

CS6120: Lecture 12

Web Search

Machine Translation

Kenneth Church

<https://kwchurch.github.io/>

Your Projects

- Oral:
 - 20-40 minutes
- Written
 - 5-20 pages
 - lots of references
- Say everything three times
 - Winston: How to Speak
 - <https://www.youtube.com/watch?v=Unzc731iCUY>
- Examples:
 - <https://aclanthology.org/2022.acl-long.60>
 - http://34.204.188.58/cgi-bin/similar?embedding=s2_recommendations&limit=20&search=An+Information-theoretic+Approach+to+Prompt+Engineering+Without+Ground+Truth+Labels
 - <https://aclanthology.org/events/acl-2022/>



Web Search & Information Retrieval

Information Retrieval

- Textbook
 - <https://www-nlp.stanford.edu/IR-book/>
- Term-by-doc matrix: M
 - Similarity of terms: $M M^T$
 - Similarity of doc: $M^T M$
- Index: Inverted file (postings)
 - Rows of M
- Solitaire → Multiplayer Game
 - Solitaire: users lose to library (casino)
 - Multiplayer Game: Ecosystems
 - Readers, writers, advertizers, market makers

Web Search

- Left rail (algo) vs. right rail (ads)
 - Instant Answers
- Crawl: wget
 - Seed URLs → sample of “everything”
 - Deep web
 - WayBack Machine
 - <https://web.archive.org/>
- User logs: queries, clicks
 - Larger than crawls
 - Healthy ecosystem → more readers than writers
 - Behavioral signals
 - (super-important feedback)

Web Crawling

<https://nlp.stanford.edu/IR-book/pdf/20craw1.pdf>

- Expensive: Limited to a few large companies
- Prioritize URLs by coverage of
 - Popular pages (estimates of demand)
 - Freshness (estimates of when the content is likely to differ from what is in cache)
- How large is the web? [Spider traps!](#)
 - (Near duplicate pages & Approximate nearest neighbors)
 - Calendar pages are not spider traps, but...
 - Ads, view counters
 - <https://www.wysiwygwebbuilder.com/phpcounter.html>
 - <https://www.geeksforgeeks.org/comparator-function-of-qsort-in-c/>
- Politeness (robots.txt): denial of service attacks
- Bragging Rights
 - Our Index is bigger than yours → Race Conditions
- How do you get seeds?
 - Toolbar, Gmail, Chrome, phones, Social Media, DNS ([Domain Name System](#))
 - Security/privacy holes & DNS



Types of Answers

- Left Rail: 10 blue links
 - algo search
 - learning to rank: https://en.wikipedia.org/wiki/Learning_to_rank
- Right Rail: ads
 - auction: https://en.wikipedia.org/wiki/Sponsored_search_auction
- Instant Answers
 - Internal groups compete for types of queries

Instant Answers

The image shows a Google search interface for the query "1+1". At the top, the Google logo is on the left, and the search bar contains "1+1" with a clear button (X), a voice search icon, and a search icon. Below the search bar are navigation tabs: Images, Videos, Perspectives, Shopping, News, Maps, Books, Flights, and Finance. On the right, there are links for "All filters" and "Tools".

The main content area features an "Instant Answer" for the query "1+1". It includes a search bar with "1 + 1 =" and the result "2". Below this is a calculator interface with buttons for mathematical operations: Rad, Deg, xl, (,), %, AC, Inv, sin, ln, 7, 8, 9, +, π, cos, log, 4, 5, 6, ×, e, tan, √, 1, 2, 3, -, Ans, EXP, x^y, 0, ., =, and +. A "Feedback" link is located at the bottom right of the calculator.

Below the calculator, there are search results:

- YouTube · BeyoncéVEVO**: 8.6M+ views · 12 years ago. **Beyoncé - 1+1 (Audio) - YouTube**. Music video by Beyoncé performing 1+1. (C) 2011 Sony Music Entertainment. Includes a video player thumbnail.
- Wikipedia**: [https://en.wikipedia.org/wiki/1+1_\(song\)](https://en.wikipedia.org/wiki/1+1_(song)). **1+1 (song)**. "1+1" is a song recorded by American recording artist Beyoncé for her fourth studio album, 4 (2011). It was released by Columbia Records in the United ...
- Genius**: <https://genius.com/Beyonce-1-1-lyrics>. **Beyoncé – 1+1 Lyrics**. Aug 8, 2022 — [Post-Chorus 1] When my days look low. Pull me in close and don't let me go, make love to me. So when the world's at war, let our love heal ...

On the right side of the page, there is a vertical sidebar for the song "1+1" by Beyoncé. It features a video player with a play button and the Vevo logo. Below the video, it lists "Listen" options: Spotify, YouTube Music, iHeart, and Deezer. At the bottom, it provides album information: "Album: 4", "Artist: Beyoncé", "Released: 2011", and "Genres: R&B/Soul, Pop". A "People also search for" section shows four related artist thumbnails.

Instant Answers 10 Blue Links

The screenshot shows a Google search for "Opium War". At the top, the search bar contains "Opium War" with a microphone icon and a search button. Below the search bar are tabs for "Images", "Videos", "News", "Shopping", "Maps", "Books", "Flights", and "Finance". To the right of these tabs are "All filters" and "Tools".

The main content area features a "Opium Wars" section with three tabs: "Overview", "Outcome", and "Purpose". Below this are four small images: a steamship, a person in a room, a battle scene, and a group of people.

The "About" section on the right provides a summary: "The Opium Wars were two conflicts waged between China and Western powers during the mid-19th century. The First Opium War was fought from 1839 to 1842 between China and Britain. [Wikipedia](#)". It also lists "Location: China, Guangzhou, Guangdong Province", "Start date: 1839", and "End date: 1860". A "Feedback" link is at the bottom right.

The "Instant Answer" section contains the following text: "The first Opium War (1839–42) was fought between China and Great Britain, and the second Opium War (1856–60), also known as the Arrow War or the Anglo-French War in China, was fought by Great Britain and France against China. Qing dynasty. Read more about the Qing dynasty." Below this is the date "Nov 6, 2023".

The first blue link is from Britannica: "Opium Wars | Definition, Summary, Facts, & Causes - Britannica". It includes the Britannica logo, the URL "https://www.britannica.com > ... > International Relations", and a "Feedback" link.

The second blue link is from the U.S. Department of State (.gov): "The Opening to China Part I: the First Opium War, ...". It includes the U.S. Department of State logo, the URL "https://history.state.gov > milestones > china-1", and a snippet: "The Opium War and these treaties were emblematic of an era in which Western powers tried to gain unfettered access to Chinese products and markets for European ...".

The third blue link is also from the U.S. Department of State (.gov): "The Opening to China Part II: The Second Opium War, ...". It includes the U.S. Department of State logo, the URL "https://history.state.gov > milestones > china-2", and a snippet: "Following the First Opium War in the 1840s, the Western powers concluded a series of treaties with China in an effort to open its lucrative markets to Western ...".

The fourth blue link is from Wikipedia: "First Opium War". It includes the Wikipedia logo, the URL "https://en.wikipedia.org > wiki > First_Opium_War", and a snippet: "The First Opium War also known as the Anglo-Chinese War, was a series of military engagements fought between the British Empire and the Qing dynasty of ...". Below the snippet is the result text: "Result: British Victory: Treaty of Nanking; Esta... Location: China and South China Sea Date: 4 September 1839 – 29 August 1842; (2... Territorial changes: Hong Kong Island cede... Opium · The Opium War (film) · Treaty of Nanking · Nerbudda incident".

Snippet

Ads

Google TV

Shopping Images Videos News Perspectives Maps Books Flights Finance All filters Tools

Filter by Smart TV Under 32 inches 32 – 43 inches 43 – 55 inches 55 – 75 inches Over 75 inches On sale 4K Roku Under \$350 LED

On sale Get it fast Used Small business

Screen Size









- Under 32 inches
- 32 – 43 inches
- 43 – 55 inches
- 55 – 75 inches
- Over 75 inches

HDTV Format

- 4K
- 1080p
- 8K
- 720p

Price Brand Features Display Type Supported Streaming Services Screen Form Color Stores

Sponsored

 Pick up today LG - 48" Class A2 Series... \$599.99 4,300 Best Buy ★★★★★ (7k+) Bluetooth, Wi-Fi, USB, HDMI,...	 Samsung 55" Class The... \$1,499.99 Samsung ★★★★★ (9k+) Bluetooth, Wi-Fi, RF, USB, HDMI,...	 Pick up today Samsung 65" Class OLED... \$1,599.99 2,700 Samsung ★★★★★ (9k+) Bluetooth, Wi-Fi, USB, HDMI,...	 Pick up today Sony - 65" Class BRAVI... \$1,899.99 2,700 Best Buy ★★★★★ (2k+) Bluetooth, Wi-Fi, IrDA, RF, USB,...	 LG C3 Series 77-Inch Clas... \$2,496.99 Amazon.com Free shipping Bluetooth, Wi-Fi, RF, USB, HDMI,...	 Gold Series 32" (Vizio) / 1 Ye... \$899.00 MirageVision... Free by 1... Wi-Fi · 4K · 3840 x 2160 · 32...	 LG 50" Class 4K UHD Sm... \$299.99 \$350 Target ★★★★★ (9k+) Bluetooth, Wi-Fi, RF, USB, HDMI,...	 Samsung 55" Class The... \$1,699.99 Samsung ★★★★★ (20) Bluetooth, USB, HDMI · QLED ...
---	---	---	---	---	---	--	--

Walmart
https://www.walmart.com › tv-home-theater
TV & Video - Walmart.com
Shop TVs and Video equipment at Walmart.com and browse Smart TVs, big screen TVs and streaming devices like Roku. Save money. Live better.
4.5 ★ seller rating (846) · \$9.95 same-day delivery over \$35 · Free 90-day returns
TV Mounts & Accessories · TV Antennas, HDTV Antennas...

Best Buy
https://www.bestbuy.com › TV & Home Theater
TVs: Televisions & HDTVs
Shop Best Buy for the latest TVs, including deals on LED, 4K, OLED & curved flat screen TVs from top-rated brands.
4.7 ★ seller rating (422) · 3–5 day delivery · Free 15-day returns
TVs Under \$500 · Flat-screen TV · Smart TVs · 65-Inch TVs

Evaluation

- Extract sample of queries from query logs
- Scrape 10 blue links from release candidates
 - (and competition)
 - (over time)
- Send Queries and candidate URLs to human judges
 - score 1-5 (plus perhaps offensive)
 - perfect, excellent, good, fair, poor, offensive/illegal/irresponsible
- Score scrapes with NDCG
- Learning to Rank
- Track scores over time
 - and hopefully get better and better over time
- Issues
 - Judges prefer Google Brand
 - Even when the links are the same
 - Learning to rank → Cheating
 - Watermarking
 - Now that Google is a monopoly
 - Little incentive to improve
 - Web search may be degrading
 - (creating opportunity for chat bots)
 - SEOs (Search Engine Optimization)

Ad Auction

<https://moz.com/blog/understanding-google-ads-auction>

- Bid
- Quality
 - Estimate of CTR (clicks)
- Market maker (Google)
 - Maximizing benefit to
 - Readers
 - Writers
 - Advertisers
- Economic Optimization
 - Different from learning to rank

	Max	x	Quality	=	Ad	Ad
	CPC Bid		Score		Rank	Position
Advertiser 1	\$3.00		8		24	1
Advertiser 2	\$4.00		5		20	2
Advertiser 3	\$6.00		2		12	3
Advertiser 4	\$8.00		1		8	4



Vickrey auction

Contents [hide]

(Top)

Properties

Self-revelation and incentive compatibility

Ex-post efficiency

Weaknesses

Proof of dominance of truthful bidding

Revenue equivalence of the Vickrey auction and sealed first price auction

Use in network routing

Generalizations

See also

References

Notes

Article Talk

Read Edit View history Tools

From Wikipedia, the free encyclopedia

A **Vickrey auction** or **sealed-bid second-price auction (SBSPA)** is a type of sealed-bid **auction**. Bidders submit written bids without knowing the bid of the other people in the auction. The highest bidder wins but the price paid is the second-highest bid. This type of auction is strategically similar to an **English auction** and gives bidders an **incentive to bid their true value**. The auction was first described academically by **Columbia University professor William Vickrey** in 1961^[1] though it had been used by **stamp collectors** since 1893.^[2] In 1797 **Johann Wolfgang von Goethe** sold a manuscript using a sealed-bid, second-price auction.^[3]

Vickrey's original paper mainly considered auctions where only a single, indivisible good is being sold. The terms *Vickrey auction* and *second-price sealed-bid auction* are, in this case only, equivalent and used interchangeably. In the case of multiple identical goods, the bidders submit inverse demand curves and pay the **opportunity cost**.^[4]

Vickrey auctions are much studied in economic literature but uncommon in practice. Generalized variants of the Vickrey auction for **multiunit auctions** exist, such as the **generalized second-price auction** used in Google's and Yahoo!'s online advertisement programmes^{[5][6]} (not **incentive compatible**) and the **Vickrey–Clarke–Groves auction** (incentive compatible).

Part of a series on

Auctions

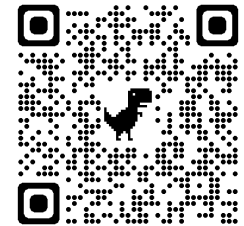


Types

- All-pay (Chinese) · Amsterdam · Anglo-Dutch · Barter double · Best/not best · Brazilian · Calcutta · Candle · Click-box bidding · Combinatorial · Common value · Deferred-acceptance · Discriminatory price · Double · Dutch · English · Forward · French · Generalized first-price · Generalized second-price · Japanese · Knapsack · Multi-attribute · Multiunit ·

Priors: Pr(URL) Query-independent

- Page Rank
 - Prior: Pr(URL)
 - Eigenvalue
 - Random Surfer Model
 - <https://en.wikipedia.org/wiki/PageRank>
- [ProNE: Fast and Scalable Network Representation Learning](#)
- [https://en.wikipedia.org/wiki/Graph neural network](https://en.wikipedia.org/wiki/Graph_neural_network)
- URL \rightarrow Noisy Channel \rightarrow Q
 - $\text{Pr}(\text{URL}) = \text{ARGMAX}_U \text{Pr}(U) \text{Pr}(Q|U)$
 - Prior: Pr(URL)
 - Channel: Pr(Q|URL)



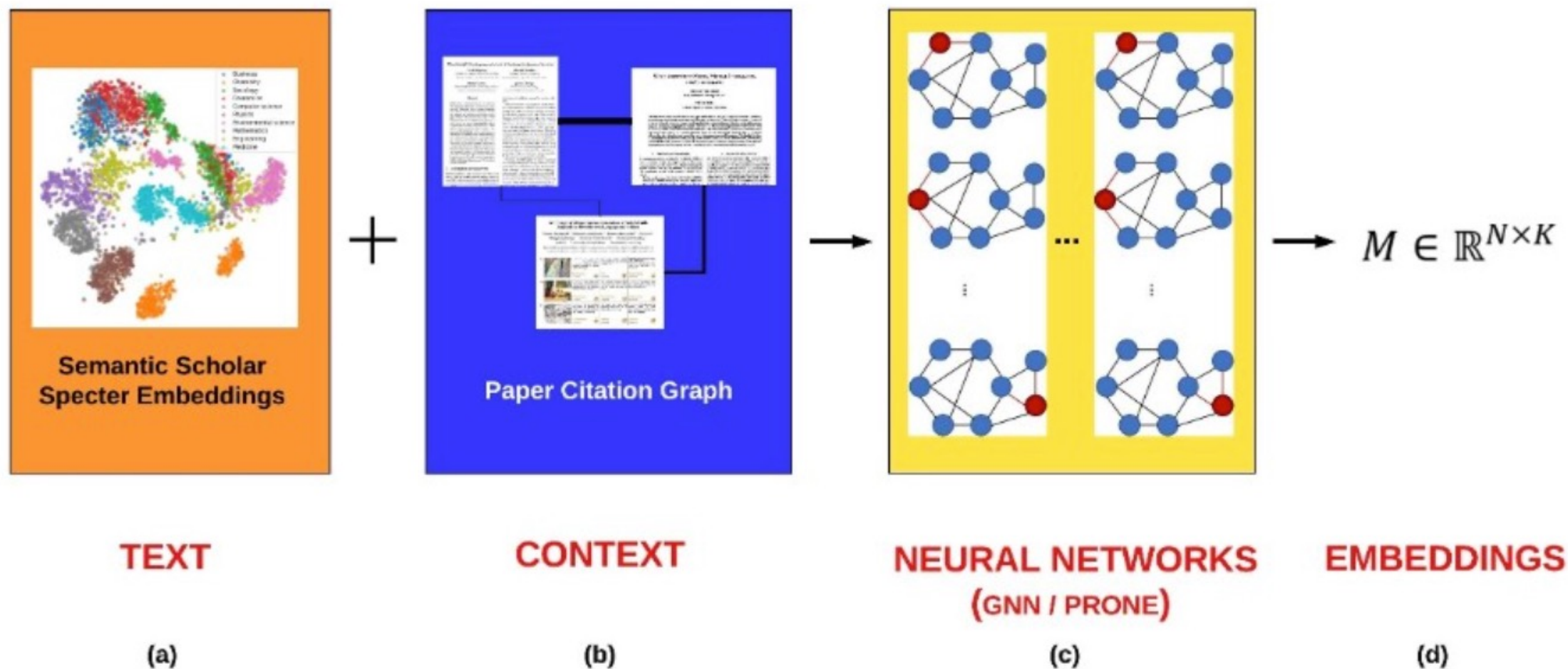
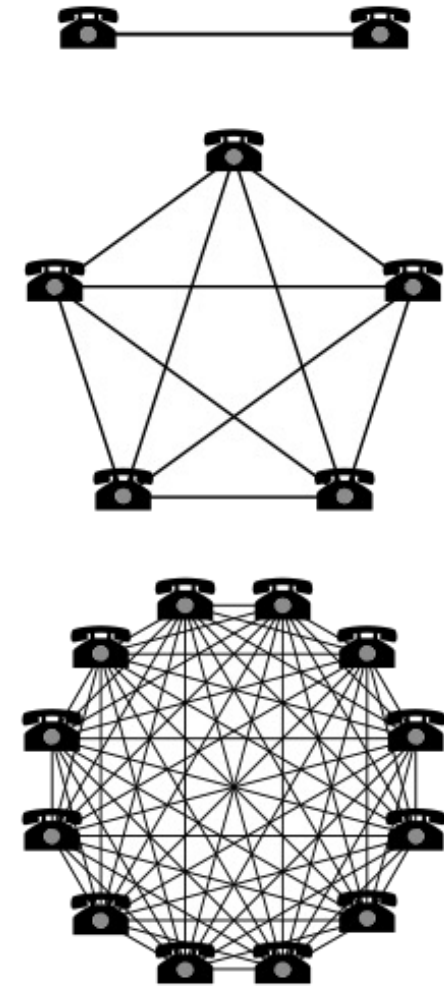


Figure 1: Embeddings are typically based on text (titles and abstracts). We will create multiple embeddings based on combinations of text and context (citations).



Metcalfe's Law (Network Effects)

- History: 3Com was selling small networks
 - 3 = 1 printer + 2 computers
 - Metcalfe argued they should sell bigger networks
 - (and more 3Com products)
 - because of economies of scale
- Economy of Scale:
 - Benefits scale faster than costs
 - Benefits: $\sim n^2$
 - Costs: $\sim n$
- Law has been good for AT&T, Google, Social Media
- Hypo: also good for Academic Search
 - Consequently, we should experiment with large graphs



Linear Algebra & Term-by-Document Matrices

- Singular Value Decomposition (SVD)
- Principal Component Analysis (PCA)
- Dimension Reduction
- Rotations
- Approximate Nearest Neighbors (ANN)

Nearly Everything To Vectors (Embeddings) (VectorDB)

- “Everything”
 - Words (Terms): word2vec
 - Documents (Text Strings):
 - doc2vec, BERT, Specter
 - Graphs (GNNs)
 - Example: citation graph
 - Semantics (“Meaning”)
 - All the world’s languages
 - Audio (Speech, Music)
 - Pictures and Videos
- Embeddings
 - Similarity \approx Cosine
 - Similar documents
 - Word Overlap
 - Nearby in citation graph
 - Similar topics, venues, authors
 - Latent (Hidden) Dimensions
- Computational Convenience
 - Dimension Reduction
 - Rotations
 - Approximate Nearest Neighbors

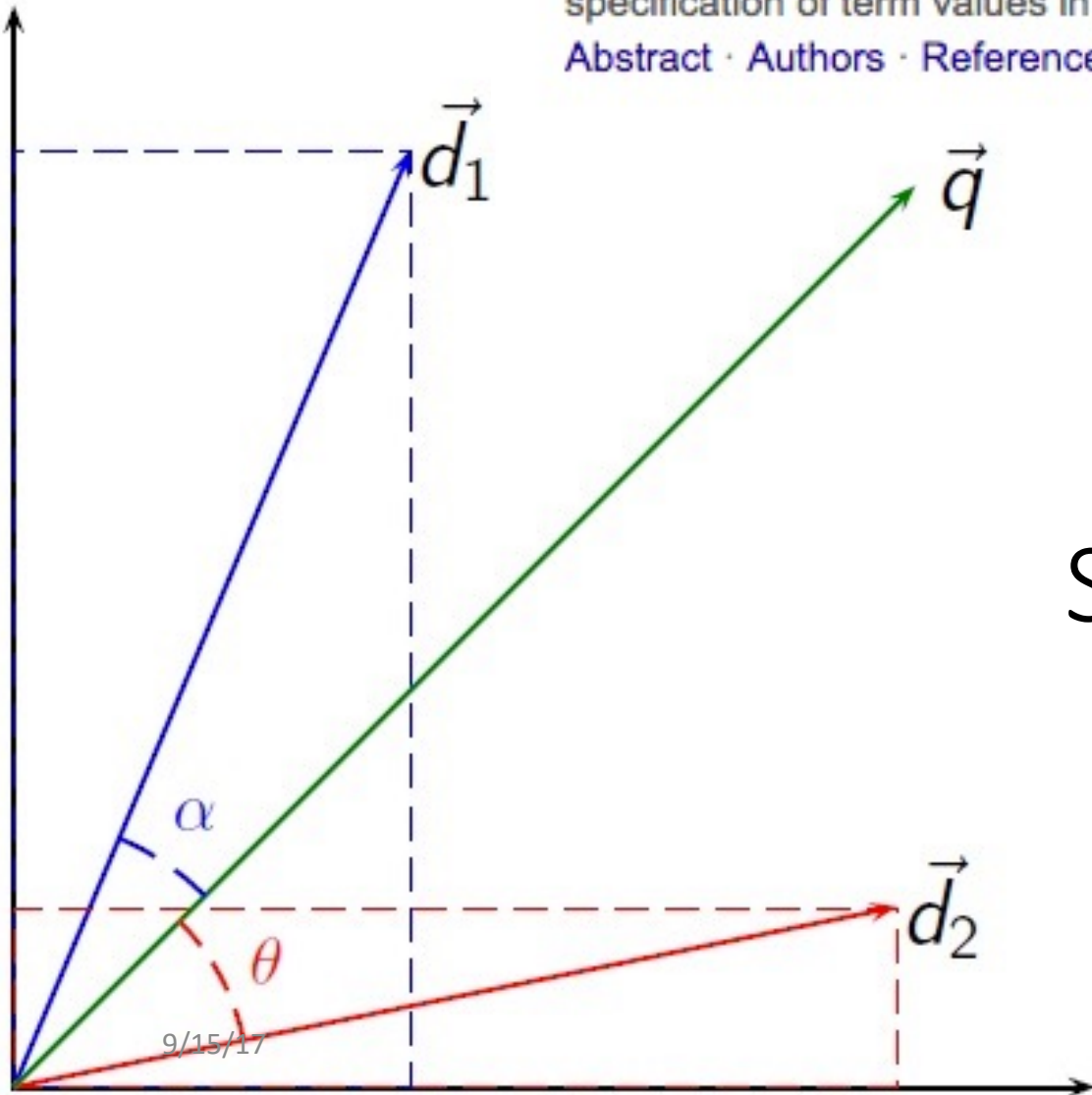
A vector space model for automatic indexing - ACM Digital Library

dl.acm.org/citation.cfm?id=361220 ▼

by G Salton - 1975 - Cited by 7464 - Related articles

A **vector space model** for automatic indexing, Published by ACM Salton, G., and Yang, C.S. On the specification of term values in automatic indexing.

[Abstract](#) · [Authors](#) · [References](#) · [Cited By](#)



Salton's Vector Space Model

Word2vec (Embeddings)

- $M \in \mathbb{R}^{V \times K}$ (tall-skinny matrix)
 - V : vocabulary size ($\approx 500k$)
 - K : hidden dimensions (≈ 300)
- $MM^T = \cos(w_i, w_j) \propto PMI(w_i, w_j)$
 - Similarity of all pairs of words in V
 - It might be infeasible to materialize MM^T
 - But there are approximations (ANNs)
 - that find many/most of the large values
- Better for capturing collocations
 - Collocations: w_i & w_j appear near one another (more than chance)
- Less appropriate for other notions of similarity
 - Both synonyms and antonyms appear near one another
 - (But they don't mean the same thing)



- For plotting purposes,
- use dimension reduction
 - to reduce K down to 2D

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

Information Retrieval (IR) notation

Term Weighting: $tf * IDF$

- t : term
- d : document
- D : # of documents in library
- Interpretation:
 - Entropy: $H = -\log(P)$
 - where $P = \Pr(t \in d)^{count(t,d)}$
- $tf(t,d)$: term frequency
 - # of times that t appears in d
- $df(t)$: document frequency
 - # of documents that contain t
 - (at least once)
- $IDF(t)$: inverse doc frequency
 - $IDF(t) = -\log_2 \frac{df(t)}{D}$
- $tf * IDF$ weighting
 - Assumes (too much) indep

Bellcore Example

- Example of term by document matrix
 - A document \approx a bag of words
 - A word \approx a bag of documents
 - *You shall know a word by the company it keeps*
- Example of SVD for dimension reduction
 - Suggestion: reducing dimensions \rightarrow better separation of classes of interest
- Motivate latent dimensions
 - as a method to embed both terms and documents
 - into a common (unified) vector space

Bellcore's Example: Bag of Words + SVD

http://wordvec.colorado.edu/papers/Deerwester_1990.pdf

- c1 Human machine *interface* for Lab ABC *computer* applications
 - c2 A *survey* of *user* opinion of computer *system response time*
 - c3 The EPS *user interface* management *system*
 - c4 *System* and *human system* engineering testing of EPS
 - c5 Relation of *user-perceived response time* to error measurement
-
- m1 The generation of random, binary, unordered *trees*
 - m2 The intersection *graph* of paths in *trees*
 - m3 *Graph minors* IV: Widths of *trees* and well-quasi-ordering
 - m4 *Graph minors: A survey*

Term by Documents Matrix

c1	Human machine interface for Lab ABC computer applications
c2	A survey of user opinion of computer system response time
c3	The EPS user interface management system
c4	System and human system engineering testing of EPS
c5	Relation of user-perceived response time to error measurement
m1	The generation of random, binary, unordered trees
m2	The intersection graph of paths in trees
m3	Graph minors IV: Widths of trees and well-quasi-ordering
m4	Graph minors: A survey

c1	c2	c3	c4	c5	m1	m2	m3	m4	
1			1						human
1		1							interface
1	1								computer
	1	1		1					user
	1	1	2						system
	1			1					response
	1			1					time
			1	1					EPS
		1						1	survey
					1	1	1		trees
						1	1	1	graph
							1	1	minors

Term by Document Matrix

c1	c2	c3	c4	c5	m1	m2	m3	m4	
1			1						human
1		1							interface
1	1								computer
	1	1		1					user
	1	1	2						system
	1			1					response
	1			1					time
		1	1						EPS
	1							1	survey
					1	1	1		trees
						1	1	1	graph
							1	1	minors

Singular Value Decomposition (SVD)

- $M \approx U D V^T$
- D is diagonal
 - Eigenvalues
 - Sorted from largest to smallest
- U and V are Eigenvectors
 - Orthogonal and unit length
 - $U^T U = I$
 - $V^T V = I$
- $\cos(M, M) = M M^T$
 - $U D V^T (U D V^T)^T$
 - $U D V^T (V D U^T)$
 - $U D^2 U^T$
- $M \rightarrow U D$
 - Plus dimension reduction
 - Replace smaller Eigenvalues with 0

Dimension Reduction

- Standard Recipe
 - Set smaller Eigenvalues to 0
 - Interpretation
 - L2 optimality (least squares)
 - Recall that Eigenvalues are sorted from largest to smallest
- Motivation for dimension reduction
 - Computational resources:
 - Space
 - Specter: $M \in \mathbb{R}^{N \times K}$
 - N is 200M documents
 - K is 768 (BERT hidden layer)
 - $MM^T \in \mathbb{R}^{N \times N}$ (**very** large)
 - Time
 - Statistical convenience:
 - Smoothing (soft thesaurus)
 - Replace zeros with small values
 - Computational convenience:
 - Approximate nearest neighbors
 - <https://pypi.org/project/annoy/>

SVD and PCA

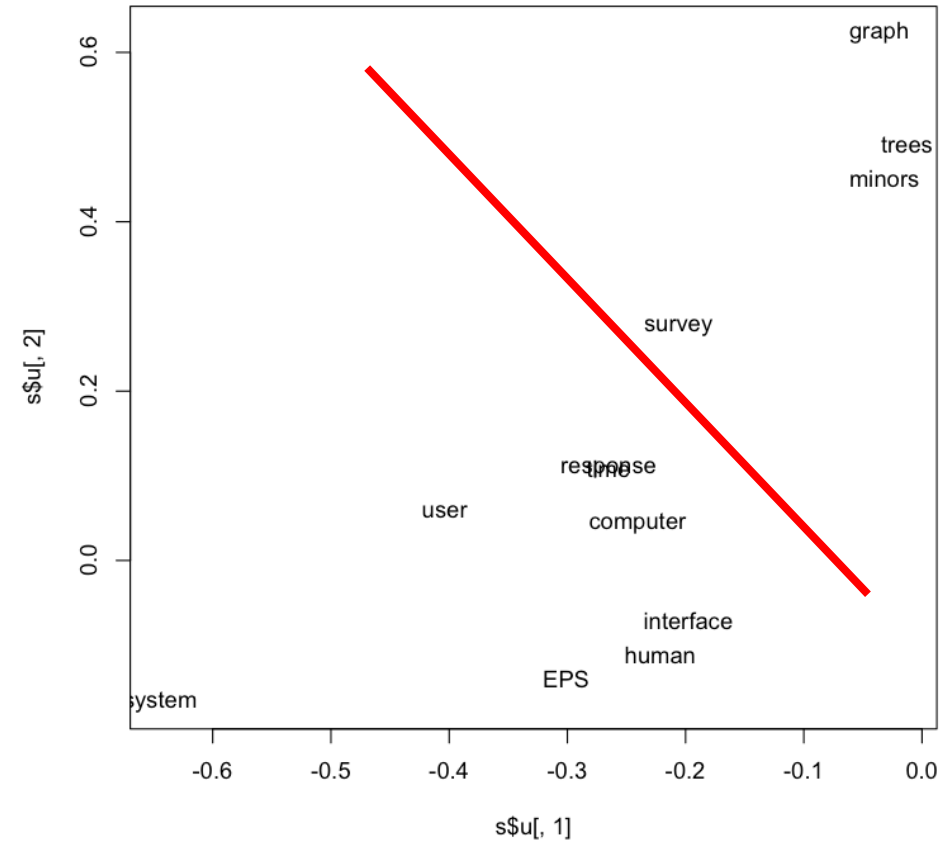
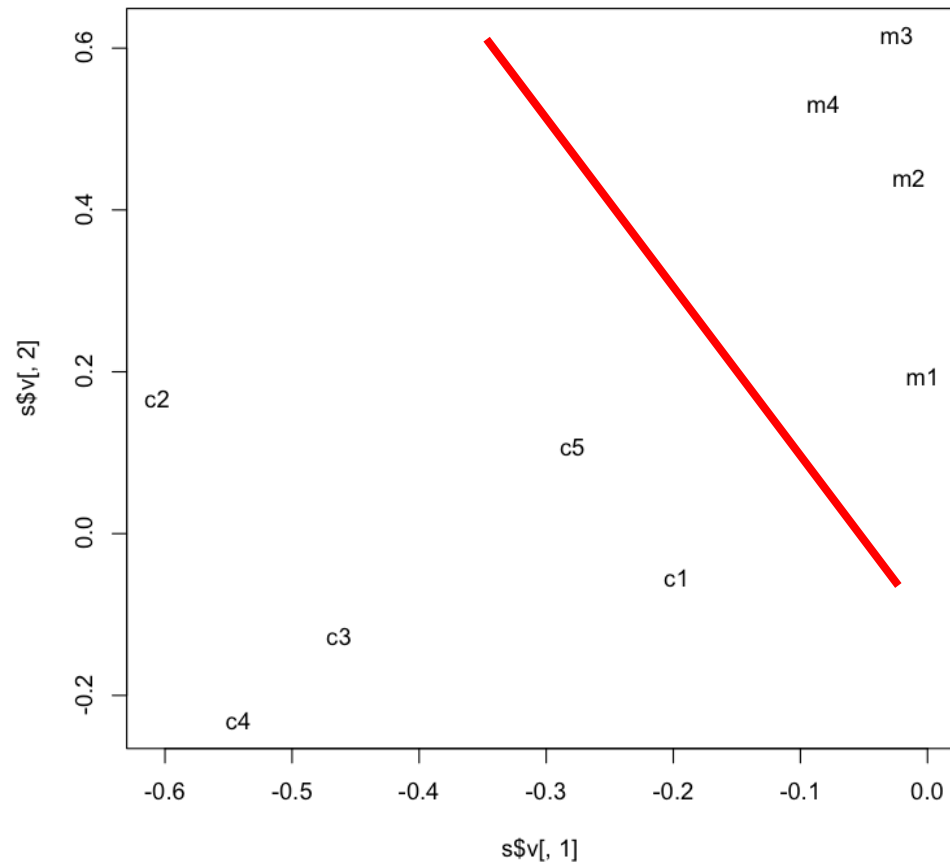
SVD (Singular Value Decomposition)

- $M \approx U D V^T$
- D : Eigenvalues
- U : Eigenvectors
- M need not be square
 - (just non-singular)

PCA (Principal Component Analysis)

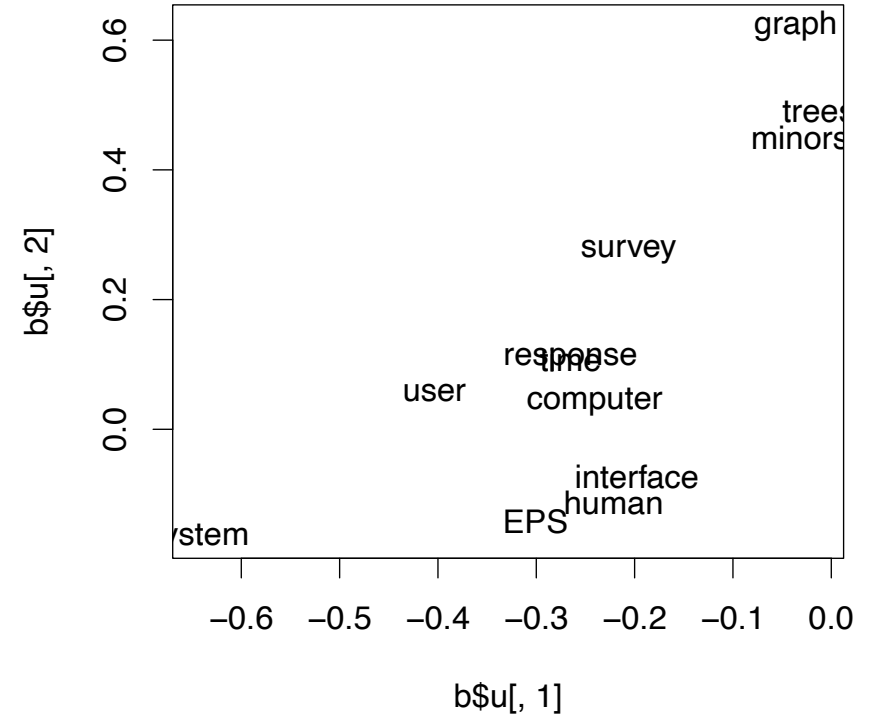
- $Q \propto X^T X = W \Lambda W^T$
- Q is square by construction
 - Λ : Eigenvalues
 - W : Covariances
 - Diagonal of W are variances

SVD maps terms & docs into internal dimensions



$$bellcore \approx U D V^T$$

c1	c2	c3	c4	c5	m1	m2	m3	m4	
1			1						human
1		1							interface
1	1								computer
	1	1		1					user
	1	1	2						system
	1			1					response
	1			1					time
		1	1						EPS
	1							1	survey
					1	1	1		trees
						1	1	1	graph
							1	1	minors



```
b = svd(bellcore)
```

```
b2 = b$u[,1:2] %*% diag(b$d[1:2]) %*% t(b$v[,1:2])
```

```
dimnames(b2) = dimnames(bellcore)
```

```
par(mfrow=c(2,2))
```

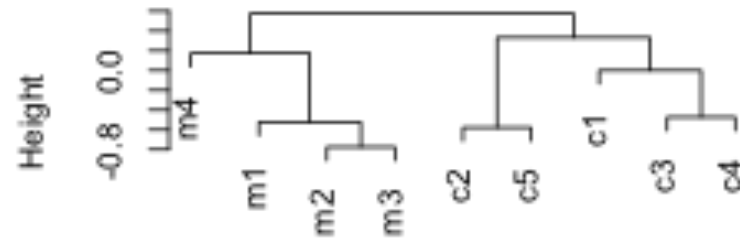
```
plot(hclust(as.dist(-cor(bellcore))))
```

```
plot(hclust(as.dist(-cor(t(bellcore)))))
```

```
plot(hclust(as.dist(-cor(b2))))
```

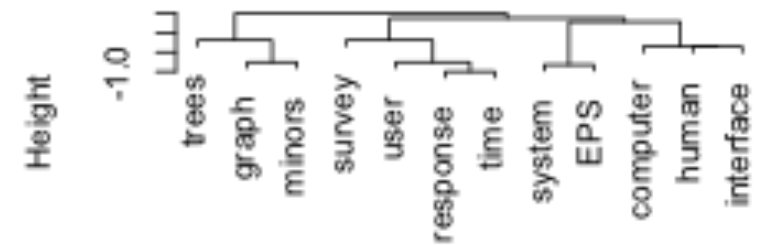
```
plot(hclust(as.dist(-cor(t(b2)))))
```

Cluster Dendrogram



as.dist(-cor(bellcore))
hclust (*, "complete")

Cluster Dendrogram

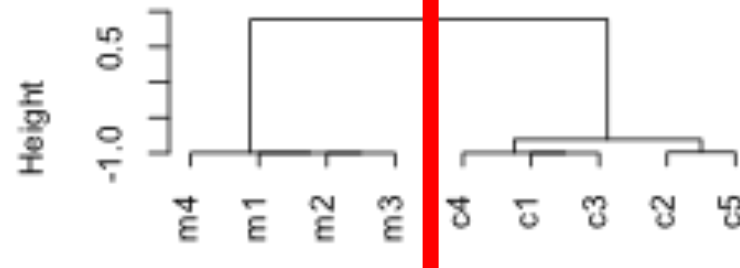


as.dist(-cor(t(bellcore)))
hclust (*, "complete")

Term by Document Matrix

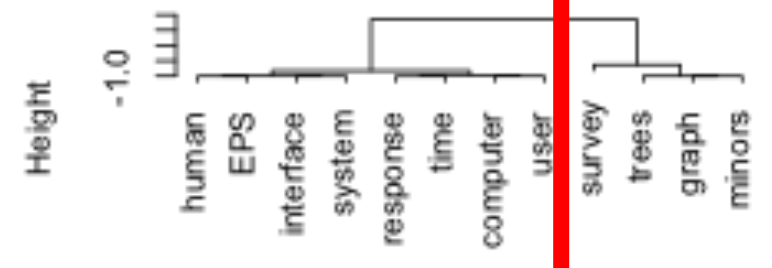
	c1	c2	c3	c4	c5	m1	m2	m3	m4	
1				1						human
1			1							interface
1	1									computer
		1	1		1					user
			1	1	2					system
						1				response
							1			time
								1		EPS
		1							1	survey
						1	1	1		trees
							1	1	1	graph
								1	1	minors

Cluster Dendrogram



as.dist(-cor(b2))
hclust (*, "complete")

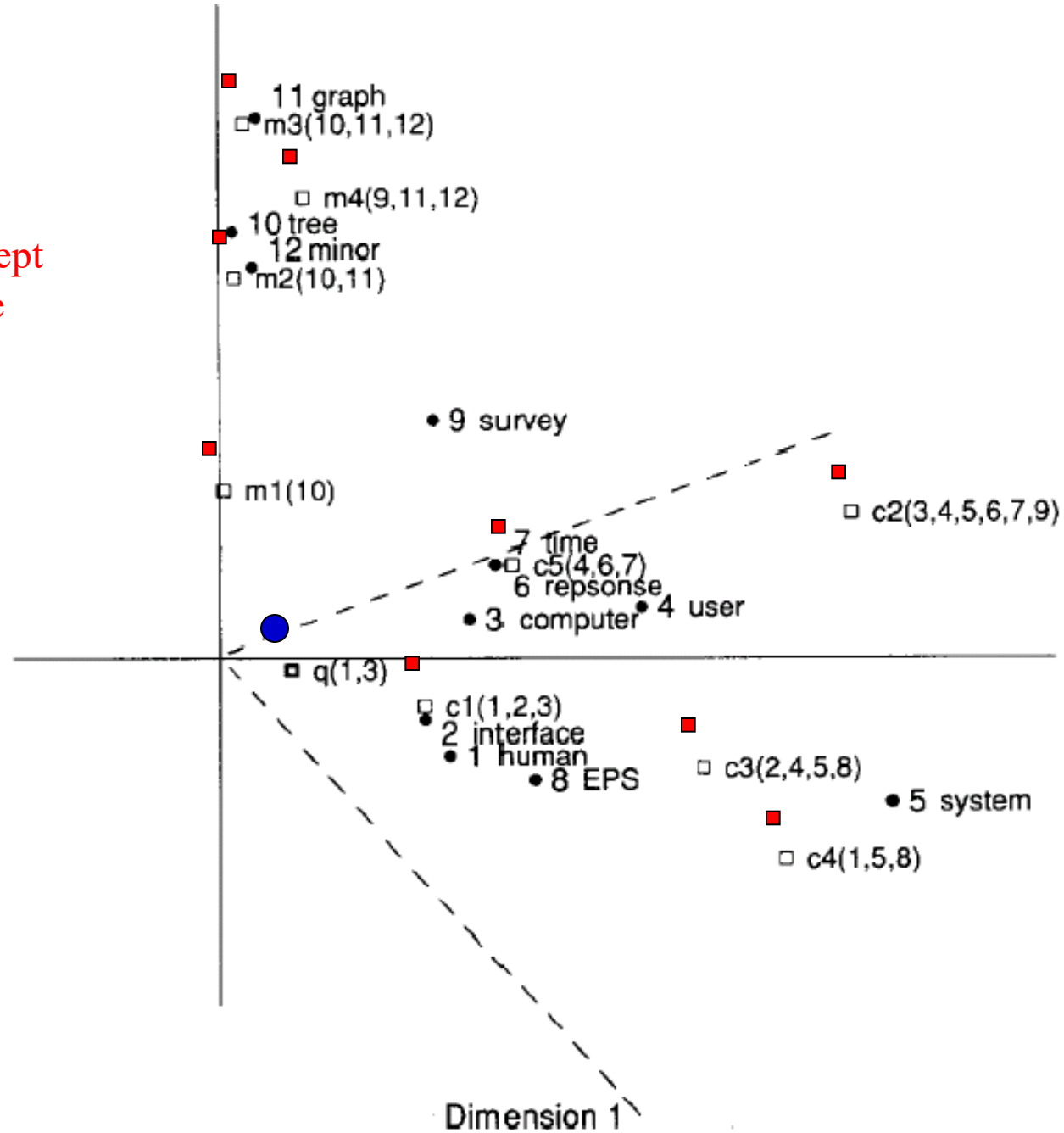
Cluster Dendrogram



as.dist(-cor(t(b2)))
hclust (*, "complete")

Latent concept
vector space

- term
- document
- query
- cosine > 0.9



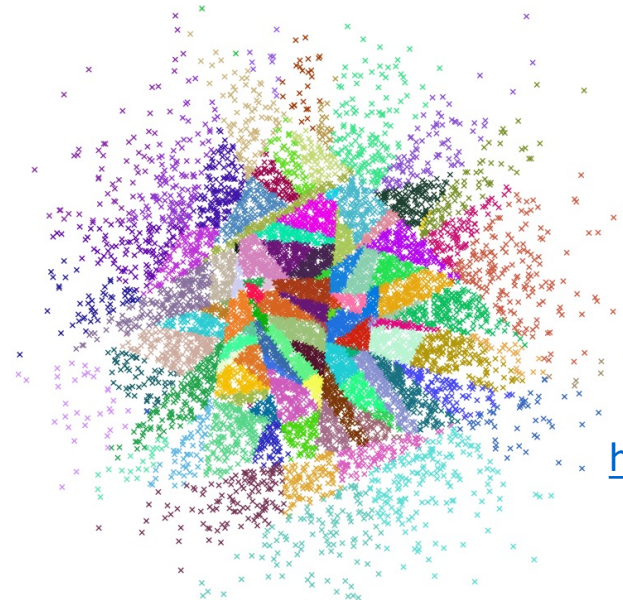
Approximate Nearest Neighbors (ANN)

- Indexing time:
 - Input: Embedding $M \in \mathbb{R}^{N \times K}$
 - Output: Indexes
- Query time:
 - Input:
 - Embedding, Indexes, query
 - Query: $q \in \mathbb{R}^K$
 - Output: candidates, $c \in \mathbb{R}^K$
 - where c is near q
 - sorted by $sim(q, c)$

```
from gensim.similarities.annoy import AnnoyIndexer

# 100 trees are being used in this example
annoy_index = AnnoyIndexer(model, 100)
# Derive the vector for the word "science" in our model
vector = wv["science"]
# The instance of AnnoyIndexer we just created is passed
approximate_neighbors = wv.most_similar([vector], topn=11, indexer=annoy_index)
# Neatly print the approximate_neighbors and their corresponding cosine similarity values
print("Approximate Neighbors")
for neighbor in approximate_neighbors:
    print(neighbor)
```

https://radimrehurek.com/gensim/auto_examples/tutorials/run_annoy.html



<https://pypi.org/project/annoy/>

Machine Translation

- <https://web.stanford.edu/~jurafsky/slp3/13.pdf>

