YELP Reviews :
Machine Learning for Natural Language Processing 2022

First Author
Hamza OULHAJ
hamza.oulhaj@ensae.fr

Abstract

We compare 2 different approaches to tackle multiple prediction tasks on reviews from the Yelp Dataset. The approaches differ by their text representation model which, in one case will be unsupervidely trained and common to all the tasks, and in the other case will be trained specifically for each task. The use of a single language model is an approach that has become very widespread due to the rise of high-performance models such as facebook's FastText (Bojanowski et al., 2017) or google's Bert (Hassan et al., 2019). We are looking to challenge and find the limits of this approach.

1 Problem Framing

In recent years, researchers in NLP have focused on the development of highly efficient language models that can best represent textual data.

In the context of information prediction from text, the approach consists in calculating embeddings using a pre-trained model that we call global model, and to use these embeddings to make the prediction. This way, a unique and common representation is used for the different prediction tasks.

The historical approach consists in training the embedding model and the prediction model simultaneously on the prediction task. Thus, each prediction task is associated with a distinct embedding model that we call local model and we will have as many representations as tasks.

Although the architectures of the local models are simpler than the architecture of the global model, the local approach is of linear space-time complexity in the number of tasks. So for a high number of tasks, the global model approach allows a consequent saving in time and storage space. Nevertheless, we can expect the global approach to be less efficient, since the representation is less adapted to each specific task.

I propose to compare these two approaches on the yelp review dataset. The datasets consist of user reviews on different businesses. I chose 3 different tasks of prediction on this review. The first task will be the prediction of the business category. It's a multi-label classification task as the business might be labelised as a restaurant and chinese food for instance. The second task is the prediction of the stars rate the user who wrote the review will give to the business. The user can rate the business on a scale from 1 to 5 stars. And finally, the third task is the prediction of the usefulness of the review for the other users. When a user shares a review, other users can read it and click on the useful button if they find it useful.

As these 3 tasks are by nature really different, they will provide us a view on how a global models manage to perform a great representation well suited for a wide range of tasks.

2 Experiments Protocol

The first step is to prepare our data. The yelp dataset is constituted of 5 different datasets with different info but only 2 are interesting for our problem. The business dataset regroups different datas as there is categories on more than 100 000 businesses. In the reviews dataset, we can find around 7 millions reviews with information such as the stars rating and the number of useful tag. The repartition of words (figure1) in the reviews is quite similar to the repartition we can get on non specialist datasets as wikipedia and it follows the zip law. I decided to sample around 1 million reviews for computing reasons. The business with less than 10 reviews are removed and we keep only the 140 most represented categories. We sample 5 reviews per business to get the final dataset with around 110 000 reviews.

For the 2 first tasks (category and stars rate),

the labels are quite explicit and provided in the dataset. The categories repartition is really balanced as some categories are strongly representated and others appear only a few times (figure2). The stars labels repartition is also balanced(figure3), and we can see that there is more good rate than bad rate. For the usefulness prediction, it's more difficult to provide a good measure of how much a review is useful. I decided to classify as useful any review that has an amount of useful tag superior to 2 thirds of the maximum of useful tag for a review on the same business. With this rule, we get around 8% of reviews considered as useful(figure4).

I will compare 2 different models on this 3 tasks. The Baseline Model is a local model with an embedding layer and a classifier of 3 fully connected layers with 50 neurons each. The global model used is a pretrained FastText, and we use for prediction a classifier with the same architecture as the Baseline Model.

3 Results

We trained the models on a train dataset that represents 76% of initial data. The training was on 5 epochs for all the models, and the learning rate was fixed to 0.001. All the results are produced from implementation you can find in my github repository [1].

Lets present the results of the models for each task :

Task 1: business categories

metrics	model	precision	recall	f1-score
micro	local	0.82	0.23	0.40
	global	0.83	0.15	0.25
macro	local	0.17	0.06	0.07
	global	0.06	0.01	0.01
weighted	local	0.47	0.27	0.30
	global	0.31	0.15	0.14

Data were strongly balanced so the model has learned to predict the most represented categories. We can see that in the detailed results in the notebook, and that explains why the micro and weighted metrics are greater than the macro metrics. Indeed, there is no mechanisms implemented here to weight the data and tackle this issue in the training. For this task, the local model is way better than the global one as he provides greater results for all the metrics. In fact, the global model cannot be used here and we should prefer a local model. To get better results, we should reduce the number of categories.

Task 2: stars rating

metrics	model	precision	recall	f1-score
macro	local	0.47	0.49	0.47
	global	0.37	0.39	0.37
weighted	local	0.59	0.53	0.55
	global	0.50	0.44	0.46

For this task, we have seen that the data were unbalanced but not as much as the previous task. That explains why the difference between macro and weighted are less significant than in the previous task. The models have a very good recall and precision for the two extreme rates 1 and 5. Indeed we can understand that it's hard to differentiate between 2 reviews that attribute the rate of 4 and 5. Here again, the local model is better as we gain around 10 points for all the metrics. However, for this task, the global model is good and provides acceptable results.

Task 3: usefulness

metrics	model	precision	recall	f1-score
macro	local	0.54	0.59	0.52
	global	0.54	0.62	0.44
weighted	local	0.85	0.71	0.77
	global	0.87	0.52	0.61

Finally for the last task, the two approaches give us quite good results and they give us the same precision. However when we look more precisely at the results, we see that the models struggle to predict the less represented class. The specific recall is high around 0.75, but the precision is low around 0.14. This problem should be linked to the weights in the training phase that give a huge weight to the less represented class.

4 Discussion/Conclusion

As predicted we can see that all the metrics are lower for the global model than the local one and the gap between the models can reach 20 points. For some tasks, the global model gives really bad results so he cannot be used. The gain of space and temporal complexity is penalyzed by a loss of preformances. What about a better global model like Bert?

[1] https://github.com/hamzaoulhaj/nlp-project.git

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Hebatallah A Mohamed Hassan, Giuseppe Sansonetti, Fabio Gasparetti, Alessandro Micarelli, and Joeran Beel. 2019. Bert, elmo, use and infersent sentence encoders: The panacea for research-paper recommendation? In *RecSys (Late-Breaking Results)*, pages 6–10.

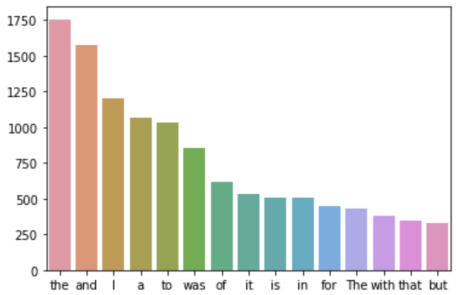

Figure 1: word repartition

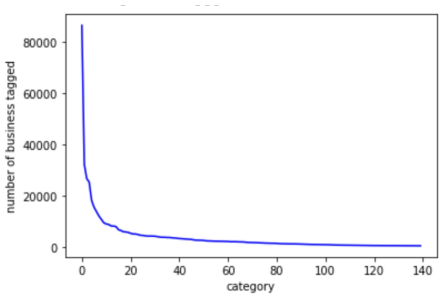

Figure 2: Categories repartition.

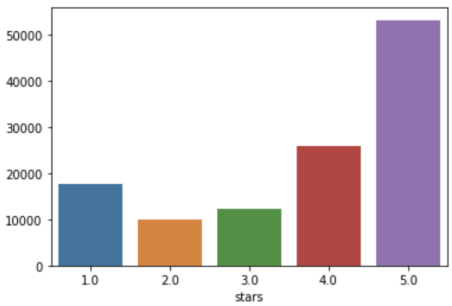

Figure 3: stars rating repartition.

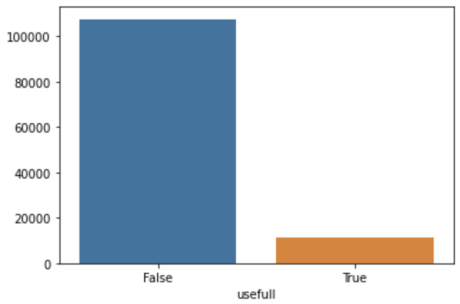

Figure 4: useful label repartition.