CS 598 Data Mining Capstone Task 7

Ampliview

Kyle Demeule

2020-08-02

Introduction

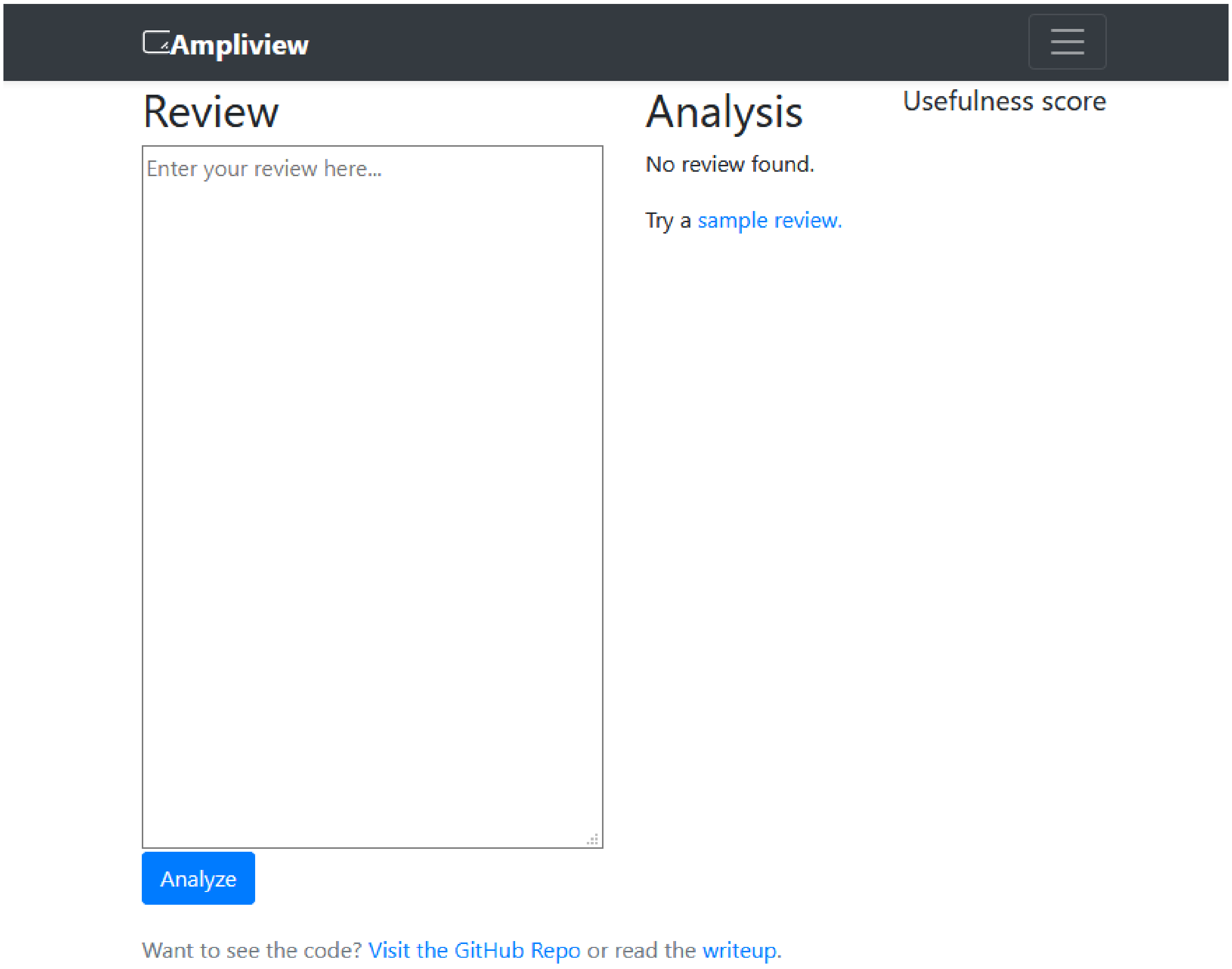
For the final task in CS 598 Data Mining Capstone we will be building a web application system that utilizes some of the functions we previously experimented with as part of the course. For my application I focused on sentiment analysis and prediction to create a system to help users write better reviews for Yelp that are more likely to be useful.

Application Scenario and Intended Use

Users are submitting reviews to Yelp to provide their perspective and to be helpful to other people in making their dining decisions. Yelp recognized this and added the ability for users to mark reviews as "Useful", which can be important feedback not only to people reading reviews, but also to the reviews author. However review writers wont get this feedback until they've submitted their review and people are reading it, creating a chicken-and-egg problem. To help with this I created Ampliview, a tool which will provide sentiment markup and a usefulness prediction score to review writers as they write their review. This immediate feedback can be used to help structure and craft their review into its best possible version.

Interface Design

The Ampliview interface is a two column interface: in the left column the user enters their review, and the right column is where analysis of the review will be displayed:



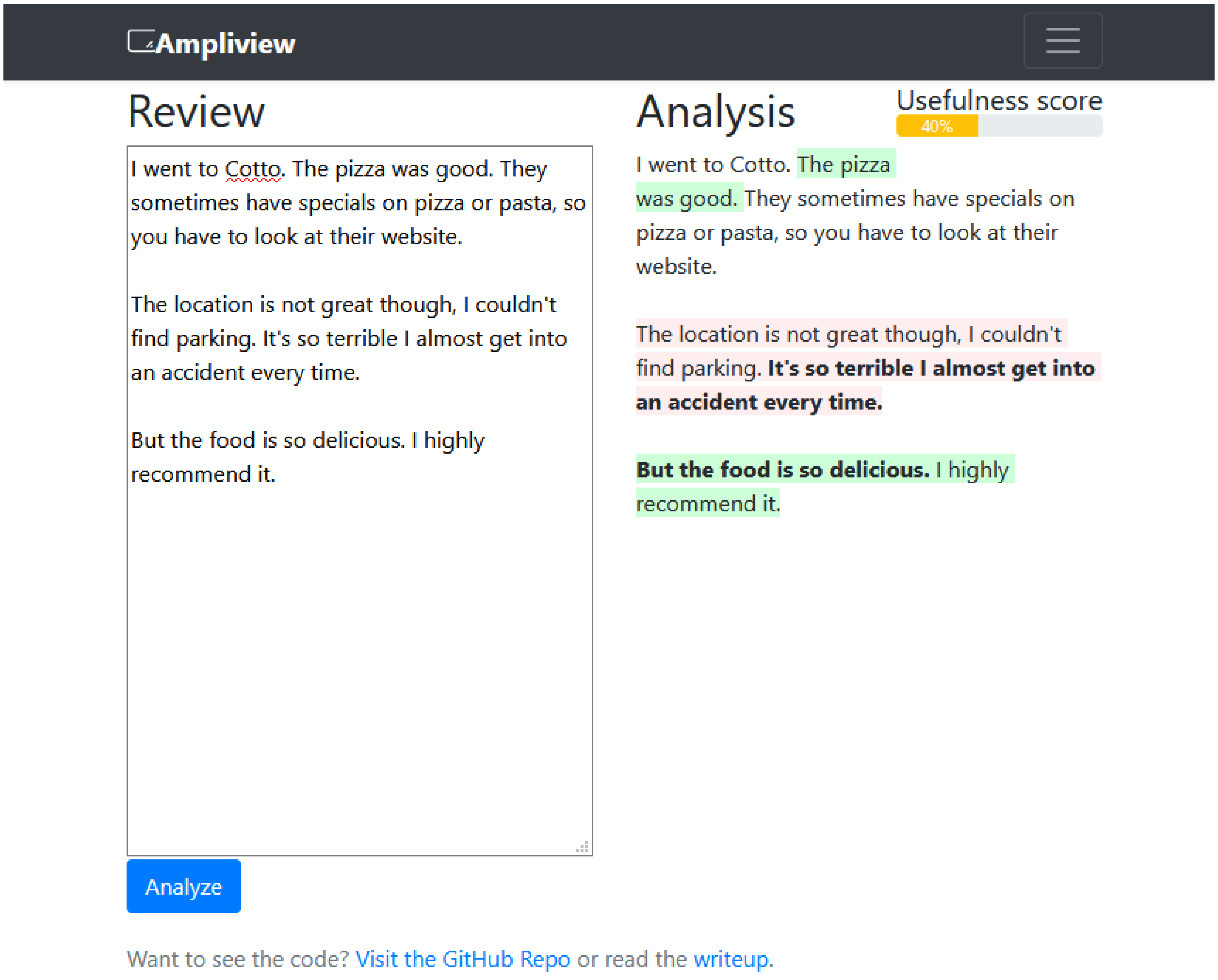
No review found.

Try a sample review.

Want to see the code? Visit the GitHub Repo or read the writeup.

Figure 1: Ampliview

As the user is writing the review, whenever they want analysis of the review they can press the "Analysis" button, and the analyzed review will appear in the right column:



I went to Cotto. The pizza was good. They sometimes have specials on pizza or pasta, so you have to look at their website.

The location is not great though, I couldn't find parking. It's so terrible I almost get into an accident every time.

But the food is so delicious. I highly recommend it.

I went to Cotto. The pizza was good. They sometimes have specials on pizza or pasta, so you have to look at their website.

The location is not great though, I couldn't find parking. **It's so terrible I almost get into an accident every time.**

But the food is so delicious. I highly recommend it.

Want to see the code? Visit the GitHub Repo or read the writeup.

Figure 2: Ampliview

Ampliview is providing two pieces of feedback: it is marking up sentences with their perceived sentiment, and is providing a score for the predicted usefulness of the review. I'll describe each in more detail.

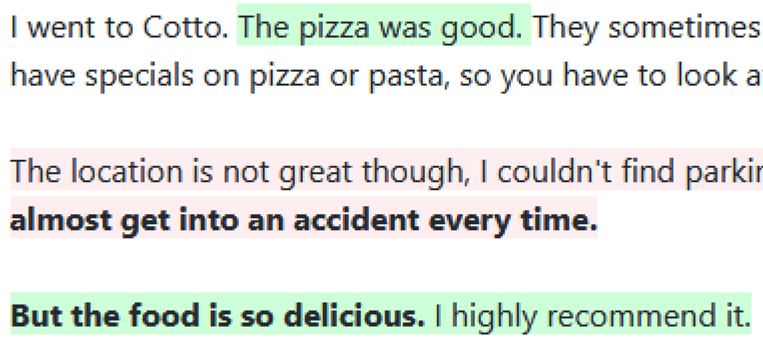
Sentiment Analysis

Ampliview will break down the review into sentences and analyze the sentiment of each sentence. This can help you better understand how people will perceive your review, as well as give you a visual representation of the structure of your review. This can help you structure your review (by having certain sentiments expressed in certain parts), can help ensure that the perception of your writing matches your intent, and can also provide an indication of what parts of your review are lacking in their descriptiveness. Sentiment analysis is something that we previously did in the capstone project, primarily for Task 4 where we were trying to determine the sentiment of particular dishes at particular restaurants

Each sentence is given a sentiment score that ranges from -1.0 to +1.0. The closer the score is to +1.0, the more positive the sentiment of the sentence. The closer the score is to -1.0, the more negative the sentiment of the sentence is. Scores near 0.0 have neutral sentiment. Sentiment analysis is done using the vaderSentiment tool[1].

[1] Hutto and Gilbert (2015)

Once each sentence is scored, Ampliview will apply highlighting based on the sentences scoring. We can look at the highlighting for an example review:

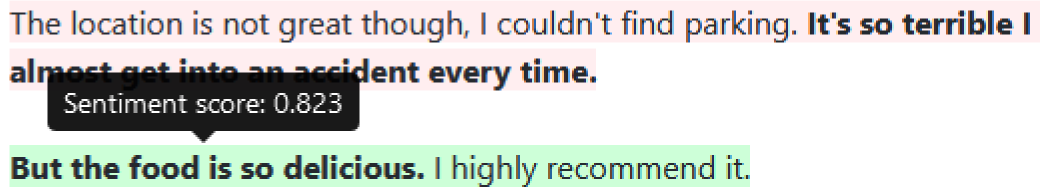


I went to Cotto. The pizza was good. They sometimes
have specials on pizza or pasta, so you have to look at their website.

The location is not great though, I couldn't find parking. **It's so terrible I almost get into an accident every time.**

But the food is so delicious. I highly recommend it.

You should be able to see that sentences with positive sentiment are highlighted in green (e.g. "The pizza was good") and sentences with negative sentiment are highlighted in red (e.g. "The location is not great though, I couldn't find parking"). Sentences with neutral sentiment have no highlighting (e.g. "I went to Cotto"). Sentences with extreme sentiment in either the positive or negative direction are bolded (e.g. "The food was so delicious"). The exact sentiment score can be seen by hovering over a sentence:



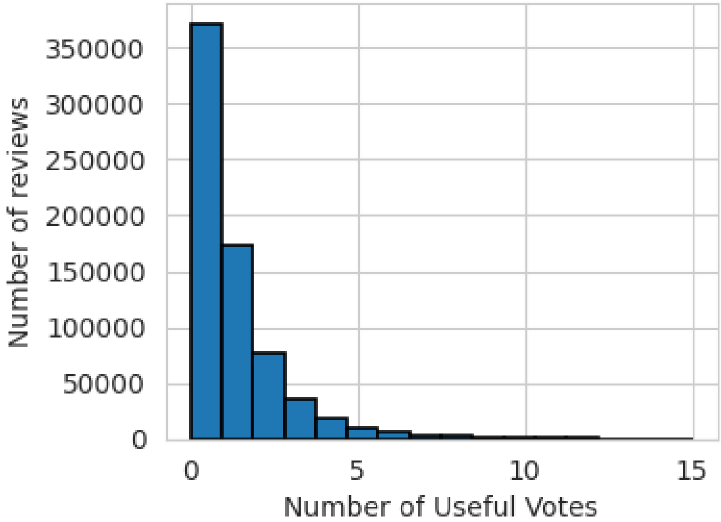
The location is not great though, I couldn't find parking. **It's so terrible I almost get into an accident every time.**

Sentiment score: 0.823

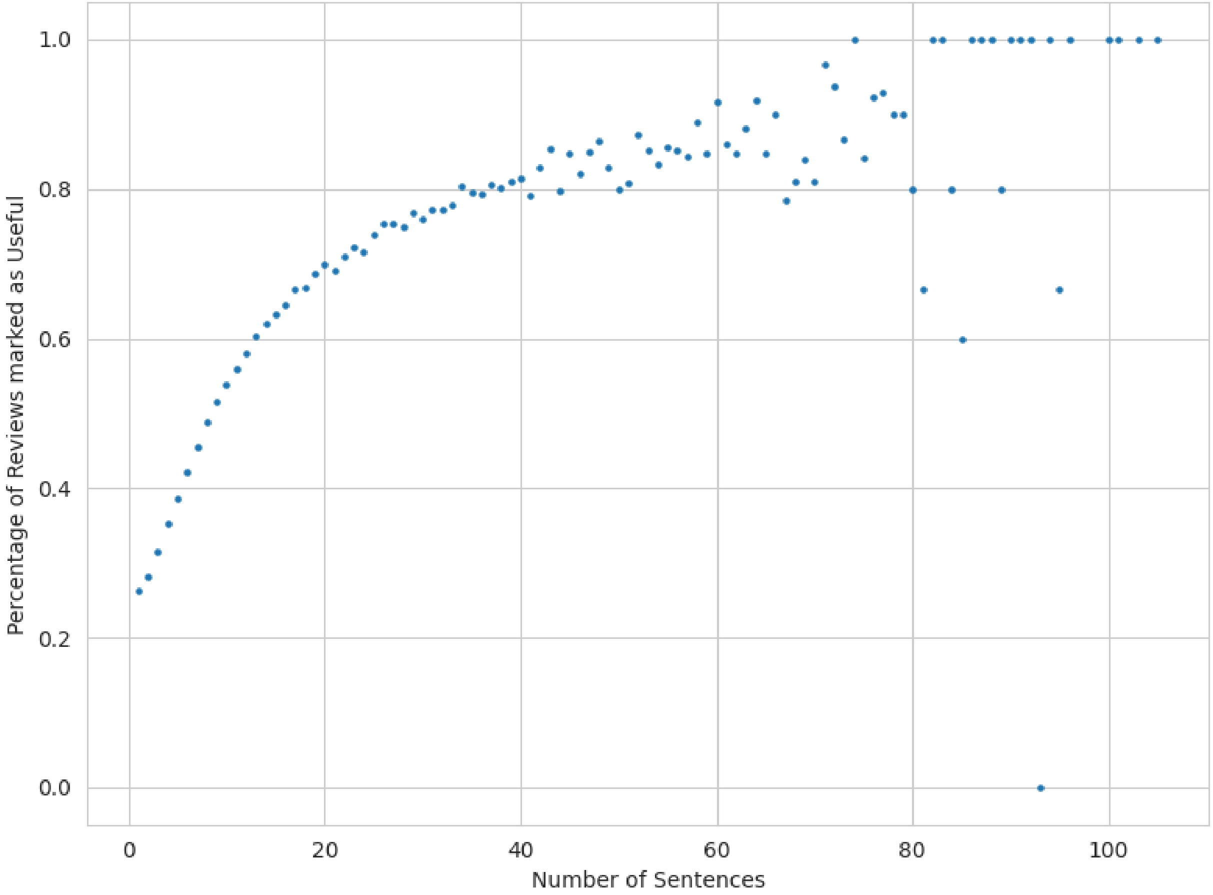
But the food is so delicious. I highly recommend it.

Usefulness Score

The second thing Ampliview is doing is providing a score for how likely it thinks the review is to be marked useful on Yelp. This is similar to Task 6 where we were trying to predict whether restaurants would pass a hygiene inspection based on reviews, except this time we are trying to predict whether a particular review would be marked as useful. We can look at the Yelp dataset and see how many "Useful" votes reviews got:

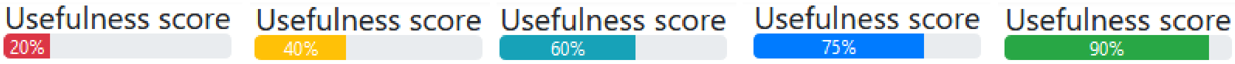


The majority of reviews received 0 votes, but around 50% received one or more useful votes. We can create an indicator variable for whether a review received at least one useful vote, and compare different properties of the reviews to this indicator variable. For example if we look at the ratio of useful reviews by the number of sentences in each review:



There appears to be a positive correlation between number of sentences and usefulness of reviews. In fact this relationship looks logarithmic: as reviews increased from 1 to 18, there is a steep incline in the ratio that are marked as useful. After around 18 sentences reviews continued to become more useful, but at a lower rate than before.

As the users continues to write and analyze their review, the updated score will be shown. The score is shown as a percentage, indicating the probability that the review would be found useful by readers at Yelp. This can help users with writing reviews in multiple ways: it can help them fine tune parts of their review by giving them immediate feedback, and can also provide motivation to keep writing their review and add more detail. The scores are shown as a percentage, with different colors as the score increases:



Two logistic regression model were trained using statsmodel. One was applied to very short reviews (under 100 characters), and longer reviews had a different model applied. The short reviews had the following model:

```
                           Logit Regression Results
==============================================================================
Dep. Variable:              is_useful   No. Observations:               706646
```

```
Model:                          Logit   Df Residuals:                   706643
Method:                           MLE   Df Model:                            2
Date:                Mon, 27 Jul 2020   Pseudo R-squ.:                 0.06707
Time:                        00:33:32   Log-Likelihood:            -4.5611e+05
converged:                       True   LL-Null:                   -4.8890e+05
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept          -4.1214      0.027   -154.771      0.000      -4.174      -4.069
log_char_length     0.6190      0.006    100.980      0.000       0.607       0.631
log_num_sentences   0.1044      0.007     14.572      0.000       0.090       0.118
==============================================================================
```

Which utlizes their total length and number of sentences. The longer model used the following coefficients:

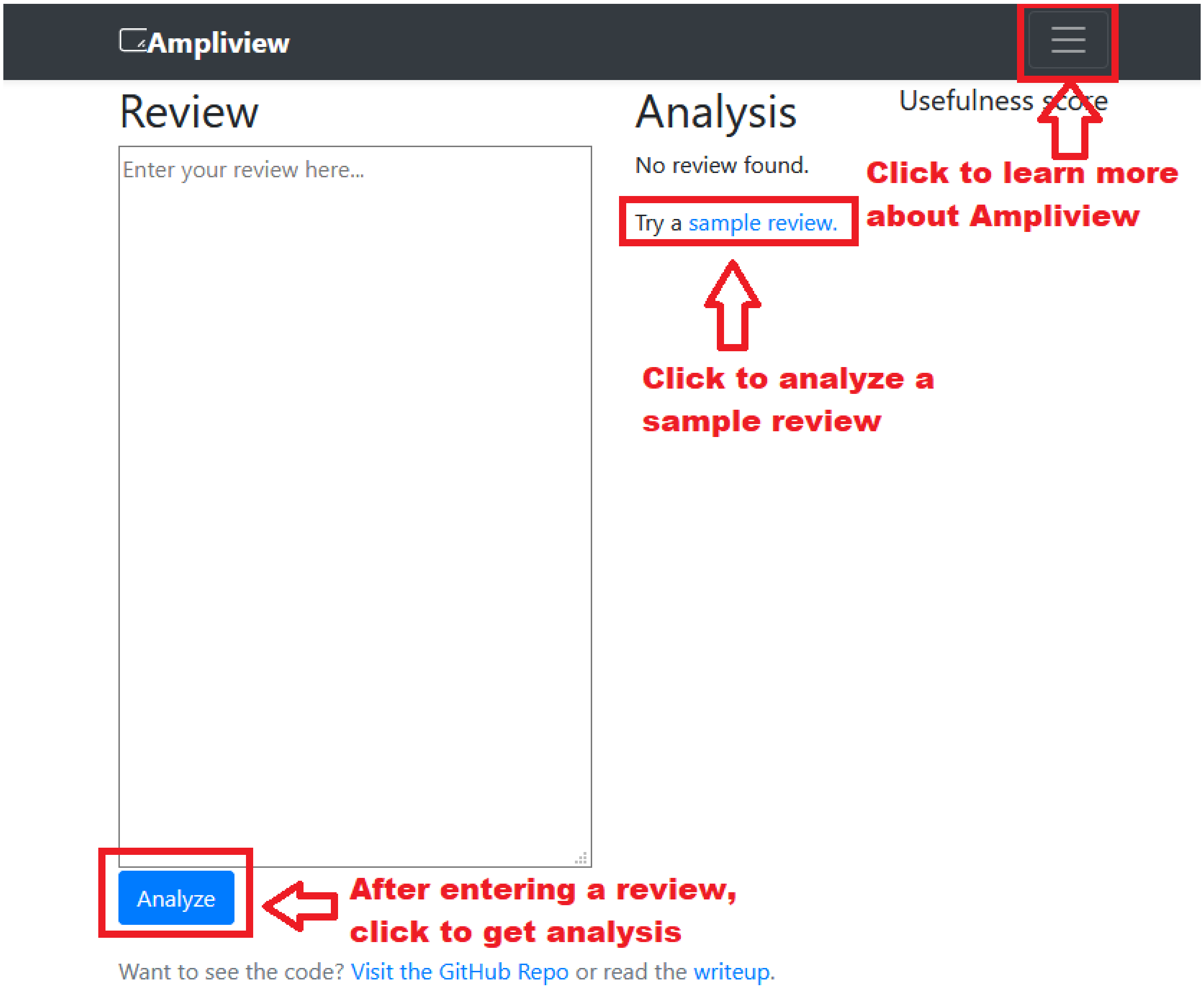
```
                        Logit Regression Results
========================================================================
Dep. Variable:              is_useful   No. Observations:                50000
Model:                          Logit   Df Residuals:                    49996
Method:                           MLE   Df Model:                            3
Date:                Fri, 31 Jul 2020   Pseudo R-squ.:                 0.05565
Time:                        19:25:27   Log-Likelihood:                -32663.
converged:                       True   LL-Null:                       -34588.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================================
                                      coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
intercept                          -1.4228      0.026    -54.234      0.000      -1.474      -1.371
log_review_distinct_pos_sentences   0.4795      0.016     29.455      0.000       0.448       0.511
log_review_distinct_neg_sentences   0.3440      0.017     20.831      0.000       0.312       0.376
log_review_distinct_neut_sentences  0.3241      0.015     21.372      0.000       0.294       0.354
==============================================================================================
```

This model utilizes the number of distinct sentences with each sentiment class. It also applies a log function to the input.

The model was trained only on reviews from Yelp, so it can only rank and compare the properties of useful reviews from less useful reviews. It was not designed to distinguish reviews from non-reviews, so if non-reviews are inputted that have similar properties to useful reviews (e.g. are fairly long) they will also score highly on the “Usefulness Score”, despite them obviously not being useful as Yelp reviews.

Other Features

Finally a few helpful parts were added to the application:



No review found.

Try a sample review.

Click to learn more about Ampliview

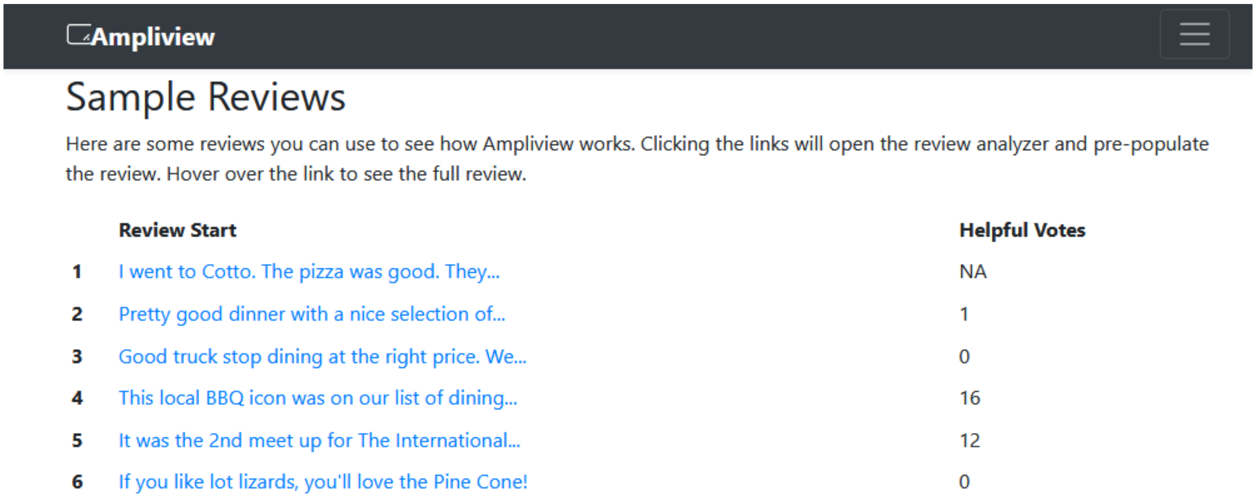
Click to analyze a sample review

After entering a review, click to get analysis

Want to see the code? Visit the GitHub Repo or read the writeup.

Figure 3: Ampliview

The empty state of the application provides a link to fill in a sample review, this can be used to easily test the application. The "Hamburger" button opens up an About page with info on Ampliview. It also has links to pages describing the sentiment analysis and usefulness score. Finally a link to a page with sample reviews is shown:



Sample Reviews

Here are some reviews you can use to see how Ampliview works. Clicking the links will open the review analyzer and pre-populate the review. Hover over the link to see the full review.

	Review Start	Helpful Votes
1	I went to Cotto. The pizza was good. They...	NA
2	Pretty good dinner with a nice selection of...	1
3	Good truck stop dining at the right price. We...	0
4	This local BBQ icon was on our list of dining...	16
5	It was the 2nd meet up for The International...	12
6	If you like lot lizards, you'll love the Pine Cone!	0

Clicking the links here will show the analysis for the selected review. This page also shows how many useful votes each review received, so you can compare useful reviews from less useful reviews.

Access URL

Ampliview is currently being hosted on Heroku and can be accessed at https://ampliview.herokuapp.com/. The code for the website is also publicly available and is hosted on GitHub at https://github.com/kyledemeule/ampliview.

References

Hutto, C. J., and Eric Gilbert. 2015. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text." In *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*.