Sentiment Analysis of Real Time Tweets

Chinmay Parikh (A059)

Qais Makani (A029)

Yash Rathod (A042)

Recap

- Using Twitter API store the tweets in database
- Use the to perform Sentiment Analysis
- Store the file in the same database
- Use the file to display the Sentiments on a WebGL Globe

Suggestions

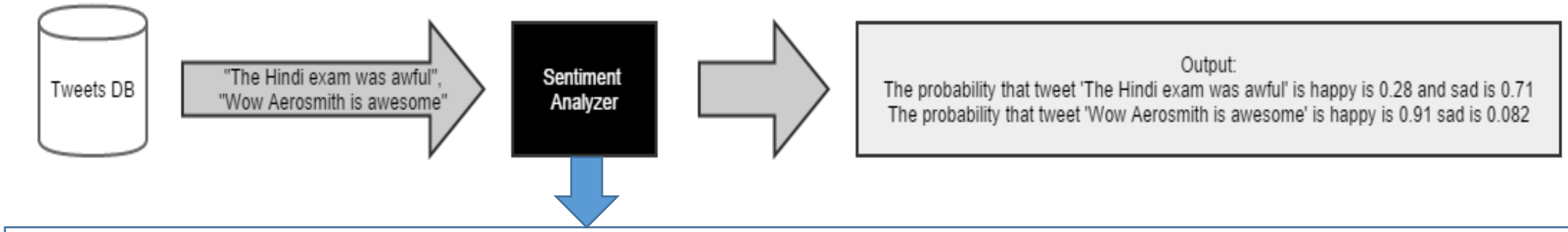
- Domain Specific or Product Specific
- Choosing to tackle either Sarcasm or Named Entity Recognition
- Choosing either Document or Sentence Level
- Selecting a Corpus

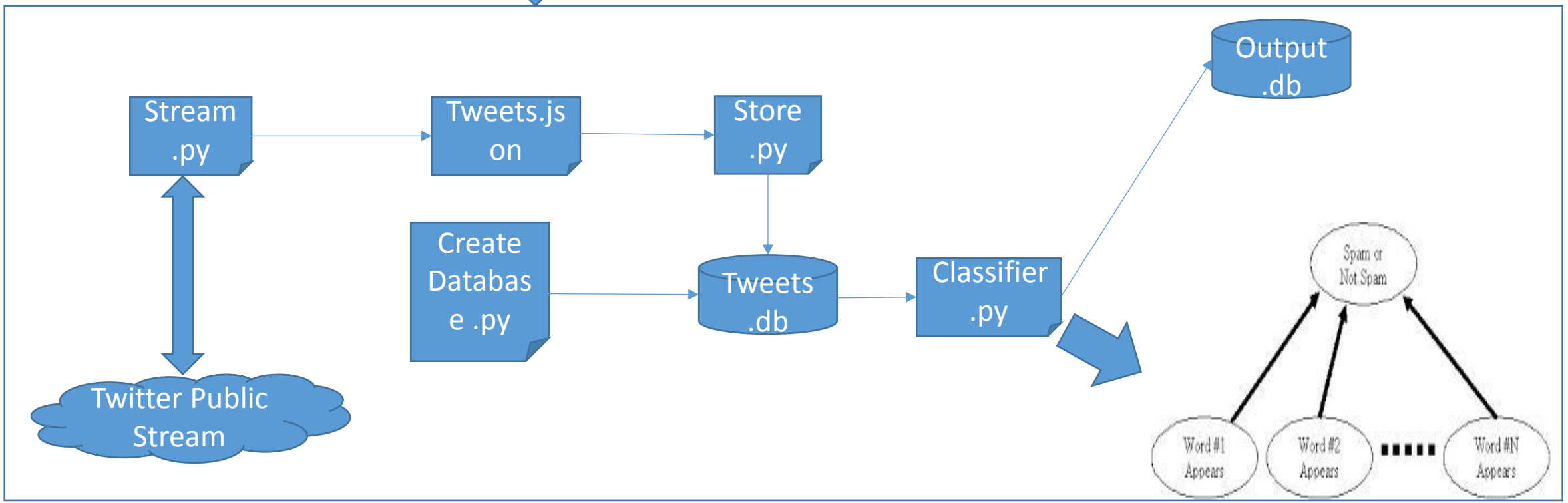
Corpus

- http://www.sananalytics.com/lab/twitter-sentiment/

- http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/
- help.sentiment140.com/for-students/

Project Overview





Bayes Classifier

- $P(c_j \mid d) = \frac{P(d \mid c_j)\, P(c_j)}{P(d)}$
- p(cj | d) = probability of instance d being in class cj,

  This is what we are trying to compute
- p(d | cj) = probability of generating instance d given class cj,

  We can imagine that being in class cj, causes you to have feature d with some probability
- p(cj) = probability of occurrence of class cj,

  This is just how frequent the class cj, is in our database
- p(d) = probability of instance d occurring

  This can actually be ignored, since it is the same for all classes

Naïve Bayes

- Naïve Bayesian is a probabilistic classifier based on Bayes theorem
- Bayes theorem states that

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

- $P(A)$, the prior probability, is the initial degree of belief in A.
- $P(A|B)$, the conditional probability, is the degree of belief in A having accounted for B.
- the quotient $P(B|A)/P(B)$ represents the support B provides for A.

Naïve Bayes

- Super simple, you're just doing a bunch of counts.
- If the NB conditional independence assumption actually holds, a Naive Bayes classifier will converge quicker than discriminative models like logistic regression, so you need less training data.
- Even if the NB assumption doesn't hold, a NB classifier still often does a great job in practice.
- A good bet if you want something fast and easy that performs pretty well.
- Its main disadvantage is that it can't learn interactions between features (e.g., it can't learn that although you love movies with Brad Pitt and Tom Cruise, you hate movies where they're together)

Choosing an Algorithm

- Better data often beats better algorithms, and designing good features goes a long way.
- If you have a huge dataset, then whichever classification algorithm you use might not matter so much in terms of classification performance
- So choosing an algorithm based on speed or ease of use would be better
- Also, If the training set is small, high bias/low variance classifiers (e.g., Naive Bayes) have an advantage over low bias/high variance classifiers, since the latter will overfit.
- http://s3.eddieoz.com/docs/sentiment_analysis/Twitter_Sentiment_Classification_using_Distant_Supervision.pdf

References

Corpus:

http://www.sananalytics.com/lab/twitter-sentiment/

http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/

help.sentiment140.com/for-students/

Papers:

http://kaikuehne.github.io/war/Pak%20and%20Paroubek%20%282010%29.%20Twitter%20as%20a%20Corpus%20for%20Sentiment%20Analysis%20and%20Opinion%20Mining.pdf

http://s3.eddieoz.com/docs/sentiment_analysis/Twitter_Sentiment_Classification_using_Distant_Supervision.pdf

http://arxiv.org/ftp/arxiv/papers/1305/1305.6143.pdf

http://www.sussex.ac.uk/Users/christ/crs/ml/lec02b.html

Thank You !!