French language difficulty detection

Moka Express

Task description

Target:

Create a classifier that predicts the level of a French text, e.g. A1, A2, B1, B2, C1, C2

To train the classifier:

- 4800 labelled texts
- Balanced data set
- Base rate: 16.94%

To test the classifier:

- 1200 unlabelled texts

Apply the basic - some 40% accuracy!

Vectorizer: tfidf vectorizer using a standard spacy tokenizer and allowing some config tuning

Classifier: tune some hyperparameters depending on the classifier

- Logistic regression
- KNN
- Decision tree
- Random forest

Learnings

-> Simple classifier with some basic config can be powerful, e.g. logistic regression (46.56% accuracy)

-> Certain classifier is not ideal for this type of classification, e.g. decision tree (32.60% accuracy)

-> Ensemble of simple classifiers, e.g. soft voting or hard voting, certainly improves the result

Get advanced - above 50% accuracy!

Vectorizer:

- Word embedding using nlp sentence vectorizer
- Each text is vectorized to a 300 dimensional vector

Classifier: tune some hyperparameters depending on the classifier

- Linear SVC
- SVC

Learnings

-> Processing surprisingly fast

-> Few lines of code

-> Impressive result

Go for a challenge - 60% accuracy possible?

Hugging face pre trained models can make a difference!

Challenges

- Make the fine tuning work
- Make the prediction work

Try out different models

- Bert models
- FlauBert models
- Camembert models

Learnings

-> Too many nerds out there in this field to pre train all those models…

-> Only use those models is by far not enough. To score high, it requires some case specific creative thoughts on top

How to go even higher?

To get to 70% accuracy I will need at least one of the following things

- More training data -> not feasible
- More powerful hardware -> not feasible
- A really creative idea given the constraints of data amount and hardware capacity

-> Well, time is up, I am out of creativity. I made it to 60% accuracy, that’s the end!

Result summary

tfidf vectorizer				
	Logreg best config, 80% data	KNN tuned hp, 80% data	Decision tree, 80% data	Random forest, 80% data
Accuracy	**45.56%**	43.75%	32.60%	42.71%
Word embedding				
	Linear SVC, 80% data	SVC, 80% data	SVC tuned hp, 80% data	SVC tuned hp, all data
Accuracy	48.44%	48.54%	50.31%	**51.17%**
Hugging face pre trained models				
	flaubert, all data	camembert-base, 80% data	camembert-base, all data	camembert-base, more data
Accuracy	54.75%	56.67%	59.58%	**60.67%**