Language Detection Using a Neural Network

Maddie Bauer

Bellevue University

DSC 680: Applied Data Science

Professor Catherine Williams

July 21, 2021

Introduction & Problem Statement

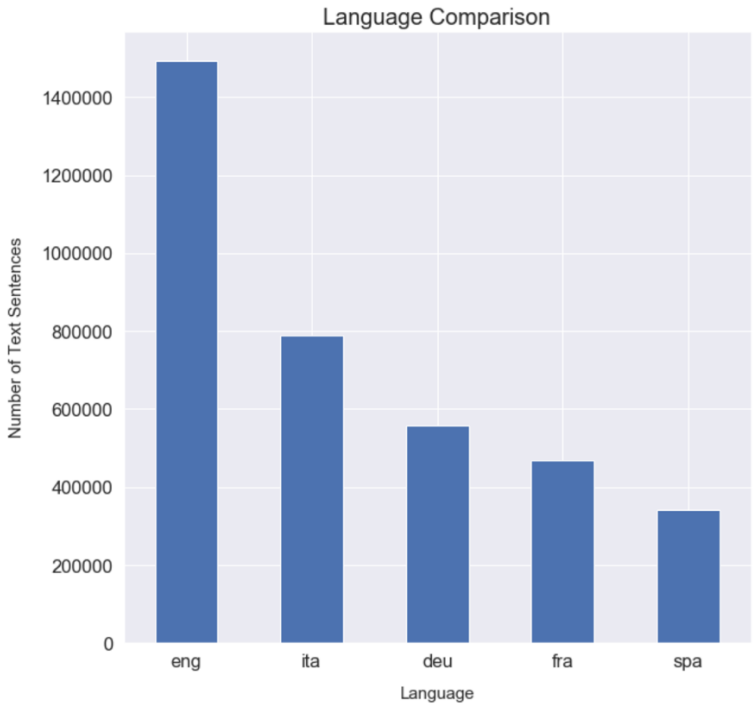
Language detection is often an important step within a Natural Language Processing (NLP) problem. The goal of this research paper is to predict the natural language of a word, sentence or other piece of text by creating a neural network. This task is important because it is necessary to know the language of the given text before any other actions, such as translating, sentiment analysis or autocompleting, can take place. In today's world, text comes from so many different sources and places, such as emails, news articles, comments, online reviews, speech-to-text applications, social networks, and many more. A language detection model is needed in order to know what to do next with the available text. For example, in a Word document the spell-checking feature must first determine what language the text is being written in before it can provide you with the correct spelling of a word (Edell, 2018). Another use case could include applying the right subtitles or closed captioning to a presentation or video as well as being able to route emails to the right geographically located customer service department for a large international company (Edell, 2018). Detecting the natural language of text data is ultimately used to understand text data better, improve a user's experience and make content more accessible.

By the end of this research paper, we will have:

- Preprocessed text data
- Built and trained a Deep Neural Network using the Keras library
- Evaluated our model on the test set

Data

The dataset being used for this research paper was retrieved from Tatoeba. Tatoeba is a free online database with example words and sentences geared towards foreign language learners (Tatoeba, n.d.). The dataset consists of a total of 9,714,915 pieces of text in 328 different unique languages. English dominates the dataset with nearly 1.5 million text entries. For my analysis, I took a subset of this dataset and used only the English, Spanish, German, French and Italian languages. Below is a breakdown of what the dataset looks like after sub-setting for these five languages.



I further subsetted this dataset to a total of 250,000 data points – 50,000 rows per language so that each language was represented equally and so that my computer will not have any issues working with too large of a dataset moving forward. There are only two variables within the dataset, language and text. I will be looking for

commonalities within the text of each language to help with the language detection model described next.

Methods

After sub-setting the dataset and removing the punctuation from the text column, I then split the dataset into training (70%), testing (10%) and validation (20%) sets. Next was conducting feature engineering in order to get the data into a format that a neural network can understand. I created a feature matrix from a list of unique n-grams for each language, specifically trigrams. An example of a list of trigrams for the phrase “hello there” follows as hel, ell, llo, lo_, o_t, _th, her, ere, re_ and so on. I also encoded each language to a numeric value as the language variable is categorical. I was then able to build a neural network using the Keras library. Let’s discuss an overview of Keras and neural networks before any further discussion with the methods used in this project.

Keras is a high-level API written in Python for deep learning purposes. Keras runs on top of the machine learning platform, TensorFlow, and was created to enable fast experimentation and implementation of neural networks (About Keras, n.d.). Keras is fairly simple to learn and work with due to its Python frontend and use of different back-ends for computation purposes (Simplilearn, 2021). Due to this structure, Keras is slower than other deep learning frameworks, but it is perfect for beginners working on neural networks. Keras is widely used today from academic researchers to engineers at large companies such as Netflix, Google, Uber, Yelp, Square and many more as well as within machine-learning competitions on Kaggle.com (Chollet, 2018, p. 61).

Neural networks fall within a subset of machine learning in which they mimic and reflect the behavior of the human brain. They are created and implemented to recognize patterns and solve common problems in the fields of artificial intelligence, machine learning and deep learning that may otherwise be time consuming tasks if completed manually. Training a neural network revolves around the following items (Chollet, 2018):

- *Layers* which are considered the core building blocks of a neural network. A layer is a data-processing module that can be thought of as a filter for the data.
- *Input Data* and their corresponding *targets*
- A *loss function* is how a network is able to measure its performance on the training data.
- An *optimizer* is the mechanism through which a network will update itself based on the loss function and the data it has already seen.

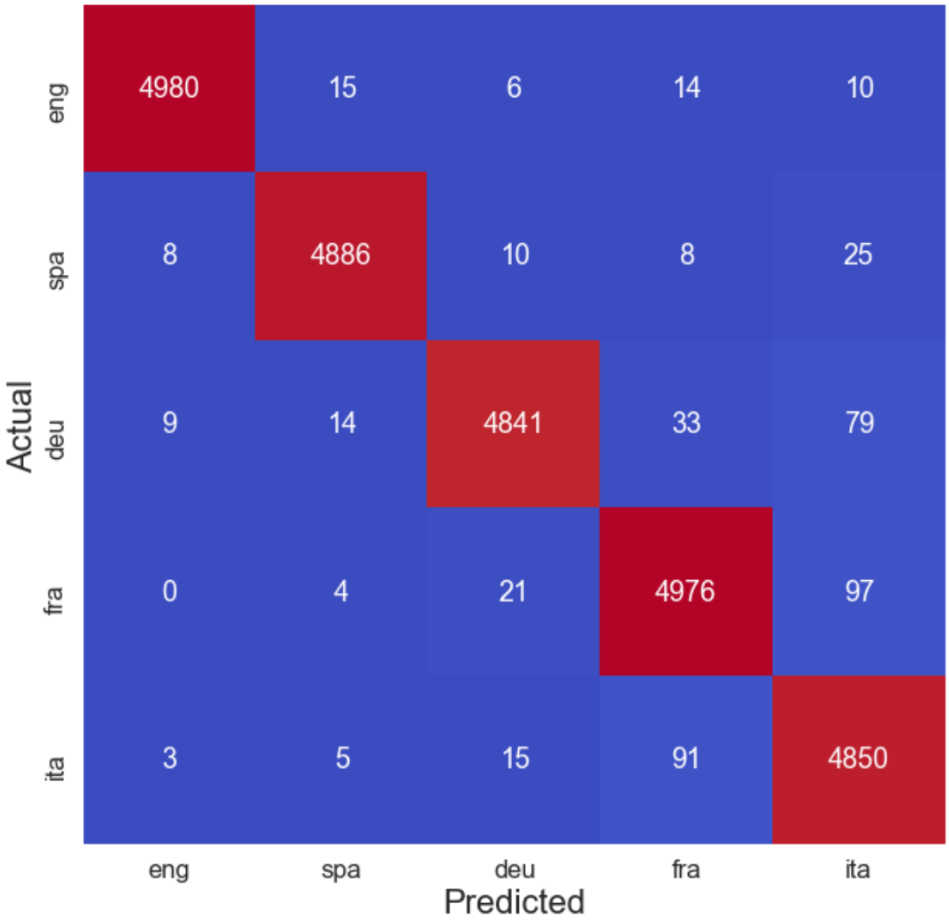
The neural network created for this analysis has four layers, three with a Rectified Linear Unit (ReLU) activation function and the final layer with a SoftMax activation. ReLU is a very common activation function in which the function returns 0 for any negative input value and simply returns the same value if the input is positive (Mujtaba, 2020). SoftMax activation is used in the final layer as it is the proper function for multi-classification problems. The neural network also uses the categorical cross entropy loss function and the RMSProp optimizer, both of which are very common as well in deep learning.

Results

During the training process, a model can become biased towards the training and validations sets. This is why we also split the original dataset into three groups, training, validation and testing sets. It is best to compute a model's accuracy on unseen data so I

evaluated the model on the testing set. The training data accuracy was 98.36% and the final accuracy on the test set was 98.132%. Minor overfitting may have occurred with the training data but overall, this is a healthy accuracy number.

We can get a good idea of how well the model performs for each language by looking at the confusion matrix below. The diagonal boxes in red give us the number of correct predictions for each of the five languages. The other boxes give us the number of times a language was incorrectly predicted. For example, English was incorrectly predicted as Spanish in 15 cases, Spanish was incorrectly predicted as French in 8 cases, and so on. The worst scenario is that French was predicted as Italian 97 times.



Conclusion

The goal of this analysis was to create and train a neural network to detect the difference between the English, Spanish, German, French and Italian languages. Natural Language Processing projects need to identify which language their input text is in order to move forward and complete any remaining tasks. For example, an online chatbot needs to determine what language the user is communicating with in order to provide their answers in the correct format. When it comes to adjusting sales and marketing strategies, businesses often apply sentiment analysis on their reviews, comments, etc. In order to properly conduct this task, the language must first be determined. While these are just a few real-life applications, businesses can take advantage of language detection models to break the language barrier between countries as well as connect with more people all around the world.

Acknowledgments

I would like to thank Tatoeba, a free online database of example sentences geared towards foreign language learners, for providing the dataset used in this project (Tatoeba, n.d.). Also, I would like to thank our Professor Catherine Williams from Bellevue University as well as the students of DSC680 for their continuous support throughout this course. Finally, I would like to express my gratitude for Wikipedia, Medium.com, kdnuggets.com, stackoverflow.com and their countless authors for their research and posts on the subject matter. This report would not have been possible without their contributions and sharing of information online.

References

About Keras. (n.d.). Retrieved July 15, 2021 from https://keras.io/about/

Bogdanov, V. (February 15, 2019). 8 thought-provoking cases of NLP and text mining use in business. Retrieved July 16, 2021 from https://becominghuman.ai/8-thought-provoking-cases-of-nlp-and-text-mining-use-in-business-60bd8031c5b5

Chollet, F. (2018). *Deep Learning with Python*. Manning Publications Co.

Edell, A. (April 25, 2018). *How I trained a language detection AI in 20 minutes with a 97% accuracy*. Retrieved July 15, 2021 from https://towardsdatascience.com/how-i-trained-a-language-detection-ai-in-20-minutes-with-a-97-accuracy-fdeca0fb7724

Mujtaba, H. (August 29, 2020). What is rectified linear unit (ReLU)? Retrieved July 16, 2021 from https://www.mygreatlearning.com/blog/relu-activation-function/

N-gram. (April 20, 2021). In Wikipedia. Retrieved July 15, 2021 from https://en.wikipedia.org/wiki/N-gram

O'Sullivan, C. (August 25, 2020). *Deep neural network language identification*. Retrieved July 15, 2021 from https://towardsdatascience.com/deep-neural-network-language-identification-ae1c158f6a7d

Tatoeba. (June 27, 2021). In Wikipedia. Retrieved July 15, 2021 from https://en.wikipedia.org/wiki/Tatoeba

Simplilearn. (March 12, 2021). *What is Keras? The best introductory guide to keras.* Retrieved July 15, 2021 from https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras

Questions:

1. Why limit to 5 languages? Why these 5 in particular?
2. Is there a machine learning algorithm that could perform the same way?
3. What about the *langdetect* function?
4. What are the benefits of detecting the language for a company?
5. Can smaller, family-owned business benefit from language detection?
6. How do you determine the number of epochs for the neural network?
7. Why trigrams? Could you have used unigrams and/or bigrams too?
8. Can you predict the language of new text with your model?
9. Would you consider using more languages in the future?
10. What are more applications/use cases?