LANGUAGE SET IDENTIFICATION IN NOISY SYNTHETIC MULTILINGUAL DOCUMENTS

Tommi Jauhiainen, Krister Lindén and Heidi Jauhiainen
Department of Modern Languages

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
HUMANISTINEN TIEDEKUNTA
HUMANISTISKA FAKULTETEN
FACULTY OF ARTS

FIN-CLARIN
KONEEN SÄÄTIÖ

INTRODUCTION

The multilingual language identification method presented here has been developed as a part of the Kone Foundation funded project The Finno-Ugric Languages and the Internet. The project aims to gather texts written in small Uralic languages from the Internet. We have downloaded and identified the language of several thousand million files, most of which are multilingual to some extent. The language identifier currently in use 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.

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 $\boldsymbol{x}$ through the document in steps of one byte. The text in each window is sent to a separate language identifier, 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 $\boldsymbol{z}$ 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.

TEST SETUP

We are using WikipediaMulti, which is a synthesized corpora of multilingual texts made available by Lui et al. [1]. It consists of training, development and test parts each with 44 languages. All the multilingual documents have been generated by randomly concatenating parts of monolingual documents together.

EVALUATION

We trained a previously developed language identifier [2] for the 44 languages in the WikipediaMulti dataset using the 5000 monolingual training documents provided. 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.

Our best results on the development set were achieved using a size $\boldsymbol{x}$ of 400 bytes and a threshold $\boldsymbol{z}$ of 100 times.

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

Recall, precision and F_1-score with the development set.

The F_1-score of 97.5 was higher than the 95.9 reported by Lui et al. [1], and had reached a local optimum.

The results on the test set can be seen in the table below. We have also included the results from the other methods tested by Lui et al. [1]. SegLang refers to a system by Yamaguchi and Tanaka-Ishii and Linguini to a system by Prager. The proposed method 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.

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

Recall, precision and F_1-score with different methods.

We took a closer look at the errors made by our system on the test set. These errors can be categorized in 6 different categories:

- Segments Written in an Unlabeled Language
- Extremely Close Languages
- More than One Writing System for a Language
- Segment Consisting Mostly of Non-Alphabetic Characters
- Place Names and Lists of Abbreviations
- Very Short Segments of Labeled Language

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.

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 first line of the table below.

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

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

We had only one language for the Indonesian/ Malaysian pair, so the results cannot be directly compared. We also tested the new method with the language identifier having 285 languages to choose from. The results can be seen in the table above.

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 [3] we tested the proposed method with our own implementation of the Cavnar & Trenkle algorithm [4] for language identification. 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. [1].

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

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

REFERENCES

[1] 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

[2] Jauhiainen, T., Lindén, K.: Identifying the language of digital text. In review, submitted 08/14, 2015.

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

[4] Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: *Proceedings of SDAIR-94, 4rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas (1994) 161-175

CONTACT INFORMATION

firstname.lastname@helsinki.fi
http://suki.ling.helsinki.fi