Language Set Identification in Noisy Synthetic Multilingual Documents

Tommi Jauhiainen, Krister Lindén and Heidi Jauhiainen

The University of Helsinki
Department of Modern Languages
firstname.lastname@helsinki.fi

Abstract. In this paper, we reconsider the problem of language identification of multilingual documents. Automated language identification algorithms have been improving steadily from the seventies until recent years. The current state-of-the-art language identifiers are quite efficient even with only a few characters and this gives us enough reason to again evaluate the possibility to use existing language identifiers for monolingual text to detect the language set of a multilingual document. We are using a previously developed language identifier for monolingual documents with the multilingual documents from the WikipediaMulti dataset published in a recent study. Our method outperforms previous methods tested with the same data, achieving an F_1-score of 97.6 when classifying between 44 languages.

1 Introduction

The method presented in this article has been developed as a part of the Kone Foundation funded project The Finno-Ugric Languages and the Internet[1]. The project has a need for word-level language identification between 300+ languages, including some closely related languages, as the project aims to gather texts written in small Uralic languages from the Internet. So far the project has downloaded and identified the language of several thousand million files, most of which are multilingual to some extent. The language identifier currently in use [1, 2] is capable of correctly handling only monolingual files, which means that text sections in small Uralic languages between text in other languages may not have been found. As an example of a multilingual text, we present a line from Finnish Wikipedia in Fig. 1. The example includes 7 words in Finnish and 6 words in Latin.

Multilingual language identification for corpora creation purposes has earlier been studied by Ludovik and Zacharski [3]. Multilingual language identification is also needed for automatic processing of multilingual documents in general, for example machine translation or information retrieval [3–10]. Stensby et al. [11] considered the problem of detecting the language while it is being written.

[1] http://suki.ling.helsinki.fi

Aasiankultakissa, (*Catopuma temminckii* eli *Profelis temminckii* eli *Felis temminckii*) on Kaakkois-Aasiassa elävä kissaeläin.
'Asian golden cat, (*Catopuma temminckii* or *Profelis temminckii* or *Felis temminckii*) is a cat living in South-East Asia.'

Fig. 1. Multilingual example from Finnish Wikipedia of a sentence in Finnish and Latin with the English and Latin gloss in quotes.

Automated methods for language identification have been improving steadily from the seventies until recent years. The current state-of-the-art language identifiers are quite efficient even with only a few characters and this gives us enough reason to evaluate the possibility of using existing language identifiers for monolingual text to detect the language set of a multilingual document.

2 Earlier Work

Here we briefly review the work already done in multilingual document identification. In 1995, Giguet [12] categorized sentences within multilingual documents. He managed to achieve 99.4% correct classification of sentences between 4 languages. In 1999, a vector-spaced categorizer called Linguini was presented by Prager [4]. Linguini identifies the languages and their proportions for the whole document and his method was evaluated by Lui et al. [10], the results of which are found later in this article. Also in 1999, Ludovik and Zacharski [3] segmented multilingual documents between 34 languages. Their 6 documents were artificially created and they each contained all the languages, so their task was not to detect the language set of a document, but to segment it according to the languages. Teahan considered segmenting multilingual text in 2000 [13]. He was using PPMD models for six languages. In 2006, the problem of multilingual web-documents was researched by Mandl et al. [6]. They were trying to identify which of the 8 languages known by the language identifier the text was written in by using a sliding window of 8 words. Their method reached 97% accuracy. Multiple language web pages were considered by Rehurek and Kolkus [14]. They evaluated their method with single sentences in 9 languages. Romsdorfer considered language identification with multilingual text-to-speech synthesis [15] [16]. In 2010, Murthy and Kumar [7] classified small text samples between two Indian languages. Also in 2010, Stensby et al. [11] classified multilingual documents between 9 languages with 97% average accuracy. Word level language identification in online multilingual communications was considered by Nguyen and Dogruöz [17]. They classified words between two languages, Turkish and Dutch, with up to 98% accuracy. Yamaguchi and Tanaka-Ishii [18] addressed the problem of segmenting multilingual text into language segments. Their method was also evaluated by Lui et al. [10]. In 2013, King and Abney [19] considered the problem of directly labeling the language of words between 31 languages. More recently, the problem was tackled by King et al. [9]. Their method achieves the highest accuracy of 89.94% when using 5-grams for clas-

sifying between 2 languages: English and Latin. Lui et al. [10] concentrated on identifying the presence of different languages in multilingual documents from a set of 44 languages, achieving the F_1-score of 95.9 on document-level. A new masters thesis on the subject was published in 2014 by Ullman [20], who experimented with multilingual documents in 5 languages.

Generally, the results of the previous studies can not be directly compared with each other, as the test setups differ considerably. The set of possible languages is usually different in size as well as in the selection of individual languages. The way the test corpora are generated or annotated is usually different, each containing language segments of different sizes. Lui et al. [10] created an openly available corpus, WikipediaMulti, for evaluating multilingual language identification. They used it to evaluate two previously introduced methods [4, 18] as well as their own. In order to provide comparable results we opted to utilize this same corpus[2] in the evaluation of the method proposed in this article.

3 Proposed Method

The proposed method is built on the idea of using already existing monolingual language identifiers in trying to identify the set of languages of a multilingual document. The basic idea is simply to slide an overlapping byte window of size x through the document in steps of one byte. The text in each window is sent to a separate language identifier algorithm, which gives the most likely language for the window. There is a variable called CurrentLanguage, which is first given the language of the first byte window as its value. CurrentLanguage changes after z consecutive window identifications have given a differing language from the CurrentLanguage. The document is given a label for each language that has been the CurrentLanguage at some point when going through the document.

The idea of using a window approach in multilingual language identification was also proposed by Mandl et al. in 2006 [6]. However, they used the number of words as the size of the window and the language was changed each time a different language (from a selection of 8 languages) was identified for the window. When we are handling noisy documents or the number of languages to be identified is large, or we handle languages without white space breaks, we need to have a sliding window frame and several frames agreeing on the language change before actually changing the CurrentLanguage.

4 Test Setup

We are using WikipediaMulti, which is a synthesized corpora of multilingual texts made available by Lui et al. [10]. It consists of three parts each with 44 languages: 5000 monolingual documents for training, 5000 multilingual documents for development and another 1000 multilingual documents for testing.

[2] Corpus can be found at "http://people.eng.unimelb.edu.au/tbaldwin/#resources" under the title "Multilingual language identification dataset".

All the multilingual documents have been generated by randomly concatenating parts of monolingual documents together. A separate metadata-file is used for marking the languages which should be found in each document, together with their respective sizes. Example of the metadata can be seen in the Figure 2, where document id is followed by part number (twice), language code and the size of the part in bytes.

```
doc001,1,1,de,1177
doc001,2,2,tr,394
doc001,3,3,el,1015
doc001,4,4,ru,315
doc001,5,5,es,728
```

Fig. 2. Example metadata for a multilingual document from WikipediaMulti dataset.

5 Evaluation

We trained a previously developed language identifier [1, 2] for the 44 languages in the WikipediaMulti dataset using the 5000 monolingual documents provided. The language identifier used has a few tunable parameters: the units used by the language identifier and their cut off in terms of their relative frequencies in the training material. The units we used are tokens and character n-grams from one to five, with a relative frequency of 0.0000005 as cut-off. The algorithm used by the language identifier is called token-based backoff. In the token-based backoff each token of the mystery text is given equal value when deciding the language of the whole text. The probabilities of languages for each token are calculated independently of the surrounding tokens and the average over the probabilities of all the tokens is used to determine the most likely language. Primarily the relative frequencies of tokens in the training corpus are used as probabilities, but when a previously unseen token is encountered the identifier backs off to using the relative frequencies of character n-grams.

We report the document-level averages of recall, precision and the F_1-score. Document-level averages are referred to as *micro-averages* by Lui et al. [10]. The F_1-score is calculated from the recall r and the precision p, as in (1).

$$F_1 = 2\left(\frac{pr}{p+r}\right) \tag{1}$$

We started the experiment by taking the first 100 bytes ($x = 100$) from the beginning of the document and identifying its language with the language identifier. The document was given a label with the language identified and the language was set as the CurrentLanguage. Then we moved forward one byte and sent the following 100 bytes to the language identifier, thus including 99

of the same bytes as the first one. We continued moving forward by one byte intervals until the end of the document. If the language identified differed from the CurrentLanguage 25 times in a row ($z = 25$), then the CurrentLanguage was set to the language identified last and the document was given a label with the identified language. We repeated the process to the end of the document. Giving the document labels this way resulted in a recall of 99.36%, precision of 88.50% and the F_1-score of 93.6.

Then we started to increase the length of the text to be identified. As can be seen in the Table 1, the F_1-score started to decrease after the window reached 400 bytes in length ($x = 400$) as the recall was decreasing quicker than precision was increasing.

Table 1. Recall, precision and F_1-score with differing length of byte-window.

x in bytes	z in times	Recall	Precision	F_1-score
100	25	99.36%	88.50%	93.6
200	25	99.12%	93.92%	96.4
300	25	98.72%	95.47%	97.1
400	25	98.33%	96.11%	97.3
500	25	97.77%	96.61%	97.2
600	25	97.27%	96.98%	97.1

Next step was to try to optimize z, the number of times the identification had to differ, with the text length x of 400. The results of these experiments can be seen in the Table 2. The F_1-score was clearly decreasing both directions from z being 100. Our best results on the development set were achieved using x of 400 and z of 100.

Table 2. Recall, precision and F_1-score with byte-window of 400.

x in bytes	z in times	Recall	Precision	F_1-score
400	200	97.13%	97.54%	97.3
400	100	97.83%	97.08%	97.5
300	100	98.31%	96.68%	97.5
400	50	98.11%	96.55%	97.3
400	25	98.33%	96.11%	97.3
400	10	98.43%	95.64%	97.0

The F_1-score of 97.5 was higher than the 95.9 reported by Lui et al. [10], and had reached a local optimum. We decided to try our method on the test set. From the test set we got micro-average recall of 97.87%, precision of 97.41%

and the F_1-score of 97.6 and macro-average recall of 97.86%, precision 97.66% and the F_1-score of 97.7. We have included the results from the other methods tested by Lui et al. [10] in Table 3. SegLang refers to a system by Yamaguchi and Tanaka-Ishii [18] and Linguini to a system by Prager [4].

Table 3. Recall, precision and F_1-score with different methods.

System	Recall	Precision	F_1-score
SegLang	97.5%	77.1%	86.1
Linguini	77.4%	83.8%	80.5
LLB	95.5%	96.3%	95.9
Proposed method	97.9%	97.4%	97.6

5.1 Errors with the Test Set

We decided to take a closer look at the errors made by our system on the test set. Our F_1-score was already 97.6, which meant that there were not that many errors and we analyzed them all. These errors can be categorized in 6 different categories.

Segments Written in an Unlabeled Language. There were 32 documents where our language identifier had detected English as a language without it being in the list of language labels for that document. In 13 documents, English had completely replaced one of the languages indicated by labels and in 9 cases the segment labeled with non-English contained more English than the labeled language. Six documents contained more than 200 character English incursions in a labeled language. In two documents, the labeled language contained many English words. In one document, Spanish had completely replaced Indonesian. One document contained 274 byte incursion in Russian at the end of a Hebrew part and one had 1500 bytes of French after a Georgian part. One document (wikipedia-multi/docsUE1/doc058), labeled only as Italian, was in fact multilingual, being a Wikipedia article about a common Slavic song in Macedonian, Croatian, Slovenian, Bulgarian, Russian and Polish. Another document had only names of books in English and Spanish in the part labeled Malaysian. One Estonian part consisted mostly of words in an unknown language. It was identified as Slovenian, Portuguese and Croatian by the language identifier with the 44 language selection, and as Breton when identified with the language identifier with 285 languages.

The errors in this category cannot be considered as errors with language identification, but are, in fact, errors in the labeling of the test set. There are several shorter incursions in English, and maybe in other languages as well, in many of the documents. We adjusted the length x of our detection window

according to the existing labels, which is why x grew so large that our language identifier no longer noticed the shorter incursions.

Extremely Close Languages. In these results, the most problematic close language pair was Indonesian and Malaysian. In 26 documents, they had been erroneously identified, most documents being labeled with both of the languages. The languages are highly similar as can be seen from the top 10 words in our training set for each language in the Table 4.

Table 4. The 10 most common words in Malaysian (ms) and Indonesian (id) in the training set.

word	number in ms	word	number in id
dan	2182	yang	2698
yang	1952	dan	2436
di	1368	di	1577
pada	870	dengan	1129
dengan	796	untuk	945
untuk	702	pada	929
dalam	681	dari	841
ini	614	dalam	754
the	579	ini	689
oleh	529	itu	631

The differences in frequencies of these words are not language specific. They are rather the result of the topics and domains of the randomly selected articles. The frequency list for Malaysian again brings to focus the previous errors with the corpora used. The ninth most common word in Malaysian is actually an English word and is the result of large English incursions in the Malaysian training texts.

In one document the beginning of Galego part was identified as Portuguese. Once the beginning part of Norwegian segment was identified as Danish. These languages are relatively close to each other, but much farther away than the Indonesian - Malaysian pair.

More than One Writing System for a Language. The Azeri language can be written using either an Arabic or Latin character set. The training partition for Azeri was mostly in Latin characters, which resulted in Azeri written with Arabic characters sometimes to be identified as Farsi. This could be corrected by creating two different language models for Azeri, one with Latin characters and another with Arabic characters.

Segment Consisting Mostly of Non-Alphabetic Characters. One document contained a segment labeled as Macedonian, which consisted of hundreds

of numbers and only less than 20 tokens in Cyrillic and another 20 tokens in Latin characters. Macedonian is written with Cyrillic characters, hence the segment was erroneously identified to contain Romanian, Bulgarian, and Russian in addition to Macedonian. One Malaysian labeled part consisted only of lots of numbers together with some U.S. place names. In one document, there was, after Hindi in a Hindi labeled part, many dates in numbers together with abbreviations of English months.

Place Names and Lists of Abbreviations. Two documents had excessive numbers of foreign place names, which were identified with their respective languages. One Slovenian part contained a large list of unknown character combinations, which could have been some sort of model numbers or abbreviations. Place names and lists of part numbers have also proven to be especially troublesome in the language identification done while crawling web pages.

Very Short Segments of Labeled Language. There were 27 language segments from 15 to 164 bytes in length which were not identified correctly. Also 10 longer segments were incorrectly identified. It is clear that these segments, which were shorter than our 400 byte window, were too short for the language identifier to notice. It is probable that our byte window grew so large, because there is a greater number of incorrectly than correctly labeled short language segments within the development set.

6 Discussion

We also tested identifying the languages with previously generated language models [2]. We took a subset of 43 languages from the 285 languages we used in our evaluation of the monolingual language identifier and the results are on the second line of the Table 5. We had only one language for the Indonesian/Malaysian pair, so the results cannot be directly compared. In these tests we also used z of 50. We also tested the new method with the language identifier having 285 languages to choose from. The results can be seen in the Table 5. It is notable how little difference there is between the scores, even though the task of categorizing between 285 languages is a lot more challenging than between 43 languages. This reflects the great accuracy we achieved when evaluating our language identifier algorithm, it reached 100.0% in both recall and precision already at the test length of 120 characters with 285 languages.

In order to provide a working prototype we tested the proposed method with our own implementation of the Cavnar & Trenkle algorithm [21] for language identification. We used language models generated from the WikipediaMulti training set and the number of n-grams in each of the language models was 20000. The language identifier using the Cavnar & Trenkle algorithm doesn't achieve as high F_1-scores as the one using our own algorithm [2], but it still outperforms the one proposed by Lui et al. [10]. It attains F_1-score of 96.2

Table 5. Recall, precision and F_1-score with different language models using the proposed method.

System	Recall	Precision	F_1-score
Language models from [2], 43 languages	98.30%	97.55%	97.9
Language models from [2], 285 languages	98.27%	97.33%	97.8

when using 400 byte window and a threshold z of 100. We also tested with the same window, but jumping every other byte when moving the window with reduced threshold z of 50. Jumping every other byte halves the time used for identifications, with only a small drop in F_1-score. The working prototype using the Cavnar & Trenkle algorithm can be downloaded from our web page[3].

Table 6. Recall, precision and F_1-score with language identifier using the Cavnar & Trenkle algorithm and language models from WikipediaMulti.

System	Recall	Precision	F_1-score
C & T algorithm with 20000 n-grams, no jump	97.23%	95.11%	96.2
C & T algorithm with 20000 n-grams, jump 2 bytes	97.27%	94.68%	96.0

Two thirds of the total amount of errors made by our system were directly or indirectly caused by incorrect labeling of languages in the test set. With the quality of the development and the test material at hand, we did not think it would be sensible to continue to token-level identifications. We will need a more precise dataset for that task. It would be easy and quite quick to find at least the most problematic unlabeled segments from the WikipediaMulti dataset using the method presented in this paper, but it wouldn't be correct to use the new derived dataset for the evaluation of at least the method itself.

When setting up a multilingual identification system, it is important to decide the minimum length for the text to be identified. If we are interested in loan-words, we might want to investigate character sequences shorter than tokens, if we are interested in foreign words used inside the sentences we might want to use tokens as the length and, if we are interested in sentences, the length should be set to a sentence. If we want to create a language corpus to research character combinations in a certain language (for example when calculating distances between languages), we might not want the foreign words polluting the language we are interested in. One of our next tasks will be to find or to create a more precisely labeled multilingual corpus for experiments with token-level language identification.

[3] http://suki.ling.helsinki.fi/MultiLI

7 Conclusions

We have presented a simple method to identify the language set of multilingual documents. The method uses existing language identifier designed for monolingual texts. We evaluated the method using a corpus designed for multilingual language identifier evaluation. The method presented in this article clearly outperforms the methods previously evaluated with the same corpus, reducing the average recall error by 53% and the average precision error by 30% when compared to the previously best method.

Acknowledgments This work was supported by Kone Foundation from its language programme[4]. We also thank Timothy Baldwin and Marco Lui for their help with the WikipediaMulti dataset.

References

1. Jauhiainen, T.: Tekstin kielen automaattinen tunnistaminen. Master's thesis, University of Helsinki, Helsinki (2010)
2. Jauhiainen, T., Lindén, K.: Identifying the language of digital text. In review, submitted 08/14 (2015)
3. Ludovik, Y., Zacharski, R.: Multilingual document language recognition for creating corpora. Technical report, New Mexico State University (1999)
4. Prager, J.M.: Linguini: Language identification for multilingual documents. In: Proceedings of the 32nd Annual Hawaii International Conference on System Sciences, Maui (1999)
5. Ozbek, G., Rosenn, I., Yeh, E.: Language classification in multilingual documents. Technical report, Stanford University (2006)
6. Mandl, T., Shramko, M., Tartakovski, O., Womser-Hacker, C.: Language identification in multi-lingual web-documents. In: Natural Language Processing and Information Systems. Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems, Klagenfurt (2006) 153–163
7. Murthy, K.N., Kumar, G.B.: Language identification from small text samples. Journal of Quantitative Linguistics **13** (2006) 57–80
8. Hughes, B., Baldwin, T., Bird, S., Nicholson, J., MacKinlay, A.: Reconsidering language identification for written language resources. In: Proceedings of the International Conference on Language Resources and Evaluation, Genoa (2006) 485–488
9. King, L., Kübler, S., Hooper, W.: Word-level language identification in The Chymistry of Isaac Newton. Literary and Linguistic Computing (2014)
10. Lui, M., Lau, J.H., Baldwin, T.: Automatic detection and language identification of multilingual documents. Transactions of the Association for Computational Linguistics **2** (2014) 27–40
11. Stensby, A., Oommen, B.J., Granmo, O.C.: Language detection and tracking in multilingual documents using weak estimators. In: Proceedings of the Joint IAPR International Workshop of SSPR&SPR. Volume 6218 of NLCS., Cesme, Springer, Heidelberg (2010) 600–609

[4] http://www.koneensaatio.fi/en/grants/language-programme/

12. Giguet, E.: Multilingual sentence categorization according to language. In: Proceedings of the European Chapter of the Association for Computational Linguistics SIGDAT Workshop "From text to tags : Issues in Multilingual Language Analysis", Dublin (1995) 73–76
13. Teahan, W.J.: Text classification and segmentation using minimum cross-entropy. In: Proceedings of the 6th International Conference "Recherche d'Information Assistee par Ordinateur", Paris (2000) 943–961
14. Řehůřek, R., Kolkus, M.: Language identification on the web: Extending the dictionary method. In: Proceedings of the 10th International CICLing Conference. Volume 5449 of NLCS., Springer, Heidelberg (2009) 357–368
15. Romsdorfer, H., Pfister, B.: Text analysis and language identification for polyglot text-to-speech synthesis. Speech communication **49** (2007) 697–724
16. Romsdorfer, H.: Polyglot text-to-speech synthesis. PhD thesis, Swiss Federal Institute of Technology, Zürich (2009)
17. Nguyen, D., Dogruöz, A.S.: Word level language identification in online multilingual communication. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Seattle (2013) 857–862
18. Yamaguchi, H., Tanaka-Ishii, K.: Text segmentation by language using minimum description length. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Jeju Island (2012) 969–978
19. King, B., Abney, S.: Labeling the languages of words in mixed-language documents using weakly supervised methods. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta (2013) 1110–1119
20. Ullman, E.: Shibboleth - a multilingual language identifier. Master's thesis, Uppsala University, Uppsala (2014)
21. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas (1994) 161–175