LEARNING TO DETECT FAKE DOCUMENTS

Team #6

Team Members:
Leonid (Leo) Boytsov
Zhou Yu
Di Wang
Hector, Zhengzhong Liu

General Approach

- Design linguistically motivated features
- Train a discriminative classifier to produce document classes (fake or real)
- Train a logistic regression model to produce posterior probabilities

Initial Hypotheses

We can detect fake articles by taking into account features computed for histories longer than three words:

- Perplexity computed using a higher-order n-gram models (*n*>3)
- Perplexity computed using a higher order n-gram model built over sequences of POS-tags
- Mutual information between distant words in a single sentence
- Mutual information in words across sentences
- Syntatctical parsing scores

Designed Features

- Original text n-gram model perplexity (2 to 7 words, Good-Turing smoothing)
- POS tag n-gram model perplexity (2 to 7 tags, Good-Turing smoothing)
- Point-wise mutual information between distant words (more than 2 words in-between)
- Point-wise mutual information between words in different sentences
- Proportion of words that repeat in more than one sentence
- Syntactical parsing scores

Additional training/testing data

- First we tried NLTK, but produced documents were very different from the provided testing and training data
- Then we back-ported the generating function from the CMU Sphinx toolkit to the CMU Cambridge toolkit
- Real data was generated by extracting a long piece that started at a random sentence
- We generated documents that were much longer than those in the training set
- The generated training/testing sets were 10x the size of the sets provided by instructors

Machine Learning Methods

- Soft-margin SVM
  - Good classifier, but no posterior probabilities
  - Tried different kernels
- Regularized logistic regression
  - Probably not as good as SVM
  - But it can produce posterior probabilities
  - LASSO (L1) and Tikhonov (L2) regularization
- Naïve Bayes
- Parameters tuned by 10-fold cross-validation
- We use two sets of features
  - Complete includes all n-gram models
  - Short includes only 3-gram and 4-gram models

Experimental results

Original Dev Set

Hard Metric (SVM)

Original training set		Generated training set	
short	complete	short	complete
0.88	0.90	0.92	0.895

$2^{\text{Soft Metric}}$ (LR-L1)

Original training set		Generated training set	
short	complete	short	complete
0.7	0.73	0.75	0.76

Generated Dev Set

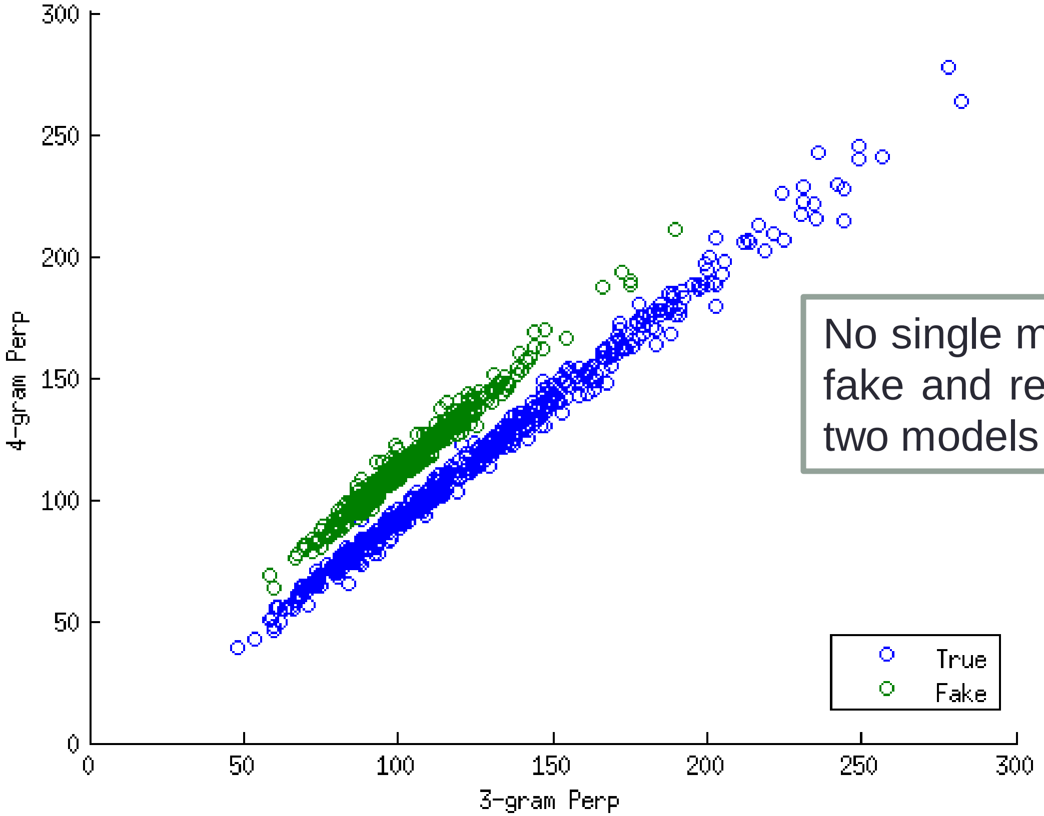
Hard Metric (SVM)

Original training set		**Generated training set**	
short	complete	short	complete
0.927	0.923	0.927	0.901

$2^{\text{Soft Metric}}$ (LR-L1)

Original training set		**Generated training set**	
short	complete	short	complete
0.8	0.78	0.8	0.73

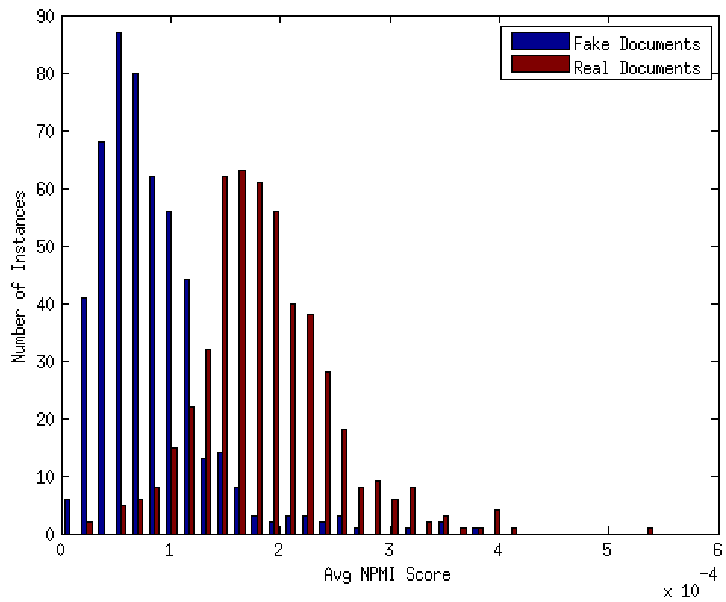
Perplexity distributions of 3- and 4-gram models for fake and real documents



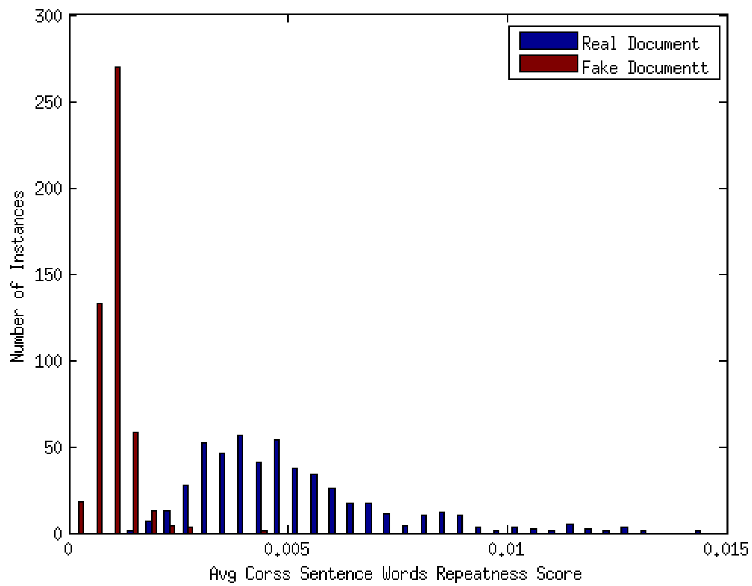
No single model can separate fake and real documents, but two models can!

Word Occurrence Features

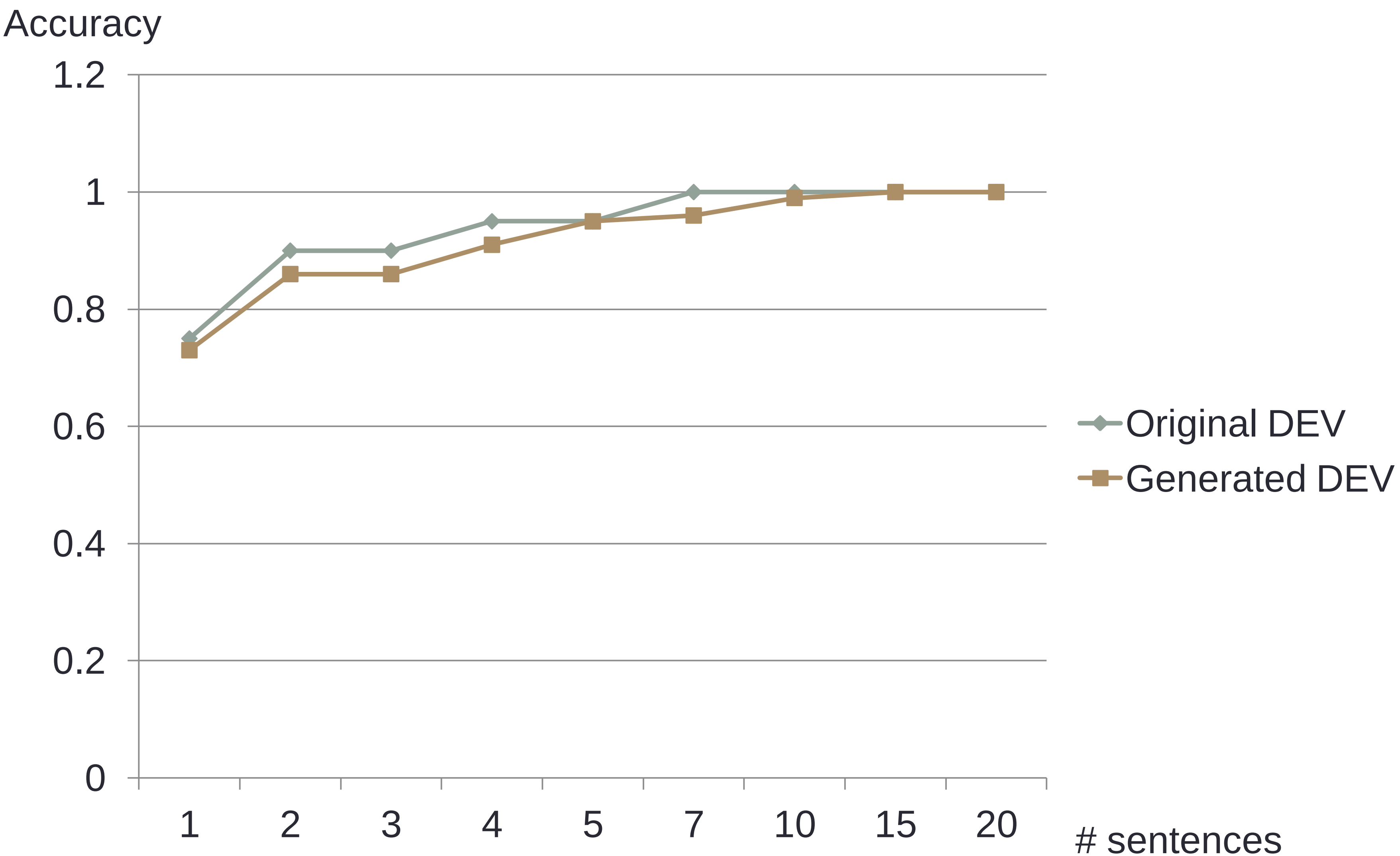
Point-wise Mutual Information



Average portion of repeated words



Accuracy vs. # Sentences



Conclusions

Best Machine Learning Methods

- Soft margin linear SVM
- Logistic regression with the LASSO regularization

Features

- Perplexities computed using n-gram models
- No single n-gram model can detect fake sentences
- Yet, two n-gram models are sufficient to classify accurately
- Probability estimation may work better with all n-gram models
- No feature works well for short sentences
- Syntactical parsing scores were not helpful

Additional data

- Was probably helpful on the development set
- Unsure about unseen data

Questions?