CAS CS 505
Lec 5

Conclusion to N-Grams; Introduction to Vector Models: BOW, Distance Metrics, TF-IDF, PCA

1. Zero probability bigrams
    a. Bigrams with zero probability
        i. mean that we will assign 0 probability to the test set!
    b. And hence we cannot compute perplexity (can't divide by 0)!
2. Another solution to 0 N-gram counts: Backoff
    a. Another solution to 0 counts for N-Grams is to use less left context:
        i. This is called Backoff; suppose you have a 4-Gram model:
            1. use quadrigram if you have "good evidence,"
            2. use trigram if you have "good evidence," otherwise bigram,
            3. if you have "good evidence," else unigram (single word),
            4. if you have "good evidence."
            5. If you have a word in test set that does not occur in the training set, use # occurrences of word in corpus / size of corpus
    b. In the testing set:
        i. <s> John likes to play Beethoven symphonies on a harmonica …
    c. In the training set:
        i. 0 occurrences of "symphonies on a harmonica"
        ii. 1 occurrence of "on a harmonica"
        iii. 5 occurrences of "a harmonica"
        iv. 10 occurrences of "harmonica"
    d. What is "good evidence" for the probability of "harmonica" following "symphonies on a"? How to calculate this probability using the evidence?
    e. Interpolation: Weighted sum of probabilities for each of trigram, bigram, and unigram; weights can be estimated, or learned from training set and choose the one with the least perplexity
    f. Stupid Backoff: Multiply (recursively) by a constant "discount factor" 0.4
        i. P("symphonies on a harmonica") = 0.4 * P("on a harmonica")
3. N-gram Smoothing Summary
    a. Used to deal with missing data
    b. Add-1 smoothing:
        i. OK for text categorization, not for language modeling
    c. Backoff and Interpolation
        i. Use thresholds for counts § Learn weights for interpolation
    d. Combination approaches
        i. Extended Interpolated Kneser-Ney (state of the art, covered in the text)
    e. BUT: when you have enough data, doesn't matter which you use…

4. Vector Models
    a. A BOW model represents a text by a vector of frequency counts over all the vocabulary
    b.

Vocabulary

BOW vector for text Julius Caesar

Texts

	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

5. BOW = Term frequency vector
    a. The term frequency TFt,d of term t in document d is defined as the number of times that t occurs in d.
    b. We want to use TF when using BOW vectors in NLP tasks such as
        i. Queries: Which play talks about religion the most?
        ii. Classification: Which plays are tragedies and which are comedies?
        iii. Similarity: Which play is most similar to Julius Caesar?
    c. There are some words that are meaningless – called stop words such as “the”, “is”
    d. Raw term frequency is not what we want:
        i. A document with 10 occurrences of the term is more relevant than a document with 1 occurrence of the term.
        ii. But not 10 times more relevant.
    e. Relevance does not increase proportionally with term frequency.
6. Document frequency
    a. Rare terms are more informative than frequent terms
        i. Recall stop words
    b. Consider a term in the query that is rare in the collection (e.g., arachnocentric)
    c. A document containing this term is very likely to be relevant to the query “What plays are arachnocentric?”
    d. → We want a high weight for rare terms like arachnocentric.
    e. Frequent terms are less informative than rare terms
    f. Consider a query term that is frequent in the collection (e.g., high, increase, line)
    g. A document containing such a term is more likely to be relevant than a document that doesn’t
    h. But it’s not a sure indicator of relevance.
    i. → For frequent terms, we want high positive weights for words like high, increase, and line
    j. But lower weights than for rare terms.
    k. We will use document frequency DF to capture this

7. IDF: Inverse Document Frequency
    a. DFt is the document frequency of a token t = the number of documents that contain t
        i. DFt is an inverse measure of the informativeness of t
        ii. DFt £ N (length of corpus in tokens)
    b. We define the IDF (inverse document frequency) of t by
        i. $$\mathrm{IDF}_t = \log_{10}\left(\frac{N}{DF_t}\right)$$
    c. We use log (N/DFt) instead of N/DFt to "dampen" the effect of IDF.
8. IDF Example: suppose N = 1 million
    a.

term	DF_t	IDF_t
calpurnia	10	1,000,000
animal	100	500,000
sunday	1,000	333,333.33
fly	10,000	250,000
under	100,000	200,000
the	1,000,000	166666.66

    b. There is one idf value for each term t in a collection
9. TF-IDF weighting
    a. The tf-idf weight of a term is the product of its tf weight and its idf weight.
        i. $$\mathrm{w}_{t,d} = \log(1+\mathrm{tf}_{t,d}) \times \log_{10}(N/\mathrm{df}_t)$$
    b. Best known weighting scheme in information retrieval
        i. Note: the "-" in TF-IDF is a hyphen, not a minus sign!
    c. Increases with the number of occurrences within a document
    d. Increases with the rarity of the term in the collection
    e. (Give rare words more weights than frequent words)
10. TF-IDF Vector Models
    a. The tf-idf weight of a term is the product of its tf weight and its idf weight.

$$\mathrm{w}_{t,d} = \log(1+\mathrm{tf}_{t,d}) \times \log_{10}(N/\mathrm{df}_t)$$

    b. Best known weighting scheme in information retrieval
        i. Note: the "-" in TF-IDF is a hyphen, not a minus sign!
    c. Increases with the number of occurrences within a document
    d. Increases with the rarity of the term in the collection

11. TF-IDF Vector Models
    a. Each document is now represented by a real-valued vector of tf-idf weights $\in R^{(|V|)}$

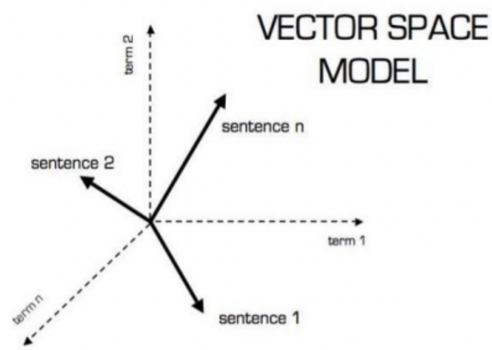
DF vector:

Julius Caesar
73
157
227
10
0
0
0

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

    b.

12. Documents as vectors (TF BOW, TF-IDF, etc…)
    a. So we have a |V|-dimensional vector space
        i. |V| = size of V, Vocabulary



    b.
    c. Terms are axes of the space
    d. Documents are points or vectors in this space
    e. Very high-dimensional: Number of dimensions = size of vocabulary, often 10,000 or more.
    f. These are very sparse vectors
        i. most entries are zero.
13. Fomalizing vector proximity
    a. Basic idea: Two documents are similar if their vectors are ”close” to each other = but how do we define “closeness”?
    b. First cut: distance between two points

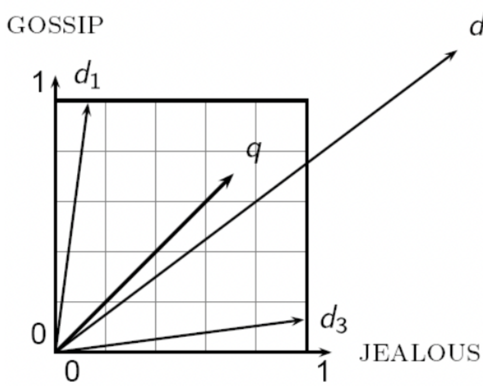
i. (= distance between the end points of the two vectors)

c. Euclidean distance?

d. Euclidean distance is a bad idea because Euclidean distance is large for vectors of different lengths.

14. Why distance is a bad idea

a. The Euclidean distance between q and d2 is large even though the distribution of terms in the query q and the distribution of terms in the document d2 are very similar.

b. 

c. It takes the size of the sentence into consideration, which is not important (the content is important)

15. Use angle instead of distance

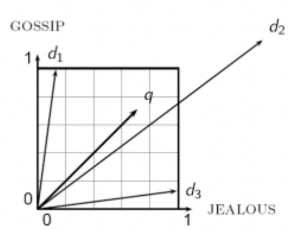
a. Thought experiment: take a document d and append it to itself. Call this document d′.

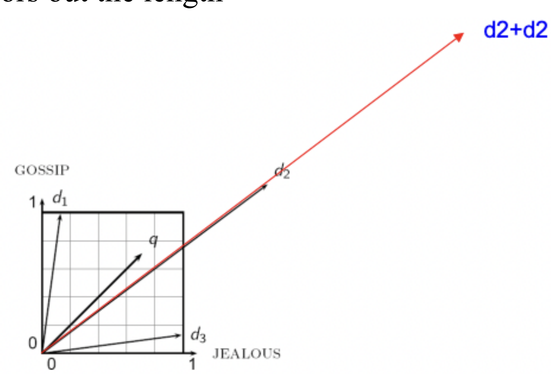
b. “Semantically” d and d′ have the same content

c. The Euclidean distance between the two documents can be quite large

d. The angle between the two documents is 0, corresponding to maximal similarity.

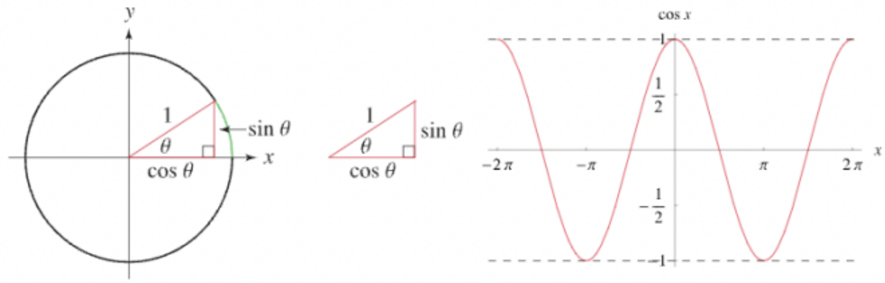
e. The proportions are the same → factors out the length

f. 



16. From angles to cosines

a. Instead of angle in degrees, we use cosine of the angle:
Cos 0 = 0 angle

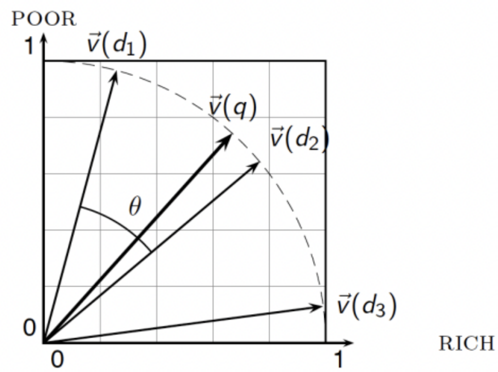


b.

17. Cosine Similiarity

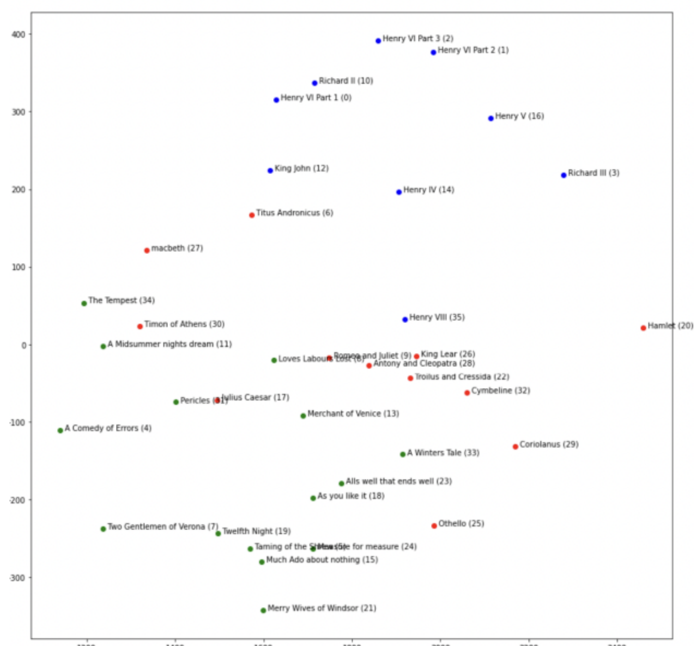
$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2 \cdot \sum\limits_{i=1}^{n} B_i^2}},$$

Cos	Angle	Meaning
1	0	Same
0	90/270	Orthogonal
-1	180	Opposite



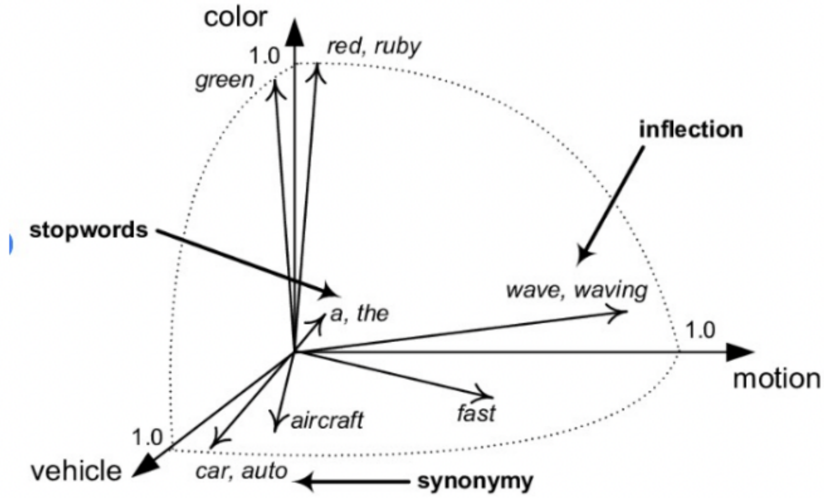
a.

18. Vector Space Models for Shakespeare



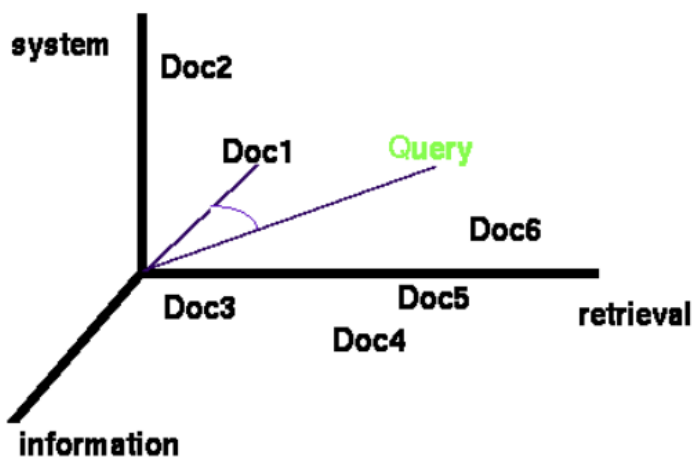
a.

19. Vector Space Models for Words



a.

20. Vector Space Models for Queries in Corpus



a.

21. Vector Space Models for Quantitative Models

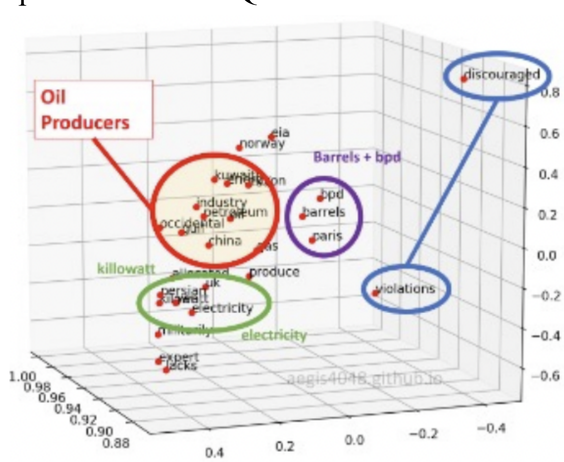


Figure 11: 3D plot - angle 2

a.