Positive or negative ? Classification of text from different contexts

Marie BRUN
ENSAE
ma.brun@ensae.fr

Abstract

Training a model typically involves using data that is similar to the context in which it will be tested. This is a logical and common practice, as text within a similar context tends to have similar lexicons. However, it is uncertain whether using similar sources is crucial to obtaining good results. This document aims to explore this question. Categorizing data is a time-consuming task, and the ability to limit it would be beneficial. If it were possible to mix sources, it could save a significant amount of time. In this study, we will compare data from Amazon, which has a diverse range of writers, with more homogeneous data such as poems, to determine if it could aid in automatic classification.

1 Introduction

In modern scientific papers on classification, reliance on a single dataset, such as the Persuasive Opinion Multimedia corpus used by [1], is the norm. In cases where multiple datasets are employed, it is typically to test the efficacy of the model, as demonstrated by [2] who used Amazon and Yelp data to this end. However, such dataframes often derive from a single source, such as review texts, resulting in a limited range of topics and vocabulary.

The process of training a model for classification necessitates the labeling of data, without which the computer cannot determine whether a sentence is positive or negative, for example. If it were possible to mix data sources, it would significantly reduce the time required for training. This is especially true for binary classification tasks such as positive or negative sentiment analysis, which is the focus of this study.

Platforms like Amazon, which host user reviews, are highly valuable data sources for classifying text of this nature. They offer a diverse range of user profiles and a global rating system, enabling a more comprehensive analysis of the data. For instance, a negative comment on a five-star rating is highly unlikely, making it evident that five-star reviews are positive, while one-star reviews are not.

2 Datasets

Now that you know the reason of this project, I will present you the chosen data-sets.
To train a model, you cannot randomly choose a data-set. Some logical point has to be think about to explain why those particular choice.

First, amazon data choice :

"Amazon us review" [1] contains several product categories. For this study, I choose to consider a book one. I did not took several category for time consideration.
Commonly, you can imagine, people who use to read might have much more vocabulary since it is a common say. Plus, thanks to the diversity of books (Comics, classics, fiction...) you will have a diversity of people who will let reviews, old and young, man and woman. And all those profile could impact words choice. You can imagine a reader of Shakespeare can use more uncommon expression than someone who buy picture books when he comments what he bought. Plus, they will not expect the same things for their books, increasing even more diversity of vocabulary.
In this data-set, we will have many feedback from customers that are already rated (with stars). Plus, since it comes from amazon, there are extra judgements comments not about the product but on the delivery, impacting even more the diversity

[1] https://huggingface.co/datasets/amazon$_{u}s_{r}eviews/viewer/Books_{v}1_{0}2/train$

of the used vocabulary.
In this data-set, we will focus on star rating category (to be our labels) and on the review body to be the text that will allow to classify our data.

Second, poem data choice :

The second chosen data-set is a bit more original and really different from the first one. It is about poems sentiment[2] and it has already been classified.
In the poem context, you can imagine that to get a poetic text, some original vocabulary can appear for a rhythm or lexical matter. It is a complete opposite to amazon, here the goal is beautyfulness of sentences.
Verses are already classified as positive, negative or neutral. Since on amazon finding neutral would be difficult (would it be more three stars ? two ? we cannot know), we will only focus on label linked to positive : 1 (equivalent to five stars on amazon) or negative : 0 (equivalent to one star on amazon).

An interesting question is will it match ? Can the diversity of amazon writer help to detect emotions found in a poem ? Balancing with two data sets of the same origin will more or less obviously lead to easy classification. What about when you mix sources ? Will those data-sets be too different to get any results ? That is what this paper search to look at.

3 Experiments Protocol

Once labelled data-set chosen, a model had to be selected to classify data. The first work has been to find an algorithm. After reading few recommended papers, it appeared BERT [2] could be helpful in this mission.
Several tests where done to find a good algorithm, several did not worked (library that did not work or algorithm not adapted to our data). In the end it has been chosen using Chat GPT because no other strategy were working.[3].

The algorithm[4], the methodology :

- **First :** Like for every algorithm, the first step was to import useful libraries. In our case, it is important to think about 'dataset' one. It allows you to import a various number of data-sets. Then you also have 'transformers' or 'torch' for the NLP and Machine Learning part.
  Finally, the function Counter from 'collection' can be a real plus to get the amount of labelled data in each category.

- **Second :** The data-sets has been load. After an import, you have to look at your data. Amazon data were grouped according to the table above :

5-stars	4-stars	3-stars	2-stars	1-star
1864807	586182	249926	166384	238221

Figure 1 : Amazon label groups

As you can see maximum and minimum have a lot of data, they should be enough for this study. You can notice there are much more 5-stars but you can imagine texts are shorter since people usually write longer message to complain.

- **Third :** Once the import has been done, treatments of the data can start. The first step of this part has been to remove middle columns that could have bias our results (column from two to four stars in the table).
  Then, imported data are turn to dataframe, an object that is easier to exploit.
  A new treatment is done to homogenize labels. Now a five stars rate will be a 1 and the one star a 0. This choice is made to make easier the analyse of the results and to have same labels in both tables.
  Then, we take as a list all important columns of both dataframes. It can be an interesting point at that level to delete amazon data-set, this is really huge so memory consuming. Now we have four lists : amazon_classes (star rating), amazon_text (body review), poem_classe (labels) and poem_text (verse text).

- **Fourth :** At that moment, we can start a more important step : the creation of the class for

[2] https://huggingface.co/datasets/poem$_sentiment$
[3] https://openai.com/blog/chatgpt
[4] https://colab.research.google.com/drive/1-1nQ3LCefkyLDgawsDLqwk9SMEeavDCU?usp=sharing

classification. This class will be useful to classify our data.
Now data can pass threw this class to link text to labels. Amazon data are now easy to train.

- **Fifth :** Before getting the results, it is important to check your device threw torch. Depending you are on cpu or not the algorithm will not work. So a little function is here to control if you are working on cpu or not.

- **Sixth :** Once the control is done, the last function is created : "predict". It will predict for each sentence if it seems to be more 0 (negative) or 1 (positive) and return a list of probability for each option.
To calculate this probability, the softmax is used (see under). Exponential sum will be we calculus of all previous token, the one from training set and the one above the text one.

$$\sigma(y_i) = \left(\frac{e^{y_i}}{\sum_j e^{y_j}} \right) \tag{1}$$

- **Seventh :** Last step, getting result response. Three different variable will be created : one to get how many exact value did the algorithm get, one for false 1 (labelled one but zero) and one for false 0.
Then we test the whole poem list. Three possible loop options :

  - in the predicted list, the position of the value corresponding to the label is the maximum. The prediction is exact, the more suspected issue is the real label.

  -the first issue is not satisfied and the label is 0.

  -the first issue is not satisfied and the label is 1.

Now that the algorithm has been explained, what about the result ? How did the tests concludes ? Is it realistic mixing those two universes to label text or are there really too different ?

4 Results

The experiment failed. The classifier does not work on the data. Most time it classifies all data in an only category.
The first time the full algorithm was launch, all data appear as 0, as negative sentences. Moreover, the border was quite clear, the prediction announced it was sure at 70% for every data they had to be negative. The closest result to 0.5 was 0.43.
For the second test results where similar about the borders. But this time data where all classified as 1, positive data. You can have a look at the result under on Figure 2. As you can see there is just one label present : label 1 that is represented by dark blue color.

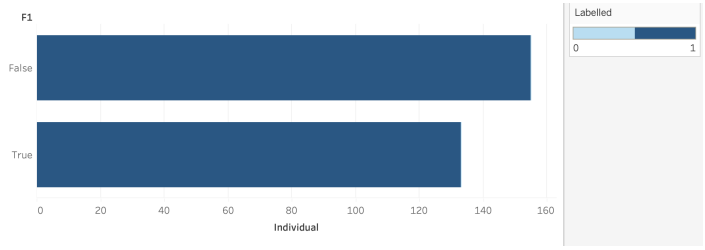

Figure 2 : Trustfulness of the result, colored depending on the true label

The third test has been a bit different about the result, still too many errors but at least both categories appeared. This time, we more or less came back to first classification with labels mainly class as one but a little get out of the row.

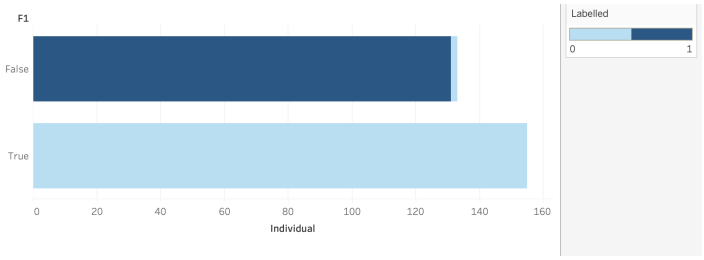

Figure 3 : Trustfulness of the result, colored depending on the true label

Few other test has been done, changing the batch size. When the batch size decrease, success rate still does not exceed 60% but you had a bit more false in all groups. If it was like Figure 4 first line at first, then it was like in second line. You have more than two data in each group.

Batching	Exact	False 1	False 0
16	155	133	0
8	133	10	145

Figure 4 : Value position

5 Discussion/Conclusion

The diversity of results suggests that the classification approach used did not yield promising outcomes in distinguishing between positive and negative poems based on training data from online shopping comments. Despite this, several potential avenues for improvement exist.

Firstly, it may have been beneficial to use a larger training dataset to expand the vocabulary available for analysis. Additionally, incorporating a filter to exclude comments that are too similar, such as those containing the word "perfect," could help to ensure greater diversity in the training data.

Another potential approach to improve the classification performance could be to explore the use of a different BERT algorithm. It is possible that the chosen algorithm may not have been the most effective for this particular classification task.

Lastly, it could be worth exploring the use of less exotic test datasets. Poems may be too distinct from other forms of literature to be easily associated with any other type of text. Using test datasets that are more closely aligned with the language and structure of poetry could lead to better classification results.

References

[1] Alexandre Garcia*, Pierre Colombo*, Slim Essid, Florence d'Alché Buc, and Chloé Clavel. From the token to the review: A hierarchical multimodal approach to opinion mining. *EMNLP 2019*, 2019.

[2] Hamid Jalalzai*, Pierre Colombo*, Chloé Clavel, Éric Gaussier, Giovanna Varni, Emmanuel Vignon, and Anne Sabourin. Heavy-tailed representations, text polarity classification & data augmentation. *NeurIPS 2020*, 2020.