Reviews' Sentiment Analysis Comparison

Aviv Pelach (ID. 208387407), Gil Knafo (ID. 318590593), Shoham Danino (ID. 204287635)

Submitted as final project report for the NLP course, IDC, 2021

1 Introduction

Sentiment Analysis (SA) is one of the cores and well-researched areas in NLP. It is a research area with rapid and constant growth, mostly because of online social media sites and services in the digital age, and the increasing amount of textual data (such as statuses, comments, reviews etc) available in them. Because of that, the need of automatic sentiment analysis is on the rise.
The goal of sentiment analysis is to identify and classify texts according to the opinion, sentiment or emotion that they convey (especially positive or negative). It requires a good knowledge of the sentiment properties of each word. This properties can be either discovered automatically by machine learning algorithms, or embedded into language resources.

Although we did not delve into this topic in the course, in one of the lectures we talked about a project done in a previous course where a group trained a model on corpus in English and then examined its accuracy on corpora in other languages, and thus examined the similarities between different languages.
We really liked this idea and thought of doing something similar and testing the similarities between different types of reviews, in terms of sentiment analysis. for example, training a model on hotel reviews, and next, test the accuracy of the model on reviews of restaurants, weather, airlines, etc.

We find this research relevant for use-cases in which we have a rich labeled data from one domain of reviews (and also other corpora), however we have very few labeled examples in other domain, which is the domain the interests us. Our goal was to show how one can achieve better performance when training on rich domains rather than just using the data in the limited domain.

1.1 Related Works

Sentiment analysis is extensively researched over the last years, researches that aim to find more accurate and fast models and to tune models to specific datasets. There are also a lot of researches specifically about reviews' sentiment analysis, which is our focus in this project. Most researches focus on

one dataset and how to predict best on it: Yelp restaurants reviews[2], app reviews[4], scientific reviews[1] etc. But, we haven't found articles referring to a comparison between reviews' sentiment analysis.
However, we found articles that compare languages' sentiment analysis[3][5]. In those articles models are trained on English or other languages corpus and tested on other languages corpus that the model were not trained by them at the first place.
In our project we will combine those two areas of sentiment analysis, and compare reviews' sentiment analysis.

2 Solution

2.1 General approach

In this work we want to measure the importance of adapting a model to a specific data and whether there can be general models in sentiment analysis questions. We chose to focus on sentiment analysis of reviews, which can be found in many different topics (hotels, movies, weather etc), and wanted to determine the similarities between different reviews of different topics.
In order to do that, We selected datasets of reviews of six different topics. Each time, we trained our model on one dataset and evaluated it (using F1 score) on the other reviews' datasets.
In this way we examined the similarities between the different topics of reviews and tried to conclude how important is to train sentiment analysis models on specific datasets, in order to get good performance.

After running our experiments we wanted to examine our results and check whether they match to real-life similarities. In order to do that we used word2vec embedding for each dataset (with 2 different techniques, more details in section 3.2) and visualized it.

2.2 Datasets

We used 8 reviews datasets of 6 different reviews' topics:

1. Movies (Large Movie Review Dataset):
   Binary sentiment classification containing substantially more data than previous benchmark datasets (25,000 highly polar movie reviews for training, and 25,000 for testing). The dataset is here.

2. Airlines (Twitter US Airline Sentiment):
   Twitter data was scraped from February 2015 and contributors were asked to first classify positive, negative, and neutral tweets, followed by categorizing negative reasons). The dataset is here.

3. Weather (Weather sentiment):
   Contributors were asked to grade the sentiment of a particular tweet re-

lating to the weather. 20 contributors graded each tweet. The dataset is here.

4. Hotels (Datafiniti Hotel Reviews and Trip Advisor Hotel Reviews):
   The first includes hotels' reviews provided by Datafiniti's Business Database, and has 2 sub datasets that we used. The dataset is here. The second contains 20k reviews crawled from Tripadvisor. The dataset is here.
   We've merged those datasets into one dataset that contains hotels' reviews.

5. Restaurants (Restaurant Customer Reviews):
   The dataset consists of 1000 reviews classify as positive (1) or negative (0). The dataset is here.

6. Ebay (Ebay reviews):
   The dataset was downloaded from ebay and contains textual reviews for ebay's products and a score. The dataset is here.

2.3 Data preprocessing

Before starting our project, we had to make sure that the datasets are similar to each other. So, we've converted all the reviews to a uniform format with 2 columns: review: the review text and sentiment: positive (1) or negative (0). Each dataset required a different preprocess for the 'sentiment' column, depends on the original labels it had. For more details, see the project's notebook.
Then, we've removed from all the 'review' columns text all tags, hyperlinks, non relevant characters and converted all higher case letters to lower case.
Eventually, we made sure that all datasets are balanced (50% of the rows are negative and 50% are positive), and removed non-english reviews.

2.4 Design

To perform our experiments we used the Google Colab platform, using Python to write our code. Our choice was made due to the ability of efficient GPU-based implementation, although we found out during the project that Google's GPU resources are too limited for us and we had to adjust ourselves to it. Within the preprocess we used langdetect and regex; For the model we used Huggingface BERT with pytorch and to evaluated it we used sklearn; We also used Gensim for word2vec (to check similarities) and NLTK in order to remove stop words.

After reading about state of the art models for sentiment analysis[1] we decided to use BERT for our experiments. Our baseline for all models was pre-trained BERT (bert-base-uncased version). On top of it we added a feed-forward classification head consists of linear dense layer, non-linearity function ReLU and another linear layer with two outputs for our binary classification. We used AdamW optimizer with 5e-5 learning rate and cross-entropy loss.

[1] http://nlpprogress.com/english/sentiment_analysis.html

One of the main difficulties in the project, that actually characterizes the whole domain of Natural Language Processing is running time. Although the datasets aren't huge, training just a few epochs takes many minutes and training and evaluating cross-datasets took us a few hours. In addition, and as stated above, Google's GPU resources are limited, and we had to come up with creative solutions in order to implement our project properly.

Another difficulty, but in our research field was datasets similarity. It is a challenge to understand how similar two (or more) datasets are, as we wanted to understand in order to evaluate our model. We have not managed to find great literature about this topic, therefore we managed to come with our own two approaches, which will be described in section 3.
Dealing with stopwords turned out to be a pretty simple challenge thanks to the NLTK package provided in python.

3 Experimental results

3.1 Base experiments

We've trained the model on one dataset each time and then tested it on the other datasets (see figure 1), and evaluated the performance with several metrics, especially on F1 score which averages precision and recall harmonically.

train/test	*airlines*	*ebay*	*hotels*	*weather*	*movies*	*restaurants*	*average*
airlines	–	0.87	0.9	0.23	0.81	0.91	0.74
ebay	0.88	–	0.88	0.18	0.85	0.91	0.74
hotels	0.83	0.85	–	0.23	0.74	0.83	0.7
weather	0.17	0.16	0.15	–	0.18	0.21	0.17
movies	0.85	0.87	0.88	0.28	–	0.89	0.75
restaurants	0.87	0.84	0.83	0.18	0.63	–	0.67
average	0.72	0.72	0.73	0.22	0.64	0.75	

Table 1: F1 scores

The performance we obtained from those experiments is presented above in table 1. It is clear to observe that training on specific dataset leads to great performance on others (for example: airlines and hotels), however, it is not the case in all datasets (especially the weather reviews) and combinations (for example: movies and restaurants or hotels).

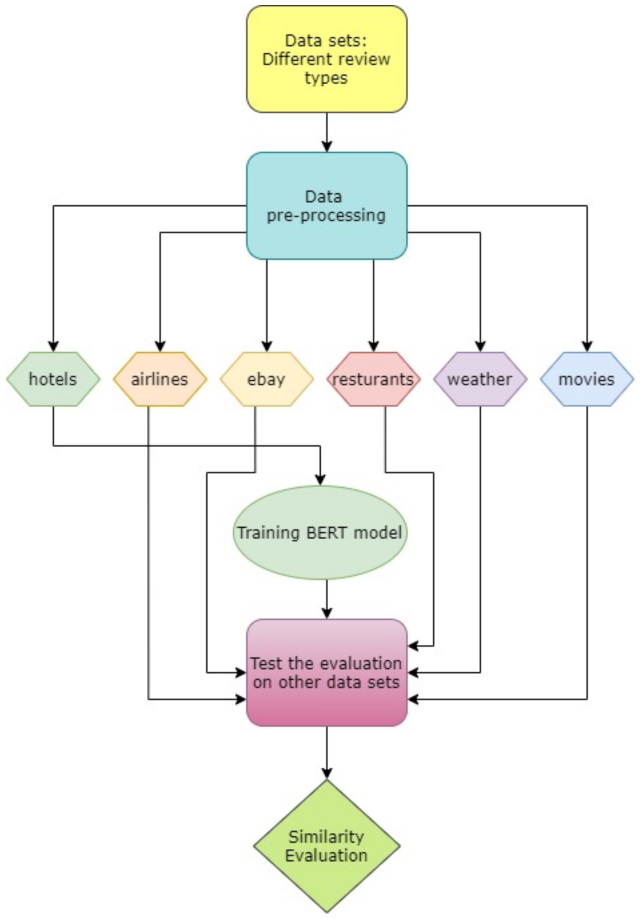

Figure 1: Example of training and evaluation (for hotels)

3.2 Datasets similarity

In order to tackle the observations presented above and get a better understanding of it we developed two techniques for evaluating datasets-similarty. Both approaches are based on word2vec embeddings which were achieved by training on the Google News corpus. The approaches are differ in the way we use the embeddings of each dataset. It should be noted that before applying both technique we've removed stop words from all datasets using NLTK.

1. The first technique takes all the words from every dataset and filter out words that appear in more than three datasets (Inverse-Document-Frequency driven approach) and then sample randomly 80 words from each dataset. based on these samples we generated a 2D-visualization after applying Latent Dirichlet Allocation technique for dimensionality reduction.

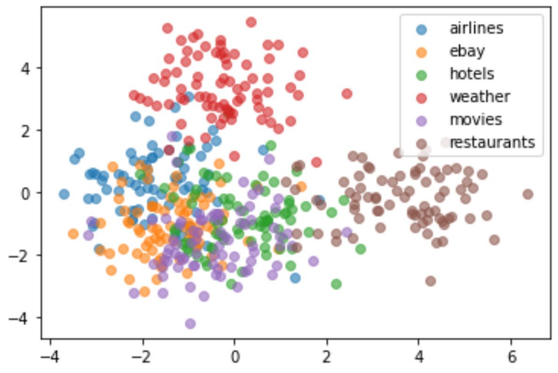

Figure 2: datasets similarity - first technique

2. The second technique takes all the words from every data set, and choose the top-200 words that appeared the most in every data set (Term-Frequency driven approach). based on these words' embeddings we generated a 2D-visualization using the same dimensionality reduction technique described above.

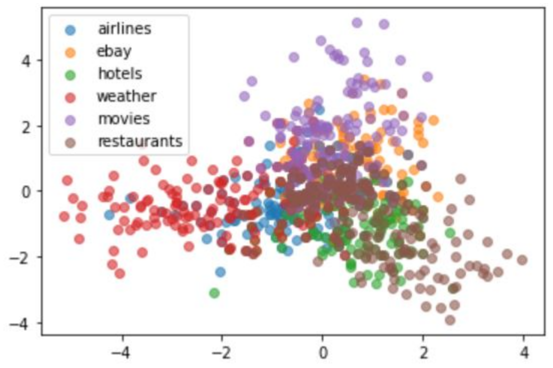

Figure 3: datasets similarity - second technique

By using these approaches we were trying to explain the results we got in the first part of the experiment, and we got some really nice insights:

1. The embedding of the weather reviews is really relatively far from the central cluster, comparing to other datasets, which explains why it's the ”weakest” dataset.

2. The embedding of the restaurants reviews is the central cluster in the second technique (with some outlayers), in accordance to its highest F1 average. However, in the first technique it's far from the center.

3. The embedding clusters of ebay and movies are pretty similar, which explains why training on one leads to high F1 score on the other.

4. With both techniques, the embedding clusters of airlines and ebay are very similar, which explains the similar F1 scores they got.

3.3 "Weak" datasets

In addition to the experiments described above, we tried to tackle the datasets which had not have good performance when training on other data set (the weather dataset in our case). We decided to implement a simulation in which we train our model with the train dataset (from other domain) and also, add a small portion of labeled examples to the train set from the dataset we wanted to get predictions on its examples. This scenario is relevant for domains in which we do not have much of labeled data, but we want to take advantage of the few labeled examples we do have in addition to the labeled data from other domains.

airlines	*ebay*	*hotels*	*movies*	*restaurants*
0.39	0.45	0.36	0.46	0.31

Table 2: F1 scores for the "weakest" dataset

We can see that the performance is still not good enough, especially compared to the rest of the F1 scores we've presented. However, improvement can be seen (compared to those presented in table 1), and this method can probably be developed and improved.

4 Discussion

In our project, we proved that training a sentiment analysis model on one domain leads to pretty good results on other domains. This, of course, affected by the similarity between the domains in "real life", as can be examined with the word2vec embedding techniques we offered. It should be noted, that we haven't found great literature about corpora similarity, and developing more accurate tests may be interesting for future work.

One of the thing we wanted to examine in this project is whether "real life" similarities are also reflected in the performance of our model. As expected, the F1 score was high for example for hotels and airlines, which are from the same field in real life. But, we were surprised to see that also the restaurants dataset, for example, gets great performances, although it isn't really in the same field as other datasets.

4.1 Future Work

As noted above, developing more accurate tests for corpora (textual datasets) similarities may be interesting for future work. Both in the field of word2vec embedding and through other methods.
In our project we used BERT model, in the future, it can be interesting to examine different models, that can be more accurate on specific datasets.
The ”few shot learning” method we used in order to improve the performance on the ”weak” dataset seems to be helpful, but not enough. This also can be an interesting subject for future work.

5 Code

Project’s Colab notebook link is here.
The notebbok includes the preprocess and the links to the datasets, the main experiments and the BERT model, the similarities tests and the ”weak” datasets (we run it in different notebook because of time complexity issues).

References

[1] Souvic Chakraborty, Pawan Goyal, and Animesh Mukherjee. Aspect-based sentiment analysis of scientific reviews, 06 2020.

[2] Eftekhar Hossain, Omar Sharif, Moshiul Hoque, and Iqbal Sarker. Sentilstm: A deep learning approach for sentiment analysis of restaurant reviews, 11 2020.

[3] Gati Martin, Medard Mswahili, and Young-Seob Jeong. Sentiment classification in swahili language using multilingual bert, 04 2021.

[4] Sakshi Ranjan and Subhankar Mishra. Comparative sentiment analysis of app reviews, 06 2020.

[5] Ali Septiandri and Arie Sutiono. Aspect and opinion term extraction for aspect based sentiment analysis of hotel reviews using transfer learning, 09 2019.