Overcoming Vader limits by identifying discussion topics and analyzing using specialized dictionaries

Vittorio Haardt, Luca Porcelli & Riccardo Fossato

June 22, 2022

Abstract

The objective of this project is to propose a new development path for sentiment analysis, in particular centered on a more accurate rendering of Vader and on the identification of topics for discussion. The project describes the main strategies tested and briefly describes the implementation of the proposed package.

1 Introduction

The project starts from the basic idea of overcoming the limits of Vader, which is too general and not applicable to specific contexts. An example of the above problem can be found in terms such as "bull" which in everyday language means "bull", but in a financial context it means that a certain stock is destined to grow. The project will be divided into two parts, which have the aim of creating the basis for an articulated and cohesive system of recognition and specific analysis of topics of discussion. The steps will be dealt with in chronological order of execution and finally a general vision of the imagined functioning will be proposed.

The first step is to train specific dictionaries for each topic, starting from the Vader base. The problem is addressed with both the use of supervised and unsupervised models. A logistic model is used to understand the most important words (supervised) and subsequently a word embedding is opted for to find the words most similar to those obtained previously (unsupervised). The words obtained will then be put into a dictionary by assigning an arbitrarily chosen score. As a last step, an evaluation was carried out between the starting dictionaries and the modified ones to understand if the change leads to a sensible result.

The second step dealt with consists in the creation and evaluation of a model that is able to grasp the context we are talking about. The optimal model that is chosen does not necessarily correspond to the optimal model in general, rather a simple and easily replicable model has been chosen. This second phase has the purpose not so much to find the optimal model for the classification of topics, given that more complex models could better manage a number of targets even higher than the one used, as much as a demonstration purpose for the potential of the idea of departure.

In general, the work done wants to have the function of inspiring a starting point for Vader's specialization work which, despite its usefulness, is very general and not very adequate for topics of discussion with specific lexicons. We therefore imagine in perspective a cohesive and articulated system for sentiment analysis, or more generally for the analysis of texts, which is able to grasp the topic or topics covered in specific texts and carry out analyzes referring to a Vader dictionary. modeled to capture all the nuances of the topic in question.

The evaluation of the results will be done by comparison with Vader, it is expected that the specialized dictionaries have a higher accuracy in understanding sentiment, however, if one were to arrive at the disregarded result, one would arrive at the interesting conclusion according to which more complex models or even manual interventions are needed to obtain specialized dictionaries.

1.1 Vader

In order to start from an authoritative and consolidated base, we chose to use the tool Vader, that is the rule-based tool for the most effective sentiment analysis (in especially for social media). Which, as mentioned above, will be updated and manipulated according to the topic, in order to specialize as much as possible.

1.2 Lime

We strongly believe that the implementation of the project should be put in the foreground, with a double purpose. The first is that the developers understand the problems otherwise not visible; the second is to be able to provide such transparency as to be able to make the hypothetical user of the system understand how the texts are classified and evaluated. For the XAI part, Lime will be used for its simplicity and intuitiveness.

1.3 PyPi

Since the project has a real working perspective it was decided to publish a package related to the project on PyPi to make it really usable on Python (at least in its primordial form). In addition, we used a hosting service for software projects, GitHub, to show the functioning of the package and the related project from which it comes. The details of the operation will be explained in a specific section later on.

2 Data

The data used come from various datasets from the well-known Kaggle platform. Separate and specific datasets have been selected for each topic. In general, the datasets are reviews made on the various topics of discussion, this type of text is crucial for the purposes of the analysis for the creation of implementations for Vader. The importance is given by the fact that it is possible to identify the positivity and negativity of the sentiment of the comments based on the vote associated with them, so as to be able to carry out sensible surveys. We now proceed to review the properties of the data used.

The part of interest of the downloaded data are the texts which will be grouped under 4 topics (which we will see shortly). As regards the first model, only the texts relating to a different topic each time and the related labels obtained from the votes will be used (they will be explained in detail for each dataset later on). Instead, the dataset for the second model is made up of a "text" column and a target "topic" column, and includes within it a union of all four datasets.

The initial goal in the search for data was to have about 40,000 lines per topic available. Of which 50% is subsequently used for the creation of specialized dictionaries and the remainder is used for the classification model. Now let's see the topics in detail.

Topics

1. *Food*
   Amazon Fine Food Reviews
   The dataset contains reviews of food products, left by consumers on the Amazon website, with a relative ranking ranging from 1 to 5. For the creation of the label, reviews with a score of 1 or 2 as "Negative", those with 4 or 5 as "Positive". Those with a score of 3 were discarded because they were not of interest for the analysis, 3 was considered a too volatile vote that risks grouping positive and negative reviews into a theoretical "Neutral" category, effectively weakening the analysis.

2. *Electronics*
   Amazon Reviews 2018 - Electronics
   The dataset contains reviews of electronic products, left by consumers on the Amazon website, with a relative ranking ranging from 1 to 5. For the creation of labels, reviews with a score of 1 or 2 as "Negative", those with 4 or 5 as "Positive". Reviews with a score of 3 were discarded for reasons similar to those shown for the previous dataset.

3. *Disneyland*
   Disneyland Reviews
   The dataset contains reviews of the three famous amusement parks of Disneyland, namely those of Paris, California and Hong Kong. The reviews were left by customers on the review site Trip Advisor, with a relative ranking ranging from scores of 1 to 5. For the creation of labels, reviews with a score of 1 or 2 as "Negative", those with 4 or 5 as "Positive". Reviews with a score of 3 were discarded for reasons similar to the previous datasets.

4. *Finance*
   Financial Sentiment Analysis
   Sentiment Analysis for Financial News
   Finance
   The dataset contains reviews and tweets on financial topics, the complete dataset was obtained by combining the three datasets listed above. All three datasets are already equipped with a label that indicates their positivity or negativity.

An attempt has been made to select topics that are as heterogeneous as possible. The selected datasets obviously do not provide a complete view of the topic. The work done does not want to be definitive but wants to be a starting point to be able to develop ideas in the sense of improvement and specialization of Vader.

3 Preprocessing

The cleanliness of the texts plays a fundamental role for the results of the analysis, especially when dealing with texts from reviews and tweets, which are often soiled by links, tags and hastags. In the case of the selected datasets, this kind of *preprocessing* had already been done, simply leaving the text clean. The *preprocessing* is slightly different for the two phases of the project, for this reason the steps will be explained separately.

The cleaning carried out equally for both models, covers the part before starting the analysis in "Python". Initially a *tokenization* was implemented by separating the words with spaces and eliminating everything that is not a word, a space or a number, later we made the words lowercase with a *lowercase*. For each single word, *lemmatization* was then applied to reduce the inflected forms of a word to their canonical form. The *stopwords*, which are not informative for the analysis purposes, have been eliminated only as regards the second model, that is the one for the classification by topic.

Having finished this fundamental phase, it is now possible to proceed with the explanation of the models used and their operation.

4 Part 1: Vader Specialization

This first part of the project aims to identify the words considered most important for the various topics in order to modify their weight in the starting dictionary, to create dictionaries specially shaped for the topics covered. We want to identify the mood of the creator of the texts, in particular the target is "Positive" and "Negative" respectively if the text is judged positively or negatively by the score, as seen in the data description phase, in order to to evaluate the effectiveness of the changes applied to the dictionaries. As previously reported, half of the data was taken to identify the most frequent words and the other half (which is also used for the part relating to the classification of topics) to evaluate the behavior of the new specialized dictionaries.

4.1 Word identification

In order to identify the most frequent words, a *bag of words* was used. The procedure was carried out using *count vectorizer* to select only the 2000 most important words of the separate texts by topic. Then an ad hoc dataset was built containing texts with relative weights (of normalized frequency) of the chosen words. Since only 2000 words have been chosen, texts with a weight of 0 will be present in this phase since they do not contain any of the selected words.

4.2 Model description and surrogate model

In order to have better performance and reduce overfitting on the data, it was decided to use a methodology using a main model to explain the target and its surrogate that was more readable and that best suited to assign weights to words.

For the choice of the so-called main model, two models were evaluated, a *Random Forest* and a *Naive Bayes*. The best rated model was chosen for its highest parameters in the worst performing dataset, however the models are comparable.

The *Random Forest* is therefore the chosen model, the tuning parameters of which have been found through a *Grid Search* each time relaunched for each dataset. For the training data, 70% was taken and the remaining 30% was used as validation. If you are interested in the parameters in particular, please refer to the LINK section at the bottom which in turn refers to the original script. In general, the model has acceptable performances on all datasets as observable from the table (Table 1).

Topics	Accuracy Random Forest	Accuracy Naive Bayes
Food	0.82	**0.85**
Electronics	**0.87**	0.85
Disneyland	0.82	**0.86**
Finance	**0.79**	0.75

Table 1: Table showing the accuracy levels obtained by the models compared on the various datasets

Before using the surrogate model, the main model was relaunched on all data and the predictions were saved. Later it was decided to apply a surrogate model to explain the main model, from which the weights were then derived. The model in question is a *Logistic Model* with default parameters. To adapt this model we used the previously saved predictions as the target.

To finally assign the weights to the words, the coefficients obtained from the surrogate model were used, which were readapted to the Vader congruent weighting method.

4.3 Weight assignment

As just explained, the coefficients of the surrogate model were used as a basis for applying weights for the rule based dictionary. Several methodology tests have been carried out to apply the weights in the best possible way, the strategies in question involve the assignment of weights from -4 to 4 depending on the values of the coefficients relating to the words.

The first method (method 1) is to assign scores to bands, as follows.

coefficient	≤ -2	≤ -1	≤ -0.5	≥ 0.5	≥ 1	≥ 2
score	-4	-3.5	-2.5	2.5	3.5	4

The second method (method 2) is to just assign the extreme weights, as shown below.

coefficient	≤ -0.6	≥ 0.6
score	-4	4

Finally the third method (method 3) is a crossing of the first two, as visible below.

coefficient	≤ -0.5	≤ -1	≥ 0.5	≥ 1
score	-4	-3	3	4

After these tests for the assignment of weights, it was decided to select only the words with "extreme" values, ie greater than 0.6 and less than -0.6, to which a weight of 4 and -4 respectively was applied. It was

observed that method 1 and method 2 were preferable to method 3 of the three alternatives proposed. The adaptation values can be seen from the table (Table 2).

	Food	Electoinic	Disneyland	Finance
Method 1	80.4%	82.3%	**83.8**%	**63.9**%
Method 2	**86.9**%	**83.0**%	83.3%	63.1%
Method 3	81.3%	82.9%	83.7%	63.5%

Table 2: Table comparing the adaptation level of the various methods on the various datasets

With the weights resulting from the operation just seen, specialized dictionaries have been created for each topic, which will now be evaluated for their performance with respect to the starting dictionaries.

4.4 Comparison with Vader

From this point forward, specialized dictionaries are referred to as *Vader New* (V.N.). It is now necessary to assess whether actually switching from Vader to V.N. led to some sort of improvement in the assessment of sentiment. What is expected is that, being the V.N. shaping ad hoc on the topics covered, are able to lead to more specific analyzes and therefore to have a greater accuracy to identify the sentiment of the texts. Otherwise, instead, we would ask ourselves about the causes of a lack of improvement, evaluating other methods for assigning weights or evaluating the insuperability of Vader.

To evaluate the accuracy of Vader and V.N., the resulting compound value was taken and the text was labeled as "Positive", if the latter is greater than 0, and "Negative" otherwise. These compund based labels were then compared with their real counterpart, obtained as explained above. As one would expect, looking at the table (Table 3) we observe that for each topic V.N. is more accurate than Vader by several percentage points.

	Vader	Vader New
Topics		
Food	72.1%	**86.9%**
Electronic	75.5%	**83.0%**
Disneyland	82.2%	**83.3%**
Finance	49.7%	**63.1%**

Table 3: Table comparing the percentage fit level between Vader and Vader New

The result obtained shows how the aim of the project can actually lead to a substantial improvement of Vader through his specialization, leading to ever more precise analyzes on increasingly capillary topics.

4.5 Word embedding and comparison with previous model

It was decided to evaluate whether the use of a *word embedding* could further increase the performances of the V.N. compared to their classic form. Two alternatives named *Vader New four* and *Vader New one change* have been evaluated.

In general, what these two changes carry out is, thanks to word embedding, the modification of the weights for the words most similar to those previously modified. We have opted to use the unsupervised model with *fastText* rather than *word2vec* as it is considered more efficient. *V.N. four* in particular selects the words with a similarity score greater than 0.99, with respect to the words contained in V.N., and assigns these words a value of 4 or -4 based on the positive or negative value of the words to which they have been associated. Instead, *V.N. one change* selects the words for which to change the weight in the same way as *V.N. four*, but change the weights by adding or subtracting 1 with respect to the value these words have in Vader, based on the positive or negative value of the words they have been associated with.

As reported in the table (Table 4), the performances of the various N.V. methods are observed, in order to choose the most appropriate methodology for each method.

Topics	Vader New	V.N. four	V.N. one change
Food	**86.9%**	81.86%	81.86%
Electronic	83.0%	83.08%	**83.23%**
Disneyland	**83.3%**	83.12%	83.12%
Finance	63.1%	**70.15%**	43.36%

Table 4: Table comparing the level of fit between Vader New and its versions that use word embedding

Having evaluated the adaptation performance, it was decided not to apply word embedding to the Food and Disneyland topics, instead as regards Electronic the best choice falls on *V.N. one change* and finally for Finance there is a clear improvement in comparison to the other two methods as regards *V.N. four*.

The specialized dictionaries are now complete and fully usable for analysis on the texts for the related topics, as seen during the steps that led to the result, the specialized dictionaries will produce more accurate analyzes than Vader did. We now want to draw attention to how, regardless of the choices made on the methods of assigning weights and on the use of word embedding, the results of specialized dictionaries still surpass those of Vader for practically all combinations of choices on all topics.

A practical example of how specialized dictionaries have improved Vader is given one of the sentences of the Electronic topic.

" *Faulty on arrival. The wire for one channel wasn't ...* "

The word *faulty* takes on a generally more negative connotation in natural language when it comes to electronic objects. Since the topic in question includes reviews of electronic objects, we see how the weight of this word went from -1.8 for Vader to -4 for Specialized Vader. By passing the entire sentence from a compound value of -0.32 to one of -0.90.

5 Part 2: Topic classification

As previously said, the second model aims to be able to classify texts according to the topics they speak of. The chosen model therefore obviously has the multi-class target variable, referring to the above arguments. In this paragraph the main characteristics will be described. Of the 50 % of the total data taken previously, a proportion of 70% and 30% was chosen for the training and test data respectively.

5.1 Model description

In order to have optimal classification performances, two different models were evaluated. The models in question are *Naive Bayes* and *Decision Tree*. For the *Naive Bayes* we have opted for an alpha value of 0.1. For the *Decision Tree* we opted for the Gini criterion, a tree with maxleafnodes without limit and a maxdeap also unlimited, so as to leave the model as loose as possible. All parameters have been tuned to select the optimal ones.

5.2 Winning model

The metric of interest mainly observed is accuracy, but still the models tend to have high values for the other metrics as well. Looking at the performance, the winning model was *Naive Bayes*, with an accuracy level of 0.95 (compared to 0.88 for the tree). Therefore the multitarget classification with the aim of identifying the topic of discussion will be entrusted to *Naive Bayes* with the parameters previously selected.

However, it is necessary to specify that this model is optimal for this type of classification of only four topics. In the case of larger discussion topic classification systems with far greater numbers of target alternatives, the optimal model is likely to be a model with greater complexity, such as a neural network. It is therefore remembered that the model is created, not so much for its actual usefulness, but rather to show the possibility of developing a cohesive system that identifies the topic of discussion and refers to a specific dictionary. The aim is therefore to be able to inspire the creation of specific dictionaries, for all the topics

of discussion, and to create a system that automatically recognizes the topic and redirects to the associated specialized dictionary.

5.3 Explainability

Explainability, as previously stated, has been entirely entrusted to Lime. The importance of this last phase is not to be underestimated, since in order to really rely on a classification model it is necessary to deeply know its functioning and the method of choice, so as to be able to apply improvements and be able to gain the trust of the non-expert user. .

By randomly sampling between the different texts, it was observed how the model classifies on the basis of really meaningful words and how on the contrary it is not based on general words, which would be applicable to any context. Below (Figure 1) is a text regarding the *Electronic* argument, in order to give a general idea of the explanatory capacity of the classification.

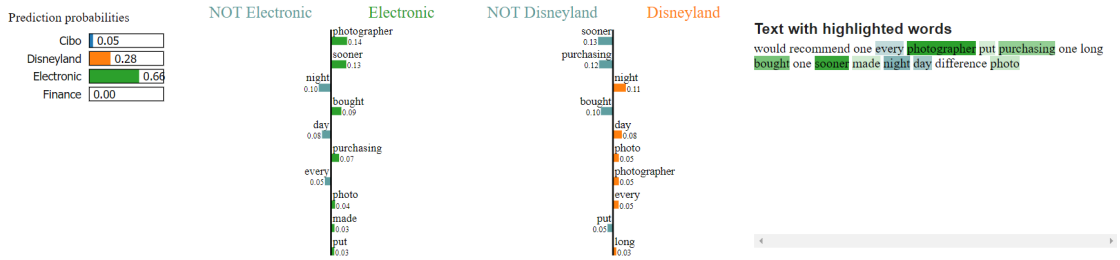

Figure 1: Text identified correctly as per the textit Electronic argument, the tokens that led the model to classification are highlighted.

6 Joint operation models

The joint functioning of models has been extensively introduced in the previous sections. To summarize, it is reported how the two parts of the study, namely the creation of the Vader New and the creation of a model that is able to identify the topic of discussion, are thought of as distinct parts of a single process. Obviously, as already specified, the study done does not aim to be definitive in any way, but rather serves as a starting point and inspiration.

A function has therefore been created that by receiving any text (obviously limiting ourselves to our four arguments) is able to accurately recognize the argument and then entrust a sentiment analysis, which produces a compund score, to the appropriate dictionary for the argument. . Thanks to functions that work like the one just explained, it is possible to provide a real leap forward for the accuracy and reliability of a system like Vader, which, although reliable and useful, needs an evolution since it remains too general, leading to unreliable analyzes for texts concerning specific topics and therefore having a specific language.

An example of the aforementioned function is now proposed, which reports its output, in order to better understand the potential of the system conceived.

	Tipo	Text	Vader_new	Compound
0	Electronic	good uv protection use keep glass lens safe harm	Positive	0.9612

Figure 2: Text identified as *Electronic*, highlighted the compound score, assigned by the specific dictionary.

7 A new Python package: "vadernew"

This section quickly presents the package introduced as a result of the analyzes carried out. "vadernew" can be used for text analysis for the above topics. The analyzes using this package instead of Vader, as explained above, will be more precise and reliable.

You start by installing the "vadernew" package via pip install.

```
In [1]: ! pip install vadernew
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting vadernew
  Downloading vadernew-2.0.tar.gz (208 kB)
     |--------------------------------| 208 kB 4.9 MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from vadernew) (2.23
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from reque
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->va
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from reques
Building wheels for collected packages: vadernew
  Building wheel for vadernew (setup.py) ... done
  Created wheel for vadernew: filename=vadernew-2.0-py2.py3-none-any.whl size=201181 sha256=73123c38b4d
  Stored in directory: /root/.cache/pip/wheels/6e/61/92/b3cf7e69a81abfdb3186292b908158e2a0590c7871fa6ad
Successfully built vadernew
Installing collected packages: vadernew
Successfully installed vadernew-2.0
```

The package contains the dictionaries and functions relating to the 4 topics separately, which can be imported individually.

```
In [2]: from vadernew import vader_food
```

```
In [3]: from vadernew import vader_electronic
```

```
In [4]: from vadernew import vader_disneyland
```

```
In [5]: from vadernew import vader_finance
```

The package contains the dictionaries and functions relating to the 4 topics separately, which can be imported individually.

That is, two replacement functions, respectively of `SentiText()` which identifies the string-level properties relevant to the sentiment of the input text, and `SentimentIntensityAnalyzer()` which instead assigns a sentiment intensity score to the phrases. The two functions are renamed for each argument.

```
In [6]: from vadernew.vader_food import Food_ST, Food_SIA
```

```
In [7]: from vadernew.vader_electronic import Electronic_ST, Electronic_SIA
```

```
In [8]: from vadernew.vader_disneyland import Disney_ST, Disney_SIA
```

```
In [9]: from vadernew.vader_finance import Finance_ST, Finance_SIA
```

For the working of the `_ST` functions, please look at the classic Vader guide for `SentiText()`, as they are not the point of the changes made.

Now let's see how the `_SIA` functions work and how with one of its sub-functions we find the compound values. The resulting values are more accurate, as they refer to specific dictionaries. For all callable sub-functions, reference is always made to the VaderSentiment guide, remember that the operation of the vadernew package is in all respects the same as that of VaderSentiment, the only change is the specificity of the dictionaries used.

```
In [11]: sentence = "Just an example"
         analyzer = vader_finance.Finance_SIA()
         vs = analyzer.polarity_scores(sentence)
         print("{:<13} {}".format(sentence, str(vs))

Out [11]:Just an example {'neg': 0.0, 'neu': 0.286, 'pos': 0.714, 'compound': 0.7184}
```

Inclusion we invite you to try and experiment the potential of the package, which, we remind you once again, only acts as a showcase of how a specialization of VaderSentiment leads to more accurate analyzes.

8 Conclusions

By reviewing all the steps of the project, we have seen how, from expected results, making changes in order to specialize Vader on topics of discussion leads only to improvements. In general, it is necessary to underline the result observed in sections 4.4 and 4.5, according to which, regardless of the methods of assigning weights and the models used to assign them, Vader is always worse than its specialized versions. This result leads to the conclusion that despite being an innovative and extremely effective system, Vader is now easily overcome, or better said, improved.

As a conclusion, we invite you to grasp the input proposed by this project, that is the development, starting from Vader, of a system that is able to identify the topic of discussion text by text, sentence by sentence and which uses specialized dictionaries in order to have performing and accurate analyzes.

As a last note, we recommend that anyone who intends to apply textual analyzes, such as sentiment analyzes, on texts whose topic of discussion is already known at the outset, to take datasets with characteristics similar to those seen previously and to train a specialized dictionary for the subject. The resulting dictionary, used as illustrated in the project, regardless of the attention paid to the choice of models and the assignment of weights, should still be more efficient than the general one used by Vader.

9 Links

The links to the python notebooks for the development of the project are reported in this final section.

- Asegnazione Pesi
- Word Embedding
- Confornto con Vader
- Classificazione

Finally, the links to the GitHub and PYPI page are reported.

- GitHub: vadernew
- PYPI: vadernew