**Data Science and Machine Learning**

**Project UNIL\_TUDOR**

**Goal:**

In this project, our aim was to create a startup that would revolutionize the way people learn and improve their command of a foreign language.

Our main goal was therefore to predict the level of difficulty of French text. To achieve this, we had to create a model for English speakers that could classify French sentences by language proficiency levels (A1, A2, B1, B2, C1, C2).

To do this, we began by concentrating on the models we had mainly seen in class, and which had been suggested to us, such as:

· kNN model

· Logistic regression model,

· Decision tree classifier model

· Random forests classifier model

After training these models, we realized that the results obtained didn't really match our expectations. That's why we set out to find a new model that would perfectly match what we were looking for and focused on new models: the BERT and CamemBERT model. CamemBERT is a natural language processing (NLP) model based on BERT, but designed specifically for understanding French texts, which is exactly what we need for this project.

**Part 1:**

For this first part, we decided to proceed in the same way for all the models. In fact, in each model we decided to use a TFIDF (Term Frequency-Inverse Document Frequency) vectorization approach because this model converts text into numerical vectors, which is necessary because the machine learning models, we use cannot work directly with plain text. After that, we assembled the vectorizer and classified it in a pipeline to finally train the model. As far as the train-test ratio is concerned, after doing some research on the internet, we realized that it was best to have an 80/20 ratio. We could also do it with another ratio, like 70/30 or 50/50, but knowing that many sites recommended the 80/20 ratio, we decided to go with that (Tokuç & Tokuç, 2023).

For the Logistic Regression we use also a TFIDF and the spacy tokenizers to tokenize the sentences. We also try to remove the stop words because in French language there is a lot of stop words around 500 in French and 326 in English and the stop words in French like “Désormais, Dorénavant…” can have a big impact on the level of a sentence. We can see that without the stop words we have the better metrics (0.444 of accuracy).

Once we had all this, we decided to show the Accuracy, Precision, Recall and F1 score of the model. For each model we realized that the scores obtained were not very satisfactory, so we tried to improve the score by performing hyperparameter optimization with GridSearchCV. We can see that this worked very well, as the score obtained after this was better than before for each model, except for the Random Forests classifier where the results obtained after hyperparameter optimization were less good. Optimization was not necessarily recommended for this project, but we did it anyway to see the difference in scores before and after optimization. After testing all the models in Part 1, i.e. the kNN model, the Logistic regression model, the Decision tree classifier model and the Random forests classifier model, we quickly realized that the scores obtained were well below our expectations, especially when compared with those of our colleagues. That's why we tried to find a model that we hadn't necessarily seen in class and that might have been suitable for the project.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquementHere you can see our results for the different models in part 1

**Part 2:**

After the part 1, we decide to focus on more complex models.

And we can separate these models in two groups. The LSTM & CNN models and the BERT and Camembert.

One member of our group discovered the LSTM and CNN model during an course of Information Security & privacy at the University of Lausanne. They did some research and they discover that these models are good for this project. In all these models we have to encode the difficulty level to be used in this model for the training, evaluation and prediction. We use the label\_Encoder to do this. Furthermore, we also used the same optimizer “AdamW” and the same loss function “CrossEntropyLoss" for a logical comparaison of the metrics.

**LSTM model :**

LSTM model: The LSTM model is a neural network model that has the capability to remember information over a long duration. This model is distinguished in three parts. In the first part, we label and encode our data with a label encoder so that our difficulty levels change from A1, A2, B1, B2, C1, C2 to 0 to 5 respectively for each label.

Next, we will create our model, evaluate it to obtain metrics, and finally predict on the unlabeled data that we have at our disposal. Embedding is what allows us to transform the words in our sentences into vectors. To do this, the parameters of this embedding can be calculated quite easily. The two pieces of information we need are the number of words in each sentence and the total number of unique words in the entire dataset. Therefore, we have calculated them to try to have number who are reprenstative for our dataset and not to use random number. The result of our model. Here are the results obtained for the length of words in each sentence:

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

As we can see, 99% of the sentences are less than 73 words, but there are a few sentences that are significantly longer than 73. The highest number recorded is 265. Therefore, we have chosen to use 100 and 265 as parameters for this model, as well as for all other models that will follow. As for the number of unique words in the entire "training dataset," it amounts to 15,773. The results of these models are not necessarily very good, as you can see below.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**CNN model :**

As can be seen in our CNN model, the structure is similar to the LSTM. As explained earlier, we tested this model because when I did my work in ISP, I also dealt with the CNN model. The structure is really similar to that of the LSTM model. However, after conducting research, we noticed that this model is not at all suited for texts but rather for images. Nevertheless, we wanted to see how it would react to text only. The result is not good for a model of this type, but it shows how powerful it is given these results. If we compare it to the results of part 1, it is above some models and even the majority.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

For this model, we reused the values in terms of sentence length and the number of unique words, but we only used 100 as the sentence length in our parameters because it was the best-performing model for the LSTM model.

**BERT model & Camembert model**

For our last two models, we will discuss them together in the same paragraph as they are closely related. Let's first introduce the BERT model.

But before discussing the BERT model, we cannot overlook the "Huggingface" community, a group in the field of artificial intelligence that can provide extensive documentation in the area and a united community. Within this community, we find the Transformers library. This library is open source and offers tools and APIs that can be easily downloaded for use. We had already used the "Transformers" library in this Data Science and Machine Learning course during a practical session to determine the sentiment of an English sentence (positive, negative, or neutral). Thus, it is thanks to this library that we had access to BERT and also Camembert, in the same vein we also have BERTA and FlowBert which we unfortunately did not have the time to experiment with.

Now we can delve into the heart of the matter. BERT utilizes a significant bidirectional mechanism worth noting. That is, it takes into account what is on the left and right when given an input. For instance, in the case of a sentence, it considers the word to the left and right of each word. For example, in the sentence "I like Machine Learning," for the word "machine," it will look at the surrounding words and thus consider the words "like" and "learning." We all know the importance of the context in which words are found; the word "machine" associated with "learning" holds great significance.

In our BERT model, we used the "bert-base-multilingual-cased" library, which includes several languages including French. For this model, we had to tokenize our various sentences to feed them to our pre-trained BERT model and then transform them into numerical values to put them into tensors. This tokenization and transformation into tensors, which is a multi-dimensional vector, was done with our "SentenceDataset" class where we applied a very important notion called "padding." Padding allows us to have tensors of a fixed size, which is crucial. To have tensors of a fixed size, we simply add zeros after the real tensors, allowing us to disregard of the size of the sentence during evaluation.

Afterward, we can configure the training. As with LSTM and CNN, we used 265 and 128 as the sentence lengths. For models like BERT and Camembert, significant computing resources are required, which is why we had to use the GPU to run our code in a reasonable amount of time. Consequently, following the model's configuration, we can now train and then evaluate it to calculate its metrics. And like all models, we will use this model to predict our unlabeled sentences.

In terms of results, BERT and Camembert perform significantly better than the other models.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Regarding Camembert, it is very similar to BERT, with one of the main differences being that it is specifically designed for French and nothing else but French. As a result, it even outperforms BERT.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

**References :**

Tokuç, A. A., & Tokuç, A. A. (2023, May 12). *Splitting a Dataset into Train and Test Sets | Baeldung on Computer Science*. Baeldung on Computer Science. <https://www.baeldung.com/cs/train-test-datasets-ratio#:~:text=If%20we%20search%20the%20Internet,even%20a%2050%3A50%20split>.

CNN :

<https://www.analyticsvidhya.com/blog/2021/01/image-classification-using-convolutional-neural-networks-a-step-by-step-guide/>

BERT :

<https://huggingface.co/docs/transformers/model_doc/bert>

<https://huggingface.co/docs/transformers/index>

<https://huggingface.co/docs/transformers/pad_truncation>

Tensor :   
<https://machinelearningmastery.com/introduction-to-tensors-for-machine-learning/>