Improving Langscape’s Text-based Language Identification Tool*

Melina A. Koukoutchos

A thesis submitted in partial fulfillment of the requirements for
the degree of Bachelor of Arts in Linguistics

Bryn Mawr College

December 2015

Abstract

Text-based language identification (LID) is the task of determining the language a piece of text is written in. Although modern LID tools achieve high accuracy using the widely-accepted n-gram method, there are several areas of LID that remain more difficult, particularly the task of distinguishing between closely related languages. Langscape, a project of the University of Maryland’s Language Science Center, has an LID tool that uses a variation on the n-gram method. In this thesis, I propose and test a modification to Langscape’s LID tool to improve its ability to distinguish between closely related languages.

*I would like to thank Tess Wood and Colin Phillips from the University of Maryland’s Language Science Center for allowing me to work on Langscape’s LID tool. I would also like to thank my advisor Nathan Sanders for his guidance and support. I am also thankful to Richard Wicentowski, Janna Coles, and Rachel Weissler for the valuable feedback they gave me on earlier drafts of this thesis.
Table of Contents

1 Introduction 3

2 The N-gram Method 4

3 Challenges to Language Identification 5
   3.1 Challenges with Short Texts 6
   3.2 Challenges with Colloquial Language 6
   3.3 Challenges with Multilingual Texts 8
   3.4 Challenges with Confusable Languages 9
   3.5 Challenges with a Large Number of Reference Languages 11

4 Langscape 12
   4.1 The Langscape Project 12
   4.2 Langscape’s Language Identification Tool 12
   4.3 Langscape’s Language Identification Reference Set 16

5 The Experiment: A Two-run Algorithm for Langscape’s LID 18
   5.1 Motivation for and Description of the Two-run Algorithm Approach 18
   5.2 Language Selection 19
   5.3 Results 21

6 Conclusions 30
1 Introduction

In computational linguistics, text-based language identification (LID) is the task of determining the language a piece of text is written in. LID is the initial step in language processing, which is important for information resources and has a number of applications: content analysis, topic identification, translation, media analysis, mining of consumer and political opinions, event detection, and more. LID is generally considered a largely solved task, with current LID tools consistently achieving high accuracy in identifying the language of given texts. The most widely-used approach to LID is the n-gram method (described in Section 2). The n-gram approach dates back to Cavnar and Trenkle (1994), who achieved 99.8% accuracy in discriminating among 8 languages across 3478 texts.

Despite the overall successful performance of LID, there are several areas of LID that remain more difficult. These problematic areas, discussed in Section 3, include identifying the language of shorter texts and of texts using colloquial language; dealing with texts that contain multiple languages; distinguishing between confusable, closely related languages; and performing LID with a large number of comparison languages. In Section 4, I examine the workings of the LID tool from the Langscape project from the University of Maryland’s Language Science Center. In this thesis, I propose and test a modification to the algorithm in Langscape’s LID tool to improve its accuracy in distinguishing between closely related languages by running LID twice on each input text, with different parameters between runs. The proposed experiment is discussed in Section 5.
2 The N-gram Method

The n-gram method for LID works by comparing a given input text in an unknown language to a database of known language samples. The LID algorithm examines the character sequences, or n-grams, that appear in a text sample for each language in the database. Generally, an n-gram is a consecutive string of n characters (with punctuation and spacing removed) taken from a longer string of text, though the size of the character string (and thus, the value of n) may vary between algorithms. Trigrams and quadgrams, with n values of 3 and 4 respectively, are generally accepted standards. In some studies, nonconsecutive character strings are also examined. For example, if the input text is “Today’s front page headline reads...”, and the LID algorithm uses trigrams (character sequences for n = 3), the trigrams would be \textbf{tod}, \textbf{oda}, \textbf{day}, \textbf{ays}, \textbf{ysf}, \textbf{sfr}, and so on. (I use boldface to indicate n-grams.)

Once the n-grams have been found for the entire text, the frequency for each n-gram is calculated. Some n-grams may appear multiple times within a text, while others may only appear once; for instance, \textbf{ead} appears multiple times in our example (from “headline reads”) while \textbf{sfr} occurs only once (from “todays front”; recall that spaces and punctuation are ignored when finding n-grams). So in much larger texts, many sequences will occur more than once with different frequencies. For example, the frequencies for 20 trigrams taken from the Declaration of Independence are given in Figure 1.

The sequences and their frequencies for an input text can be compared to those of the reference texts for the languages in the database. The languages that have n-grams that are most similar and occur with a similar frequency to those in the input text will be offered
as the best matches. These results are normally ranked in order of similarity with a score indicating the degree of similarity. For example, running the basic Langscape LID tool on the Declaration of Independence results in the following partial results: 1. English (score 0.681), 2. French (score 0.108), 3. Afrikaans (score 0.083), 4. Pampanga (score 0.074), 5. Danish (score 0.073), 6. Dutch (score 0.071), 7. Hakka (score 0.065), 8. Frisian (score 0.055), 9. German (0.051), and 10. Bislama (score 0.048).

3 Challenges to Language Identification

Some of the biggest challenges in LID today include identifying the language of shorter texts, texts using colloquial language, and texts using multiple languages, as well as distinguishing between genetically related languages and other “confusable” languages. Each of these problems is discussed in turn in this section, with a focus on the specific challenges they pose for the n-gram method.
3.1 Challenges with Short Texts

In the early 2000s, the challenge to using LID on shorter strings of text became a research interest, as seen in the work of Hammarström (2007), Baldwin and Lui (2010), Vatanen et al. (2010), and Carter et al. (2013). Short texts are challenging for LID using the n-gram method since there are fewer n-grams in shorter texts. As a result, there is less information for the algorithm to examine, which can make the comparison to the reference language documents less accurate. Baldwin and Lui (2010) demonstrate that LID becomes more difficult as the length of the texts decreases by testing LID on Wikipedia pages. Hammarström (2007) presents an alternative model for LID aimed at identifying the language of texts as short as one word. Hammarström’s model uses both a frequency dictionary (based on the training corpus) and an affix table that tries to identify the language of partial words by examining affixes. The model, tested using a Bible corpus for 32 languages, achieves competitively accurate results. Vatanen et al. (2010) focus on creating an LID tool aimed at texts with only 5-21 characters, which they created using a corpus of the Universal Declaration of Human Rights in 281 languages. Carter et al. (2013) observe that microblogs, social media sites to which users make short, often frequent posts, present a two-fold challenge for LID because not only are the texts shorter in length but the texts also typically use less formal language, a second problem for LID.

3.2 Challenges with Colloquial Language

Difficulties with texts using colloquial language, such as internet pages, blogs, and social media posts, may occur when an LID tool uses a reference set consisting of language
documents using formal language that has been structured and edited, uses consistent
standardized spelling conventions, and uses vocabulary more traditional in formal registers.
When this is the case, LID works successfully for documents that are similarly structured
and edited but finds difficulties with texts using less formal language, which can be rather
different and more variable than the language of formal texts. For instance, French and Latin
vocabulary is more frequent in more formal registers in English. In addition, in informal
contexts people may use features never used in formal documents, such as abbreviations,
slang, deliberate or accidental misspellings, or emoticons. For example, in modern informal
texts, the exclamation *omg* is used frequently in place of *oh my god*, and the slang truncation
*whatevs* may be used in lieu of *whatever*. There are often many variations in spellings of the
same word or expression as well, making informal texts more variable in one sense. For
instance, the expression *oh my god* may be used directly, but in addition to the abbreviation
*omg*, numerous variations like *ohh myyy goddd* and *oh mah gawd* are also found. The
differences in vocabulary and spelling or the use of abbreviations lead to a change in the
appearance of certain character sequences as well as the frequency with which certain
character sequences occur in texts using more informal language. The disparity between
the character sequences of the formal language documents in the reference set and the less
formal input texts can lead to inaccurate language matches under the n-gram method if the
reference documents and the text to be identified are in different registers.

The challenge of LID on texts using colloquial language interests current researchers
because texts of this variety are prominent on the internet and can provide a vast amount
of information if the content can be accessed and understood, which requires LID. Martins
and Silva (2005) modify the n-gram approach for use on web documents. Testing on 6000
documents in 23 languages, they obtain between 80% and 100% accuracy depending on the language. Carter et al. (2013) explore ways to improve accuracy of LID on microblogs by incorporating currently unused data into the language matching. Such data includes information like the language a particular blogger used previously, the language of webpages that the blogger links to, and the language used by other bloggers who have conversations with the blogger.

3.3 Challenges with Multilingual Texts

With over 7000 languages spoken across the world and technology like the internet making language contact increasing frequent, it may not be surprising that texts using multiple languages are appearing with increasing frequency. Multilingual speakers may code-switch, that is, they may alternate between two or more languages or language varieties within a single conversation. Speakers code-switch for many reasons. For instance, Holmes (2013) explains that code-switching may be used to signify membership to a particular group or address a specific topic, or may be the result of lexical borrowing or a change in the relationship between speakers or the formality of their interaction. Multilingual texts create a problem for the n-gram method since the character sequences in the text are from more than one language. According to Řehůřek and Kolkus (2009) this may even result in identifying the input text as being written in a language that does not appear in the text at all. The most common approach to adjusting the n-gram method of LID in order to deal with texts in multiple languages involves segmenting the input text into (presumed) monolingual blocks. This segmentation has been examined both at word level, as done by Das and Gambäck (2014), and by character sequences, as done by Řehůřek and Kolkus (2009).
3.4 Challenges with Confusable Languages

The last major challenge to modern LID is distinguishing between “confusable” languages, which I define as languages that are easily misidentified as each other by LID tools. Due to the nature of examining the character sequences used in each language, the n-gram approach to LID is extremely successful in distinguishing between languages that use disparate character sets, like Chinese and Arabic. When dealing with languages using somewhat similar character sets, LID can still be successful if the specific character sequences used in each language are very dissimilar, like in English and Hungarian. Those languages that use identical or similar character sets and have similar n-gram frequencies are confusable languages for LID.

Sets of confusable languages typically include closely-related languages in the same language family or subfamily. Closely-related languages often have grammatical and lexical overlap, which leads to the presence of similar character sequences and character sequence frequencies. For example, Iranian languages like Persian and Dari or Scandinavian languages like Norwegian, Danish and Swedish, are more difficult for LID tools to distinguish between than English and Hungarian, which use a similar character set but have dissimilar character sequences. The degree of “closeness” between two languages that are confusable for LID may vary, but typically confusable languages are in the same immediate subfamily. Attempts to improve LID accuracy in distinguishing between close languages have been made in small scale experiments, as seen in Ljubešić et al. 2007, Huang and Lee 2008, Tiedemann and Ljubešić 2012, and Malmasi and Dras 2015a.

Ljubešić et al. (2007) present a tool aimed to distinguish specifically between Croatian,
Serbian, and Slovenian and found that the tool outperforms current tools. Huang and Lee (2008) examine ways to distinguish between three varieties of Chinese. Tiedemann and Ljubešić (2012) advocate for a token-based approach, which allowed them to increase accuracy in distinguishing between Bosnian, Croatian, and Serbian by 9% over the best public LID tools. Malmasi and Dras (2015a) conduct an experiment to improve LID distinction between Persian and Dari by increasing the size of the language documents in the reference set for both languages by thousands of characters. This provides the algorithm with more data about the character sequences used in each language, which ultimately allow it to distinguish between Persian and Dari with 96% accuracy on 28000 sentences. Although their method was successful, it may not be practical to implement for a larger set of languages due to the amount of manual work required to build the large language documents.

While many confusable languages for LID are closely related languages, not all closely related languages are confusable for LID. Since clusters of confusable languages have character sequences that are relatively similar, languages that are closely related but use different writing systems will not be confused by LID using the n-gram method, which specifically examines the characteristics of written text. For example, Urdu and Hindi are very closely related languages but are not problematic for LID tools because Urdu uses the Arabic script while Hindi uses the Devanagari script. The problem of distinguishing between confusable languages continues to be one of the biggest challenges to modern LID, particularly for tools using reference sets for a large number of languages. This problem inspired the experiment on Langscape’s LID presented in this paper.
3.5 Challenges with a Large Number of Reference Languages

LID becomes increasingly difficult as the number of languages being compared to the input text increases. An increase in the number of languages in the reference database not only requires the algorithm to distinguish between more languages but also often introduces languages that are confusable with other languages in the set, which present further difficulties. Baldwin and Lui (2010) claim the idea of LID being a "solved task" is a misconception formed from the success of isolated experiments using a small number of languages. They demonstrate that LID becomes much harder for a larger number of languages through experimentation with seven LID models using three datasets in their experiment: one with 10 languages represented, one with 60 languages, and one with 67 languages. Baldwin and Lui show that accuracy in LID decreases with a change in less than 100 languages, but the difficulties grow as the number of languages increases by several hundred or more. Xia et al. (2009) use the n-gram algorithm of Cavnar and Trenkle (1994) to show that the 99.8% accuracy achieved when testing on news articles in eight languages drops to 1.66% when the number of languages reaches a few hundred.

Xia et al. also note that the amount of language data used for training the algorithm greatly affects accuracy. As a result, these LID methods do not work as well for so-called low-density languages, those for which few online or computational resources exist (Megerdoomian 2009). Therefore, LID accuracy can greatly depend on the number of languages represented as well as the specific languages represented. In distinguishing between confusable languages, a successful technique used by Malmasi and Dras (2015b) was to increase the amount of training data for each language. This method, though
successful, is hard to implement for a larger number of languages since it is labor intensive and difficult for low-density languages.

4 Langscape

4.1 The Langscape Project

Langscape, a project of the University of Maryland’s Language Science Center, has an interactive map that allows users to explore the languages spoken across the globe and find resources for learning about them. General information about the languages and their speakers can be found along with sample texts, speech recordings, linguistic analysis, word lists, and more. The site incorporates data from many sources, including SIL's Ethnologue, the Rosetta Project, and the University of Maryland Center for Advanced Study of Language. The goal of the project is to increase understanding of language diversity, specifically by expanding the data available through Langscape and developing tools for applying the data in new ways in different areas, such as research, education, technology, and government. Langscape’s LID is one of these tools.

4.2 Langscape’s Language Identification Tool

Langscape’s LID tool uses a variation on the traditional n-gram method. The description here is based on the explanation given in Huffman 1995. The first step is creating document vectors for each language document in the reference set. Essentially, each document vector characterizes the n-grams (character sequences) that are found in the document and counts the frequency with which each n-gram occurs.
Let us consider the sample text “The time I met...” to demonstrate how to create a document vector. The algorithm works its way through the document examining each \( n \)-gram. For this example, we will use trigrams. When the first trigram the is found (“The time I met”), an entry is created in the document vector corresponding to the and is filled with the number 1 (since one of that trigram has been found thus far). The vector is \(<1>\) so far. Then the trigram het is found (“The time I met”). This trigram has not appeared previously in the document, so a new entry is created in the document vector with the value 1 so that the vector is now \(<1,1>\). Likewise, entries with the value 1 are created for eti, tim, ime, mei, and eim to produce \(<1,1,1,1,1,1,1>\). Then when ime is encountered for the second time, the previous entry created for ime is incremented by one, changing the vector to \(<1,1,1,1,2,1,1>\). This process continues until all trigrams in the document have been tabulated. So the final vector produced from the string “The time I met...” would be \(<1,1,1,1,2,1,1,1>\), with each entry corresponding to a specific trigram. (The order of the trigrams is stored in a separate vector.)

Once the frequency count for each trigram has been added to the document vector, the vector is normalized by dividing each entry by the total number of trigrams in the document. In our example there were nine trigrams in the string so the final vector would be \(<\frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{2}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}>\), so that the entries of the vector sum to one. This process is completed for every language document in the reference set. So in a reference set with 50 languages, the algorithm will produce 50 such document vectors. These document vectors do not need to be recalculated unless the reference texts are changed. The same process is also used to create what are called test vectors, which characterize the trigram frequencies of the input texts of which we are trying to identify the language.
Once the document vectors and test vectors have been calculated, the next step is the subtraction of a centroid vector, which is the key difference between Langscape’s algorithm and the traditional n-gram method, as well as the key to understanding my experiment design. In Huffman 1995, the centroid is described as a calculation of the “center of mass” of all document vectors produced from the reference set. That is, the centroid vector captures the features (frequencies of trigrams) that are relatively common to all the documents.

To calculate the centroid vector, the frequency trigram counts from each document vector are added to the corresponding entries in the centroid vector; for instance if the trigram \textit{ing} has a frequency count of \( \frac{67}{140} \) in document #1 and a frequency count of \( \frac{25}{140} \) in document #2, then the centroid vector including only these two documents would have \( \frac{67+25}{140} \) as the entry corresponding to the trigram \textit{ing}. Note that the document vectors do not necessarily have the exact same set of trigrams as each other. Consider a document vector for Arabic and one for English; the two languages use entirely different scripts, so the trigrams found in each would likely be entirely different (unless code switching is present or foreign words are used). Thus, the centroid vector will likely have more entries than many of the document vectors because it characterizes the trigrams found in any of the documents. Once the frequency trigram counts for all documents have been added to the centroid vector, this vector is also normalized by dividing each entry by the total number of documents that the centroid characterizes.

Once the centroid vector has been calculated, the document vectors and test vectors are adjusted based on the common features characterized by the centroid vector. Generally, the idea is to subtract the centroid vector from each document vector and test vector in order to remove the features that are common to all the languages being compared and
focus on the features that set each vector apart from the “normal” centroid vector. Then,
the similarity between the language documents and the test document is evaluated using
geometric techniques. Each language is given a similarity score based on the similarity
of the $n$-grams and $n$-gram frequencies of the language document and the test document.
The scores are produced by calculating the cosine similarity between the document vectors
and the test vectors. The cosine of the angle between the document vector and test vector
measures the similarity in the orientation of the vectors produced from the two texts, which
will mark the degree of similarity between the character sequences and frequencies appearing
in the two texts. The equation used for this comparison will not be given here, but it is
included in Huffman 1995.

Consider the illustration of the cosine similarity in Figure 2 from Perone 2013 as we
discuss the scores produced from the cosine values. If the vectors produced from two
documents are very similar, it means the two documents have nearly identical character
sequences. Then the angle between the two vectors would be close to zero, which would
produce a cosine value and similarity score close to 1. This is seen in the first graph in
Figure 2. If the angle between the vectors produced for the two documents is close to 90
degrees, the cosine value is close to 0 and the vectors are mostly uncorrelated, as seen in
the second graph in Figure 2. This means there seems to be no connection between the
character sequences appearing in the two documents, indicating that the language of the
two documents is likely different. Lastly, if the vectors are oriented in opposite directions,
with an angle close to 180 degrees between them, then the vectors are anticorrelated, as
seen in the third graph in Figure 2. This means the character sequences present in one
document do not appear in the second document. Thus, a similarity score of -1 would be
produced, indicating that the languages of the two documents are different (probably using completely different writing systems). So, the top results produced by an LID run will be the languages of the documents that produced the score closest to 1, which would indicate complete similarity between the language document and the test document.

Figure 2: Cosine Values and Similarity Scores

Note that the algorithm used by Langscape’s LID tool can be used in conjunction with different reference sets. The algorithm always works as described, but the language documents that the tool accesses for comparison when computing language scores for a given input text may be different when using different reference sets.

4.3 Langscape’s Language Identification Reference Set

Langscape’s LID, which is publicly available on the Langscape website, uses a reference set consisting of language documents for about 900 languages. The language documents for each language in Langscape’s original reference set (henceforth, LORS) principally consist of formal documents, such as the United Nation’s Universal Declaration of Human Rights or translations of the Bible. As discussed in Section 3, this allows the tool to work more successfully for documents that are structured, are edited, and use vocabulary more traditional in formal registers; however, this also causes difficulty in identifying texts using
less formal language, such as internet pages, blogs, online news articles, and tweets. Due to
the size of LORS, the difficulties with LID for a larger number of languages are essential
when examining Langscape’s LID.

As a summer intern for the University of Maryland Center for Advanced Study of
Language in 2015, I created an alternative reference set in order to improve the tool’s
accuracy when dealing with shorter, less formal input texts. The alternative reference set,
which I will refer to here as the Koukoutchos Informal Reference Set (KIRS), consists
of language documents composed of less formal texts, including tweets, blogs, and news
articles, in 53 languages. The 53 languages chosen for KIRS are a subset of the more
commonly spoken languages from the languages in LORS.

The creation of the new reference set was its own challenge in terms of determining
the ideal type of reference text as well as manually distinguishing between all of the
languages. When finding reference samples, texts using more topic-specific terminology,
like science journals, were avoided because they may contain character sequences that
are generally uncommon in the language and may skew the tool’s results. Travel blogs
were also discovered to be problematic because they often contain names of foreign places
that have unusual character sequences, while texts like recipes that have numbered lists
for ingredients or step-by-step instructions were also problematic. I concluded that the
best reference texts were those that covered topics about more regular activities, such as
blogs about daily life activities. These texts contained vocabulary that was commonly
used in informal contexts and not skewed too heavily towards a particular topic. Once
a non-problematic sample for a language was found, the correct language identification
was determined manually. Examining the differences in alphabets, writing systems, high-
frequency words such as pronouns, common word endings and notable consonant clusters was helpful in differentiating between languages. Once at least 3000 characters worth of sample text was collected for each language, the reference documents were created by combining the different entries into one file per language. The distinction between LORS and KIRS comes into play in the experiment conducted for this thesis.

5 The Experiment: A Two-run Algorithm for Langscape’s LID

5.1 Motivation for and Description of the Two-run Algorithm Approach

The aim of this experiment is to improve the ability of Langscape’s LID to distinguish between confusable languages by implementing a two-step algorithm approach. Using the KIRS reference set, the LID algorithm currently runs once on an input text using the reference texts for all 53 languages for comparison before showing the results. The proposed alternative approach will first run the algorithm in the same manner and then run the input text through the LID tool a second time using only the top 10 language results from the first run as the new reference set. This approach might produce different results because of the calculation and subtraction of the centroid, which characterizes the common features of all the languages in the reference set, which changes between runs. The common features of the languages in the reference set will be different depending on the group of languages used, leading to different centroids when using different groupings of languages. The idea is that running the algorithm again with a smaller number of languages will help the tool tease apart the more unique features of each language in order to match the input text to a language document.
With a general understanding of the experiment goal, the reasoning behind the idea (and why it may improve the LID results) can be examined in Huffman’s terms. The document vectors, which categorize the n-grams and n-gram frequencies of each language document in the reference set, will remain the same; however, the centroid vector will change. For example, imagine LID using KIRS is run on an input text. KIRS consists of language documents for 53 languages, including both languages that use the Latin script and languages that use the Devanagari script. A centroid vector will be calculated by finding the common features of all 53 languages in the reference set, which may not be much considering the variety of scripts used. Now, consider running LID on a subset of the initial 53 languages consisting of only 10 languages, all using the Latin script. In this case, the overlap (in terms of n-grams and n-gram frequencies) between the 10 language documents will likely be much greater than it was with all 53 language documents using different scripts. Therefore, the centroid vector for the 10 language documents will be different than the centroid for all 53 language documents. Thus, the effect of subtracting the centroid is greater when only 10 languages are compared because the relevant values in the centroid are larger. As a result, different language scores are produced, and the subtraction of a larger centroid essentially allows the tool to focus on the most unique features of each document for comparison.

5.2 Language Selection

The two-run algorithm approach was tested using KIRS. For each test, an input text was run on the reference set using both the current one-run approach and the proposed two-run
approach using quadgrams. In the two-run approach, the first run used all 53 languages in KIRS, and the second run used the top 10 language results produced from the first run.

Testing was done on a select group of languages: Hindi, Norwegian, Danish, Estonian, and Serbian. Several factors determined the selection of these languages, though each was chosen from a cluster of confusable languages. The languages were chosen to include several different scripts: Hindi uses the Devanagari script, Serbian uses the Cyrillic script, and the others use the Latin script. The languages were also chosen, in part, since they are easier to find test texts for and easier to identify and distinguish from their respective confusable languages. For instance, Hindi is better represented online in news sources and social media sites than the languages it is often confused with by LID, such as Chhattisgarhi. On the other hand, Serbian written in the Cyrillic script is confusable with Bulgarian, Ukrainian, and Russian in Cyrillic script but is easy to distinguish because it uses a unique character. Norwegian and Danish, which are often confused with each other by LID, were both chosen because the misidentification of closely related languages may not always be symmetric; repeated tests using Langscape’s LID have shown that the tool is more likely to identify Danish as Norwegian than vice versa, though the reason for this asymmetry is unknown. Regardless, testing two languages that are in the same group of confusable languages may be interesting. Lastly, Estonian, which is confusable with Finnish and Hungarian, was chosen to ensure the inclusion of at least one non-Indo-European language.

For each of the five languages discussed, twenty tests were conducted. The set of test texts for each language consisted generally of 8 tests from news articles, 6 from blogs, and 6 from tweets; however, this division varied slightly by language when blogs or tweets were more difficult to find. Each test text contained between 200 and 500 characters. The results
of the experiment are examined in Section 5.3, and conclusions about whether or not the two-run approach is a viable way to improve the accuracy of Langscape’s LID tool are drawn in Section 6.

5.3 Results

In order to compare the one-run approach and the two-run approach, I examined the language identified by the tool for each input text as well as the similarity scores produced for the top two language results. In particular, I analyzed how often the LID tool correctly identified the language of the test text, how the top similarity scores changed from the one-run approach to the two-run approach, and how confidently the tool distinguished between the top two language results. First, we examine the tool’s accuracy in correctly identifying the language of the test texts. Contrary to my hypothesis, the top language result produced for all 100 tests was the same for the one-run approach and the two-run approach. Overall, the tool correctly identified the language of the test texts with 91% accuracy. This accuracy varied by language, as shown in Table 1. Estonian and Serbian achieved the highest accuracy of 100%, and Danish achieved the lowest accuracy of 75%.

Table 1: LID Accuracy by Language

<table>
<thead>
<tr>
<th>Language</th>
<th>Hindi</th>
<th>Norwegian</th>
<th>Danish</th>
<th>Estonian</th>
<th>Serbian</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-run Approach</td>
<td>85%</td>
<td>95%</td>
<td>75%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Two-run Approach</td>
<td>85%</td>
<td>95%</td>
<td>75%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

In addition to variation by language, LID accuracy varied by the type of test text (news article, blog, or tweet). Examining accuracy by the type of test text is interesting due to the use of KIRS, which was designed to improve the tool’s performance on less formal texts.
Table 2 shows the accuracy on tests from news articles, blogs, and tweets, for each algorithm approach. Notice again that the accuracy for the one-run approach and the two-run approach is identical. Performance on news articles was better than performance on blogs and tweets, indicating that the difficulties with texts using colloquial language remain even when using a reference set consisting of texts using colloquial language. Previous testing comparing KIRS and LORS showed improved accuracy in identifying the language of texts using more colloquial language when using KIRS; however, it would be interesting to test LID accuracy for the test texts used in this experiment using LORS for further comparison.

Table 2: LID Accuracy by Type of Test Text

<table>
<thead>
<tr>
<th></th>
<th>News Article</th>
<th>Blog</th>
<th>Tweet</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-run Approach</td>
<td>97%</td>
<td>79.2%</td>
<td>77.8%</td>
</tr>
<tr>
<td>Two-run Approach</td>
<td>97%</td>
<td>79.2%</td>
<td>77.8%</td>
</tr>
</tbody>
</table>

Next, we examine the languages identified for the test texts with respect to each language. Table 3 shows the number of texts identified for each language result obtained. Each time LID misidentified a test, the language selected was written in the same script as the correct language and more closely related to the correct language than the other languages in KIRS. For instance, tests in Hindi were misidentified as either Nepali or Marathi, which are two out of six Indo-Aryan languages represented in KIRS besides Hindi and the only two languages in KIRS that use the Devanagari script apart from Hindi. The results from the Norwegian and Danish tests confirmed the suspicion that confusable languages are not always misidentified as each other in a symmetric fashion. While only one Norwegian text was misidentified as Danish, five Danish texts were misidentified as
Norwegian. Meanwhile, both Estonian and Serbian were identified correctly in all cases under both algorithm approaches.

<table>
<thead>
<tr>
<th>Test Language</th>
<th>One-run Approach</th>
<th>Two-run Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hindi</td>
<td>Hindi</td>
<td>Hindi</td>
</tr>
<tr>
<td>Marathi</td>
<td>1</td>
<td>Marathi</td>
</tr>
<tr>
<td>Nepali</td>
<td>2</td>
<td>Nepali</td>
</tr>
<tr>
<td>Norwegian</td>
<td>Norwegian</td>
<td>Norwegian</td>
</tr>
<tr>
<td>Danish</td>
<td>Danish</td>
<td>Danish</td>
</tr>
<tr>
<td>Norwegian</td>
<td>5</td>
<td>Norwegian</td>
</tr>
<tr>
<td>Estonian</td>
<td>Estonian</td>
<td>Estonian</td>
</tr>
<tr>
<td>Serbian</td>
<td>Serbian</td>
<td>Serbian</td>
</tr>
</tbody>
</table>

Since every test produced the same top language result using both the one-run and two-run approach, the scores of the top language result are particularly interesting to compare between the one-run and two-run approach. Figure 3 plots the difference in the top score between the one-run and two-run approach for the 20 tests in each of the five languages. This difference was calculated by subtracting the top score produced by the two-run approach from the top score produced by the one-run approach. Therefore, a positive result indicates that the top score was higher using the one-run approach. So, Figure 3 shows that the top language result produced from the one-run method scored higher than the top result produced from the two-run method in every test. First, these results show that the score produced for the top language was never the same using both approaches, which indicates that changing the set of languages used for comparison does affect the LID results. Second,
we examine why the scores produced using the one-run approach were consistently higher than those produced using the two-run approach.

Figure 3: Difference in Top Language Scores Between One-run and Two-run Approach

(a) Hindi
(b) Norwegian
(c) Danish
(d) Estonian
(e) Serbian

I predicted that the second run of the two-run approach would involve a comparison of the most unique \(n\)-grams found in the documents due to the subtraction of a larger centroid. Recall that the centroid calculated in the first run characterizes the \(n\)-grams shared among all 53 language documents in KIRS, and the centroid calculated in the second run characterizes the \(n\)-grams shared among only 10 of the language documents. If the centroid in the second run is larger and characterizes more \(n\)-grams as predicted, then the centroid subtraction in the second run removes more of the \(n\)-grams shared by the top result’s language document and the test document than the centroid subtraction in the first run. With even more \(n\)-grams that are common to these two documents now absent, a lower similarity score in the second run indicates that part of the similarity score produced in the first run was determined by
the n-grams that were common not only between the top language document and the test
document but also among the top 10 language documents.

It is clear from these results that the value of the top language score is not by itself a
good measure of how well LID is functioning. Since the language result in the second run
is selected based on the similarity of the most unique n-grams found in the documents, a
language document would have to be much more similar in n-grams and n-gram frequencies
to the test document in order to receive a score as high or higher than in the first run. The
comparison of similarity scores and the selection of the top language result are relative to
the other language documents in the reference set. For example, a language with a score of
0.3 could be the top result, even though the score is not that close to 1, which would indicate
complete similarity, as long as all other languages being compared scored lower than 0.3.
On the other hand, a language with a score of 0.7 could be the second result, even though it
has a high score, if there is a language scoring slightly higher. Thus, the raw value of the top
language score is not by itself a good measure of the performance of LID.

A more important measure of the performance of LID is the “confidence” with which it
distinguishes between the top two language results. This can be investigated by examining
the difference between the score of the top result and the score of the second result, which I
call the score gap. The larger the score gap, the greater the perceived difference between:
1. the similarity between the n-grams of the test document and the top result’s language
document and 2. the similarity between the n-grams of the test document and the second
result’s language document. Thus, a larger score gap indicates that the algorithm was more
confident that the language of the test document matches the top language result and not
the second language result. I examined changes in the score gap between the one-run and
two-run approach in two cases: when the target language was correctly identified and when the target language was not correctly identified. For all five languages, Table 4 shows the average score gap for tests correctly identified by LID and the average score gap for tests incorrectly identified by LID under both the one-run and two-run approach.

Table 4: Average Score Gaps

<table>
<thead>
<tr>
<th></th>
<th>One-run Approach</th>
<th>Two-run Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Correct LID</td>
<td>Incorrect LID</td>
</tr>
<tr>
<td>Hindi</td>
<td>0.0519</td>
<td>0.0543</td>
</tr>
<tr>
<td>Norwegian</td>
<td>0.0452</td>
<td>0.0280</td>
</tr>
<tr>
<td>Danish</td>
<td>0.0411</td>
<td>0.0324</td>
</tr>
<tr>
<td>Estonian</td>
<td>0.1226</td>
<td>-</td>
</tr>
<tr>
<td>Serbian</td>
<td>0.1104</td>
<td>-</td>
</tr>
</tbody>
</table>

In Figure 4, we examine the difference in the average score gap between the correct LID tests and the incorrect LID tests, both for the one-run approach and the two-run approach. Since all Estonian and Serbian tests in the experiment were correctly identified using both the one-run and two-run approach, no comparison can be made with respect to the difference in average score gap between correct and incorrect results. Therefore, Figure 4 only includes the remaining three languages used in the experiment. For the other languages, the tool would ideally be more confident when selecting a correct language and less confident when selecting an incorrect language. This would mean that the score gap would be larger for correct LID tests than incorrect LID tests. The difference between correct and incorrect LID tests is calculated by subtracting the average score gap of incorrect tests from the average score gap of correct tests. For example, the one-run approach for Hindi found an average score gap of 0.0519 for correct results and an average score gap of 0.0543 for incorrect
results; so, the one-run score gap between correct and incorrect LID for Hindi is $-0.0024$, as seen in Figure 4, since $0.0519 - 0.0543 = -0.0024$. If the difference is positive, then the score gap is larger for correct tests as desired. If the difference is negative, then the score gap is larger for incorrect tests, which is not ideal.

Figure 4: Score Gap Comparison Between One-run and Two-run Approach

Figure 4 shows that the difference in the average score gap between the correct and incorrect LID tests is positive for both the one-run and two-run results for Norwegian and Danish. So, the tool more confidently distinguishes between the top two results when LID is correct than when LID is incorrect, as desired. Furthermore, this trend holds for both the one-run and two-run approach, which is also a favorable result. Unfortunately, the difference was negative for both the one-run and two-run results for Hindi, indicating that the score gap was larger for incorrect LID tests. This result is problematic but may be linked to the selection of languages included in the reference set, as discussed next.

Out of the 53 languages in KIRS, only three languages use the Devanagari script: Hindi, Nepali, and Marathi. Therefore, each of these languages is likely to receive a relatively high similarity score for a text in any language written in the Devanagari script. Since only three languages in KIRS use the Devanagari script, both the first run and second run
include comparison languages that use a different script than the input text. As a result, the centroid likely characterizes very few n-grams, which means the tool is still limited in its ability to focus on the most unique features of the input text, even when restricted to 10 languages. Further testing on a reference set that includes at least 10 languages in the Devanagari script could help determine whether the undesirable effect of having a larger score gap for incorrect LID tests is due to the fact that the centroid remains ineffective in this experiment for Hindi tests.

Next, we examine the difference in the average score gap between the one-run results and the two-run results, both for correct LID tests and incorrect LID tests. The difference in the average score gap between the one-run results and the two-run results, calculated by subtracting the average score gap of the one-run results from the average score gap of the two-run results, is illustrated in Figure 5. A positive result shows that the score gap increased using the two-run approach, which indicates that the tool more confidently distinguished between the top two results. Similarly, a negative result shows that the score gap decreased using the two-run approach, which indicates that the tool less confidently distinguished between the top two results. Ideally, the average score gap between the one-run results and the two-run results would be positive when LID is correct and negative when LID is incorrect.

Since the tool correctly identified all Estonian and Serbian tests in the experiment, only the difference in the average score gaps between the one-run results and the two-run results for correct LID tests can be examined, which explains the absence of a second bar for Estonian and Serbian in Figure 5. The difference in the average score gap between the one-run results and the two-run results was positive in all cases, though the degree of change
varied. In the case that LID was correct, this positive result is helpful because the tool is more confidently distinguishing between the top two results in order to select the correct language as the top result. However, the score gap also increased in the two-run approach for incorrect tests, contrary to my prediction.

Looking more closely, Figure 5 shows that, in Hindi, the change in score gap from one-run approach to the the two-run approach was greater for incorrect tests than for correct tests. This indicates that while the score gap increased in all cases in the second run, it increased more for incorrect tests than correct tests in Hindi. This indicates that the tool more clearly selected an incorrect top result in the second run than the first run, which is problematic. In contrast to the Hindi results, however, both Norwegian and Danish found the change in score gap from the one-run approach to the two-run approach to be greater for correct LID tests, which is a helpful result. This demonstrates that while the score gap increased in all cases in the second run, it increased more for correct tests than incorrect tests in Norwegian and Danish, which is a beneficial result.

In the end, the two-run approach more confidently distinguishes between the top two
results when LID is correct, as desired. Although the two-run approach also more clearly
distinguishes between the top two results when LID is incorrect, the increase in the score
gap is not as dramatic as for correct LID tests for two out of three languages for which
the comparison can be made. For these two languages, the increase in score gap from the
one-run approach to the two-run approach for correct tests was greater than 0.01 each time.
Additionally, the two-run approach produced a larger score gap for incorrect tests than the
one-run approach, but the increase in score gap was less than or equal to 0.007. Further
testing is needed to determine the significance of such an increase in score gap; however, if
only a change of at least 0.01 is significant, then the two-run approach is more confidently
selecting the top results for correct tests (by a score gap of at least 0.01) and selecting the
top result for incorrect tests by an insignificantly larger amount (less than 0.007) than the
one-run approach. This would indicate that the two-run approach has beneficial effects on
the similarity scores both when LID is correct and incorrect.

6 Conclusions

The language identified by the tool for each test using the two-run approach was identical to
the language identified using the one-run approach; however, the two-run approach showed
improvements in the similarity scores produced for the top two results. It seems that running
the LID a second time on a smaller group of languages allows the algorithm to make finer
distinctions between the top 10 languages produced from the first run, which are presumably
more similar to the language of the test document. This finer distinction did not lead to a
different top language result in the second run but consistently produced language results
with a lower top score. A lower top language score indicates that the centroid is, in fact, removing more n-grams from consideration before comparing the test vector to the language document vectors. The two-run approach also produced a larger average score gap for correct tests than for incorrect tests in two out of three languages for which the comparison could be made (recall that all Estonian and Serbian tests were identified correctly). So, the two-run approach is more clearly distinguishing between the top two language results when LID is correct using both the one-run and two-run approach, which is desirable. Additionally, there was a smaller score gap change from the one-run approach to the two-run approach when LID was incorrect in two of the three languages, suggesting that the two-run approach selects incorrect LID results similarly to the one-run approach and selects correct LID results more clearly than the one-run approach, which is also an improvement.

The particular effects of the two-run approach found in this experiment may be specific to the number of languages used for comparison, the size of the input texts, and the number of languages used in the second run of the two-run approach. Further research is needed to determine the merit of the two-run approach when these variables are altered. I speculate that the two-run approach would have more noticeable effects when the number of languages included in the reference set increases by several hundred or when the test texts are shorter in length. Preliminary testing using tweets with fewer than 200 characters shows that the top language results produced by the two-run approach may even differ from the results produced by the one-run approach. In several cases, the second language result in the first run became the top language result in the second run, producing a correct language identification only after the second run. Each time this change occurred, the initial score gap between the top two languages in the first run was less than 0.01, indicating a lower
confidence in the language selection. Further testing is needed to determine the full effect of
the two-run approach under different circumstances, but preliminary testing indicates that
the effects may be more beneficial than those found in this experiment.
References


Huffman, Stephen. 1995. Language-independent document categorization by n-grams. In


