Robust Language Identification of Noisy Texts
- Proposal of Hybrid Approaches

Kheireddine Abainia
USTHB University
Algiers
abainia@hotmail.fr

Siham Ouamour
USTHB University
Algiers
siham.ouamour@uni.de

Halim Sayoud
USTHB University
Algiers
halim.sayoud@uni.de

***Abstract*—This paper deals with the problem of automatic language identification of noisy texts, which represents an important task in natural language processing. Actually, there exist several works in this field, which are based on statistical and machine learning approaches for different categories of texts. Unfortunately, most of the proposed methods work fine on clean texts and/or long texts, but often present a failure when the text is corrupted or too short. In this research work, we use a typical dataset consisting of short texts collected from several discussion forums containing several types of noises. Our dataset contains 32 different languages; where we notice that some languages are quite different while some others are too closed.**
In this investigation, we propose two types of methods to identify the text language: term-based method and character-based method. Moreover, we propose two hybrid methods to enhance the performances of those techniques. Experiments show that the proposed hybrid methods are quite interesting and present good language identification performances in noisy texts.

Keywords—Natural Language Processing; Text categorization; Automatic Language Identification; Noisy Text; Hybrid Approach.

I. INTRODUCTION

The wide use of numerical data and textual information facilitates the sharing of information between people, where the important size of the shared information leads to increase the size of such textual information and databases in different fields. Hence, the access to the information becomes difficult or expensive, and in this respect, many research works were performed to extract the information automatically from databases. That task needs a natural language processing (NLP) or computational linguistics based processing of the written texts, which requires knowing the language in advance, to select the best features and the appropriate language processing procedures. Then, language identification is an important step in the information extraction process. That reason prompted many researchers to deal with the field of *automatic language identification* during the last years.

Nowadays, there are several categories of numerical media text information in over the web news, education, forums…etc. The more interesting categories for many researchers are the social network messages and forums. There are some difficulties presented in these two categories, where texts have different noises (*Abbreviations, tags…etc*). In some cases it is impossible to identify the language of a given text, because the whole text is abbreviated or written as SMS style.

In this investigation, we propose five statistical approaches: three basic methods and two hybrid ones. The first basic method, called CBA (Characters Based identification Algorithm), is based on characters. Another similar method, called WBA (Words Based identification Algorithm), is based on frequent/common terms. A new other method, based on special characters, has been proposed too: it is called SCA (Special characters Based identification Algorithm) and is based on two stages of classification. Furthermore, two hybrid methods have been proposed in order to enhance the performances of the previous techniques. The first one is a sequential combination of CBA and WBA; and the second one is a parallel combination of those two techniques. For the evaluation process, we have created a small dataset, called DLI-32, which contains 320 text documents collected manually over several discussion forums in 32 different languages. Some of those 32 languages are too closed and share many features, which obviously increases the difficulty of identification. On the other hand, the evaluation experiments are performed on two subsets of the previous dataset: one subset contains 10 languages and another one contains 32 languages. This diversity is interesting to know the limits of the proposed approaches and to reveal their performances. Finally, a comparison is made between our best identification approaches and some well-known tools of identification, such as Google Translator and Microsoft Word.

The paper is organized as follows. In section 2, we present some related works in the field of language identification. In section 3, we give an overview of our Dataset and character encoding used in this work. In section 4, we present some basic statistical methods of language identification that are implemented; thereafter we introduce two hybrid methods to identify the text language. Section 5 describes the experimental investigation in which we have compared the three basic methods and the two hybrid methods. At the end of this section, we present a comparison of our best results with these of Google Translator and Microsoft Word. Section 5 summarizes the present work and gives some suggestions and perspectives.

II. RELATED WORKS

During the last years, several researchers have been interested in the field of language identification. For instance, in a study of language identification applied to web pages [1], the authors used four languages and they applied some basic approaches like trigrams (top 1000 words). They tested their approaches on different collections of datasets (eg. Wikipedia and Europarl) with different text sizes (ie. short and long texts). The results in long texts are quite good unlike in short texts. Another work based on distance measures was reported by Cavnar & Trenkle [2], by using an Out-Of-Place distance with N-grams to measure the distance between two documents. The approach is based on creating an N-gram profile for each category, and reordering the profile elements by a descending order. To classify a document, they computed the distance between a test profile and all reference profiles, which make the approach more expensive and not very robust. A study of some distance measures was also performed in [3] to compare the Out-Of-Place distance with other distance measures. The authors used 39 languages, where the dataset is collected from Gutenberg and CIIL, consisting in brut text documents without noise (ie. books). The training data size ranges from 2495 to 102377 words. Other research works are based on machine learning algorithms [4], the authors used the discrete HMMs with three models for comparison. It was the first work based on HMM in text document analysis. The dataset used in their experiments is a collection of HTML pages collected over hotel web sites in 5 languages. The texts are not noisy, because this genre of web sites presents the information as well as possible. A decision tree-ARTMAP model [5] was used in language identification for Arabic script web documents. The authors used two closed languages (ie. Arabic and Persian), and their method brings a quite good result. However, their method works fine only with these two languages, and it is not suitable if the dataset is extended to other languages, like Urdu for instance, due to the letters similarity. A study between different classification methods was also performed in [6], the authors used the nearest neighbor model with three distances, Naive Bayes and SVM model. They used three datasets with different properties (ie. Wikipedia with 67 languages, TCL with 60 languages and EuropoGov with 10 languages), and the documents size ranges between 1480 and 39353 bytes which corresponds to long texts. Their results show that the nearest neighbor model using N-grams is the most suitable in the three datasets. Some researchers worked on short messages in social networks like twitter for instance [7], the authors used supervised learned approaches like Prediction by Partial Matching and Logistic Regression with different features. They used three families of languages (i.e. Arabic, Devanagari and Cyrillic). They did not classify the text to sub languages (e.g. Arabic, Farsi or Urdu) and they classified the language into the family of languages. Another work conducted on web pages was performed by [8] using the similarity between string-based N-grams by employing heuristics for text categorization. Their techniques were experimented on 12 different languages, which are not very closed, and a training dataset collected from newsgroups was used. Also, they cited some limitations of their experimental results. A graph based approach was proposed in [9], where the authors used a graph representation based on N-gram features and graph similarity. However, their approach is quite expensive due to the construction of the different N-gram graphs.

In this research work we propose some basic statistical approaches of language identification, which are quite simple and easy to implement, furthermore, the proposed techniques have to be robust in noisy texts.

III. DATASET AND CHARACTER ENCODING

The textual documents processing requires a judicious choice of the character encoding. In fact, this is a primordial step before starting any process. The same character set encoding has been used in this work: it is the UTF8 encoding, because 32 different languages have been used in this research work, with different character encoding in the beginning. The UTF8 encoding covers more than 1900 characters in different languages, where each character is coded in 1 byte to 4 bytes (*e.g. Chinese characters are coded in 3 bytes*).

The construction of the Dataset was the hardest task, because it is constructed manually. The Dataset contains 320 texts in 32 different languages, where the texts are collected from several forums. Concerning the difficulties we noticed that some languages have very few forums and in some others, it is very difficult and hard to find forums. Our global Dataset contains 320 texts of about 100 words per text and is formatted according to UTF8 format. Furthermore there are 10 different texts for every language. The dataset is unbalanced and the texts are noisy with the following different noises: URLs, Citations in other language, Tags, Abbreviations, Unaccented characters, Typing errors and Insignificant characters.

IV. LANGUAGE IDENTIFICATION METHODS

Generally, before starting any identification process, it is preceded by the creation of the reference profiles (e.g. list of frequencies) based on a training process using a subset of the dataset (training dataset). The global scheme of the language identification process is summarized in four global steps: *loading reference profiles, reading the text file, preprocessing and finally the identification*. The preprocessing step consists of removing the insignificant characters, which may cause some problems of identification. In the identification step, we create a profile of the given text based on the selected feature, and we compare it with all reference profiles (ie. all languages). Finally, we classify the text according to the language which has the nearest profile.

In our experiments we use several basic statistical methods to identify the language of a given text. Firstly, we describe

two methods based on characters and terms to identify the language, respectively, and another one which is based on special characters. Moreover, we propose two hybrid approaches to improve the performances of the previous algorithms and increase their accuracy.

A. Character Based Algorithm (CBA)

The first algorithm implemented in our investigation is based on the characters (*or letters*) of the languages. The CBA algorithm uses a set of predefined characters for each language, and it consists in computing the frequency of each character in the text. Next, we add all frequencies together (*computing the sum of the frequencies*) using equation 1, and finally we classify the text according to the language that has the highest sum of frequencies.

$$Sum_i = \sum_{j=1} frequency_{ij} \quad (1)$$

where $frequency_{ij}$ is the frequency of the j[th] character of language i.

However, when two or more languages share the same set of predefined characters or the same set characters in the text, there will be a problem, since many languages will have the same highest frequencies. To deal with this problem, we follow a special order to classify the languages. Reference characters are usually retrieved from the web (letters of each language), except the Chinese language, because the Chinese language contains more than 1300 characters. To deal with Chinese characters, we test the code point of the character if it is in the range of the Chinese UTF8 code. After computing the additive frequencies for each language, we classify the text in the corresponding language. The first test performed is: if the highest sum of frequencies is equal to Chinese frequency, Greek frequency, Thai frequency, Hebrew frequency or Hindi frequency then return one of those languages. Those languages are very distinctive in our experiments.

If no language is returned then we perform the following tests:

$$\begin{cases} Return \text{ Arabic} \\ \quad if \text{ max\_freq} = \text{arabic\_freq } and \text{ max\_freq} != \text{english\_freq} \\ \quad else \text{ pass to next test} \end{cases} \quad (2)$$

Where max_freq is the highest sum of frequencies, arabic_freq and english_freq are the Arabic and English frequencies respectively.

Repeat the equation (2) with the following order of languages: Persian, Urdu, Bulgarian, Russian, French, Italian, Irish, Spanish, Portuguese, Albanian, Czech, Finnish, Hungarian, Swedish, German, Norwegian, Danish, Icelandic, Turkish, English, Dutch, Indonesian, Malay, Latin, Roman, Polish.

B. Word Based Algorithm (WBA)

The WBA algorithm is roughly similar to the previous one. However, instead of using characters as features, we use common terms (words) to identify the language. After the text preprocessing, we extract all the terms from the text in order to compute each word frequency of the reference profiles, next we follow the previous classification process presented in CBA (described in section 4.1).

C. Special Character based Algorithm (SCA)

This new algorithm uses the special characters characterizing each language. It is decomposed into two classification steps:

The first step consists in classifying the language into a global class of languages. Hence, there are 8 classes of languages, and each one regroups one or several languages together. Each class is defined by a set of languages and a set of characters. Hence, to classify a particular text into one of the predefined classes, we use the characters of each class and we compute the sum of frequencies of those characters, next we classify the text corresponding to the class which has the highest sum of frequencies. If the class is class 1, class 2, class 6, class 7 or class 8 then the algorithm returns the language of the classified class directly, because those classes contain only one language per class. Else, the algorithm moves on to the second classification step.

The second classification step consists in determining exactly what language the text belongs to. Each language is defined by a set of characters which do not exist in the global class. Hence, some languages are not characterized by any characters (all characters are defined in the global class). To classify the text into the corresponding language of the class languages, a sum of frequencies of the characters is computed (language characters), and next, the classification is assigned to the language with the highest sum of frequencies and following the same order described in CBA.

D. First Hybrid Approach (HA1)

A hybrid approach has been proposed to identify the language and increase the score of identification. This approach is based on a sequential combination of the two previous algorithms CBA and WBA. Firstly, the CBA algorithm is performed, then, if the algorithm returns the English language then it will execute the WBA algorithm. This hybrid combination usually corrects the weaknesses of the first identification algorithm, where there are some confusion errors between some Latin-based languages (*English, Spanish, German, French, etc.*).

E. Second Hybrid Approach (HA2)

Another hybrid approach based on the fusion between CBA and WBA algorithms has been implemented. The proposed method uses a parallel fusion, where the two algorithms (CBA and WBA) are executed in parallel (at the same time), and once the two processes are finished without classifying the language, we add the sum of frequencies of the two algorithms for each language using equation (3). Finally, we classify the text using the same process described previously (section 4.1).

$$Sum_i = freqCBA_i + freqWBA_i \quad (3)$$

where $freqCBA_i$ is the sum of character frequencies in language i, and $freqWBA_i$ is the sum of word frequencies in language i.

V. EXPERIMENTS AND RESULTS

Our experiments are performed on two subsets of the *DLI-32* dataset:

- The first subset contains only 10 different languages;
- The second subset contains 32 different languages.

The main reason for selecting two subsets is to evaluate the efficiency of our approaches on a small number of languages and on large number of languages.

Firstly, we present the results obtained on the 10 languages, after that, we present the results got on the 32 languages. Subsequently, we will examine the performances of the different algorithms and try to make a comparison between them. Finally, we complete those experiments by making a comparison of our best results with other well-known tools of language identification (*i.e. Google Translator and Microsoft Word*).

Note: "Test 1" is performed using 20 common words collected from the web for each language, and "Test 2" is performed using 20 common words obtained from a training process using 4 texts per language. On the other hand 40% of the dataset is used to extract common words and 60% is used for the evaluation.

TABLE 1. COMPARATIVE SCORES OF LANGUAGE IDENTIFICATION WITH 10 LANGUAGES.

	CBA	**WBA** *test 1*	**WBA** *test 2*	**SCA**	**HA1**	**HA2**
Identification Score in %	100	90	90	100	100	100

The WBA algorithm uses white space and line feed to extract words from the text, and that causes some confusion problems in the Chinese language. Hence, the Chinese words are not spaced like the other languages, and each Chinese character represents a word. The identification score of WBA is 90% in the two tests: herein, all texts are recognized correctly, except Chinese texts, which are not recognized.

The identification score of CBA and SCA algorithms is 100% when the evaluation is performed on the whole dataset. The two algorithms are more accurate than WBA, and they analyze the text "character by character", which avoids the previous problem of the Chinese language. However, if two or more languages share the same set of characters in the text then that may cause some classification problems. To deal with that issue, a sequence of tests is performed by respecting a specific order of tests (described previously). The proposed order may cause another classification problem, when two languages have the same highest sum of frequencies. For instance, a Persian text could be recognized as an Arabic text, and an Italian text could be recognized as a French one, because the two couples of languages are quite closed and because the Arabic and French tests precede the Persian and Italian tests respectively.

On the other hand, the hybrid approaches are more precise and present excellent performances in the 10 languages (*100% of good identification*).

TABLE 2. COMPARATIVE SCORES OF LANGUAGE IDENTIFICATION WITH 32 LANGUAGES.

	CBA	**WBA** test 1	**WBA** test 2	**SCA**	**HA1** test 1	**HA1** test 2	**HA2** test 1	**HA2** test 2
Identif. Score %	72.40	92.19	83.85	71.35	89.06	87.85	97.92	94.27

Comparing table 2 with the previous one (table 1), we notice that the identification scores are reduced for the all algorithms, which means that the number of languages impacts directly the accuracy. The main reason comes from the fact that many closed languages are used in the second subset of the dataset, and the language identification process becomes more difficult obviously.

In the 32 languages, we notice that there is a difficulty in some languages, which is due to the inclusion of some characters within other different languages, like Arabic characters, which are included in the Persian and Urdu language. We also notice another similar case: some languages have almost the same character set (letters) like English, Roman, Latin, Malay, Indonesian and Dutch language. There is another difficulty, which is due to the similarity between the two couples of languages Indonesian/Malay and Danish/Norwegian, since, those languages share the same character set, grammar and vocabulary.

From the two previous tables (table 1 and 2), we notice that the accuracy of the WBA algorithm is slightly reduced compared with the previous results (10 languages). However, test 1 is better than test 2, because the words in test 1 are collected from the web, which are obtained by other researchers using large datasets, and the words in test 2 are obtained from a small subset (i.e. 4 texts per language).

Contrariwise, the accuracy of CBA and SCA algorithms are highly reduced in the second test (32 languages), and the performance is highly lower than the WBA performance. As described above, there are many languages sharing a set of characters which impacts directly the performance (i.e. identification based on characters). An accuracy of 72.40% and 71.35% for the two algorithms, respectively, are obtained from an evaluation on the whole dataset.

Concerning the hybrid approaches, we notice the following points:

- Comparing the hybrid approach with the basic approaches, we see that the hybrid techniques present better results than the basic ones, because of the fusion advantages.

Therefore, the combination does provide a great advantage, and the hybrid algorithms present high performances in the overall.

- Comparing the two hybrid approach each other, we notice that HA2 is quite better in language identification. This method takes more advantages from the two fused methods (*CBA and WBA*), then the results are obviously better than those of the other one.

TABLE 3. COMPARATIVE PERFORMANCES OF THE HYBRID METHODS WITH SOME UNIVERSAL TOOLS: ON THE 1ST SUBSET (SMALL DATASET).

	Google Translator	**Microsoft Word**	**HA1** *test 1*	**HA2** *test 1*
Identification Score in %	98	90	100	100

TABLE 4. COMPARATIVE PERFORMANCES OF THE HYBRID METHODS WITH SOME UNIVERSAL TOOLS: ON THE 2ND SUBSET ET, WITHOUT LATIN AND URDU.

	Google Translator	**Microsoft Word**	**HA1** *test 1*	**HA2** *test 1*
Identification Score in %	93.67	62.22	95	97.78

Concerning the comparison of our approaches with universal identification tools, we have compared our best results (HA1 and HA2) with universal identification tools (i.e. Google Translator and Microsoft Word) in the two subsets of the dataset to reproduce the same evaluation. We see from the two tables (table 3 and table 4) that the two methods HA1 and HA2 (with test 1) provide high performances and seem to be better than Google Translator and Microsoft Word. We notice that Microsoft Word did not manage to recognize several documents, for instance; Persian texts were recognized as Arabic texts. On the other hand, Turkish texts were recognized as French texts. Also it recognized some Hindi texts as French ones. There are several reported cases of false identification with Microsoft Word. Google Translator was also mistaken in several cases, for instance; some Malay texts were recognized as Indonesian ones, because the two couples of languages are too closed. On the other hand, some Latin texts were recognized as Italian ones, whereas the 2 languages are not closed. So, we have reported some cases of false identification with Google Translator too.

According to these last results, we can say that the 2 proposed hybrid approaches are more accurate than Google Translator and Microsoft Word for the identification of noisy texts.

VI. CONCLUSION

In this investigation, several experiments of language identification, in noisy text, have been conducted and commented. The text documents used in the different experiments are collected from several web forums and contain different types of noises. These textual documents, which constitute our experimental dataset DLI-32, correspond to 320 texts written in 32 different languages.

In this research work, we have proposed three basic language identification algorithms: characters based identification (CBA), special character based identification (SCA) and common words based identification (WBA). Furthermore, we have proposed two hybrid approaches based on the combination of the two previous methods (*character based identification and common words based identification*): HA1 and HA2. These two combinations have presented good performances, especially the parallel fusion based approach (*HA2*), which got an identification score of 100% with 10 languages and a score of 97.78% with 30 languages. The results of HA2 are better than those obtained by HA1, which shows that the parallel fusion is quite interesting.

On the other hand, results show that the proposed hybrid techniques are more accurate than Google Translator and Microsoft Word, especially for the approach HA2, with a difference of 4% in the identification score, with regards to Google Translator and a difference of 35% with regards to Microsoft Word. Consequently, the proposed approaches seem to be quite interesting and could be used efficiently to recognize the language of noisy texts. In perspective, we suggest trying other types of combinations and fusions to increase the language identification performances.

REFERENCES

[1] Rachel Mary Milne, Richard A. O'Keefe and Andrew Trotman. A Study in Language Identification. *ADCS '12, December 05 - 06 2012, Dunedin, New Zealand*.

[2] William. B. Cavnar and John. M. Trenkle. N-gram-based text categorization. *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, pages 161-175, 1994*.

[3] Anil Kumar Singh. Study of Some Distance Measures for Language and Encoding Identification. *Proceedings of the Workshop on Linguistic Distances, pages 63–72, Sydney, July 2006*.

[4] A. Xafopoulos, C. Kotropoulos, G. Almpanidis and I. Pitas. Language identification in web documents using discrete HMMs. *The Journal Pattern of Recognition Society, Pattern Recognition 37 (2004) 583 – 594*.

[5] Ali Selamat, Ng Choon Ching and Yoshiki Mikami. Arabic Script Web Documents Language Identification Using Decision Tree-ARTMAP Model. *Proceedings of IEEE Computer Society, 2007 International Conference on Convergence Information Technology*.

[6] Timothy Baldwin and Marco Lui. Language Identification: The Long and the Short of the Matter. Human Language Technologies: *The 2010 Annual Conference of the North American Chapter of the ACL, pages 229–237, Los Angeles, California, June 2010*.

[7] Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink and TheresaWilson. Language Identification for Creating Language-Specific Twitter Collections. *Proceedings of the 2012 Workshop on Language in Social Media (LSM 2012), pages 65–74, Montreal, Canada, June 7, 2012*.

[8] Bruno Martins and Mário J. Silva. Language Identification in Web Pages. *2005 ACM Symposium on Applied Computing*.

[9] Erik Tromp and Mykola Pechenizkiy. Graph-Based N-gram Language Identification on Short Texts. *Proceedings of the 20 th Machine Learning conference of Belgium and The Netherlands, 2011*.