Technology Review

Title:

Sentiment Analysis Tools & Techniques

Inspiration:

Technology revolution in past few years have given us loads of information that is readily available and easily accessible on the web. This includes sharing our views/perspective with others. The view that we have about a product/business/entity could be the result of our personal experience or it could be inspired from other's perspective about the same entity. Let's say we want to buy a product then we can leverage the experience of others who might have experience with the product. This might help us discover the popularity of that product. To acquire that information, we need to understand what others feel about a particular entity. As we explained with loads of information available and not enough time to go through every piece of information, we want to get to the general sentiment about the product to make our decision. This thought inspired us to work on a problem of looking at reviews of different users for a business and then generate a popularity score for the business. Based on the sentiment score generated we can show the user whether this business is recommended, or not recommended by other users.

Background:

Technology review will enable us to select the right set of tools for our project. We are working on a project "Business Search and Recommender" with restaurants data from yelp. The core functionality of the project is facilitating the user with search query results and recommend the nearby restaurants. To achieve the recommendation, we need to have good understanding about the entity/business and general understanding about the user, who provide their perspective about the entity. We worked on the understanding the features of the business based on the reviews provided by the user as this is the rich source of information about user's view related to the business. This review information along with other attributes in yelp data like rating, categories can help us to guide the user with good suggestions/recommendations in the similar category. We have confined scope of this project to be the Restaurants category.

Goal:

Since we will be working very closely on review sentiment analysis, we would like to concentrate our technology review around the same topic. We explored different sentiment analysis tools/techniques available. This exploration led to a comparison between the top few approaches for sentiment analysis and why would we choose one over the other/ how we can improve the existing technologies to achieve more. The top few sentiment analysis libraries that we worked with are mentioned in the section below.

Review Analysis:

https://www.kaggle.com/yelp-dataset/yelp-dataset

For this discussion of Sentiment analysis, we will use the Yelp's business dataset. This dataset includes core attributes for the businesses like name, address, categories etc. This dataset includes reviews of the users for a business, reviews are one of the most important attributes that contains the sentiment of the users about the business. To understand the review sentiment analysis, we have selected few sample reviews from the Yelp data with user ratings is shown in the table below. We have added the comments based on what kind of review comment it is.

#	ReviewText	UserRating (1-5)	Comments
1	**Delicious**, friendly staff, cool atmosphere, all around great experience! They have a pretty **great** selection of crepes and omelettes, among other things. I enjoyed the Cherry Omelette (cherry tomatoes, swiss cheese, potato slices, and bacon) which was pretty big and cooked **perfectly**! Nice side salad and bread as well. Coffee was great and our server was warm and polite. **Would recommend**!	5	Positive sentiment
2	**Good**, simple, familiar flavor combinations. **Top** quality ingredients. **Can't ask** for more!	5	Positive statement with negation
3	Ive been going here for a while (regularly), but lately I haven't seen the owner and it seems like theyve endlessly been rotating deadbeat employees through there. They're **screwing up a ton** of orders, and they oftentimes dont know the specials or how to run the register....not to mention theyre **unprofessional**. This place is going **downhill** QUICK and it's a shame"	1	Negative sentiment
4	Good service but not so good food. I already ate on many tgi Fridays and this is the first one where the food was **not good**. Will **not come** back again.	2	Negative sentiment
5	**Excellent** dinner. Everything was delicious, especially the charcuterie. Waiter was very helpful in checking on ingredients. Pigs feet/cheek croquette was easily best dish of the night, we ordered multiple times**:)**	5	Emoticons
6	I love pretzels and this place is the best for getting fresh pretzels--which can be a downside. They more often than not run out of pretzels and the wait can be 20 minutes or so for new ones. #N#If you have time, order and take advantage of their wifi (password is their phone number!) and wait for a fresh pretzel. The sandwiches are awesome and the prices are **not bad at all**.#N#Pretzel Poppers are great as well!	4	Capitalized negation
7	Good food, but **definitely not the best** meat I've eaten. Wife **loves** the grilled zucchini, and the squid in the hot bar is very tasty. I like how attentive the people are in the restaurant, but when the bill came I realized that the dessert and coke were both additional and they were pretty **overpriced** (3.75 for a coke). Overall? Lots of meat and not a bad meal at all, but not worth the money we spent **:(**	3	Mixed sentiment
8	This place has **terrible service and worse food**. Complain about anything and you don't get a receipt so you can't complain. On a previous visit they were out of French fries and day before yesterday they were out of patties for all the large beef sandwiches, you know, like a Whopper? Don't know who owns this place but your employees don't give a darn and your managers are covering up. I've never used this word before in a review, but **sux** is not too strong.	1	Slang negation

Now let's dive into different libraries available for sentiment analysis and see how they perform with our sample dataset.

1. VADER (Valence Aware Dictionary and sEntiment Reasoner)
https://github.com/cjhutto/vaderSentiment

VADER is a lexicon and rule-based sentiment analysis tool. The VADER sentiment lexicon takes care of polarity and intensity of the sentiments expressed in the review/sentence. The advantage of using VADER is that there is no need to have training on related text before use. The setup for VADER is very easy as you will see from the installation steps below. VADER is one the most interesting sentiment analysis library that provides multi-dimensional measures of sentiment for a given review/sentence. This helps to understand the weightage of a review in 3 different categories: Positive, Negative & Neutral. Generally, the score for the 3 different categories should sum close to 1. Along with these multidimensional categories this library provides us with Compound score for a review/sentence. The score value is normalized to range from -1 (extremely negative) to +1 (extreme positive).

Installing instructions - Use the command line to do an installation from [PyPI] using pip, e.g.,

```
Anaconda Prompt

(base) C:\Users\vakak>pip install -U vadersentiment
```

Once the installation is done, we are ready to execute the commands to check the sentiment polarity for all the different category of reviews. Code snippet from Jupyter notebook:

```
In [1]: from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

In [31]: reviews = ["Delicious, friendly staff, cool atmosphere, all around great experience! They have a pretty great selection of crepes
                    "Good, simple, familiar flavor combinations.  Top quality ingredients. Can't ask for more!", #Positive statement wit
                    "Ive been going here for a while (regularly), but lately I haven't seen the owner and it seems like theyve endlessly
                    "Good service but not so good food. I already ate on many tgi Fridays and this is the first one where the food was n
                    "Excellent dinner.  Everything was delicious, especially the charcuterie.  Waiter was very helpful in checking on in
                    "I love pretzels and this place is the best for getting fresh pretzels--which can be a downside. They more often tha
                    "Good food, but definitely not the best meat I've eaten. Wife loves the grilled zucchini, and the squid in the hot b
                    "This place has terrible service and worse food. Complain about anything and you don't get a receipt so you can't co
                    ]

In [34]: %%time
         sentiAnalyzer = SentimentIntensityAnalyzer()
         for review in reviews:
             vader = sentiAnalyzer.polarity_scores(review)
             print("score {}".format(str(vader)))

         score {'neg': 0.0, 'neu': 0.497, 'pos': 0.503, 'compound': 0.9926}
         score {'neg': 0.0, 'neu': 0.667, 'pos': 0.333, 'compound': 0.6114}
         score {'neg': 0.163, 'neu': 0.789, 'pos': 0.048, 'compound': -0.8271}
         score {'neg': 0.218, 'neu': 0.729, 'pos': 0.053, 'compound': -0.7984}
         score {'neg': 0.0, 'neu': 0.594, 'pos': 0.406, 'compound': 0.9524}
         score {'neg': 0.021, 'neu': 0.676, 'pos': 0.303, 'compound': 0.9793}
         score {'neg': 0.106, 'neu': 0.665, 'pos': 0.228, 'compound': 0.9168}
         score {'neg': 0.142, 'neu': 0.819, 'pos': 0.039, 'compound': -0.8487}
         Wall time: 43 ms

In [ ]:
```

Score for VADER analysis is updated in **green** from output of the Jupyter notebook [Result (Table2) below].

Analysis: VADER can take care of most of the categories of review sentiments from the above table like Positive sentiment, Positive statement with negation, Negative sentiment, Emoticons, Capitalized negation & Mixed sentiment. The documentation for VADER states that it can handle mixed sentiment example with slang and constrastive conjunction but we were not able to find any concrete review with this scenario.

Performance Numbers: Execution time for VADER for the above dataset is 43ms.

2. TextBlob Sentiment Analysis library

https://textblob.readthedocs.io/en/dev/

TextBlob is a Python library for processing textual data. TextBlob helps to achieve most of the common natural language processing tasks but we will concentrate on sentiment analysis for this discussion. The sentiment property which is part of TextBlob will be invoked and the result is in the form of a named tuple - **Sentiment**(polarity, subjectivity). The tuple has 2 properties-

1. **The polarity score** which is the orientation about the emotions expressed in a sentence. It ranges from -1.0 to 1.0 (-1 being very negative & +1 being very positive)
2. **The subjectivity score** which is about the expressions in a sentence. It ranges from 0.0 to 1.0 (where 0.0 being very objective and 1.0 is very subjective).

Installation instructions

```
Anaconda Prompt

(base) C:\Users\vakak>pip install -U textblob
Requirement already up-to-date: textblob in c:\users\vakak\appdata\local\continuum\anaconda3\lib\site-packages (0.15.2)
Requirement already satisfied, skipping upgrade: nltk>=3.1 in c:\users\vakak\appdata\local\continuum\anaconda3\lib\site-p
ackages (from textblob) (3.3)
Requirement already satisfied, skipping upgrade: six in c:\users\vakak\appdata\local\continuum\anaconda3\lib\site-package
s (from nltk>=3.1->textblob) (1.11.0)

(base) C:\Users\vakak>python -m textblob.download_corpora
[nltk_data] Downloading package brown to
[nltk_data]     C:\Users\vakak\AppData\Roaming\nltk_data...
[nltk_data]   Package brown is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\vakak\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\vakak\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\vakak\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package conll2000 to
[nltk_data]     C:\Users\vakak\AppData\Roaming\nltk_data...
[nltk_data]   Package conll2000 is already up-to-date!
[nltk_data] Downloading package movie_reviews to
[nltk_data]     C:\Users\vakak\AppData\Roaming\nltk_data...
[nltk_data]   Package movie_reviews is already up-to-date!
Finished.

(base) C:\Users\vakak>
```

Once the installation is done, we are ready to execute the commands to check the sentiment polarity for all the different category of reviews like we did for VADER above.

There are 2 sentiment analyzers supported with TextBlob, **PatternAnalyzer** from the Pattern library (default). The other one is **Naive Bayes analyzer**, that we can pass to TextBlob. In our case we will stick to the default analyzer.

Code snippet from Jupyter notebook would be extended to run sentiment analysis for TextBlob:

```
In [35]: from textblob import TextBlob
```

```
In [36]: %%time
for review in reviews:
    blob = TextBlob(review)
    for sent in blob.sentences:
        print("{}".format(str(sent.sentiment)))
```

```
Sentiment(polarity=0.68125, subjectivity=0.725)
Sentiment(polarity=0.30833333333333335, subjectivity=0.7083333333333334)
Sentiment(polarity=0.4375, subjectivity=0.7)
Sentiment(polarity=0.6, subjectivity=1.0)
Sentiment(polarity=0.7, subjectivity=0.675)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.35833333333333334, subjectivity=0.48571428571428577)
Sentiment(polarity=0.5, subjectivity=0.5)
Sentiment(polarity=0.625, subjectivity=0.5)
Sentiment(polarity=-0.14166666666666666, subjectivity=0.47564102564102567)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.3333333333333333, subjectivity=0.5)
Sentiment(polarity=0.7, subjectivity=0.6000000000000001)
Sentiment(polarity=0.13333333333333333, subjectivity=0.4777777777777778)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=1.0, subjectivity=1.0)
Sentiment(polarity=0.5, subjectivity=1.0)
Sentiment(polarity=0.2, subjectivity=0.3)
Sentiment(polarity=0.5, subjectivity=0.43333333333333335)
Sentiment(polarity=0.6, subjectivity=0.4666666666666666)
Sentiment(polarity=0.3181818181818182, subjectivity=0.4772727272727273)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.3, subjectivity=0.5)
Sentiment(polarity=0.7833333333333332, subjectivity=0.8055555555555555)
Sentiment(polarity=0.5666666666666667, subjectivity=0.46666666666666673)
Sentiment(polarity=0.225, subjectivity=0.5750000000000001)
Sentiment(polarity=0.325, subjectivity=0.95)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=-0.16250000000000003, subjectivity=0.4666666666666667)
Sentiment(polarity=-0.7, subjectivity=0.8)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.015873015873015872, subjectivity=0.1984126984126984)
Sentiment(polarity=0.0, subjectivity=0.0)
Sentiment(polarity=0.4333333333333333, subjectivity=0.7333333333333333)
Wall time: 77 ms
```

I have updated the score for TextBlob analysis in **red** from the output of Jupyter notebook to the Result (Table2) below.

Analysis:
Not able to handle the punctuations (. and !) and actually provided score for individual sections of the review text. It doesn't provide the individual positive, negative and neutral multidimensional scores but just provides overall score for the sentiment.

Performance Numbers: Execution time for TextBlob for the above dataset is 77ms.

3. Pattern Sentiment Analysis library

Pattern is another Python library for processing textual data. Similar to other sentiment analysis libraries discussed above, the pattern.en module also performs NLP tasks. Part of speech being the basic and then applying database of adjectives/adverb to classify it as positive or negative sentiment. We will look at Sentiment method from Pattern which is very similar to TextBlob explained above and returns a named tuple with Polarity & Subjectivity.

Installation Instructions

Anaconda Prompt

```
(base) C:\Users\vakak>pip install pattern
Collecting pattern
  Downloading https://files.pythonhosted.org/packages/1e/07/b0e61b6c818ed4b6145fe01d1c341223aa6cfbc3928538ad1f2b890924a3/Pattern-3.6.0.tar.gz (22.2MB)
    100% |████████████████████████████████| 22.3MB 1.2MB/s
Requirement already satisfied: future in c:\users\vakak\appdata\local\continuum\anaconda3\lib\site-packages (from pattern) (0.17.0)
Collecting backports.csv (from pattern)
  Downloading https://files.pythonhosted.org/packages/71/f7/5db9136de67021a6dce4eefbe50d46aa043e59ebb11c83d4ecfeb47b686e/backports.csv-1.0.6-py2.py3-none-any.whl
Collecting mysqlclient (from pattern)
  Downloading https://files.pythonhosted.org/packages/6f/d5/fee36b78367c275d7542181bd522d7204dbfb4bc2ad2c6de72a738059a92/mysqlclient-1.3.14-cp36-cp36m-win_amd64.whl (266kB)
    100% |████████████████████████████████| 276kB 3.7MB/s
```

Once the installation is done, we are ready to execute the commands to check the sentiment polarity for all the different category of reviews like we did for VADER & TextBlob above.

Code snippet from Jupyter notebook would be extended to run sentiment analysis for Pattern:

```
In [38]: from pattern.en import sentiment

In [39]: %%time
         for review in reviews:
                 print("{}".format(str(sentiment(review))))

(0.5038461538461538, 0.7057692307692307)
(0.44000000000000006, 0.49142857142857144)
(-0.022916666666666667, 0.48173076923076924)
(0.3, 0.43888888888888894)
(0.5285714285714286, 0.6571428571428571)
(0.5689393939393939, 0.5856902356902357)
(0.17916666666666667, 0.5263888888888888)
(-0.15317460317460319, 0.48809523809523814)
Wall time: 99.5 ms
```

I have updated the score for Pattern analysis in **purple** from the output of Jupyter notebook to the Result (Table2) below.

Analysis:

It seems to be very similar to VADER but again it doesn’t provide the individual positive, negative and neutral multidimensional scores but just provides overall score for the sentiment. If you look closely at the results, this is not able to handle the “negations” like in sample #4.

Performance Numbers: Execution time for Pattern for the above dataset is 99.5ms.

Result (Table 2)

#	Scores	UserRating (1-5)	Comments
1	{'neg': 0.0, 'neu': 0.497, 'pos': 0.503, 'compound': 0.9926} **(polarity=0.68125, subjectivity=0.725)** **(polarity=0.30833333333333335, subjectivity=0.7083333333333334)** **(polarity=0.4375, subjectivity=0.7)** **(polarity=0.6, subjectivity=1.0)** **(polarity=0.7, subjectivity=0.675)** **(polarity=0.0, subjectivity=0.0)** **(0.5038461538461538, 0.7057692307692307)**	5	Positive sentiment
2	{'neg': 0.0, 'neu': 0.667, 'pos': 0.333, 'compound': 0.6114} **Sentiment(polarity=0.35833333333333334, subjectivity=0.48571428571428577)** **Sentiment(polarity=0.5, subjectivity=0.5)** **Sentiment(polarity=0.625, subjectivity=0.5)** **(0.44000000000000006, 0.49142857142857144)**	5	Positive statement with negation
3	{'neg': 0.163, 'neu': 0.789, 'pos': 0.048, 'compound': -0.8271} **(-0.022916666666666667, 0.48173076923076924)**	1	Negative sentiment
4	{'neg': 0.218, 'neu': 0.729, 'pos': 0.053, 'compound': -0.7984} **(0.3, 0.43888888888888894)**	2	Negative sentiment
5	{'neg': 0.0, 'neu': 0.594, 'pos': 0.406, 'compound': 0.9524} **(0.5285714285714286, 0.6571428571428571)**	5	Emoticons
6	{'neg': 0.021, 'neu': 0.676, 'pos': 0.303, 'compound': 0.9793} **(0.5689393939393939, 0.5856902356902357)**	4	Capitalized negation
7	{'neg': 0.106, 'neu': 0.665, 'pos': 0.228, 'compound': 0.9168} **(0.17916666666666667, 0.5263888888888888)**	3	Mixed sentiment
8	{'neg': 0.142, 'neu': 0.819, 'pos': 0.039, 'compound': -0.8487} **(-0.15317460317460319, 0.48809523809523814)**	1	Slang negation

Conclusion:

TextBlob is a simple and makes it easy to understand and work with Sentiment Analysis. Pattern was able perform well on most of the queries, but VADER was able to handle all different types of review types mentioned above. If we look at the performance numbers VADER is ahead there as well in comparison to TextBlob and Pattern. VADER also provides the compound sentiment that in our case is useful if we want to feed this to a recommender system. This makes VADER our choice for the project that we completed for Sentiment Analysis of review dataset from Yelp.

Team members:

- Varun Kakkar [**vkakkar2**] – Project coordinator
- Prem Prakash [**premp3**]