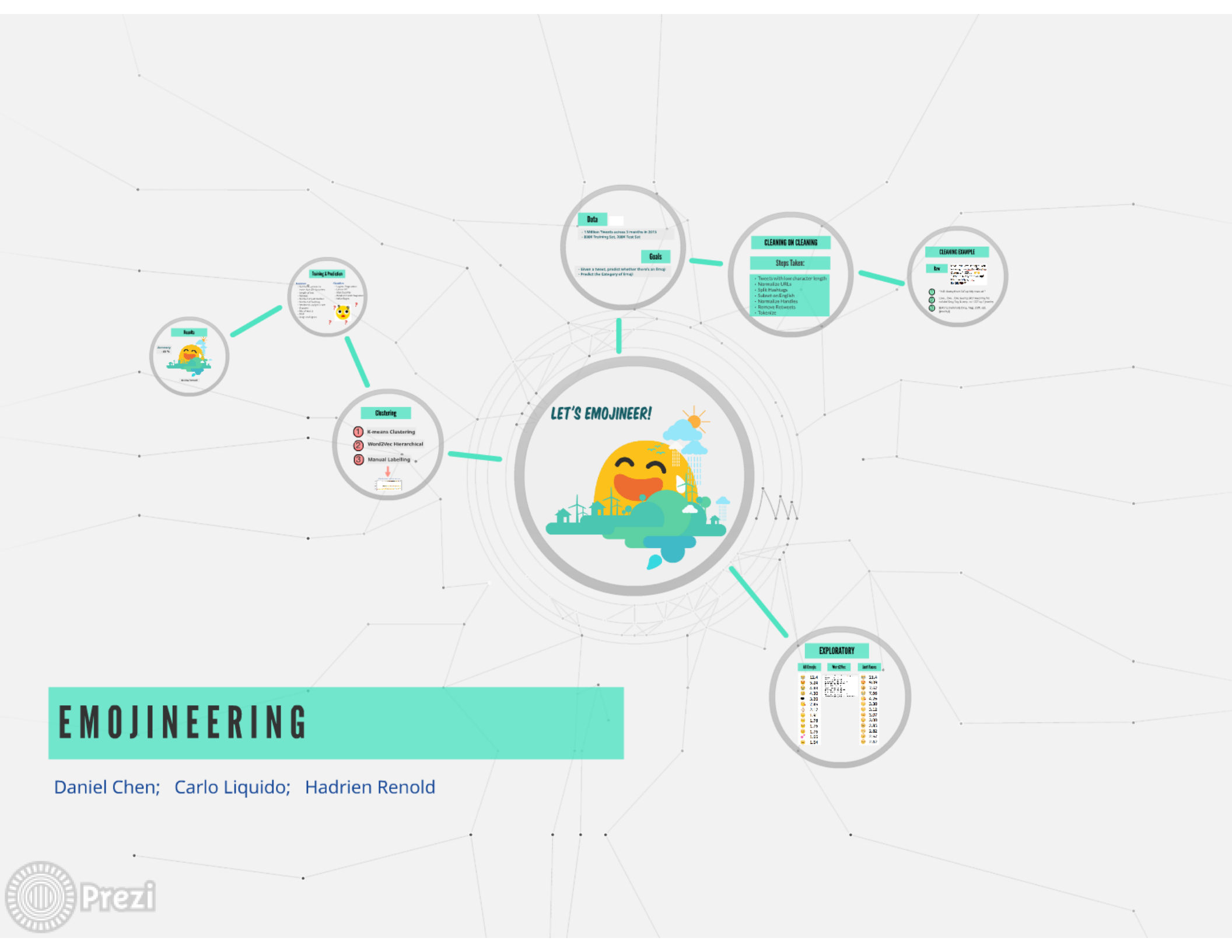
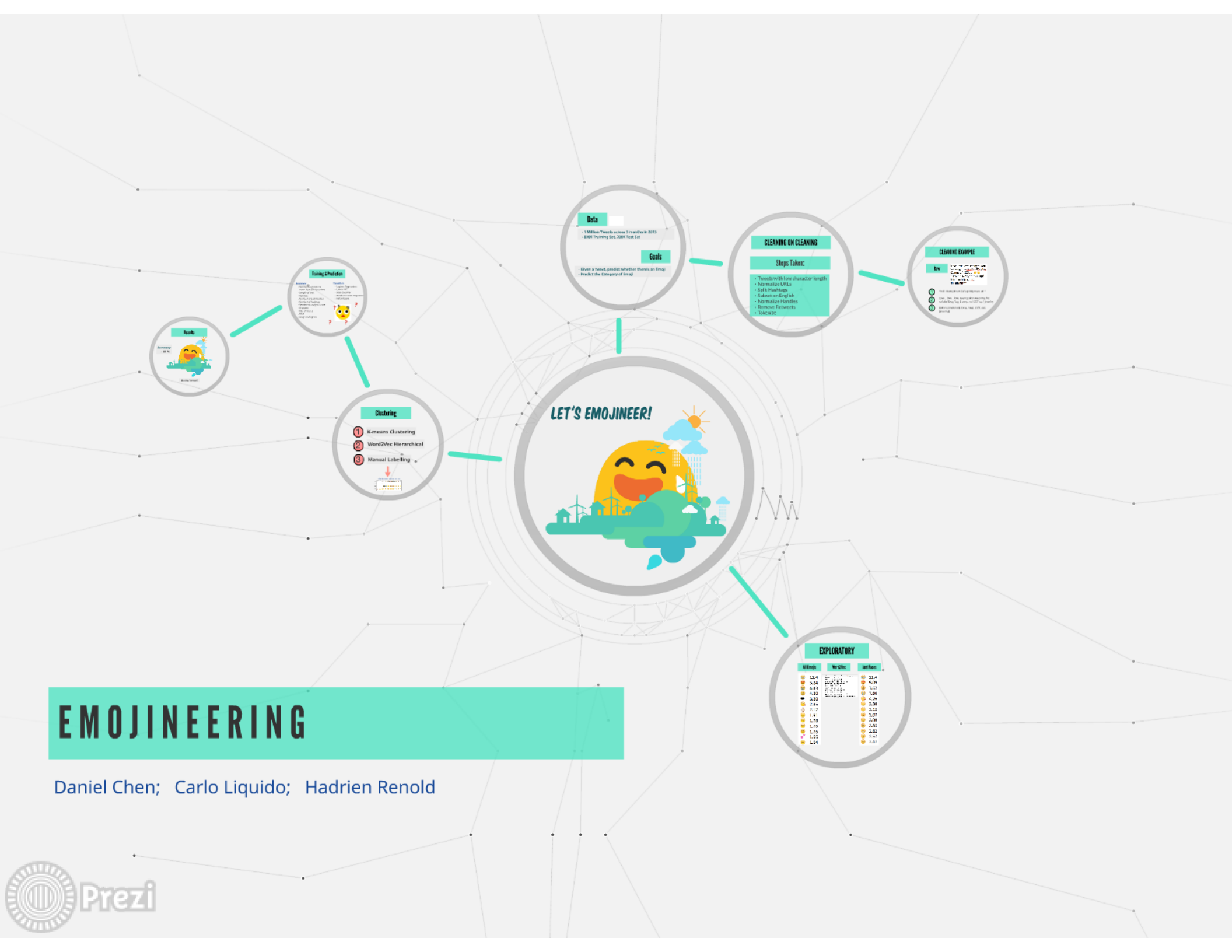


EMOJINEERING

Daniel Chen; Carlo Liquido; Hadrien Renold



EMOJINEERING

Daniel Chen; Carlo Liquido; Hadrien Renold

LET'S EMOJINEER!

Data

- 1 Million Tweets across 3 months in 2013
- 800K Training Set, 200K Test Set

Goals

- Given a tweet, predict whether there's an Emoji
- Predict the Category of Emoji

CLEANING ON CLEANING

Steps Taken:

- Tweets with low character length
- Normalize URLs
- Split Hashtags
- Subset on English
- Normalize Handles
- Remove Retweets
- Tokenize

CLEANING EXAMPLE

Raw

Love, love, love seeing #Akon wearing his #Nalukai #DogTag & w/ #ZZTop! 😍 😍 “@Akon: Dusty from ZzTop! My man http://t.co/5biGfDRBH7”

1. “ hdl : Dusty from ZzTop! My man url ”
2. Love , love , love seeing akon wearing his nalukai Dog Tag & amp ; w / ZZT op ! jewelry
3. [[akon], [nalukai], [Dog, Tag], [ZZT, op], [jewelry]]

Love, love, love seeing #Akon wearing his #Nalukai #DogTag & w/ #ZZTop! 😍 😍 “@Akon: Dusty from ZzTop! My man http://t.co/5biGfDRBH7”

& w/ #ZZTop! 😍 😍
“@Akon: Dusty from ZzTop!
My man http://t.co/
5biGfDRBH7”

1. “ hdl : Dusty from ZzTop! My man url ”
2. Love , love , love seeing akon wearing his nalukai Dog Tag & amp ; w / ZZT op ! jewelry
3. [[akon], [nalukai], [Dog, Tag], [ZZT, op], [jewelry]]

EXPLORATORY

All Emojis

😂	12.4
😍	5.24
😭	4.43
😃	4.30
❤	3.20
😘	2.45
👌	2.17
😞	1.91
😏	1.78
😒	1.76
😊	1.76
💕	1.66
😩	1.64

Word2Vec

```
emoji_model.most_similar(positive = ['😻'], topn = 20)
```

```
[('cmmmmng', 0.7462140321731567),
 ('#daad', 0.6925972700119019),
 ('maagan', 0.6692696809768677),
 ('lmaoooooooooo', 0.6570378541946411),
 ('inseparable', 0.6502166986465454),
 ('#lrt', 0.6376644372940063),
 ('#playa', 0.6227041482925415),
 ('#dying', 0.5953817367553711),
 ('#aika', 0.5952829122543335),
 ('#coldworld', 0.5948764681816101),
 ('#blama', 0.5885862112045288),
 ('#icant', 0.5870974063873291),
 ('#workaholics', 0.5856915712356567),
 ('laffin', 0.5782320499420166),
 ('hahahahahahahahahahahaha', 0.5772393345832825),
```

Just Faces

😂	21.4
😍	9.09
😃	7.67
😭	7.66
😘	4.26
😞	3.30
😏	3.12
😊	3.07
😒	3.00
😩	2.85
😁	2.82
😉	2.57
😔	2.42

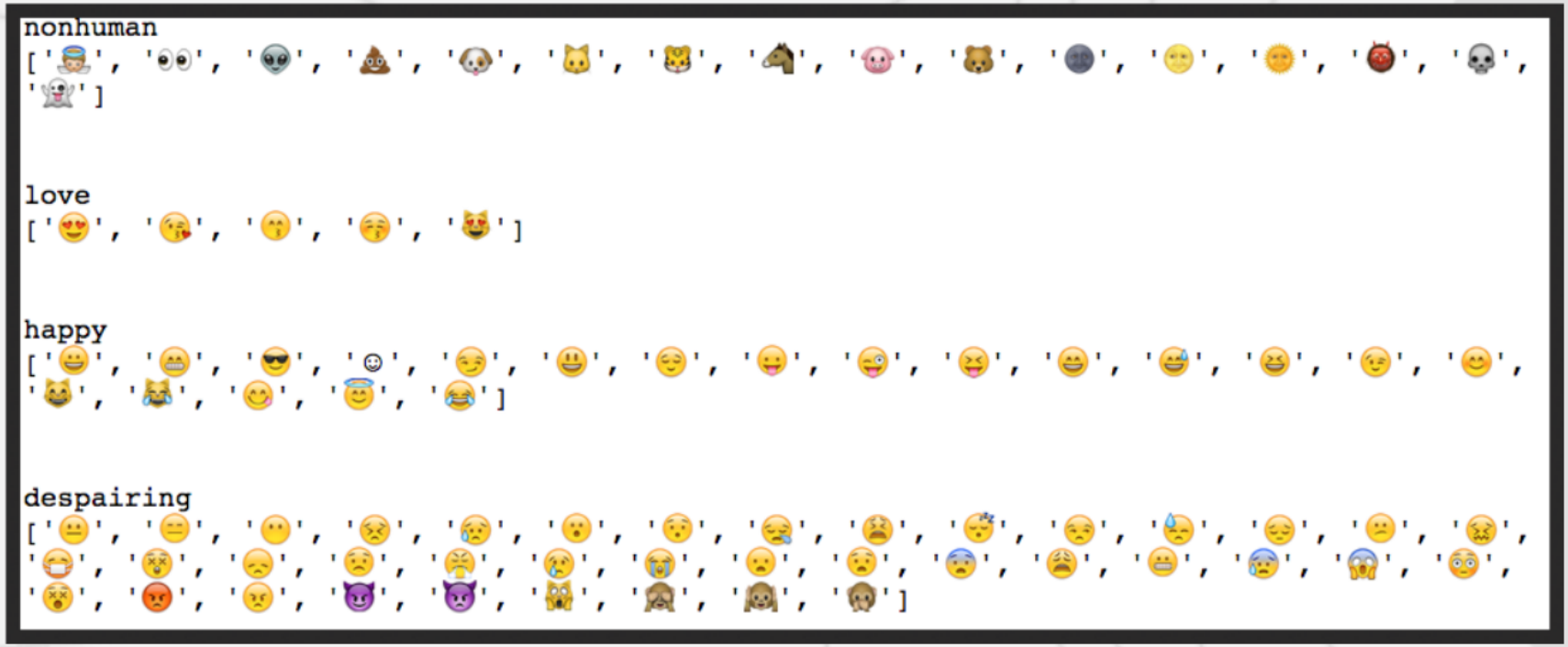
```
emoji_model.most_similar(positive = ['😂 '], topn = 20)
```

```
[('ommmmmg', 0.7462140321731567),
 ('#dead', 0.6925972700119019),
 ('meagan', 0.6692696809768677),
 ('lmaooooooooooo', 0.657037854194641 1),
 ('inseparable', 0.6502166986465454),
 ('#lrt', 0.6376644372940063),
 ('#playa', 0.6227041482925415),
 ('#dying', 0.5953817367553711),
 ('#sike', 0.5952829122543335),
 ('#coldworld', 0.5948764681816101),
 ('#blama', 0.5885862112045288),
 ('#icant', 0.5870974063873291),
 ('#workaholics', 0.585691571235656 7),
 ('laffin', 0.5782320499420166),
 ('hahahahahahahahahahahaha', 0.5772393345832825),
```

Clustering

1. K-means Clustering
2. Word2Vec Hierarchical
3. Manual Labelling

~ 68% of all tweets non-conflicting Face emojis



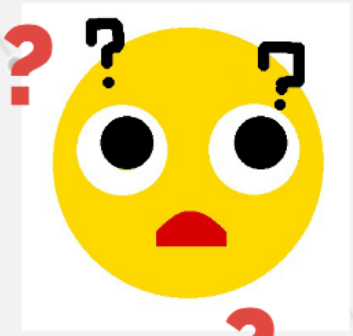
Training & Prediction

Features:

- Normalize on tweets more than 40 characters
- Length of text
- Retweet
- Number of punctuation
- Number of hashtag
- Sentiment analysis score
- Q-grams
- Bag of words
- Tfidf
- Unigram/bigram

Classifier:

- Logistic Regression
- Linear SVC
- XGB Classifier
- Random Forest Regressor
- Naïve Bayes

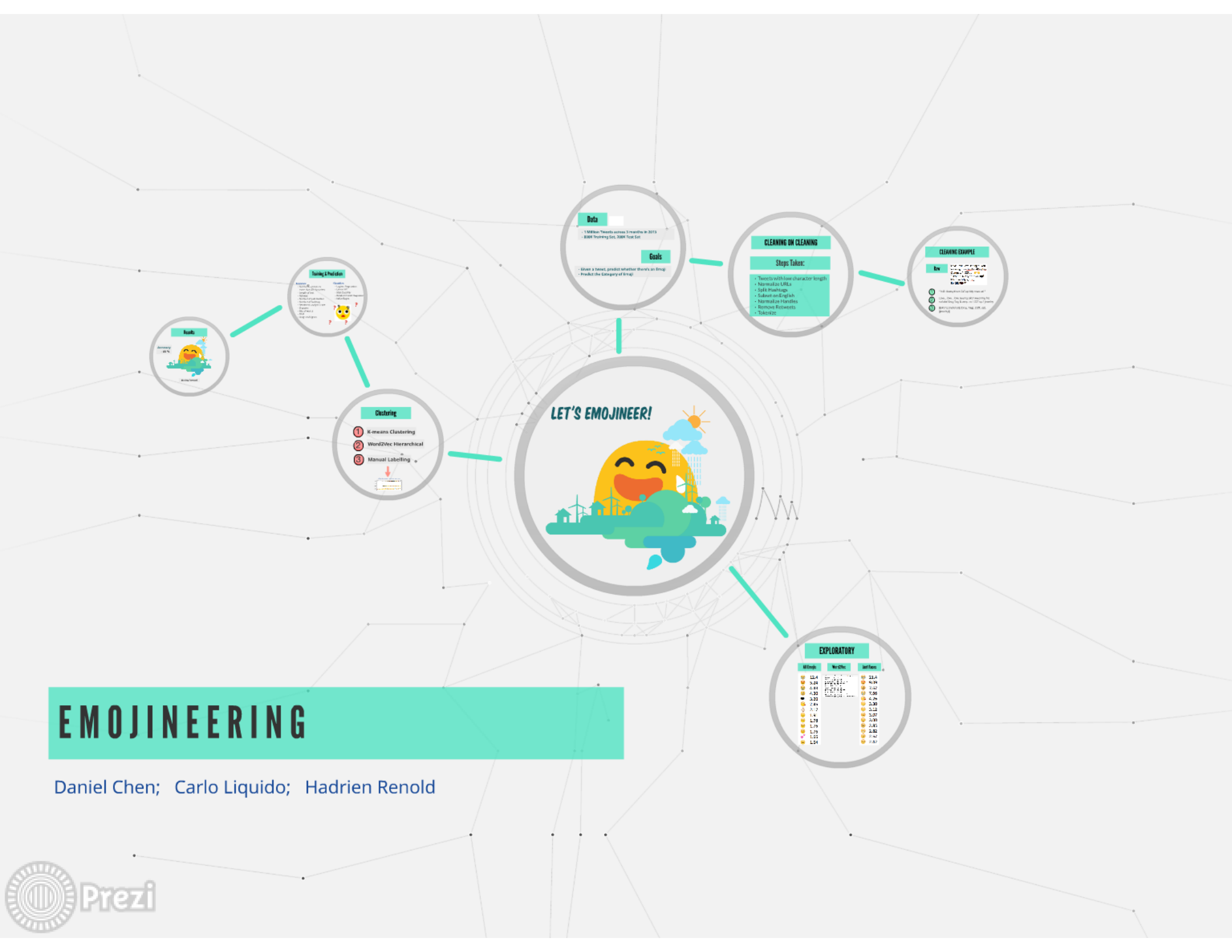


Results

Accuracy

- 85 %

Moving Forward



EMOJINEERING

Daniel Chen; Carlo Liquido; Hadrien Renold