Final Project of Team 18

Name	PID
Chenghao Tsai	A53274551
Yu-Han Chen	A53270592
Wei-Cheng Huang	A53267614

Part 1 Classifier from PA2

Comment Classification

Classifier: Logistic Regression
Vectorizer: TfidfVectorizer
Data Size:

- Training
  * POSITIVE: 2307
  * NEGATIVE: 2304
- Dev
  * POSITIVE: 231
  * NEGATIVE: 234

Overview of our website

Home Page:

Content Page:

Comment Classification

Significant Positive Words

Significant Negative Words

Input Sentence **Example**

The restaurant is fantastic. The service there is awesome and the food is the best I've ever had.

Classify

Result

Analysis

Significant Words

Sentence Similarity

Sentence Vector

Result

POSITIVE👍 with 98.3% confidence

Analysis

Significant Words

Significant **positive** word occurrences in this sentence:
is, awesome, and, best, the best

Significant **negative** word occurrences in this sentence:

Sentence Similarity

With **positive** corpus vector: **0.446**

With **negative** corpus vector: **0.262**

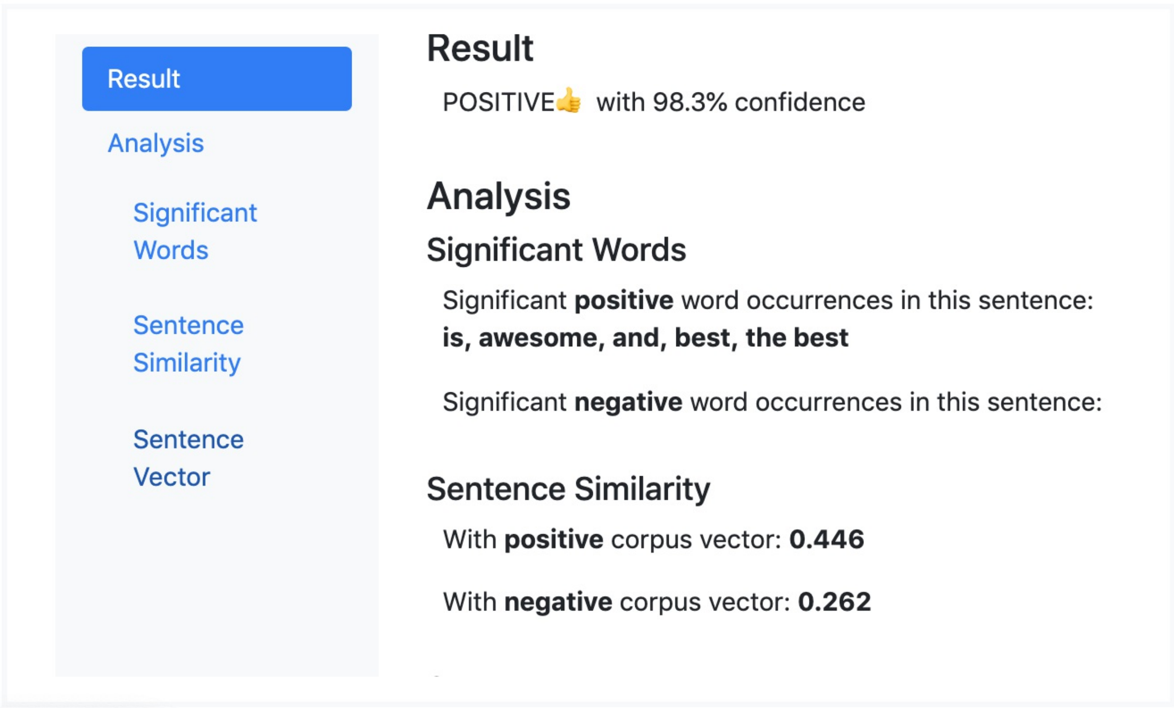
1.1

- ## WordCloud

The wordcloud on the top displayed the most important words determining the comments' label. Here we choose top 50 words with highest coefficeints.

- ## Result

It gives the predicated label of the comment as either POSITIVE or NEGATIVE along with the probability of the label produced by our model. In this case it's 98.3%.



- ## Analysis

  - ### Significant words

    With the final weight of the model's paramter, we can know that the features with the top positive and negative weights might be the ones that has more significant influence on the model's predictions. Therefore, we extract the top k (k=50) and bottom k features out and define the top k features (words) as the Significant positive words and the bottom k as the Significant negative words.

We believe that if the sentence is predicted as POSITIVE, it is very likely that there are more words in the input sentence that occured in the significant positive words than significant negative word.

As for this example, we can see that there are multiple positive words but no significant negative words occurances, thus, this sentence is highly confident as predicting as POSITIVE.

- ### Sentence Cosine Similarity

  Cosine simliarity is used to calculate the similarity of two vectors, so we calculate the cosine similarity of the input vector with both the positive and negative vectors.

  Positive, negative vector generation:
  We utilize the TfidfVectorizer to fit on the whole training corpus, and then transform on both the positive, negative corpus which contains only the positive reviews and the negative ones. Therefore, we get a matrix for both positive and negative corpus. But both with a dimension of 70965 features.We then get the average on all examples and finally retreive a 1 * 70965 dimension vector.

  Since we believe the top and bottom k features are more important, we retrieve those features out of all features (70965) and finally a 1 * 2k vector for both positive and negative.

  The input sentence follows the same idea and is transformed to a 1 * 2k dimension vector.

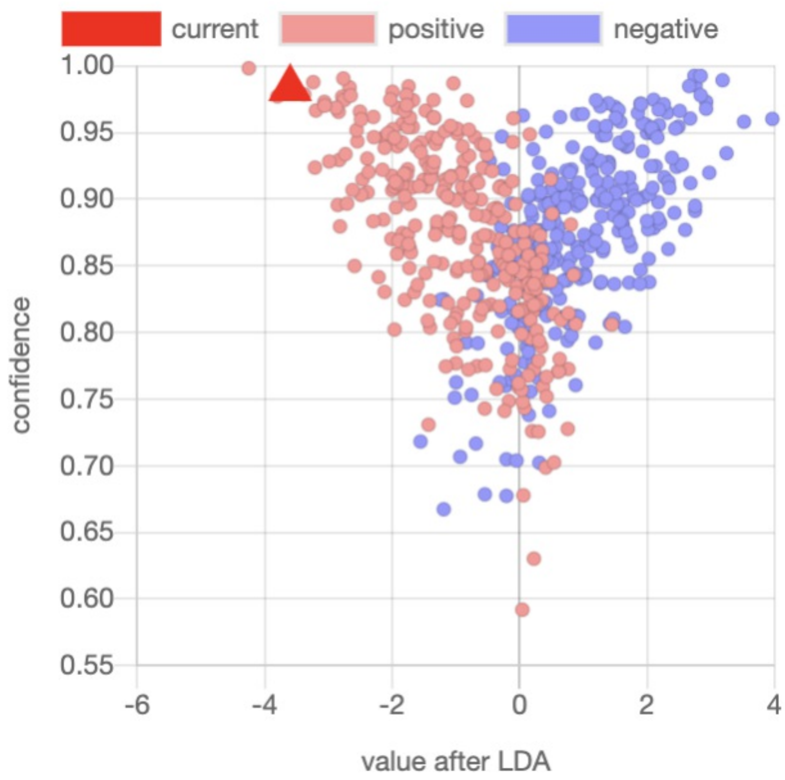
  The cosine similarity is calculated as the follow:

$$cosine\_similarity(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||||\vec{v}||}$$

- Sentence Vector on LDA

  The graph shows the data distribution of the dataset and the current sentence vector using LDA. See detailed description in part 3.2.

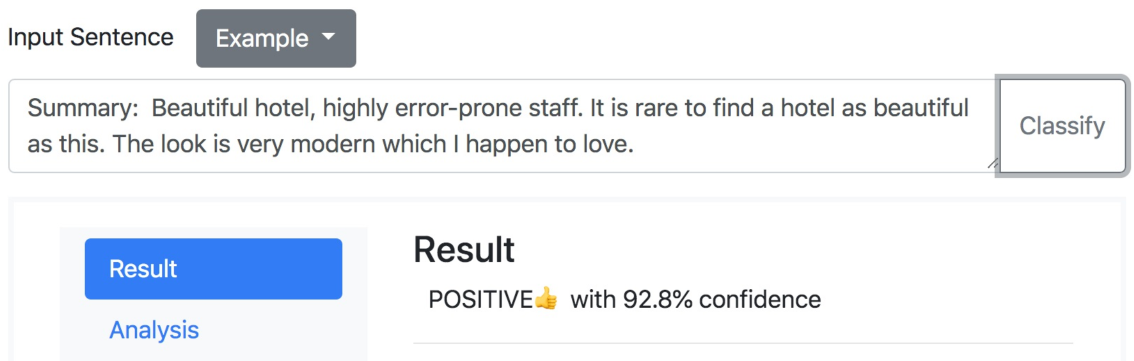
Sentence Vector



1.2

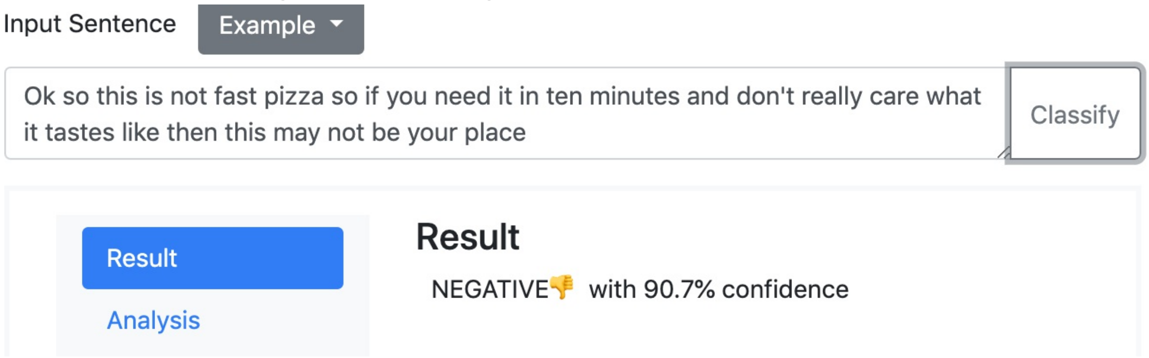
We give two examples of overconfident reviews of our model.

1. The following is an example that our classifier is overconfident in. The true label is negative, while predicted as 92.8% positive.
   From the input sentence you can see that it contains mostly compliments of the hotel but just mentioned an drawback of it on its staff. However, the comment overall doesn't seemed like a NEGATIVE review to us but it is labeled as NEGATIVE.



2. The below review is does not have a obvious sentiment as whether it's positive or negative. It is somewhat vague literally since it's saying that it's slow but on the other

hand, it's also inferring that it tastes good since it says that if you don't care about what it tastes like then this might be the wrong place.



We concluded that the overconfident examples are very likely to be vague in its context and whether it's positive or negative might be controversial.

Part 2

Toxic Comment Classifier

Classifier: Logistic Regression
Vectorizer: TfidfVectorizer
Data Size:

- Train: 158450
- Dev: 17606

2.1

Our model is used to determine if a comment is toxic or not. The reason why we chose this task is because nowadays everyone uses internet, and some of us are suffering from cyber bullying. People can insult or humiliate others just by typing annonymously. Such phenomenon even somehow increase the suicide rate. Therefore, we hope to construct a tool that could predict toxic comments in advance and prevent such tragedy.

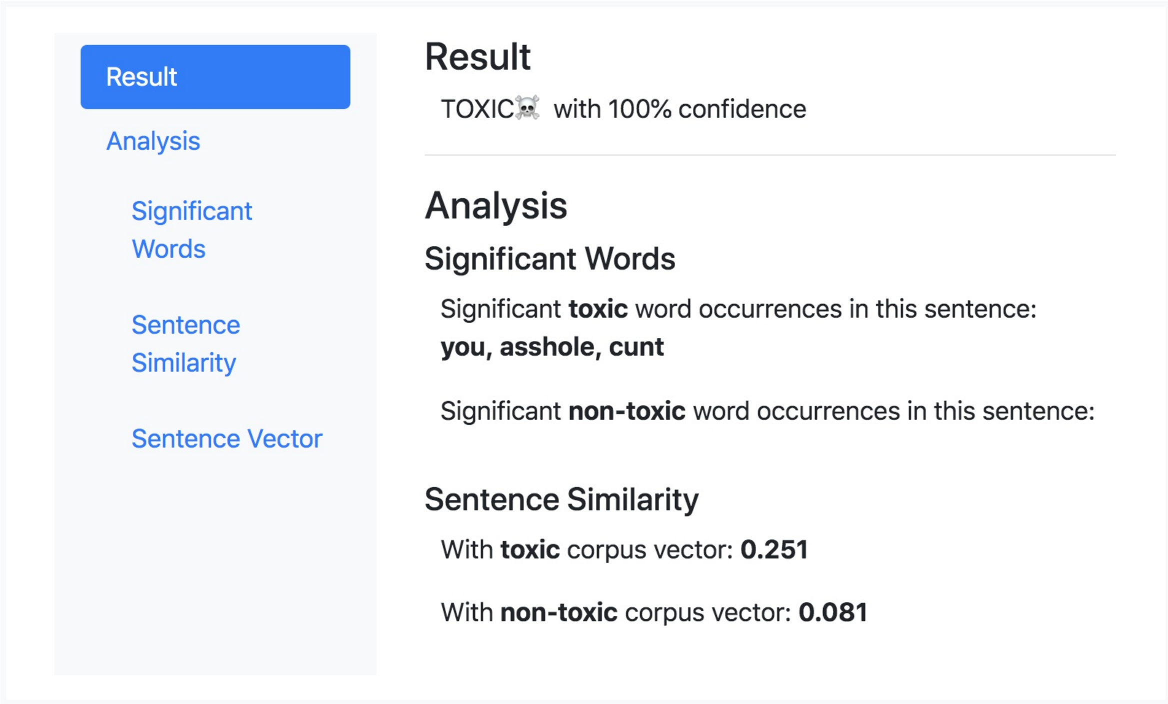
We combined two dataset: toxic comments and cybertroll as our training data. If a comment is considered to be toxic, our model would label it as "toxic". Otherwise, "non-toxic".

2.2

Home Page

Content Page

Our example, which is predicted 100% TOXIC. We also show the significant toxic/non-toxic words in the analysis, as well as compare the sentence vector with corpus vectors, indicating the similarities toward the average toxic/non-toxic vector.

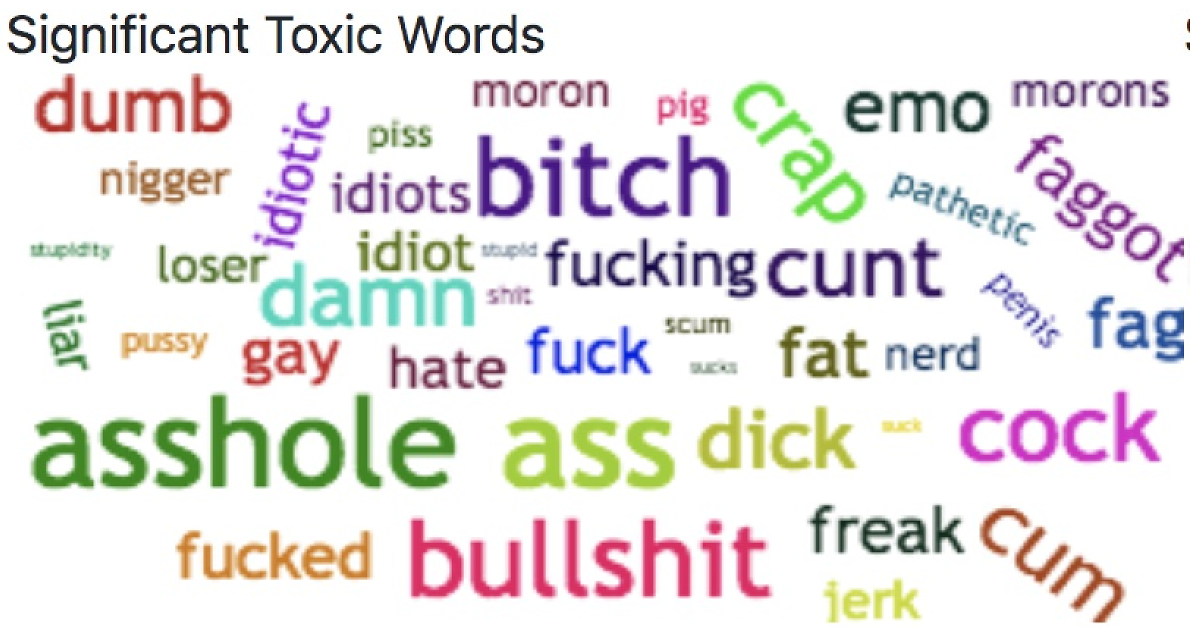


Part 3

3.2

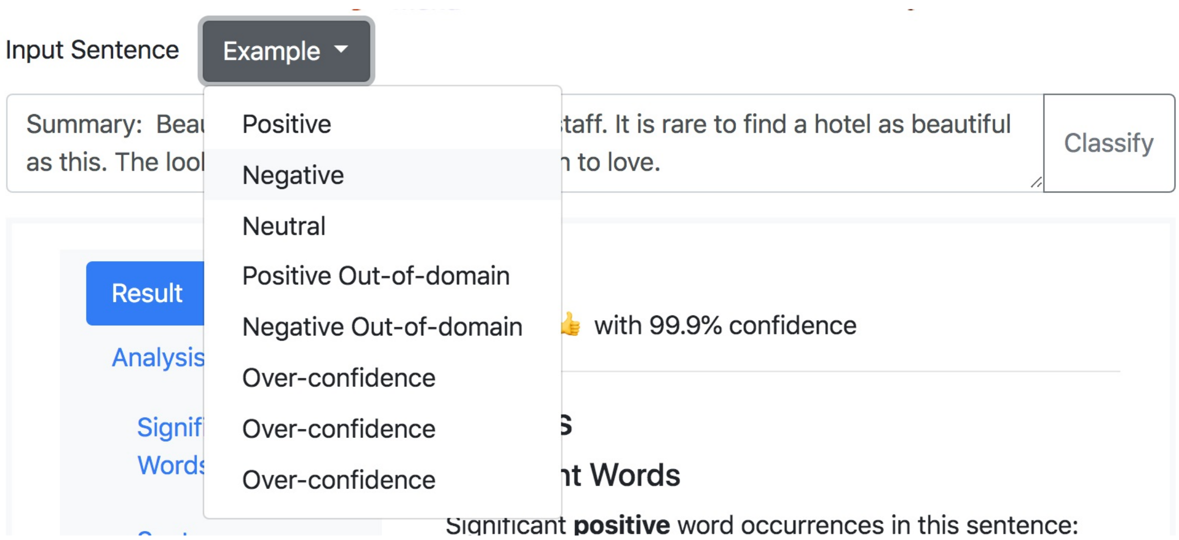
WordCloud Demonstration

WordCloud clearly shows the most significant words in the whole corpus. At the same time, the different size of each word represents the extent of significance, which helps the users understand in a direct manner.

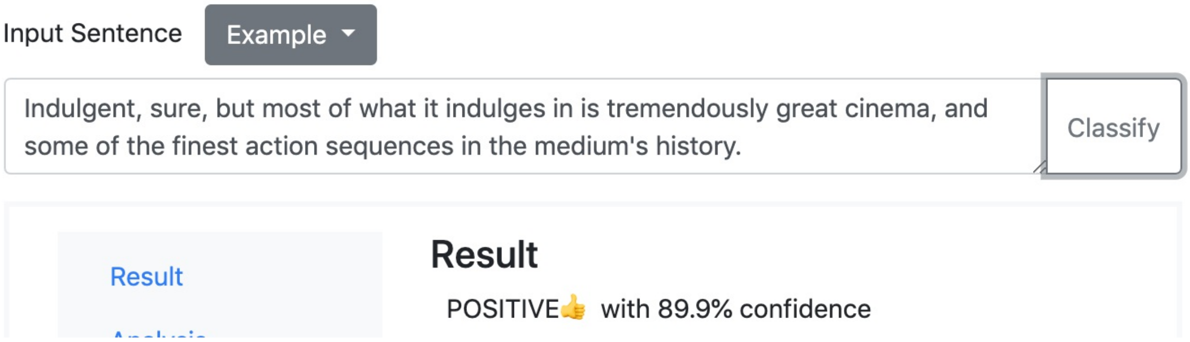


Easy Example Selection

We provide example options for users who are not familiar with or unsure of what kind of sentence can be passed into our model. Users tend to input only simple sentences, and by providing example instances, they can be impressed by how our classifier can cope with sentences that are much more complicated.



We also provide some out-of-domain sentences retrieved from Rotten tomoatoes, which is a totally different domain from restaurant comments. With the result, we find that our model is able to classify it correctly, showing that our model works pretty well on performing out-of-domain tasks as well.



Sentence Vector Visualization

In order to help users understand how data points are distributed in the data space, we provide a visualization produced by applying dimensionality reduction on our datasets using Linear Discriminant Analysis (LDA) technique. LDA tries to maintain the distance between each pair of data points, as well as maximize the distance between groups of different labels.

X coordinates represent the values of vectors after LDA. The farther a point is away from another on the x axis, the farther it is away in the original space. Y coordinates show the confidence of our predictions. As you can see, the farther a positive sentence is away from negative sentences, the higher its prediction is.

This graph gives users a rough but easy-understanding sketch of how our classifier is able to classify an instance based on its sentence vector.