COSE474-2024F: Final Project Proposal
Image Language Identification using Pre-Trained Models

James Watson

Abstract

My project proposal is a two step image language identification model, the goal is to take an image with text, convert the text embedded within the image to a string, and then identify the language(s) of the text in the string.

1. Motivation

Automatic language identification (LID) is a very important task in natural language processing. By quickly identifying the language of text in an image we can quickly pass the data to machine translation or content moderation systems.

Throughout my time in Korea I regularly rely upon the use of image based translation apps like Microsoft Lens and Papago, and I understand the importance of how quickly these applications must identify language.

By using a pre-trained model we can reduce the need for very large labeled datasets, simplifying the process of language identification.

2. Related Work & SOTA

Traditional LID systems relied on statistical techniques to model distributions of different languages, such as Cavnar & Trenkle (1994). (Cavnar and Trenkle, 2001)

Neural networks have seen much use in natural language processing and LID systems, particularly Joulin et al.'s FastText, which uses word embeddings trained on large text corpora. (FastText, 2015)

Pre-trained models are a good way to perform language identification, for example mBERT is a multilinual transformer model based on BERT, which performs well on LID tasks, but struggles on underrepresented languages in its training data, additionally mBERT was not trained for language identification. (Devlin et al., 2018; Google, 2018a)

There is also the XLM-R model, which is a state of the art model for multilingual NLP tasks. (Google, 2018b)

3. Challenges

There are a few major challenges with language identification:

1. Languages with similar grammatical structures or scripts can be hard to differentiate (i.e. Spanish and Portuguese, or the use of Chinese characters in Japanese)
2. Lack of data for some languages
3. Lack of data on the language of text embedded in images, which means I cannot directly verify the final results

4. Problem Definition, Models & Datasets

The goal is to build a image language identification system that will classify the text on an image into a language. Google Cloud Vision API (Google, 2019) or EasyOCR (JaidedAI, 2019) will be used to read the text into a string. XLM-R or mBERT will then be trained on Tatoeba (Tatoeba, 2018), JW300 (Agić and Vulić, 2019) or a Kaggle Language Identification Dataset where the target is the language, and use the model to determine the language of the string.

As we are using two separate models, we should test them individually, and calculate their F1-score for classification performance. We will then test them together, but since we don't have data on the language of images, we will compare our data against another pre trained LID model like FastText.

5. Goals and Schedule

1. Download data and pre trained models
2. Check efficiency and capability of models on their existing tasks
3. Fine-tune language model for LID
4. Setup pipeline image > string > LID
5. Compare results of language model with results generated through FastText

References

Agić, Ž., and Vulić, I. *JW300: A Wide-Coverage Parallel Corpus for Low-Resource Languages*, 2019.

Cavnar, W., and Trenkle, J. *N-Gram-Based Text Categorization* (p.), 2001.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2018. http://arxiv.org/abs/1810.04805

FastText. *https://fasttext.cc/*, 2015.

Google. *https://huggingface.co/google-bert/bert-base-multilingual-cased*, 2018a.

Google. *https://huggingface.co/google-bert/bert-base-multilingual-cased*, 2018b.

Google. *https://cloud.google.com/vision*, 2019.

JaidedAI. *https://github.com/JaidedAI/EasyOCR*, 2019.

Tatoeba. *https://tatoeba.org/en/*, 2018.