Cross-domain Sentiment Analysis

Niccolò Benedetto MAT. 7024656

Supervisor: Paolo Frasconi

1 Development Environment

The project was developed using the cloud service *Google Colab*, in its standard version, thus with limited access to the computing resources provided directly by Google (GPUs, RAM). To run the code in the Python language (in its latest version, Python 3), Jupyter Notebooks are used, an open-source web application that allows the creation and sharing of documents containing live code and various other multimedia resources.

2 Analysis and Implementation

The reference dataset, on which the model will operate, is presented in the following form:

	Unnamed: 0	drugName	condition	review	rating	date	usefulCount
0	206461	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9.0	May 20, 2012	27
1	95260	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8.0	April 27, 2010	192
2	92703	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5.0	December 14, 2009	17
3	138000	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8.0	November 3, 2015	10
4	35696	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9.0	November 27, 2016	37

Figure 1: original dataset

The ultimate goal is to perform cross-domain sentiment analysis with reference to the *review* column of the dataset above. By cross-domain sentiment analysis, we mean analyzing how the classifier reacts when it is trained and tested on reviews belonging to different domains. In our case, the referenced domains are the conditions that appear most frequently in the dataset: *Anxiety*, *Birth Control*, *Depression*, *Diabetes Type 2* and *Pain*. It was also decided to train the model not only on the raw data from the *review* column, but also on a new column added to the dataset, called *cleaned_review*, which contains the data from the original column processed after a text pre-processing phase. This phase was carried out using the suite of libraries and programs for symbolic and statistical analysis in the field of natural language processing (mainly in English), also known as NLTK. In order to prepare the text for a 'clean' analysis phase, the pre-processing used involves several steps including the removal of non-alphanumeric characters, removal of stop-words, removal of HTML tags, tokenization, and conversion to lowercase. In **Figure 2** a row of the dataset is shown with the addition of the *cleaned_review* column.

The texts of the reviews will therefore represent the content of the model's variables. The label values, on the other hand, are generated following an analysis of the texts performed through the tool VADER, a tool included in the NLTK library, which utilizes a vast dictionary of words labeled with sentiment

review	rating	date	usefulCount	cleaned_review
"It has no side effect, I take it in combinati...	9.0	May 20, 2012	27	side effect take combination bystolic 5 mg fis...
"My son is halfway through his fourth week of ...	8.0	April 27, 2010	192	son halfway fourth week intuniv became concern...
"I used to take another oral contraceptive, wh...	5.0	December 14, 2009	17	used take another oral contraceptive 21 pill c...
"This is my first time using any form of birth...	8.0	November 3, 2015	10	first time using form birth control 039 glad w...
"Suboxone has completely turned my life around...	9.0	November 27, 2016	37	suboxone completely turned life around feel he...

Figure 2: dataset after pre-processing phase on *review* column

scores. VADER evaluates the polarity of a text, i.e. performs a sentiment analysis on it, generating a dictionary with 4 keys:

- **negative key**, indicates the proportion of negativity expressed by the text.
- **positive key**, indicates the proportion of positivity expressed by the text.
- **neutral key**, indicates the proportion of neutrality expressed by the text.
- **compound key**, which is a composite value generated by combining the three values above.

The first three values of this dictionary range between 0 and 1, where 0 indicates the absence of the corresponding sentiment and 1 its maximum presence. The fourth value, however, varies in the range [-1, 1], where values > 0 indicate a positive sentiment, < 0 a negative sentiment, and values close to 0 indicate neutrality or the absence of a clear positive or negative sentiment. It was decided, in relation to the meaning of the compound key values defined by the VADER analysis, to express the sentiment of a review through the following intervals: if the value of compound key falls within the interval [-1; -0.3] then the corresponding label value is 0 (corresponding to a negative sentiment), if it falls in (-0.3; 0.3) the label value is 1 (neutral sentiment), while if it is contained in [0.3; 1], the label takes the value 2 (positive sentiment).

review	rating	date	usefulCount	cleaned_review	cleaned_review-dict-VD	cleaned_review-numberlist-VD	standard_review-dict-VD	standard_review-numberlist-VD
"It has no side effect, I take it in combinati...	9.0	May 20, 2012	27	side effect take combination bystolic 5 mg fis...	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	(0.0, 1.0, 0.0, 0.0)	{'neg': 0.121, 'neu': 0.879, 'pos': 0.0, 'comp...	(0.121, 0.879, 0.0, -0.296)

Figure 3: dataset after VADER analysis

A new column is then added to the dataset, called *rating_model*, which maps the values of compound key based on the logic described above.

review	rating	date	usefulCount	cleaned_review	cleaned_review-dict-VD	cleaned_review-numberlist-VD	key_neg	key_neu	key_pos	key_compound	rating_model
"It has no side effect, I take it in combinati...	9.0	May 20, 2012	27	side effect take combination bystolic 5 mg fis...	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	(0.0, 1.0, 0.0, 0.0)	0.0	1.0	0.0	0.0	1

Figure 4: adding column *rating-model*

The dataset, after being manipulated to contain all the necessary information to train the model, is then split with a rate test_size = 0.33 (i.e. 67% of the data is used for training and the remaining 33% for testing). The model has 4 variables: X_train (training reviews), X_test (test reviews), y_train (training labels, which correspond to the values of the new *rating_model* column), and y_test (test labels). The logic behind extracting lexical features from the review texts, which must be provided as input to the Perceptron algorithm, relies on the working principle of the TfidfVectorizer class from the scikit-learn module, which extracts features from a text to convert them into a vector representation based on the frequency of words and their importance in the overall corpus. Thus, the words in the reviews are considered as units of analysis; more specifically, unigrams (individual words) and bi-grams (pairs of adjacent words) are evaluated, unlike the work described in Gräßer et al. 2018, which also considers tri-grams. This choice was forced due to the limited resources offered by the standard version of *Google Colab*.

3 Study of the Results

The cross-domain analysis then requires us to iteratively use Perceptron to train and test the model created on reviews belonging to portions of the original dataset, obtained by considering all possible combinations resulting from pairing domains.

```
[PERCEPTRON ON CLEAN DF]
------------------------------------------------------------
['X_BC_trainCl', 'y_BC_trainCl', 'X_BC_testCl', 'y_BC_testCl']
------------------------------------------------------------
Accuracy of Perceptron is 0.875381538785391
------------------------------------------------------------
['X_D_trainCl', 'y_D_trainCl', 'X_BC_testCl', 'y_BC_testCl']
------------------------------------------------------------
Accuracy of Perceptron is 0.6879275865698348
------------------------------------------------------------
```

Figure 5: partial output of Perceptron

For example, the first accuracy value in **Figure 5** indicates the result of running Perceptron when the model is trained on reviews belonging to the *Birth Control* domain and tested on labels also belonging to the same domain. The second accuracy value refers to the predictive result of Perceptron when the model is

trained on reviews belonging to the *Depression* domain and tested on labels belonging to the *Birth Control* domain.

test train	Anxiety	Birth Control	Depression	Diabetes, Type 2	Pain
Anxiety	0.800924	0.676981	0.710992	0.659549	0.678501
Birth Control	0.731657	0.875382	0.753091	0.696323	0.700690
Depression	0.707542	0.687928	0.850317	0.666667	0.689349
Diabetes, Type 2	0.612622	0.638354	0.650184	0.792408	0.646450
Pain	0.636737	0.652247	0.655530	0.625148	0.813116

Figure 6: cross-domain analysis on clean dataset

test train	Anxiety	Birth Control	Depression	Diabetes, Type 2	Pain
Anxiety	0.765008	0.622566	0.649516	0.604982	0.592209
Birth Control	0.651616	0.856962	0.678249	0.655991	0.659763
Depression	0.659826	0.652563	0.821918	0.601423	0.631164
Diabetes, Type 2	0.574654	0.612988	0.586702	0.744958	0.610454
Pain	0.598769	0.607515	0.634815	0.589561	0.784024

Figure 7: cross-domain analysis on dataset without text pre-processing

Thus, the table in **Figure 6** collects the accuracy values of the algorithm when the model is trained on reviews from a specific domain (the rows of the table) and tested on labels of reviews from another specific domain (the columns of the table). It can be observed that the numerical prediction values are considerably high when the model is trained and tested on the same domains, which is known as in-domain analysis (for example, consider the value 0.875382 obtained for *Birth Control*), while the accuracy can drop to 0.612622 (Perceptron correctly predicts with a probability of approximately 60%) when training on the *Diabetes, Type 2* domain and testing on *Anxiety*. The numbers also suggest that there is a high correlation between the *Birth Control* and *Anxiety* domains, meaning that those who wrote reviews referring, for example, to contraceptive methods are likely to also mention feelings of anxiety, as well as between *Depression* and *Anxiety*. It should be noted that these results are closely linked to the choices made during the project's analysis phase. Consider that the range of intervals used to define the model's labels is consistent with the meaning of

the compound value generated by the VADER tool after evaluating the text's polarity, but it is most likely not optimized to achieve the best predictive accuracy. Manipulating them differently will undoubtedly lead to a different set of results.
Finally, note the cross-domain analysis obtained on the dataset in the absence of the text pre-processing phase (**Figure 7**). While the trend of the values and the relationships between domains mirror those in **Figure 6**, the predictive accuracy of the algorithm appears less precise.