Corpus Linguistics
N-Gram Models

What is Corpus Linguistics?

- A methodology to process text ….. AND…..
- Resource provision of (possibly annotated) text

- The Corpus is a collection of text
  - Utilizes a maximally representative sample of machine-readable text of a language or a particular variety of text or language
- Statistical analysis
  - Word frequencies
  - Collocations
  - Concordances
- Text is sometimes annotated, or the goal of the research is to annotate it

Questions to keep in mind for any NLP application:

- Will you need a corpus?
- What are the requisite characteristics of the corpus?
- What type of annotations are needed in this corpus?
- Is there an appropriate annotated corpus already available for your use?

Questions … : (cont'd)

- Will a human or a computer do the annotations?
- What tag-set will you use?
- Does the tag-set need to be specialized for your application or domain?
- If computer annotates, how will you train the system?

Preliminary Text Processing Required :

- Find the words:
  - Filter out ‘junk data’
    - Formatting / extraneous material
    - First be sure it doesn’t reveal important information
  - Deal with upper / lower case issues
  - Tokenize
    - Decide how you define a ‘word’
    - How to recognize and deal with punctuation
      - Apostrophes
      - Hyphens
      - Periods

Preliminary Processing Required: (cont'd)

- Word segmentation
  - No white space in Japanese language
  - Compound words – *"Lebensversicherungsgesellschaftsangestellter"*
- Additional issues if OCR'd data or speech transcripts
- Morphology (To stem or not to stem?)
  - Depends on the application

Preliminary Processing Required: (cont'd)

- May also find additional properties of the text:
  - Mark up the text using chosen tag set:
    - Tokens (words, punctuation, etc.)
    - Part of Speech
    - Named entities
    - Phrase / Clause
    - Sentence
    - Semantics
      - Sense Identification
      - Referent Resolution tagging
    - Discourse Components

Word Counting in Corpora

- After corpus preparation, additional decisions
  - Ignore capitalization at beginning of sentence? Is "They" the same word as "they"?
  - Ignore other capitalization? is "Company" the same word as "company"
  - Stemming? Is "cat" the same word as "cats"
- Terminology for word occurrences:
  - Tokens – the total number of words
  - Distinct Tokens – the number of distinct words, not counting repetitions
    - The following sentence from the Brown corpus has 16 tokens and 14 distinct tokens: *They picnicked by the pool, then lay back on the grass and looked at the stars.*

Word Frequencies

- Count the number of each token appearing in the corpus (or sometimes single document)
- A frequency distribution is a list of all tokens with their frequency, usually sorted in the order of decreasing frequency

- Used to make "word clouds"
  - For example, http://www.tumblr.com/tagged/word+cloud
- Used for comparison and characterization of text
  - See the article on the State of the Union (SOTU)Speeches by Nate Silver

Zipf's Law

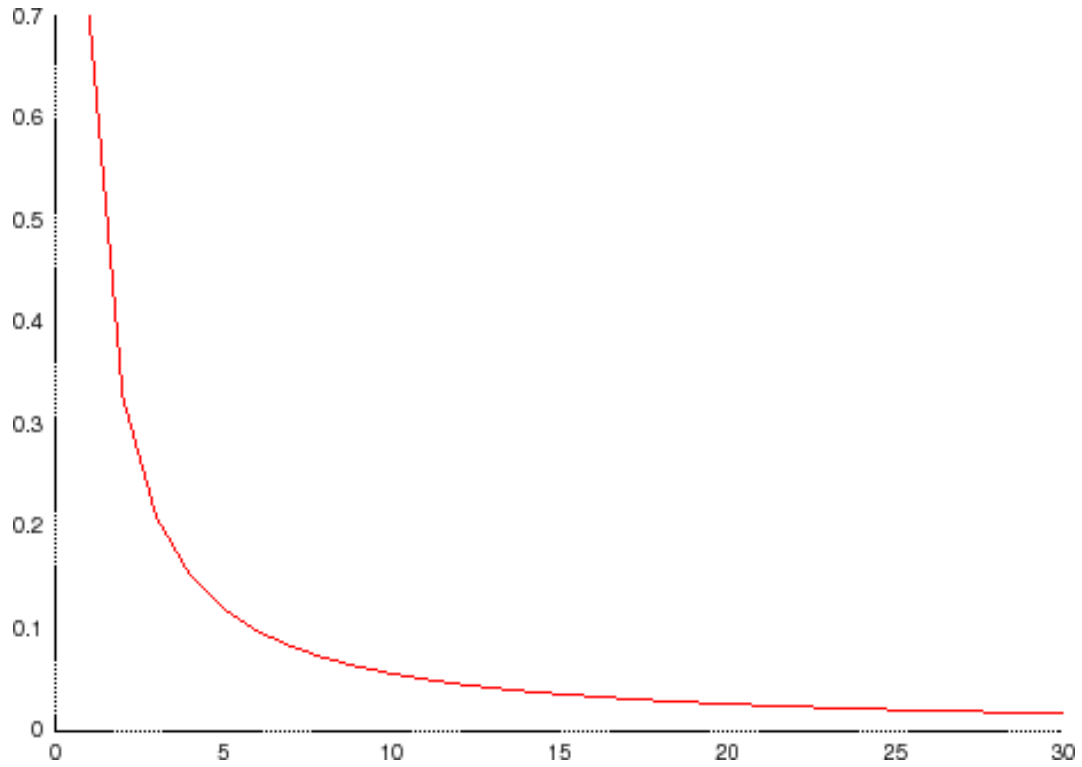
- Rank (r): The numerical position of a word in a list sorted by decreasing frequency (f).
- Zipf (1949) "discovered" that:

$$f \cdot r = k \quad (\text{for constant } k)$$

- If probability of word of rank r is p_r and N is the total number of word occurrences:

$$p_r = \frac{f}{N} = \frac{A}{r} \quad \text{for corpus indp. const. } A \approx 0.1$$

Zipf curve



A typical Zipf-law rank distribution. The y-axis represents word occurrence frequency, and the x-axis represents rank (highest at the left).

* Diagram from planetmath.org.

Zipf's Law Impact on Language Analysis

- Good News: Stopwords (commonly occurring words such as "the") will account for a large fraction of text so eliminating them greatly reduces size of vocabulary in a text

- Bad News: For most words, gathering sufficient data for meaningful statistical analysis (e.g. for correlation analysis for query expansion) is difficult since they are extremely rare.

N-Gram frequencies

- N-grams are used informally as a sequence of N objects, almost always words, in a row
  - Examples of bigrams are any two words that occur together
    - In the text: “two great and powerful groups of nations”, the bigrams are “two great”, “great and”, etc.
- The frequency of an n-gram is the percentage of times the n-gram occurs in all the n-grams of the corpus
  - For bigram xy:
    - Count of bigram xy / Count of all bigrams in corpus

Google N-Gram Release

All Our N-gram are Belong to You

By Peter Norvig - 8/03/2006 11:26:00 AM

Posted by Alex Franz and Thorsten Brants, Google Machine Translation Team

Here at Google Research we have been using word n-gram models for a variety of R&D projects, such as statistical machine translation, speech recognition, spelling correction, entity detection, information extraction, and others. While such models have usually been estimated from training

to share this enormous dataset with everyone. We processed 1,024,908,267,229 words of running text and are publishing the counts for all 1,176,470,663 five-word sequences that appear at least 40 times. There are 13,588,391 unique words, after discarding words that appear less than 200 times.

Google n-gram viewer

- In 2010, Google placed on on-line n-gram viewer that would display graphs of n-gram frequencies of one or more n-grams, based on a corpus defined from Google Books
  - http://ngrams.googlelabs.com/
  - And see also the "About Google Books NGram Viewer" link

N-gram Models

- The n-gram model uses the previous N – 1 '*things*' to predict the next one
  - Can be letters, words, parts-of-speech, etc
- Based on context-sensitive likeliness of occurrence
- We use n-gram word prediction more frequently than we are aware
  - Finishing someone else's sentence for them
  - Deciphering hard-to-read handwriting
  - Correcting spelling mistakes
- *Do all sounds / letters / words have an equal likelihood of being the next sound / letter / word in a particular situation?*
  - No! so incorporate normal frequency of occurrence
    - The relative frequency used to assign a probability distribution across potential next words

N-gram Models

- Look not just at individual relative frequencies, but at **conditional probability of a word given the previous words**

- Not really possible for long sequences, so we simplify and approximate
  - Probability given just the single 1 or 2 previous words
  - Known as the Markov assumption

- **Bigram Model** – approximates the probability of a word given all the previous words by the conditional probability of the preceding word

N-gram Essential Facts

1. N-gram models become increasingly accurate as the value of N is increased
   - Quadrigams are more accurate than Trigrams, which are more accurate than Bigrams, but are seldom used because of the computational cost and the scarcity of examples of the longer length

2. N-gram models are very dependent on their corpus, in terms of:
   - Genre
   - Size

N-Gram probabilities

- For N-Grams, we need the conditional probability:

  P(<next word> | <preceding word sequence of length n>)

  e.g. P (*the* | *They picnicked by*)

- We define this as
  - the observed frequency (count) of the whole sequence divided by
  - the observed frequency of the preceding, or initial, sequence (sometimes called the maximum likelihood estimation (MLE):

  P(<next word> | <preceding word sequence of length n>)
  = Count (<preceding word sequence> <next word>)
  / Count (<preceding word sequence>)
  Count (*They picnicked by the*) / Count (*They picnicked by*)

- Normalization
  - divide the ratio, aka the relative frequency, by the total word counts to get a probability between 0 and 1

Markov Assumption

- Counting all the long sequences in a corpus is too compute intensive, so we make the assumption that we can predict the next word based on a short piece of the past
- Bigrams often used.
  - Instead of computing

    $$P(\textit{the} \mid \textit{They picnicked by})$$

    we estimate this probability with the bigram

    $$P(\textit{the} \mid \textit{by})$$

Example of Bigram probabilities

- Example mini-corpus of three sentences, where we have sentence detection and we include the sentence tags in order to represent the beginning and end of the sentence.

  <S> I am Sam </S>

  <S> Sam I am </S>

  <S> I do not like green eggs and ham </S>

- Bigram probabilities:

  P (I | <S>) = 2/3 = .67 (probability that I follows <S>)

  P (</S> | Sam) = ½ = .5

  P (Sam | <S>) = 1/3 = .33

  P (Sam | am) = ½ = .5

  P (am | I) = 2/3 = .67

Using N-Grams for sentences

- For a bigram grammar $\prod_{k=1}^{n} P(w_k \mid w_{k-1})$
  - –P(sentence) can be approximated by multiplying all the bigram probabilities in the sequence
- Example:

P(I want to eat Chinese food) =

P(I | <S>) P(want | I) P(to | want) P(eat | to)
P(Chinese | eat) P(food | Chinese)

Bigrams from the restaurant corpus

Eat on	.16	Eat Thai	.03
Eat some	.06	Eat breakfast	.03
Eat lunch	.06	Eat in	.02
Eat dinner	.05	Eat Chinese	.02
Eat at	.04	Eat Mexican	.02
Eat a	.04	Eat tomorrow	.01
Eat Indian	.04	Eat dessert	.007
Eat today	.03	Eat British	.001

Examples due to Rada Mihalcea

Additional Bigrams

<S> I	.25	Want some	.04
<S> I'd	.06	Want Thai	.01
<S> Tell	.04	To eat	.26
<S> I'm	.02	To have	.14
I want	.32	To spend	.09
I would	.29	To be	.02
I don't	.08	British food	.60
I have	.04	British restaurant	.15
Want to	.65	British cuisine	.01
Want a	.05	British lunch	.01

Computing Sentence Probabilities

- P(I want to eat British food) = P(I|<S>) P(want|I) P (to|want) P(eat|to) P(British|eat) P(food|British) = .25×.32×.65×.26×.001×.60 = .000080
- vs.
- P(I want to eat Chinese food) = .00015
- Probabilities seem to capture “syntactic" facts, “world knowledge"
  - eat is often followed by a NP
  - British food is not too popular
- N-gram models can be trained by counting and normalization

In-Class Exercise

Smoothing Techniques

- Every N-gram training matrix is sparse, even for very large corpora (remember Zipf's law)
  - There are words that don't occur in the training corpus that may occur in future text
  - These are known as the unseen words

- Solution: estimate the likelihood of unseen N-grams

Smoothing

- Add-one smoothing
  - Given: $P(w_n|w_{n-1}) = C(w_{n-1}w_n)/C(w_{n-1})$
  - Add 1 to each count: $P(w_n|w_{n-1}) = [C(w_{n-1}w_n) + 1] / [C(w_{n-1}) + V]$
- Backoff Smoothing for higher-order N-grams
  - Notice that:
    - N-grams are more precise than (N-1)grams
    - But also, N-grams are more sparse than (N-1) grams
  - How to combine things?
    - Attempt N-grams and back-off to (N-1) if counts are not available
    - E.g. attempt prediction using 4-grams, and back-off to trigrams (or bigrams, or unigrams) if counts are not available

N-gram Application - Spell Correction

- Frequency of spelling errors in human typed text varies
  - 0.05% of the words in carefully edited journals
  - 38% in difficult applications like telephone directory lookup
- Optical character recognition
  - Higher **error** rates than human typists
  - Make different kinds of errors

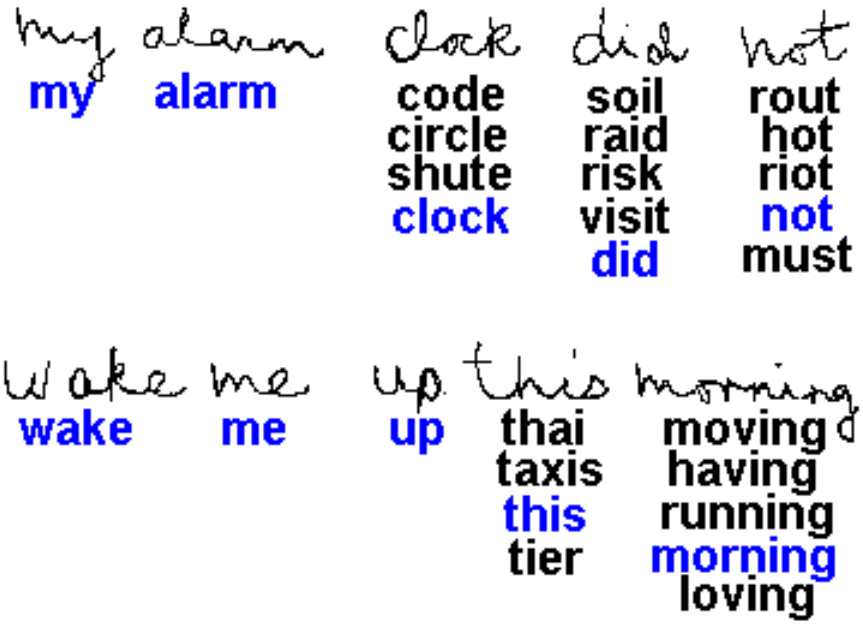
Spelling Error Detection & Correction Types

- **Non-word error detection**
  - Detecting spelling errors that result in non-words
  - *mesage* -> *message* by looking only at the **word** in isolation
  - Simple lexicon look-up
  - May fail to recognize an error (**real-word errors**)
    - Typographical errors e.g. *there* for *three*
    - Homonym or near-homonym e.g. *dessert* for *desert*, or *piece* for *peace*

- **Context-dependent error detection** and correction
  - Using the context to detect and correct spelling errors
  - Real-word errors are most difficult and would benefit most from n-gram analysis which utilizes context

N-gram Analysis of Handwritten Sentence

- Results of offline isolated word recognition
- Lists of up to top 5 choices of the handwritten word recognizer, with correct choice highlighted
- Using language models with collocational (*alarm clock*) & syntactic (POS) information, correct sentence is extracted:



Language Models

- Language models are a closely related idea that use the N-Gram prediction of the next word to predict sequences of words
- N-Gram language models were first used in large vocabulary speech recognition systems to
  - Provide the recognizer with an a-priori likelihood **P(W)** of a given word sequence **W**.
- The N-Gram language model is usually derived from large training texts that share the same language characteristics as the expected input.
  - This information complements the acoustic model that models the articulatory features of the speakers.
  - Together, these two components allow a system to compute the most likely input sequence
    W' = argmax$_W$ P(W|O), where **O** is the input signal observations
    as **W' = argmax$_W$ P(O|W) P(W).**

A Statistical Technique: Mutual Information (MI)

- N-Gram probabilities predict the next word – Mutual Information computes probability of two words occurring in sequence
- A technique for determining which co-occurrences of words are significant collocations.
  - Based on corpus statistics
  - MI is borrowed from information theory
- Given a pair of words, compares probability that the two occur together as a joint event to the probability they occur individually & that their co-occurrences are simply the result of chance
  - The more strongly connected 2 items are, the higher will be their MI value

Association Ratio
(informally also called Mutual Information)

- Based on work of Church & Hanks (1990), generalizing MI to apply to words in sequence
- P(x) and P(y) are estimated by the number of observations of x and y in a corpus and normalized by N, the size of the corpus
- P(x,y) is estimated by the number of times that x is followed by y in a window of w words
- Mutual Information:

$$I(x,y) = \log_2 \left(P(x,y) / P(x)\,P(y) \right)$$

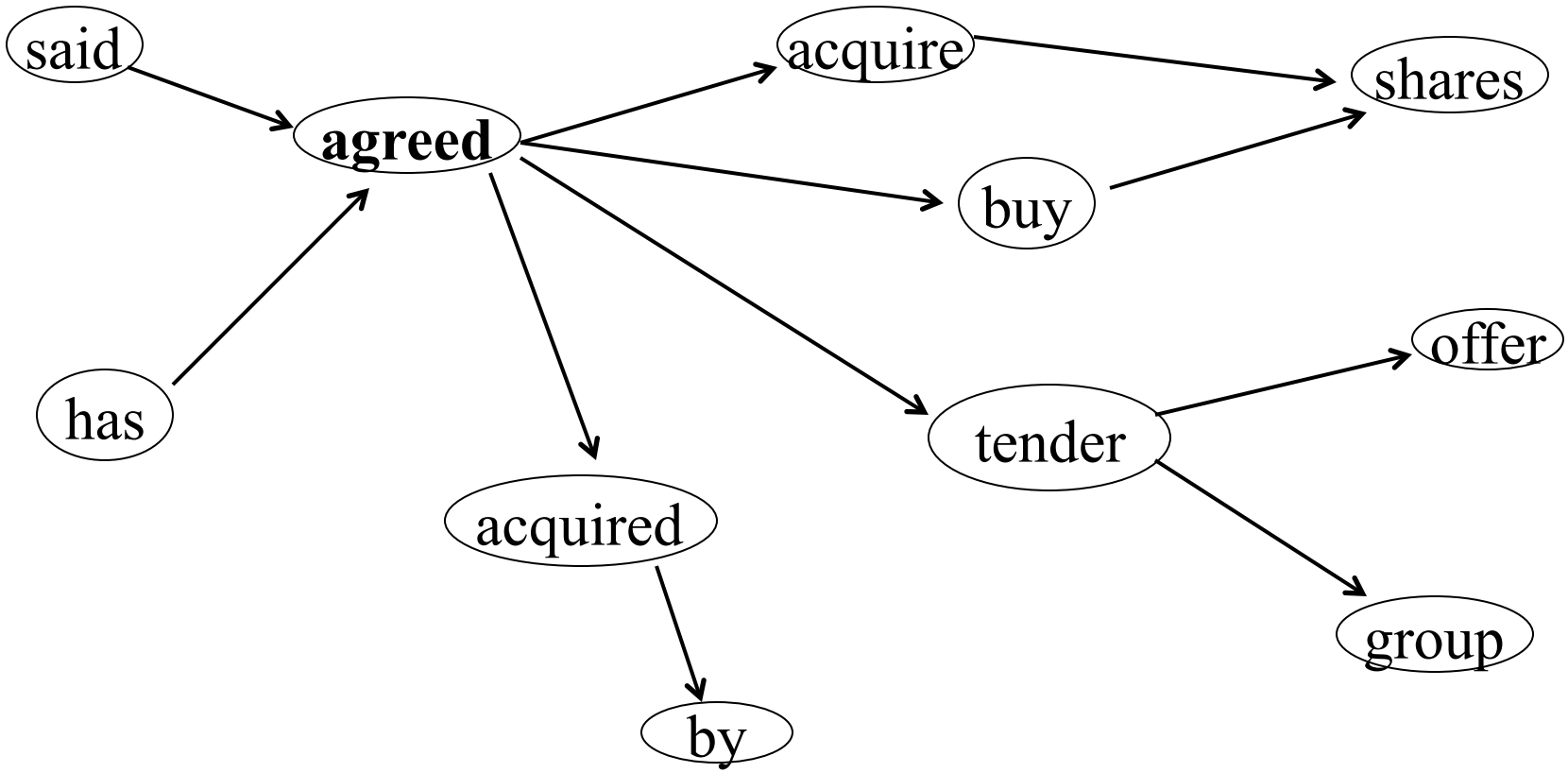
Association Ratio
(informally also called Mutual Information)

- If there is no relationship between words x and y, then I(x,y) is the same as P(x) P(y), i.e. the probability that they occur together by chance
  - Values can be negative or positive

-	**0**	+
Complementary	Chance	Highly Assoc.

MI values based on 145 WSJ articles

x	freq (x)	y	freq (y)	freq (x,y)	MI
Gaza	3	Strip	3	3	14.42
joint	8	venture	4	4	13.00
Chapter	3	11	14	3	12.20
credit	15	card	11	7	11.44
average	22	yield	7	5	11.06
appeals	4	court	47	4	10.45
…..					
said	444	it	346	76	5.02



Highest word associations with *agreed* in WSJ corpus

Uses of Mutual Information or Assocation Ratio

- Lexicographic analysis for dictionary development
- Facilitate development of features to be captured in symbolic applications
  - Idiomatic phrases for MT
  - Semantic word classes for query expansion
  - Lexical candidates for Case Role frames for detecting relations
- Sense disambiguation (both statistical and symbolic approaches)
- Error detection & correction in speech analysis and spell-checking