Pattern Recognition Letters 33 (2012) 2077–2084

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A high performance centroid-based classification approach for language identification

Hidayet Takçı [a,*], Tunga Güngör [b]

[a] Department of Computer Engineering, GYTE, Kocaeli 41400, Turkey
[b] Department of Computer Engineering, Boğaziçi University, İstanbul 34342, Turkey

ARTICLE INFO

Article history:
Received 3 November 2010
Available online 28 June 2012

Communicated by S. Sarkar

Keywords:
Language identification
Centroid-based classification
IDF (inverse document frequency)
ICF (inverse class frequency)

ABSTRACT

Centroid-based classification is a machine learning approach used in the text classification domain. The main advantage of centroid-based classifiers is their high performance during both the training stage and the classification stage. However, the success rate can be lower than the other classifiers if good centroid values are not used. In this paper, we apply the centroid-based classification method to the language identification problem, which can be considered as a sub-problem of text classification. We propose a novel method named as inverse class frequency to increase the quality of the centroid values, which involves an update of the classical values. We also use a feature set formed of individual characters rather than words or *n*-gram sequences to decrease the training and classification times. The experiments were performed on the ECI/MCI corpus and the method was compared with other methods and previous studies. The results showed that the proposed approach yields high success rates and works very efficiently for language identification.

1. Introduction

Language identification has been considered as a form of text classification since 1990s. Several classification approaches have been used for identifying the language of documents, including Bayesian classification (Dunning, 1994), relative entropy-based classification (Sibun and Reynar, 1996), centroid-based classification (Kruengkrai et al., 2005), Markov models (Xafopoulos et al. 2004), decision trees (Hakkinen and Tian, 2002), neural networks (Tian and Suontausta, 2003), support vector machines (SVM) (Zhai et al., 2006), and multiple linear regression (Murthy and Kumar, 2006). As a result of plenty of works, language identification is considered as a solved problem. However, most of these studies offer off-line solutions to the problem and there is only a limited number of studies that work on-line. Moreover, while the methods proposed in these studies were analyzed in terms of their success rates, the run-time performance of the classifiers was not taken into account.

Centroid-based classifier is a vector space-based classifier (Salton, 1989). In centroid-based classification, each document and class is represented by a vector, each entry of which corresponds to a term (feature) in the document collection and contains the weight of the term in that document. It is a simple and efficient classifier and these properties make it preferable when compared with mathematically more complex classifiers. The high run-time performance of centroid-based classification is quite useful for on-line tasks like language identification. Han and Karypis (2000) compared the centroid-based classifier with k-nearest neighbor (k-NN), C4.5, and naive Bayes classifiers, and showed that it yields comparable results with better time complexities. In another study, Chuang et al. (2000) proposed a method similar to centroid-based classification and they employed hierarchical classifiers based on a concept hierarchy. The time complexity analysis of the algorithm showed that it is a computationally efficient algorithm.

Centroid-based classifiers yield high success rates in several topics related to text classification. Soonthornphisaj et al. (2002) proposed a centroid-based algorithm for filtering spam e-mail messages and compared it with naive Bayes and k-NN classifiers. Although centroid-based classification is a successful approach in general, it has the tendency to be affected from small variations in the data. In a related work, it was shown that filtering the outliers in the data improves the classification performance of the classifier by about 10% when compared to the classical centroid-based approach (Shin et al., 2006). A variation of centroid-based classification, referred as nearest centroid classification, has been applied to the DNA microarray classification problem (Wang and Zhu, 2007). The centroid-based approach has been used for large scale text classification, which can be defined as the classification problem in the case of a very large number of classes (Miao and

[*] Corresponding author. Tel.: +90 212 3597094; fax: +90 212 2872461.
E-mail address: htakci@bilmuh.gyte.edu.tr (H. Takçı).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.06.012

Qiu, 2009). Two characteristics of this domain, data sparsity and highly imbalanced classes, cause a decrease in the performance of the traditional text classification schemes. Centroid-based classifiers operate in linear time that is proportional to the number of data samples and are thus quite efficient. Based on this property, they are suitable for classification tasks that work on large volumes of data.

Although centroid-based classification shows a successful behavior in a variety of tasks, it has two main problems (Tan, 2007). The first one is the model misfit in the sense that the algorithm is too sensitive to the training data. The second problem is its relatively lower success performance in some domains when compared to other classification algorithms. One solution for the performance problem is applying normalization and smoothing techniques on the data, which was shown to increase the success rates significantly (Lertnattee and Theeramunkong, 2006). Another approach is increasing the discriminative power of the centroid values that represent the classes. Term weighting methods can be used for this purpose. Adjusting the weights of the centroids using inter-class and intra-class term frequencies increases the classification performance of the classifier (Tan, 2008; Guan et al., 2009). The most widely used term weighting scheme is tf-idf (term frequency – inverse document frequency), in which terms common to most of the documents are less discriminative for classification (Salton and Buckley, 1988). There are some more sophisticated term weighting schemes such as Okapi BM25 (Robertson and Walker, 2000), LTU (Buckley et al., 1996), TF-ICF (term frequency – inverse corpus frequency) (Reed et al., 2006), and STW (supervised term weighting) (Debole and Sebastiani, 2003).

Another problem of the centroid-based classifier is the large number of features in the feature vector. The problem can be solved by either designing a small feature set or feature selection. A number of techniques for feature selection such as information gain, chi square, and mutual information have been applied successfully in the literature (Chizi et al., 2009). Syntactic and semantic information in the sentences can also be incorporated into the feature selection model (Özgür and Güngör, 2010).

In this paper, we propose a novel approach for language-based text classification. The decisions underlying the proposed approach and the properties of the method are listed below:

- The main problem in vector space-based models is the high dimensionality of the feature set. In language identification systems that use common words or n-gram sequences, the number of features is between 1000 and 3500 (Grefenstette, 1995). The method we propose is based on characters (letters and special characters) that occur in the documents and we use a feature vector formed of 54 features. Since it is a character-based approach, feature extraction and processing can be done very efficiently.
- Language identification systems are generally used on-line in the preprocessing stages of several application areas such as information retrieval, machine translation, and text summarization. Thus, the time complexity of these systems should be as low as possible. In this respect, in addition to using a low dimensional feature space, selection of the classification method is critical. For this purpose, we use the centroid-based classification approach, which is highly efficient for training and testing of data.
- To overcome the accuracy problem of centroid-based classification, we propose a method named as ICF (inverse class frequency). ICF determines the importance of a term based on the number of occurrences of the term in the classes. In this respect, it differs from inverse document frequency by making use of the term frequency distributions within the classes rather than the whole corpus.

The rest of the paper is organized as follows. Section 2 gives a literature survey related to language identification studies. Section 3 explains briefly the centroid-based text classification method. Section 4 gives the details of the methodology used in this work. Section 5 explains the data set and the experiments and compares the results of the proposed method with previous studies. Section 6 concludes the paper.

2. Related work on language identification

There has been a significant amount of work on language identification and success rates close to 100% were obtained for large documents. However, similar performance ratios cannot be obtained for short documents like query texts. In most of the studies, some common words and character and word n-grams are accepted as the features that best distinguish different languages. Since language identification is a classification problem, several statistical and machine learning techniques used in text classification have been successfully applied to this problem.

One of the pioneering studies in language identification used common words and unique combinations of characters (Grefenstette, 1995). Using common words or letter combinations is known as a linguistically-oriented language identification technique. A widely-known method that is frequently used in language identification is the *n*-grams approach. Cavnar and Trenkle (1994) employed an *n*-gram method that made use of the list of the most frequently observed character *n*-grams. Another *n*-gram method was proposed by Adams and Resnik (1997). The implementation in this work was based on the character *n*-gram approach of Dunning (1994). Statistical approaches have also been applied for language identification. One of these studies is the work of Dunning (1994), in which Markov models over *n*-gram frequencies were used. Another language identification study which uses a Markov model was done by Xafopoulos et al. (2004).

Several machine learning techniques have been used in the domain of language identification. Combrinck and Botha (1995) used histogram method for language based classification on 12 languages. Prager (2000) employed a vector space model and performed experiments on 13 languages using *n*-grams. Suzuki et al. (2002) used a method based on *n*-grams that was capable of identifying the language and character encoding schemes for a web document together. Takçı and Soğukpınar (2004) used a character-based method for language identification. In this work, 22 characters that occur frequently in different languages were used and the method was applied to four languages. Ng and Selamat (2009) performed language identification on Arabic texts by using letters as the features. They proposed a new letter weighting approach and obtained about 99.75% success rate.

As stated previously, although language identification can be regarded as a solved task for long documents, this is not the case for texts formed of a limited number of words. Winkelmolen and Mascardi (2011) applied statistical methods for identification of language in short documents such as SMS texts. In another work, the language of query texts was identified by analyzing The European Library (TEL) logs (Bosca and Dini, 2010). Gottron and Lipka (2010) compared different approaches for language identification related to queries. They employed naive Bayes, Markov chain, frequency rank and vector space models.

Vatanen et al. (2010) employed different *n*-gram models for the analyses of short texts formed of 5–21 characters. In order to decrease the execution time of the methods, they also employed the model pruning technique with different pruning parameters. In another work, Grothe et al. (2008) showed that changing the out-of-place approach proposed by Cavnar and Trenkle (1994) dynamically causes an increase in the classification accuracy. They performed experiments for comparison of different language identification algorithms. An interesting study on language identification was done by Bhargava and Kondrak (2010), which aims at identifying the language using data of proper names by SVMs. The experiments on the Transfermarkt corpus (a corpus containing European soccer player names) and 13 languages yielded about 80% success.

Although language identification studies mostly involve supervised learning strategies, there are some works based on unsupervised methods. Amine et al. (2010) use artificial ants and k-means algorithms together for language identification. The authors presented a method based on the behavior of ants having collective and individual characteristics and ability to gather and sort objects. The method does not require the number of classes to be given a priori and this is determined by the artificial ants algorithm. The experiments were performed using three different similarity measures and it was observed that the choice of the similarity measure has an important role in this domain.

3. Centroid-based text classification

The concept of centroid denotes the central value of a set of data. The data that belong to a class are represented by a point that is at the center of the class (the centroid value). In this way, each class is represented by a single centroid, based on the assumption that the central value of a set of data is the best representative of these data. In centroid-based text classification, during the training phase, simply the centroid value of each class is calculated. To identify the correct class of a new data sample during testing, the similarity of the sample to each centroid is calculated and it is assigned to the most similar class.

Centroid-based text classification is an efficient method based on the vector space representation model. Each document is represented by a vector $\vec{d}$ and each element of the vector corresponds to a term in the document collection. Usually the tf-idf metric is used to assign weights to the terms. An element in a centroid vector corresponding to a term denotes an average value for this term in all the documents in this class and is accepted as a representative value for the whole class with respect to the term. We will use the following notation throughout the paper:

k	number of classes
n	number of terms
d	a document
$\vec{d}$	term weighting vector of document d
c_i	class i
$\vec{c}_i$	centroid vector of class c_i. $\vec{c}_i = [c_{i1}, c_{i2}, \ldots, c_{in}]$
t_j	term j
D	the set of documents in the corpus
D_{c_i}	the set of documents belonging to class c_i
C	the set of classes in the corpus
C_{t_j}	the set of classes containing term t_j
D_{c_i,t_j}	the set of documents belonging to class c_i and containing term t_j

There are mainly two methods for forming the centroid vectors from the training data (Guan et al., 2009). The first one is arithmetic average centroid (AAC), where the elements in a centroid are simply the mean values of the corresponding term weights in the document vectors belonging to the class. The centroid vector $\vec{c}_i$ of class c_i is formed as follows:

$$\vec{c}_i = \frac{1}{|D_{c_i}|} \sum_{d \in D_{c_i}} \vec{d} \tag{1}$$

The second method is cumuli geometric centroid (CGC) that uses the sum of the term weights rather than their average:

$$\vec{c}_i = \sum_{d \in D_{c_i}} \vec{d} \tag{2}$$

There are different variations of the basic AAC and CGC methods. An effective one of these variations is the CFC (class feature centroid) method (Tan, 2008), where the weights in the centroid vector of a class are computed by taking into account the document and class ratios for the terms. The weight of term t_j in class c_i, c_{ij}, is calculated as follows:

$$c_{ij} = b^{\frac{|D_{c_i,t_j}|}{|D_{c_i}|}} \log\left(\frac{|C|}{|C_{t_j}|}\right) \tag{3}$$

where b is a constant greater than one. The first part of the equation is referred as the inner-class term index and the second part as the inter-class term index.

As each class is represented by a centroid, a document in the test set is categorized using a similarity measure. The test document is compared to each centroid and it is assigned to the class yielding the maximum similarity value. A commonly used similarity measure is cosine similarity. Given a document d, the similarity between d and each class is calculated and the maximizing similarity value is selected:

$$sim(d, c_i) = \frac{\vec{d} \cdot \vec{c}_i}{|\vec{d}| \cdot |\vec{c}_i|}, \quad \text{for } i = 1, \ldots, k \tag{4}$$

$$\text{argmax}_i sim(d, c_i)$$

4. The methodology

We propose a new method, inverse class frequency, to improve the discriminative power of the centroids and thus the training and classification performances. Similar approaches that take the term distributions in classes into account for text classification have been proposed in the literature (Min, 2003; Keim et al., 2009). The approach we propose in this study differs from the previous works in three ways. First, we use the class frequencies in centroid-based learning and for language identification rather than document classification. Second, instead of simply incorporating a ratio of the number of classes in the formula, we use class frequencies as an updating factor in the centroids. Third, we use a character-based feature set and there is no need for a feature selection step. To the best of our knowledge, this is a novel approach in centroid-based classification and language identification.

4.1. Inverse class frequency

In text classification systems, a critical factor that affects the performance of the classifier is term weighting (Leopold and Kindermann, 2002). The weight of a term for a particular document indicates how much the term is related to that document. Debole and Sebastiani (2003) and How and Kulathuramaiyer (2004) use supervised term weighting and give an analysis of some popular term weighting metrics. In the former one, a weighting score is computed for a term with respect to its relation to class labels. In the latter one, term weighting is done by choosing the most relevant items in the classes using a relevance score. An unsupervised approach is used by Ko and Seo (2004) that is based on the bootstrapping framework and a feature projection technique.

In this work, we use a new term weighting scheme named as ICF (inverse class frequency). In ICF, the relevance of a term for a class depends on its average frequencies in that class and in other classes in the corpus. This is different from the idf method and its variations which consider the number of documents or classes a term occurs in. Note that, although they share the same name, ICF and TF-ICF which is another method used in text classification (Reed et al., 2006) are different methods.

ICF makes use of the term frequency distributions within the classes. In ICF, we first form the centroid values using Eq. (1) (c_{ij}^{old}) and then make the following modification on the centroid values to obtain new centroids (c_{ij}^{new}):

$$c_{ij}^{new} = \log\left(\frac{(c_{ij}^{old}+\varepsilon)}{\sum_{i=1\ldots k} c_{ij}^{old}}{}^{*}\, sf\right) \tag{5}$$

where sf denotes a smoothing factor from the set {1, 10, 100, 1000, 10,000} and $\varepsilon < 0.001$. Note that, unlike Eqs. (1) and (2), this formula takes into account both the term frequencies in a class and the ratio of these frequencies to the frequencies in the corpus. A term that appears frequently in the documents of a class but rarely in the documents of other classes will be deemed as an important term for that class. We use the logarithm function and a smoothing factor to prevent large deviations in the centroid values. The training and the testing algorithms for the ICF method are shown in Fig. 1.

As stated previously, when a test document is to be classified, the similarity of the document to each class is computed by using the test document vector and the class centroid vectors as parameters in the similarity measure. In classical methods like AAC and CGC, the test document vector is formed using the tf-idf values of the terms. On the other hand, in ICF, the term weights in the document vector are tf values only. We can show the difference between the two approaches as follows, where the subscripts indicate the term weighting scheme:

For AAC and CGC: Similarity between $\vec{d}_{tf-idf}$ and $\vec{c}_{tf}$.

For ICF: Similarity between $\vec{d}_{tf}$ and $\vec{c}_{tf-icf}$

Since centroid-based classification is a class-based approach and idf involves a corpus-based calculation, in classical methods, we form the centroid vectors using term frequencies and we cannot incorporate corpus-based document frequencies (idf values). In ICF, however, the class frequencies are explicitly used in centroid calculations. In this respect, the importance of a term for a class is obtained with respect to its distribution in the classes. A term used in a class gets a high score if it occurs rarely in other

```
Module Train
Input
    Training set
Output
    Centroid vector $\vec{c}_i$ for each class $c_i$
Local variables
    $freq_{i,k}$ : total frequency of letter $l_k$ in class $c_i$
    $freq_k$ : total frequency of letter $l_k$ in the corpus
begin
    for each class $c_i$
        for each document $d_j$ in class $c_i$
            for each letter $l_k$ in document $d_j$
                $freq_{i,k} = freq_{i,k} + 1$          // Increment class frequency of letter $l_k$
                $freq_k = freq_k + 1$                  // Increment corpus frequency of letter $l_k$
            end for
        end for
    end for
    for each class $c_i$
        for each letter $l_k$
            $c_{ik} = \log(((freq_{i,k} + 0.001) / freq_k) * 10)$   // Calculate centroid value $c_{ik}$ ($sf$=10)
        end for
    end for
end

Module Test
Input
    Test document $d$
    Centroid vector $\vec{c}_i$ for each class $c_i$
Output
    Class index of document $d$
Local variables
    $\vec{f}$ : frequency vector of letters in the corpus
begin
    for each letter $l_k$ in document $d$
        $f_k = f_k + 1$                                // Increment document frequency of letter $l_k$
    end for
    for each class $c_i$
        compute $sim(\vec{f}, \vec{c}_i)$
    end for
    return $\arg\max_i sim(\vec{f}, \vec{c}_i)$
end
```

Fig. 1. Training and testing algorithms for ICF (sf = 10 and ε = 0.001)

Table 1
idf and icf values of example characters ($sf = 10, \varepsilon = 0.001$)

Character	Language								
	DUT	ENG	FRE	GER	ITA	POR	SPA	SWE	TUR
a	0	0	0	0	0	0	0	0	0
	2.68	2.70	2.74	2.55	2.93	2.91	2.92	2.58	2.91
ç	0	0	1	0	0	0.88	0	0	0.32
	0	0	2.67	0	0	3.43	0	0	3.84
ö	0	0	0.47	0	0	0	0.18	0	1.09
	0	0	3.15	1.53	0	0	3.55	0	3.47

classes, but its score decreases if it is also used frequently in the other classes. In addition, the icf value is more sensitive to the training data than the idf value as required by a high performance classification method.

4.2. Discriminative power of inverse class frequency

For language identification, the most important property of ICF is its high discriminative power for language-specific characters (characters that are in the alphabets of one or a few languages only). For instance, considering the set of languages in the European corpus initiative multilingual corpus (the corpus used in this work), the letter ç appears in the alphabets of French, Portuguese and Turkish languages only, while the letter ö appears in the alphabets of French, German, Spanish and Turkish. If we can increase the importance of such characters during classification the performance will be higher, since the classes will be separated more from each other. Table 1 compares the idf values (first row for each character) and the icf values (second row for each character) in different languages of some example characters. The values in the table were obtained using the training data in the corpus used in this work (see Section 5.1). We used $sf = 10$ which gave the best results in the experiments and $\varepsilon = 0.001$ in calculating the icf values.

As can be seen from the table, the importance of the language-specific characters becomes clearer in the ICF weighting scheme. For instance, the icf values of *ç* and *ö* are much higher than the corresponding idf values in the languages they are used. This is a desired property since these letters are important clues in determining the correct language. In addition, while the idf weight of *a* is zero in all these languages showing that it is used in all the documents, the corresponding icf values are slightly different among the languages. Thus the ICF method increases the discriminative power of both the language-specific characters and the common characters.

5. Experiments and results

In this section, we first explain the data set used in the experiments and the performance measures used for evaluation. Then we compare the proposed method in detail with several other methods and previous studies.

5.1. Data set and performance measures

In the experiments we used the European Corpus Initiative Multilingual Corpus I (ECI/MC1),[1] which is one of the most widely used corpora in language identification studies (Armstrong-Warwick et al., 1994). The ECI/MC1 corpus contains almost 100 million words in 27 (mainly European) languages. The corpus is formed of documents from several genres such as newspaper texts, novels, scientific papers, transcribed speech, legal texts, and dictionary entries.

We performed the experiments on nine languages in this corpus. The languages and the number of characters used for each language are as follows: Dutch (DUT-291 K), English (ENG-108 K), French (FRE-108 K), German (GER-171 K), Italian (ITA-99 K), Portuguese (POR-107 K), Spanish (SPA-107 K), Swedish (SWE-91 K), and Turkish (TUR-109 K). For each language, we used 90% of the data for training and 10% for testing, and we employed 10-fold cross validation in all the experiments.

The feature set consists of all the letters (Latin letters, extra letters, diacritics, and ligatures) in the alphabets of the nine languages. There are 54 features in the feature set which are as follows: a,à,á,â,ã,ä,å,b,c,ç,d,e,è,é,ê,ë,f,g,ğ,h,ı,i,ì,í,ï,j,k,l,m,n,ñ,o,ò,ó,ô,õ, ö,p,q,r,s,ş,t,u,ù,ú,û,ü,v,w,x,y,ß,z. We refer to the 26 letters of the English alphabet as common letters since each of them is used in a majority of the other languages. The other letters are referred as language-specific letters. As in other classification tasks, the success of language identification methods depends on the size of the test data. In order to determine the performance of the methods on data of varying sizes, we divided the test data into groups of different lengths ranging from 12 characters to 500 characters. For each length parameter, data of the appropriate size were taken randomly from the test data portion of the corpus.

For evaluating the success rates of the methods, we used the commonly used precision, recall, and f-measure metrics. For a language (class) l, let $correct_l$ and $predicted_l$ denote the set of test documents, respectively, that actually belong to the language l and that are classified as belonging to the language l by the method. Then the success rates are calculated as follows:

$$precision_l = \frac{|correct_l \cap predicted_l|}{|predicted_l|}$$

$$recall_l = \frac{|correct_l \cap predicted_l|}{|correct_l|}$$

$$f - measure_l = 2\frac{precision_l * recall_l}{precision_l + recall_l}$$

5.2. Comparison of ICF with centroid-based methods

In order to observe the performance of the proposed method in language identification, we compared it with two other centroid updating methods, AAC and CFC. The results are shown in Table 2. The values in the table are averages of the results obtained for the nine languages. The test data were formed of 100-byte (100 characters) long documents taken randomly for each language.

When we also analyze the results on a language basis, we observe that ICF outperforms AAC and CFC in nearly all of the languages. The AAC method has a performance close to the performance of ICF for English, Swedish, and Turkish. CFC also seems successful for Turkish, but it does not yield high results for other languages. For English, the success of the proposed method is similar to the success ratios of the other methods. This is due to the fact that the English alphabet includes only the Latin letters which

Table 2
Average success rates of centroid-based methods (100-byte long test data).

Method	Precision	Recall	F-measure
AAC	0.890	0.887	0.889
CFC	0.701	0.645	0.672
ICF	0.971	0.975	0.973

[1] http://www.elsnet.org/eci.html. The corpus is being distributed by ELSNET and can be ordered as CD-ROM.

Table 3
Comparison of the proposed method with related work.

Method	Training size	Test size	Result (%)
Dunning (1994)	50 K	20 bytes	92.00
Bayesian classifier	50 K	500 bytes	99.90
	5 K	500 bytes	97.00
Grefenstette (1995)	–	100 bytes (trigrams)	98.96
N-grams and short-terms		100 bytes (short-terms)	98.68
Adams and Resnik (1997)	220 K	100–500 bytes (trigrams)	98.32
Character *n*-grams	220 K	100-500 bytes (five-grams)	98.68
Prager (2000)	–	100 bytes (bigrams)	93.50
Vector space model over *n*-grams		100 bytes (trigrams)	97.70
		100 bytes (four-grams)	98.20
Xafopoulos et al. (2004)	-	140 bytes	99.00
Hidden Markov model			
Takçı and Soğukpınar (2004)	500 K	100 bytes	99.00
Centroid-based classification			
Kruengkrai et al. (2005)	–	50 bytes (kernelized centroid)	95.90
Kernelized centroid and support vector machines		50 bytes (support vector machine)	99.70
Vojtek and Bielikova (2007)	25–200 K	50 bytes (Dunning (1994))	97.95
Markov models of Dunning (1994) and Teahan and Harper (2001)		50 bytes (Teahan and Harper (2001))	98.20
Gottron and Lipka (2010)		45.1 bytes (Naïve Bayes)	98.52
Naïve Bayes, multinomial, Markov model, frequency rank, and vector space		45.1 bytes (Multinomial)	97.63
		45.1 bytes (Markov model)	73.13
		45.1 bytes (Frequency rank)	59.93
		45.1 bytes (Vector space)	61.04
Our method	50 K	100 bytes	98.00

serve as a default alphabet for other languages. However, the success of ICF largely depends on the special characters in the languages. ICF shows a significant improvement over other methods especially for languages in which language-specific characters occur frequently.

Although CFC is a more sophisticated approach than AAC, it yields worse results, especially for recall. This is probably due to the nature of CFC that gives too much importance to rare terms and biases against frequent terms. Thus, by decreasing the importance of some frequent but discriminative terms, it may miss the correct classes of some documents when some of the important (rare) terms for a class do not occur in the test document. The CFC method gives successful results in subject-based text classification (e.g. Guan et al., 2009), but its success falls behind the other methods in the domain of language identification. It seems that CFC is a more suitable approach when the documents to be classified are long documents.

5.3. Comparison of ICF with previous work

In this section, we compare the centroid-based classification method proposed in this study with previous studies that yielded successful results in language identification. We included in the comparisons different types of classifiers (Bayesian, HMM, etc.) in order to observe the performance of the proposed methodology with respect to other approaches. The results are shown in Table 3. The performances shown in the table are averages of the success rates of the languages used in the experiment.

The work of Dunning (1994) is one of the earliest statistical language identification studies. In this work, a Bayesian classifier was used. An accuracy rate of 92% was obtained with 50 K training data and 20 bytes test data. Increasing the test data size to 500 bytes improved the accuracies up to 99.9%. The high success rates in this early work were mainly due to incorporating only two languages in the system. The performances of the works based on the *n*-gram approach are around 97–98% for 100 bytes test data (for $n = 3–5$). Although the *n*-gram models give successful results for trigram-five-gram sequences, they employ a high dimensional feature set and yield much higher time complexities than the centroid-based models.

Xafopoulos et al. (2004) used an HMM classifier for identifying the language of web documents. They tested the method on five European languages and obtained 99% success rates on 140 bytes texts. A centroid-based model that uses a small feature set was proposed by Takçı and Soğukpınar (2004). The method was applied to four languages and a success rate of 99% was obtained. Kruengkrai et al. (2005) proposed a language identification approach based on a concept named as string kernels. They used a kernelized centroid method and support vector machine which yielded 95.90% and 99.70% success rates, respectively, for 50-byte long data.

In another work, the authors focused on the use of the Markovian approach and compared two notable Markov models (Vojtek and Bielikova, 2007). Experiments on the Reuters corpus that includes languages from several different families showed about 98% accuracies. Gottron and Lipka (2010) compared several different approaches for language identification on short texts in the form of queries. They used *n*-character long features in the feature set and tested the models with different n values. They observed that the naïve Bayes approach outperformed the others.

5.4. Comparison of ICF with short-term and n-gram methods

We compared the ICF method with the short-term and *n*-gram approaches used frequently in language identification studies. The short-term approach is a linguistic method. The short-terms are formed of words that occur frequently in a language such as conjunctions, prepositions and adverbs. We identified 400 short-terms that have the highest frequencies for each language and the feature set was formed of the combination of all the short-terms. In the case of the *n*-gram approach, we used $n = 3$ and identified 400 character trigrams with the highest frequencies for each language. Similarly, the set of all the trigrams formed the feature set. For the experiments, 100 documents were selected randomly for each language from the ECI/MC1 corpus. The tests were performed on 100-byte long documents.

Fig. 2 shows the success rates of the three methods for each language. The average success rates of all the languages are 96.85%, 95.44% and 95.88% for short-term, *n*-gram and ICF, respectively. We see that the short-term method outperforms the other

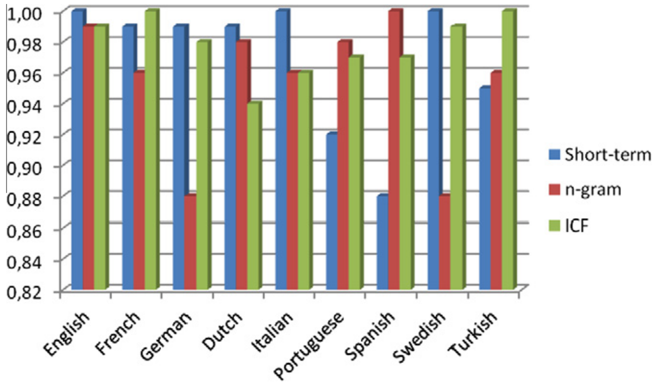

Fig. 2. Accuracies of language identification methods for each language.

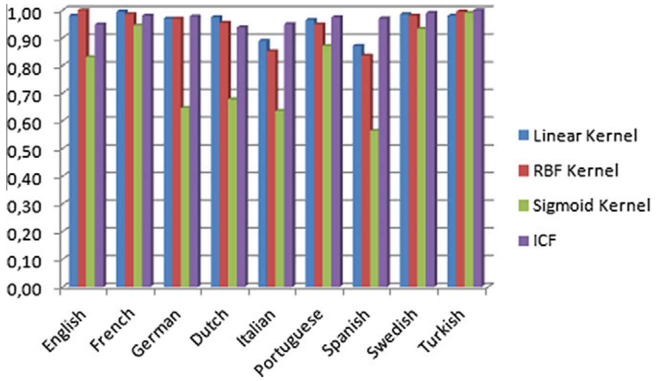

Fig. 3. Success rates (f-measure) of SVM methods and ICF method.

Table 4
Accuracies of MI, CHI and ICF methods (100-byte long test data).

Method	Accuracy
Mutual information	89.20
Chi-square	83.20
ICF	97.50

methods. This is probably due to the sizes of the feature sets and the test data. While 400 short-terms can cover most of the common terms used in a language, 400 character trigrams seem insufficient to represent a language. Although the performance of ICF is worse than the short-term method, the feature set consists of only 54 single characters, which is a much smaller feature set than the other models. In addition, the number of features is independent of the number of languages involved in the system. Thus, we can regard the proposed approach as a simple approach that yields high success rates and owns a low time complexity.

5.5. Comparison of ICF with SVM

We also compared the proposed method with SVM, which is one of the most successful approaches used in text classification and language identification. We used the SVM-light implementation with three kernel types: linear, radial basis function (RBF) and sigmoid. The experiments were conducted on the ECI/MC1 corpus for the same set of languages with 100-byte long test documents. The same feature set was used in the experiments.

The results in terms of f-measure are shown in Fig. 3. It can be seen that for most of the languages ICF outperforms all the SVM implementations. It is followed by the linear and RBF SVMs, and the sigmoid SVM gives the worst results. The experiments indicate that ICF, which is a quite simple and efficient approach, shows better behavior than one of the state-of-the-art kernel methods.

5.6. Comparison of ICF with mutual information and chi-square methods

The centroid-based classification approach proposed in this paper is a term weighting method that can be employed in several text classification problems. In this respect, we compare the ICF method with the mutual information (MI) and the chi-square (CHI) term weighting schemes that are widely used in the text classification domain. We use the standard formulas for these two metrics as given by Sebastiani (2002).

We repeated the experiments using the MI and CHI metrics on the same training and (100-byte long) test data with the same feature set. The results are shown in Table 4. The results indicate that ICF significantly outperforms the other two methods and is a more suitable approach for character-based language identification.

5.7. Analysis of results

The results of the experiments in the previous sections indicate that the centroid-based classifier that makes use of inverse class frequencies gives in general more successful results than the other classifiers for language identification. An important property of the method is its low (training and testing) time and space complexities. Having *k* languages and *m* features (terms), the time complexity is in the order $O(km)$. We used a very low value for the feature vector size ($m = 54$). Thus, the documents can be processed, the centroids can be formed, and the similarities can be calculated very efficiently.

We also measured the training and testing (classification) time complexities of the proposed method empirically and compared it with the other methods discussed previously. The executions were performed on a desktop PC with Intel Core 2 Quad CPU, 2 GB RAM and 32-bit operating system. We measured the classification time as the total execution time for a set of 100 test documents where each document is 100-byte long. The classification times for the short-term, *n*-gram, SVM and ICF methods were obtained as 2.44 secs, 71.88 secs, 4.28 secs and 0.85 secs, respectively. For the training times, we used a training set of size 900 K formed of approximately 100 K training data for each language. The training times were obtained as 1300.09 secs, 2162.22 secs, 428.50 secs and 78.38 secs, respectively. Having a linear time complexity and operating on a small feature set, the ICF method gives the best execution times. The short-term and *n*-gram methods need more training and testing time since they use larger feature vectors. Although SVM uses the same feature set as the ICF method, it has a higher time complexity. Thus, we can state that the ICF approach yields more successful results and has lower time complexities than the other approaches for language identification.

6. Conclusions

Language-based text classification is an important text classification problem and language identification systems are used in the preprocessing stages of several other systems. In this paper, we proposed a novel approach for language identification that is based on centroid-based classification and that uses a low dimensional feature space formed of individual characters. Although centroid-based classification is a fast classification approach, its success is in general lower than those of other classifiers. In order to increase its performance, we employed a method called inverse class frequency that makes an update on the centroid vectors. The results obtained on the ECI/MC1 multilingual corpus were compared with other methods and previous studies. The results showed that the ICF method mostly yields better success rates than the other methods and operates in less time.

A possible future work is applying the proposed approach to other text classification tasks in addition to language identification, where high performance classification is an important factor for the success. We can cite spam e-mail detection, categorization of texts in newsgroups, and identification of similar contents in social networks as a few examples of such problems. The centroid-based method proposed in this work can also be applied to these areas as an efficient classification approach.

References

Adams, G., Resnik, P., 1997. A language identification application built on the java client/server platform. In: Proc. ACL/EACL, Madrid, pp.43–47.

Amine, A., Elberrichi, Z., Simonet, M., 2010. Automatic language identification: an alternative unsupervised approach using a new hybrid algorithm. Int. J. Comput. Sci. Appl. 7 (1), 94–107.

Armstrong-Warwick, S., Thompson, H.S., McKelvie, D., Petitpierre, D., 1994. Data in your language: the ECI multilingual corpus I. In: Proc. International Workshop on Sharable Natural Language Resources, Japan, pp.97–106.

Bhargava, A., Kondrak, G., 2010. Language identification of names with SVMs. In: Proc. HLT-NAACL, pp. 693–696.

Bosca, A., Dini, L., 2010. Language identification strategies for cross language information retrieval. In: Proc. CLEF.

Buckley, C., Singhal, A., Mitra, M., 1996. New retrieval approaches using SMART. In: Proc. 4thText Retrieval Conference (TREC), Gaithersburg.

Cavnar, W.B., Trenkle, J.M., 1994. N-gram-based text categorization. In: Proc. 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR), Las Vegas, pp. 161–175.

Chizi, B., Rokach, L., Maimon, O., 2009. A survey of feature selection techniques. Encyclopedia of Data Warehousing and Mining, p. 1888–1895.

Chuang, W.T., Tiyyagura, A., Yang, J., Giuffrida, G., 2000. A fast algorithm for hierarchical text classification. In: Proc. International Conference on Data Warehousing and Knowledge Discovery, p.409–418.

Combrinck, H.P., Botha, E.C., 1995. Text-based automatic language identification. In: Proc. 6th Annual South African Workshop on Pattern Recognition, Gauteng.

Debole, F., Sebastiani, F., 2003. Supervised term weighting for automated text categorization. In: Proceedings of SAC. ACM, pp. 784–788.

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

Gottron, T., Lipka, N., 2010. A comparison of language identification approaches on short, query-style texts. In: Proc. ECIR, p. 611–614.

Grefenstette, G., 1995. Comparing two language identification schemes. In: Proc. 3rd International Conference on Statistical Analysis of Textual Data.

Grothe, L., Luca, E.W.d., Nürnberger, A., 2008. A comparative study on language identification methods. In: Proc. LREC, 2008.

Guan, H., Zhou, J., Guo, M., 2009. A class-feature-centroid classifier for text categorization. In: Proc. WWW, Madrid.

Hakkinen, J., Tian, J., 2002. N-gram and decision tree based language idetntification for written words. In: Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, Trento, pp. 335–338.

Han, E.-H., Karypis, G., 2000. Centroid-based document classification: analysis and experimental results. Principles of Data Mining and Knowledge Discovery, pp. 424–431.

How, B.C., Kulathuramaiyer, N., 2004. An empirical study of feature selection for text categorization based on term weightage. In: Proc. International Conference on Web Intelligence, p.599–602.

Keim, D.A., Oelke, D., Rohrdantz, C., 2009. analyzing document collections via context-aware term extraction. In: Proc. NLDB, Germany.

Ko, Y., Seo, J., 2004. Learning with unlabeled data for text categorization using a bootstrapping and a feature projection technique. In: Proc. ACL, p. 255–262.

Kruengkrai, C., Srichaivattana, P., Sornlertlamvanich, V., Isahara, H., 2005. Language identification based on string kernels. In: Proc. 5th International Symposium on Communications and Information Technologies (ISCIT), Beijing, pp. 896–899.

Leopold, E., Kindermann, J., 2002. Text categorization with support vector machines: how to represent texts in input space? Mach. Learn. 46 (1–3), 423–444.

Lertnattee, V., Theeramunkong, T., 2006. Class normalization in centroid-based text categorization. Inf. Sci. 176 (12), 1712–1738.

Miao, Y., Qiu, X., 2009. Hierarchical centroid-based classifier for large scale text classification. In: Proc. LSHTC.

Min, K., 2003. Related factors of document classification performance in a highly inflectional language. In: Liu, J. et al., (Eds.), Proc. IDEAL, pp. 645–652.

Murthy, K.N., Kumar, G.B., 2006. Language identification from small text samples. J. Quantitative Linguistics 13 (1), 57–80.

Ng, C.-C., Selamat, A., 2009. Improved letter weighting feature selection on arabic script language identification. In: Proc. ACIIDS, pp. 150–154.

Özgür, L., Güngör, T., 2010. Text classification with the support of pruned dependency patterns. Pattern Recognition Lett. 31, 1598–1607.

Prager, J.M., 2000. Linguini: language identification for multilingual documents. J. Manage. Inform. Syst. 16 (3), 71–102.

Reed, J.W., Jiao, Y., Potok, T.E., Klump, B.A., Elmore, M.T., Hurson, A.R., 2006. TF-ICF: A new term weighting scheme for clustering dynamic data streams. In: Proc. Internat. Conf. on Machine Learning Applications (ICMLA), Orlando, p. 258–263.

Robertson, S.E., Walker, S., 2000. Okapi/Keenbow at TREC-8. In: Voorhees, E.M., Harmann, D.K. (Eds.), Gaithersburg Proc. 8th Text Retrieval Conf. (TREC).

Salton, G., Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. 24 (5), 513–523.

Salton, G., 1989. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison Wesley.

Sebastiani, F., 2002. Machine learning in automated text categorization. ACM Comput. Surv. 34 (1), 1–47.

Shin, K., Abraham, A., Han, SY., 2006. Enhanced centroid-based classification technique by filtering outliers. Lecture Notes in Artificial Intelligence, Springer-Verlag, 4188, pp. 159–163.

Sibun, P., Reynar J.C., 1996. Language identification: examining the issues. In: Proc. 5th Symposium on Document Analysis and Information Retrieval, Las Vegas, p. 125–135.

Soonthornphisaj, N., Chaikulseriwat, K., Tang-On, P., 2002. Anti-spam filtering: a centroid-based classification approach. In: Proc. ICSP, pp. 1096–1099.

Suzuki, I., Mikami, Y., Ohsato, A., Chubachi, Y., 2002. A language and character set determination method based on n-gram statistics. ACM Trans. Asian Lang. Inf. Process. 1 (3), 269–278.

Takçı, H., Soğukpınar, İ., 2004. Centroid-based language identification using letter feature set. In: Proc. CicLing, pp. 635–645, Springer-Verlag.

Tan, S., 2007. Large margin dragpushing strategy for centroid text categorization. Expert Syst. Appl. 33 (1), 215–220.

Tan, S., 2008. An improved centroid classifier for text categorization. Expert Syst. Appl. 35 (1–2), 279–285.

Teahan, W.J., Harper, D.J., 2001. Combining PPM models using a text mining approach. In: Proc. Data Compression Conf. (DCC), Washington, pp. 153–162.

Tian, J., Suontausta, J., 2003. Scalable neural network based language identification from written text. In: Proc. IEEE Internationl Conference on Acoustic, Speech and Signal Processing, Hong Kong, pp.48–51.

Vatanen, T., Väyrynen, J.J., Virpioja, S., 2010. Language identification of short text segments with n-gram models. In: Proc. LREC.

Vojtek, P., Bielikova, M., 2007. Comparing natural language identification methods based on Markov processes. In: Proc. Slovko, International Seminar on Computer Treatment of Slavic and East European Languages, pp. 271–282.

Wang, S., Zhu, J., 2007. Improved centroids estimation for the nearest shrunken centroid classifier. Bioinformatics 23 (8), 972–979.

Winkelmolen, F., Mascardi, V., 2011. Statistical language identification of short texts. In: Proc. ICAART, pp. 498–503.

Xafopoulos, A., Kotropoulos, C., Almpanidis, G., Pitas, I., 2004. Language identification in web documents using discrete HMMs. Pattern Recognition 37 (3), 583–594.

Zhai, L-F., Siu, M-H., Yang, X., Gish, H., 2006. Discriminatively trained language models using support vector machines for language identification. In: Proc. Speaker and Language Recognition Workshop, San Juan, pp.1–6.