_{\mathrm{s}\&\Delta 0\&\Delta 1}$ as features for DHMMs. By comparing the data of these tables we reach the following conclusions.

As for the size of the training corpus it appears that the bigger the size is, the higher rates are achieved. Nevertheless, the effect does not seem to be very big regarding the increase ratio. On the other hand, the average length of the test sentences appears to play a significant role, since its increase leads to a significant increase in the rates. Although the variable character n-grams attain better results, there is a small difference in the rates. What is more, in some cases for Spanish and Italian the DHMMs yielded higher rates, which enables the consideration of a hybrid technique for better results. Experiments using non-tourism related plain text documents for the training gave worse average results for both methods. Finally, it should be noted that these percentages include a small error percentage since the ground truth incorporated is not absolutely correct.

For further evaluation the identifier is tested with full html documents in the following manner. First, the HTML document d is cleaned in the way that was described in the corpus preparation procedure. The cleaned document $d_{(\mathrm{cln})}$ that

Table 2. Average identification rate in five languages. The training sentences have at least 20 characters and their total size is 22133 characters for each language.

Method	Range of test sequence length in characters			
	20–100	100–200	50–150	20–200
	(avg) 68	142	100	102
	(total) 18804	31198	33550	49796
DHMMs	94%	99%	97%	96%
Var. n-grams	96%	100%	99%	97%

Table 3. Average identification rates in five languages. The sampled training sentences (1 out of 10) have at least 20 characters and their total size is 2183 characters for each language.

Method	Range of test sequence length in characters			
	20–100	100–200	50–150	20–200
	(avg) 68	142	100	102
	(total) 18804	31198	33550	49796
DHMMs	90%	97%	95%	93%
Var. n-grams	92%	100%	98%	95%

resulted is segmented into its sentences s_{di}. For each sentence, an identification result is found. In the final stage, the sentences are merged into a language annotated document $d_{(\mathrm{rec})}$ using an HTML-like tag. In $d_{(\mathrm{rec})}$, whenever a group of more than one contiguous sentences is classified into the same language, the annotation is given for the whole group. One such example is given in Fig. 3.

5 Conclusions and Future Work

We demonstrated by experiments that the use of DHMMs for TLI on web documents attains high recognition rates, taking account of the inherent domain difficulties, and can therefore be regarded as a viable alternative of the other techniques used for TLI. Judging from some non-overlapping wrong results, especially for the smaller ranges of lengths of test sentences, there seems to be room for a hybrid technique between DHMMs and variable character n-grams which would yield better results than either of them. The proposed method should also be tested on other corpora.

References

1. Y. Muthusamy, E. Barnard and R. Cole, "Reviewing Automatic Language Identification," *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 33–41, Oct. 1994.
2. M. Zissman, "Automatic Language Identification Using Gaussian Mixture and Hi dden Markov Models," in *Proc. of ICASSP'93*, vol. 2, pp. 399–402, 1993.
3. P. Sibun and J. Reynar, "Language Identification: Examining the Issues," in *Proc. of SDAIR'96*, pp. 125–135, 1996.

<lang name="it">
Hotel Gabriella : Presentazione L'Hotel Gabriella accogliente albergo a conduzione familiare, è situato nel centro storico della città. Le camere tutte recentemente ristrutturate, sono provviste di servizi privati, tv color, cassette di sicurezza, aria condizionata (su richiesta), asciugacapelli e telefono. Inoltre, l'albergo dispone di bar con servizio di prima colazione a buffet. Prenotazioni per gite turistiche per la città e dintorni. Si accettano carte di credito e traveller's cheque. </lang>
<lang name="en">
The Hotel Gabriella is a comfortable, family-run establishment located in the town's old centre. the rooms, which have all recently been restructured, have a private bathroom, colour television, strong-box, air conditioning (on request), hair dryer and telephone. What is more, the hotel has a bar with breakfast and buffet service. Tours of the town and surrounding area can be booked here. Credit cards and travellers cheques welcome. </lang>

Fig. 3. The output of the algorithm having as input the multilingual web page `http://www.expoitalia.com/hotelgabriella/presentazione.htm`. 10 out of 10 small sentences are correctly identified.

4. G. Grefenstette, "Comparing two Language Identification Schemes," in *Proc. of JADT'95*, 1995.
5. H. Combrinck and E. Botha, "Text-based Automatic Language Identification," in *Proc. of the 6th Annual South African Workshop on Pattern Recognition*, 1995.
6. W. Cavnar and J. Trenkle, "N-Gram-Based Text Categorization," in *Proc. of SDAIR'94*, pp. 161–175, 1994.
7. J. Prager, "LINGUINI: language identification for multilingual documents," in *Proc. of HICSS-32*, 1999.
8. T. Dunning, "Statistical Identification of Language," Tech. Rep. CRL MCCS-94-273, New Mexico State University, Las Cruces, NM, 1994
9. J. Häkkinen and J. Tian, "N-gram and Decision Tree Based Language Identification for Written Words," in *Proc. of ASRU'01*, 2001.
10. L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
11. S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev and P. Woodland, *The HTK Book (for HTK version 3.0)*, Microsoft Corporation, 2000.