# Distribution based Stemmer Refinement

B. L. Narayan and Sankar K. Pal
Machine Intelligence Unit
Indian Statistical Institute
http://www.isical.ac.in/~bln_r/

# Contents

# What is Stemming ?

- Grouping morphologically similar words
- Clubbing inflections, derivationally related forms, *etc.*
- Assumption: morphological similarity $\equiv$ semantic similarity.
- Similar terms: Conflation, Lemmatization, Normalization
- Clustering, Feature reduction, etc.

# Examples

- visit, visits, visiting, visited $\longrightarrow$ visit
- complete, completes, completed, completing, completion, completeness, completely $\longrightarrow$ complet
- ring, rings, ringed, ringing $\longrightarrow$ ring
- range, ranges, rang, ranging, ranged $\longrightarrow$ rang

# Why Stem ?

- Information Retrieval
  - Smaller inverted index
  - Increase in Recall
  - Query Expansion
    - Natwar Iraq visit
    - Natwar Iraq visit OR visited OR visits OR visiting
- Text categorization/classification
  - Fewer number of features
  - Simpler classification models

# Stemmers and Errors

- Lovins, Dawson, Porter, Paice/Husk, Truncate(3), Truncate(5), Krovetz, Xu and Croft
- Strength: amount of reduction in dictionary size
- Stemming Errors
  - Under-stemming: words that should have been grouped into the same class are not so.
  - Over-stemming: unrelated words are merged together.

# Under-stemming

- A part of the suffix is left untouched
  - visit, visits, visiting, visited $\rightarrow$ visit
  - visitor, visitors $\rightarrow$ visitor
- Related words are not merged
  - dwell, dwells, dwelling, dwellings $\rightarrow$ dwell
  - dwelt $\rightarrow$ dwelt
- Information is under-utilized, but not harmful

# Over-stemming

- Too large a suffix is removed
  - divide, divides, division, divine $\rightarrow$ divi
- Introduces new (possibly, wrong) relations between words
  - range, ranges, rang, ranging, ranged $\rightarrow$ rang
- Merges together words with different senses
  - complete, completes, completed, completing, completion, completeness, completely $\rightarrow$ complet

# Stemmer Refinement

- Exception List
- Rule modification
- Dictionary based correction
- Co-occurrence Based Stemming
- Distribution Based Stemming

# Co-occurrence based refinement

- Corpus-based
- Refines an existing stemmer by partitioning some of the equivalence classes
- An equivalence class should contain only words with high co-occurrences
- Computes expected and actual number of co-occurrences of pairs of words

# Co-occurrence based refinement

$$em(w_i, w_j) = \max\left(\frac{n(i,j) - En(i,j)}{n_i + n_j}, 0\right)$$

- Creates a graph with edge weights being the *em*-scores
- Optimal graph partitioning: Computationally expensive
- Connected component labelling (edge weights are thresholded)

# Scope for Improvement

- Creation of new (and sometimes meaningless) words
- Complexity and speed
- Substitute words
- Cross-corpus stemming

# Distribution Based Stemming: Overview

- Refines an existing stemmer
- Corpus-based
- Documents are classified into a number of topics
- Words are assumed to be arising from a multinomial distribution
- Words having similar distributions may be stemmed to the same stem

# Distribution Based Stemming: Initialization

- Choose a strong stemmer
  - More over-stemming, less under-stemming
- For each stem equivalence class:
  - Estimate the distributions of all the words
  - Order the words in descending order of their frequency in the corpus
  - The first word is defined to stem to itself

# Distribution Based Stemming: Refinement

- For each stem equivalence class:
  - For every new word, check if it may be merged with any of the existing stems
  - If it is very *different* from each of the existing stem classes, create a new one for the current word
- Measuring Difference: Sequential Hypothesis Testing
- Two thresholds $t_1 < t_2$ chosen

# Distribution Based Stemming: Procedure

- Compute statistic to test if $i$th word may be merged with $j$th stem class

$$T = \frac{m_j}{n_i} \sum_{l=1}^{K} \frac{n_{il}^2}{m_{jl}} - n_i$$

  - $T \sim \chi^2_{K-1}$
  - If $T < t_1$, merge with corresponding stem class
  - If $T > t_2$, create new class for current word
  - Else leave decision for later.

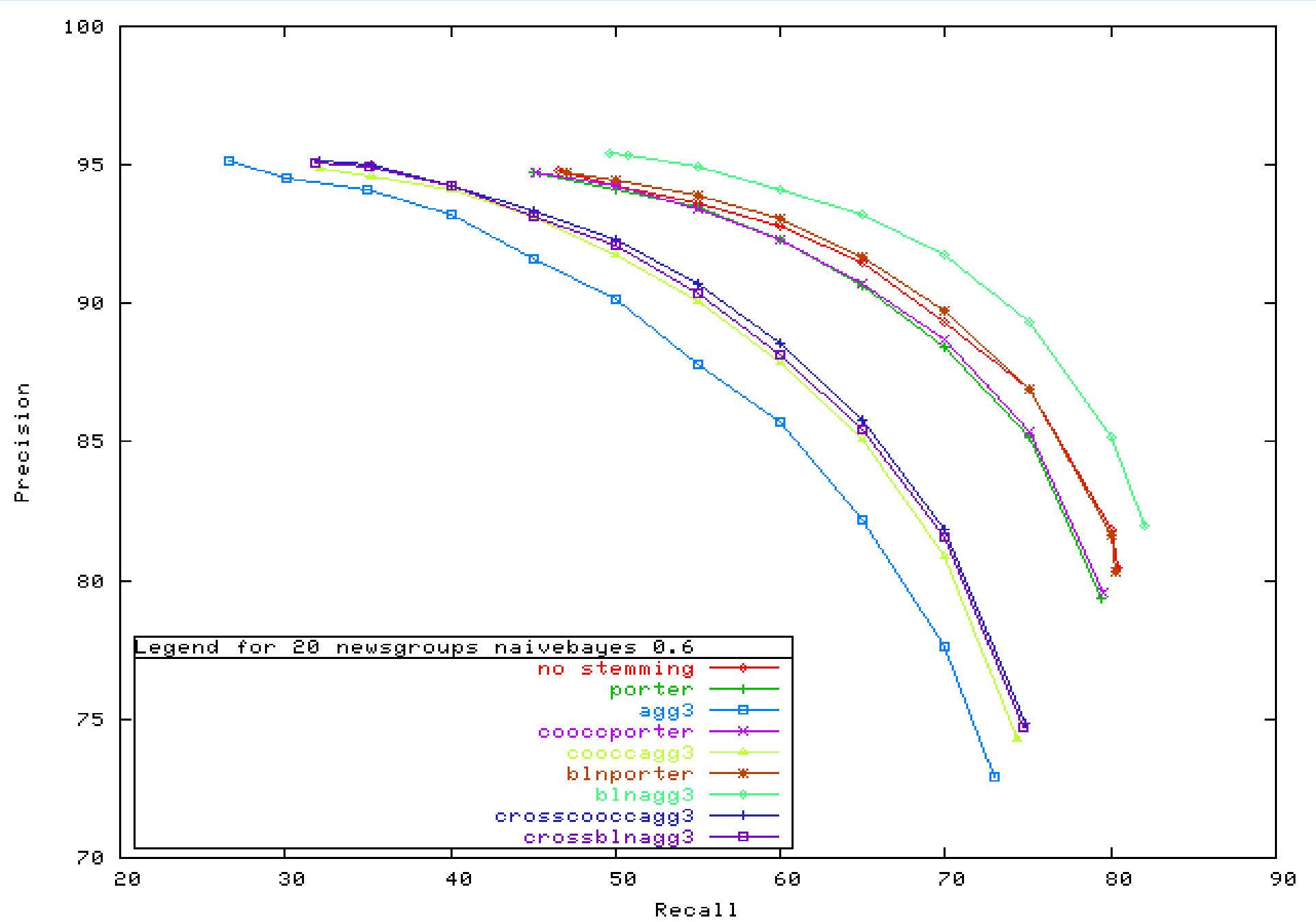
- Modify thresholds (reduce $t_1$, increase $t_2$) and repeat.

# Characteristicss

- No new words created
- Fast, as the test statistic needs to be computed just once per class
  - Can handle (during refinement) any strong stemmer
  - Does not need a stemmer to begin with (it would be conflation then)
- Can stem substitute words to the same stem
- Since the criterion for stemming is more general than just co-occurrence, cross-corpus stemming is supported.

## Experimental Results

- 20NG: About 20,000 documents, 20 topics
- $t_1 = \chi^2_{19,0.05}$, $t_2 = 2\chi^2_{19,0.05}$
- Porter: 40821 classes, singletons: 32479, rest: 8342
- Truncate(3): 8158 classes
  - Co-occurrence: 30710 classes
  - Distribution: 22623 classes
  - *angle, angular* and *angstrom* were kept together
  - *war, ware* and *ward* were separated out




Classification Results: 20 Newsgroups, NaiveBayes

# References


- Porter, M.F.: An algorithm for suffix stripping. Program 14 (1980) pp. 130-137
- Krovetz, R.: Viewing morphology as an inference process. In Proc. 16th ACM SIGIR conference, Pittsburgh (1993) pp. 191-202
- Xu, J., Croft, W.B.: Corpus-based stemming using coocurrence of word variants. ACM Transactions on Information Systems 16 (1998) pp. 61-81
- `http://snowball.tartarus.org/`
- `http://www.cs.cmu.edu/~mccallum/bow/`

# Thank You