Layered approach for language identification of multilingual documents

Team 11, AlphaDogs

Name	USC ID	E-mail
Minchul Park	8210695817	minchulp@usc.edu
Meng-Yu Chung	9208398418	mengyuch@usc.edu
Nithin Chandrashekhar	1922846582	nithinch@usc.edu
Samual Krish Ravichandran	6334599483	samualkr@usc.edu

1. Introduction

A language is more often comprised of features that extend geographically and social space. Language identification (LangID) plays a key part in many Language Processing tasks that require linguistic assumptions, one of them being machine translation. The major question that needs to be addressed: "Is it sufficient to determine the unknown language with the set of the possible language". This task can be difficult; even for commercial-level translators like Google translate or Bing Translator which often wrongly identify languages that are closely related by script (Wallon and French, Punjabi and Urdu, Hindi and Kashmiri) or lexicon (Devanagari family contains about 120 languages).

Our approach is to find those similar language groups in a systematic way by gradually include specialized identifiers to each language group by considering specific linguistic knowledge. This allows us to break down a hard problem of LangID with more than 100 languages into small modularized problems and incrementally improve the overall performance with ease. In addition to this, we also bring the idea of entropy [1] to determine a discriminative power of each word in a context of LangID. Since entropy is a measure of information contained in a probabilistic distribution, we can use it to distinguish words more useful for LangID task and give more weight on them.

On the other hand, multilingual documents pose a major problem during document translation. Previous works on multilingual identification are done mostly on a document-level [2] [3] and there have been only a few attempts at identifying multiple languages in a fine-grained way [4] [5]. Typically, sentence is a minimal text segmentation to maintain monolingual assumption; thus, sentence-level granularity is sufficient for identifying exact span of each language in multilingual documents. Hence, the problem is formulated as a sentence-level LangID here.

2. Method

- Materials

Our dataset is generated from the Wikipedia corpus. First, we downloaded entire text dump files from the Wikimedia Downloads [6] site and removed irrelevant texts, such as tags or URLs, by using wikiextractor [7]. Then we excluded namespaced pages, as many of those pages are written in secondary languages. We assumed all documents in the corpus are monolingual, although this is not necessarily true.

In order to embrace lexical ambiguity that arises from similar languages, we collected comparable monolingual documents by using interlanguage links, which member of them deals with the same topic in a different language. We randomly sampled sentences from the documents and concatenated them in a sentence level. Sentence sampling have been done in favor of minor languages, in order to guarantee every languages have an adequate size of data.

The resulting dataset contains 20,000 documents of 123 languages, each of them consists of maximum 3 languages and 10 to 15 sentences. Each sentence contains minimum 20 characters. These documents are separated into three sets: 16,000 for training, 2,000 for development and 2,000 for a test.

- Procedure

Since sentence-level granularity is sufficient for most of the multilingual identification, we formulate this problem as a sentence-level LangID task. Unicode text segmentation algorithm [9] is used for detecting sentence/word-level boundary detection.

The system is composed of several components. The first component is a script-level identifier. This component tries to identify a language by using only Unicode script information. We calculate the count of Unicode script for each document and language, and build corresponding vectors based on this information. Then we calculate cosine similarities [8] between language and document vector and select the language that yields the highest score.

$$\mathbf{V}_{\mathrm{s}} = [C(st, script_0), \ldots, C(st, script_n)]$$

$$\mathbf{V}_{\mathrm{l}} = \sum_{st_i \in l} \boldsymbol{V}_{st_i}$$

$$\mathrm{R(st)} = \underset{l \in S_l}{\operatorname{argmax}} \cos(\boldsymbol{V}_{st}, \boldsymbol{V}_l)$$

$\mathrm{C(st, script_n)}$ is the number of codepoints in a sentence st that belongs to Unicode script n. $\mathbf{V}_{\mathrm{st}}$ and $\mathbf{V}_{\mathrm{l}}$ are vector representations of a sentence st and a language l. A language l that maximizes $\cos(\mathbf{V}_{\mathrm{st}}, \mathbf{V}_{\mathrm{l}})$ is selected as a language of st.

Since Unicode script itself does not provide sufficient information for language identification, the result is supposed to be highly erroneous. The agglomerative clustering algorithm [8] is used for finding a cluster structure that minimizes the error between clusters by merging clusters in a greedy way. Documents in each cluster are subject to the next LangID task.

The next stage is a word-level identification. In this stage, we incorporate a concept of entropy for gauging a discriminative power of each word. If we have probability distribution of a certain word over languages, its entropy should display its informational value on LangID task. For instance, a low entropy value of a certain word means that appearances of this word are focused on a small set of languages. Therefore, this word is likely to carry more information for LangID than other words.

$$\mathrm{E(w)} = -\sum_{l \in S_l} p(w_l) \log p(w_l)$$

$$\mathrm{score}(w, l) = \frac{c(w_l)}{E(w) + C_e}$$

$$\mathrm{score}(st, l) = \sum_{w \in st} score(w, l)$$

$$\mathrm{R(st)} = \underset{l \in S_l}{\operatorname{argmax}}\, \mathrm{score(st, l)}$$

$p(w_l)$ and $c(w_l)$ is a probability and an occurrence count of the word w appearing in a sentence written in language l. E(w) is an entropy value of a word w. C_e is an entropy coefficient for the purpose of both a weight parameter and avoiding division-by-zero. A language l that maximizes a summation value of $c(w_l)$ divided by the entropy value of w is selected as an estimated language of st. For this task, only unigram model is used.

For the case of unknown words, we also build a word-level language identifier. In order to discover morphological properties of a particular language, we collect every possible substring for all words in each language. Word models are constructed as vectors over these morphological statistics. The models are also augmented by entropy values of each substring.

$$\mathbf{V}_{\mathrm{w}} = [C(w, s_o), \ldots, C(w, s_n)], \qquad substr = \{s_0 \ldots s_n\}$$

$$\mathbf{V}_{\mathrm{l}} = \sum_{w_i \in l} \boldsymbol{V}_{w_i}$$

$$\mathrm{p}(\mathrm{w}_{\mathrm{l}}) = \frac{\cos(\boldsymbol{V}_w, \boldsymbol{V}_l)}{\sum_{l \in S_l} \cos(\boldsymbol{V}_w, \boldsymbol{V}_l)}$$

$$\mathrm{c}(\mathrm{w}_{\mathrm{l}}) = p(w_l) \cdot |S_l|$$

$substr$ is a set of all possible substrings observed in the dataset. $\boldsymbol{V}_w$ and $\boldsymbol{V}_l$ are vector representations of a word w and a morphological model for language l. We estimate probability distribution $\mathrm{p}(\mathrm{w}_l)$ from cosine similarity values. Occurrence count values $\mathrm{c}(\mathrm{w}_l)$ are normalized to fit their average to 1. For the sake of efficiency, we limit the number of candidate languages to five for each word in our implementation.

- Evaluation

We evaluate our system described using a sentence-level accuracy of the predicted language labels, along with a baseline method of simple dictionary lookup based on word counting. We also evaluate the performance of the language identifier for unknown words.

3. Results

Due to the length limit, the detailed result are uploaded to our Github.

Baseline	Without entropy	Without word	Use all
6.12%	65.06%	89.99%	90.65%

Table 1. Performance of language identifiers

The baseline system has a low performance of 6.12% as shown in the Table 1. This is an expected result since the corpus has a considerable amount of lexical ambiguity due to the way it is generated.

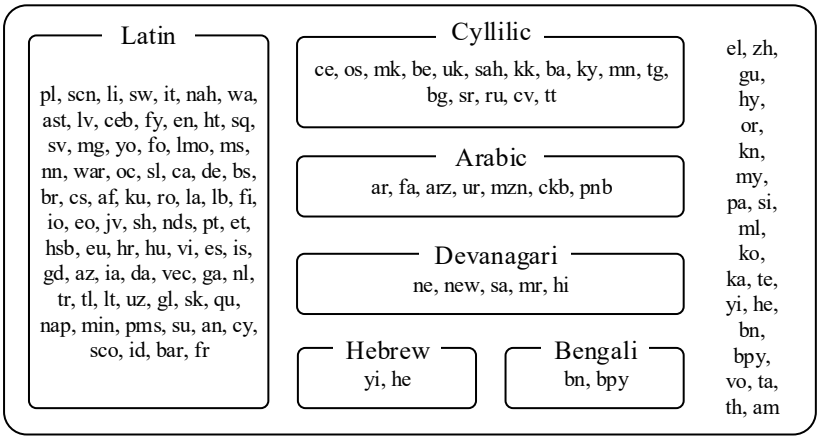

Figure 1. Generated clusters

The script-level identifier serves its purpose almost perfectly. The figure 1 displays the generated clusters based on the script-level identification result. In consideration of this identifier's original purpose is to divide the dataset according to their scripts, we can see that this objective is achieved. However, further attempts to recursively cluster languages using same script was not successful. It is discussed in the next section in detail.

Script	Accuracy on each word	Not using word ID	Using word ID
Arabic	66.78%	91.21%	94%
Devanagari	62.04%	89.68%	90.67%
Cyrillic	63.84%	94.34%	95.22%
Hebrew	94.79%	99.13%	100%
Bengali	95.49%	97.2%	99.6%
Latin	43.40%	86.55%	87.11%

Table 2. Performance of language identifier for unknown words

The performance of language identifiers for unknown words are described in the Table 2. While it is extremely challenging to estimate the language solely from a word itself without furthermore context, our result indicates that rather inaccurate estimation can bring performance improvement for LangID task. Although total improvement is marginal because the portion of unknown words is small, the performance gain on Arabic and Cyrillic languages is noticeable.

word \ sentence	0.1	0.01	0.001
0.1	80.32%	87.69%	90.65%
0.01	80.27%	87.64%	90.60%
0.001	80.23%	87.61%	90.59%

Table 3. Performance with entropy coefficient settings

We measured the performance of a sentence-level identifier with various settings for entropy coefficient C_e. As described in the Table 1 and Table 3, the result shows effectiveness of our approach. Using entropy of each word to give weight gives substantial performance gain of 15% and further parameter tuning yields 10% of improvement. While 90% of accuracy is not a very satisfactory result for a language identification task, it is worth to note that we only used unigram word features. Since we initially focused on enabling gradual improvement than improving overall performance, we believe this result can be easily improved when the clustering problem is solved and similar language sets are found in a systematic way.

4. Discussion

Our core contributions in this research can be summarized in two parts. The first one is emphasizing a discriminative power of certain features by applying entropy value of each word. The second part is dividing a larger problem into smaller problems by clustering similar languages.

The first part works fairly well. The primary reason is that our dataset has high lexical similarity, it is important to reduce effects caused by words less indicative of its language. For instance, the word "York" has a very high entropy value, because use of "New York" is so prevalent regardless of languages. We can guess that giving same weight to such words will negatively affect to the performance. In our observation, named entities usually have low discriminative power whereas long adjective/adverbs are opposite. We think future studies of language identification utilizing full dictionary/tag information may yield these kinds of interesting information. Investigation of this idea on other classification tasks could be also fruitful.

Although clustering for finding similar languages worked almost perfectly on the script level, we realized that a more sophisticated clustering scheme is required for the later stage. While we could find several obvious clusters such as a Serbo-Croatian family (*sh*, *hsb*, *bs*) in the Latin script cluster, our relatively simple algorithm failed to appreciate them. Due to sparseness of minor languages, their characteristics are insufficiently represented in the model. Thus, they are occasionally misidentified in a large set of random languages. This greedy algorithm tries to merge two clusters that can eliminate as many errors as possible in a single step. When the algorithm eventually merges the minor language, then this accidental relationship based on error rate propagates to all the other languages. In the result, we have a single large cluster of irrelevant languages. Future works should deal

with this problem.

On the other side, we found the surprising result; our relatively simple model using cosine similarity outperformed other sophisticated models by a noticeable margin. While we were not able to present concrete numbers due to the schedule constraint, we initially experimented with some other models including HMM, MEMM and CRF. Our conclusion is that these methodologies may not work very well without appropriate understanding of the model, precise tuning of hyperparameters and thoughtful feature engineering.

In addition, it is worth to note that many advanced models demand significant computational power, especially in this task mainly due to the huge number of vocabularies. For instance, nltk Naive-Bayes classifier crashed on the full dataset due to OOM error even with 16 GB of memory. A five-gram model on CRF generated billions of features which most CRF implementations are unable to deal with and even a three-gram model took 20~30 minutes to train the entire dataset. In the perspective of a problem size, we expect solving the clustering problem stated above can alleviate this computational problem.

5. References

[1] A Mathematical Theory of Communication, Bell System Technical Journal, 1948.

[2] Reconsidering Language Identification for Written Language Resources, LREC, 2006.

[3] Automatic Detection and Language Identification of Multilingual Documents, Tran. ACL, 2014.

[4] A Fine-Grained Model for Language Identification, Proc. SIGIR, 2007.

[5] Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods, NAACL HLT, 2015.

[6] Wikimedia Downloads, https://dumps.wikimedia.org/

[7] wikiextractor, https://github.com/attardi/wikiextractor

[8] Introduction to Information Retrieval, Cambridge University Press. 2008.

[9] Unicode® Standard Annex #29 UNICODE TEXT SEGMENTATION, http://unicode.org/reports/tr29/, 2015

6. Division of labor between the teammates

	Theory	Coding	Data	Writing
Minchul	O	O	O	O
Meng-Yu		O	O	
Nithin		O	O	O
Samual	O	O		O

This document contains 1959 words except the reference part.