Three-stage Short Text Language Identification Algorithm

Maimaitiyiming Hasimu[1,2,3,*], Wushour Silamu[1,2]
[1] School of Information Science and Engineering, Xinjiang University
Urumqi, Xinjiang 830046, China
[2] Key Multi-lingual Laboratory of Xinjiang, Urumqi, Xinjiang 830046, China
[3] Hotan Teachers College, Hotan, Xinjiang 848000, China
*Correspondence: mamtimin116@163.com, wushour@xju.edu.cn

JDIM
Journal of Digital Information Management

ABSTRACT: *Text on the internet is written in different languages and scripts, and a language identification system is used to analyze and identify them. To improve the performance of text language identification, this paper proposes a three-stage short text language identification algorithm. The script of a given text is identified in the first stage of the algorithm. The language group to which it belongs, consisting of languages written in the same script, is identified in the second stage. In the third stage, the specific language of the given text is recognized from within the language group. Experimental results showed that our proposed method improves the accuracy of text language identification systems stage by stage, reduces the time and the size of the feature set needed to make a prediction, and achieves optimal accuracy.*

Subject Categories and Descriptors
[H.5.2 User Interfaces]: Natural language; **[I.2.7 Natural Language Processing]**

General Terms:
Language Processing, Language Identification

Keywords: Language Identification, Character n-gram, Script Identification, Language Group Identification

Received: 24 September 2017, Revised 28 October 2017, Accepted 7 November 2017

1. Introduction

Text language identification (TLI) is the process that attempts to classify a text in a language to one in a pre defined set of known languages [1]. TLI is often the first step in many text processing systems. It is widely used in text mining, information retrieval, speech processing, and machine translation [2-4]. Although TLI is often portrayed as a solved problem, there is considerable room for improvement in terms of selecting, reducing, and weighting features, reducing training and testing times, and increasing the accuracy of the identification of similar languages.

Languages are written in different scripts. Each script in Unicode has a defined code range. This information helps us identify different parts of a script within a document [5]. Scripts are easy to identify based on form because their characters have different code points in Unicode. Hence, there is no need to analyze all languages in TLI when identifying a language in text: the script of the text can be identified in advance, making it easier to identify the language from among a group of languages written in the same script.

Languages belong to different families or groups. Languages in a group are related to one another as they have a common ancestral or parental language, and errors in TLI often occur in the case of similar languages in the same language group. Languages in such a family are similar in their vocabulary and structure [6, 32]. We can use this fact to discriminate different language groups written in the same script to narrow the range of identification of TLI.

Social media has become ubiquitous in recent years, and its content often consists of short multilingual text. Improving the accuracy of TLI can help the acquisition, search, and analysis of social media data. The main desiderata of a TLI system are high-speed real-time processing, efficiency, minimal storage, and robustness against textual errors [7]. In light of these preferences, we propose a three-stage short TLI algorithm in this paper. The script of a given text is identified in the first stage, and the language group to which the text belongs—consisting of languages written in the same script—is detected in the second. In the third stage, the specific language of the text within the language group is recognized. This procedure helps shorten the range of identification, and can help reduce processing time, remove noisy features, reduce the size of the feature set, and hence improve identification accuracy.

2. Related Work

Language identification (LI) is generally viewed as a form of text categorization. The authors of [8,9] provided a survey of research on the identification of written languages. In TLI, statistical models can be generated using the number of words [4] or letters in the given text [2], n-gram statistics [3,10], or a combination of the two [2]. Statistical methods require no prior linguistic knowledge and are highly accurate. The dominant statistical approach used in the literature is the character-based n-gram model, which is superior to the word-based model for small text fragments and performs equally well on large fragments. It is also tolerant to errors in text, and is easy to create and compute for any given text. Hence, most TLI systems use character n-grams [2,3,10]. We have therefore restricted the feature sets we use to character trigrams.

Many TLI studies have been carried out on a large number of languages with different scripts. The authors of [11] used the character n-gram language profile based on the most frequent character n-grams in each language. They introduced an ad-hoc "out-of-place" ranking distance to classify specific texts as belonging to a language. This system recorded an accuracy of 99.8% with an n-gram profile of 400 and text consisting of more than 300 words in 14 Indo–European languages, but accuracy decreased when identifying short texts. In [12], the authors compared eight distance measures for LI using combinations of different types of character n-grams. They analyzed 38 languages with different scripts and belonging to different language groups, and compared corpora of different sizes. They verified that decreasing text size can reduce the accuracy of TLI. The authors of [13] compared three distance measure-based methods, naïve Bayes, and a support vector machine with three datasets containing different numbers of languages and sizes of documents. Experimental results showed that the TLI task becomes considerably more complex for larger numbers of languages, shorter documents, higher class skew, and multilingual documents. The author of [10] used the character n-gram model to identify 923 languages, with impressive results. Experimental results verified that confusion errors often occur among languages in the same family. To improve the accuracy of TLI online, the authors of [14] proposed a byte-sequence-based HTML parser and an HTML character entity converter for webpages prior to LI. They tested their method on webpages featuring 182 languages, where TLI accuracy increased from 86.99% to 94.04%. The authors of [4] proposed a word length algorithm for the identification of under-resourced languages. They tested it on 15 languages, and found that confusion errors occurred among closely related languages. In [2], the authors proposed five high-frequency approaches and conducted comparative tests using 11 similarity measures combined with several types of character n-grams. Their datasets consisted of 32 languages, and their experimental results verified that LI is more accurate on small datasets containing fewer languages. The authors of [15] used relative entropy to discriminate language similarity; they analyzed 27 languages written in the Roman script, and the results showed that increasing the size of training data can increase the accuracy for bigram and unigram models; moreover, when few languages are involved, the accuracy was higher. The authors of [3] analyzed factors influencing the accuracy of text-based LI. Their datasets consisted of 11 official languages of South Africa belonging to two language families. The experimental results verified that a vast majority of errors result from confusions within the same language family. The authors of [16] used five algorithms for LI experiments on 30, 60, and 90 languages, and verified that increasing the number of languages reduces the accuracy of TLI systems, and increases the time needed for prediction and training. In [17], the authors define two quantitative distances to measure how far apart two languages are. They compare the distance between two languages in forty-four languages. They found that in many cases languages within the same family or sub-family have low distances as expected. Although the above-mentioned studies analyzed languages belonging to different scripts or families, they did not classify text into different scripts or families during training and testing.

Some researchers have studied how to cluster natural languages but their methods do not apply to TLI. The authors of [18] explored the correlation between the frequency of bigrams and trigrams for each of nine European languages. The results do tend to adhere to Indo-European family tree which have been proposed by historical linguists. In [19], the authors selected character five-grams and measured similarities among documents in 31 languages to reveal a similarity-based clustering of languages. They concluded that accurate family groups can be formed by grouping together languages with similar scores. However, they verified only their system's impressive performance at language discrimination, and did not provide any data pertaining to LI. The authors of [20] compared 108 languages and clustered them using character trigrams and the most frequently used words, and reported that the former yielded better results. The clustering results indicated that language comparisons

based on simple orthographic profiles can yield genealogical relations among languages, and can be used to detect similar languages written in the same script.

Some researchers have recently studied the identification of languages belonging to the same group. In [21, 33], the authors proposed a two-stage identification technique for similar languages. Their proposed language group prediction algorithm predicts the group to which a given language belongs, and then discriminates among languages within it. This language group classifier used character four-grams as features, and exhibited excellent performance. However, its prediction time was longer in tests than in training. The authors of [22] proposed a similar technique that uses a simple token-based maximum-entropy classifier to predict language groups. They did not provide any testing data to evaluate the efficiency of their method. On the Internet, similar languages are often presented as mixed with other languages belonging to same group. Hence, the above-mentioned approaches cannot be directly applied to TLI tasks.

Every script in Unicode has its own code range. The authors of [5] used this to traverse every letter in a document to find the starting and ending code points in a given script, detect different parts of a script in the document, and distinguish the language used in each. In LI, however, they analyzed all languages in the system rather than those using the same script.

Some studies have been conducted on the prediction, clustering, and identification of languages belonging to the same group, but there is a lack of comparative research to assess the efficiency of LI following script identification or language group prediction in TLI. Therefore, we aim to address this gap using the proposed method.

3. Preliminaries and Methods

TLI is a classification process, and our proposed method performs this classification in three stages: script identification, language group identification (LGI), and LI.

3.1 Script Identification

Each script in Unicode has a defined code range, and this allows us to detect different parts of a script in text or a sentence. Unlike the algorithm proposed in [5], we use the regular expression matching method to identify different scripts. Based on the code range of scripts in Unicode, we created a regular expression for every script. The proposed script identification stage consists of the following steps:

Step 1: Remove punctuation and nonalphanumeric items from the text.

Step 2: Ensure that the remaining text matches the regular expression of the script.

Step 3: Calculate the length of each matching result and, if the length is nonzero, save it in a list. Sort the list in order of descending length.

Step 4: Select the top item in the list as the main script of the text and return it for use in the LGI and the LI stages.

3.2 Classification Method and Evaluation

A range of classification algorithms have been used for TLI. Of these, support vector machines (SVMs), multinomial naïve Bayes (NB), and logistic regression (Maxent) have yielded the best results [3,21,22]. Therefore, we chose these three classifiers for LGI and LI. The SVM is among the most effective classification algorithms and linear SVM is one of the most successful text classification algorithms [23]. The naïve Bayes classifier is a simple probabilistic classifier based on Bayes' theorem. Multinomial NB is a widely accepted model for text classification [24]. In experiments, we used scikit-learn's multinomial NB, LinearSVC, and the logistic regression toolbox with default parameter settings, excluding $C = 180$ for LinearSVC.

To test the proposed method, we used the F1 score widely used for text classification. The F1 score is defined as the harmonic mean of precision and recall. Precision is defined as the ratio of correct categorization of text to the total number of attempted classifications. Recall is the ratio of correct classifications of text to the total numbers of labeled data items in the test set. A good classifier is assumed to have a high F1 score, which indicates that the classifier performs well with respect to both precision and recall:

$$F_1 = \frac{2*precision*recall}{precision+recall} \quad (1)$$

3.3 Character-level n-gram Feature

An n-gram is a sequence of n consecutive letters. The n-gram-based approach for TLI divides text into character strings of equal size [11]. It is assumed that some languages use certain n-grams more frequently than others. This idea is based on Zipf's law, which states that the size of the r-th largest occurrence of an event is inversely proportional to its rank r [25]. An example of the decomposition of the sentence "good boy" into character n-grams is shown in Table 1. The symbol "-" represents space and is used to capture the beginning and end of words.

N-gram type	N-gram
Unigram	g, o, o, d, b, o, y
Bigram	-g, go, oo, od, d-, -b, bo, oy, y-
Trigram	-go, goo, ood, od-, d-b, -bo, -boo, boo, ook, ok-
Quadgram	-goo, good, ood-, od-b, d-bo, -boy, boy-

Table 1. Example of the decomposition of a sentence into character n-grams

4. Experiments and Results

All experiments were conducted using software written in Python 3.5. The execution environment was Microsoft Windows 10 on a computer with 32 GB of RAM and a 3.4 GHz CPU.

4.1 TLI Corpora

We selected the Leipzig Corpora Collection [26,27]. The corpora were identical in format, and similar in size and content. They contained randomly selected sentences in the language of the corpus. The sources were newspapers or text randomly collected from the Web and split into sentences.

In this study, we analyzed three widely used scripts: Latin, Cyrillic, and Arabic. According to their use on Wikipedia [28-30], we selected 51 widely used languages shown in Table 2, with 10,000 sentences for each. Of these, 49 languages were selected from the Leipzig Corpora Collection and the other two were based on Kazakh and Kirghiz, which have Arabic script (bolded items in Table 2). Sentences in these two languages were collected from relevant websites. We also used language family-related information on Wikipedia [6] and divided languages using the same script into different language groups. Tables 2–5 provide the description of the datasets. In Table 2, we list the names of the languages, their relevant ISO language code as described in [31], and the average sentence length in the corpora for each language; the minimum length for most languages was approximately 20 characters.

Language	Code	AVG	Min	Language	Code	AVG	Min
Afrikaans	afr	101	19	Uzbek	uzb	114	20
Azerbaijani	aze	109	18	Vietnamese	vie	111	20
Catalan	cat	133	19	Bashkir	bak	95	20
Czech	ces	107	20	Belarusian	bel	101	18
Danish	dan	114	19	Bulgarian	bul	110	20
German	deu	119	20	Chuvash	chv	82	20
English	eng	125	20	Kazakh	kaz	119	20
Faroese	fao	101	20	Kirghiz	kir	104	19
French	fra	127	22	Macedonian	mkd	126	19
Western Frisian	fry	90	20	Mongolian	mon	105	18
Indonesian	ind	117	19	Ossetian	oss	96	20
Icelandic	isl	99	35	Russian	rus	115	20
Italian	ita	126	19	Yakut	sah	87	20
Malay	msa	143	19	Serbian	srp	105	19
Dutch	nld	102	20	Tatar	tat	113	20
Norwegian Nynorsk	nno	104	20	Tajik	tgk	121	18
Norwegian Bokmål	nob	114	20	Ukrainian	ukr	113	20
Polish	pol	106	20	Arabic	ara	120	19
Portuguese	por	107	20	**Kazakh* (ara)**	**kaz***	132	11
Romanian	ron	124	18	**Kirghiz* (ara)**	**kir***	55	11
Slovak	slk	111	20	Persian	fas	112	19
Slovenian	slv	121	20	Kurdish	kur	175	21
Spanish	spa	133	20	Pushto	pus	109	19
Swedish	swe	109	17	Uighur	uig	120	20
Turkmen	tuk	104	20	Urdu	urd	115	19
Turkish	tur	108	16				

Table 2. Average lengths of sentences of every language in the corpora (unit is character)

Language Group	ISO Code List
Indo–European/Germanic	dan, fao, isl, nno, nob, swe, afr, deu, eng, fry, nld
Altaic/Turkic	tur, tuk, uzb, aze,
Indo–European/Italic	cat, fra, por, spa, ita, ron,
Indo–European/Balto–Slavic	slv, pol, cze, slk
Austronesian/Malayo-Polynesian (MP)	msa, ind
Austroasiatic/Vietic	Vie

Table 3. Latin script dataset

Language Group	ISO Code List
Indo–European/Indo–Iranian	fas, kur, pus, urd
Altaic/Turkic	uig, kaz*, kir*
Afro–Asiatic/Semitic	ara

Table 4. Arabic script dataset

Language Group	ISO Code List
Indo–European/Balto–Slavic	bel, rus, ukr, bul, mkd, srp
Altaic/Turkic	chv, sah, bak, kaz, kir, tat
Indo–European/Indo–Iranian	tgk, oss
Altaic/Mongolic	mon

Table 5. Cyrillic script dataset

4.2 Script Identification

To evaluate our script identification method, we estimated the main script of each sentence and calculated the script identification results for the corpus of each script. Since our data was extracted from the Leipzig Corpora Collection, newspapers, and websites, some sentences might have contained the contents of two or more scripts. Tables 6–8 show the results of the identification of the main script of the sentences. We find that sentences in Arabic and Cyrillic contained some contents from Latin script, and those in Latin contained contents from Cyrillic script. Hence, we needed to identify the main script of a sentence before identifying its language and remove content from other scripts from the sentence. Our script identification algorithm yielded ideal efficiency in terms of time, taking only a few microseconds to identify the scripts of 510,000 sentences. This did not influence the training and testing times of the LI system.

ID	Latin	Cyrillic	ID	Latin	Cyrillic	ID	Latin	Cyrillic	ID	Latin	Cyrillic
tur	1.0	0	nno	1.0	0	nld	1.0	0	ind	1.0	0
tuk	1.0	0	nob	1.0	0	cat	1.0	0	slv	1.0	0
uzb	0.7525	0.2475	swe	1.0	0	fra	1.0	0	pol	1.0	0
aze	1.0	0	afr	1.0	0	por	1.0	0	ces	1.0	0
dan	1.0	0	deu	1.0	0	1.0	1.0	0	slk	1.0	0
fao	1.0	0	eng	1.0	0	ita	1.0	0	vie	1.0	0
isl	1.0	0	fry	1.0	0	msa	1.0	0	ron	0.9999	0.0001

Table 6. Script identification results for the Latin dataset.

ID	Cyrillic	Latin	ID	Cyrillic	Latin	ID	Cyrillic	Latin	ID	Cyrillic	Latin
chv	0.9918	0.0082	kir	0.0991	0.9009	mon	0.9949	0.0051	bul	0.9991	0.0009
sah	0.9964	0.0036	tat	0.9999	0.0001	bel	0.9997	0.0003	mkd	0.9984	0.0016
bak	0.8665	0.1335	tgk	0.9373	0.0627	rus	0.9956	0.0044	srp	0.9948	0.0052
kaz	1.0	0	oss	0.9984	0.0016	ukr	0.9993	0.0007			

Table 7. Script identification results for the Arabic dataset

ID	Arabic	Latin	Cyrillic	ID	Arabic	Latin	Cyrillic	ID	Arabic	Latin	Cyrillic
ara	0.9964	0.0036	0	pus	0.9946	0.0054	0	Kaz* (ara)	1.0	0	0
fas	0.9952	0.0048	0	urd	0.9978	0.0022	0	kir * (ara)	1.0	0	0
kur	1.0	0	0	uig	0.9987	0.0011	0.0002				

Table 8. Script identification results for the Arabic dataset

4.3 Language Group Identification (LGI)

LGI consisted of preprocessing, feature selection, feature weighting, and training and testing.

4.3.1 Preprocessing

To evaluate the proposed LGI, we used the 10-fold classification method. We used sklearn's Kfold toolbox to split the corpus of each language into 10 parts, and conducted training and testing 10 times. Nine blocks were used for training and other for testing each time. The average test score was used as the final test score for classification. Following this, we cleaned the corpus of each script, and removed contents from other scripts as well as nonalphanumeric items from the text. We then extracted trigrams or bigrams to create an n-gram profile for each language group. Tables 9–10 show the number of trigrams for different sizes of corpora for each script. We can conclude from this that increasing the number of sentences in a language corpus increases the number of n-grams. The number of n-grams of every script was smaller than that of the entire TLI system. The number of n-grams of every language group was smaller than that of the relevant script as well. Narrowing the classification range reduced the number of combinations of n-grams, and helped remove noise and reduce feature size.

4.3.2 Feature Selection and Weighting

In this work, we selected the frequency distribution of n-grams as the feature selection method for LGI and LI. We calculated the frequency of each n-gram in the corpus of

Language corpora size (sentence number)	500	1000	2000	3000	4000
Arabic script (includes 8 languages)	28252	34892	42105	46685	50673
Cyrillic script (includes 15 languages)	21770	25079	28754	30922	32639
Latin script (includes 28 languages)	34575	41506	49022	53821	57487
LI in the entire TLI (includes 51 languages)	93786	116769	145253	164850	177285

Table 9. Number of trigrams for different sizes of corpora

Group (language number)	Trigram Number	Group (language number)	Trigram Number
Indo–European/Germanic (11)	19207	Indo–European/Balto–Slavic(4)	15248
Altaic/Turkic (4)	15328	Austronesian/MP (2)	5735
Indo–European/Italic (6)	12871	Austroasiatic/Vietic (1)	7514

Table 10. Number of trigrams for different language groups of Latin script

each language group, organized the n-grams in reverse order according to their frequencies, and selected the top k n-grams as features. Following feature selection, we used the occurrence of a term in a sentence as the weighting method for both LGI and LI, and calculated the occurrence of each feature in a sentence by converting the sentence into a vector:

$$tf(t_i) = Max_{j \in LG}\, t_{ij} \quad (2)$$

$$t_{ij} = \frac{\text{Num}\,(n\_gram_{ij})}{\text{Sum}_j} \quad (3)$$

4.3.3 Training and Testing

To investigate the effect of corpus size on LGI performance, we used five sizes of sentences in each language by employing bigrams and trigrams. From Tables 11 and 12, we can conclude that an increase in sentence size in the training corpora improved LGI performance in the Maxent classifier. However, once corpus size reached 2,000 sentences in a language, LGI improved only slightly for additional sentences when using both bigram and trigram features. A similar situation occurred with both the NB and the SVM classifiers. We only provide the results for the Maxent classifier owing to limitations of space. To test the effect of LGI on TLI on the same corpus size, for the remainder of this paper, we consider 2,000 sentences in each language for LI.

Tables 13 and 14 show the results of LGI on the three scripts when corpus size was 2,000 sentences in each language. The results verified that increasing feature size can improve classification accuracy. When the size of the features grew to a certain point (the feature size shown in boldface score in Tables 13 and 14), accuracy only improved slightly or even decreased, when feature size was increased to several times that of the original. This is described here as the "optimal feature size (OFS)." The accuracy of Maxent was higher than that of naïve Bayes and SVM, and the results using bigram features were similar to those obtained using trigram features, except that the OFS in the former was relatively shorter than that in the latter. Hence, in our experiments, we selected the Maxent classifier and bigram features for LGI.

To analyze the number of sentences misclassified into other language groups than the correct one, we inspected confusion matrices 1 and 2 while training the Maxent classifier, where the size of the bigram features was equal to the OFS. In matrix 1, only 13 sentences in 15 languages of a total of 3,000 test sentences were misclassified into other language groups, and this is less than one per language on average. In matrix 2, only 29 sentences of 5,600 in 28 languages were misclassified. The classification accuracy of languages using the Arabic script was very high, reaching 100% with a feature size of only 700. For languages using the Latin script, sentences were often misclassified into the Germanic language group in

Script	Corpora Size	Feature Size									
		100	300	500	700	800	1000	1200	2100	2400	3300
Arabic	500	0.998	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
	1000	0.998	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
	1500	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
	2000	**0.999**	**0.999**	**0.999**	**1.000**	**0.999**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	2500	0.999	0.999	0.999	1.000	1.000	1.000	1.000	0.999	0.999	0.999
Cyrillic	500	0.978	0.989	0.993	0.993	0.993	0.993	0.993			
	1000	0.979	0.992	0.994	0.994	0.994	0.994	0.994			
	1500	0.977	0.991	0.993	0.994	0.994	0.994	0.994	0.994		
	2000	**0.976**	**0.992**	**0.994**	**0.994**	**0.995**	**0.995**	**0.995**	**0.995**		
	2500	0.975	0.991	0.994	0.994	0.994	0.994	0.994	0.994		
Latin	500	0.969	0.987	0.989	0.991	0.991	0.991	0.992	0.992	0.992	0.992
	1000	0.968	0.989	0.992	0.992	0.993	0.993	0.993	0.994	0.994	0.994
	1500	0.969	0.990	0.993	0.994	0.994	0.994	0.994	0.995	0.995	0.995
	2000	**0.970**	**0.990**	**0.993**	**0.994**	**0.994**	**0.994**	**0.995**	**0.995**	**0.995**	**0.995**
	2500	0.971	0.991	0.993	0.994	0.994	0.995	0.995	0.995	0.995	0.995

Table 11. LGI F1 score when using bigrams in Maxent

matrix 2. A similar situation occurred in the case of languages using the Cyrillic script in matrix 1, where sentences were often misclassified as belonging to the Slavic language group. As Germanic and Slavic languages are widely used in the world, other language groups contain words in these languages, thus creating noise during LGI. From our analysis, we conclude that our proposed LGI algorithm has a very high efficiency and can be used in LI projects.

Script	Corpora Size	Feature Size									
		400	600	800	1000	1500	2000	3000	4000	8000	12000
Arabic	500	0.995	0.996	0.996	0.996	0.996	0.996	0.997	0.997	0.997	0.997
	1000	0.997	0.997	0.997	0.997	0.998	0.998	0.998	0.998	0.998	0.998
	1500	0.998	0.998	0.998	0.998	0.999	0.999	0.999	0.999	0.999	0.999
	2000	**0.998**	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**
	2500	0.998	0.998	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
Cyrillic	500	0.987	0.990	0.991	0.992	0.993	0.993	0.994	0.994	0.995	0.995
	1000	0.988	0.992	0.993	0.994	0.995	0.995	0.996	0.996	0.996	0.996
	1500	0.985	0.990	0.992	0.993	0.995	0.995	0.995	0.995	0.996	0.996
	2000	**0.986**	**0.990**	**0.992**	**0.993**	**0.994**	**0.995**	**0.996**	**0.996**	**0.996**	**0.996**
	2500	0.985	0.990	0.992	0.993	0.994	0.995	0.996	0.996	0.996	0.996
Latin	500	0.974	0.983	0.985	0.987	0.991	0.992	0.992	0.993	0.994	0.994
	1000	0.976	0.985	0.988	0.990	0.992	0.993	0.994	0.995	0.995	0.995
	1500	0.979	0.986	0.989	0.991	0.993	0.994	0.995	0.996	0.996	0.996
	2000	**0.980**	**0.988**	**0.990**	**0.992**	**0.994**	**0.995**	**0.996**	**0.996**	**0.997**	**0.997**
	2500	0.981	0.988	0.990	0.992	0.994	0.995	0.996	0.997	0.997	0.997

Table 12. LGI F1 score when using trigrams in Maxent

Script	Method	Feature Size										
		200	400	600	700	800	1000	1200	1500	2100	2400	3300
Arabic	MaxTent	0.999	0.999	0.999	**1.000**	0.999	1.000	1.000	1.000	1.000	1.000	1.000
	NB	0.997	0.997	0.998	0.998	0.998	**0.999**	0.999	0.999	0.999	0.999	0.999
	SVM	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	**1.000**	1.000	1.000
Cyrillic	Maxent	0.988	0.993	0.994	0.994	**0.995**	0.995	0.995	0.995	0.995		
	NB	0.961	0.977	0.979	0.979	0.981	0.983	0.984	**0.985**	0.985		
	SVM	0.981	0.989	0.988	0.989	**0.990**	0.990	0.990	0.990	0.990		
Latin	Maxent	0.987	0.992	0.993	0.994	0.994	0.994	**0.995**	0.995	0.995	0.995	0.995
	NB	0.981	0.987	0.990	0.990	0.990	0.991	0.991	**0.992**	0.992	0.992	0.992
	SVM	0.981	0.987	0.990	**0.991**	0.991	0.991	0.991	0.990	0.991	0.991	0.991

Table 13. LGI F1 score when using bigrams in the three classification methods

Script	Method	Feature Size									
		100	300	500	600	1000	2000	3000	5000	10000	12000
Arabic	Maxent	0.992	0.998	0.998	**0.999**	0.999	0.999	0.999	0.999	0.999	0.999
	NB	0.986	0.997	0.998	0.998	**0.999**	0.999	0.999	0.999	0.999	0.999
	SVM	0.992	0.996	0.998	0.997	0.998	**0.999**	0.999	0.999	0.999	0.999
Cyrillic	Maxent	0.943	0.981	0.989	0.990	0.993	**0.995**	0.996	0.996	0.996	0.996
	NB	0.920	0.961	0.979	0.982	0.988	0.993	**0.994**	0.995	0.996	0.996
	SVM	0.933	0.975	0.984	0.981	0.988	**0.991**	0.992	0.995	0.996	0.996
Latin	Maxent	0.929	0.973	0.984	0.988	0.992	**0.995**	0.996	0.997	0.997	0.997
	NB	0.911	0.961	0.977	0.981	0.988	0.993	**0.994**	0.995	0.996	0.996
	SVM	0.910	0.964	0.980	0.981	0.983	0.991	**0.994**	0.995	0.995	0.995

Table 14. LGI F1 score when using trigrams in the three classification methods

Turkic	Iranian	Mongolic	Slavic
1194	0	0	6
3	393	0	1
0	1	199	0
1	2	0	1197

Confusion Matrix 1. LGI confusion matrix for languages using the Cyrillic script

Germanic	Turkic	Italic	MP	Slavic	Vietic
2192	0	7	0	1	0
5	794	0	1	0	0
9	0	1191	0	0	0
1	0	0	399	0	0
0	1	1	1	798	0
1	0	0	0	0	199

Confusion Matrix 2. LGI confusion matrix for languages using the Latin script

4.4 Language identification (LI)

The preprocessing for LI was identical to that for LGI. To assess the efficiency of the proposed three-stage TLI, we compared three types of TLI tests:

• LI within whole languages in the TLI system without script identification and LGI. This is referred to as "LI in the entire TLI."

• LI of languages using the same script. Having identified the script of a text, a language is distinguished from other languages using the same script. This is referred to as "LI in the same script."

• LI within a language group. Following script and language group identification, the language of the text is distinguished from those within the language group. This is referred to as "LI in the LG."

From Tables 15–19, we observe that increasing feature size can improve the accuracy of identification in the three types of TLI tests, but feature size more than the OFS (the feature size relevant to boldface score in Tables 15–19). LI accuracy improves very little, despite the feature size increasing several times. From the results of the TLI tasks, we can conclude that the SVM required more

features to reach a high score than Maxent and NB for the three kinds of TLI. Maxent was slightly better than NB for overall LI as well as "LI in the same script," but the "LI in the LG" performance of NB was better than that of Maxent. We only provide the high scores relevant to OFS (boldface and underlined numbers in Tables 21–26) for NB here owing to space restrictions.

Method	Feature Size										
	200	600	1000	2000	3000	4000	5000	6000	8000	10000	12000
Maxent	0.745	0.895	0.927	0.949	0.956	**0.959**	0.960	0.962	0.964	0.966	0.968
NB	0.733	0.887	0.919	0.944	0.951	0.955	**0.957**	0.958	0.960	0.961	0.962
SVM	0.665	0.855	0.890	0.915	0.924	0.931	0.935	**0.939**	0.942	0.947	0.951

Table 15. LI F1 score of the entire TLI system

Script	Method	Feature Size									
		200	600	1000	2000	3000	4000	5000	6000	9000	12000
Arabic	Maxent	0.972	0.985	0.985	0.987	**0.988**	0.988	0.988	0.988	0.989	0.990
	NB	0.967	0.987	0.988	**0.990**	0.990	0.990	0.990	0.990	0.990	0.990
	SVM	0.953	0.978	0.970	0.979	0.983	**0.985**	0.986	0.987	0.987	0.987
Cyrillic	Maxent	0.894	0.939	0.947	0.954	**0.956**	0.956	0.956	0.957	0.957	0.958
	NB	0.888	0.938	0.946	0.952	**0.954**	0.955	0.957	0.957	0.957	0.958
	SVM	0.859	0.913	0.925	0.930	0.928	0.937	**0.940**	0.941	0.945	0.948
Latin	Maxent	0.859	0.934	0.948	0.958	**0.964**	0.967	0.969	0.969	0.971	0.972
	NB	0.888	0.938	0.946	0.952	**0.954**	0.955	0.957	0.957	0.957	0.958
	SVM	0.797	0.897	0.913	0.929	0.941	0.949	**0.954**	0.956	0.960	0.961

Table 16. LI F1 score for the same script

Language Group	Method	Feature Size									
		100	400	600	700	800	900	1000	2000	3000	12000
Indo–Iranian	Maxent	0.967	0.974	0.973	0.974	**0.975**	0.973	0.974	0.978	0.980	0.980
	NB	0.969	0.978	0.979	**0.980**	0.980	0.980	0.981	0.981	0.981	0.981
	SVM	0.960	0.964	0.947	0.953	0.953	0.956	0.960	**0.969**	0.971	0.977
Turkic	Maxent	0.987	0.998	0.999	**1.000**	1.000	1.000	1.000	1.000	1.000	1.000
	NB	0.981	0.999	0.999	0.999	**1.000**	1.000	1.000	1.000	1.000	1.000
	SVM	0.984	0.999	0.999	**1.000**	1.000	1.000	1.000	1.000	1.000	1.000

Table 17. LI F1 score for language groups using the Arabic script

Language Group	Method	Feature Size									
		100	300	500	700	900	1000	2000	3000	10000	12000
Turkic	Maxent	0.839	0.885	0.894	0.900	0.903	**0.906**	0.906	0.908	0.905	0.907
	NB	0.827	0.870	0.885	0.889	0.890	0.892	**0.897**	0.899	0.903	0.904
	SVM	0.802	0.855	0.864	0.879	0.882	0.886	0.866	**0.878**	0.893	0.894
Iranian	Maxent	0.999	0.999	0.999	0.999	**1.000**	1.000	1.000	1.000	1.000	
	NB	0.999	**1.000**	1.000	1.000	1.000	1.000	1.000	1.000	1.000	
	SVM	0.997	0.999	0.999	0.999	**1.000**	1.000	1.000	1.000	1.000	
Slavic	Maxent	0.922	0.975	0.978	0.981	0.984	0.985	**0.988**	0.989	0.991	0.991
	NB	0.911	0.970	0.978	0.982	0.988	0.988	**0.993**	0.993	0.996	0.996
	SVM	0.892	0.957	0.961	0.969	0.974	0.975	**0.982**	0.984	0.987	0.987

Table 18. LI F1 score for the language groups using the Cyrillic script

Language Group	Method	Feature Size									
		100	500	700	800	1000	2000	3000	4000	6000	12000
Turkic	Maxent	0.958	0.987	0.988	0.989	**0.991**	0.992	0.993	0.993	0.993	0.993
	NB	0.954	0.989	0.989	0.990	0.990	**0.992**	0.993	0.993	0.993	0.993
	SVM	0.947	0.979	0.980	0.982	0.983	**0.986**	0.987	0.988	0.988	0.988
Germanic	Maxent	0.840	0.945	0.953	0.954	0.956	**0.964**	0.969	0.970	0.971	0.972
	NB	0.821	0.943	0.953	0.955	0.959	0.969	**0.973**	0.975	0.977	0.977
	SVM	0.770	0.913	0.927	0.924	0.923	0.945	**0.955**	0.959	0.962	0.963
Italic	Maxent	0.954	0.984	0.985	0.986	0.987	**0.990**	0.991	0.992	0.992	0.992
	NB	0.950	0.986	0.989	0.990	0.990	**0.993**	0.994	0.994	0.994	0.994
	SVM	0.934	0.970	0.976	0.979	0.980	**0.987**	0.988	0.988	0.989	0.989
MP	Maxent	0.724	0.805	0.813	0.819	0.831	**0.846**	0.844	0.848	0.851	
	NB	0.678	0.786	0.800	0.807	0.823	**0.854**	0.863	0.859	0.864	
	SVM	0.625	0.748	0.782	0.784	0.786	0.818	**0.823**	0.826	0.826	
Slavic	Maxent	0.940	0.983	0.987	0.987	0.988	**0.992**	0.993	0.994	0.994	0.994
	NB	0.937	0.981	0.987	0.988	0.990	**0.994**	0.996	0.996	0.996	0.996
	SVM	0.924	0.967	0.975	0.976	0.981	**0.988**	0.988	0.990	0.990	0.991

Table 19. LI F1 score of language groups using the Latin script

4.4.1 LI in Latin Script

Our TLI corpus consisted of 28 languages using the Latin script and belonging to six language groups as shown in Table 3. Of these, the Austroasiatic/Vietic language group had only one member, and was identified in the LGI stage; its F1 score on the Maxent method was 0.997, as shown in Table 20, and only one of 200 sentences were misclassified as belonging to the Germanic language group in confusion matrix 2.

From the results of comparison tests for Germanic languages given in Tables 21–23, we can observe that hierarchical LI that narrows the range of LI improved identification accuracy, LI accuracy within a language group was higher than that for language in the same script, and LI accuracy of languages using the same script was higher than the overall TLI. Hierarchical LI to narrow the range of LI can significantly reduce feature size as shown in Tables 21–23. For example, for the Germanic languages (Table 21), the OFSs were 2000, 3000, and 5000 for the three types of comparison tests.

Script	Language Group	Feature Size								
		200	600	700	800	1000	1200	2100	2400	3300
Arabic	Semitic	0.998	0.999	**0.999**	0.999	0.999	0.999	0.999	0.999	0.999
	Indo–Iranian	0.999	0.999	**0.999**	0.999	0.999	0.999	0.999	0.999	0.999
	Turkic	1.000	1.000	**1.000**	1.000	1.000	1.000	1.000	1.000	1.000
Cyrillic	Turkic	0.987	0.994	0.994	**0.995**	0.995	0.995	0.995		
	Indo–Iranian	0.984	0.993	0.993	**0.993**	0.993	0.993	0.993		
	Mongolic	0.985	0.992	0.992	**0.993**	0.993	0.992	0.993		
	Slavic	0.989	0.995	0.995	**0.995**	0.995	0.995	0.995		
Latin	Germanic	0.987	0.994	0.994	0.994	0.994	**0.995**	0.995	0.995	0.995
	Turkic	0.987	0.994	0.995	0.995	0.996	**0.996**	0.997	0.997	0.997
	Italic	0.983	0.991	0.991	0.991	0.991	**0.992**	0.992	0.992	0.993
	MP	0.992	0.995	0.995	0.995	0.996	**0.995**	0.995	0.995	0.995
	Slavic	0.987	0.995	0.995	0.996	0.996	**0.996**	0.996	0.996	0.996
	Vietic	0.995	0.996	0.996	0.996	0.996	**0.997**	0.997	0.997	0.997

Table 20. LGI F1 score for every language group when using bigrams in Maxent.

There were some highly similar languages, such as Norwegian Nynorsk (nno) and Norwegian Bokmål (nob) (Table 20), and Malay (msa) and Indonesian (ind) (Table 21). Their identification accuracy values were lower than those of languages using the Latin script as confusion errors often occurred in these cases. For example, we inspected the confusion matrix for one-time naïve Bayes classification when the feature size was 2,000. In matrix 3, eight Norwegian Nynorsk sentences of a total of 200 sentences were misclassified as Norwegian Bokmål, and 13 sentences of Norwegian Bokmål were misclassified as those of Norwegian Nynorsk language. We also investigated the confusion matrix for Malay and Indonesian using one-time naïve Bayes classification. A total of 28 Indonesian sentences out of 200 sentences were misclassified as those of Malay, and 28 Malay sentences were misclassified as those of Indonesian.

Feature Size	afr	dan	Deu	eng	fao	Fry	isl	nld	nno	nob	Swe
LI in Germanic Languages											
2000	**0.987**	**0.948**	**0.993**	**0.990**	**0.995**	**0.973**	**1.000**	**0.963**	**0.933**	**0.898**	**0.980**
3000	0.988	0.955	0.995	0.989	0.996	0.975	1.000	0.966	0.941	0.913	0.984
5000	0.989	0.968	0.996	0.989	0.997	0.975	1.000	0.967	0.948	0.928	0.988
LI in Latin Script											
2000	0.981	0.933	0.990	0.980	0.991	0.971	1.000	0.957	0.916	0.875	0.970
3000	**0.985**	**0.943**	**0.991**	**0.980**	**0.994**	**0.973**	**1.000**	**0.961**	**0.923**	**0.891**	**0.976**
5000	0.987	0.956	0.992	0.978	0.995	0.973	1.000	0.963	0.937	0.910	0.980
LI in the entire TLI											
2000	0.961	0.889	0.983	0.968	0.985	0.963	1.000	0.940	0.887	0.827	0.952
3000	0.973	0.917	0.988	0.970	0.990	0.967	1.000	0.951	0.902	0.857	0.960
5000	**0.982**	**0.933**	**0.989**	**0.977**	**0.992**	**0.972**	**1.000**	**0.959**	**0.916**	**0.876**	**0.969**

Table 21. Comparative TLI results for Germanic languages in NB.

Feature Size	Cat	fra	ita	por	ron	spa	Feature Size	ind	msa
LI in Italic languages							LI in MP languages		
2000	**0.987**	**0.997**	**0.995**	**0.995**	**0.996**	**0.986**	2000	0.856	0.851
3000	0.988	0.997	0.997	0.996	0.997	0.988	**3000**	**0.863**	**0.862**
5000	0.989	0.998	0.997	0.996	0.997	0.989	5000	0.862	0.864
LI in Latin Script							LI in Latin Script		
2000	0.981	0.993	0.988	0.990	0.989	0.980	2000	0.798	0.805
3000	**0.984**	**0.995**	**0.989**	**0.993**	**0.992**	**0.983**	**3000**	**0.812**	**0.817**
5000	0.987	0.996	0.993	0.995	0.993	0.984	5000	0.830	0.835
LI in the entire TLI							LI in the entire TLI		
2000	0.974	0.984	0.980	0.984	0.983	0.974	2000	0.740	0.759
3000	0.979	0.988	0.984	0.988	0.985	0.976	3000	0.767	0.784
5000	**0.983**	**0.993**	**0.989**	**0.992**	**0.990**	**0.981**	**5000**	**0.799**	**0.810**

Table 22. Comparative TLI results for Italic and Malayo-Polynesian (MP) languages in NB

Feature Size	Aze	tuk	tur	uzb	Feature Size	ces	pol	slk	slv
LI inTurkic languages					LI in Slavic languages				
1000	**0.984**	**0.998**	**0.984**	**0.994**	**2000**	**0.991**	**0.999**	**0.990**	**0.996**
3000	0.987	0.999	0.990	0.996	3000	0.994	0.999	0.993	0.997
5000	0.988	0.999	0.991	0.996	5000	0.995	0.999	0.993	0.997
LI in Latin Script					LI in Latin Script				
1000	0.975	0.992	0.966	0.988	2000	0.972	0.994	0.976	0.987
3000	**0.985**	**0.998**	**0.983**	**0.992**	**3000**	**0.980**	**0.997**	**0.985**	**0.992**
5000	0.987	0.998	0.987	0.993	5000	0.985	0.998	0.987	0.992
LI in the entire TLI					LI in the entire TLI				
1000	0.951	0.962	0.903	0.941	2000	0.951	0.987	0.951	0.974
3000	0.970	0.989	0.932	0.945	3000	0.959	0.993	0.965	0.981
5000	**0.977**	**0.991**	**0.937**	**0.944**	**5000**	**0.974**	**0.995**	**0.977**	**0.986**

Table 23. Comparative TLI results for Turkic and Slavic languages in NB

language	afr	dan	deu	eng	fao	fry	isl	nld	nno	nob	swe
afr	199	0	0	0	0	0	0	1	0	0	0
dan	0	186	1	1	0	0	0	0	1	11	0
deu	0	0	199	1	0	0	0	0	0	0	0
eng	0	0	0	199	0	1	0	0	0	0	0
fao	0	0	0	0	200	0	0	0	0	0	0
fry	0	0	0	0	0	194	0	5	1	0	0
isl	0	0	0	0	0	0	200	0	0	0	0
nld	4	0	0	1	0	1	0	194	0	0	0
nno	0	1	0	0	0	0	0	0	185	13	1
nob	0	6	0	0	0	0	0	0	8	185	1
swe	1	1	0	1	0	0	0	0	2	2	193

Confusion Matrix 3. LI confusion matrix for languages using the Latin script

4.4.2 LI in Cyrillic and Arabic Scripts

Our TLI corpus contained texts in l5 languages using the Cyrillic script belonging to four language groups as shown in Table 5. Of these, the Mongolian language group had only one member: Mongolian. Its F1 score on the Maxent method was 0.993, as shown in Table 20, where only one Mongolian sentence was misclassified as belonging to the Common Turkic languages on the Maxent classifier in confusion matrix 1.

Feature Size	Bak	chv	kaz	kir	sah	Tat	Feature Size	oss	tgk
LI in Turkic languages							LI in Iranian languages		
2000	**0.678**	**0.995**	**0.998**	**0.995**	**0.993**	**0.720**	**300**	**1.000**	**1.000**
3000	0.690	0.995	0.998	0.996	0.994	0.721	3000	1.000	1.000
5000	0.696	0.995	0.998	0.997	0.994	0.719	5000	1.000	1.000
LI in Cyrillic Script							LI in Cyrillic Script		
2000	0.672	0.995	0.997	0.993	0.989	0.717	300	0.991	0.981
3000	**0.672**	**0.995**	**0.998**	**0.995**	**0.989**	**0.722**	**3000**	**0.999**	**0.999**
5000	0.693	0.995	0.998	0.997	0.989	0.731	5000	0.999	0.999
LI in the entire TLI							LI in the entire TLI		
2000	0.264	0.995	0.991	0.951	0.973	0.726	300	0.960	0.819
3000	0.285	0.995	0.996	0.969	0.983	0.732	3000	0.999	0.954
5000	**0.295**	**0.996**	**0.997**	**0.978**	**0.985**	**0.731**	**5000**	**0.999**	**0.955**

Table 24. Comparative TLI results for Turkic and Iranian languages in NB

Feature Size	bel	bul	mkd	Rus	Srp	ukr
LI in Group						
2000	**1.000**	**0.989**	**0.989**	**0.992**	**0.991**	**0.995**
3000	1.000	0.991	0.988	0.994	0.991	0.996
5000	1.000	0.993	0.990	0.996	0.994	0.997
LI in Cyrillic Script						
2000	0.999	0.983	0.981	0.978	0.986	0.993
3000	**0.999**	**0.986**	**0.986**	**0.979**	**0.990**	**0.994**
5000	0.999	0.991	0.989	0.982	0.992	0.995
LI in the entire TLI						
2000	0.993	0.952	0.954	0.983	0.968	0.979
3000	0.996	0.965	0.966	0.985	0.980	0.987
5000	**0.999**	**0.977**	**0.976**	**0.990**	**0.985**	**0.991**

Table 25. Comparative TLI results for Slavic languages in NB

Language	bak	chv	kaz	Kir	sah	tat
Bak	151	0	0	0	0	49
Chv	0	200	0	0	0	0
Kaz	0	0	200	0	0	0
Kir	0	0	0	200	0	0
Sah	0	1	0	0	199	0
Tat	46	0	0	0	0	154

Confusion Matrix 4. LI in Turkic language group.

There were three different language groups of languages using the Arabic script, as shown in Table 4. Of these, the Semitic language group had only one member, Arabic. It was identified during the LGI process for the Arabic script. Its identification F1 score attained optimal accuracy.

The identification accuracy of the languages improved gradually following script identification and language group identification for languages using the Cyrillic and the Arabic scripts. From Tables 24–26, it can be seen that the OFS decreased significantly after hierarchical LI. For the Slavic languages, as can be seen in Table 25, the OFS occurred at 2000, 3000, and 5000 for three types of tests.

Highly similar languages were also present among those using the Cyrillic script; the Bashkir and Tatar languages belong to the Common Turkic language group. Confusion errors often arise between Bashkir and Tatar. For example, in confusion matrix 4 (in a one-time naïve Bayes classification when the feature size was 2000), 49 Bashkir (bak) sentences out of 200 sentences were misclassified as belonging to Tatar (tat), and 46 sentences in Tatar out of 200 sentences were misclassified as those in Bashkir.

4.5 Time Efficiency Analysis for Hierarchical LI

To evaluate the temporal efficiency of our proposed three-stage TLI, we compared the training and testing times of the three types of TLI tests when the feature size was equal to the OFS. We selected the Maxent classifier with bigrams as the feature for LGI because the Maxent classifier requires fewer features to reach its OFS when using bigrams, as shown in Table 13. We used Python's datetime Toolbox to calculate the training and testing times

Feature Size	fas	kur	pus	Urd	Feature Size	Kaz(ara)	Kir(ara)	uig
LI in Indo–Iranian					LI in Turkic			
700	**0.963**	**0.999**	**0.995**	**0.963**	**800**	**1.000**	**0.999**	**1.000**
2000	0.963	1.000	0.996	0.963	2000	1.000	1.000	1.000
5000	0.963	1.000	0.996	0.963	5000	1.000	1.000	1.000
LI in Arabic Script					LI in Arabic Script			
700	0.961	0.997	0.993	0.963	800	0.998	0.998	0.998
2000	**0.963**	**0.999**	**0.996**	**0.963**	**2000**	**1.000**	**1.000**	**0.999**
5000	0.963	1.000	0.995	0.963	5000	1.000	1.000	1.000
LI in the entire TLI					LI in the entire TLI			
700	0.952	0.994	0.984	0.958	800	0.936	0.939	0.978
2000	0.960	0.995	0.992	0.961	2000	0.990	0.989	0.994
5000	**0.961**	**0.998**	**0.994**	**0.963**	**5000**	**0.998**	**0.998**	**0.998**

Table 26. Comparative TLI results for Indo–Iranian and Turkic languages in NB

Method	Arabic script			Cyrillic script			Latin script		
	OFS	train	test	OFS	train	test	OFS	train	test
Maxent	700	00:02.9	00:00.1	800	00:05.8	00:00.1	1200	00:13.5	00:00.3

Table 27. Training and testing times for LGI for the three scripts

Method	Arabic script			Cyrillic script			Latin script		
	OFS	train	test	OFS	train	test	OFS	train	test
Maxent	3000	00:09.5	00:00.3	3000	00:30.3	00:00.5	3000	01:32.2	00:00.9
NB	2000	00:01.8	00:00.3	3000	00:04.8	00:00.6	3000	00:09.0	00:01.0
SVM	4000	00:05.5	00:00.4	5000	00:14.5	00:00.8	5000	00:33.6	00:01.4

Table 28. Training and testing times for LI in the same script

taken for classification. The unit format is minute: second. millisecond. Tables 27–31 show the values for the training and testing times in our experiments. We can conclude that the training time of the NB was significantly shorter than those of Maxent and SVM. The training time for Maxent was longer than that of the other two classifiers but the time to make a prediction (testing) was shorter for Maxent. The greater the number of members in a language group or a group of languages of the same script, the longer are the training and testing times required.

To compare temporal efficiency, we used the experimental data in Tables 27–32 and Equations 4–7 to calculate the total training and testing times for LI for the same script and the language group. Table 31 shows the total training and testing times. The results showed that the training time for the NB classifier on LI in the LG was longer than that on LI in the script and LI for the overall TLI test. We used the Maxent classifier for LGI; its training time was 00:22.2 and total training time for LI in the LG was 00:30.7 when using NB for LI. Most of the training time was consumed by the LGI process, but the training time for the NB in LI in the LG was shorter than those for the Maxent and the SVM classifiers. The training time for Maxent and SVM in LI in the LG was shorter than those on LI in the script and LI for the overall TLI test.

Comparing the times of the three types of TLI tests, we find that following script identification and LGI, language prediction time (testing time) was significantly shorter for the three classification methods. The testing time in hierarchical language classification was shorter than LI for overall TLI and LI in the same script. Testing in the same script was faster than in overall LI. The testing time of Maxent for LI in the LG was shorter than those of the NB and the SVM classifiers.

Method	Indo–European			Turkic		
	OFS	train	Test	OFS	Train	test
Maxent	800	00:00.9	00:00.0	700	00:00.5	00:00.0
NB	700	00:00.3	00:00.0	800	00:00.2	00:00.0
SVM	2000	00:01.1	00:00.1	700	00:00.3	00:00.0

Table 29. Training and testing times for LI in LG for the Arabic script

Method	Turkic			Iranian			Slavic		
	OFS	train	test	OFS	train	test	OFS	train	test
Maxent	1000	00:03.6	00:00.1	900	00:00.3	00:00.0	2000	00:03.6	00:00.1
NB	2000	00:01.0	00:00.1	300	00:00.1	00:00.0	2000	00:01.1	00:00.1
SVM	3000	00:03.1	00:00.2	900	00:00.3	00:00.0	2000	00:01.9	00:00.1

Table 30. Training and testing times for LI in LG for the Cyrillic script

Method	OFS	train	test	OFS	train	test	OFS	train	test
	Germanic			Italic			MP		
Maxent	2000	00:14.3	00:00.2	2000	00:04.1	00:00.1	2000	00:00.6	00:00.0
NB	2000	00:02.0	00:00.3	2000	00:01.1	00:00.1	2000	00:01.1	00:00.1
SVM	3000	00:05.7	00:00.3	2000	00:01.8	00:00.1	3000	00:01.0	00:00.1
	Slavic			Turk					
Maxent	2000	00:01.6	00:00.1	1000	00:00.8	00:00.0			
NB	2000	00:00.9	00:00.1	2000	00:00.7	00:00.1			
SVM	2000	00:01.1	00:00.1	2000	00:01.2	00:00.1			

Table 31. Training and testing times for LI in LG for the Latin script

LI Method	LI in LG		LI in script		LI in the entire TLI	
	train	test	train	test	train	test
Maxent	00:52.5	00:01.1	02:12.0	00:01.7	05:25.2	00:02.1
NB	00:30.7	00:01.4	00:15.6	00:01.9	00:24.4	00:02.6
SVM	00:39.7	00:01.6	00:53.6	00:02.6	01:15.2	00:02.8

Table 32. Training and testing times for LI in the three LI tests

$$Total\_train\_\text{time\_in\_script} = \sum_{script_i=1}^{3} train\_time\_in\_script_i \quad (4)$$

$$\text{Total\_test\_time\_in\_script} = \sum_{\text{script}_i=1}^{3} \text{test\_time\_in\_script}_i \quad (5)$$

$$\text{Total\_train\_time\_in\_LG} = \sum_{\text{script}_i=1}^{3}(\text{train\_time\_for\_LGI}_i + \sum_{\text{group}_{i=1}}^{m} \text{test\_time\_in\_LG}_{ij}) \quad (6)$$

$$\text{Total\_test\_time\_in\_LG} = \sum_{\text{script}_i=1}^{3}(\text{test\_time\_for\_LGI}_i + \sum_{\text{group}_i=1}^{m} \text{test\_time\_in\_LG}_{ij}) \quad (7)$$

5. Conclusions

Languages are written in different scripts, each of which has its own range in Unicode. Languages in the same language group are similar in their vocabularies and structure. We used these facts to propose a hierarchical short text LI system. When identifying a sentence as belonging to a language, our method identifies its script, the group of languages with the same script, and then the language itself within the language group. Experimental results showed that our method significantly improves LI accuracy, requires shorter training and testing times, and needs a smaller feature size to achieve optimal accuracy than traditional LI systems.

Some highly similar languages were in the same language group. Confusion errors often occurred among them, and reduced their identification accuracy compared to other languages in the same group. Other methods to improve the identification accuracy for highly similar languages need to be studied. Some noisy features were also found in both LGI and LI stages. In future work, we will examine ways to remove noisy features and select the most efficient ones.

Acknowledgment

This work was partially supported by the National Program on Key Basic Research Projects of China (2014CB340506) and the Key Project of the National Natural Science Foundation of China (61331011).

References

[1] Choong, C. Y., Mikami, Y., Marasinghe, C. A., Nandasara, S. T. (2009). Optimizing n-gram order of an n-gram based language identification algorithm for 68 written languages. *International Journal on Advances in ICT for Emerging Regions*, 2 (2) 21-28.

[2] Abainia, K., Ouamour, S., Sayoud, H. (2016). Effective language identification of forum texts based on statistical approaches. *Information Processing & Management: an International Journal*, 52 (4) 491-512.

[3] Botha, G. R., Barnard, E. (2012). Factors that affect the accuracy of text-based language identification. *Computer Speech & Language*, 26 (5) 307-320.

[4] Selamat, A., Akosu, N. (2015). Word-length algorithm for language identification of under-resourced languages. Journal of King Saud University -*Computer and Information Sciences*, 28 (4) 457-469.

[5] Hanif , F., Latif, F., Khiyal, M. S. H. (2007). Unicode Aided Language Identification across Multiple Scripts and Heterogeneous Data. *Information Technology Journal* , 6 (4) 534-540.

[6] https://en.wiktionary.org/wiki/language_family

[7] Tran, D., Sharma, D. (2005). Markov models for written language identification. *In*: Proceedings of the 12th International Conference on Neural Information Processing, p. 67-70.

[8] Garg, A., Gupta, V., Jindal, M. (2014). A survey of language identification techniques and applications. *Journal of Emerging Technologies in Web Intelligence* 6 (4).

[9] Kranig, S. (2016). Evaluation of language identification methods. Bakalárska Práca.

[10] Brown, R. D. (2012). Finding and identifying text in 900+ languages. Digital Investigation 9 (15) 534-543.

[11] Cavnar, W. B., Trenkle, J. M. (1994). N-Gram-Based Text Categorization. *In*: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US, 161-175.

[12] Singh, A. K. (2006). Study Some Distance Measures

for Language and Encoding Identification. *In*: Proceedings of the Workshop on Linguistic Distance, (July), 63-72.

[13] Boldwin, T., Lui, M. (2010). Language Identfication: The Long and the Short of the Matter. Human Language Technologies: *In*: The 2010 Annual Conference of the North American of the ACL, Los Angeles, California, (June), 229-237.

[14] Chew, C.Y.,Mikami, Y., Nagano, R. L. (2011). Language identification of web pages based on improved n-gram algorithm. *IJCSI International Journal of Computer Science Issues*, 8(3) 1694-1814.

[15] Sibun, P., Reynar, J. C. (1996). Language identification: examining the issues. *In*: Proceedings of the 5th Symposium on Document Analysis and Information Retrieval, p. 125-135.

[16] Majlis, M. (2012). Yet another language identifier. Student Research Workshop at the Conference of the European Chapter of the Association for Computational Linguistics.

[17] Gamallo, P., Pichel, J. R., Alegria, I. (2017). From language identification to language distance. *Physica A Statistical Mechanics & Its Applications*, 484, 152-162.

[18] Souter, C., Churcher, G., Hayes, J., Hughes, J., Johnson, S. (1994). Natural language identification using corpus-based models. *Hermes Journal of Linguistics* 13, 183-203.

[19] Damashek, M., (1995). Gauging similarity with n-grams: language-independent categorization of text. Science, 267(5199) 843-848.

[20] Goldhahn, D., Quasthoff, U. (2014). Vocabulary-Based Language Similarity using Web Corpora, In Proceedings of the Ninth International Conference on Language Resource and Evaluation, Reykjavik, Iceland, (May).

[21] Goutte, C., Léger, S., Carpuat, M. (2014). The NRC System for Discriminating Similar Languages. *In*: Proceedings of the First Workshop on Applying NLP Tools to Similar languages, Varieties and Dialects, Dublin, Ireland, August 23, p. 139-145.

[22] Porta, J., Sancho, J. L. (2014). Using Maximum Entropy Models to Discriminate between Similar Languages and Varieties. *In*: Proceedings of the First Workshop on Applying NLP Tools to Similar languages, Varieties and Dialects, Dublin, Ireland, August 23, p. 120-128.

[23] Uysal, A. K., Gunal, S. (2012). A novel probabilistic feature selection method for text classification. *Knowledge-Based Systems*, 36 (6) 226-235.

[24] Jiang, L., Cai, Z., Zhang H., Wang, D. (2013). Naïve Bayes text classifiers: a locally weighted learning approach. *Journal of Experimental & Theoretical Artificial Intelligence*, 25 (2) 273-286.

[25] Ha, L., Sicilia-Garcia, E., Ming, J., Smith, F. (2003). Extension of zips law to word and character n-grams for English and Chinese. *Journal of Computational Linguistics and Chinese Language Processing*, 8 (1) 77-101.

[26] Goldhahn, D., Eckart, T., Quasthoff, U. (2012). Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 languages. *In*: Proceedings of the 8th International Language Resources and Evaluation (LREC'12).

[27] http://wortschatz.uni-leipzig.de/en/download

[28] https://en.wikipedia.org/wiki/Latin_script

[29] https://en.wikipedia.org/wiki/Arabic_script

[30] https://en.wikipedia.org/wiki/Cyrillic_script

[31] http://www.loc.gov/standards/iso639-2/php/code_list.php

[32] https://simple.wikipedia.org/wiki/Language_family

[33] Khan, Imtiaz Hussain., Siddiqui, Muazzam Ahmed (2015). Do Speakers Produce Different Referring Expressions in Their Native Language Than A Non-native Language?, *International Journal of Computational Linguistics Research* 6 (2) 41-47.