Language Identification with Deep Learning

Ahmet Tarık KAYA
Computer Engineering
Hacettepe University
Çankaya, Ankara
ahmet.tarik.kaya@gmail.com

Kaan Mersin
Computer Engineering
Hacettepe University
Çankaya, Ankara
kaanmersin07@gmail.com

***Abstract*—Language Identification is one of the first tasks of Natural Language Processing according to many different papers. Nowadays most of the effort of the scientist that are working on this topic is focused on methods that has Neural Networks and Deep Learning approaches. We are focusing on the same topic. In this paper we are going to explain the information that we learn from similar papers and the main approach that we are going to use while we develop a model.**

1. Related Work

Language Identification is a popular topic that computer scientists have been working on for many years. Because Language Identification is a old topic, many of the models that have been developed on the topic doesn't work with a Neural Network approach. Our goal is developing a model that has a Neural Network, so we focused on the papers that has same approach.

We mainly focused on three different papers. These papers are focusing on different problems that data scientists faced while dealing with Language Identification.

The first paper that we studied was a paper with title ”Language Identification: a Neural Network Approach” [1]. This model is using a dataset that contains 25 different languages. This is a basic Neural Network model. With this approach, model was able to work with a precision of 95% and only failed in identification of Portuguese language. Model uses the dataset in two steps; using training set for alphabet identification and creating a trigram for the languages itself. For alphabet identification, model creates a trigram and then merges it with trigram that contains languages. For the merge operation, models firstly merge these two trigrams for each language and then does the same operation for the entire dataset. They used forward propagation in output layer and gradient descent as the models cost function.

The second paper that we studied was a paper with title ”Text Language Identification Using Attention-Based Recurrent Neural Networks” [2]. This model is using Attention-Based Recurrent Neural Networks(ARNN). Other model are using Neural Networks and this models developers are saying that basic Neural Networks requires large texts as input to make a good language identification but this will slow down the training process. ARNN is a solution for this problem. The model starts with Bidirectional Long Short-Term Memory Neural Network (BiLSTM). BiLSTM has two hidden states and when a hidden state is calculating, the Neural Networks is using both previous and next hidden state. This process is one of the main solutions for short texts in training dataset. In the identification process, ARNN uses softmax activation function for detecting the characters and relations between characters with the highest effect in the language identification and after returns vector of probability as an output for each texts. The language with the highest probability will be taged as the language of that text.

The third paper that we studied was a paper with title ”Language Identification from Text Documents” [3].This model achieved 95.12% accuracy. In this model, they tried three different approaches and find the best qualities of each approaches. First approach is Multinomial Naive Bayes(MNN). This approach is working good in the identification between two non-similar languages. They experimented MNN with word n-grams and character n-grams. But this approach is not useful in this topic because MNN is using n-grams. In different languages same n in n-grams are not working sufficiently. Some n's are too small for the language and some n's are too large. As a result of only using MNN, character and word n-grams works in nearly same performance. Second approach is using Logistic Regression while the MNN model is working. With Logistic Regression character level n-gram's performance increased and it become a better method then word n-grams. Last approach was Recurrent neural Network(RNN). As I mentioned before, MNN is a good method when the dataset contains non-similar languages. So to solve this problem, they used RNN. Like the ARNN in the previous paper, RNN is a cyclic neural network between the hidden states. So as a result, they used single hidden layer RNN with gated recurrent units in their model.

2. Approach with figure of architecture

Our model consists of 3 main layers.

- Embedding Layer with 16 embedding dimensions
- Long Short-Term Memory(LSTM) Layer with 32 hidden layers

- Dense Layer

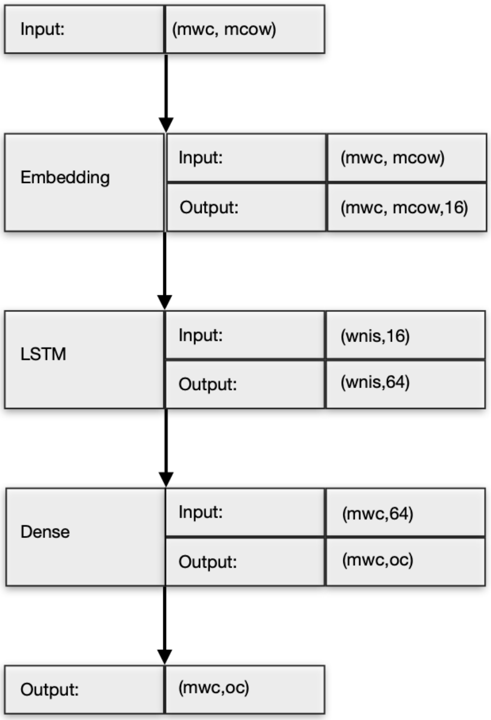


Figure 1. Architecture of our model

Here is the meanings of abbreviations:

- oc: Output class counts
- mwc: Max word number vectorizer can represent
- mcoc: Max char number a word can contain
- wnis: Number of words the text contains

Here is how our model works step by step

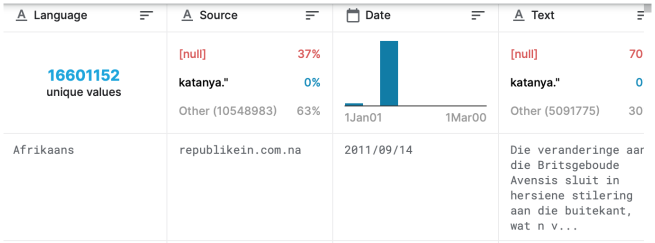
- Our input text is translated into a vector using a pre-trained vectorizer from seqtolang [4] model.
- Resulting vector has a size of [986,199]. 986 is the max number of words our vectorizer can handle and 199 is the max number of chars a word of text can contain. This values are decided by training operation of our dataset.
- Vector of the input text is fed into to model.
- First layer of our model is an embedding layer with 16 dimensions.
- When our vector with the size of [986,199] fed into embedding layer, it's output is a vector with size [986,199,16]
- Embedder's output values for every possible char field is summed which results with a vector sized [986,16]
- As said before, 986 represents maximum number of cells that could be stored in vector for the text. If the text contains 4 words, only first 4 of those 986 cells are filled. Filled cells of the tensor is detected and gets extracted.
- Resulting vector that has the shape [number_of_words,16] is feeded into our LSTM as a sequence.
- LSTM outputs a vector with shape [number_of_words,64]. This output is again fitted into a tensor with max word count. Which means it is transformed to a vector with shape [986,64]
- Every word in this vector is fed into a dense layer. This dense layer takes 64 inputs an outputs with the number of output classes. In our model, this number is 40.
- So when the vector with shape [986,64] is fed into that dense layer, it results with a vector shaped [986,40]. This is output of our model.
- Softmax is applied to that output. As it is said earlier, 986 represents maximum numbers the text could contain. If the input text doesn't have 986 words, filled cells of the tensor is detected and gets extracted again.
- Mean of every words probability for every class is taken. Resulting vector with shape contains the text's probability for every class.

We take the class that has highest probability as the prediction of the model. This model is developed using PyTorch [5].

3. Dataset and Evaluation Metric

As we stated in the Proposal Report, we are using Old Newspapers Dataset. This dataset is a created by using HC Corpora. HC Corpora contains many texts of different languages from newspapers, social media posts and blog pages. Old Newspapers Dataset is a cleaned version of HC Corpora which only contains data from newspapers.

Our dataset is in ".tsv" format. This format stores data in tables. "old-newspapers.tsv" contains a table with four columns. These columns are "Language", "Source", "Date" and "Text". In our model we are not using "Source" and "Date" data. They are irrelevant to our models purpose. We are organizing our data by "Language" columns and using the data in the "Text" column as a training and testing date.



Language	Source	Date	Text
16601152 unique values	[null] 37% katanya." 0% Other (10548983) 63%	1Jan01 1Mar00	[null] 70 katanya." 0 Other (5091775) 30
Afrikaans	republikein.com.na	2011/09/14	Die veranderinge aar die Britsgeboude Avensis sluit in hersiene stilering aan die buitekant, wat n v...

Figure 2. A example row of the Old Newspapers Dataset Table

Old Newspapers Dataset contains 16,806,041 sentences/paragraphs in 67 different languages. There are different types of alphabets in these dataset. These many languages is making it hard to develop a model in language identification. Therefore we decided that, in the first steps of development we are only going to use 6 languages. These languages are: ”English”, ”Spanish”, ”Italian”, ”French ”, ”Portuguese” and ”German”.

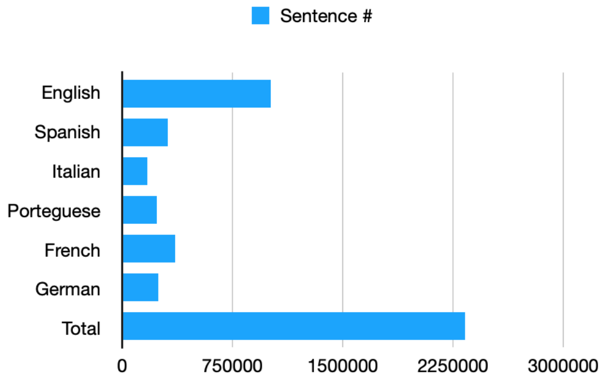


Figure 3. Language Distribution of Filtered Dataset

After we filter our dataset with the languages that we stated before by uploading our data to a SQL database, to total number of sentences reduced to 2329178. 1010242 of them are in English, 309918 of them are in Spanish, 358001 of them are in French, 236644 of them are in Porteguese, 169630 of them are in Italian and 244743 of them are in German. It didn’t suprise us that the nearly half of sentences are in English. Because in our six languages, English is the most used one by biggest population.

4. Results - Baseline

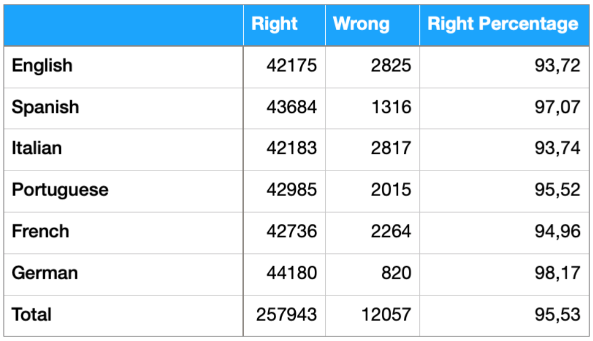
This section of the report contains the data that we collect by running our model with the test dataset that we produced from Old Newspapers dataset. As we mentioned beforem, we are only using ”English”, ”Spanish”, ”Italian”, ”French ”, ”Portuguese” and ”German” languages. Before we talk about the outputs, we want to write our predictions of outcomes of the test.

All six languages that we picked are in the same alphabet. When we searched about the structures and word types in these languages wrote down some important notes and create some predictions from those notes. These predictions(we numbered them in order to recall them easier in rest of the paper) are:
Prediction 1- Some words in English language and German language are similiar.
Prediction 2- Sentence structures of Spanish language and Portuguese language are similiar.
Prediction 3- English words are commonly used by other languages. Therefore our model will wrongly tag sentences as English language when they are not.
Prediction 4- Sentences in Italian language will be wrongly taged as Spanish or Portuguese languages.
Prediction 5- French is the only language that we can’t pair in our picked six languages. Therefore we are expecting that our model will taged sentences in French with a high accurancy.

These are our predictions from our model. As a result of our predictions we can say that, we are expecting French language to have the highest truth rate. Our predicted order is French, English, German, Spanish, Portuguese and Italian.

In the testing phase, we used same amount of sentences from each language. English language is the language that has highest number of sentences and Italian language is the one with the lowest number of sentences. So we didn’t want test results to get effected by these numbers. We want to see the homogenuesly distrubuted results to see the parts that we need to change in our model for better results.

These is the table that we generate with our results from our models test. We made some predictions before the test phase. The order of the languages by truth rate is German, Spanish, Portuguese, French, English and Italian. This order is different in many points from our predicted order. German language is the language with the highest truth rate and it has substantially high truth rate then other languages. When we looked our our sentences in German, we saw that most of the sentences contains more then 10 words. We were expectiong Italian to has the lowest rate because Italian has very similiar structures to other languages.



	Right	Wrong	Right Percentage
English	42175	2825	93,72
Spanish	43684	1316	97,07
Italian	42183	2817	93,74
Portuguese	42985	2015	95,52
French	42736	2264	94,96
German	44180	820	98,17
Total	257943	12057	95,53

Figure 4. Test Output

It was logical for us to talk about the German and Italian languages according to ”Figure 3. Test Output” but for the other languages we need some extra data. So we produced a Confussion Matrix to see the tags that our model taged to sentences wrongly.

With our Confussion Matrix it is easier to talk about our previous predictions and produce new conclusions that will help us in increasing the truth rate of our model. In Prediction 1, we stated that English and German languages will be wrongly taged as each other and our prediction was right. English is the language with highest percentage that taged wrongly when the sentence was in German language. German was the second highest percentage for English in the same situation. In Prediction 2, we stated that same

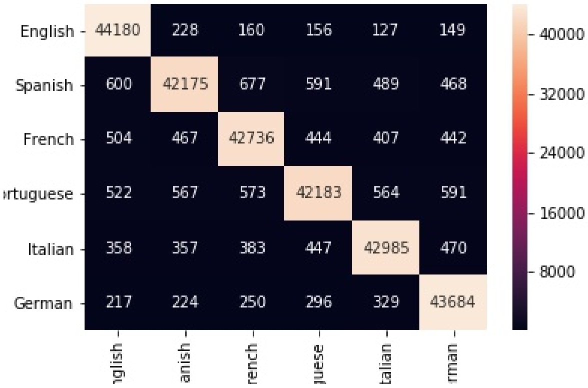
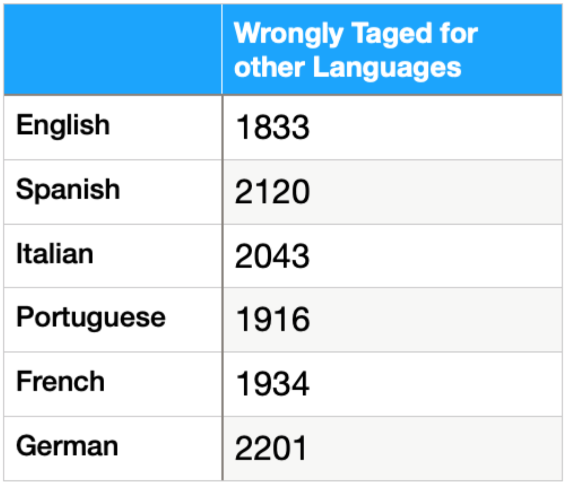


Figure 5. Confusion Matrix

problem will occur between Spanish and Portuguese, and this prediction is true too. Too solve this too problem we tought that we will set a minimum lenght for sentences. So that our model will learn the meanings of the word sets of languages better. In Prediction 4, we said that Italian sentences will be wrongly taged as Spansh or Portuguese and this was right too. But we don 't think that using longer sentences are the right solution for this problem. We didn't made much changes in our model because in the first step we wanted to find the most common result. Our model is not using punctuation signs. We think that this problem of Italian will be developed by using the punctuation signs in our models train and test phases. In Prediction 5, we said that French will be the language with he highest truth value. But we were wrong. But in this point we are not sure that the solutions that we produced for other problems are going to solve this problem. So we are going to make some research about this topic.

This table (table in Figure 6) contains information about each language when they wrongly taged when the sentence was in another language. In Prediction 3, we thought that English will be the language with the highest numbers in this table. But on the contrary English is the language with lowest numbers in this table. The reason of this is, our model is very good in taging English sentences. German is the language that has highest numbers in this table. We think that reason of this is the same reason of the problem in Prediction 1. German sentences are not long enough for our model to learn the most key notes of German language structure and words.

As a results of the test, we decided to run our model with longer sentences as inputs and try using the punctuation signs. After trying this assumptions that we made, we will try for more solutions that can make our model. This is the point that we are at in this phase of the project. We believe that our truth rate will increase to 97% for these six languages at the end of this term project.



	Wrongly Taged for other Languages
English	1833
Spanish	2120
Italian	2043
Portuguese	1916
French	1934
German	2201

Figure 6. Languages Wrongly Taged Datas

References

[1] https://pdfs.semanticscholar.org/17b2/c9ae27d2ef1b6b901a0afd5aa8649f7795bf.pdf

[2] https://www.researchgate.net/publication/333392357_Text_Language_Identification_Using_Attention-Based_Recurrent_Neural_Networks

[3] http://cs229.stanford.edu/proj2015/324_report.pdf

[4] https://pypi.org/project/seqtolang/

[5] https://pytorch.org/docs/stable/nn.html