Under consideration for publication in Knowledge and Information Systems

Language Identification of Multilingual Posts from Twitter: A case Study

Ferran Pla and Lluís-F. Hurtado

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí Vera s/n 46022 València (Spain)
{fpla|lhurtado}@dsic.upv.es

Abstract. This paper describes a method for handling multi-class and multi-label classification problems based on the Support Vector Machine formalism. This method has been applied to the Language Identification problem in Twitter. The system evaluation was performed mainly on a Twitter dataset developed in the TweetLID workshop. This dataset contains bilingual tweets written in the most commonly used Iberian Languages (i.e., Spanish, Portuguese, Catalan, Basque, and Galician) as well as the English language.
We address the following problems: 1) Social media texts. We propose a suitable tokenization that processes the peculiarities of Twitter; 2) Multilingual tweets. Since a tweet can belong to more than one language, we need to use a multi-class and multi-label classifier; 3) Similar languages. We study the main confusions among similar languages; 4) Unbalanced classes. We propose threshold-based strategy to favor classes with less data. We have also studied the use of Wikipedia and the addition of new tweets in order to increase the training dataset.
Additionally, we have tested our system on Bergsma corpus, a collection of tweets in nine languages, focusing on confusable languages using the Cyrillic, Arabic, and Devanagari alphabets. To our knowledge, we obtained the best results published on the TweetLID dataset and results that are in line with the best results published on Bergsma dataset.

Keyword: Natural Language Processing, Language Identification, Multi-label Classification, Support Vector Machines, Twitter.

Received xxx
Revised xxx
Accepted xxx

1. Introduction

Twitter has become one of the most widely used platforms in which users express opinions about many subjects. This justifies the great interest in the automatic processing of this information in order to extract different content such as opinions, hobbies, political trends, reputation, etc. in real-time (Liu, 2012; Rao, Yarowsky, Shreevats and Gupta, 2010; Pla and Hurtado, 2014).

One important problem to solve in these applications is to filter the huge amount of data retrieved from Twitter in order to discard the information that is not of interest. Twitter supplies some tools (APIs) that allow us to filter information based on the user, the date, the geographical location, some content such as hashtags, specific words, etc. Nevertheless, for some applications, these tools are not enough and we need to apply more specific filters on the retrieved tweets if we are interested in more specific information.

One example of filters of this kind is the identification of the language of a text. Language Identification (LID) is a crucial problem to solve when we plan to do a thorough analysis of texts and we want to apply appropriate Natural Language Processing (NLP) techniques and linguistic resources for a specific language. Even though Twitter provides information about the language of the tweets, for our purposes, this information is not accurate enough, and moreover, it does not supply language information for tweets that belong to some of the languages studied in this paper.

LID has been well-studied (Cavnar and Trenkle, 1994) and it achieves very good results for normative and long texts. However, tweets are short texts of a maximum of 140 characters and the kind of language used in them does not have any restriction on the form or content. Therefore, the LID task is more difficult in Twitter. Also, for the kind of text we study in this paper, in some cases, the tweets are written with ungrammatical sentences with a lot of emoticons, abbreviations, specific terminology, slang, etc. In addition, one characteristic that makes the problem more difficult is the fact that a tweet can be written in more than one language (multilingual tweets). Finally, another aspect that can make the LID task difficult is to consider similar languages, that is, languages with a similar origin that share some words and grammatical constructions.

In this work, we address the LID problem in Twitter by considering the difficulties mentioned above. We used the corpus defined at the TweetLID[1] workshop (Zubiaga, Vicente, Gamallo, Campos, Loinaz, Aranberri, Ezeiza and Fresno-Fernández, 2014). The corpus contains tweets that are written in one of the five most commonly used languages of the Iberian Peninsula (tweets in English were also included in the corpus). Four of these languages (Spanish, Portuguese, Catalan, and Galician) are Romance languages and the Basque language is a very different language with different theories about its origin. These languages are spoken in different bilingual regions of the Iberian Peninsula. This corpus includes multilingual tweets from regions in which the users sometimes mix words from the different official languages of their region in a single tweet.

Due to the characteristics of the task that is addressed in this paper, we need to propose a solution for the following: 1) Social media texts. Specially posts from Twitter. We need to propose a suitable tokenization of tweets that correctly processes the peculiarities of Twitter (e.g, hashtags, user mentions, retweets, slag,

[1] http://komunitatea.elhuyar.org/tweetlid/

Table 1. Difficult tweets in the corpus.

Twitter peculiarities	**Examples**
en	Hii!!!! :-) @MariaEscot pic.twitter.com/as78df
Similar languages	**Examples (Hello, how are you?)**
es	Hola, cómo estás?
ct	Hola, com estàs?
gl	Ola, como vai?
pt	Olá, como vai?
Multilingual tweets	**Examples**
es+**en**	Vamos a echar un partido de Fifa contra **my brother** :)
ca+**en**	@ilove ja no tindré examens, **can we meet up and watch it together plsss???**
Ambiguous tweets	**Examples**
ca/es	La vida es un carnaval!!!!

emoticons, etc.); 2) Multilingual tweets. Since a tweet can belong to more than one language, we need to use a multi-class and multi-label classifier; 3) Similar languages. We studied the main confusions among similar languages, mainly by exploring the confusion matrix of our system for similar languages; 4) Unbalanced classes. We propose a correction method to increase the accuracy of the classifiers for classes with less data in the corpus by using thresholding strategies to favor minority classes. We also studied the use of external resources such as Wikipedia, or the inclusion of additional tweets that are automatically classified by our system, this is presented in the experimental work section (Section 7).

Table 1 shows some examples of difficult tweets for the task that is considered in this paper. The first one corresponds to tweets that contain information that is irrelevant for determining the language, such as URLs, references to pictures, user mentions, emoticons. It only contains one useful word (Hiii!!!), but in this case with repeated characters. The second example, "Hello, how are you?" is very similar in the Romance languages that are present in the corpus: Spanish (es), Catalan (ct), Galician (gl), and Portuguese (pt). The third group contains examples of multilingual tweets: tweets that merge Spanish and English texts and Catalan and Spanish texts, respectively. The last example is an ambiguous tweet because the sentence, "La vida es un carnaval" (Life is a carnival), is written the same way in both Spanish and Catalan.

To address these problems, we have proposed an approach that uses the Support Vector Machine (SVM) formalism to solve the LID problem and we evaluated it on the TweetLID copus. We also performed a comparison with the published results using this data set.

The rest of this paper is organized as follows. Section 2 presents some works related to LID. In Section 3, we describe the LID task. In Section 4, we formalize the multi-class and muli-label approach to the LID task. In Section 5, we define the metric to be used for evaluating the system proposed in this work. Section 6 presents a short description of the system developed. Section 7 presents the experimental work conducted in this paper. In Section 8 we present the evaluation

of our system in a very different data set. Finally, in Section 9, we present some conclusions and possible directions for the future work.

2. Related Works

Text Categorization (TC) is a well-studied problem for classifying textual documents into categories (Joachims, 1998; Sebastiani, 2002). The goal of TC is the classification of documents into a fixed number of predefined categories. In general, each document can be classified into one or several categories. For this reason, TC can be formalized as a multi-class and multi-label classification problem. An overview of different approaches and strategies to handle this problem can be seen in (Tsoumakas and Katakis, 2007).

Language Identification can be considered as a particular case of the Text Categorization problem in which we must determine the language or languages that appear in a given text from a list of candidate languages. Since a tweet can be written in more than one language, we used a multi-label classification approach in the TweetLID task.

One of the most influential and widely used approach for both TC and LID is the work of (Cavnar and Trenkle, 1994). Their method consisted in learning an n-gram model of characters for every language considered in the collection of texts. Given an input text, the classification consisted in assigning the language with the minimum distance to all of the learned models. They reported hight accuracy, around 99.8% for long and well-written documents.

Some works perform studies to test different sets of features or different machine learning approaches. The work presented in (Grefenstette, 1995) compares two systems: one based on letter trigrams and the other based on common short words. Their experimental work, which was performed on a set of 10 languages, shows that the best results for short texts was obtained by using the trigram approach. Baldwin and Lui (2010) compared different tokenization strategies and different machine-learning models such as Nearest-Neighbour (NN), Naive Bayes, and Support Vector Machines (SVM). The best results were obtained by using 1-NN model or SVM with a linear kernel considering bigrams or trigrams of characters to represent the Wikipedia documents. They also reported that accuracy depends on the length of Wikipedia documents. The accuracy decreased from 90% for long texts to 60-70% for short texts.

In recent years, different LID works on microblogging texts, and specially on Twitter, have demonstrated the difficulty of texts of this kind. The work of Carter, Weerkamp and Tsagkias (2013) reported that accuracy for LID decreases on average, from 99.4% on formal texts (EuroParl) to 92.4% on microbloging texts (tweets in five European languages). Their approach is based on a character n-gram distance metric. Additionally, they take into account five microblogging characteristics to enrich the textual content of tweets: the content of the links of the tweet, the author, hashtags, mentions, and replies. Goldszmidt, Najork and Paparizos (2013) used a boot-strapping mechanism to adapt a system to Twitter posts. Initially, they constructed a LID systems that was trained on Wikipedia texts. Then, they used the location information from tweets to retrieve tweets from a certain language in order to improve the initial model. Bergsma, McNamee, Bagdouri, Fink and Wilson (2012) present a system that combines n-gram information and metadata from Twitter. They also perform an experimental comparison with the best available LID systems. Lui and Baldwin

(2014) proposed a method for building Twitter datasets by minimizing the manual annotation task. They tested existing LID techniques over Twitter messages and combined them by using simple voting techniques to outperform individual systems.

Recently, there has also been growing interest in the study of multilingual tweets. Prager (1999) presented Linguini, which is a vector-space-based categorizer that uses n-grams of characters and words as features. He used this system to identify the languages in bilingual and trilingual documents. Lui, Lau and Baldwin (2014) introduced a method for detecting multilingual documents, identifying the languages present, and estimating their relative proportion. This method uses a generative mixture model for performing multi-label classification. Nguyen and Dogruoz (2014) treated the Language identification task at the word level as a sequence labeling problem using a Conditional Random Fields approach that adds the previous and the next token to each word as context. Jauhiainen, Lindén and Jauhiainen (2015) proposed a method whose main idea is to slide an overlapping window of a fixed length through the document. The text in each window is classified in the most likely language. Finally, the document is classified with the different languages that are found in all of the windows considered.

Another aspect that is related to the characteristics of the task addressed in this work is the problem of identifying similar languages. Ljubešić, Mikelić and Boras (2007) presented a study about similar languages such as Slovenian, Serbian, Slovak, and Croatian that shows that introducing some heuristics about the most frequent words in these languages increases the performance of the LID system. Bergsma et al. (2012) also developed a collection of tweets in nine languages focusing on confusable languages using the Cyrillic, Arabic, and Devanagari alphabets.

An overview of the approaches used by the participants at the TweetLID competition can be found in (Zubiaga et al., 2014). Different machine-learning approaches such as SVM and Naive Bayes were used by the participants. All the participants agreed on the importance of suitable tokenization and a technique to properly clean tweets by removing specials tokens such as mentions, user references, retweets, or irrelevant text for language identifications (e.g., web addresses, emoticons, etc.). In the experimental work section (Section 7), we discuss the results of teams participating at the TweetLID workshop.

3. LID Task Description

The aim of this task consists in identifying the language or languages in which the tweets are written. This task was proposed at the TweetLID workshop (Zubiaga et al., 2014) focusing on the most commonly used languages of the Iberian Peninsula (Spanish (es), Portuguese (pt), Catalan (ct), Basque (eu), and Galician (gl)) as well as the English (en) language.

Figure 1 shows the geographical distribution of languages in the Iberian Peninsula. Three officially bilingual regions were considered: Catalonia and the Valencian Community (ct+es), the Basque Country (eu+es), and Galicia (gl+es). The official languages for these three regions are likely to co-occur along with news and events that are relevant to the Iberian Peninsula in the bilingual regions. The rest of the Iberian peninsula is considered to be monolingual (i.e., the rest of the Spain (es) and Portugal (pt)).

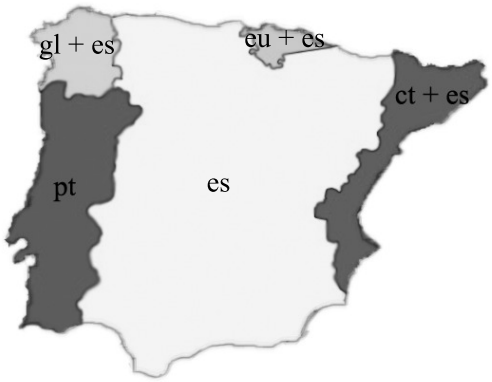

Figure 1. Languages of the Iberian Peninsula.

Table 2. Distribution of tweets per language in the training and test sets.

Language		Training		Test	
1 language	ca	1409	(10.12%)	1426	(7,77%)
	es	7916	(56.87%)	11752	(64.03%)
	en	940	(6.75%)	910	(4.96%)
	eu	363	(2.61%)	358	(1.95%)
	gl	487	(3.50%)	423	(2.30%)
	pt	1933	(13.89%)	1929	(10.51%)
# monolingual tweets		13048	(93.74%)	16798	(91.53%)
2 languages		346	(2.49%)	347	(1.89%)
3 languages		8	(0.06%)	7	(0.04%)
ambiguous		323	(2.32%)	261	(1.42%)
undefined (und)		174	(1.25%)	555	(3.02%)
other		20	(0.14%)	385	(2.10%)
# tweets		13919	(100%)	18353	(100%)

The TweetLID organization supplied a corpus of tweets that was collected within the Iberian Peninsula and manually annotated with the correct language or languages (Zubiaga et al., 2014).

Table 2 shows the distribution of tweets per language in the training and test set, respectively. The set of labels considered in the corpus is: *eu* for Basque, *ca* for Catalan, *ga* for Galician, *es* for Spanish, *pt* for Portuguese, and *en* for English. Additionally, some tweets are annotated with the *other* label indicating that they are written in a language that is not considered in the task (i.e, German, French, etc.). Tweets that did not have enough information to determine the language are annotated with the *und* (undefined) label. Finally, some tweets are *ambiguous*, that is, tweets that may have been written the same way in some languages.

For evaluation purposes, the organization decided to merge the tweets that are annotated as *other* and *und* in a single label.

The TweetLID organization proposed two task: the *constrained* task, in which you can only use the supplied training partition for learning the models; and the *unconstrained* task, in which other resources can be used in the learning phase. Both tasks must be evaluated on the test part of the corpus. Therefore, to properly deal with the classification of multilingual tweets we needed to develop a multi-label classifier like the one that we formulate in Section 4. In addition, due to the unbalanced languages of the corpus, we propose a thresholding strategy to mitigate this problem and to favor the classes with less data.

4. LID Formulation as a Multi-Class and Multi-Label Problem

Following the notation used in (Tsoumakas and Katakis, 2007) and (Ramón Quevedo, Luaces and Bahamonde, 2012), we formalize the classification problem as the problem of learning a function $f : X \to Y$ from a data set of labeled samples D, where

- L is a finite set of disjoint labels $L = \{l_1, \cdots, l_{|L|}\}$, and $|L| > 1$
- X is an input space
- $\mathcal{P}(L)$ is the label power set of L
- Y is the set of considered labels ($Y \subseteq \mathcal{P}(L)$), and
- $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ is a data set of samples, where $x_i \in X$ and $y_i \in Y$

When $|L| = 2$, the problem is called binary classification, which is a well-studied problem in Text Classification. Since $|L| > 2$ for most practical problems, we need to extend the problem. This is usually done by combining binary classifiers in order to perform multi-class classification. In these cases, the combination is performed by using the well-known one-versus-rest or the one-versus-one strategies.

We assume that we have $|L| > 2$ classes and we need to learn $|L|$ classifiers. To do this, we uses a transformation method of the data set D that consists of the construction of $|L|$ sample sets, in which every set considers those samples that belong to the ith class l_i as positive samples and the rest of the samples belong to the $\neg l_i$ class. From these new data sets, we learn $|L|$ classifiers denoted as f_k, where $k : 1 \cdots |L|$. We denote $P_k(x_i)$ as the probability assigned by the classifier f_k to the input sample x_i. In order to classify a new sample x_i, we apply all the classifiers to it, and we select the classifier that maximizes Equation 1:

$$\hat{y}_i = \underset{k:1\cdots|L|}{\arg\max} P_k(x_i) \tag{1}$$

There are some problems (e.g., the Language Identification task) in which we need to associate more than one label to an example (multi-label classification). In this case, we have a set of labels $Y \subseteq \mathcal{P}(L)$ that can be assigned to the input sample x_i.

According to the solutions presented in (Tsoumakas and Katakis, 2007), we formalized multi-label classification as a transformation method by following two solutions called *label power set* and *binary relevance*. We selected the transfor-

mation strategy for this problem because it is more general and is independent of the type of classifier used.

– Following the *label power set approach*, we only consider as labels those combinations of labels from the set $\mathcal{P}(L)$ that have been seen in the training set. With this new set of labels, in our approach, we re-label the training set and we construct a multi-class classifier following the one-versus-all strategy. Due to the way we have defined the set Y, the classifier can assign a set of labels to an input sample x_i by maximizing Equation 2.

$$\hat{y}_i = \arg\max_{k:1\cdots|Y|} P_k(x_i), \text{where } y_i \in Y \tag{2}$$

– Following the *binary relevance approach*, we construct one classifier per class. To do this, we consider the samples in the training set that belong to a particular class l_i, and we assign the rest of the training set to the class $\neg l_i$. This is a one-versus-all strategy for multi-class problems. A *threshold* is used to determine those labels will be selected (the labels that are assigned a probability greater than the threshold by the binary classifier). In this approach, $Y = L = \{y_1, \cdots, y_{|L|}\}$. We defined a *threshold*, ϵ, and we applied all the classifiers learned (f_k, where $k : 1 \cdots |L|$) for an input sample x_i. We chose those labels l_i that satisfies the probability assigned to the classifier $f_k(x_i)$ is greater than ϵ, as shown in Equation 3.

$$y_i = \{l_k \in Y : P_k(x_i) > \epsilon, \forall k : 1 \cdots |L|\} \tag{3}$$

The *threshold* is usually set to 0.5, but other values of $\epsilon > 0.5$ or non-constant values of ϵ can also be considered (Ramón Quevedo et al., 2012). As we explain in the experimental work section (Section 7), we tested different values of ϵ that depend on the class, $\epsilon(k)$, in order to favor minority classes.

Now, we present an example of these transformations methods for a subset of languages of the TweetLID task.

Let L be a finite set of disjoint labels, for example, $L = \{es, ct, en\}$ which represents the set of Spanish (es), Catalan (ct), and English (en) languages. Let $D = \{(x_1, (es, ct)), (x_2, (en, es)), (x_3, es), (x_4, en), (x_5, ct)\}$ be a data set of samples.

– For the *label power set* approach, we consider the following set of labels Y in the data set: $Y = \{(es, ct), (en, es), es, en, ct\}$. We consider the following new data sets by applying the one-versus-rest strategy,

$$D_1 = \{(x_1, (es, ct)), (x_2, \neg(es, ct)), (x_3, \neg(es, ct)), (x_4, \neg(es, ct)), (x_5, \neg(es, ct))\}$$
$$D_2 = \{(x_1, \neg(en, es)), (x_2, (en, es)), (x_3, \neg(en, es)), (x_4, \neg(en, es)), (x_5, \neg(en, es))\}$$
$$D_3 = \{(x_1, \neg es), (x_2, \neg es), (x_3, es), (x_4, \neg es), (x_5, \neg es)\}$$
$$D_4 = \{(x_1, \neg en), (x_2, \neg en), (x_3, en), (x_4, en), (x_5, \neg en)\}$$
$$D_5 = \{(x_1, \neg ct), (x_2, \neg ct), (x_3, \neg ct), (x_4, \neg ct), (x_5, ct)\}$$

– For the *binary relevant* approach, the set of labels considered is: $Y = \{es, ct, en\}$, and we construct the following new data sets:

$$D_1 = \{(x_1, es), (x_2, es), (x_3, es), (x_4, \neg es), (x_5, \neg es)\}$$
$$D_2 = \{(x_1, ct), (x_2, \neg ct), (x_3, \neg ct), (x_4, \neg cat), (x_5, ct)\}$$
$$D_3 = \{(x_1, \neg en), (x_2, en), (x_3, \neg en), (x_4, en), (x_5, \neg en)\}$$

The *binary relevance* approach has the advantage that is more general than the *label power set* approach because it can assign sets of labels that are not in the training set. However, the main drawback of this approach is that relationships between classes are lost.

5. Evaluation metric

For the experimental evaluation, we used the well-known *precision* (π), *recall* (ρ), and F_1 measures. Nevertheless, considering that this task is a multi-label text classification task, we consider two different ways of calculating these measures: *macroaveraging* and *microaveraging*. These two different types of measures can be described as follows (Sebastiani, 2002):

$$micro\_\pi = \frac{\sum_{i=1}^{|T|} TP_i}{\sum_{i=1}^{|T|} TP_i + \sum_{i=1}^{|T|} FP_i}$$

$$micro\_\rho = \frac{\sum_{i=1}^{|T|} TP_i}{\sum_{i=1}^{|T|} TP_i + \sum_{i=1}^{|T|} FN_i}$$

$$micro\_F1 = \frac{2 \sum_{i=1}^{|T|} TP_i}{2 \sum_{i=1}^{|T|} TP_i + \sum_{i=1}^{|T|} FN_i + \sum_{i=1}^{|T|} FP_i}$$

$$macro\_\pi = \frac{\sum_{i=1}^{|T|} \frac{TP_i}{TP_i + FP_i}}{|T|}$$

$$macro\_\rho = \frac{\sum_{i=1}^{|T|} \frac{TP_i}{TP_i + FN_i}}{|T|}$$

$$macro\_F1 = \frac{\sum_{i=1}^{|T|} \frac{2\,TP_i}{2\,TP_i + FN_i + FP_i}}{|T|}$$

In the measures previously defined, TP_i, FP_i, and FN_i represent *True Positives*, *False Positives*, and *False Negatives* for a certain class i, respectively.

From the *macroaveraging* point of view, all classes count the same when calculating the precision and recall of the system. From the *microaveraging* point of view, the classes count proportionally to the number of tweets of this class when calculating global precision and recall. Because our goal is to evaluate the performance of our system for each class, we chose the macro-F1 measure as the evaluation measure. In addition, this is the measure that was chosen by the TweetLID workshop organization to rank the participants. Therefore, we can compare our results with those obtained by the TweetLID participants.

6. System Design

We developed a system that uses the SVM formalism (Cortes and Vapnik, 1995) because of its ability to handle a large feature space and to determine the relevant features. We used a bag-of-words approach to represent each tweet as a feature vector that contains the td-idf factors of the selected features of the training set.

To optimize our system, we can distinguish among three important aspects

that should be taken: 1) Tokenization, 2) Feature selection, and 3) Optimization of the parameters of the model.

Tokenization is a very important preprocessing step in Twitter. As mentioned above, this is due to the nature of the language used in Twitter (ungrammatical sentences, absence of punctuation, specific terminology, slang, etc.). Although there are a lot of tokenizers available, they need to be adapted in order to address the segmentation of tweets. Furthermore, most of these resources are for the English language, which adds a degree of difficulty for their use in processing tweets in other languages.

In the design of our system, we decided to use and adapt some tools that are available for tokenization. We adapted the package *Tweetmotif*[2] that is described in (O'Connor, Krieger and Ahn, 2010) to process tweets written in the set of languages considered in the task. We made some modifications in *Tweetmotif*. We modified some regular expressions to take into account Latin characters: accents (*á, é, í, ...*), specific letters(*ñ, ü, ...*), etc. We adapted the emoticon detector, and we added some functions to process special tokens (e.g., grouping all *web* directions into a single token). For the LID task, we converted all text to lowercase.

In the feature selection process, we used a 10-fold cross validation process that optimizes the performance of the system on the training set in terms of F1 measure. We considered a wide set of features such as the tokens extracted in the tokenization phase (basically words), the n-grams of these words (where n ranged from 1 to 4), and the n-grams of characters (where n ranged from 1 to 6). We also tested the use of external information resources (words and sentences from Wikipedia). In addition, we automatically downloaded and classified additional tweets to increase the data set. In the experimental work section (Section 7) we explain in more details the set of features used and the results achieved using them.

The parameters of the SVM were also optimized during the 10-fold cross validation phase. We also estimated a threshold for minority classes on the corpus in order to increase the performance of the system as we show in the experimental work section.

The system was implemented in Python using the *scikit-learn* package (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot and Duchesnay, 2011) and the LibLinear[3] external library. The developed system supports the mono-label (ML) classification approach and the multi-label approach, which are both label power set (LP) and binary relevance (BR) strategies.

7. Experimental Work

For the purpose of evaluating our different systems, we used the TweetLID official corpus described in Section 3. We developed up to 26 different systems. All of the systems were built using SVM with a linear kernel, but they differ from each other in three main characteristics: the multi-label strategy selected, the features used by the SVM, and the amount of corpus used (the TweetLID corpus only or

[2] https://github.com/brendano/tweetmotif.
[3] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

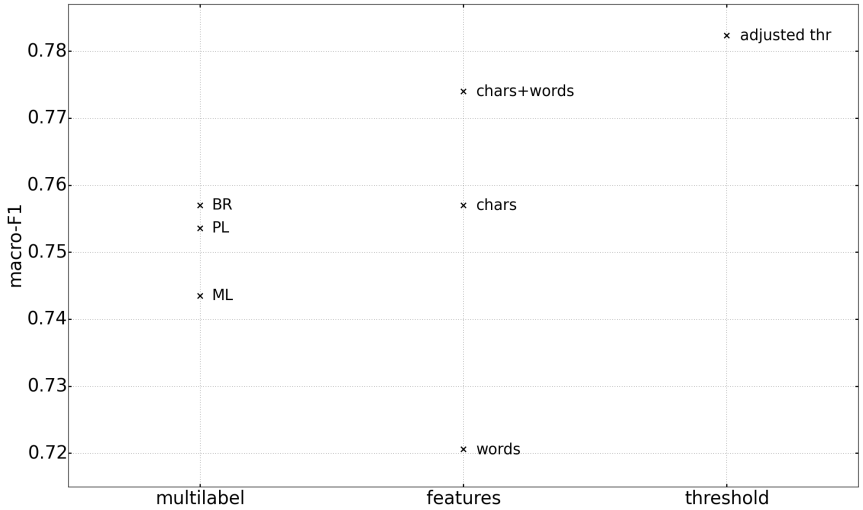

Figure 2. Macro-F1 of the *constrained* systems during the tuning process (average of the 10-fold cross-validation).

TweetLID plus additional corpora). The same tokenization described in Section 6 was used in all of the systems.

7.1. Tuning Process

In order to compare the behavior of the different systems, we used the macroaveraging F1 measure (*macro-F1*) because it was the official measure of the TweetLID competition. For each developed system, a 10-fold cross validation process was carried out to determine the best set of features and the best value of the linear SVM classifier c parameter. We developed several systems for both the *constrained* task and the *unconstrained* task.

7.1.1. Constrained task

In the *constrained* task only the supplied training partition could be used for learning the models. Figure 2 summarizes the results obtained by our *constrained* systems during the tuning process in terms of macro-F1 measure.

First, we conducted a study of the behavior of different strategies to address the multi-label problem. The results obtained can be seen in the **multi-label** column of Figure 2. Three systems were developed using the Mono-label approach (ML), the Label Power Set approach (LP), and the Binary Relevance approach (BR). All of the systems used n-gram of characters as features, and the cross validation process determined the best n and c for each system. The best result (0.757) was achieved by the system using the BR approach. Therefore, we used the Binary Relevance approach in the rest of the systems. In the BR approach,

we assume that if any class reaches its threshold, the *und* class is selected. This assumption was experimentally determined.

Second, three systems were developed in order to determine the best set of features to use (see the **features** column of Figure 2). The systems were: a system that uses n-grams of words (words) as features, a system that uses n-grams of chars (chars) as features, and a system that uses both n-gram of words and n-gram of chars (chars+words) as features. The cross validation process determined the best n for words and chars and the best c for each system. As expected, the worst results were obtained by the system using only n-grams of words. However, the addition of n-grams of words to the n-grams of characters considerably improved the results going from 0.757 to 0.774. Thereafter, we used n-grams of words together with n-grams of chars for the subsequent systems.

Third, we tried to improve the performance of the classes with less accuracy. With that in mind, we explored setting different thresholds per class ($\epsilon(k)$). The process for determining the best threshold for each class was as follows. Based on the best system in the previous step, we modified the threshold of just one class in each iteration by increasing or decreasing it. Those threshold values that improved the macro-F1 measure (using the cross validation process) were kept. Subsequently, all possible combinations of the stored individual thresholds were made. The combination of thresholds that increased the macro-F1 the most was selected. As shown in the **threshold** column of Figure 2, setting different thresholds per class greatly improved the system performance, achieving 0.782 of macro-F1 on the 10-fold cross validation using the training set. Only four thresholds were different from the default threshold: the threshold of *es*, which increased by 0.05, and the thresholds of *eu*, *gl*, and *pt*, which decreased by -0.2, -0.15, and -0.1, respectively. In other words, the class with the most samples is slightly penalized and three classes with fewer samples are favored.

7.1.2. Unconstrained task

In the *unconstrained* task other resources could be used in the learning phase. Figure 3 summarizes the results obtained by our *unconstrained* systems during the tuning process in terms of macro-F1 measure.

We addressed the *unconstrained* task using two different information sources: Wikipedia and tweets from specific users. The goal was to have words, sentences, and tweets for which we already know the language without having to label them by hand.

We explored the use of Wikipedia as a source of additional corpora in two different ways: adding the most frequent words and adding whole sentences. We built a set of dictionaries with the most frequent words obtained from the versions of Wikipedia for the languages considered in the TweetLID competition. We added a new feature to the SVM models for each dictionary built. Each feature was the number of words in the tweet that appear in the corresponding dictionary. We tested with different amount of words. The **wikiwords** column of Figure 3 shows the results obtained when the 1000, 5000, 10000, 20000, 30000, 40000, and 50000 most frequent words of each language were considered to build the dictionaries. It can be observed that there was an increase in the system behavior (0.7795), but without reaching the level obtained by the system with the adjusted thresholds (0.782, see Figure 2). There was also a slight decrease in the system performance over 40000 words. It should be noted that setting the same number of words for all languages may cause an imbalance in the quality

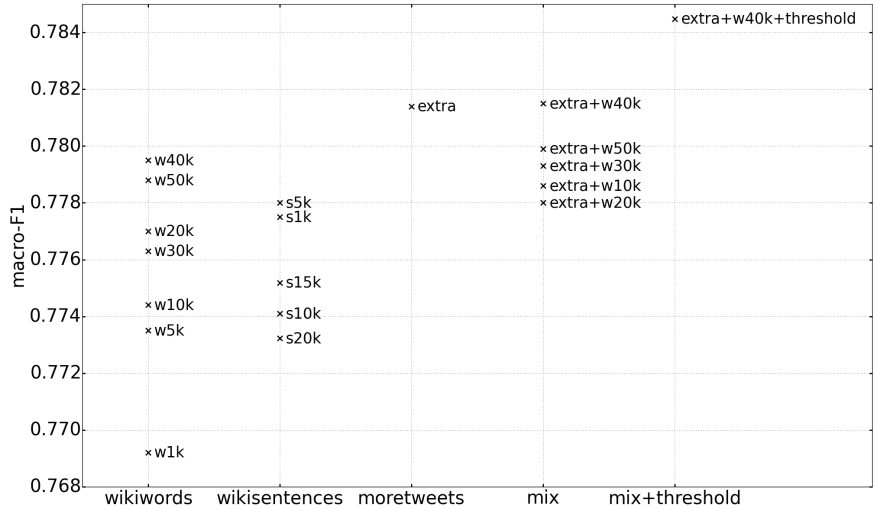

Figure 3. Macro-F1 of the *unconstrained* systems during the tuning process (average of the 10-fold cross-validation).

and the representativeness of the words considered in those minority languages. The English version of Wikipedia has more than 39 million different words, while in the Galician version (the smallest version of those considered), there are fewer than half a million words. There are over 80 times more words in the English version that in the Galician version.

The other way of using Wikipedia to increase the coverage of the model was to expand the corpus by directly adding complete sentences extracted from Wikipedia. The sentences were to meet a length criterion similar to the Twitter post. The length of the sentence was to be between 50 and 150 characters including spaces. The sentences were randomly selected from all of the articles of the different versions of Wikipedia as long as they met this length criterion. The **wikisentences** column of Figure 3 shows the results obtained when the 1000, 5000, 10000, 15000, and 20000 sentences per language were added to the corpus. Adding more than 5000 sentences of every language (30000 in total) did not help to improve the system. Moreover, 20000 sentences even worsened it (from 0.774 to 0.7732). This may be counterintuitive because it seems that the greater the corpus, the better the results. Perhaps the explanation is that the type of text is very different.

We also tried to increase the corpus by adding new tweets (we added 2221 extra tweets to the corpus, denoted as *extra* in Figure 3). Specifically, we added 1640 new tweets in Galician and 581 in Basque. To do this without having to label the tweets manually, we proceeded as follows. First, we selected Twitter user accounts that were well-known for their defense of the Galician or Basque culture. They were mostly cultural associations or nationalist parties. We got over 10000 tweets. We discarded the tweets that were retweets of other users because retweets are often in languages other than the language normally used

by the user. Finally, we added only the tweets that our own baseline system classified as written in Galician or Basque. We did not do a manual review of the tweets, but we are relatively confident that they have the characteristics we were looking for. As the **moretweets** column of Figure 3 shows, the system improved by adding these *extra* tweets (0.7814). However, this was not the great leap forward that we expected. The new tweets may not have had enough variability in topics or vocabulary.

In an attempt to take advantage of the potential complementarity of the models, we decided to join the wikiwords models and the *extra* tweets. Therefore, we created new models that included dictionaries of words from Wikipedia as features and also the 2221 *extra* tweets in Galician and Basque in the learning corpus. The **mix** column of Figure 3 shows the results of mixing wikiwords and extra tweets. Again, the best results were obtained when the 40000 most frequent words of each language were added to the dictionaries. However, the result was almost equal to that obtained by adding only the *extra* tweets (0.7814 instead of 0.7815). Furthermore, amounts different from 40000 words produced even worse results. It seems that the models were not so complementary after all.

The last system developed consisted of setting the thresholds on the best system previously obtained. From the *extra+w40k* system, we adjusted the thresholds of each class as described above. As before, only the thresholds of *es*, *eu*, *gl*, and *pt* were changed. The threshold of *es* was increased and the thresholds of the *eu*, *gl*, and *pt* were decreased. This last system achieved the best result of all of the developed systems (macro-F1 of 0.7845 in the 10-fold cross validation). It was the system that contained the most information and the one that had better tuning. The result (*extra+w40k+threshold*) can be found in the **mix+threshold** column of Figure 3.

7.2. Results Using the Test Set

Once all of the best developed systems using cross validation on the training set had been evaluated, we selected those systems with the best results in order to evaluate them using the test set of the TweetLID corpus. We selected eight systems for this evaluation (three for the *constrained* task and five for the *unconstrained* task).

We looked for previous results on the same corpus to compare the performance of the systems developed in this work. In this regard, there were two baselines provided by the TweetLID organization (Zubiaga et al., 2014): a) Twitter's metadata, where the language identification is performed using the language information provided by Twitter with each tweet; b) TextCat, a state-of-the-art n-gram-based language identification system developed for formal texts. Although for different reasons neither of the two baselines achieved a macro-F1 higher than 0.5, which indicates the difficulty of the task. We also considered the winner systems of the TweetLID competition as reference in both the *constrained* task and in the *unconstrained* task. The winner of the *constrained* task was a system developed by the authors of this work. Our system (Hurtado, Pla, Giménez and Arnal, 2014) was based on SVM and n-grams of characters but it used a tokenization without considering the specific characteristics of Twitter and a feature selection process that was not as exhaustive as the one carried out in this work. It could be considered as a preliminary version of the systems presented in this paper. The system winner of the *unconstrained* task, (Gamallo, García,

Table 3. Results obtained by the reference systems.

Reference system	*macro-F1*
Twitter's metadata baseline	0.463
TextCat baseline	0.447
Constrained TweetLID winner	0.752
Unconstrained TweetLID winner	0.753

Sotelo and Campos, 2014), was based on a state-of-the-art Bayesian algorithm. Gamallo et al. (2014) used a news corpora extracted from on-line journals to learn the model.

Table 3 shows *macroaveraging* F1 measure for the four reference systems. As can be consulted in the overview of the workshop (Zubiaga et al., 2014), 11 runs were submitted for the *constrained* task and 9 runs were submitted for the *unconstrained* task. For the *constrained* task, the macro-F1 values ranged from 0.752 to 0.498; for the *unconstrained* task the macro-F1 values ranged from 0.753 to 0.501. Note that except for the system of Gamallo et al. (2014), the rest of the participating systems achieved worse results for the *unconstrained* approach than for the *constrained* approach. This conclusion and other features and peculiarities of the participating systems at TweetLID competition can be seen in (Zubiaga, Arkaitz, naki San Vicente, Gamallo, Pichel, Alegria, naki, Aranberri, Ezeiza and Fresno, 2015).

In Figure 4 we summarize the results of our best systems both for the *constrained* and the *unconstrained* tasks that we will discuss in next Subsections. In order to verify if the improvements achieved by the developed systems were statistically significant, we have added to Figure 4 the confidence intervals for both macroaveraging precision ($Macro$-π) and macroaveraging recall ($Macro$-ρ) at 95% of confidence. Due to macroaveraging F-1 measure ($Macro$-$F1$) was chosen by the TweetLID workshop organization to rank the participants, we used this measure to compare the results of our different systems with the results obtained by the TweetLID participants.

7.2.1. Results of the Constrained Approaches on the Test Set

The systems selected for the *constrained* task were: i) *BR*, a system that uses the Binary Relevance approach and 5-grams of characters as features; ii) *chars+words*, a system that uses 3-grams of characters and 3-grams of words as features; iii) *chars+words+threshold*, the system that uses a combination of characters and words as features, but setting a separate threshold for each class, maximizing macro-F1 as described above. These systems were the best performers for the *constrained* task in the tuning phase (Figure 2). The features and parameters of the SVMs were obtained during the 10-fold cross-validation process.

For the *constrained* systems (the first three systems in Figure 4) the macroaveraging recall ($Macro$-ρ) of the *BR* system was the lowest of all the systems. The addition of n-grams of words as features in the *chars+words* system allows $Macro$-ρ and even the precision ($Macro$-π) to be increased. Finally, adjusting the threshold per class greatly improves the recall, at the expense of a slight

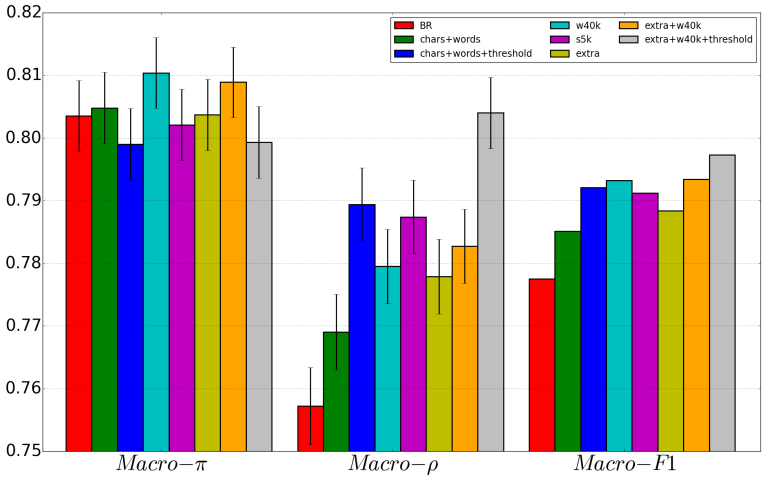

Figure 4. Results of the best systems on the test set for the macroaveraging measures.

decrease in precision, allowing the *chars+words+threshold* system to obtain the best result of all of the *constrained* systems. Compared with the winner of Tweet-LID, the *chars+words+threshold* system achieved a relative improvement of more than 5% in $Macro\text{-}F1$ (from 0.752 to 0.792). Note that the behavior of the *constrained* systems on the test was the same as these systems had during the tuning phase; adding words improved the system and adjusting the thresholds got the most improvement.

7.2.2. Results of the Unconstrained Approaches on the Test Set

The systems selected for the *unconstrained* task were: i) *w40k*, a system that uses n-grams of characters and words and the 40000 most frequent words for each language extracted from Wikipedia as features; ii) *s5k*, a system that incorporates 5000 randomly selected sentences from Wikipedia to the training corpus, for each language; iii) *extra*, a system that adds 2221 more tweets to the training corpus (1640 in Galician and 581 in Basque); iv) *extra+w40k*, a system that mixes the 2221 extra tweets in the training corpus and the 40000 most frequent words per language as features; v) *extra+w40k+threshold*, the system (*extra+w40k*) but adjusting the thresholds of each class to maximize the macro-F1 measure. As in the *constrained* task, we used the features and parameters obtained during the 10-fold cross-validation process.

For the *unconstrained* systems (the last 5 systems in Figure 4) it can be observed that unlike the *constrained* systems, the behavior of the *unconstrained* systems on the test set was a bit different from their behavior during the tuning phase. For the test, using Wikipedia as a source of information (both words and sentences) improves the performance more than adding new tweets to the corpus. The lack of variability in the added tweets is more noticeable on the test set. In contrast to what happened in the tuning phase, mixing new tweets and the most frequent words of Wikipedia did not improve the performance of the individual

systems; the *extra+w40k* system obtained 0.793 in the $Macro\text{-}F1$ measure, *w40k* obtained the same 0.793, and *extra* achieved only 0.788. Adjusting the thresholds is also the best alternative in this case. Although its precision decreases, the huge increase in recall helped the *extra+w40k+threshold* system obtain the best result in terms of $Macro\text{-}F1$ of all the system developed in this work. Compared with the best baseline, the winner of the *unconstrained* task at TweetLID, the *extra+w40k+threshold* system achieved a relative improvement of almost 6% in $Macro\text{-}F1$ (from 0.753 to 0.797).

It can be observed that, while the improvements achieved in recall are statistically significant (especially for the *extra+w40k+threshold* system), the variations in precision are not. We can conclude that the *extra+w40k+threshold* system improves the results obtained to date with this corpus and also that improvement is statistically significant at 95% of confidence.

7.3. Evaluation at the Class Level on the Test Set

We also conducted a study of the performance of the systems at the class level. Tables 4, 5, and 6 show the results per language obtained by the eight systems tested on the test set. Each row in the tables represents the results for one language, except the last one (amb) which joins the results of the ambiguous tweets. In addition, the macroaveraging (macro) evaluation measures are also included in the last row.

As expected, the best results were obtained for languages with more tweets in both the training set and the test set (*es, pt, ca*) and the worst results were obtained for languages with fewer samples. A particular case is the Basque language, the language with the least training samples (3.37% of the training corpus) which achieved better results than Galician (4.95% of the training corpus) and English (7.54% of the training corpus). We think it is because the morphosyntactic features of Basque language are very different from the Romance languages and the English language.

It can be observed that in all of the systems the worst results were obtained for the undefined (*und*) category. This is not a surprising result if we consider that this category joins tweets with languages that are not included in the task (i.e., German, French, etc.) and tweets without enough information to determine the language. The ambiguous category is the easiest case since just guessing any of the possible classes is considered a hit.

When the results of the *constrained* systems shown in Table 4 are compared, it can be observed that the *chars+words* system was able to increase the F1 measure for Galician by more than 5 points (from 0.497 to 0.551). Galician is practically the only class that considerably benefited from the inclusion of words to the model features. The only class that was worsened by this inclusion was the *und* class, which went from 0.410 to 0.397 of F1 measure. For the system with the thresholds adjusted per class, the *chars+words+threshold* system, it is clearly evident that the favored classes (*eu*, *gl*, and *pt*) greatly increased their recall at the expense of a loss of precision (the F1 measure remained approximately equal to the F1 measure achieved by the same classes in the *chars+words* system). The greatest F1 variation occurred in the *und* class. Some difficult samples that were incorrectly classified as *und* in the *chars+words* system are now correctly classified as *eu*, *gl*, or *pt* (due to their decrease in the threshold); therefore, the precision of the *und* class increased.

Table 4. Results obtained by the best *constrained* systems at the class level.

Class	BR			chars+words			chars+words+threshold		
	π	ρ	*F1*	π	ρ	*F1*	π	ρ	*F1*
ca	0.856	0.868	0.862	0.835	0.887	0.860	0.843	0.884	0.863
en	0.835	0.779	0.806	0.834	0.799	0.816	0.842	0.796	0.819
es	0.940	0.951	0.946	0.935	0.958	0.946	0.943	0.954	0.948
eu	0.928	0.807	0.863	0.943	0.796	0.863	0.912	0.854	0.882
gl	0.554	0.451	0.497	0.556	0.551	0.553	0.496	0.618	0.551
pt	0.943	0.912	0.927	0.932	0.927	0.930	0.928	0.939	0.933
und	0.371	0.459	0.410	0.403	0.392	0.397	0.428	0.438	0.432
amb	1.000	0.831	0.908	1.000	0.843	0.915	1.000	0.831	0.908
macro	0.803	0.757	0.777	0.805	0.769	0.785	0.799	0.789	0.792

Table 5. Results obtained at class level by the *unconstrained* systems that use external sources of information individually.

Class	w40k			s5k			extra		
	π	ρ	*F1*	π	ρ	*F1*	π	ρ	*F1*
ca	0.860	0.893	0.876	0.834	0.893	0.863	0.841	0.881	0.861
en	0.830	0.795	0.812	0.838	0.790	0.813	0.838	0.796	0.817
es	0.936	0.959	0.948	0.941	0.949	0.945	0.936	0.954	0.945
eu	0.945	0.815	0.875	0.951	0.802	0.870	0.956	0.796	0.868
gl	0.546	0.588	0.566	0.516	0.676	0.585	0.523	0.613	0.564
pt	0.937	0.922	0.929	0.939	0.928	0.934	0.947	0.923	0.935
und	0.429	0.425	0.427	0.397	0.439	0.417	0.389	0.413	0.401
amb	1.000	0.839	0.913	1.000	0.824	0.903	1.000	0.847	0.917
macro	0.810	0.779	0.793	0.802	0.787	0.791	0.804	0.778	0.788

When the systems that use Wikipedia as an additional information source (*w40k* and *s5k* in Table 5) are compared with *chars+words* system, there are no significant improvements in any class except in the *gl* and *und* classes. Contrary to what was expected, adding 2221 tweets in Galician and Basque to the training corpus improved the performance of the *extra* system very little for these languages. The recall for Galician increased from 0.551 to 0.613 (F1 from 0.553 to 0.566) and it increased for Basque from 0.796 to 0.815 (F1 from 0.863 to 0.875). Note that the behavior of these three systems was different in test phase than in tuning phase. During the tuning phase, the *extra* system was the best of the three, while in the test phase it was the worst.

Now we analyze the results of the *unconstrained* systems that mix different information sources, which can be seen in Table 6. The *extra+w40k* system adds

Table 6. Results obtained at class level by the *unconstrained* systems that mix external sources of information.

Class	extra+w40k			extra+w40k+threshold		
	π	ρ	*F1*	π	ρ	*F1*
ca	0.864	0.889	0.876	0.864	0.889	0.876
en	0.836	0.790	0.812	0.836	0.790	0.812
es	0.938	0.956	0.947	0.943	0.951	0.947
eu	0.952	0.813	0.877	0.921	0.857	0.887
gl	0.525	0.627	0.572	0.466	0.715	0.564
pt	0.941	0.920	0.930	0.928	0.937	0.932
und	0.415	0.434	0.424	0.437	0.450	0.444
amb	1.000	0.831	0.908	1.000	0.843	0.915
macro	0.809	0.783	0.793	0.799	0.804	0.797

the 2221 extra tweets to the training corpus and uses the 40000 most frequent words per language as features. The *extra+w40k+threshold* system also adjusts the threshold of each class. The *extra+w40k* system did not improve the performance of the *w40k* system. Only the performance for the *gl* class improved very slightly, but this improvement was not enough to improve the overall performance. Once again, the best result was obtained by adjusting the thresholds for classes with fewer samples. As in the case of the *chars+words+threshold* system, the improvement of the *extra+w40k+threshold* system was due to the improvement in the recall of the classes favored by the threshold, *eu*, *gl*, and *pt*. The performance of the *es* class (penalized by the increase of its threshold) actually was unaffected, probably because its model was very well estimated.

7.4. Confusion Matrix Analysis

In order to evaluate the behavior of our best system (*extra+w40k+threshold*) for each one of the classes on the test set, and to study its performance in multi-label samples, we constructed the confusion matrix that is shown in Table 7. The table also includes the accuracy measure (Acc). The accuracy for multi-label tweets is calculated as an exact matching between the set of languages assigned to the tweet and the set predicted by the system. Note that this matrix represents 99.7% of all unambiguous tweets; the rest are not shown due to space restrictions (the full matrix is very sparse).

Some conclusions that we can draw from the confusion matrix are the following:

– The confusion among monolingual tweets mostly occurs between languages that are linguistically related (e.g., between *gl-es* and *ca-es*).
– The accuracy of multilingual tweets is low. However, in most cases the system is able to predict at least one of the languages of the tweet. For example, the class *en-es* only has an accuracy of 31% but the system predicts correctly *es*

Table 7. Extract of the confusion matrix of *extra+w40k+threshold* system considering unambiguous tweets.

Labels	Acc.	es	pt	ca	en	gl	eu	und	en + es	es + eu	ca + es	ca + en	es + gl	pt + gl	pt + es	Total
es	0.94	11003	28	21	68	123	5	259	26	16	57	-	105	7	19	11737
pt	0.89	45	1722	-	2	6	-	62	-	-	-	-	-	73	18	1928
ca	0.88	47	3	1255	1	-	-	68	-	-	45	1	1	-	-	1421
en	0.85	44	4	4	770	-	-	64	13	1	-	3	-	-	-	903
gl	0.57	72	12	2	-	242	-	32	-	-	-	-	46	16	-	422
eu	0.83	14	-	-	-	2	297	12	-	30	-	-	-	-	-	355
und	0.45	287	28	108	58	10	5	425	6	2	6	-	-	1	2	938
en+es	0.31	69	-	-	13	-	-	7	40	-	-	-	-	-	-	129
es+eu	0.30	34	-	-	-	-	32	1	-	29	-	-	-	-	-	96
ca+es	0.17	38	-	16	-	-	-	3	-	-	12	-	-	-	-	69
en+pt	0.08	1	13	-	2	1	-	4	-	-	-	-	-	1	-	22
ca+en	0.15	-	-	6	1	-	-	4	-	-	-	2	-	-	-	13
es+gl	0.29	3	-	-	-	2	-	-	-	-	-	-	2	-	-	7
Total		11657	1810	1412	915	386	339	941	85	78	120	6	154	98	39	18040

in 53% of the tweets and *en* in 10%; this represents a partial success of over 94%. This behavior is similar in the other multilingual labels.

– The confusion of the Romance languages with the Basque language is very low. This is also true for the confusion between English and Basque.

– The language that most other languages are confused with is Spanish. This

may be because it is the language that has the largest number of samples in the training set and because of its preponderance even in the bilingual regions.
– The Galician language has the worst results of all languages. The confusion matrix shows that the most frequent confusion of Galician is with Spanish and Portuguese (67% of misclassification errors of Galician involve Spanish and 17% involve Portuguese). We think this is due to the close similarity among these Languages. Even though Galician and Portuguese are closer languages to each other than Spanish, two factors can justify the fact that there are more confusions with Spanish: 1) Spanish and Galician share similar symbols, for example, the symbol "ñ" is used in Spanish and Galician while Portuguese uses "nh"; 2) Spanish is the predominant language in the Galician region and, consequently, it has a great influence on the Galician language.

Finally, Table 8 shows the confusion matrix considering the ambiguous tweets. Note the high accuracy achieved by these tweets. Most errors in ambiguous tweets occur because of the *und* class, that is, when no language obtains a probability greater than its threshold.

8. Evaluation on a different data set

In order to evaluate our approach on other data sets, we investigated some available tweet collections similar to the TweetLID corpus. Among the data sets for language identification on Twitter used in (Lui and Baldwin, 2014), we decided to use the corpus developed in the work of Bergsma et al. (2012). This corpus (Bergsma corpus) is public-available and it contains very different languages that those included in TweetLID corpus.

Bergsma corpus is a collection of tweets in nine languages, focusing on confusable languages using the Cyrillic, Arabic, and Devanagari alphabets. We considered this corpus interesting because the set of languages included in it is very different to the one used in TweetLID corpus, and because the languages involved are very confusable. However, one of the shortcomings of Bergsma corpus is that it only contains monolingual tweets and the distribution of tweets per language is quite balanced. Nevertheless, we think that this collection allow us to evaluate the generality of the system developed in our work and if our system is over-fitted for the TweetLID corpus.

Table 9 shows the statistics of the corpus at the moment we downloaded it. Some tweets from the original data set could not be accessible, only 85% of the tweets of this collection was available in January 2016. Despite the amount of data is not exactly the same as the used in (Bergsma et al., 2012) or (Lui and Baldwin, 2014), we think that the decrease of tweets per category is homogeneous and therefore does not significantly affect the experimental comparison performed.

To compare our approach with those presented in (Bergsma et al., 2012) and (Lui and Baldwin, 2014), we decided not to use external resources for two reasons: i) Lui and Baldwin (2014) work did not use external resources, and ii) although, Bergsma et al. (2012) used external resources, not all these resources are detailed enough to be obtained. Consequently, we choosed our best constrained system for this comparison. In addition, to contrast our results with the related works, we used the same preprocessing method described in (Bergsma et al., 2012). That

Table 8. The confusion matrix of *extra+w40k+threshold* system considering the ambiguous tweets.

Labels	Acc.	es	pt	ca	en	gl	und	ca + es	es + gl	pt + es	pt + gl	ca +es +gl	pt +es +gl	Total
gl/pt	0.86	3	32	1	-	3	9	-	-	-	48	-	-	96
ca/es	0.78	25	1	4	2	-	10	20	1	-	-	-	-	63
es/gl	0.89	34	1	-	-	-	1	2	2	-	-	-	1	41
es/gl/pt	0.80	17	2	-	-	-	6	-	1	2	1	-	1	30
ca/es/gl/pt	0.85	9	-	-	-	-	2	1	-	-	-	1	-	13
ca/es/gl	0.75	3	-	-	-	-	1	-	-	-	-	-	-	4
ca/en/es	0.67	2	-	-	-	-	1	-	-	-	-	-	-	3
ca/en/es/eu/gl/pt	1.00	2	-	-	-	-	-	-	-	-	-	-	-	2
ca/es/eu/pt	0.00	-	-	-	-	-	1	-	-	-	-	-	-	1
ca/es/pt	1.00	1	-	-	-	-	-	-	-	-	-	-	-	1
en/es	0.00	-	-	1	-	-	-	-	-	-	-	-	-	1
en/es/gl/pt	1.00	1	-	-	-	-	-	-	-	-	-	-	-	1
en/gl	0.00	1	-	-	-	-	-	-	-	-	-	-	-	1
en/gl/pt	1.00	-	-	-	-	1	-	-	-	-	-	-	-	1
es/eu	1.00	1	-	-	-	-	-	-	-	-	-	-	-	1
es/eu/gl/pt	1.00	1	-	-	-	-	-	-	-	-	-	-	-	1
es/pt	1.00	-	-	-	-	-	-	-	-	1	-	-	-	1
Total		100	36	6	2	4	31	23	4	3	49	1	2	261

is, we removed from the tweets URLs, hash-tags, user mentions, punctuation, and digits.

In (Bergsma et al., 2012) the results are grouped by alphabet and the systems are ranked, within each alphabet, according to *accuracy* measure.

We trained one system per alphabet using the best *constrained* approach (*chars+words+threshold*) described in subsection 7.2.1. Table 10 shows the results of our system (*chars+words+thr* column) compared with two results from (Bergsma et al., 2012). On one hard, the *Bergsma chars* column shows the results from *Bergsma* work when only the text of the tweets is used for training the

Table 9. Statistics of the Bergsma corpus that were available in January 2016.

Arabic alphabet						
Language	#training		#development		#test	
Arabic	474	(25.92%)	236	(24.21%)	235	(25.19%)
Farsi	917	(50.14%)	438	(44.92%)	471	(50.48%)
Urdu	438	(23.95%)	301	(30.87%)	227	(24.33%)
Total	1829	(100.0%)	975	(100.0%)	933	(100.0%)
Devanagari alphabet						
Language	#training		#development		#test	
Hindi	547	(28.31%)	291	(33.03%)	243	(28.29%)
Marathi	566	(29.30%)	212	(24.06%)	281	(32.71%)
Nepali	819	(42.39%)	378	(42.91%)	335	(39.00%)
Total	1932	(100.0%)	881	(100.0%)	859	(100.0%)
Cyrillic alphabet						
Language	#training		#development		#test	
Bulgarian	851	(44.53%)	455	(46.96%)	449	(46.67%)
Russian	804	(42.07%)	395	(40.76%)	394	(40.96%)
Ukrainian	256	(13.40%)	119	(12.28%)	119	(12.37%)
Total	1911	(100.0%)	969	(100.0%)	962	(100.0%)

Table 10. Accuracy results of our approach on Bergsma corpus compared with the results published in (Bergsma et al., 2012), considering one model per alphabet.

Alphabet	chars+words+thr	Bergsma chars	Bergsma more
Arabic	0.979	0.971	0.979
Devanagari	0.977	0.962	0.979
Cyrillic	0.971	0.961	0.983

models, similar to the *constrained* task in TweetLID. This is the column directly comparable with our *constrained* approach. On the other hand, the *Bergsma more* column shows the results from *Bergsma* work when, in addition to text, tweets meta-data and/or external resources are used. Due to the impossibility of obtaining the same external resources, we could not train a model directly comparable with this one. However, this column has been added for completeness, because it represents the best result using this corpus.

It can be seen that when the same data is used (*chars+words+thr* and *Bergsma chars* columns) our system outperforms Bergsma approach for all alphabets. Moreover, the results of our *constrained* approach are in line with the results obtained by Bergsma *unconstrained* system using extra resources (*Bergsma more*). It should be noted that in the case of Cyrillic, the best results published in (Bergsma et al., 2012) (0.983) were achieved using only text and metadata from Twitter, when external resources were used the value of accuracy came down to 0.960.

In the work of Lui and Baldwin (2014) a comparison among different state-

Table 11. Results of our approach on Bergsma corpus compared with the results published in (Lui and Baldwin, 2014), considering only one model for all languages.

System	$macro-\pi$	$macro-\rho$	$macro-F1$
chars+words+thr	0.972	0.972	0.972
Lui majority voting	-	-	0.935

of-the-art language identification systems for Twitter is presented. For the comparison, a variety of corpora, including Bergsma corpus, is used. In (Lui and Baldwin, 2014) a unique model is learned for all languages in Bergsma corpus. Consequently, we also trained one model joining the training sets of the three alphabets. Table 11 shows the results of our system (*chars+words+thr* row) compared with the best result from (Lui and Baldwin, 2014) that consisted of a majority voting among the different considered systems (*Lui majority voting*). It can be observed that our system outperforms the best results obtained by Lui and Baldwin (2014) in terms of macro-F1.

9. Conclusions

In this paper, we have presented an approach for Language Identification in Twitter. We tested the approach on the freely available corpus developed at the TweetLID workshop. The corpus contains bilingual tweets that are written in one of the five most commonly used languages of the Iberian Peninsula (Spanish, Portuguese, Catalan, Basque, and Galician) as well as the English language.

We developed several systems based on the Support Vector Machine formalism using a lineal kernel. Based on the characteristics of the task, we proposed a solution the following problems: 1) Social Media Texts, proposing a suitable tokenization of tweets that correctly processes the peculiarities of Twitter; 2) Multilingual Tweets. Since a tweet can belong to more than one language, we developed a multi-class and multi-label classifier; 3) Similar Languages. We studied the main confusions among similar languages to mitigate this difficulty; 4) Unbalanced Classes. We proposed a correction method to increase the accuracy of the classifiers for classes with less data in the corpus by using thresholding strategies to favor minority classes.

We conducted an exhaustive study in order to determine the best set of features and parameters in our systems. We tested n-grams of words, n-grams of characters, and combinations of n-grams of words and n-grams of characters. The addition of n-grams of words to the n-grams of characters considerably improved the results. In addition, we also tested the use of external resources (such as Wikipedia) or the inclusion of additional downloaded tweets (automatically classified by our system) in order to increase the training data set. Contrary to what one might expect, only slight improvements were obtained regarding the constrained approach. These conclusions are in line with the results reported at the TweetLID competition in which participating systems achieved worse results for the unconstrained approach than for the constrained approach.

With our approach, we obtained competitive results for the TweetLID task. We achieved 0.792 of macro-F1 for the *constrained* task and 0.797 for the unconstrained one. Compared with the winner of TweetLID, the *chars+words+threshold*

system achieved a relative improvement in the *constrained* task of more than 5% in *macro-F*1 (from 0.752 to 0.792). For the *unconstrained* task at TweetLID, the *extra+w40k+threshold* system achieved a relative improvement of almost 6% in *macro-F*1 (from 0.753 to 0.797).

We also conducted a study of the performance of the systems at the class level. As expected, the best results were obtained for languages with more tweets in both the training set and the test set (*es, pt, ca*) and the worst results were obtained for languages with fewer samples. A particular case is the Basque language, the language with the least training samples (3.37% of the training set), which achieved better results than Galician (4.95% of the training set) and English (7.54% of the training set). Our thresholding strategy mitigates the problem of languages with less data in the corpus.

Moreover, we tested our system on the Bergsma corpus. This corpus contains tweets in nine confusable languages using the Cyrillic, Arabic, and Devanagari alphabets. The obtained results (similar and in some cases better than the best results published using this data set) show that our approach is independent from the corpus used.

As future work, we plan to continue working on this task taking into account new features and resources that can improve our system as well as other methods than can improve performance for minority languages in the corpus. Regarding the use of external resources, we think that there is still much work to do. The results achieved by the *unconstrained* systems in the TweetLID task or by Bergsma et al. (2012) probe that it is not easy to adapt external resources (Wikipedia and others) to be used on Twitter tasks.

Finally, we want to emphasize the importance of the TweetLID corpus for the scientific community for contrasting different approaches to identify multilingual tweets. We plan to freely distribute all of the resources developed in this work (i.e., the new additional data set of tweets obtained and the lexical resources extracted from Wikipedia).

Acknowledgments

This work has been partially funded by the project ASLP-MULAN: Audio, Speech and Language Processing for Multimedia Analytics (MINECO TIN2014-54288-C4-3-R).

References

Baldwin, T. and Lui, M. (2010), Language identification: The long and the short of the matter, *in* 'Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics', HLT '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 229–237.

Bergsma, S., McNamee, P., Bagdouri, M., Fink, C. and Wilson, T. (2012), Language identification for creating language-specific twitter collections, *in* 'Proceedings of the Second Workshop on Language in Social Media', LSM '12, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 65–74.

Carter, S., Weerkamp, W. and Tsagkias, M. (2013), 'Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text', *Lang. Resour. Eval.* **47**(1), 195–215.

Cavnar, W. B. and Trenkle, J. M. (1994), N-gram-based text categorization, *in* 'In Proceedings

of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval', pp. 161–175.

Cortes, C. and Vapnik, V. (1995), 'Support-vector networks', *Machine learning* **20**(3), 273–297.

Gamallo, P., García, M., Sotelo, S. and Campos, J. R. P. (2014), Comparing ranking-based and naive bayes approaches to language detection on tweets., *in* 'TweetLID@SEPLN', pp. 12–16.

Goldszmidt, M., Najork, M. and Paparizos, S. (2013), Boot-strapping language identifiers for short colloquial postings, *in* 'Proc. of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2013)', Springer Verlag.

Grefenstette, G. (1995), Comparin two language identification schemes, *in* 'In 3rd International Conference on Satatistical Analysis of Textural Data'.

Hurtado, L. F., Pla, F., Giménez, M. and Arnal, E. S. (2014), Elirf-upv en tweetlid: Identificación del idioma en twitter, *in* 'Proceedings of the Tweet Language Identification Workshop co-located with 30th Conference of the Spanish Society for Natural Language Processing, TweetLID@SEPLN 2014, Girona, Spain, September 16th, 2014.', pp. 35–38.

Jauhiainen, T., Lindén, K. and Jauhiainen, H. (2015), Language set identification in noisy synthetic multilingual documents, *in* A. Gelbukh, ed., 'Computational Linguistics and Intelligent Text Processing', Vol. 9041 of *Lecture Notes in Computer Science*, Springer International Publishing, pp. 633–643.

Joachims, T. (1998), Text categorization with support vector machines: learning with many relevant features, *in* C. Nédellec and C. Rouveirol, eds, 'Proceedings of ECML-98, 10th European Conference on Machine Learning', number 1398, Springer Verlag, Heidelberg, DE, Chemnitz, DE, pp. 137–142.

Liu, B. (2012), *Sentiment Analysis and Opinion Mining. A Comprehensive Introduction and Survey*, Morgan & Claypool Publishers.

Ljubešić, N., Mikelić, N. and Boras, D. (2007), Language identification: How to distinguish similar languages, *in* V. Lužar-Stifter and V. Hljuz Dobrić, eds, 'Proceedings of the 29th International Conference on Information Technology Interfaces', SRCE University Computing Centre, Zagreb, pp. 541–546.

Lui, M. and Baldwin, T. (2014), Accurate language identification of twitter messages, *in* 'Proceedings of the EACL 2014 Workshop on Language Analysis in Social Media (LASM 2014)', pp. 17–25.

Lui, M., Lau, J. H. and Baldwin, T. (2014), 'Automatic detection and language identification of multilingual documents', *Transactions of the Association for Computational Linguistics* **2**, 27–40.

Nguyen, D. and Dogruoz, A. S. (2014), 'Word level language identification in online multilingual communication', *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* .

O'Connor, B., Krieger, M. and Ahn, D. (2010), Tweetmotif: Exploratory search and topic summarization for twitter, *in* W. W. Cohen and S. Gosling, eds, 'Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010', The AAAI Press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011), 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research* **12**, 2825–2830.

Pla, F. and Hurtado, L.-F. (2014), Political tendency identification in twitter using sentiment analysis techniques, *in* 'Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers', Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pp. 183–192.

Prager, J. M. (1999), 'Linguini: Language identification for multilingual documents', *J. Manage. Inf. Syst.* **16**(3), 71–101.

Ramón Quevedo, J., Luaces, O. and Bahamonde, A. (2012), 'Multilabel classifiers with a probabilistic thresholding strategy', *Pattern Recogn.* **45**(2), 876–883.

Rao, D., Yarowsky, D., Shreevats, A. and Gupta, M. (2010), Classifying latent user attributes in twitter, *in* 'Proceedings of the 2Nd International Workshop on Search and Mining User-generated Contents', SMUC '10, ACM, New York, NY, USA, pp. 37–44.

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

Tsoumakas, G. and Katakis, I. (2007), 'Multi-label classification: An overview', *Int J Data Warehousing and Mining* **2007**, 1–13.

Zubiaga, A., Vicente, I. S., Gamallo, P., Campos, J. R. P., Loinaz, I. A., Aranberri, N., Ezeiza, A. and Fresno-Fernández, V. (2014), Overview of tweetlid: Tweet language identification at SEPLN 2014, *in* 'Proceedings of the Tweet Language Identification Workshop co-located with 30th Conference of the Spanish Society for Natural Language Processing, TweetLID@SEPLN 2014, Girona, Spain, September 16th, 2014.', pp. 1–11.

Zubiaga, Arkaitz, naki San Vicente, I., Gamallo, P., Pichel, J. R., Alegria, naki, I., Aranberri, N., Ezeiza, A. and Fresno, V. (2015), 'TweetLID: a benchmark for tweet language identification', *Language Resources and Evaluation* (First Online).

Author Biographies

Ferran Pla received his Ph.D. degree in Computer Science from the Universitat Politècnica de València in 2000. He is currently an Associate Professor in the Departament de Sistemes Informàtics i Computació of the Universitat Politècnica de València. He is member of the Natural Language Engineering and Pattern Recognition (ELiRF) research group at the same institution. He has published over 50 papers being involved in many research projects. His research interests cover many areas within natural language processing, including: POS tagging, parsing, name entity recognition, word sense disambiguation, and sentiment analysis in social media.

Lluí F. Hurtado received his Ph.D. degree in Computer Science from the Universitat Politècnica de València in 2004. He is currently a Associate Professor in the Departament de Sistemes Informàtics i Computació of the Universitat Politècnica de València. He is member of the Natural Language Engineering and Pattern Recognition (ELiRF) research group at the same institution. He has published over 80 research papers being involved in many research projects. His research interests cover many areas within speech processing and natural language processing, including: spoken dialog systems, voice-activated question answering, spoken language understanding, and sentiment analysis.

Correspondence and offprint requests to: Ferran Pla, Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València, Camí de Vera sn, 46600 València. Email: fpla@dsic.upv.es