ABSTRACT
We present herein a language identifier developed for metadata records. Trained with European Parallel Corpora, this program can distinguish English metadata records from those of other languages. We evaluated the program’s accuracy with a test collection of 800 metadata records and compared the program’s accuracy to that of an open-source language identification program.

Keywords
Language identification, text classification, natural language processing, metadata records, N-gram, language model.

INTRODUCTION
Language identification is the act of determining to which language a text or audio belongs. Language identifiers seldom classify all text inputs correctly, though accuracy often depends upon the underlying language model, itself a function of the amount of text in each language available to the classifier. Language identification is used in many applications. For example, many online machine translation systems have a language identification program built in to determine the language of textual input. While there are many methods for performing language identification, an N-gram approach, as proposed by Cavnar and Trenkle (1994) has proven effective.

An N-gram is a sequence of N consecutive tokens taken from an input text. In the case of language identification, these tokens are usually words. N-gram models predict the last word of an N-gram from the previous one(s) using some empirical probability taken from training input text. Jurafsky and Martin (2009) refer to such a statistical mechanism as a language model, or LM. N-grams are useful in areas other than language identification, including information retrieval, machine translation, speech recognition, and author identification, among others.

As part of an investigation into multilingual information access of digital objects in digital collections, the author and three others acquired two million textual metadata records from a pair of governmental and university libraries with the aim of translating the English metadata records into Spanish and Chinese. Many of these records, however, were not English, and since human inspection of all the records was infeasible, we required a language identifier to streamline the workflow.

Many open-source language identifiers are available, such as Apache Tika (http://tika.apache.org/1.5/detection.html), a language identification library in Java (https://code.google.com/p/language-detection/), and Perl's Lingua::Identify (http://search.cpan.org/~ambs/Lingua-Identify-0.56/lib/Lingua/Identify.pm). The author’s prior experience with Perl led to Lingua::Identify being selected for separating English and non-English metadata records to enable English to Chinese and English to Spanish translation.

Lingua::Identify uses 4 methods of language identification and a total of 13 variations on those methods. The methods are: Smallwords, prefixes2, prefixes4, suffixes2, suffixes4, suffixes3, suffixes4, ngrams1, ngrams2, ngrams3, and ngrams4. Each of these is described in the Methods of Language Identification at http://search.cpan.org/~ambs/Lingua-Identify-0.56/lib/Lingua/Identify.pm.

The following outlines a study performed to determine whether better accuracy could be achieved using an in-house language identification program rather than Perl’s Lingua::Identify.

METHOD
Our team agreed that the default method in Lingua::Identify, which consists of smallwords, prefixes2, suffixes3, and ngrams3, would be sufficient for this experiment. Smallwords, prefixes2, and suffixes3 were suitable for language identification in this task. However, many of the metadata elements in our collection contain only one to three words. We expected ngrams3 might fail on those metadata elements containing only one to three
words, or N-grams. Early experiments demonstrated that changing the default ngrams3 to ngrams1 or ngrams2 resulted in no more than a 1% variance in the overall accuracy of the program, so we used the default method.

Smallwords considers the stop words of a language; prefixes2 considers the first two letters of prefixes; suffixes3 considers the last three letters of suffixes; and ngrams3 considers sequences of three words or tokens.

To evaluate the accuracy of Lingua::Identify, we developed a test collection with 800 records randomly selected from the two million total records. The author and three other humans manually inspected the records. Each human classified 200 records to be either English or non-English; a second human subsequently validated the first human’s judgments. For our purposes, the languages of the non-English records were not of import, only that the records’ languages were non-English. Our manual inspection of the 800 total records resulted in a total of 250 non-English records and a total of 550 English records. This implied that about 68-69% of the total two million records were likely to be written in English. The agreement between reviewers was 98%, with disagreement arising mostly in cases where a metadata record was partially in English. For example, many records had <title> and <creator> elements that were not in English, while the <subject> and <coverage> elements of the same records were in English.

The low accuracy of Lingua::Identify in a few trials served as an impetus for building an in-house program. The in-house program utilizes a bigram LM at the letter-level rather than the word level, justified by the word level requiring far more computational time and being less appropriate for metadata, being that metadata is often in fragmented sentences or even single words. A bigram is an N-gram consisting of two items, or letters, in this case. The LMs were built using the Europarl corpus (http://www.statmt.org/europarl/), which consists of political proceedings of the European Parliament in 21 European languages. This corpus is commonly used in machine translation research. The languages include Slovene, Slovak, Latvian, Portuguese, Italian, Spanish, etc. The language files differ in size. The smallest, Romanian, is 66.4 MB. The largest, Greek, is 390.6 MB. The various sizes of the language files may have affected the accuracy of the program, with smaller language models likely producing less accurate results than larger language models. Political proceedings are, of course, a different domain from metadata records. However, we believed that in-domain data was less necessary when implementing a letter-level bigram model.

The nature of the data we are working with, i.e., metadata records, makes it impractical to use a word-level N-gram language model, i.e., an N-gram model considering entire words as tokens. Many language identification programs work with N-gram models usually consisting of words, pairs of words, or three or more words. Table 1 displays both a sample English metadata record and a sample non-English record taken from the test collection.

<table>
<thead>
<tr>
<th>Sample Metadata Elements</th>
<th>English Record</th>
<th>Non-English Record</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;title&gt;Terrorism: ## a guide to events and documents / &lt;/title&gt;</td>
<td>&lt;title&gt;Proust, un amour de Swann: ## dossier du professeur / &lt;/title&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;creator&gt;Kronenwetter, Michael. &lt;/creator&gt;</td>
<td>&lt;creator&gt;Richer, Edmond. &lt;/creator&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;subject&gt;Terrorism. ## Terrorism ## United States. ## Terrorists. &lt;/subject&gt;</td>
<td>&lt;subject&gt;Proust, Marcel. ## 1871-1922. ## Amour de Swann. &lt;/subject&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;publisher&gt;Westport, Conn. : ## Greenwood Press, &lt;/publisher&gt;</td>
<td>&lt;publisher&gt;[Paris?]: ## Hachette, &lt;/publisher&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;description&gt;The philosophy of terror -- A short history of terrorism -- Turning to terror -- Varieties of terror -- Weapons of mass destruction -- Their name is legion: a selection of terrorist groups -- Four aspects of terror -- Chronology -- Documents. &lt;/description&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Sample metadata record from test collection.

The “##” between items in the <title>, <subject>, and <publisher> elements is simply to separate subfields within the metadata records’ elements. The impracticality of a word-level N-gram model can be illustrated by the <subject> element of the English record in Table 1. A word-level N-gram model would compute the probability of “Terrorism Terrorism United States Terrorism” occurring as an English string, unless the program considered the subfields individually. This probability would likely be fairly low. A letter level N-gram language model, however, can easily handle instances such as this. A letter level N-gram model considers tokens of letters rather than of words. For example, the probability is calculated of ‘e’ following ‘t’. This sort of LM also suffers less from data sparsity than a word-level N-gram model may. The computational time of such a model is also speedier than a word-level N-gram LM: There are far fewer letters in a language than words, and storage and calculations of probabilities of sequences of letters is less time consuming computationally than the same is for words.

We used the Perl scripting language to develop our program and used the Europarl corpus to develop the LMs. We created a letter-level bigram LM for each of the languages in the Europarl corpus which recorded all the probabilities of one letter following another for each language. The several gigabytes of textual data resulted in robust LMs. After constructing the LMs, the program tracked each element's and each record's language probability using the LMs. The “winner” was the highest probability language
for each element and whole record. If the winner was not English, the program stored the record into a non-English corpus, leaving behind only records the program determined to be English.

Although we expected instances of data sparsity to be very few or none, we applied Laplace, or 'add one', smoothing to the data to handle unseen instances if they were to occur. Laplace smoothing adds one to each probability to eliminate instances of zero probabilities.

RESULTS
Inputting the test records into the Perl module—Lingua::Identify, resulted in the following: 281 of 550 English records correctly identified, and 195 of 250 non-English records correctly identified. This means that the program was 51% accurate in determining English records and 78% accurate in determining non-English records. Therefore, the total accuracy of Lingua::Identify on differentiating English from non-English records was 59.5%. While the total accuracy of this program was better than chance, the English accuracy above is only slightly better than chance.

The results of the in-house program's accuracy are as follows: 384 of 550 English records correctly identified, and 167 of 250 non-English records correctly identified. This means our program's accuracy at determining English records was 69.8%, and the accuracy at determining non-English records was 66.8%. So the total accuracy of the in-house program at determining English from non-English records was 68.875%. The difference between this program’s accuracy and the Perl module’s accuracy in determining non-English records might be the consequence of the disparity between languages considered. The in-house program uses only the 21 languages in the Europarl corpus, while the Perl module utilizes 33 languages, only ten of which are from the Europarl corpus. Since an estimated 30% of the total records will be non-English, we chose the in-house program to separate the English and non-English records.

Using the 800 test records, we found that the accuracy of the in-house program could be further improved by adding additional in-domain data to the English LM. We randomly selected another 20,000 records from the records that remained of the original 2 million records for this purpose. After adding the 20,000 records to the English LM, the program accurately identified 422 of 550 records as English, yet accurately identified only 147 of 250 records as non-English. This is a 76.7% accuracy rate for English and a 58.8% accuracy for non-English, with an increased total accuracy of 73.625%. It is unclear why the accuracy of non-English identification dropped with this addition to the English LM.

Table 2 displays the accuracy rates for the Perl module, Lingua::Identify, the in-house program, and the in-house program with augmented English LM.

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Perl</th>
<th>Lingua::Identify</th>
<th>In-House Program</th>
<th>Augmented In-House Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>English:</td>
<td>51%</td>
<td>69.8%</td>
<td>66.8%</td>
<td>58.8%</td>
</tr>
<tr>
<td>Non-English:</td>
<td>78%</td>
<td>66.8%</td>
<td>58.8%</td>
<td>58.8%</td>
</tr>
<tr>
<td>Total:</td>
<td>59.5%</td>
<td>68.875%</td>
<td>68.875%</td>
<td>73.625%</td>
</tr>
</tbody>
</table>

Table 2. Comparison of accuracy rates for each program.

SUMMARY AND FUTURE WORK
While language identification is seldom 100% accurate, we found that an in-house program performed with higher accuracy than an open-source program in determining the language of metadata records. Using this program will allow us to proceed with our larger investigation of metadata records translation with less chance of error stemming from non-English records remaining in the dataset.

Future work will include investigating ways to improve the program's accuracy at identifying English and non-English metadata records. We will explore possible data to add to the English LM, including more in-domain data, i.e., metadata records. We also plan to replicate this experiment using alternative open-source language identification programs to provide us more data for comparison with our own program.

ACKNOWLEDGMENTS
This project is supported by the Institute of Museum and Library Services (IMLS) under the National Leadership Grant LG-06-13-0187-13. The author thanks the team members of the MLIA project (http://txcdk-iia.unt.edu/MLIA) who participated in generating the gold standard for evaluation.

REFERENCES