Feature Engineering

- Tokenization
- Contextual Features
  - n-grams
  - position information
- Linguistic Features
  - Stemming
  - Noun phrases
- Structural Features
  - structural markups
  - hypertext
- Feature Subset Selection
  - Frequency-based
  - TF-IDF
  - Machine Learning methods (*not* class-blind)
- Feature Construction
  - Latent Semantic Indexing
- Stop Lists
  - Removal of frequently occurring words

Tokenization

- Identification of basic document entities („words“)
  - typically performed in indexing phase
- Issues in tokenization:
  - ***Finland's capital*** →

    Finland? Finlands? Finland's?
  - ***Hewlett-Packard*** → ***Hewlett*** and ***Packard*** as two tokens?
    - ***State-of-the-art***: break up hyphenated sequence.
    - ***co-education*** ?
    - ***the hold-him-back-and-drag-him-away-maneuver*** ?
    - It's effective to get the user to put in possible hyphens
  - ***San Francisco***: one token or two? How do you decide it is one token?

Numbers

- Many different formats
  - *3/12/91* *Mar. 12, 1991*
  - *55 B.C.*
  - *B-52*
  - *My PGP key is 324a3df234cb23e*
  - *100.2.86.144*
- Also in abbreviations:
  - We want to match ***U.S.A.*** and ***USA***
- Typically, periods etc. are removed
- Special recognizers for dates, IP addresses, etc.

Tokenization: Language issues

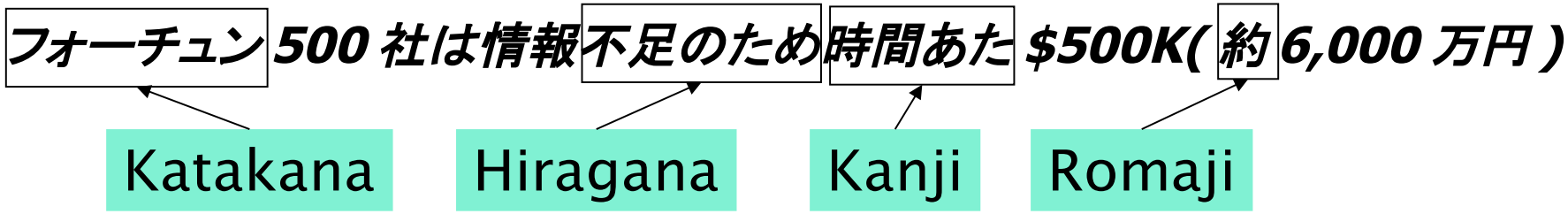
- ***L'ensemble*** → one token or two?
  - ***L*** ? ***L'*** ? ***Le*** ?
  - Want ***l'ensemble*** to match with ***un ensemble***

- German noun compounds are not segmented
  - Lebensversicherungsgesellschaftsangestellter
  - 'life insurance company employee'

- Special Characters:
  - Umlauts: ***Tuebingen*** vs. ***Tübingen***
  - Accents: ***résumé*** vs. ***resume***.

Tokenization: language issues

- Chinese and Japanese have no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - Not always guaranteed a unique tokenization
- Further complicated in Japanese, with multiple alphabets intermingled



  - Dates/amounts in multiple formats

Tokenization: language issues

- Arabic (or Hebrew) is basically written right to left, but with certain items like numbers written left to right
- Words are separated, but letter forms within a word form complex ligatures

Example:

- استقلت الجزائر في سنة 1962 بعد 132 عاما من الاحتلال الفرنسي.

← → ← → ← start

- 'Algeria achieved its independence in 1962 after 132 years of French occupation.'

- With Unicode, the surface presentation is complex, but the stored form is straightforward

Case folding

- Reduce all letters to lower case
- Exception: upper case (in mid-sentence?)
  - e.g., ***General Motors***
  - ***Fed*** vs. ***fed***
  - ***SAIL*** vs. ***sail***
  - ***MIT*** vs. ***mit***
- Typically, everything is converted to lower case anyways
  - automatic disambiguation via context

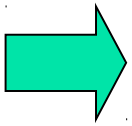
Lemmatization

- Reduce inflectional/variant forms to base form, e.g.,
  - *am, are, is* → *be*
  - *car, cars, car's, cars'* → *car*
- *the boy's cars are different colors*
  → *the boy car be different color*
- Lemmatization implies doing “proper” reduction to dictionary headword form

Stemming

- Reduce terms to their “roots” before indexing
- “Stemming” suggest crude affix chopping
  - language dependent
  - e.g., ***automate(s), automatic, automation*** all reduced to ***automat***.

for example compressed and compression are both accepted as equivalent to compress.



for exampl compress and compress ar both accept as equival to compress

Porter's algorithm

- Most popular algorithm for stemming English
  - Bad results from a linguistic point of view
  - but results suggest that for IR and text classification, it is at least as good as other stemming options
- Conventions + 5 phases of reductions
  - phases applied sequentially
  - each phase consists of a set of commands
- Example Rules:
  - *sses* → *ss*
  - *ies* → *i*
  - *ational* → *ate*
  - *tional* → *tion*
- Sample Convention:
  - select the rule that applies to the longest suffix
  - what is a suffix is determined by word length
  - Example:
    - *replacement* → *replac*
    - *cement* → *cement*

Stop Words

- Remove most frequent words in the (English) language
  - a, about, above, across, after, afterwards, again, against, all, almost, alone, along, already, also, although, always, am, ... yet, you, your, yours, yourself, yourselves
  - http://www.ranks.nl/stopwords/
- Assumption:
  - These words occur in all documents and are irrelevant for retrieval
- Stop lists used to be popular, but are nowadays often avoided, because important information may be lost
  - polysemous words: „can“ as a verb vs. „can“ as a noun
  - phrases: “Let it be”, “To be or not to be”, pop group „The The“
  - relations: “flights to London” vs. „flights from London“

Stemming and Stop Words: Example

- Original Text

> Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals.

- After Porter stemming and stopwords removal

> market strateg carr compan agricultur chemic report predict market share chemic report market statist agrochem

Example taken from T. Joachims, http://www.cs.cornell.edu/Courses/ cs630/2004fa/lectures

Stemming: Evaluation

- Sometimes too aggressive in conflation
  - e.g., policy/police, execute/executive, university/universe
- Sometimes miss good conflations
  - e.g., European/Europe, matrices/matrix, machine/machinery
- Abbreviations, polysemy and names maybe problematic
  - E.g.: Stemming “Gates” to “gate”, may be bad !
- In general:
  - Stemming may increase recall
    - more documents will be indexed under fewer terms
  - but at the price of precision
    - some terms may be too general to discriminate documents
- Stemming may be good combination with n-grams
  - stemming increase recall, n-grams decrease them
  - simple alternative to noun phrase extraction

Thesauri and soundex

- Handle synonyms and homonyms
  - Hand-constructed equivalence classes
    - e.g., ***car*** = ***automobile***
    - ***color*** = ***colour***
  - can be looked up in Thesauri
    - Wordnet (http://wordnet.princeton.edu/)
    - Wiktionary (http://en.wiktionary.org)
- Soundex:
  - Traditional class of heuristics to expand a query into phonetic equivalents
    - Language specific – mainly for names
    - E.g., ***chebyshev*** → ***tchebycheff***
  - American standardized SoundEx (from the 1920's)
    - map each name into one letter and three digits
    - letters that are pronounced similar have the same target

Feature Subset Selection

- Using each word as a feature results in tens of thousands of features
- Many of them are
  - irrelevant
  - redundant
- Removing them can
  - increase efficiency
  - prevent overfitting
- Feature Subsect Selection techniques try to determine appropriate features automatically

Unsupervised FSS

- Using domain knowledge
  - some features may be known to be irrelevant, uninteresting or redundant
- Random Sampling
  - select a random sample of the feature
  - may be appropriate in the case of many weakly relevant features and/or in connection with ensemble methods
- Frequency-based selection
  - select features based on statistical properties
  - TF: term frequency
    - keep the *n* most frequent words (fixed number)
    - keep all words that occur at least *k* times (thresholding)
  - TF-IDF: trade off term frequency with document frequency

Supervised FSS

- Filter approaches:
  - compute some measure for estimating the ability to discriminate between classes
  - typically measure feature weight and select the best n features
  - problems
    - redundant features (correlated features will all have similar weights)
    - dependant features (some features may only be important in combination
- Wrapper approaches
  - search through the space of all possible feature subsets
  - each search subset is tried with the learning algorithm
  - good results, but typically too expensive for practice

Supervised FSS: Filters

- foreach term t
  - W[t] = term weight according to some criterion measuring discrimination
- select the n terms with highest W[t]

- basic idea of term weights:
  - a good term should discriminate documents of different classes
  - there must be some correlation between the class and the occurrence (t) or non-occurrence ($\bar{t}$) of a term.
- examples for discrimination measures:
  - **information gain**: $IG(T) = E(C) - [p(t)E(C|t) + p(\bar{t})E(C|\bar{t})]$

    where $E(C) = -\sum_{c \in C} p(c) \log p(c)$
  - **log-odds ratio:** $LO(T) = \log \frac{p(t|c_1)}{p(\bar{t}|c_1)} - \log \frac{p(t|c_2)}{p(\bar{t}|c_2)}$

The χ^2 test

- Build a 2 x 2 contingency table for each class-term pair

	D does not contain t	D contains t
D is of class 0	k_{00}	k_{01}
D is of class 1	k_{10}	k_{11}

- Basic idea
  - Aggregates the deviations of observed values from expected values if the occurrence of term were independent of class
  - expected value: how many occurrences of the term could we expect if the terms occurs with the same frequency as in all documents

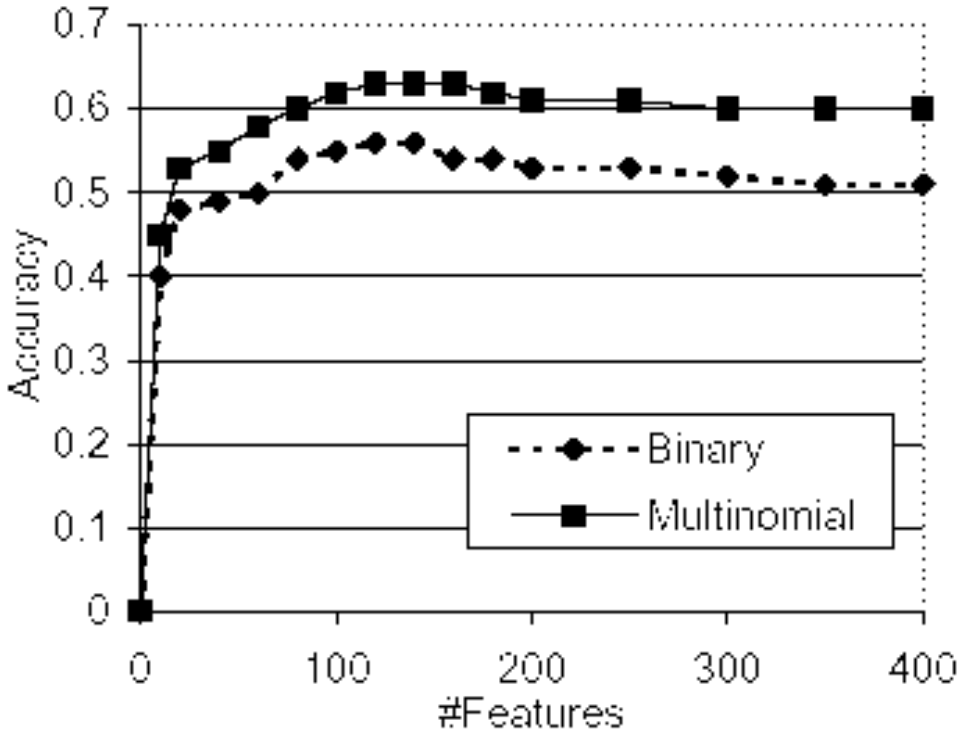
$$E(k_{ij}) = (k_{i0} + k_{i1}) \frac{k_{0j} + k_{1j}}{n}$$

- Test Statistic:

$$\chi^2 = \sum_{i,j} \frac{(k_{ij} - E(k_{ij}))^2}{E(k_{ij})} = \frac{n(k_{11}k_{00} - k_{10}k_{01})^2}{(k_{11}+k_{10})(k_{01}+k_{00})(k_{11}+k_{01})(k_{10}+k_{00})}$$

Feature Selection Results

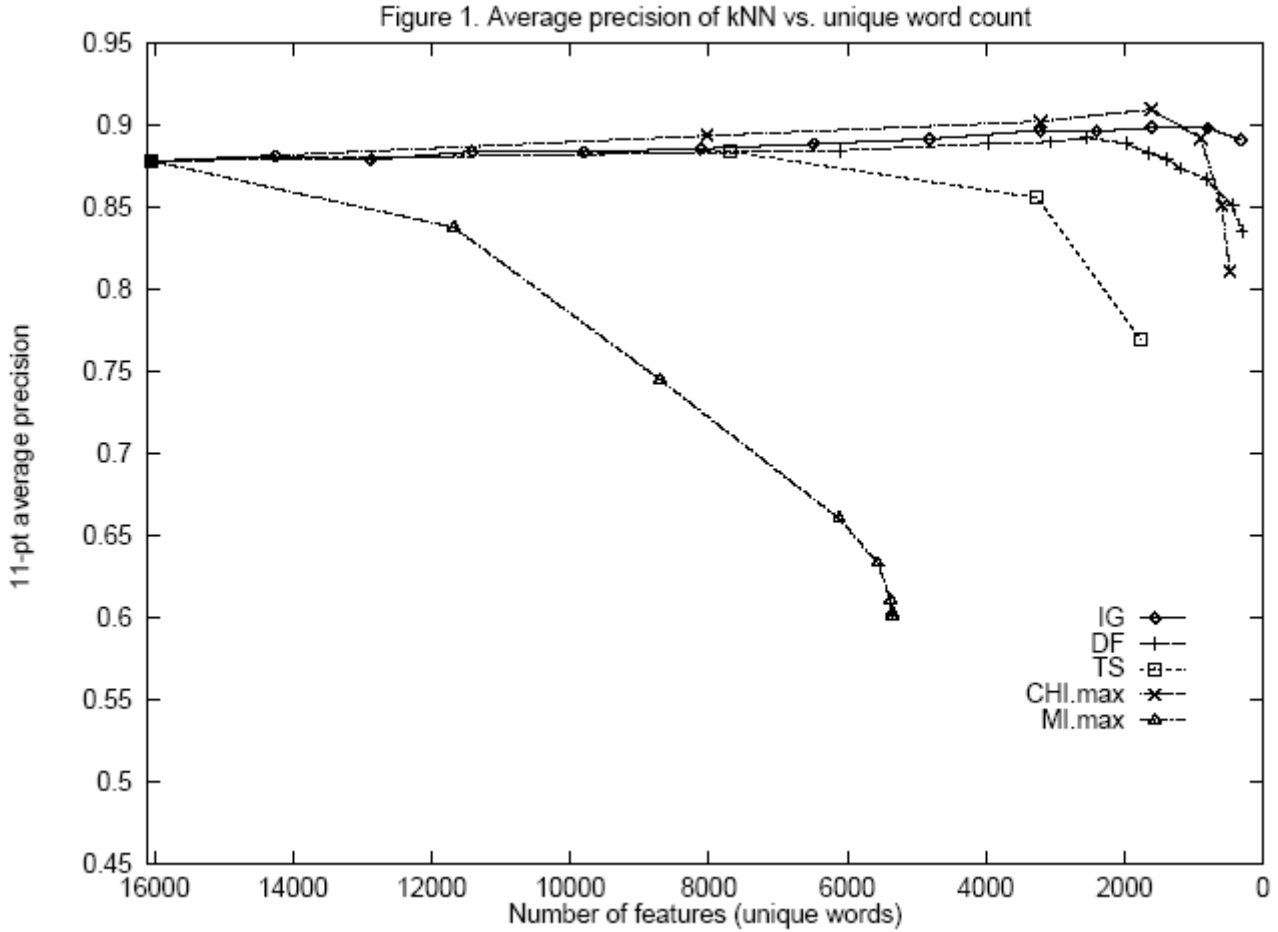
- Naive Bayes classifier cannot overfit much
  - but clearly feature subset selection improves the result



Effect of feature selection on Bayesian classifiers

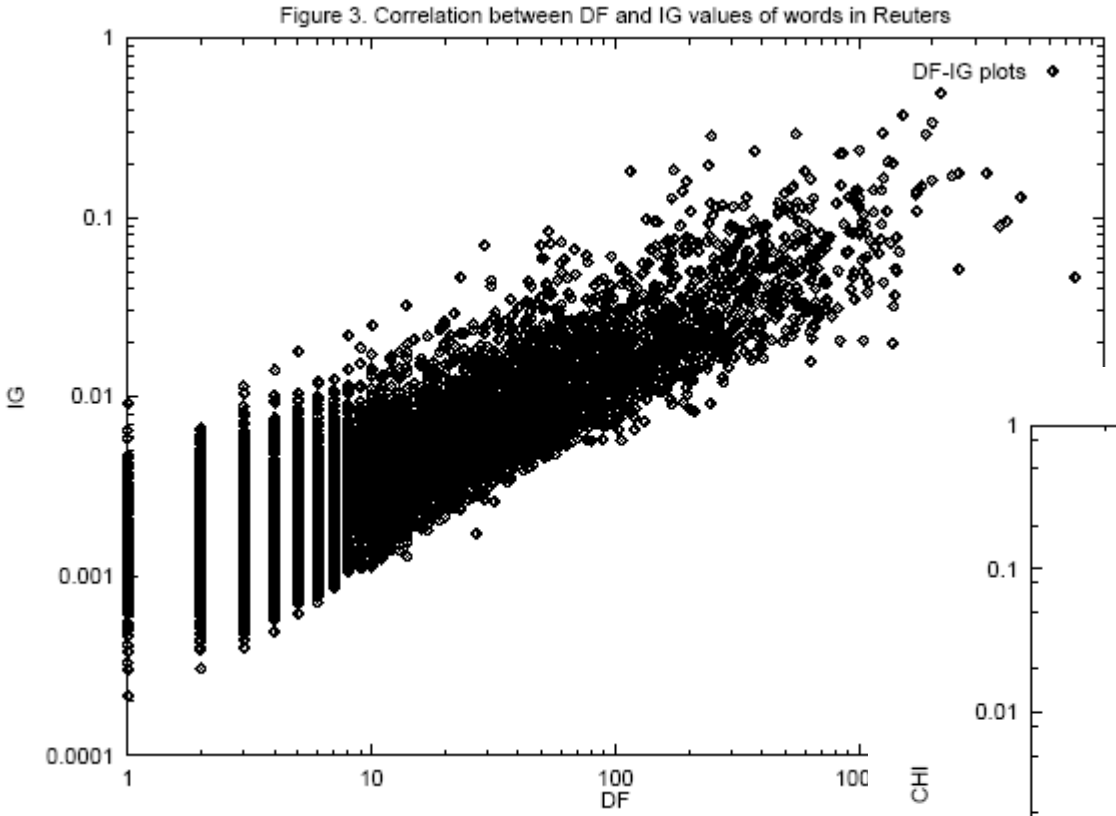
Corpus: US. Patent database, feature selection by Fisher's discriminant

FSS Results

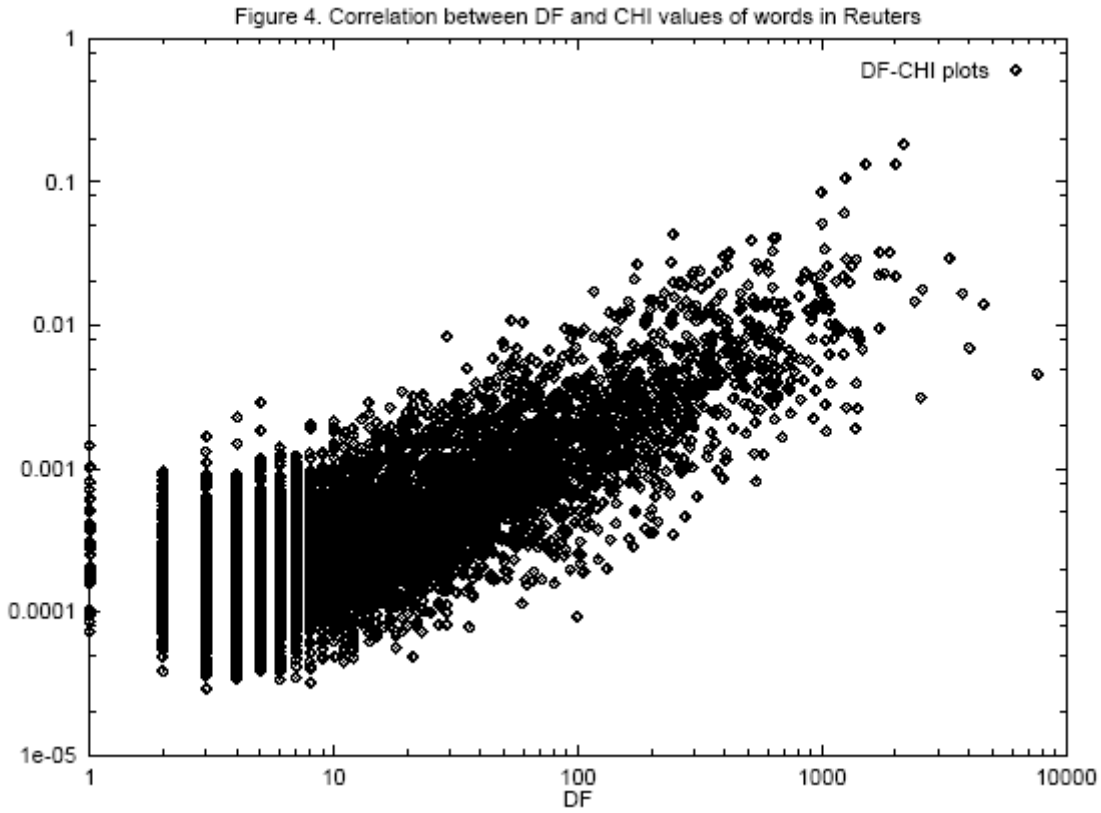


(Yang & Pedersen, ICML-97)

Correlation beetween Measures



DF = document frequency
IG = information gain
CHI = chi^2



- different measures measure similar properties
- when one is high, the others tend to be high as well

(Yang & Pedersen, ICML-97)

n-grams

- Exploit context by using sequences of n words instead of single words
  - "coal mining" vs. "data mining" ($n = 2$, *bigrams*)
- Observation:
  - number of possible n-grams increases with n
  - but their frequency of occurrence decreases
- Subsequence Property:
  - If a sequence of words occurs n times, each of its subsequences occurs at least n times
  - this holds for term frequency and/or document frequency

Finding Frequent *n*-grams

- Problem:
  - Find sequences of words that occur with a given minimum frequency (a frequent n-gram)

- Finding frequent n-grams
  - based on Apriori Algorithm for finding frequent itemsets (Agrawal et al., 1995)

1. assume we have all frequent n-grams of length $n-1$
2. build all pairwise extensions by overlapping two sequences of length $n-1$ to one sequence of length n
3. only count the frequency of those
4. repeat for finding freqent n+1-grams, etc.

Evaluation on 20 Newsgroups

Pruning	n	Error	#features
no		47.07	*71,731*
	1	46.18	*36,534*
DF: 3	2	45.28	*113,716*
TF: 5	3	45.05	*155,184*
	4	45.18	*189,933*
	1	45.51	*22,573*
DF: 5	2	45.34	*44,893*
TF: 10	3	46.11	*53,238*
	4	46.11	*59,455*

Pruning	n	Error	#features
no		47.07	*71,731*
	1	45.88	*13,805*
DF: 10	2	45.53	*20,295*
TF: 20	3	45.58	*22,214*
	4	45.74	*23,565*
	1	48.23	-
DF: 25	2	48.97	-
TF: 50	3	48.69	-
	4	48.36	-

DF = minimum document frequency TF = minimum term frequency

a term must satisfy both constraints

Error = Classification Error (10-fold x-val) with Ripper rule learner

Evaluation of Frequency-Based Selection

- A little context improves performance
  - bigrams are usually better than unigrams
  - trigrams are sometimes better
  - no gain for $n > 3$
- Frequency pruning
  - most frequent features need not be good (typically placeholders for numbers and stop words)
  - too much pruning hurts
- Overfitting through repetition of parts of texts
  - the phrase "*closed roads mountain passes serve way escape*" occurs 153 times and gives the 4 most frequent 4-grams.
- Other measures (TF-IDF, CHI^2, Log-Odds, ...) might produce better results
  - but subsequence property does not hold

  → much more candidates would have to be evaluated
  - results of (Yang & Pedersen, 97) for DF were not so bad

Statistical Tests for Filtering Bigrams

- Frequency-based pruning alone may not be enough
  - the most frequent sequences will be sequences consisting of the most frequent words
- What is interesting is
  - whether the probability of occurrence for a pair of words differs from the product of the individual probabilities
  - H0: terms t_1 and t_2 occur independently: $p(t_1, t_2) = p(t_1)\, p(t_2)$
  - H1: there is a dependency: $p(t_1, t_2) \neq p(t_1)\, p(t_2)$
- Likelihood ratio test:
  - statistical test for determining whether H0 holds or not
- Alternatives:
  - one could also use a χ^2-test for testing whether the observed number of bigrams of t_1 and t_2 differs from the expected

Extracting Noun Phrases

- the focus of frequent n-grams can be improved, if only n-grams that are likely to be phrases are used
- can be realized with a simple filter that attaches to each word its „part-of-speech“ (lexical category)
  - e.g.: only admit combinations Noun-Noun and Adverb-Noun
  - can be looked up in a dictionary, but is very often ambigous (e.g. „can“: auxiliary verb or noun)
- Example: (Manning & Schütze, 2001) after (Justeson & Katz, 1995)
  - most frequent bigrams w/o and with filter

frequency	bigram
80871	of the
58841	in the
26430	to the
21842	for the
21839	and the

frequency	bigram	pattern
11487	New York	AN
7261	United States	AN
5412	Los Angeles	NN
3301	last year	AN
3191	Saudi Arabia	NN

Linguistic Phrases: Motivation

"I am a student of Computer Science
at Carnegie Mellon University."

- Among home pages that typically occur in a Computer Science Department
  (for students, faculty, staff, department, courses, projects,...)

  Which are the words that are most characteristic for recognizing this as a student home page?

AutoSlog (Riloff, 1996)

- Originally built for information extraction
- Detects all instantiations of syntactic templates in a text
  - part-of-speech tagging is necessary
- These can be used as features

Syntactic Heuristic	*Phrasal Feature*
`noun aux-verb <d-obj>`	I am <_>
`<subj> aux-verb noun`	<_> is student
`noun prep <noun-phrase>`	student of <_>

Mixed Results

	Rainbow	Ripper
words	45.70	77.78
phrases	51.22	74.51
both	46.79	77.10

- Rainbow: Increase
  - Rainbow is a Naive Bayes implementation
  - Rainbow misclassifies too many pages of class OTHER.
  - The lower coverage of the phrase features improves *precision* in the other classes.
- Ripper: Decrease
  - Ripper is a rule learning algorithm
  - Ripper uses the class OTHER as the default class
  - The lower coverage of the phrase features decreases *recall* in the other classes.

Best Bigrams vs. Phrases

3 Best Features	*Phrases*	*Stemmed Bigrams*
	I am <_>	home page
student	<_> is student	comput scienc
	student in <_>	depart of
	university of <_>	comput scienc
faculty	professor of <_>	of comput
	<_> is professor	univ of
	department of <_>	comput scienc
department	undergraduate <_>	the depart
	graduate <_>	scienc depart

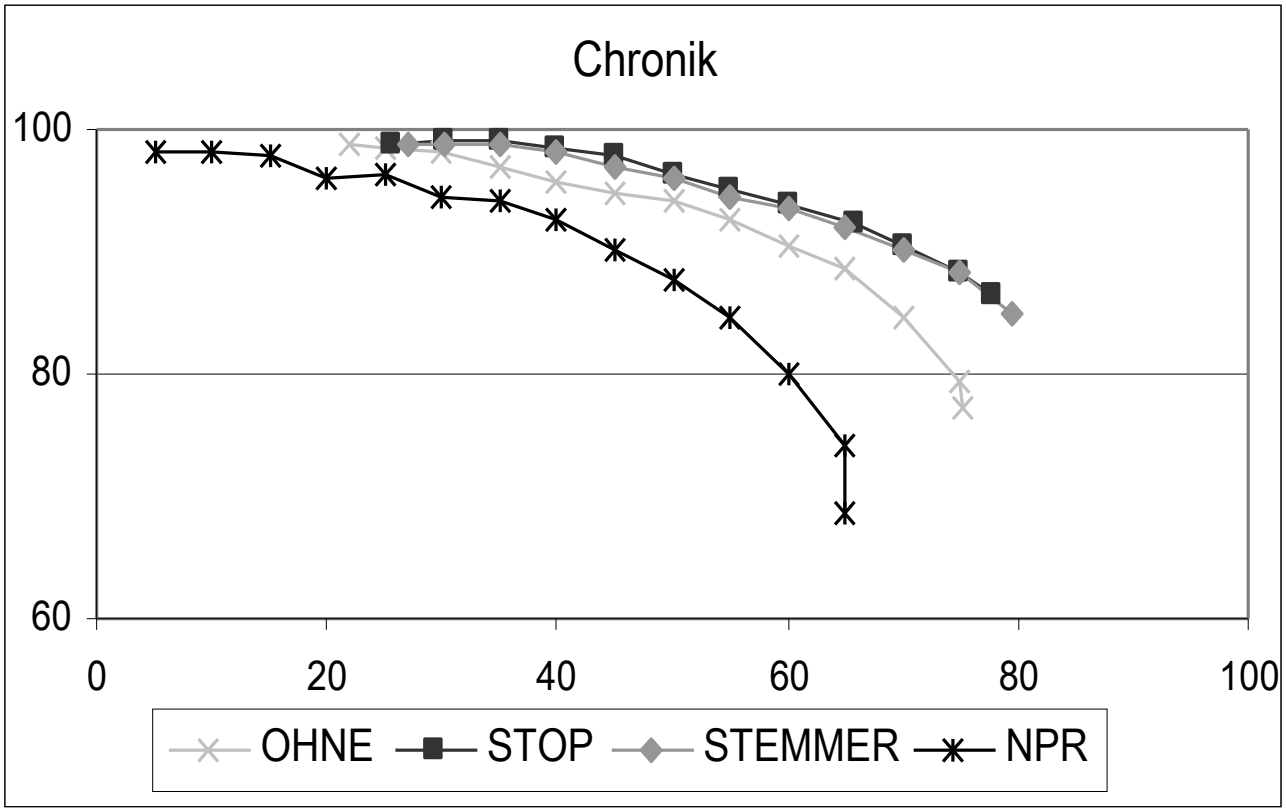
terms are sorted by $p(t|c)$

Evaluation

- Phrases seem to help when the word-based classifier over-generalizes
  - lower recall
  - higher precision
- Phrases vs. Bigrams
  - phrases seem to make more sense
  - only slightly more phrase features than word features
  - no difference in accuracy

Stemming and Phrases in German

OHNE
rechtsextreme gruppe bekennt sich zu anschlag in london nm zwei tote und verletzte attentat richtete sich gegen homosexuelle offenbar viele auslaender unter den verletzten eine rechtsextreme gruppe hat sich zu dem anschlag in london bekannt bei dem freitag abend zwei menschen getoetet und mehr als verletzt wurden die gruppierung namens weisse woelfe habe sich in einem anonymen anruf bei einem bbclokalsender der tat bezichtigt teilte ein polizeisprecher mit dieselbe organisation sowie andere rechtsextremistengruppierungen hatten sich bereits zu den beiden fremdenfeindlichen anschlaegen vom vergangenen und vorvergangenen samstag bekannt bei denen insgesamt menschen verletzt worden waren
STOP
rechtsextreme gruppe bekennt anschlag london nm zwei tote verletzte attentat richtete homosexuelle offenbar auslaender verletzten eine rechtsextreme gruppe anschlag london freitag zwei menschen getoetet verletzt die gruppierung weisse woelfe anonymen anruf bbc lokalsender tat bezichtigt teilte polizeisprecher dieselbe organisation rechtsextremisten gruppierungen fremdenfeindlichen anschlaegen vergangenen vorvergangenen samstag menschen verletzt
STEMMER
rechtsextreme gruppe bekennen sich zu anschlag i londo nm zwei tote u verletzte attentat richten sich geg homosexuell offenbar viele auslaend unter d verletzte eine rechtsextreme gruppe haben sich zu d anschlag i londo koennen bei d freitag ab zwei mensche getoetet u mehr als verletzen werden di gruppierung namens weisse woelfe haben sich i ein anonyme anruf bei ein bbc lokalsend d tat bezichtigen teilte ein polizeisprech mit dieselbe organisation sowie ander rechtsextremist gruppierung haben sich bereits zu d beid fremdenfeindlich anschlaege vom gehen u vorvergangene samstag koennen bei dene insgesamen mensche verletzen werden war
NPR
rechtsextreme_gruppe anschlag london_nm tote verletzte_attentat homosexuelle auslaender verletzten rechtsextreme_gruppe anschlag london freitag menschen gruppierung weisse_woelfe anonymen_anruf bbclokalsender_der_tat polizeisprecher organisation andere_rechtsextremistengruppierungen fremdenfeindlichen_anschlaegen vergangenen_und_vorvergangenen_samstag menschen



Results

© Markus Mayer

- Task:
  - Classification of German newswire articles into categories like sports, politics, culture, etc.
- Stemming and Stoplists improve accuracy
  - +5.14% Rainbow, +3.46% Ripper
- Noun phrases decrease performance
  - -9.5% Rainbow, -15.75% Ripper
  - mostly due to overfitting and resulting low recall

Latent Semantic Indexing

- PROBLEM
  - Words may capture the *latent semantic* content of a document in different ways
    - **Synonyms:** different words may describe the same concept (⇒ poor *recall*)
    - **Polysemy:** the same word may describe different concepts (⇒ poor *precision*)
- Suggestion for SOLUTION (Deerwester et al., JASIS 1990)
  - transform term-document matrix into a lower-dimensional space using *singular value decomposition*
  - each dimension of the lower-dimensional space is a linear combination of the original dimensions
    - representing a meaningful combination of words
  - terms and documents are vectors in this new space

LSI - Example

- Example Documents: (Flexer & Puig, 2001)
  - A1: Die Beamtin schenkte ihrer Mutter nur rote Rosen und blaue Nelken.
  - A2: Rosen, Tulpen, Nelken, alle drei verwelken. Nur eine nicht, die heißt Vergißmeinnicht.
  - B1: Menschen, die auf Hunde und Katzen allergisch reagieren, sind nur überempfindlich.
  - B2: Nur Hunde, die bellen beissen nicht, und bei Nacht sind alle Katzen grau.
- Projection into 2 dimensions

LSI - Example (Ctd.)