A Study in Language Identification

Rachel Mary Milne
University of Otago
Dunedin, New Zealand
rmilne@cs.otago.ac.nz

Richard A. O'Keefe
University of Otago
Dunedin, New Zealand
ok@cs.otago.ac.nz

Andrew Trotman
University of Otago
Dunedin, New Zealand
andrew@cs.otago.ac.nz

ABSTRACT

Language identification is automatically determining the language that a previously unseen document was written in. We compared several prior methods on samples from the Wikipedia and the EuroParl collections. Most of these methods work well. But we identify that these (and presumably other document) collections are heterogeneous in size, and short documents are systematically different from large ones. That techniques that work well on long documents are different from those that work well on short ones. We believe that improvement in algorithms will be seen if length is taken into account.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Linguistic Processing; H.3.3 [**Information Search and Retrieval**]: clustering; I.2.7 [**Natural Language Processing**]: Language models

General Terms

Experimentation

Keywords

Language identification

1. INTRODUCTION

Almost anything you might want to do with a natural language document requires that you know what language it is in, even determining whether "1,234" is larger or smaller than 10. If you want to return documents relevant to a query, documents the user cannot read are not relevant.

It would seem that this problem could be solved at the source in many cases. Most text-holding elements in HTML, [11], for example, may have a "`lang`" attribute with an RFC 1766 [1] language code as value. XML [3] builds this into the XML framework as "`xml:lang`" [3, section 2.12]. Word processors such as Word and Symphony allow the author to set the language of any region of text.

This kind of annotation permits fine-grained classification. A document such as Olivier Lauffenburger's "Hittite Grammar" [7], for example, has raisins of Sumerian and Akkadian as well as plums of Hittite in a large custard of English.

However, such annotations are often omitted or misused. A document written in New Zealand English, for example, may be left with the word processor's default setting of American English (en_US), to the detriment of spelling and grammar checkers. Phillip Koehn [6] reported, for example, of the EuroParl corpus that "part of the 'English' part of the proceedings contain[ed] actually French texts [in May 1996]".

The language identification problem, then, is to automatically determine from the text itself what language it is written in. This problem has been addressed many times in the literature, but we uniquely identify that non-textural characteristics of a document can affect the accuracy of the algorithms. Specifically, we identify that the two corpora we use, Wikipedia and EuroParl, have multi-modal length distributions and this affects result quality. Short documents are obviously *harder* to classify just because they provide less evidence, but it turns out that they use language differently from long documents, so different algorithms are needed for each case.

It is obviously impossible to identify documents written in a language you have no knowledge of. It is also obviously difficult to discriminate between closely related languages given short documents: "The cat sat on the mat" is perfectly good American and perfectly good English. It is also clearly difficult to do fine-grained automatic classification, because of accidental similarities between languages ("come" is both Italian and English) and borrowings ("Matariki" is Māori, but is used in New Zealand English). What we can realistically hope for is automatic classification of whole documents into one of a small group of not too similar known languages.

2. RELATED WORK

Language identification is a well studied problem and space requirements prevent a thorough literature survey. However, several prior algorithms are discussed in this section.

Cavnar & Trenkle [4] developed an n-gram based text classifier, which they used to classify Usenet articles in English,

ADCS '12, December 05 - 06 2012, Dunedin, New Zealand

Portuguese, French, German, Italian, Spanish, Dutch and Polish. Here an n-gram is a contiguous subsequence of n characters. For each language, they create a list of the common most n-grams in descending order of frequency, called an n-gram profile. These profiles are mixed in n-gram length with n ranging from 1 to 5 (for example, the word "the" contributes "`t`", "`h`", "`e`", "`th`", and "`he`" as well as "`the`".)

Cavnar & Trenkle compare the n-gram profile of a new document with the n-gram profiles of the known languages, summing the absolute differences of the ranks ascribed to each n-gram. For example, if "the" is rank 1 in English and rank 9 in the unknown document, the absolute rank difference is 8. The language with the lowest sum is reported as the class of the new document. They reported an accuracy of 99.8% using this method.

Hayati [5] applied Cavnar & Trenkle's algorithm to a collection of Web documents in Danish, German, English, Spanish, Finnish, French, Italian, Dutch, Norwegian, Portuguese and Swedish. She found the accuracy to be lower than reported, at 86.8%. She suspected that their technique did not choose representative n-grams with sufficient power to distinguish between similar languages, so she used the Fisher discriminant function to choose n-grams and cosine similarity to compare document profiles to language profiles, and accuracy improved to 93.9%.

Langdetect [12] is an open-source Java library for language identification. It uses a naïve Bayes algorithm with character n-grams. McCandless [9] compared three classifiers, of which langdetect was the best. He reported an accuracy of 99.2% on a collection with 17 languages, ranging from 97.2% for Danish to 100.0% for Greek. Langdetect comes with built-on profiles, so we did not train it in our experiments. However we used our own markup removal algorithm, as it does not do this itself, Langdetect reports several possibilities; we only used the language with the highest score.

Mayer [8] looked at tweets and e-Bay messages, which are very short. He used the first two and last two words in each document and looked them up in dictionaries for each language.

It is Mayer's work that first alerted us to the possibility that different algorithms might perform better on different length documents. To this end we analysed the length of documents in the collections we used and identify that they are mixed-modal. We consequently tested several algorithms on these different lengths and show here that, indeed, different algorithms are suited to different lengths of document.

3. LANGUAGES USED

This study primarily worked with four languages: German (de), English (en), Spanish (es), and French (fr). They were chosen because one author was able to read all these languages and consequently articulate why the classifiers were failing (recall that Koehn reported misclassified documents in EuroParl). In section 6 Dutch and Italian were further examined.

<sing languages that could be written in the ISO Latin 1 character set simplified the coding but also makes the identification problem harder. For this set of languages, conversion to lower case is language-independent. Equally, separating Chinese from Russian from English can be done by mechanically examining the codepage used in the majority of the document.

4. DOCUMENT COLLECTIONS USED

We worked with two collections: the Wikipedia [13] and the EuroParl collection [6].

September 2012 Wikipedia dumps were obtained for German, Dutch, English, Spanish, Italian, and French. They were decompressed, parsed as XML, the <text> elements were extracted, Wikimedia markup was removed, and they were tokenised. Stemming and stopping were not performed because they cannot be performed until the language is known. Wikimedia markup removal was done with *ad-hoc* code. Words were converted to lower case. The results of section 6 were obtained using one author's Wikimedia stripper, and randomly selected training sets and test sets of 100,000 documents each for each language, only documents with more than 10 words being selected. The results of section 8 were obtained using another author's Wikimedia stripper, and a training set of 1,000 documents in each language. Although we used two different strippers we do not believe that this substantially affects the results. Equally, we do not believe that the different numbers of test documents will substantially affect results either.

The mid-2012 edition of the EuroParl collection was obtained, decompressed, and the XML markup removed. Stemming and stopping were not done (recall that they are language dependent). Words were converted to lower case. Test sets of 1,000 documents were used. In section 6 only documents with more than 10 words were selected.

Document lengths were also checked in the Wall Street Journal and INEX IEEE article corpora.

5. NONUNIFORMITY OF COLLECTIONS

Our preliminary investigatory experiments suggested that each technique performed worse than expected. Closer inspection showed that most of the difficulty was with very short documents.

Figure 1 shows a histogram of (log Wikipedia article length in words + 1) for each of German, English, Spanish, and French. The overlaid curves show normal distributions fitted to the data using the `normalmixEM` function from the "mixtools" package [2] in R [10]. This simple mixture distribution appears from visual inspection to fit well. The fact that the document length *distribution* can be modelled as a mixture of two simpler distributions strongly suggests that the *collection* is actually a mixture of two collections with different properties.

If one of these distributions were small relative to the other the it might be effective to simply ignore the bi-modality of the collection. Table 1 shows the proportion of documents that fell into the "small" (about 2 words) group and proportion that fell into the "large" group (averaging about 160 words) for each of the four languages. Neither group is negligible.

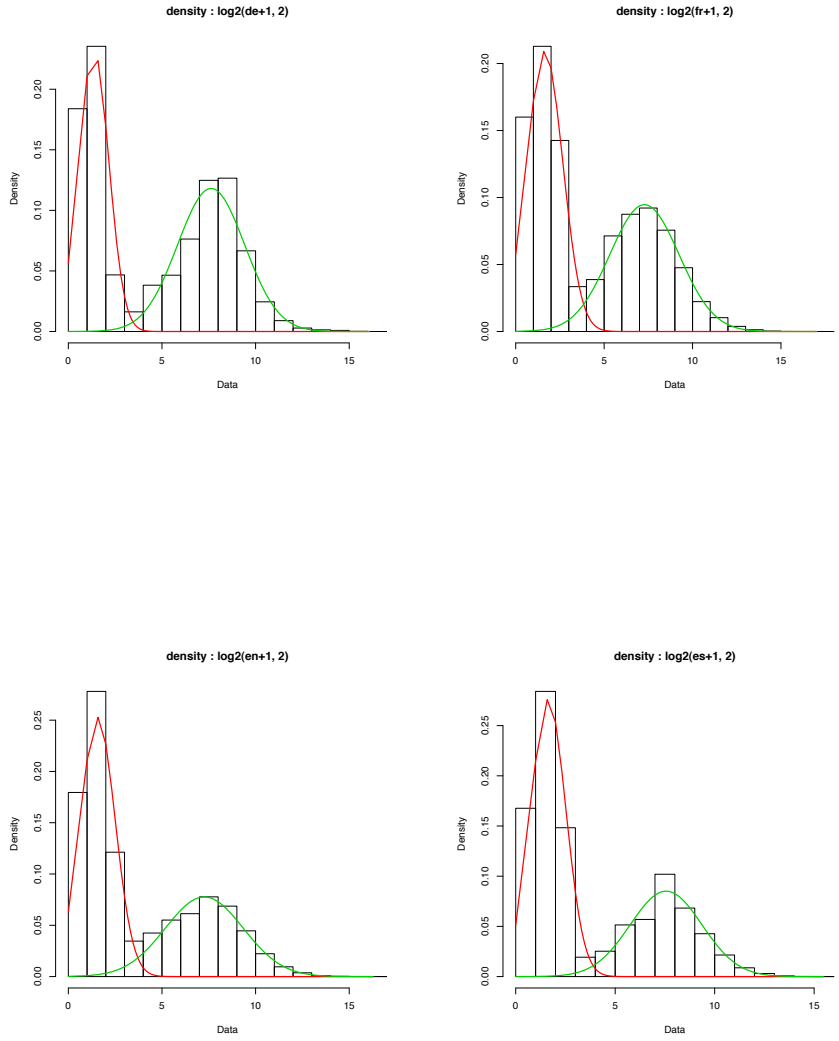

Figure 1: Length distribution of Wikipedia articles showing a bi-modal distribution in all four languages examined

Figure 2 shows the equivalent histogram (log document length in words + 1 against proportion of documents) for the same four languages, in the EuroParl collection. Again the overlaid curves show normal distributions fitted using `normalmixEM`. This time, a mixture of four normals fitted best (by visual inspection).

Table 2 shows the proportion of EuroParl documents that fell into the "small" (about 10 words), "medium" (about 170 words), "large" (about 8,000 words), and "huge" (about 80,000 words) groups for each of the four languages.

We found other collections to be mixed as well. For example, the INEX IEEE collection appears to be a mixture of three groups (about 800, about 1900, and about 5000 words). In further experiments we will examine further collections to determine whether this is a pattern we can expect of whether it is characteristic of just these collections.

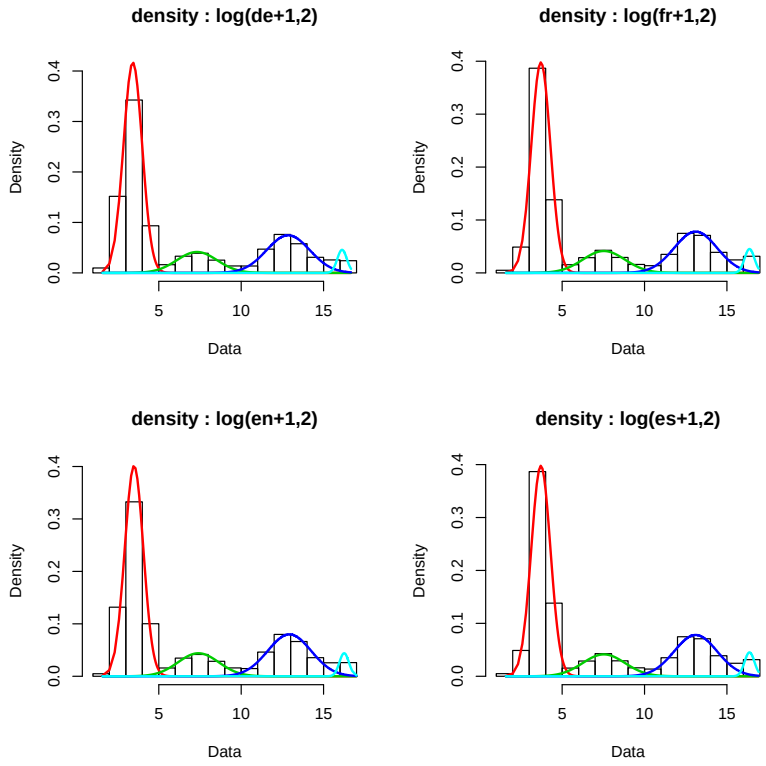

Figure 2: Length distribution of EuroParl documents showing a quad-modal distribution in all four languages examined

Language	Small	Large
de	47%	53%
en	59%	41%
fr	53%	47%
es	61%	39%

Table 1: Proportion of Wikipedia documents of each modality for each of the four languages

Language	Small	Medium	Large	Huge
de	60%	13%	24%	3%
en	57%	14%	26%	3%
fr	58%	13%	26%	3%
es	58%	14%	24%	3%

Table 2: Proportion of EuroParl documents of each modality for each of the four languages

We observe that short documents tend to use language differently from long ones and very short documents more so. In the EuroParl collection we observe headlines or stylised "see XXX committee minutes". Similarly within the Wikipedia, redirect articles are common. Table 3 shows that *as a whole*, short Wall Street Journal articles do not look like longer ones. When judging the effectiveness of language classifiers, we believe it is important to judge the size classes separately. A language identification may do well on one size class and badly on another.

Top 20 words			Top 20 trigrams		
Small	Medium	Large	Small	Medium	Large
of	the	the	of	the	the
a	of	of	a	ion	ing
the	a	to	the	of	of
from	to	a	ent	ing	to
in	and	in	ion	a	ion
and	in	and	ing	and	and
to	said	s	and	to	a
was	s	that	ear	ent	in
said	million	for	ice	ill	ent
this	for	is	rom	in	tio
president	it	on	in	com	for
year	its	it	fro	tio	s
named	that	as	res	for	tha
director	from	at	to	lli	ter
earlier	company	with	was	res	hat
s	is	by	ect	aid	ate
vice	will	said	ill	ter	ati
for	by	mr	inc	sai	ill
million	on	he	ati	lio	ers
board	year	from	aid	ati	com

Table 3: Wall Street Journal; words and trigrams; small = 1 to 30 words, medium = 31 to 300 words; large = over 300 words.

	de	nl	en	es	it	fr
de	99.4%	0.1%	0.3%	0.0%	0.1%	0.1%
nl	0.4%	99.3%	0.2%	0.0%	0.1%	0.0%
en	0.7%	0.5%	98.4%	0.1%	0.3%	0.0%
es	0.3%	1.5%	0.5%	97.4%	0.3%	0.1%
it	1.3%	0.1%	9.4%	1.5%	87.6%	0.1%
fr	0.7%	0.5%	0.5%	0.3%	0.4%	97.5%

Table 4: Wikipedia confusion matrix, $k = 20$. Row = true language, column = assigned language.

	de	nl	en	es	it	fr
de	99.5%	0.1%	0.3%	0.1%	0.0%	0.0%
nl	0.1%	99.3%	0.3%	0.2%	0.0%	0.0%
en	0.3%	0.1%	99.3%	0.1%	0.1%	0.1%
es	0.1%	0.1%	0.6%	99.1%	0.0%	0.1%
it	0.1%	0.1%	1.3%	8.0%	90.4%	0.1%
fr	0.2%	0.1%	1.0%	0.3%	0.1%	98.3%

Table 5: Wikipedia confusion matrix, $k = 1000$. Row = true language, column = assigned language.

6. BASELINE

It is reasonable to conclude from the document length analysis and literature survey that collections may not mixed, and that if it were possible to identify which component a document belongs to then it might be possible to increase the performance of language identification. This, in turn, might (although we leave it for further work to demonstrate this) increase the precision of a search engine.

In order to get a clear view of the merits of each method when applied to long documents (Documents shorter than 11 words were excluded), we implemented a simple term-based baseline.

A language profile was built for each language by taking the top k most frequent word from the training set for that language. For each document in the test set, the unique words were identified and compared to these language profiles. This comparison was a straight set intersection. The language of the document was chosen as the language of the largest set. That is, if the document contained 23 of the top 50 words in English but only 12 of the top 50 words in German then the document was identified as bring written in English.

This approach resembles that of Cavnar & Trenkle, but pays no attention to relative ranks. In later sections we refer to this method (with $k = 1000$) as the "top 1000 words" method.

Table 4 presents the confusion matrix for the Wikipedia test set, trained on the Wikipedia training set when $k = 20$. Cells were rounded to the nearest 0.1%. It is reasonable to expect more evidence to produce better results and so the experiment was re-performed with $k = 1000$ (see Table 5). Percentages are presented (rather than absolute counts) in these tables so that they can be more easily compared.

It is astonishing that such a crude technique performs so well (typically over 95% accuracy). Further analysis is required to explain why, however we believe that it is due to a combination of Zipf's law and Heap's law — that is, the most frequent words will be frequent and we're not really expecting to see many new ones in a new document

We make further observations from these results: if Italian had not been included, the results would have been better; Italian is often mistaken for Spanish; and everything is mistaken for English.

These observations highlight two difficulties in language identification of European languages. First, Italian genuinely has a number of high frequency words which match English and Spanish words, so an approach such as Cavnar & Trenkle's absolute rank difference or Kulback-Liebler divergence should improve results. Second, it is difficult to accurately remove all markup and any remaining Wikimedia markup will be identified as "English", as markup is frequent this adds to the confusion.

An error analysis on by document size suggests that most errors in classifying Italian occurred in documents of between 121 and 175 words in length where about 40% were wrongly classified.

To eliminate any error introduced as a consequence of markup a further experiment was conducted using the EuroParl collection. In this experiment the two collections were tested against themselves and each other. Rather than presenting more confusion matrices, Table 6 shows the accuracy (percent of correctly classified documents) we observed when $k = 20$. The last line is the main diagonal of Table 4 represented for clarity.

This table shows that the classifiers developed for each collection work well on the other. Indeed, the column for Italian suggests that the baseline method effectively manages markup contamination in the training documents and the test documents, but suffers when both are contaminated the same way.

Increasing k does not always improve (sometimes worsening) accuracy as table 7 for $k = 200$ shows. As before, contamination of the Wikipedia data resulting from inferior Wikimedia markup removal is a problem.

The purpose of this experiment was to establish a *lower bound* that better methods should beat. What we found is that data cleansing (removal of markup from XML, removal of menus from Web pages, and so on) can substantially affect the performance of the language detection algorithm. This happens because the markup tends to be in one language, but becomes frequent in all languages making all languages look (to the classifier) more similar than they really are.

7. TRIGRAMS

We conducted two experiment with tri-grams. In the first we examined which tri-grams were frequent, and in the second we measured the performance of the approach.

The algorithm we used, "Padded trigrams", is a reduced version of the Cavnar & Trenkle algorithm using only tri-grams (recall that they used 1-5 grams). Words that were shorter

Training	Testing	de	nl	en	es	it	fr
EuroParl	EuroParl	99.4%	99.5%	99.5%	98.3%	98.0%	96.7%
EuroParl	Wikipedia	99.5%	99.3%	97.9%	97.8%	94.1%	97.4%
Wikipedia	EuroParl	99.5%	99.5%	99.5%	98.4%	96.4%	96.6%
Wikipedia	Wikipedia	99.4%	99.3%	98.4%	97.4%	86.7%	97.5%

Table 6: Accuracy for each language, varying training and test collections, k=20

Trained	Tested	de	nl	en	es	it	fr
EuroParl	EuroParl	99.5%	99.6%	99.8%	99.8%	100.0%	99.4%
EuroParl	Wikipedia	99.4%	99.3%	98.4%	97.5%	88.6%	97.6%
Wikipedia	EuroParl	99.6%	99.1%	96.0%	98.5%	99.5%	97.4%
Wikipedia	Wikipedia	99.5%	99.4%	98.6%	99.2%	90.1%	98.1%

Table 7: Accuracy for each language, varying training and test collections, k=200

than three grams were padded with space (on the right). For example, "a bag of meal" contributes "a␣␣", "bag", "of␣", "mea", and "eal".

Table 8 shows the top ten trigrams for the Wikipedia collection in four languages. Articles and conjunction are high in all four lists. The trigrams "ing", "ion", "tio", and "ent" are characteristic of nominalisations in English; there are hints of the same pattern in the other languages. This suggests that the Wikipedia may not be typical of language use. Also seen in this list are some highly frequent words (for example the articles, "the" and "a" in English).

Table 9 shows the top ten trigrams for the EuroParl collection. While the trigrams are different, a similar pattern is seen.

The presence of whole words in the top tri-gram lists may be providing additional reason for the success of the baseline approach in the previous section (and *vice versa*). Highly frequent words (at least in English) appear to be short and the short words appear to be frequent tri-grams.

To examine the performance of this algorithms, the Wikipedia collection was split into three groups: 1000 randomly chosen short documents (10 words or fewer) for testing; 1000 randomly chosen long documents (more than 10 words) for testing; and the remainder used for training. All of the EuroParl articles were used for testing.

As this algorithm can result in ties, a document is reported as "Tie" if two or more languages tied for best score.

Table 10 shows the confusion matrix for the trigram method applied to long Wikipedia documents. As before, rows show the language ascribed to a document in its collection and columns show the language it was classified as. Cells are percentages. A large percentage on the main diagonal is good and non-zero results off the diagonal show mistakes being made. The results for long Wikipedia documents are generally good. Table 11 shows what happens with short Wikipedia documents. The results are generally not good. Table 12 shows the confusion matrix for long EuroParl documents. The results are good. Table 13 shows what happens with short EuroParl documents. The results are also not good, but they are better than the results for short

en	de	es	fr
2.36% the	1.31% sch	2.31% de	1.75% de
1.00% and	1.21% der	1.09% la	0.96% ent
1.00% of	1.05% ein	0.93% en	0.88% la
0.85% ing	0.98% ich	0.86% el	0.84% ion
0.78% ion	0.88% che	0.84% ent	0.68% le
0.73% in	0.80% die	0.72% y	0.67% que
0.63% to	0.76% und	0.64% con	0.65% les
0.62% a	0.61% den	0.63% nte	0.64% et
0.58% tio	0.59% ter	0.61% que	0.61% tio
0.56% ent	0.58% ung	0.60% ado	0.59% à

Table 8: Top 10 trigrams by language: Wikipedia

en	de	es	fr
3.04% the	1.56% ich	2.03% de	1.59% de
1.32% ion	1.44% die	1.38% la	1.58% ent
1.17% of	1.42% der	1.22% que	1.40% ion
1.08% to	1.20% sch	1.17% ent	1.08% que
1.03% and	1.12% ein	1.02% ión	1.04% la
0.93% ent	1.03% ung	1.02% est	0.96% tio
0.89% tio	0.99% che	0.91% nte	0.87% ons
0.89% ing	0.92% den	0.88% en	0.87% men
0.77% in	0.83% und	0.85% con	0.86% les
0.66% hat	0.83% cht	0.79% el	0.76% l

Table 9: Top 10 trigrams by language: EuroParl

Wikipedia documents; short EuroParl documents tend to be longer than short Wikipedia ones.

8. EXPERIMENT

So far we have show that the document collections we are using are mixtures and the the performance of the baseline algorithms on short documents is substantially worse than on longer documents. In this section we show that this disparity of performance is not a characteristic of our baselines by comparing approaches of others on different length documents.

Four methods were evaluated: Cavnar & Trenkle's method, our padded trigrams method, langdetect, and the top 1000 words method. To save space, Tables 14–17 show accuracies, not entire confusion matrices. These tables show that on the Wikipedia collection the classifiers are effective on long

	en	de	es	fr	Tie
en	99.2%	0.1%	0.2%	0.5%	0.0%
de	0.7%	99.2%	0.1%	0.0%	0.0%
es	0.5%	0.2%	99.3%	0.0%	0.0%
fr	1.3%	0.0%	1.3%	97.4%	0.0%

Table 10: Confusion matrix: Wikipedia long documents

	en	de	es	fr	Tie
en	48.0%	14.8%	17.2%	13.8%	6.2%
de	14.5%	55.0%	13.3%	11.0%	6.2%
es	12.4%	9.0%	57.3%	15.9%	5.4%
fr	13.1%	9.5%	17.0%	56.0%	4.4%

Table 11: Confusion matrix: Wikipedia short documents

	en	de	es	fr	Tie
en	99.4%	0.3%	0.0%	0.2%	0.0%
de	0.3%	99.5%	0.1%	0.1%	0.0%
es	0.1%	0.0%	99.8%	0.0%	0.0%
fr	0.2%	0.1%	0.0%	99.7%	0.0%

Table 12: Confusion matrix: EuroParl long documents

	en	de	es	fr	Tie
en	74.9%	9.9%	1.0%	14.2%	0.0%
de	0.6%	99.0%	0.2%	0.3%	0.0%
es	0.2%	4.5%	94.9%	0.5%	0.0%
fr	2.9%	5.3%	1.5%	90.3%	0.0%

Table 13: Confusion matrix: EuroParl short documents

Method	de	en	es	fr
Padded trigrams	99.2%	99.2%	99.3%	97.4%
Cavnar & Trenkle	99.6%	98.6%	96.4%	97.3%
langdetect	99.5%	97.1%	96.2%	97.4%
Top 1000 Words	99.3%	98.9%	98.8%	98.4%

Table 14: Accuracy of the four methods on Wikipedia long documents

Method	de	en	es	fr
Padded trigrams	48.0%	55.0%	57.3%	56.0%
Cavnar & Trenkle	59.4%	56.4%	66.3%	61.1%
langdetect	67.0%	54.7%	69.8%	67.0%
Top 1000 Words	32.3%	25.8%	35.8%	39.6%

Table 15: Accuracy of the four methods on Wikipedia short documents

Method	de	en	es	fr
Padded trigrams	98.4%	99.5%	99.8%	99.6%
Cavnar & Trenkle	99.9%	99.3%	99.8%	99.2%
langdetect	99.4%	99.6%	99.8%	99.6%
Top 1000 Words	99.3%	100.0%	99.4%	99.5%

Table 16: Accuracy of the four methods on EuroParl long documents

Method	de	en	es	fr
Padded trigrams	74.9%	99.0%	94.9%	90.3%
Cavnar & Trenkle	93.0%	88.1%	95.2%	89.4%
langdetect	93.5%	96.2%	95.5%	93.5%
Top 1000 Words	91.6%	94.6%	90.6%	91.9%

Table 17: Accuracy of the four methods on EuroParl short documents

Method	Len	de	en	es	fr
Cavnar & Trenkle	>10	98.4%	99.7%	96.3%	96.2%
Padded trigrams	>10	99.3%	99.3%	97.1%	97.8%
Cavnar & Trenkle	≤ 10	56.2%	58.8%	62.9%	43.2%
Padded trigrams	≤ 10	56.8%	40.3%	54.2%	52.3%

Table 18: Crossover accuracy, trained on EuroParl, tested on Wikipedia

documents but not on short documents. A similar pattern is seen on EuroParl collection, however the accuracy on short documents on that collection degrades more gracefully.

To determine whether this is a collection-specific characteristic or not we performed one further experiment. In that experiment the classifiers were trained on one collection and tested on the other. Tables 18 and Table 19 show how well the algorithms performed when used in this way. It appears as though identifying the language of short Wikipedia documents is substantially harder than doing so for EuroParl documents.

9. CONCLUSIONS

In this work we examined two document collections and identified that documents do not follow a normal distribution in size, but are instead multi-modal in size. In the Wikipedia we identified two components (short and long) and observed that short documents were often redirect pages. The EuroParl corpus contained 4 components we called small, medium, large, and huge. In this collection short pages were similarly redirect pages; "see the minutes of" pages.

When we tested a simple baseline language identification technique we observed unexpectedly high performance. We also observed that performance was inhered by markup whose language model pollutes the models of each of the languages we were testing for.

Our experiments show that short documents may need different techniques from long ones. If such documents are redirect (or equivalent) pages then this suggests that an effective way of classifying short documents might be through the classification of the pages they point to. We leave for

Method	Len	de	en	es	fr
Cavnar & Trenkle	>10	99.6%	99.4%	99.8%	99.3%
Padded trigrams	>10	99.4%	07.0%	99.8%	99.7%
Cavnar & Trenkle	≤ 10	93.1%	94.2%	94.5%	91.0%
Padded trigrams	≤ 10	97.9%	57.5%	94.4%	91.9%

Table 19: Crossover accuracy, trained on Wikipedia, tested on EuroParl

further work the exploration of such an approach.

As we expected, some languages are more similar than others. The classifiers we used considered Italian and Spanish to be similar - indeed English variants may be extremely hard to distinguish as some documents could be correct both syntactically and semantically in more than one variant. We leave for further work the construction of a taxonomy of languages that might be used by a classifier to improve performance.

Finally we observe that word-based and character n-gram based classifiers respond to different aspects of a language — we leave for further work methods to combine them.

10. REFERENCES

[1] H. Alvestrand. Tags for the Identification of Languages. RFC 1766 (Proposed Standard), Mar. 1995. Obsoleted by RFCs 3066, 3282.

[2] T. Benaglia, D. Chauveau, D. R. Hunter, and D. Young. mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6):1–29, 2009.

[3] T. Bray, J. Paoli, and C. M. Sperberge-McQueen. Extensible markup language (xml) 1.0. REC-xml-19980210 (W3C recommendation), February 1998.

[4] W. B. Cavnar and J. M. Trenkle. n-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

[5] K. Hayati. Language identification on the world-wide web. Master's thesis, Computer Science, University of California, Santa Cruz, 2004.

[6] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit 2005*, 2005.

[7] O. Lauffenburger. Hittite grammar, 2006. [online, last visited on October 2012].

[8] U. Mayer. Bootstrapped language identification for multi-site internet domains. In *Proceedings of KDD'12*, August 2012.

[9] M. McCandless. Accuracy and performance of google's compact language detector, October 2011. [blog entry; last visited in October 2012].

[10] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[11] D. Raggett, A. Le Hors, and I. Jacobs. Html 4.01 specification. REC-html40 (W3C recommendation), December 1999.

[12] N. Shuyo. Language detection library for java, 2010.

[13] Wikipedia. Wikipedia:database download, 2012.