Baseline:

**GloVe word vectors – words in vector space (semantics)**

Some labeled data:

Naïve Bayes

**Multinomial naïve bayes**

Each topic is prob distrib over words, calc prob of seeing words in doc based on bias

Linear classifier of word vectors

Clustering – but might cluster on topic

With some labeled data

**Basic semisupervised – trying to get nothing near the boundary (decision line) (only classify ones it is super confident about) – gather some training labels, train b=model, whatever the prediction is on data assume so and retrain**

**look for convergence**

Active learning – instead of random, label something you’re most uncertain

Use source to supervise – objective is to classify articles from the same sources

Add term for optimizing to get all the data in the same news source to be the same bias

Unsupervised based on news source

As long as we have 10 articles for each news source, at least 20 news sources

Predict what news source it is from

Confusion matrix to see how to mistake one news source from another – then see how the news sources fall on the spectrum

Averaging out over the topic – don’t want to look at the topic (idk how to filter out), bias is the only thing left

**Use news source label**

**Take 15 news sources, think for any splitting sources into two camps**

**Try to find best split (start with random split, go from there)**

Local search 9constraint to do 5/10 grouping) – suppose a 7/8 split, iterating through all 8 to see if adding one to the 7 improves the error score. Find clustering that gives you cleanest split

For evaluation, pick one very right, one very left, some in the middle

CREATE UTIL FILE TO CLEAN UP DOCUMENTS?

GET RID OF HARD-CODING IN READFILE

CURRENTLY HAVE BOTH CLASSIFICATION METHODS IMPLEMENTED (add classification to example tuple and also pass in klass with example to naïve bayes)

You limit number of examples you are willing to further classify, it trains model on limited data then asks for you to label a few more that it is very unconfident about; iterate

STOCHASTIC GRADIENT DESCENT