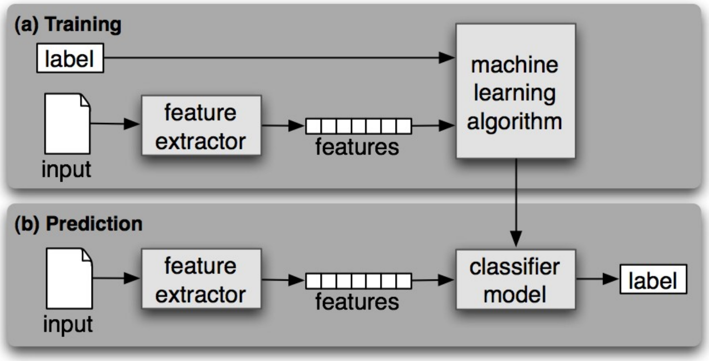
Delvison Castillo
Cog376
Sentiment Analysis (Final Project)

Summary

This is a sentiment analysis program that can classify a movie critique as either a positive or negative critique. The program is written entirely in python (version 2.7.3) and utilizes components of the Natural Language Toolkit.

Explanation



This program utilizes the method of supervised classification. The data set that was used for the training component is listed below. Initially, I had intended to use full critique examples taken from the web but the amount of irrelevant words within these critiques proved for very little accuracy. Instead, I chose to use brief one-line critiques as the training *input* simply because they were straight to the point and larger fuller critiques can be atomically comprised of these smaller ones. I start by reading my sample input, removing punctuation and any words that contain a length less than 3 from every live. This is mainly to get rid of neutral words such as 'it' and 'the'. The resulting words are then considered to be the *features*. I generate a training set by utilizing the `nltk.classify.apply_features` function with my *feature extractor* function and list of critiques as parameters. The *feature extractor* of this program takes in a tokenized review, checks what words it contains in relation to the features that were extracted from my sample data and returns a dictionary of this information. The dictionary essentially has boolean values of what words the tokenized review contains in relation to the training feature set. After I have my training set I generate my *classifier model* by utilizing the `nltk.NaiveBayesClassifier.train` function with my training set as a parameter. Once I have my classifier model, I am ready to start making predictions. I take a movie critique as an input, tokenize it, extract its features using my feature extractor and pass the list of features into the classifier model's classify function. The result is a *label* which in this case can be 'positive' or 'negative'.

Heres example of what the feature extractor of this program returns when considering a movie review:

Example: “It was a great movie.” --> (tokenize) --> ['It', 'was', 'a', 'great', 'movie'] → pass this set into feature extractor. The return resembles something like this:

```
{'contains(great)' : True,
  'contains(movie)' :True
  'contains(terrible)': False,
  (etc)...}
```

Data set used

Positive Critiques (positive_reviews.txt)	Negative Critiques (negative_reviews.txt)
loved it.	hated it.
a great time.	worst movie ever.
definitely watch it.	waste of time.
best movie of the year.	waste of money.
awesome film.	bad movie.
laughed out loud.	don't watch it.
extremely funny.	wouldn't recommend it.
two thumbs up.	very disappointing.
can't wait for the next one.	a complete let down.
great fun.	a failure.
it made me smile.	a failed attempt.
exceeded my expectations.	I was disappointed.
will watch it again.	it didn't live up to the hype.
the perfect movie.	crappy movie.
my favorite of all time.	worst ending.

Examples

In order to test the program I pulled random movie critiques from rottentomatoes.com. The following are examples that utilize 4 different critiques.

```
Enter a critique >>> Iron Man 3 was ultimately a disappointing experience. There
are a number of reasons, but the main issues included grandiose action, cop outs in
the plot, and a distancing from the comic itself.

negative

Enter a critique >>> A mindless, milling mess. Just what you would expect from
somebody schooled in the Michael Bay/Jerry Bruckheimer academy of bad directing.

negative

Enter a critique >>> "Toy Story 3" is a bit on the safe side, hemmed in by its
concept from offering the imagination that makes the best Pixar has to offer so
good ... but the filmmakers still understand real heart.

positive

Enter a critique >>> Full of riches, thematic and visual, and the best way to
understand them is simply to watch the film and take it all in.

positive
```

References

Chapter 6 of the NLTK book. (http://nltk.googlecode.com/svn/trunk/doc/book/ch06.html)