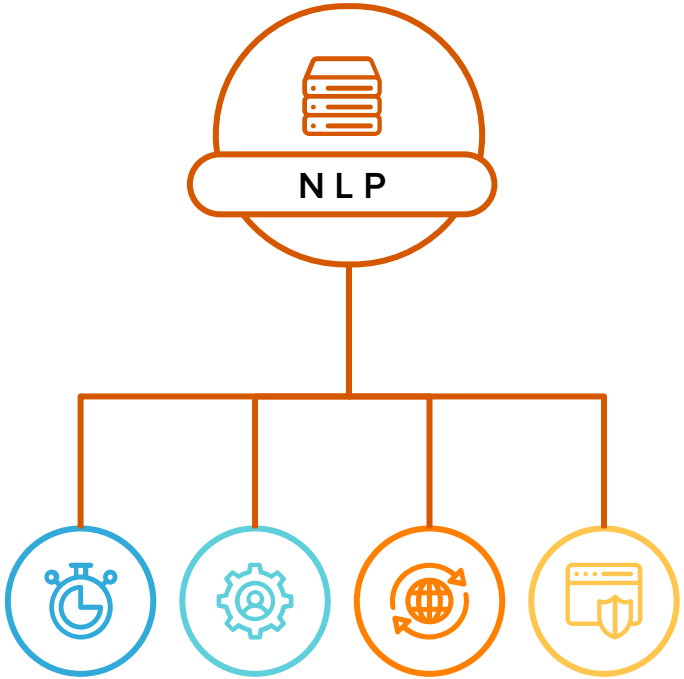
IDS 566 Advanced Text Analytics

Final Project



Team 1:

Evelyn Huang (UIN: 655772191)
Hoang Nha Nguyen (UIN: 671491808)
Anusha (UIN:670194968)

Content

For this final project, we will go through the questions in both mini project one and two.
And will Focusing more on the “Application” Part

- **General Context** (taught in the lecture)
  - Mini Project 1: N-gram + smoothing
  - Mini Project 2: WSD procedure

- **Our Open-ended Application**
  - The Apple Store App Review Organizer

General Context

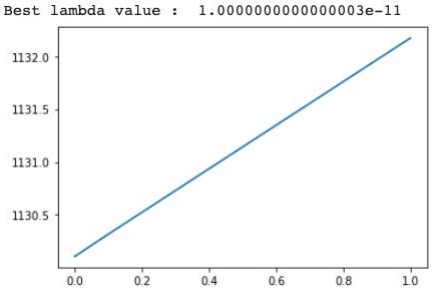
Mini Project 1

ngram+smoothing

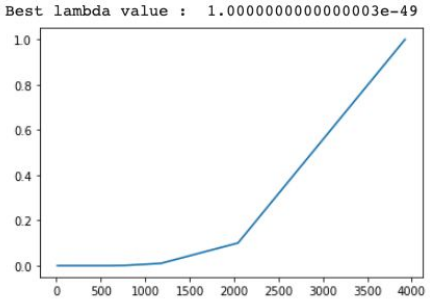
Perplexity Performance in different model

Train on	Trn	Trn	Trn+Dev
Smoothing?	**No**	**Yes**	**Yes**
Test on	test data	developing data	test data
Unigram	inf	1130.11	1148.50
Bigram	inf	11.86	7.505
Trigram	-	-	1.455e-13

Unigram



Bigram



Mini Project 1

ngram+smoothing

Bigram	**Sentence 2 :** Were you " . **Sentence 4 :** This gives a 280-yard drive it must have a pint of his associates Jewishness with the patriot in our capabilities " in great problem. **Sentence 5 :** She sounded like `` because excess egalitarianism , and unless the United Nations .
Trigram	**Sentence 1 :** Friend is off to the Congress . **Sentence 3 :** Cover the whole earth with peas , about that item anyway . **Sentence 4 :** To settle this slight , wiry frame , which often come up . **Sentence 5 :** Despite Giffen's warning , the monomer , and the good one) , storage (piers) and Gallet (chap. 14) .

Mini Project 2

Problem 1: supervised WSD

1. **Probability of each sense**

2. **Model** **Performance**

- In our model, the process of deciding model parameters becomes training. we have to train a separate model per each target word in the training data.

- Train the model using the training data and predict the model to obtain the **accuracy = 84.89%**

3. **Predict the model on test data**
   -predict the senses

4. **Use add-lambda smoothing (λ = 0.0001)**
   Accuracy with lambda smoothing = 0.8478027867095391

	target_word	vocab	sense	vocab_count	p(f/s)	p(s)
0	affect	that	1	7	0.038889	1.0
1	affect	that	2	0	0	0.0
2	affect	that	3	0	0	0.0
3	affect	that	4	0	0	0.0
4	affect	that	6	0	0	0.0
...	...	...	...	...	...	...
1750	affect	my	11	0	0	0.0
1751	affect	my	13	0	0	0.0
1752	affect	my	14	0	0	0.0
1753	affect	my	15	0	0	0.0
1754	affect	my	12	0	0	0.0

```
unique_word_sense_count

{'affect': {1: 45,
  2: 0,
  3: 0,
  4: 0,
  5: 0,
  6: 0,
  7: 0,
  8: 0,
  9: 0,
  10: 0,
  11: 0,
  12: 0,
  13: 0,
  14: 0,
  15: 0},
 'allow': {1: 98,
  2: 10,
  3: 0,
  4: 0,
  5: 0,
  6: 0,
  7: 0,
  8: 0,
  9: 0,
  10: 0,
  11: 0,
  12: 0,
  13: 0,
  14: 0,
  15: 0},
 'announce': {1: 87,
  2: 1,
  3: 0,
  4: 0,
  5: 0,
  6: 0,
  7: 0,
  8: 0,
  9: 0,
  10: 0,
  11: 0,
  12: 0,
  13: 0,
  14: 0,
  15: 0},
```

Mini Project 2

Problem 2: Ontological WSD (Simple Lesk algorithm)

1. Design a **metric** that rewards consecutive overlaps

   We decide to build bigram and trigram model for the “gloss” of each sense in the dictionary XML file and test it whether considering consecutive words or longer consecutive will give better outcome.

2. Implement a dictionary-based WSD system (consider consecutive words)

   Test bigram and trigram model on **validating** data:

   Accuracy of bigram model = 0.0
   Accuracy of trigram model = 0.0

   Test bigram and trigram model on **training** data:

   Accuracy of bigram model = 0.0
   Accuracy of trigram model = 0.0

```
print(df_predict_tri)
```

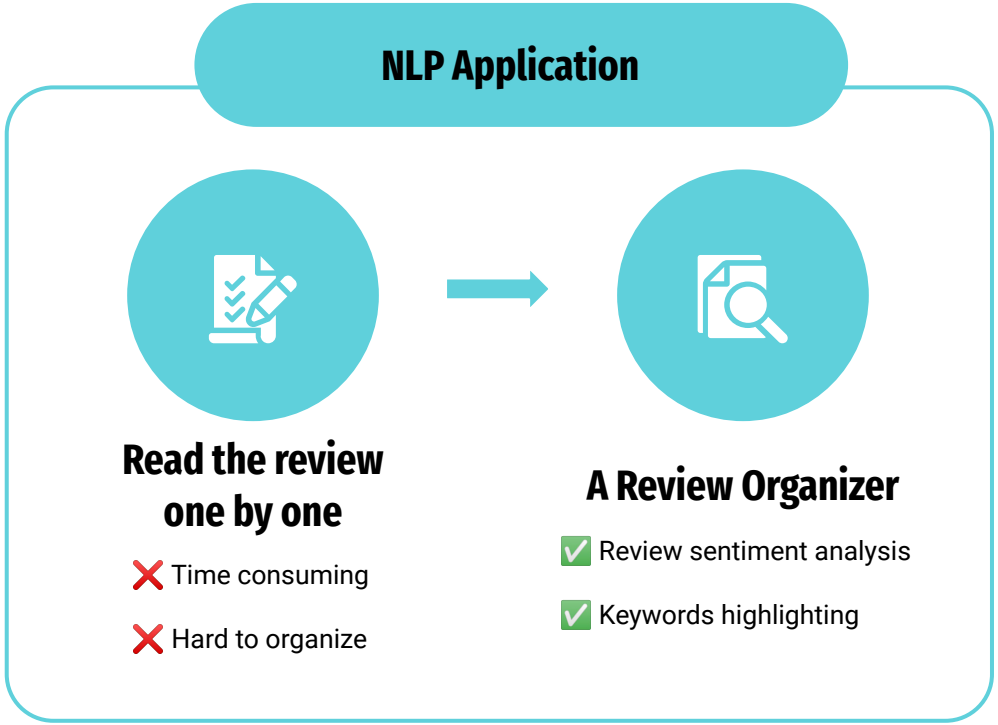
	Given	Predicted
0	capital.n.1	capital.n.0
1	capital.n.1	capital.n.0
2	capital.n.1	capital.n.0
3	capital.n.1	capital.n.0
4	capital.n.1	capital.n.0
..	...	...
928	keep.v.1	keep.v.0
929	keep.v.4	keep.v.0
930	keep.v.1	keep.v.0
931	maintain.v.1	maintain.v.0
932	maintain.v.1	maintain.v.0

Our Open-ended Application

Application Idea

- The mobile application market is highly competitive.
- To find the niche in your category, competitor research is a must!
- G2.com

Application Idea



How it works?

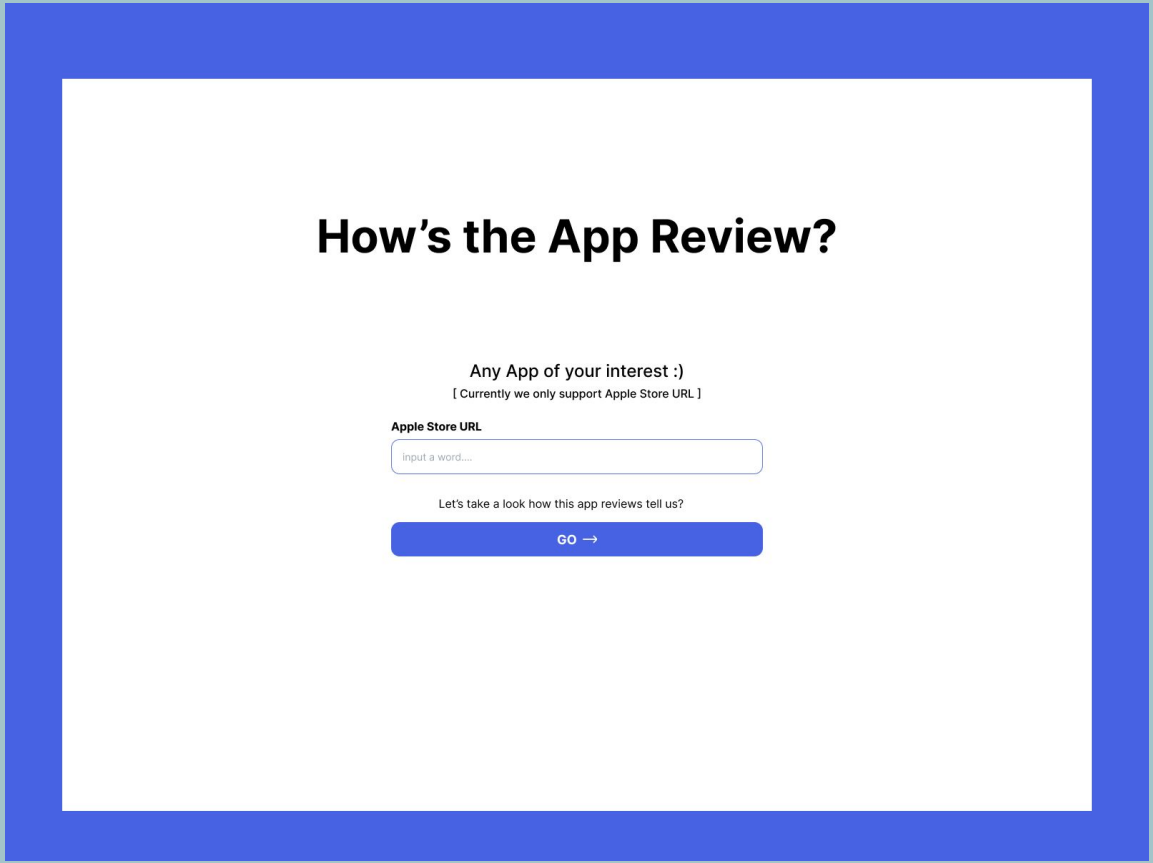
Scrap the review from Apple Store

Tokenize, Normalize, and Lemmatize the Text

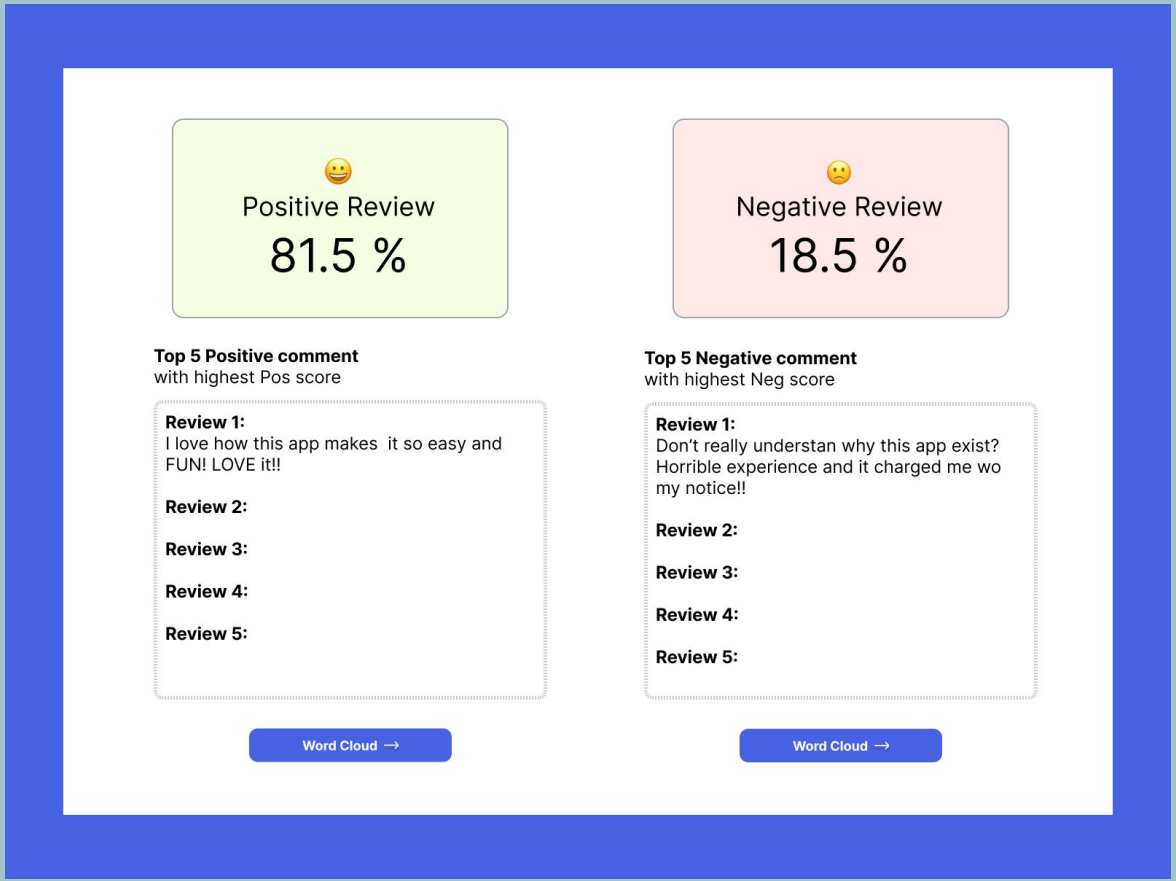
WordNet Sentiment Analysis

Categorize Reviews to showcase result !

Application UI design



Application UI design



Application UI design

←

😀 Positive Review

😕 Negative Review

Common Words in Positive Reviews

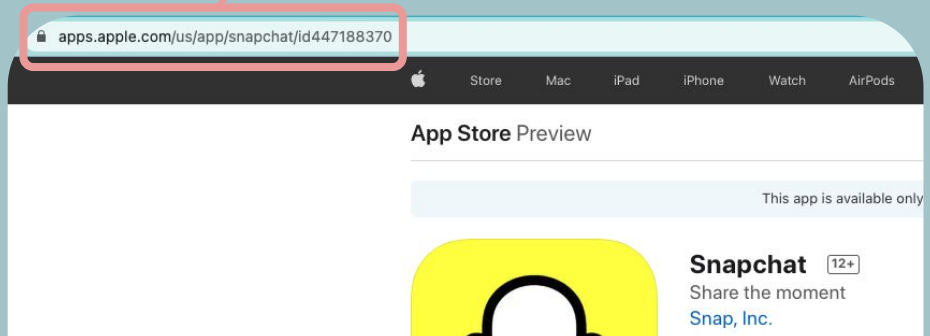
Common Words in Negative Reviews

Search Another App

Application Demo

THE APP REVIEW ORGANIZER

Apple Store App Link of your choice 🌷 :
(tab "Enter" to see the result ✨)



Thank you for your listening