Querying

Introduction to Information Retrieval
INF 141
Donald J. Patterson

Content adapted from Hinrich Schütze
http://www.informationretrieval.org

Full text queries

- To use zone combinations for free text queries, we need:
  - A way of scoring = Score(full-text-query, zone)
  - Zero query terms in zone -> zero score
  - More query terms in a zone -> higher score
  - Scores don't have to be boolean (0 or 1) anymore
- Let's look at the alternatives...

Building up our query technology

- "Matching" search
  - Linear on-demand retrieval (aka grep)
  - 0/1 Vector-Based Boolean Queries
  - Posting-Based Boolean Queries
- Ranked search
  - Parametric Search
  - Zones
  - Scoring
  - Term Frequency Matrices

Incidence Matrices

- Recall how a document, d, (or a zone) is a (0,1) column vector
  - A query, q, is also a column vector. How so?

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Anthony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0
...						

Incidence Matrices

- Using this formalism, score can be overlap measure:

$$|q \cap D|$$

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Anthony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0
...						

Incidence Matrices

- Example:
  - Query "ides of march"
  - Shakespeare's "Julius Caesar" has a score of 3
  - Plays that contain "march" and "of" score 2
  - Plays that contain "of" score 1
- Algorithm:
  - Bitwise-And between q and matrix, D
  - Column summation
  - Sort

Incidence Matrices

Incidence Matrices

- What is wrong with the overlap measure?

Incidence Matrices

- What is wrong with the overlap measure?
- It doesn't consider:

Incidence Matrices

- What is wrong with the overlap measure?
- It doesn't consider:
  - Term frequency in a document

Incidence Matrices

- What is wrong with the overlap measure?
- It doesn’t consider:
  - Term frequency in a document
  - Term scarcity in corpus

Incidence Matrices

- What is wrong with the overlap measure?
- It doesn't consider:
  - Term frequency in a document
  - Term scarcity in corpus
    - "ides" is much rarer than "of"

Incidence Matrices

- What is wrong with the overlap measure?
- It doesn't consider:
  - Term frequency in a document
  - Term scarcity in corpus
    - "ides" is much rarer than "of"
  - Length of a document

Incidence Matrices

- What is wrong with the overlap measure?
- It doesn’t consider:
  - Term frequency in a document
  - Term scarcity in corpus
    - “ides” is much rarer than “of”
  - Length of a document
  - Length of queries

Toward better scoring

- Overlap Measure
- Normalizing queries
  - Jaccard Coefficient
    - Score is number of words that overlap divided by total number of words
    - What documents would score best?
  - Cosine Measure
    - Will the same documents score well?

Toward better scoring

$$|q \cap d|$$

- Overlap Measure
- Normalizing queries
  - Jaccard Coefficient
    - Score is number of words that overlap divided by total number of words
    - What documents would score best?
  - Cosine Measure
    - Will the same documents score well?

Toward better scoring

- Overlap Measure $|q \cap d|$
- Normalizing queries
  - Jaccard Coefficient $\dfrac{|q \cap d|}{|q \cup d|}$
    - Score is number of words that overlap divided by total number of words
    - What documents would score best?
  - Cosine Measure
    - Will the same documents score well?

Toward better scoring

- Overlap Measure
- Normalizing queries
  - Jaccard Coefficient
    - Score is number of words that overlap divided by total number of words
    - What documents would score best?
  - Cosine Measure
    - Will the same documents score well?

$$|q \cap d|$$

$$\frac{|q \cap d|}{|q \cup d|}$$

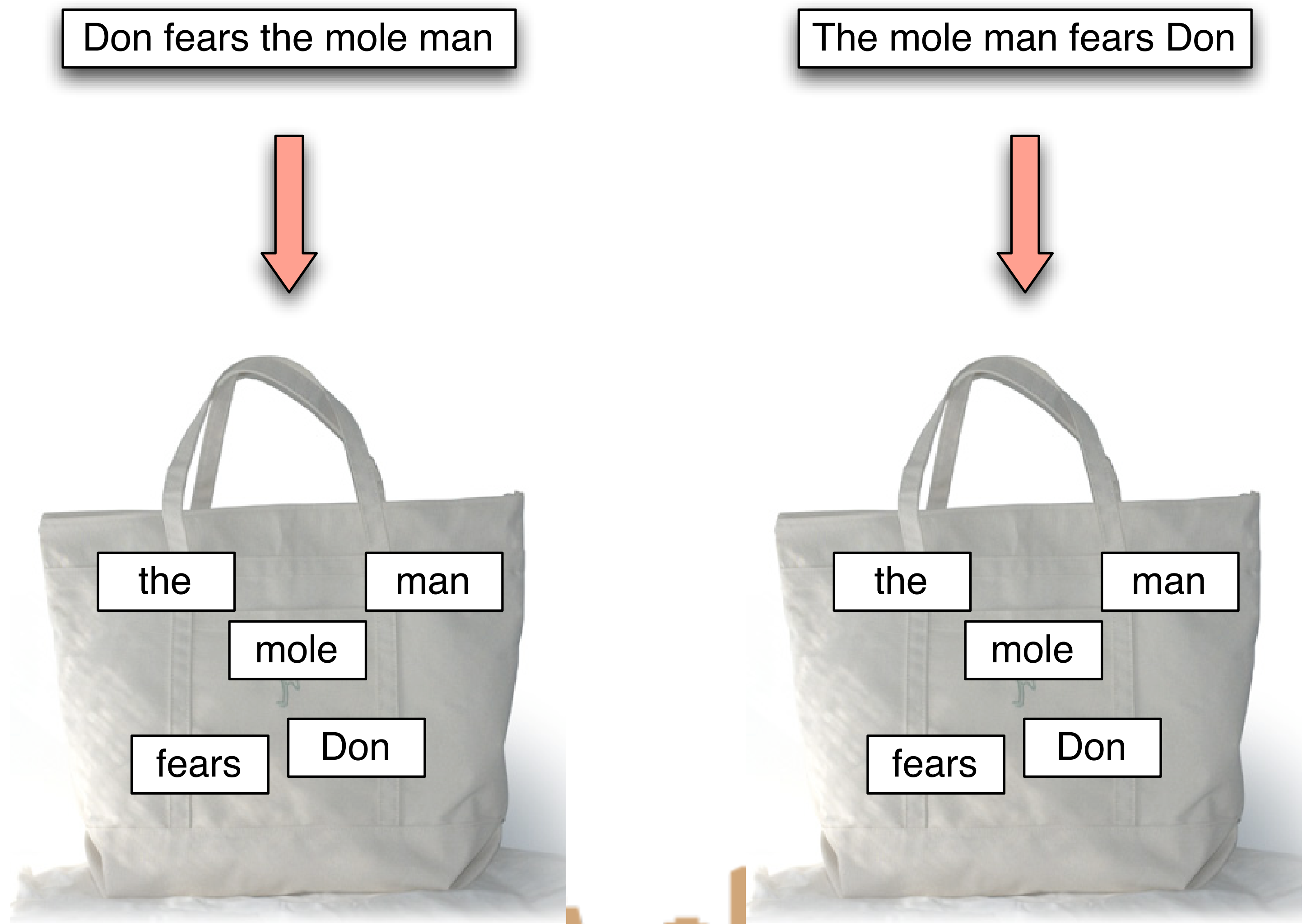
$$\frac{|q \cap d|}{\sqrt{|q||d|}}$$

Toward Better Scoring

- Scores so far capture position (zone) and overlap
- Next step: a document which talks about a topic should be a better match
    - Even when there is a single term in the query
    - Document is relevant if the term occurs a lot
    - This brings us to term weighting

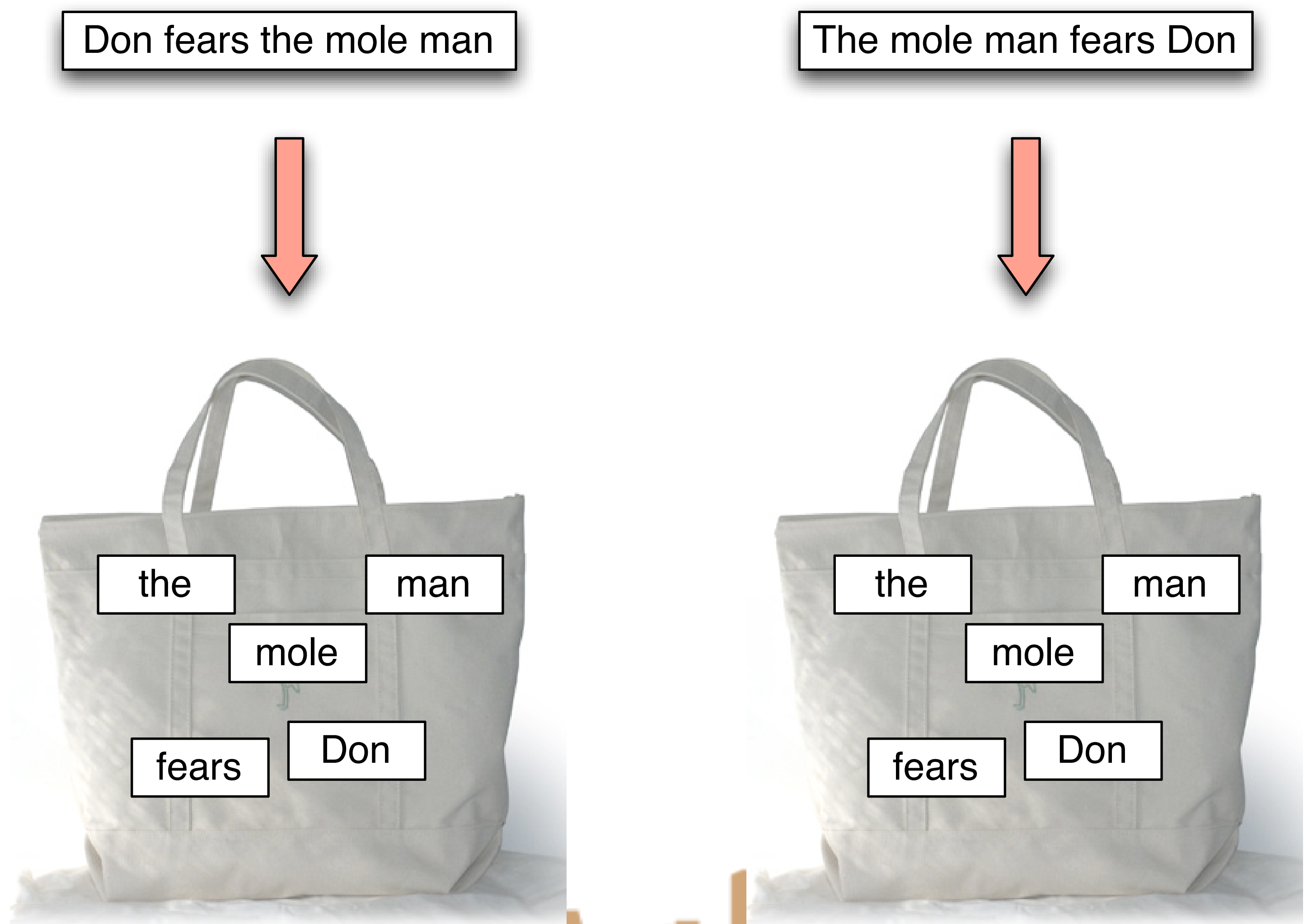
Bag of Words Model

- “Don fears the mole man” equals “The mole man fears Don”
- The incidence matrix for both looks the same



Bag of Words Model

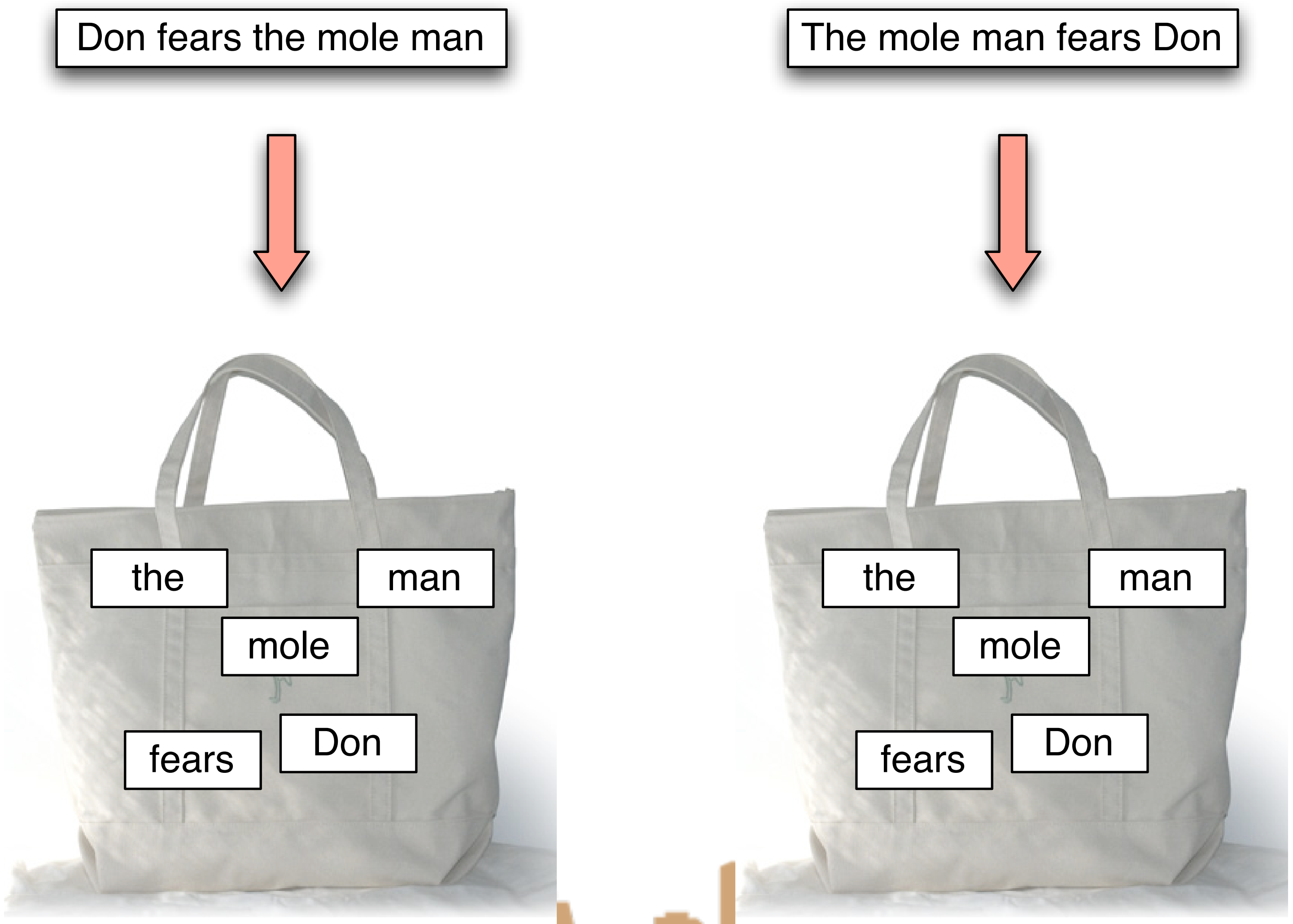
- "Don fears the mole man" equals "The mole man fears Don"
- The incidence matrix for both looks the same



Querying

Bag of Words Model

- "Don fears the mole man" equals "The mole man fears Don"
- The incidence matrix for both looks the same



	d_1	d_2
Don	1	1
$fears$	1	1
man	1	1
$mole$	1	1
$mule$	0	0
the	1	1
zoo	0	0

Term Frequency Matrix

- Bag of words
- Document is vector with integer elements

	Antony and Cleopatra	*Julius Caesar*	*The Tempest*	*Hamlet*	*Othello*	*Macbeth*
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

Term Frequency - tf

- Long documents are favored because they are more likely to contain query terms
- Reduce the impact by normalizing by document length
- Is raw term frequency the right number?

Weighting Term Frequency - WTF

Weighting Term Frequency - WTF

- What is the relative importance of
  - 0 vs. 1 occurrence of a word in a document?
  - 1 vs. 2 occurences of a word in a document?
  - 2 vs. 100 occurences of a word in a document?

Weighting Term Frequency - WTF

- What is the relative importance of
  - 0 vs. 1 occurrence of a word in a document?
  - 1 vs. 2 occurences of a word in a document?
  - 2 vs. 100 occurences of a word in a document?
- Answer is unclear:
  - More is better, but not proportionally
  - An alternative to raw tf:

Weighting Term Frequency - WTF

- What is the relative importance of
  - 0 vs. 1 occurrence of a word in a document?
  - 1 vs. 2 occurences of a word in a document?
  - 2 vs. 100 occurences of a word in a document?
- Answer is unclear:
  - More is better, but not proportionally
  - An alternative to raw tf:

$$
\begin{array}{l}
\mathrm{WTF}(t,d) \\
\quad \textbf{if } tf_{t,d} = 0 \\
\quad\quad \textbf{then } return(0) \\
\quad\quad \textbf{else } \; return(1 + log(tf_{t,d}))
\end{array}
$$

Weighting Term Frequency - WTF

Weighting Term Frequency - WTF

- The score for query, q, is

Weighting Term Frequency - WTF

- The score for query, q, is
    - Sum over terms, t

Weighting Term Frequency - WTF

- The score for query, q, is
    - Sum over terms, t

$$\mathrm{WTF}(t, d)$$

$$\textbf{if } tf_{t,d} = 0$$

$$\quad \textbf{then } return(0)$$

$$\quad \textbf{else } \; return(1 + log(tf_{t,d}))$$

Weighting Term Frequency - WTF

- The score for query, q, is
  - Sum over terms, t

$$
\begin{array}{l}
\mathrm{WTF}(t, d) \\
\quad \textbf{if } tf_{t,d} = 0 \\
\quad\quad \textbf{then } return(0) \\
\quad\quad \textbf{else } \; return(1 + log(tf_{t,d}))
\end{array}
$$

$$Score_{WTF}(q, d) = \sum_{t \in q} (WTF(t, d))$$

Weighting Term Frequency - WTF

- The score for query, q, is
  - Sum over terms, t

$$\mathrm{WTF}(t,d)$$

1. **if** $tf_{t,d} = 0$
2. **then** $return(0)$
3. **else** $return(1 + log(tf_{t,d}))$

$$Score_{WTF}(q,d) = \sum_{t \in q}(WTF(t,d))$$

$$\begin{aligned}
Score_{WTF}(\text{"}bill\ rights\text{"}, declarationOfIndependence) &= \\
WTF(\text{"}bill\text{"}, declarationOfIndependence) &+ \\
WTF(\text{"}rights\text{"}, declarationOfIndependence) &= \\
0 + 1 + log(3) &= 1.48
\end{aligned}$$

Weighting Term Frequency - WTF

$$Score_{WTF}(q,d) = \sum_{t \in q} (WTF(t,d))$$

Weighting Term Frequency - WTF

$$Score_{WTF}(q,d) = \sum_{t \in q} (WTF(t,d))$$

$$\begin{aligned} Score_{WTF}(\text{"}bill\ rights\text{"}, declarationOfIndependence) &= \\ WTF(\text{"}bill\text{"}, declarationOfIndependence) &+ \\ WTF(\text{"}rights\text{"}, declarationOfIndependence) &= \\ 0 + 1 + log(3) &= 1.48 \end{aligned}$$

Weighting Term Frequency - WTF

$$Score_{WTF}(q,d) = \sum_{t \in q} (WTF(t,d))$$

$$\begin{aligned}
Score_{WTF}(\text{"bill rights"}, declarationOfIndependence) &= \\
WTF(\text{"bill"}, declarationOfIndependence) &+ \\
WTF(\text{"rights"}, declarationOfIndependence) &= \\
0 + 1 + log(3) &= 1.48
\end{aligned}$$

$$\begin{aligned}
Score_{WTF}(\text{"bill rights"}, constitution) &= \\
WTF(\text{"bill"}, constitution) &+ \\
WTF(\text{"rights"}, constitution) &= \\
1 + log(10) + 1 + log(1) &= 3
\end{aligned}$$

Weighting Term Frequency - WTF

- Can be zone combined:

$$\begin{aligned} Score \quad = \quad & 0.6(Score_{WTF}(''instant\ oatmeal\ health", d.title) + \\ & 0.3(Score_{WTF}(''instant\ oatmeal\ health", d.body) + \\ & 0.1(Score_{WTF}(''instant\ oatmeal\ health", d.abstract) \end{aligned}$$

- Note that you get 0 if there are no query terms in the document.
  - Is that really what you want?
  - We will eventually address this

Unsatisfied with term weighting

Unsatisfied with term weighting

- Which of these tells you more about a document?
  - 10 occurrences of “mole”
  - 10 occurrences of “man”
  - 10 occurrences of “the”

Unsatisfied with term weighting

- Which of these tells you more about a document?
  - 10 occurrences of "mole"
  - 10 occurrences of "man"
  - 10 occurrences of "the"
- It would be nice if common words had less impact
  - How do we decide what is common?

Unsatisfied with term weighting

- Which of these tells you more about a document?
  - 10 occurrences of “mole”
  - 10 occurrences of “man”
  - 10 occurrences of “the”
- It would be nice if common words had less impact
  - How do we decide what is common?
- Let’s use corpus-wide statistics

Corpus-wide statistics

Corpus-wide statistics

- Collection Frequency, cf
  - Define: The total number of occurences of the term in the entire corpus

Corpus-wide statistics

- Collection Frequency, cf
  - Define: The total number of occurences of the term in the entire corpus
- Document Frequency, df
  - Define: The total number of documents which contain the term in the corpus

Corpus-wide statistics

Word	*Collection Frequency*	*Document Frequency*
insurance	10440	3997
try	10422	8760

Corpus-wide statistics

Word	*Collection Frequency*	*Document Frequency*
insurance	10440	3997
try	10422	8760

- This suggests that df is better at discriminating between documents

Corpus-wide statistics

Word	*Collection Frequency*	*Document Frequency*
insurance	10440	3997
try	10422	8760

- This suggests that df is better at discriminating between documents
- How do we use df?

Corpus-wide statistics

Corpus-wide statistics

- Term-Frequency, Inverse Document Frequency Weights

Corpus-wide statistics

- Term-Frequency, Inverse Document Frequency Weights
  - “tf-idf”

Corpus-wide statistics

- Term-Frequency, Inverse Document Frequency Weights
  - "tf-idf"
  - tf = term frequency

Corpus-wide statistics

- Term-Frequency, Inverse Document Frequency Weights
  - “tf-idf”
  - tf = term frequency
    - some measure of term density in a document

Corpus-wide statistics

- Term-Frequency, Inverse Document Frequency Weights
  - "tf-idf"
  - tf = term frequency
    - some measure of term density in a document
  - idf = inverse document frequency

Corpus-wide statistics

- Term-Frequency, Inverse Document Frequency Weights
  - "tf-idf"
  - tf = term frequency
    - some measure of term density in a document
  - idf = inverse document frequency
    - a measure of the informativeness of a term

Corpus-wide statistics

- Term-Frequency, Inverse Document Frequency Weights
  - "tf-idf"
  - tf = term frequency
    - some measure of term density in a document
  - idf = inverse document frequency
    - a measure of the informativeness of a term
    - it's rarity across the corpus

Corpus-wide statistics

- Term-Frequency, Inverse Document Frequency Weights
  - “tf-idf”
  - tf = term frequency
    - some measure of term density in a document
  - idf = inverse document frequency
    - a measure of the informativeness of a term
    - it’s rarity across the corpus
    - could be just a count of documents with the term

Corpus-wide statistics

- Term-Frequency, Inverse Document Frequency Weights
  - “tf-idf”
  - tf = term frequency
    - some measure of term density in a document
  - idf = inverse document frequency
    - a measure of the informativeness of a term
    - it’s rarity across the corpus
    - could be just a count of documents with the term
    - more commonly it is:

Corpus-wide statistics

- Term-Frequency, Inverse Document Frequency Weights
  - "tf-idf"
  - tf = term frequency
    - some measure of term density in a document
  - idf = inverse document frequency
    - a measure of the informativeness of a term
    - it's rarity across the corpus
    - could be just a count of documents with the term
    - more commonly it is: $$idf_t = log\left(\frac{|corpus|}{df_t}\right)$$

TF-IDF Examples

$$idf_t = log\left(\frac{|corpus|}{df_t}\right) \qquad idf_t = log_{10}\left(\frac{1,000,000}{df_t}\right)$$

$term$	df_t	idf_t
$calpurnia$	1	6
$animal$	10	4
$sunday$	1000	3
fly	$10,000$	2
$under$	$100,000$	1
the	$1,000,000$	0

TF-IDF Summary

- Assign tf-idf weight for each term t in a document d:

$$tfidf(t,d) \quad = \quad \underbrace{WTF(t,d)}_{(1+log(tf_{t,d}))} * log\left(\frac{|corpus|}{df_{t,d}}\right)$$

- Increases with number of occurrences of term in a doc.
- Increases with rarity of term across entire corpus
- Three different metrics
  - term frequency
  - document frequency
  - collection/corpus frequency

Now, real-valued term-document matrices

- Bag of words model
- Each element of matrix is tf-idf value

	Antony and Cleopatra	*Julius Caesar*	*The Tempest*	*Hamlet*	*Othello*	*Macbeth*
Antony	13.1	11.4	0.0	0.0	0.0	0.0
Brutus	3.0	8.3	0.0	1.0	0.0	0.0
Caesar	2.3	2.3	0.0	0.5	0.3	0.3
Calpurnia	0.0	11.2	0.0	0.0	0.0	0.0
Cleopatra	17.7	0.0	0.0	0.0	0.0	0.0
mercy	0.5	0.0	0.7	0.9	0.9	0.3
worser	1.2	0.0	0.6	0.6	0.6	0.0

Fix this slide so that the numbers are correct with the previous slide

Vector Space Scoring

- That is a nice matrix, but
  - How does it relate to scoring?
  - Next, vector space scoring