Yelp Review Tourist Classifier:
an NLP System
Methods in Computational Linguistics

Tyler Wilbers

Outline

- The Goal
  - Create NLP system that can classify whether or not a Yelp review written in English was written by a local or visitor to the region.
- Creating the Corpus
- Training the classifier
  - Feature Extraction
  - Increasing Accuracy
- Results

The Goal

1/14/2016

1 check-in

Although I sense danger, for there's now a bar downstairs; I'm pretty stoked about about family owned bar with good taps, Whiskey drinks, food and atmosphere--finally. We've been here for years and this is the first decent chill spot close to the Gates and Kosciusko stop. B52 bus is a block away. The twins (no you weren't that drunk) and a Chicagoan bartender were super abputwayttentive and the pretzel dude was off the chain.

Was this review ...?

Useful 3 Funny 1 Cool

Local

6/10/2014

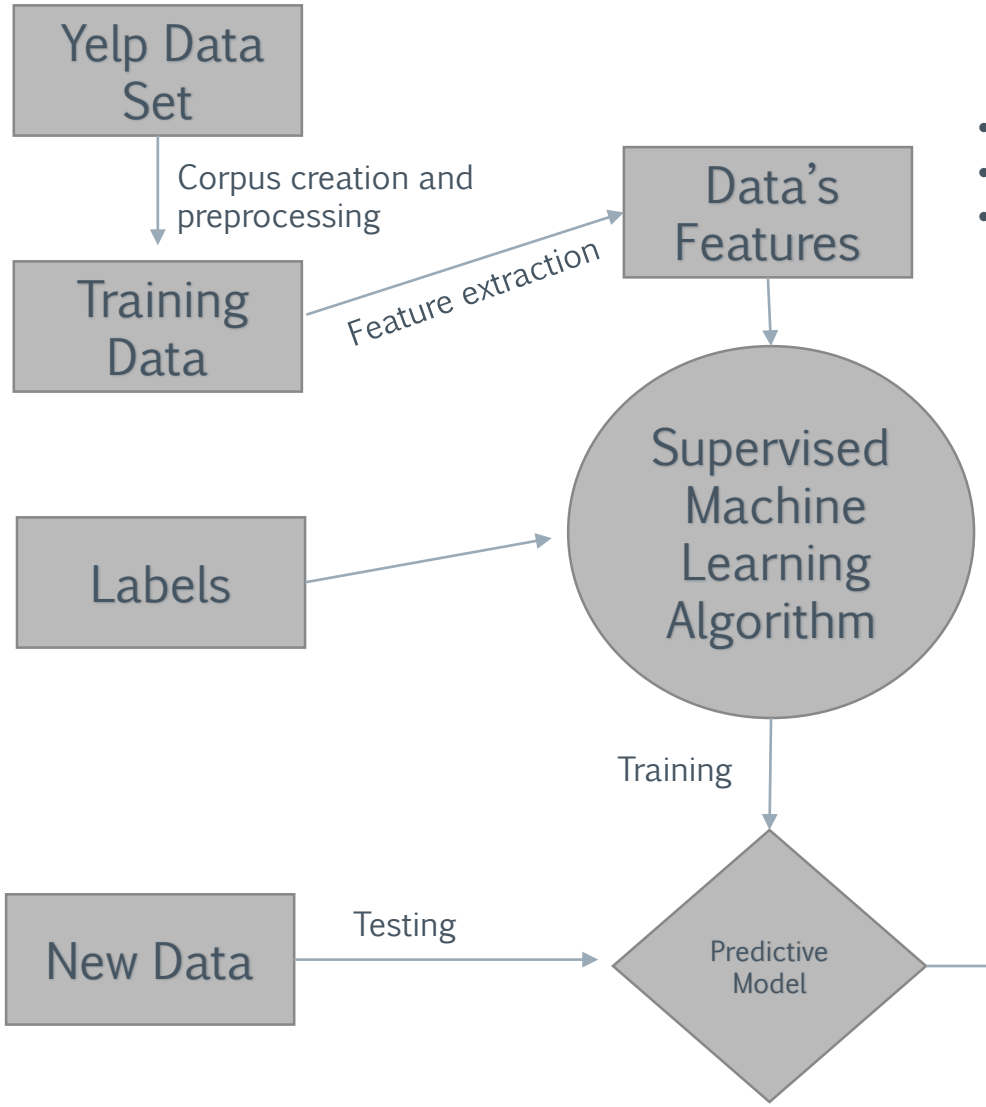
We entered the restaurant at 1:30 on a Sunday and it was pretty packed. We were still seated quickly though and service was perfect. I can't wait for my next trip to south Georgia so I can stop at Bay South for lunch.

Was this review ...?

Useful 4 Funny Cool

Remote

The Goal



- Tokenizer = nltk.word_tokenize()
- N-grams = sklearn.feature_extraction.text.CountVectorizer()
- POS tagging = nltk.tag.stanford.StanfordPOSTagger()

- LR = sklearn.linear_model.LogisticRegression()
- NB =sklearn.naive_bayes.GaussianNB()
- SVM = sklearn.svm.SVC()

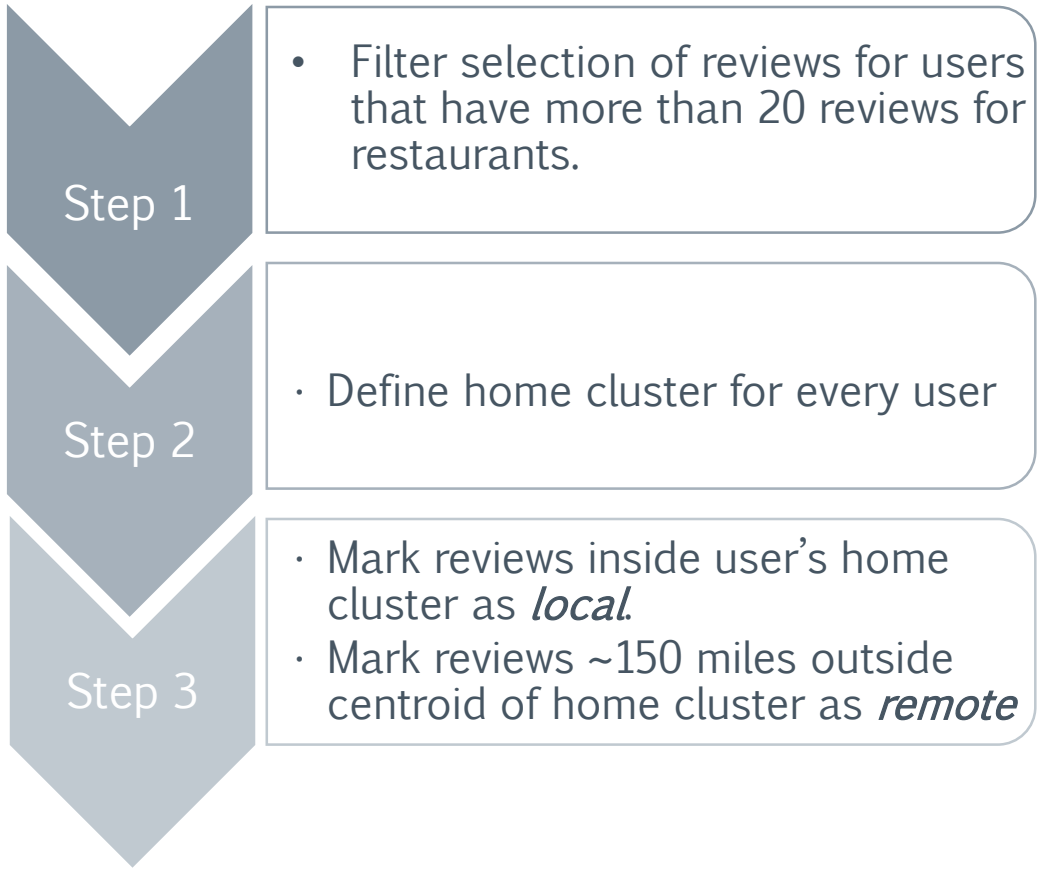
The Corpus

- I used the data provided by the Yelp Dataset Challenge to construct a corpus:
  - Their data set included 2.2M reviews by 552K users for 77K businesses.
- Two problems:
  - The Yelp Dataset does not include the locations of users (only businesses).
  - This data set is entirely too large for my project.
- I approached these problems in tandem to create a corpus of yelp reviews with proper labels to train a supervised model.

Constructing Training Labels

› The Yelp Dataset does not include the locations of users (only businesses).

› So I used the following algorithm to find the home cluster for ever user:



Home Cluster Algorithm

- Every review has a geo-point (latitude, longitude).
- For every user *u*, for every review *r* written by *u*, add a proximity counter for *r*:
  - This counts the number of reviews written by *u* that are in either 1 degree longitude or 1 degree latitude from *r*.
  - This is a distance of ~60 miles max.
- Ever user *u*, now has a review with a geo-point *g* with the highest proximity counter.
  - If the proximity counter is greater than half of their total number of reviews, set their home as *g*.
  - The home cluster is every point within 1 degree of latitude or longitude from *g*.
- If a review by *u* is written within 1 degree of latitude or longitude from *g*, the review is marked as local and added to the corpus.
- If a review is ~150 miles from *g*, mark those review as remote and add it to the corpus.

The Final Corpus

- The end result was corpus of 2840 reviews with 935 distinct users,1266 business, and 73 cities.
  - Test Set

    A random sample of 2130 of these reviews equally split between local and remote would be dedicated to a training set.
  - Train Set:

    A random sample of 719 reviews would be used for a test set.

Yelp Review Metadata Features

- City of review:
  - Categorical variable that ranges over every city in the corpus.
  - Reason: reviews written in certain cities are more likely to be local/remote. (e.g., Paris Texas will not have as many remote review as New York).

- Week:
  - Categorical variable that rages over ever week in the year.
  - Reason: reviews written during certain times are more likely to be remote.

Linguistic Features: City Mentioned

- Location mentions:
  - Binary feature based on whether the location of the business is mentioned in the review.

  Prediction:
  - Local reviews are less likely to mention the city the review is in because they see it is more implied knowledge.

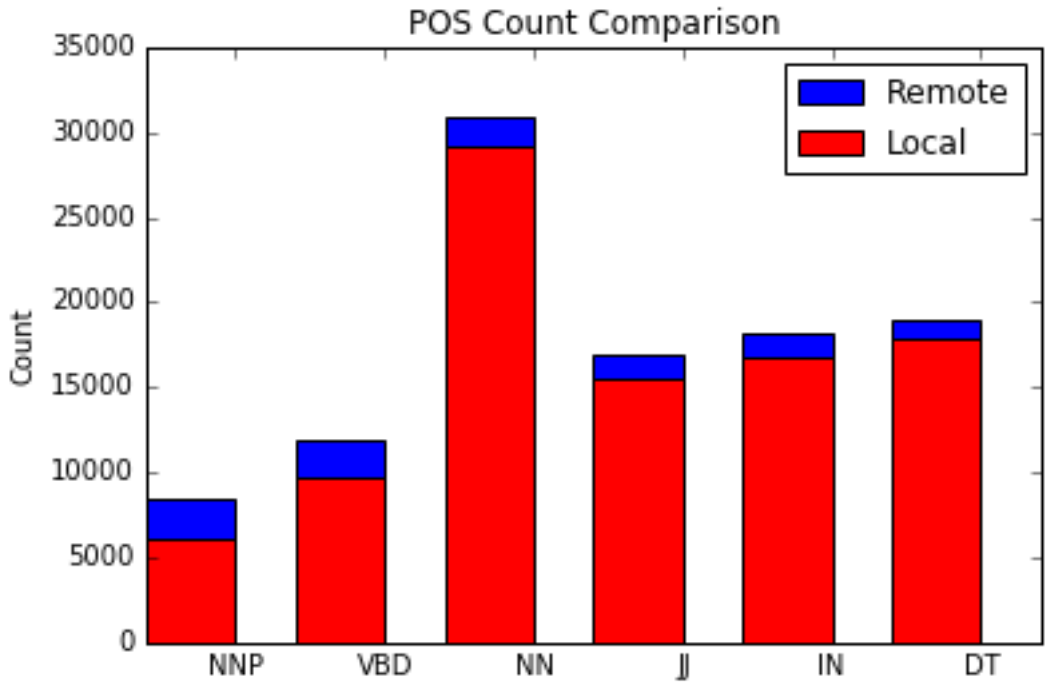
  Results:
  - Local reviews: 12% mention the city
  - Remote reviews: 16% mention the city

Linguistic Features: Length

- Review length:
  - Character count of the review.

  Results:
  - The average character count of remote reviews is %7 larger than the average character count of local reviews.

Linguistic Features: Tense, Aspect, and POS



Linguistic Features: Tense, Aspect, and POS

- Proper Noun Count (NNP, NNPS)
- Noun Count (NN, NNS)
- Preposition Count (IN)
- Tensed Verbs Count
  - Count Past Participle (VBN)
  - Count Simple Past (VBD)
  - Count Simple Present (VBP, VPZ)
- Adverb count (RB, RBR, RBS):
  - The number adverb occurrences in a review.

Introduce Concepts

› Decision (Adda et al., 1998):

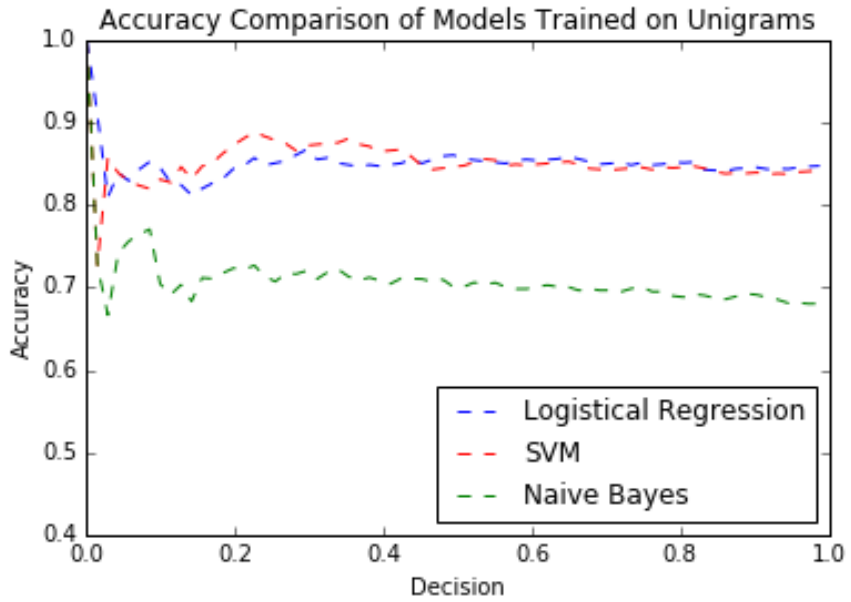
$$decision = \frac{N(\text{retrieved documents})}{N(\text{all documents})}$$

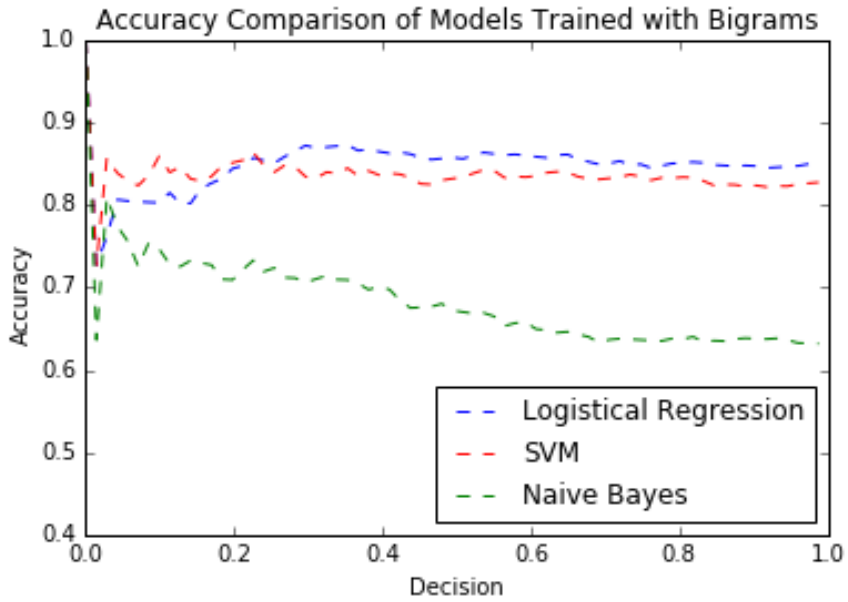
› Accuracy (Manning and Schütze, 1999):

$$accuracy = \frac{N(\text{correct classifications})}{N(\text{all classifications})}$$

Linguistic Features: N-grams

› Preliminary testing showed that bigrams and unigram models yielded similar results across multiple ML algorithms:





Increasing Accuracy

- To discriminate common n-grams I used Pak's (2010) strategy of introducing a salience threshold:

$$salience(g) = \frac{1}{N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} 1 - \frac{\min(P(g|s_i), P(g|s_j))}{\max(P(g|s_i), P(g|s_j))}$$

- Suppose that a n-gram occurs twice as often in remote reviews.
  - The salience measure for that n-gram would be the one minus the sum of probability distribution for local reviews over the sum of the probability distribution for remote reviews (i.e. 0.5).

Increasing Accuracy

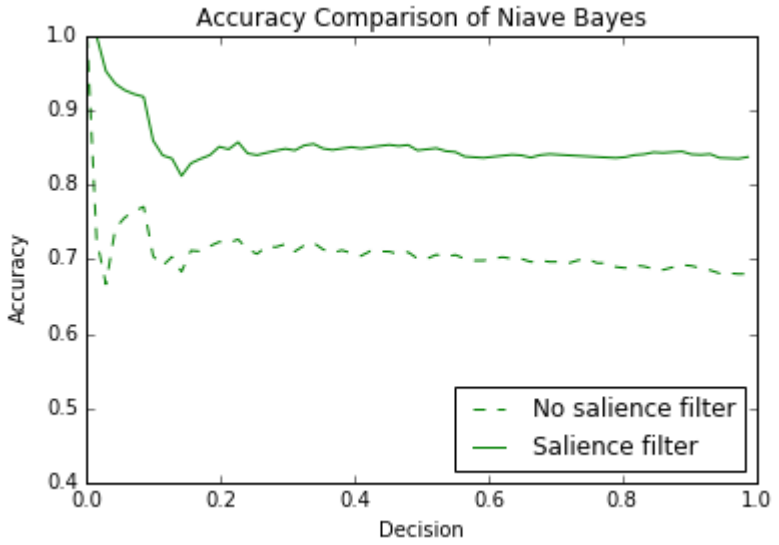
› The following are some examples of unigrams with high salience:

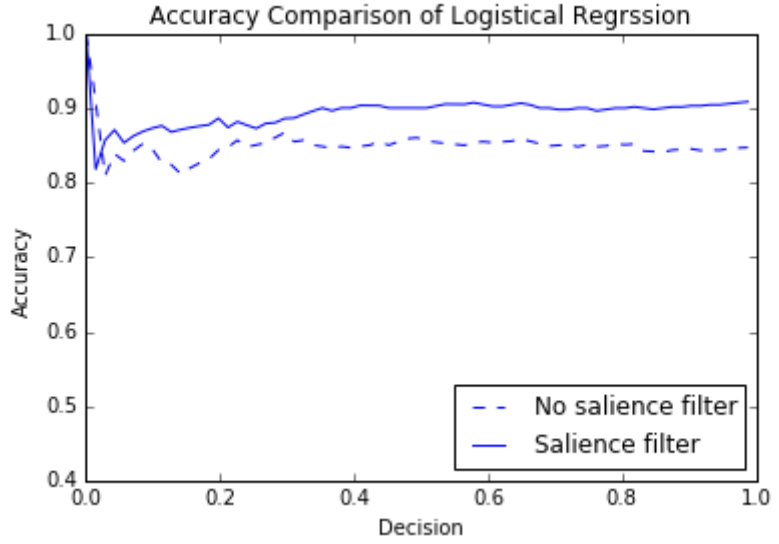
unigram	Salience
wedding	.909
hotel	.967
coupons	.909
golf	.941
tuesday	.875
staying	.939

bigram	Salience
charlotte airport	.857
sunday buffet	.857
time visit	.833
new york	.8
ranch dressing	.9
never bad	.818

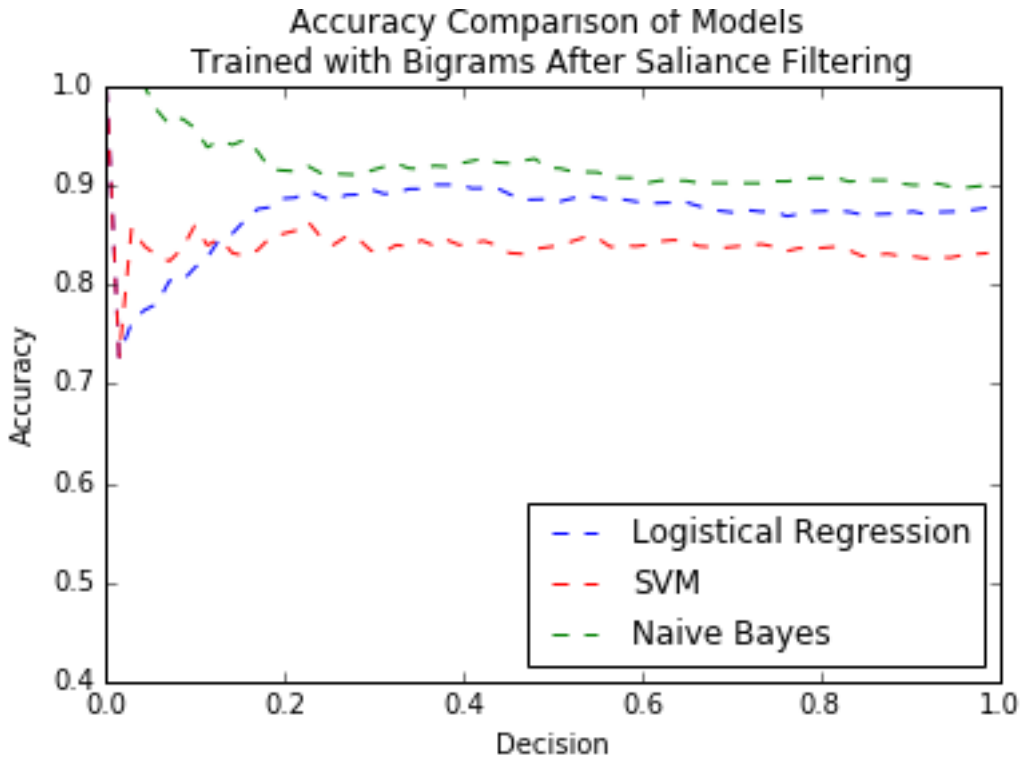
Unigram Results

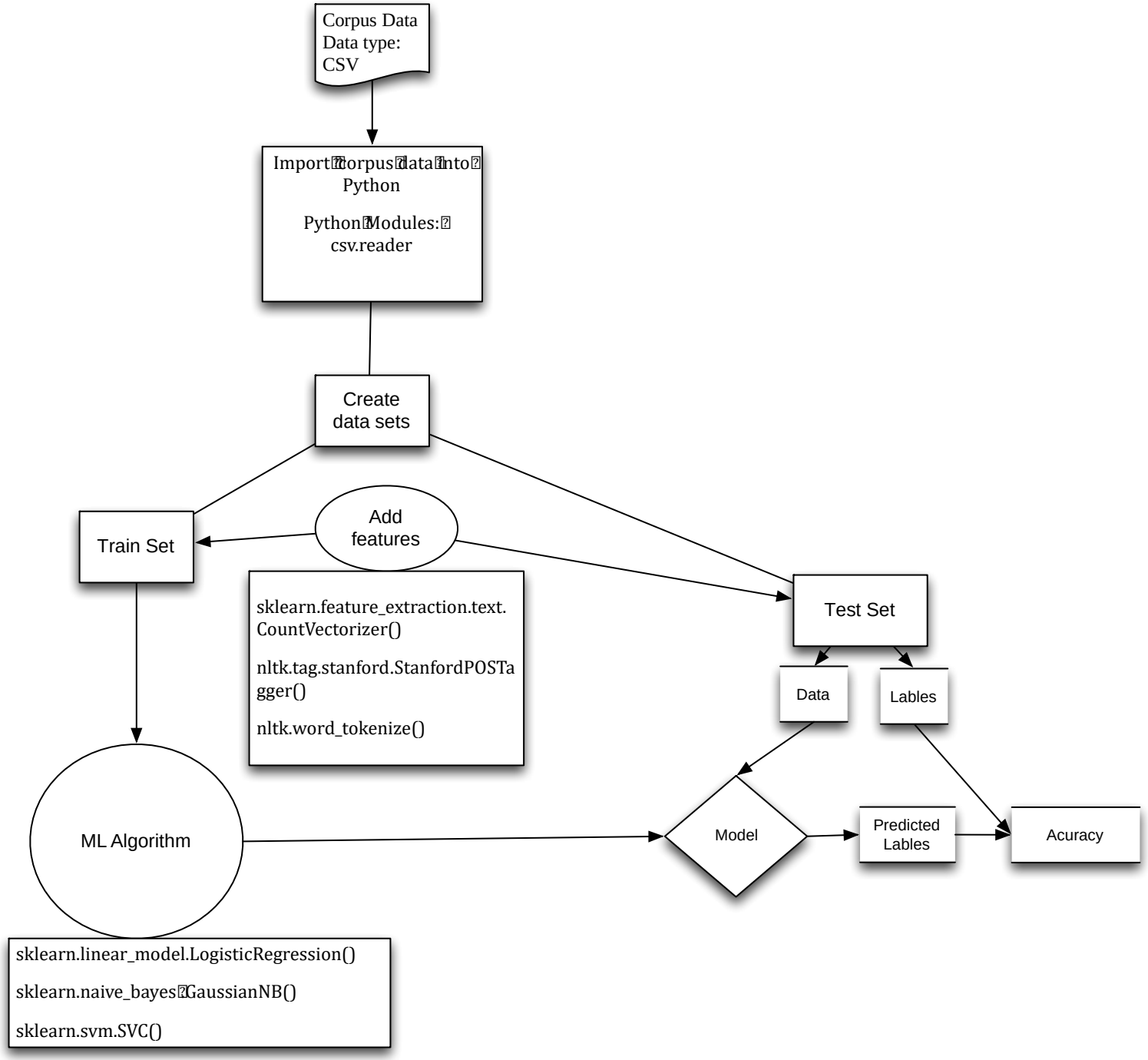
- By using a salience threshold, ϑ, I was able to eliminate common n-grams.
- Before filtering the average salience was .738 with a standard deviations of .362.
- Setting the ϑ *to.75* helped me to significantly improve accuracy.





Bigram Results




Corpus Data
Data type:
CSV
Import corpus data into Python
Python Modules:
csv.reader
Create data sets
Train Set
Add features
Test Set
sklearn.feature_extraction.text.CountVectorizer()
nltk.tag.stanford.StanfordPOSTagger()
nltk.word_tokenize()
Data
Lables
ML Algorithm
Model
Predicted Lables
Acuracy
sklearn.linear_model.LogisticRegression()
sklearn.naive_bayes.GaussianNB()
sklearn.svm.SVC()

Citations

- G. Adda, J. Mariani, J. Lecomte, P. Paroubek, and M. Rajman. 1998. The GRACE French part-of-speech tagging evaluation task. In A. Rubio, N. Gallardo, R. Castro, and A. Tejada, editors, LREC, volume I, pages 433–441, Granada, May.
- Christopher D. Manning and Hinrich Schutze. 1999. Foundations of statistical natural language processing. MIT Press, Cambridge, MA, USA.
- Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).