Review classification and conditional tip generation based on Yelp data
Machine Learning for Natural Language Processing 2020

Julien BONNET
ENSAE
julien.bonnet@ensae.fr

Laurent BROCHET
ENSAE
laurent.brochet@ensae.fr

Abstract

We train various NLP models for sequence classification and text generation on a sample of reviews from Yelp. We generate short texts ("tips") partially reflecting the characteristics we identified throughout the classification task[1].

1 Problem Framing

Yelp is one of the most popular web platforms for "crowd-sourced" evaluation of small businesses (restaurants, shops...). On Yelp, users rate businesses from 1 to 5 and write texts ("reviews") describing their experience as well as short sentences ("tips") providing one particular piece of information, a suggestion or trivia about the business.

Our main hypothesis is that the language of the reviews written by the users are somehow correlated with socio-economic parameters, and with the nature of the business reviewed. Based on this hypothesis, we will try to automatically generate short messages (tips) which would fit the characteristics (nature of the business, socio-economic characteristics, positivity or negativity) of a given review. In the end, we use the predictors calculated in the classification step to assess whether we can identify characteristics in the tips we generate.

2 Experiments Protocol

2.1 Exploring and sorting data

We use the Yelp data from the challenge Dataset, which contains information about businesses in a dozen American States and Canadian provinces. Socio-economic characteristics are extracted from US Internal Revenue Service data[2], then merged with the Yelp data using ZIP codes.

In order to keep a workable dataset, we restrict ourselves to only a few classes per variable to be predicted. In particular, we choose to keep only the four most frequent restaurant categories. Finally, we predict the following three variables[3]:

Categories, the "national" type of the restaurant in 4 classes: American (traditional), Chinese, Italian or Mexican

Rating, the evaluation of the restaurant, 3 classes: excellent (5 yelp stars), good (3 or 4 stars) or bad (1 or 2 stars)

Social, 3 classes: computed with the average revenue of the postcode area of the restaurant: rich, average or poor

2.2 Predicting reviews characteristics with SVMs and random forests

We try many different setups on a part of the dataset to determine the best non-neural predictor: with a pre-trained vectorization (Sklearn CountVectorizer) or with a vectorization learnt on an other part of the dataset (Gensim Word2Vec), with a different predictor for each of the three variables, with a single multilabel predictor, or with a monolabel predictor for each of the $4 \times 3 \times 3$ possible combinations of classes.

All those different predictors are compared with the same metrics: global accuracy, binary indicators for each of the 10 classes (precision, recall and F^1 score), confusion matrices.

We use a development set to tune some parameters with regard to these metrics, such as the number of iterations and the regularization parameter for the SVMs, or the number of trees for the random forests.

[1] see https://github.com/jb8794/NLP_ENSAE and https://colab.research.google.com/drive/1X5bVH81JxRYmmw9qRqIZQSW9Uz-1xxJD

[2] https://www.irs.gov/statistics and choose Individual Income Tax then Zip Code Data (SOI)

[3] Summary statistics and graphs about the dataset are provided in the Colab file.

2.3 Predicting reviews characteristics with neural predictors

We use Google's BERT to predict characteristics with neural networks. Starting from pre-trained weights, we fine tune the model to each specific classification task of the variables of interest.

2.4 Performing and evaluating tip generation

For this task, we try two different causal methods based on Hugging Face's transformers pre-trained token weights for text generation, *gpt2*.

The first one (beam search) is deterministic: at each step, a certain number of different possible next paths (beams) including multiple words are compared, and the most likely according to pre-trained weights is chosen. The second one (top-k sampling) is probabilistic: at each step, the k most likely next words are filtered and the next word is chosen according to a probability distribution where pre-trained masses are redistributed among only those k next words.

Both methods outputs are compared qualitatively (are the sentences coherent?), as well as quantitatively with BERT score, an indicator of similarity between sentences. Our aim is to have a corpus where sentences are not too similar between each other, but still reflect at least some of the characteristics defined above.

Then we generate a corpus of tips from a corpus of reviews of different types (categories, rating, social). In the end, we use the predictors of part 2 (trained on a different part of the whole dataset) to check whether the tips still reflect the characteristics of the reviews they were made from.

3 Results

3.1 Non-neural review classification

The global (accuracy) and binary indicators we computed show that most of the classes can be predicted in the majority (60 to 80 %) of cases, except for "good" (in variable ratings) and "rich" (in social). Perhaps those classes do not exhibit enough vocabulary specificity. Still, most of the time, diagonal terms dominate the confusion matrices.

Learning vectorization on the dataset seem to provide a small advantage in the prediction of categories and ratings (maybe limited by the size of the training set). We also note that SMVs and random forest tend to overpredict larger classes, such as "American" ; we add the option *class weight = balanced* in the models to address this issue.

3.2 Neural review classification

Neural networks have very good results: around 80 % accuracy for ratings and categories predictions. For social conditions (60 % global accuracy), prediction of the "rich" class is still difficult, maybe because there are too few reviews in this category (and not distinctive enough).

3.3 Tip generation

In the end, we chose top-k sampling to generate our tips, since the results of beam search were very repetitive. Human subjective analysis shows that almost all the sequences are syntactically and grammatically correct, and that most of them make sense when talking about a restaurant. Running a previously trained SVM multilabel predictor on the tips, we found that most of the tips exhibit at least some characteristics of the reviews they were generated from. Anyway, having only moderate characteristic accuracy is not a problem: in a "natural" environment, the tips not are always specific of the context in which they appear.

4 Discussion/Conclusion

We managed to show that the reviews on Yelp exhibit characteristics related to the businesses their own characteristics are as well as the social conditions of their neighborhood (to a moderate extent).

We observe that the performances of our models are limited by RAM availability in Google Colab: they were still improving as we were increasing the size of the train set up to the memory limit.

On another note, we could replicate our method on a more general scale, browsing through texts on websites to add short comments (or tweets) from a previously generated database - these comments being coherent with the characteristics of the text found on the websites as well as with the positivity (or negativity) we would want to convey.

References

Chris McCormick. BERT fine-tuning tutorial with PyTorch [online].

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert.

Thomas Wolf et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.