IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 6 Issue 6, June 2019
ISSN (Online) 2348 – 7968
www.ijiset.com

Unicode Based Language Identification Using Trigrams

Fathima Sana K[1] and Farzana T[2]

[1] Department of Computer Science and Engineering, MES College of Engineering, Kuttipuram, Kerala , India

[2] Department of Computer Science and Engineering, MES College of Engineering, Kuttipuram, Kerala , India

Abstract
In recent years, with the widespread of internet and digitized processing of both monolingual and multilingual documents worldwide, the requirement of having effective automated language identifier has increased further. Where the language identification is the task of automatically detecting the language(s) present in documents. To improve the performance of language identification, here propose a method with active usage of Unicode information about scripts. The efficiency of algorithm was evaluated on sentences from 94 languages.The experimental results verified that, after script identification, language group identification and similar language group identification, the performance of the language identification algorithm improved with each stage. Notably, the language identification accuracy between similar languages improved by using character trigram.

Keywords: *Language identification, Multilingual documents, Unicode, Trigram*

1. Introduction

In recent years, with the widespread of internet and digitized processing of documents worldwide, script identification techniques have become more important in the pattern recognition field.
The task of identifying languages dates back to the ages when people started using language for communication. Human beings are very good at identifying spoken languages but not all are good at identifying languages in written form and also one cannot expect every human being to know all the languages.Language identification(LI) is generally viewed as a form of text categorization. It is a process that attempts to classify text in a language into a predefined set of known languages. Identification of languages has become an important pre processing step for a number of applications such as machine translation, Part of-Speech tagging, linguistic corpus creation, supporting low-density languages, accessibility of social media or user-generated content, search engines and information extraction in addition to processing multilingual documents. In a multilingual country like India, Language Identification has wider scope to bridge the digital divide between different language users.
Studies have verified that language identification accuracy rapidly drops when identifying short text, and confusion errors often occur between languages in the same family or in similar language groups. Therefore, considerable room for improvement exists in terms of improving short-text LI performance and similar language identification performance. Languages are written in different scripts, and each script has a unique defined code range in Unicode. This helps identify different parts of a script within a document.Languages belong to different families, and language families can be divided into similar phylogeny units. A remarkably similar pattern is exhibited by languages within a phylogeny unit [5], and this can be leveraged in LI to allow discrimination between different language groups written in the same script to narrow the range of identification. This paper presents a brief overview of the challenges involved in automatic language identification, existing methodologies and some of the tools available for language identification.
LI is a well-established research topic with state-of-the art algorithms achieving over 95 percentage accuracy. Text collection (web page or text document) which goes as input to Language Identification may be written in only one language (monolingual) or multiple languages (multilingual). Processing monolingual documents is fairly simple compared to that of processing multilingual documents as the former requires the knowledge of only one language while the later requires the knowledge of several languages and also other related issues.

2. Existing System

There are a number of cooperative routing techniques for language identification. Language identification was arguably established as a task by Gold(1967), who construed it as a closed class problem: given data in each of a predefined set of possible languages, human subjects were asked to classify the language of a given test document. It wasn’t until the 1990s, however, that the task was popularized as a text categorization task. The text categorization approach to language identification applies a standard supervised classification framework to the task. Perhaps the best known such model is that of Cavnar and Trenkle (1994), as popularized in the text cat tool.
Like the approaches there are many language identification techniques are invented later.

3. Proposed System

Computational approaches in Language Identification (LI) often result low accuracy and high error rate. Especially if the languages come from the same subfamily. In this paper, we have implemented improved variations of Unicode aided script identification and trigram based language detection to improve accuracy and to decrease error rate in LI. Like the most LI techniques, It is also a two phased process.The first phase is being training and the second is identification. Before going on to identifying documents, the system must be trained for the languages that we wish to detect. It only detects the languages it has been trained for. In identification phase, we present a Unicode based script identification for language and trigram used language identification.

3.1 Unicode based script identification

Unicode I is a computing industry standard for the consistent encoding, representation,and handling of text expressed in most of the worlds writing systems. The use of Unicode in our research has proved to be very useful. The global acceptance of Unicode and the well thought out placement of different languages in the Unicode code-set hasgivenusmoreopportunitiestodetectlanguageseffectively. Each script/language in Unicode has defined code ranges.The Unicode standard divides the Unicode character map into different blocks or ranges of code points. Each block is used to define characters of a particular script like ”Tibetan” or belonging to a particular group like ”Braille Patterns”. Most blocks include unassigned code points, reserved for future expansion of the Unicode standard. An essential difference between blocks and scripts is that a block is a single contiguous range of code points, as in below.

1.Basic Latin: U+0000U+007F
2.Latin1 Supplement: U+0080U+00FF
3.Latin Extended A: U+0100U+017F
4.Latin Extended B: U+0180U+024F
5.IPA Extension: U+0250U+02AF
6.Spacing Modifier Letters: U+02B0U+02FF
7.Combining Diacritical Marks: U+0300U+036F
8.Greek and Coptic: U+0370U+03FF
9.Cyrillic: U+0400U+04FF
10.Cyrillic Supplementary: U+0500U+052F
11.Armenian: U+0530U+058F
12.Hebrew: U+0590U+05FF
13.Arabic: U+0600U+06FF

Scripts consist of characters taken from all over the Unicode character map. Blocks may include unassigned code points. Scripts never include unassigned code points.
Although some languages are written by multiple scripts that have different Unicode code ranges like there is no Japanese Unicode script. Instead, Unicode offers
the Hiragana, Katakana, Han, and Latin scripts that Japanese documents are usually composed of and they share common scripts or words with other languages using the same script. Text script is easier to identify than its language. ISO 639-1 and ISO 639-2 codes are used to annotate language. In this coding schemes, each language
is represented by a two-letter code and three letter code respectively[6] as shown in Table 1

Table 1:ISO 639-1 and ISO 639-2 codes annotate languages

ISO language name	Native name	639-1	639-2
Afar	Afaraf	aa	aar
Afrikaans	Afraikaans	af	afr
Akan	Akan	ak	aka
Albanian	Shqib	sq	sqi
arabic	ا لعربيۃ	ar	ara

Languages in the same phylogeny unit are similar in terms of vocabulary and structures and the majority of confusion errors in language identification occur between similar languages. This advantage can be used to discriminate between different language groups among languages using the same script.Here we implemented SLGI (Similar Language Group Identification) in script. Where the languages with similar scripts are belongs to same group.

This helps reduce the number of languages in LI and should improve the LI accuracy.
For script identification, here uses a REMSI (Regular Expression Matching-based Script Identification) Algorithm using Unicode Regex Engines[7] . Based on the code range of the scripts in Unicode, there is a regular expression for every script. The proposed SI stage consists of the following steps:
Step 1. Remove items that are not alphabets, punctuation spaces and characters from the input sentence.
Step 2. Input data encoding into Unicode block format using Unicode Regex engines. This process is called Unicode normalization. Every character has unique Unicode block format. Eg: a is encoded as U+0061. Whereas `a encoded as U+0061 U+0300
Step 3. Using Unicode Regex Engine, use regular expressions to match the sentence to each scripts regular expressions to judge whether relevant script contents are in the sentence.
Step 4. Calculate the length of each matching result, and if the length is nonzero, save it in a list. Sort the list by decreasing length.
Step 5. Select the top item on the list as the main script of the text. If a script only has one member, return the language; otherwise, return its script, and further identify its language within languages using the same script. If the language ISO code is returned, the LI has identified and returned the sentence’s language.

3.2 Trigram based language identification

In this system, an approach that aims to measure language similarity using trigrams is presented. A trigram is an example of an n-gram or an n-character slice of a word[10]. As an example, the list of trigrams that can be generated from the word test are _te,tes,est,st_ .The process of performing LI involves the following:

3.2.1 Training data

Before going on to identifying documents, the system must be trained for the languages that we wish to detect. It only detects the languages it has been trained for.
To measure language similarity, training data is needed. The trigram data sets of 94 languages involved have to be stored to the database. Each trigram model for a language contain commonly occurring trigram with high frequency.rained trigram data sets some of languages is shown below.

tmodels[”af”]={”ie”,”di”,”die”,”en”,”ing”,”an”,”en”…..}
tmodels[”az”]={”lr”,”in”,”n”,”lar”,”da”,”an”,”ir”,”ki”,..}
tmodels[”ca”]={”de”,”es”,”de”,”la”,”el”,”que”,”el”,…..}

Where” tmodels” represent trigram data sets of languages.

3.2.2 Creating language models using the data collected

Trigram models are generated from input data and they contain frequency count of trigrams. Where the most occurring high frequency trigrams are taken for to check the similarity measures. If the Input Data is Mary and Samantha arrived at the bus station early but waited until noon for the bus, then the generated trigrams will be like{ _ma,mar,ary,ry_ ,...............us_ }

3.2.3 Using similarity measures to determine which among these to languages the input is in

The language that yields the lowest distance measure is identified as the language of the input. Model-based LID works by comparing a text input with trained models extracted from large corpora. It can be mathematically described as

dist=Math.Abs(model.Trigrams[i].score−knownmodel.Get Score(model.Trigrams[i].t)) (1)

Where ’dist’ is the distance between the generated trigram model and trained trigram model is measured by the absolute math function and ’known model’ indicate the trained language model. Using this function the lowest distance measure between generated trigram and trained trigram model is taken as the language of input.

4. Testing and Results

Tests were conducted for 94 languages belongs different groups with different string
lengths:
1. less than 10 characters
2. 11 to 20 characters
3. 21 to 40 characters
4. more than 40 characters. Tests are conduct for all languages of all groups. Where every languages other than from the group Basic Latin shows above 95% to the character length of 10 and shows accuracy over 98% on character length more than 10.

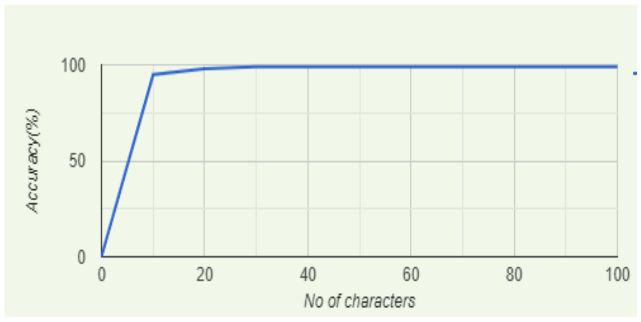

Figure 1: Performance analysis for groups other than Basic Latin

The system shows accuracy over 95% for the group Basic Latin, if it satisfy the minimum character length as 20.This group shows less than 50% accuracy if the character length is below 20. Group Basic Latin acquire accuracy over 98%if the character length of input sentence is more than 30.

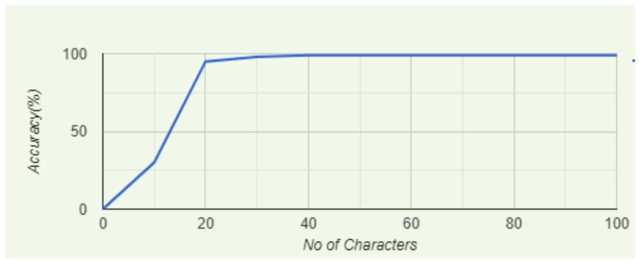

Figure 2: Performance analysis for groups Basic Latin

5. Conclusion

We carried out a thorough (re)examination of the task of language identification that is predicting the language that a given document is written in, focusing on multilingual documents. The system experimented with a identification of 94 global languages with different scripts in a two phased process.The first phase is Unicode based script identification from input sentence and second phase is language identification using trigram features of languages. By performance analysis it shows that how the Unicode script detection and language identification using trigrams can be used to improve accuracy. Results show that LID Can obtain about 98% of accuracy at minimum character length of 20characters. The Language group Basic Latin shows only a variation in results.As future work, other languages can be considered. The system can be trained by more languages. An approach that can increase the accuracy of group Basic Latin can also be considered.

References

[1] J. K. Sarungbam, B. Kumar and A. Choudhary, “Script identification and language detection of 12 Indian languages using DWT and template matching of Frequently Occurring Character(s)”, 5thInternationalConference(Confluence),2014.

[2] Zhou, L., Ping, X.J., Zheng, “Script Identification Based on Wavelet Energy Histogram Moment Features”, IEEE Journal and Magazines, 2010.

[3] Nathaniel Oco, JoelIlao, Rachel Edita Roxas, “Measuring Language Similarity using Trigrams”, International Conference on Recent Trends in Information Technology (ICRTIT), 2017.

[4] K. Ubul, G. Tursun, D. Pirlo and T. Yibulayin, “Script Identification of Multi Script Documents”, IEEE Access,vol. 5,pp.6546-6559.

[5] D. Kosmajac and V. Keselj, “Language identication in multilingual, short and noisy texts using commom N-grams”, IEEE Conference on BigData,2017.

[6] Unicode Regular Expression Guidelines, https://www.regular expressions.info/unicode.html

[7] Regex Tutorial - Unicode Characters and Properties, https://www.regularexpressions.info/unicode.html

First Author Presently pursuing Post Graduation in Computer Science and Engineering.Completed Bachler Degree in Information Technology in 2017 from the University of Calicut.

Second Author Working as Assistant Professor in Computer Science and Engineering Department, MES College of Engineering, Kuttippuram, India.