Vector Space Model

Jaime Arguello
INLS 509: Information Retrieval
jarguell@email.unc.edu

February 13, 2013

The Search Task

- Given a query and a corpus, find relevant items

  query: a textual description of the user's information need

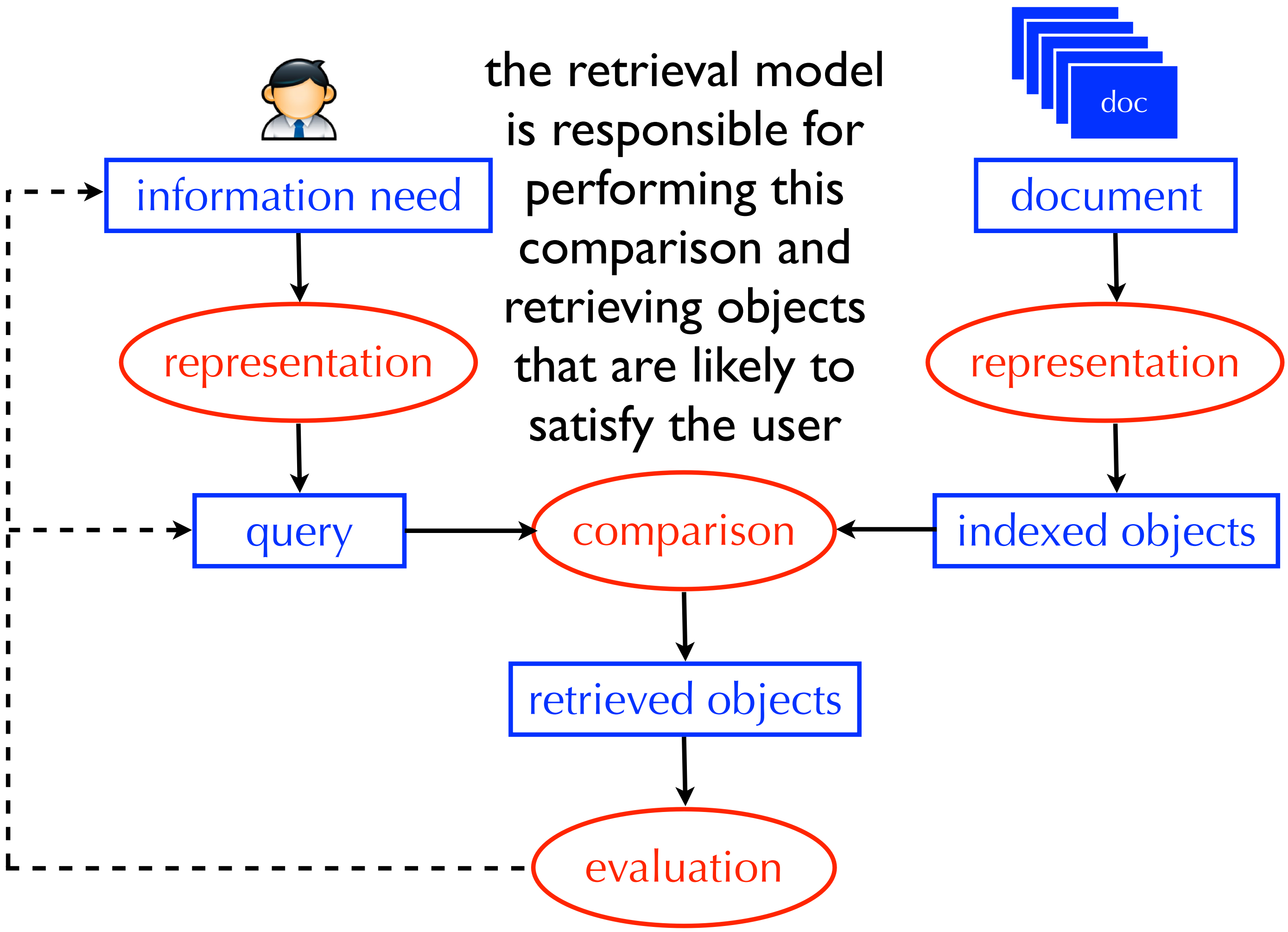
  corpus: a repository of textual documents

  relevance: satisfaction of the user's information need

What is a Retrieval Model?

- A formal method that predicts the degree of relevance of a document to a query

Basic Information Retrieval Process



Boolean Retrieval Models

- The user describes their information need using boolean constraints (e.g., AND, OR, and AND NOT)
- Unranked Boolean Retrieval Model: retrieves documents that satisfy the constraints in no particular order
- Ranked Boolean Retrieval Model: retrieves documents that satisfy the constraints and ranks them based on the number of ways they satisfy the constraints
- Also known as 'exact-match' retrieval models
- Advantages and disadvantages?

Boolean Retrieval Models

- Advantages:
  - Easy for the system
  - Users get transparency: it is easy to understand why a document was or was not retrieved
  - Users get control: it easy to determine whether the query is too specific (few results) or too broad (many results)
- Disadvantages:
  - The burden is on the user to formulate a good boolean query

Relevance

- Many factors affect whether a document satisfies a particular user's information need
- Topicality, novelty, freshness, authority, formatting, reading level, assumed level of prior knowledge/expertise
- Topical relevance: the document is on the same topic as the query
- User relevance: everything else!
- For now, we will only try to predict topical relevance

Relevance

- Focusing on topical relevance does not mean we're ignoring everything else!
- It only means we're focusing on one (of many) criteria by which users judge relevance
- And, it's an important criterion

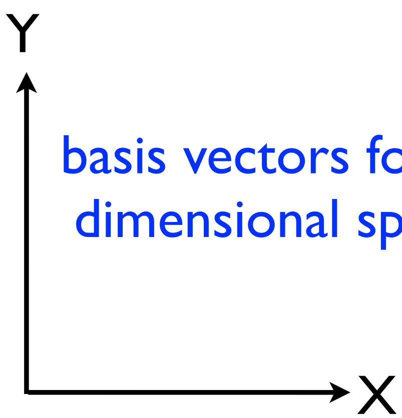
Introduction to Best-Match Retrieval Models

- So far, we've discussed 'exact-match' models
- Today, we start discussing 'best-match' models
- Best-match models predict the degree to which a document is relevant to a query
- Ideally, this would be expressed as RELEVANT(q,d)
- In practice, it is expressed as SIMILAR(q,d)
- How might you compute the similarity between q and d?

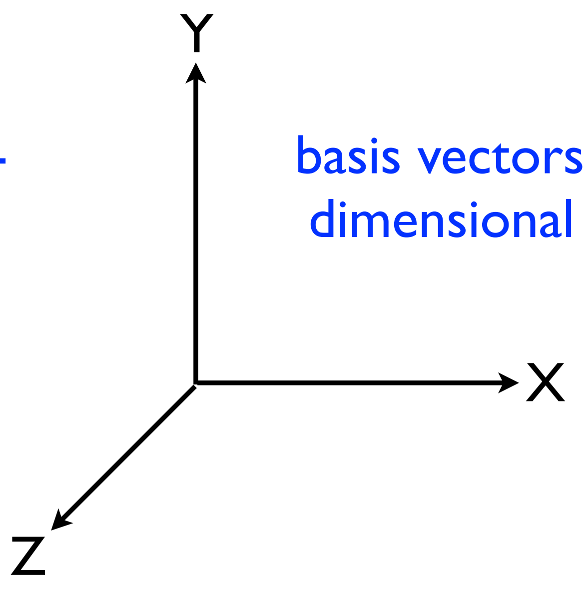
Vector Space Model

What is a Vector Space?

- Formally, a vector space is defined by a set of <u>linearly independent</u> basis vectors
- The basis vectors correspond to the dimensions or directions of the vector space



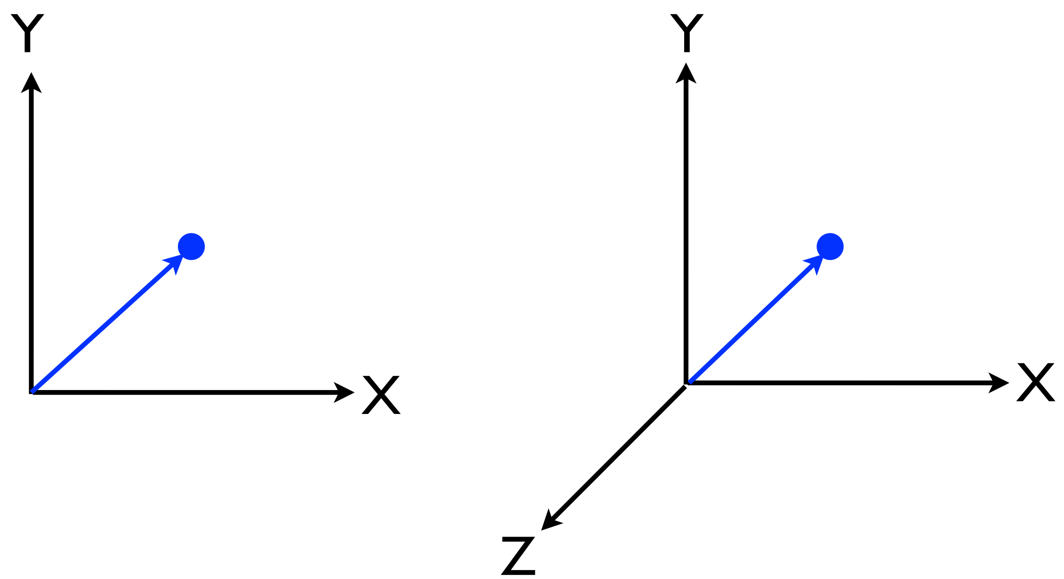
basis vectors for 2-dimensional space



basis vectors for 3-dimensional space

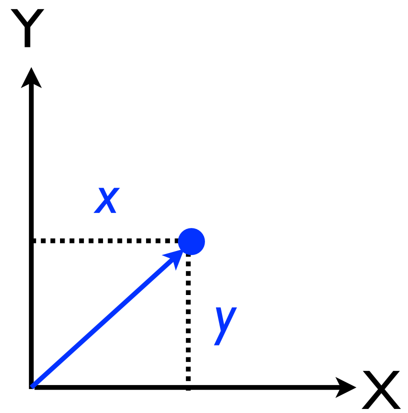
What is a Vector?

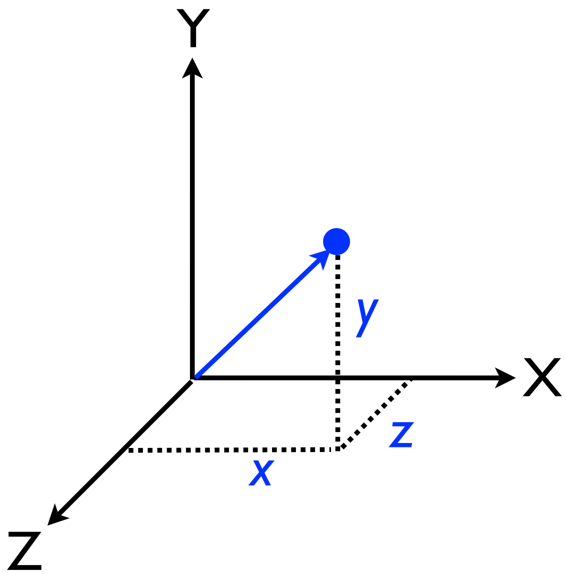
- A vector is a point in a vector space and has length (from the origin to the point) and direction



What is a Vector?

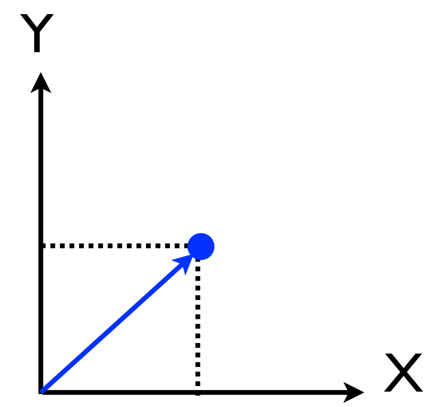
- A 2-dimensional vector can be written as **[*x*,*y*]**
- A 3-dimensional vector can be written as **[*x*,*y*,*z*]**



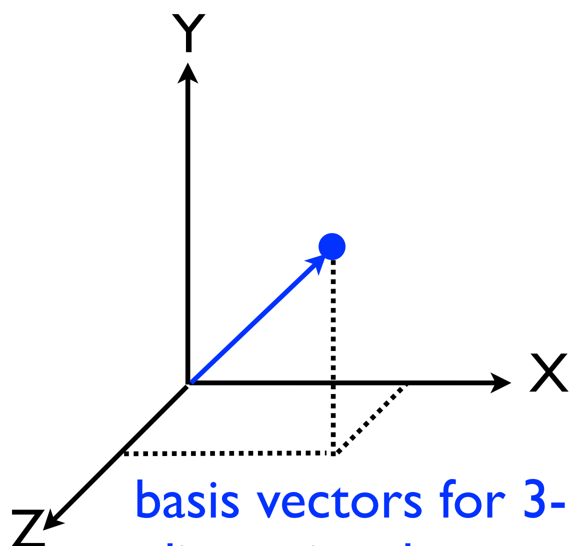


What is a Vector Space?

- The basis vectors are <u>linearly independent</u> because knowing a vector's value on one dimension doesn't say anything about its value along another dimension



basis vectors for 2-dimensional space



basis vectors for 3-dimensional space

Binary Text Representation

	a	*aardvark*	*abacus*	*abba*	*able*	...	*zoom*
doc_1	1	0	0	0	0	...	1
doc_2	0	0	0	0	1	...	1
::	::	::	::	::	::	...	0
doc_m	0	0	1	1	0	...	0

- 1 = the word appears in the document
- 0 = the word does <u>not</u> appear in the document
- Does not represent word frequency, word location, or word order information

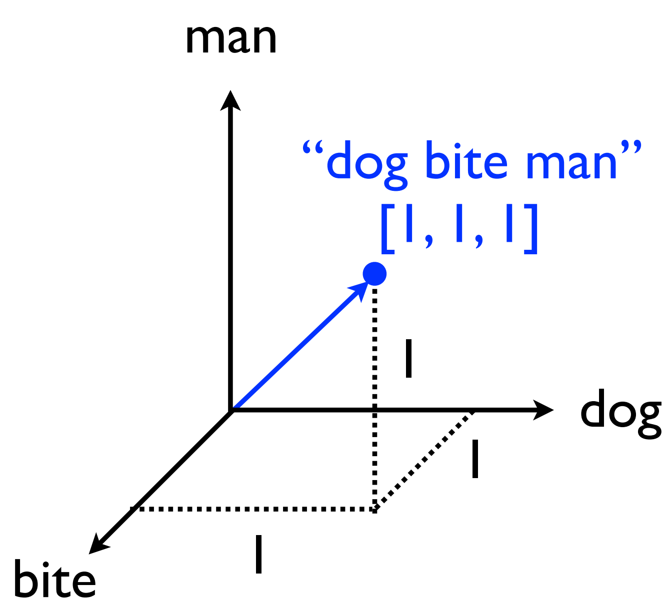
Vector Space Representation

- Let V denote the size of the indexed vocabulary
  - V = the number of unique terms,
  - V = the number of unique terms excluding stopwords,
  - V = the number of unique stems, etc...
- Any arbitrary span of text (i.e., a document, or a query) can be represented as a vector in V-dimensional space
- For simplicity, let's assume three index terms: dog, bite, man (i.e., **V=3)**
- Why? Because it's easy to visualize 3-D space

Vector Space Representation
with binary weights

- 1 = the term appears at least once
- 0 = the term does <u>not</u> appear

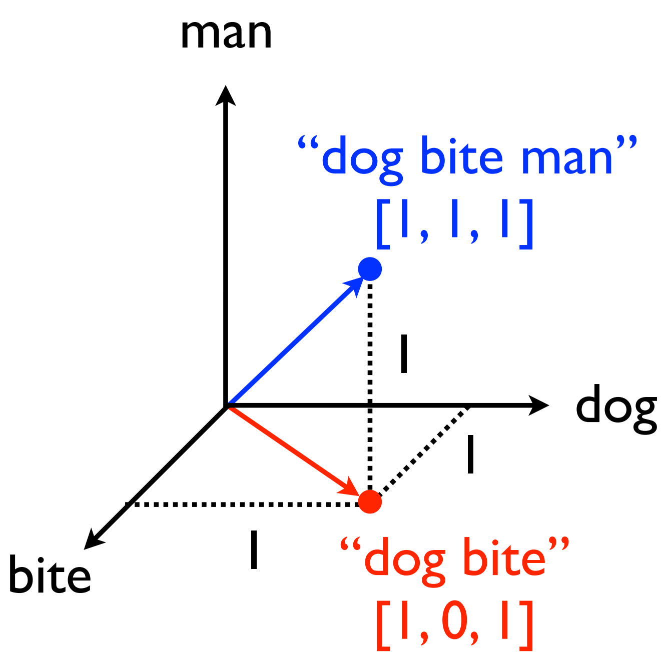
	dog	*man*	*bite*
doc_1	1	1	1



Vector Space Representation
with binary weights

- 1 = the term appears at least once
- 0 = the term does not appear

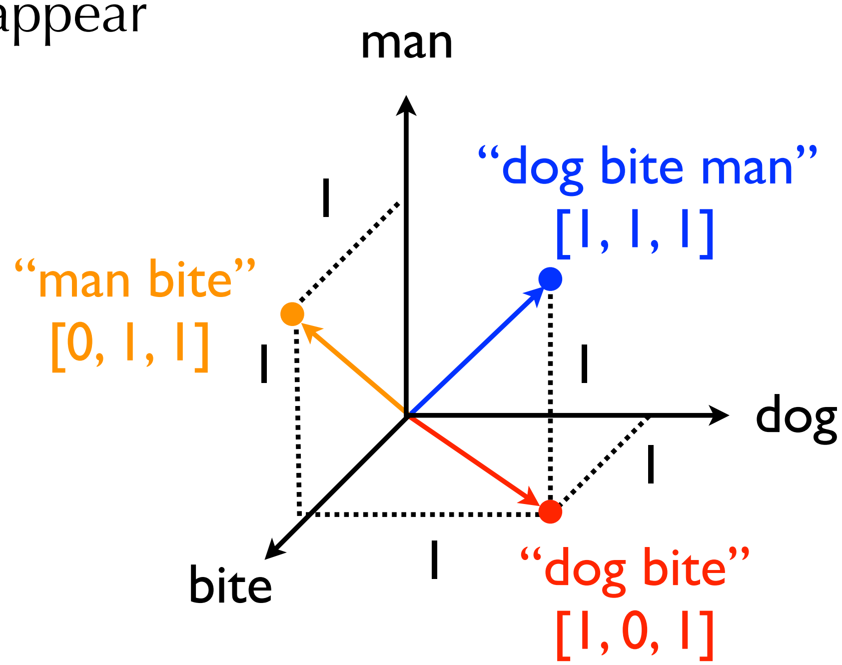
	dog	*man*	*bite*
doc_1	1	1	1
doc_2	1	0	1



Vector Space Representation
with binary weights

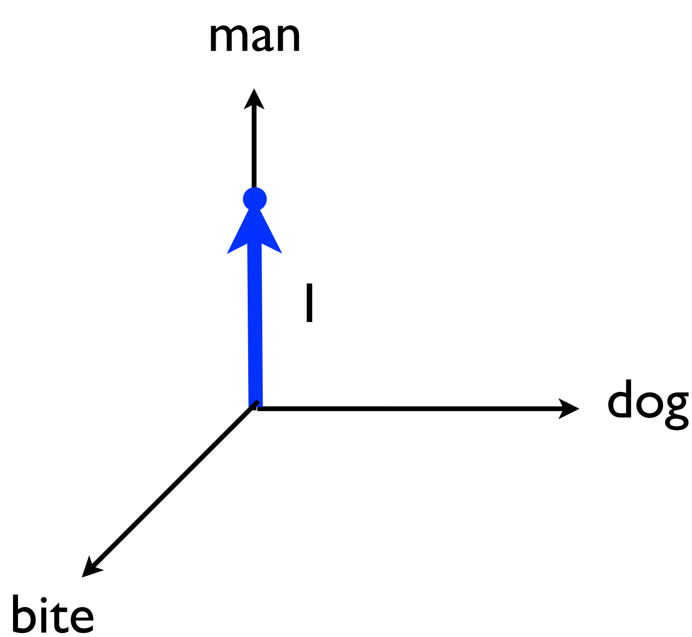
- 1 = the term appears at least once
- 0 = the term does not appear

	dog	*man*	*bite*
doc_1	1	1	1
doc_2	1	0	1
doc_3	0	1	1



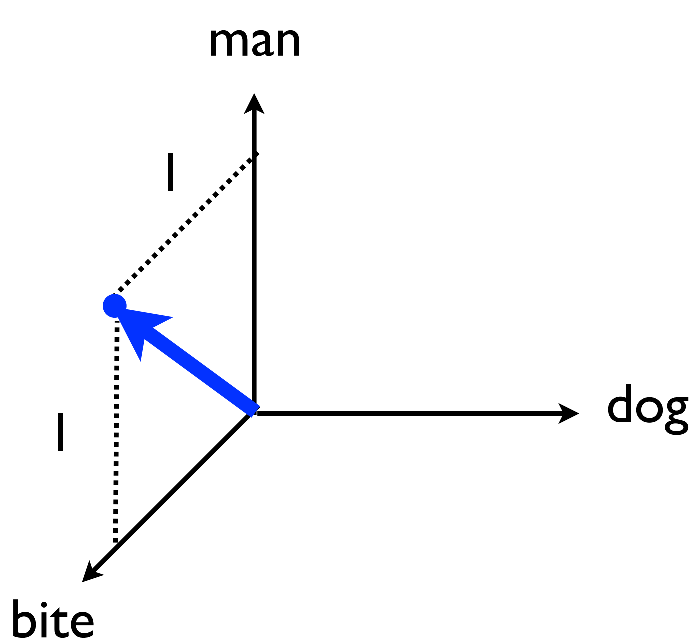
Vector Space Representation
with binary weights

- What span(s) of text does this vector represent?



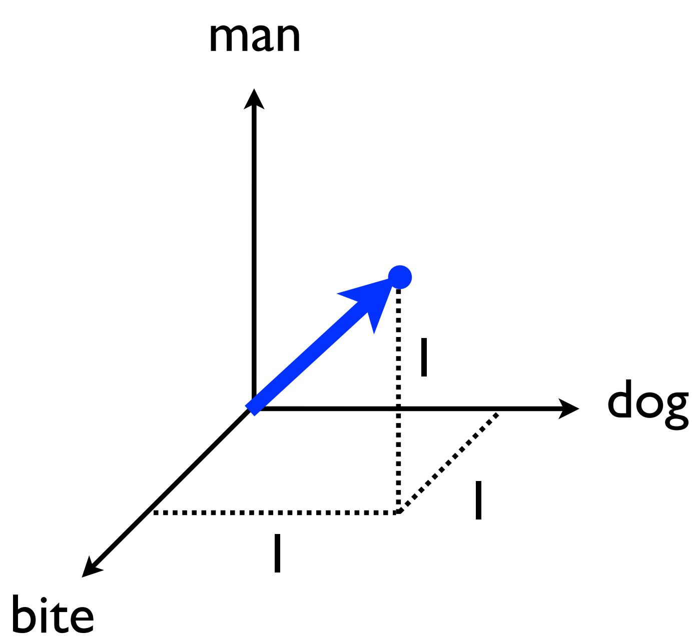
Vector Space Representation
with binary weights

- What span(s) of text does this vector represent?



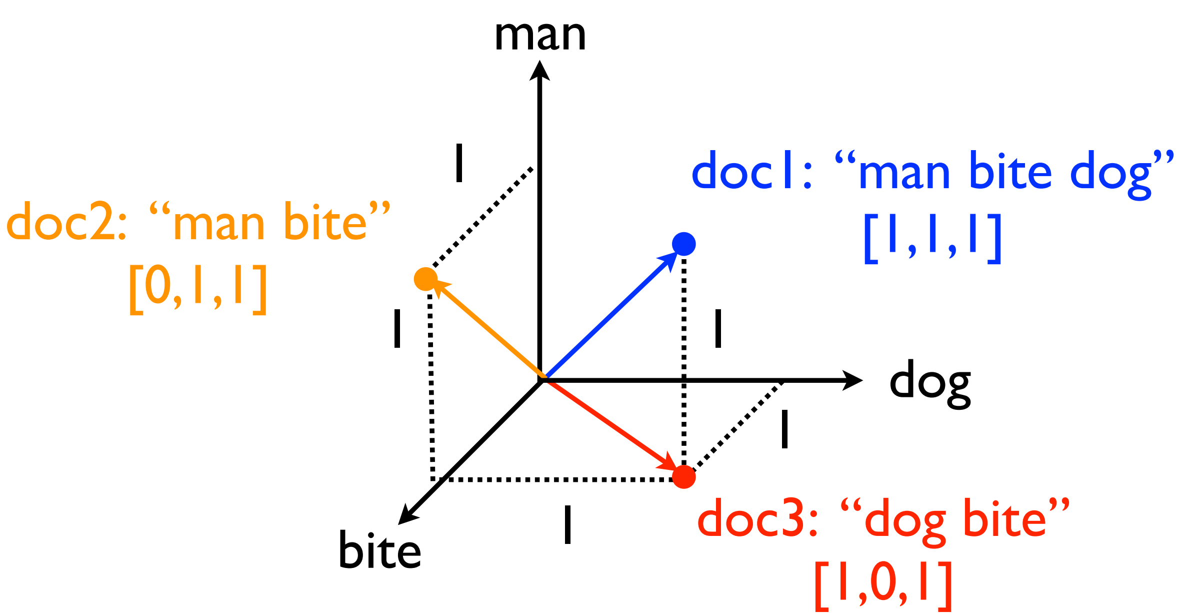
Vector Space Representation
with binary weights

- What span(s) of text does this vector represent?



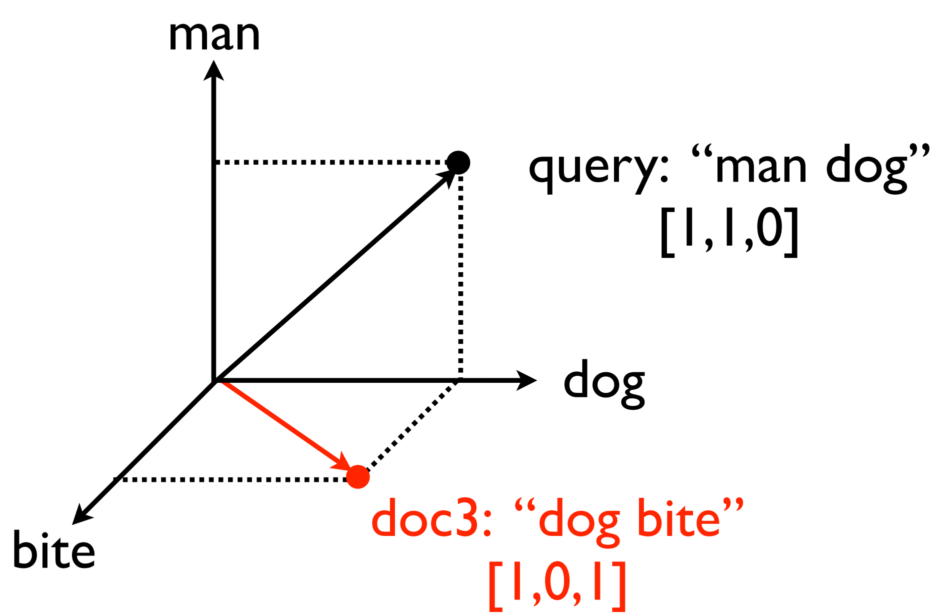
Vector Space Representation

- Any span of text is a vector in V-dimensional space, where V is the size of the vocabulary



Vector Space Representation

- A query is a vector in V-dimensional space, where V is the number of terms in the vocabulary



Vector Space Similarity

- The vector space model ranks documents based on the vector-space similarity between the query vector and the document vector
- There are many ways to compute the similarity between two vectors
- One way is to compute the inner product

$$\sum_{i=1}^{V} x_i \times y_i$$

The Inner Product

- Multiply corresponding components and then sum of those products

$$\sum_{i=1}^{V} x_i \times y_i$$

	x_i	y_i	$x_i \times y_i$
a	1	1	1
aardvark	0	1	0
abacus	1	1	1
abba	1	0	0
able	0	1	0
::	::	::	::
zoom	0	0	0
		inner product =>	2

The Inner Product

- When using 0's and 1's, this is just the number of terms in common between the query and the document

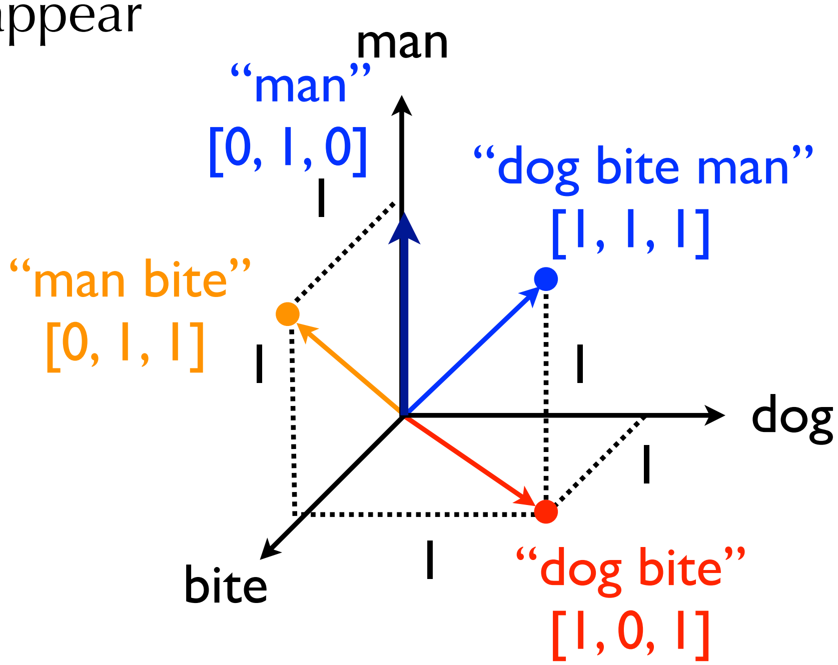
$$\sum_{i=1}^{V} x_i \times y_i$$

	x_i	y_i	$x_i \times y_i$
a	1	1	1
aardvark	0	1	0
abacus	1	1	1
abba	1	0	0
able	0	1	0
::	::	::	::
zoom	0	0	0
		inner product =>	2

The Inner Product

- 1 = the term appears at least once
- 0 = the term does not appear

	dog	*man*	*bite*
doc_1	1	1	1
doc_2	1	0	1
doc_3	0	1	1
doc_4	0	1	0



The Inner Product

- Multiply corresponding components and then sum those products
- Using a binary representation, the inner product corresponds to the number of terms appearing (at least once) in both spans of text
- Scoring documents based on their inner-product with the query has one major issue. Any ideas?

The Inner Product

- What is more relevant to a query?
  - A 50-word document which contains 3 of the query-terms?
  - A 100-word document which contains 3 of the query-terms?
- The inner-product doesn't account for the fact that documents have widely varying lengths
- All things being equal, longer documents are more likely to have the query-terms
- So, the inner-product favors long documents

The Cosine Similarity

- The numerator is the inner product
- The denominator is the product of the two vector-lengths
- Ranges from 0 to 1 (equals 1 if the vectors are identical)

$$\frac{\sum_{i=1}^{V} x_i \times y_i}{\underbrace{\sqrt{\sum_{i=1}^{V} x_i^2}}_{\text{length of vector x}} \times \underbrace{\sqrt{\sum_{i=1}^{V} y_i^2}}_{\text{length of vector y}}}$$

$$\frac{\sum_{i=1}^{V} x_i \times y_i}{\sqrt{\sum_{i=1}^{V} x_i^2} \times \sqrt{\sum_{i=1}^{V} y_i^2}}$$

In Class Exercise

- For each document, compute the inner-product and cosine similarity score for the query: Jill

doc_1	Jack and Jill went up the hill
doc_2	To fetch a pail of water.
doc_3	Jack fell down and broke his crown,
doc_4	And Jill came tumbling after.
doc_5	Up Jack got, and home did trot,
doc_6	As fast as he could caper,
doc_7	To old Dame Dob, who patched his nob
doc_8	With vinegar and brown paper.

$$\frac{\sum_{i=1}^{V} x_i \times y_i}{\sqrt{\sum_{i=1}^{V} x_i^2} \times \sqrt{\sum_{i=1}^{V} y_i^2}}$$

In Class Exercise

- For each document, compute the inner-product and cosine similarity score for the query: Jack

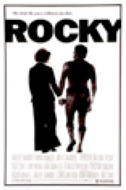
doc_1	Jack and Jill went up the hill
doc_2	To fetch a pail of water.
doc_3	Jack fell down and broke his crown,
doc_4	And Jill came tumbling after.
doc_5	Up Jack got, and home did trot,
doc_6	As fast as he could caper,
doc_7	To old Dame Dob, who patched his nob
doc_8	With vinegar and brown paper.

Vector Space Representation

	a	*aardvark*	*abacus*	*abba*	*able*	...	*zoom*
doc_1	1	0	0	0	0	...	1
doc_2	0	0	0	0	1	...	1
::	::	::	::	::	::	...	0
doc_m	0	0	1	1	0	...	0

	a	*aardvark*	*abacus*	*abba*	*able*	...	*zoom*
query	0	1	0	0	1	...	1

- So far, we've assumed binary vectors
- 0's and 1's indicate whether the term occurs (at least once) in the document/query
- Let's explore a more sophisticated representation

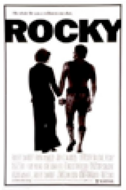


Term-Weighting

what are the most important terms?

- Movie: Rocky (1976)
- Plot:

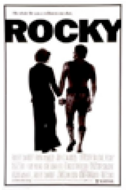
Rocky Balboa is a struggling boxer trying to make the big time. Working in a meat factory in Philadelphia for a pittance, he also earns extra cash as a debt collector. When heavyweight champion Apollo Creed visits Philadelphia, his managers want to set up an exhibition match between Creed and a struggling boxer, touting the fight as a chance for a "nobody" to become a "somebody". The match is supposed to be easily won by Creed, but someone forgot to tell Rocky, who sees this as his only shot at the big time. Rocky Balboa is a small-time boxer who lives in an apartment in Philadelphia, Pennsylvania, and his career has so far not gotten off the canvas. Rocky earns a living by collecting debts for a loan shark named Gazzo, but Gazzo doesn't think Rocky has the viciousness it takes to beat up deadbeats. Rocky still boxes every once in a while to keep his boxing skills sharp, and his ex-trainer, Mickey, believes he could've made it to the top if he was willing to work for it. Rocky, goes to a pet store that sells pet supplies, and this is where he meets a young woman named Adrian, who is extremely shy, with no ability to talk to men. Rocky befriends her. Adrain later surprised Rocky with a dog from the pet shop that Rocky had befriended. Adrian's brother Paulie, who works for a meat packing company, is thrilled that someone has become interested in Adrian, and Adrian spends Thanksgiving with Rocky. Later, they go to Rocky's apartment, where Adrian explains that she has never been in a man's apartment before. Rocky sets her mind at ease, and they become lovers. Current world heavyweight boxing champion Apollo Creed comes up with the idea of giving an unknown a shot at the title. Apollo checks out the Philadelphia boxing scene, and chooses Rocky. Fight promoter Jergens gets things in gear, and Rocky starts training with Mickey. After a lot of training, Rocky is ready for the match, and he wants to prove that he can go the distance with Apollo. The 'Italian Stallion', Rocky Balboa, is an aspiring boxer in downtown Philadelphia. His one chance to make a better life for himself is through his boxing and Adrian, a girl who works in the local pet store. Through a publicity stunt, Rocky is set up to fight Apollo Creed, the current heavyweight champion who is already set to win. But Rocky really needs to triumph, against all the odds...



Term-Frequency

how important is a term?

rank	term	freq.	rank	term	freq.
1	a	22	16	creed	5
2	rocky	19	17	philadelphia	5
3	to	18	18	has	4
4	the	17	19	pet	4
5	is	11	20	boxing	4
6	and	10	21	up	4
7	in	10	22	an	4
8	for	7	23	boxer	4
9	his	7	24	s	3
10	he	6	25	balboa	3
11	adrian	6	26	it	3
12	with	6	27	heavyweigh	3
13	who	6	28	champion	3
14	that	5	29	fight	3
15	apollo	5	30	become	3



Term-Frequency

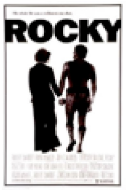
how important is a term?

rank	term	freq.	rank	term	freq.
1	a	22	16	creed	5
2	rocky	19	17	philadelphia	5
3	to	18	18	has	4
4	the	17	19	pet	4
5	is	11	20	boxing	4
6	and	10	21	up	4
7	in	10	22	an	4
8	for	7	23	boxer	4
9	his	7	24	s	3
10	he	6	25	balboa	3
11	adrian	6	26	it	3
12	with	6	27	heavyweigh	3
13	who	6	28	champion	3
14	that	5	29	fight	3
15	apollo	5	30	become	3

Inverse Document Frequency (IDF)
how important is a term?

$$idf_t = \log(\frac{N}{df_t})$$

- N = number of documents in the collection
- df_t = number of documents in which term t appears



Inverse Document Frequency (IDF)

how important is a term?

rank	term	idf	rank	term	idf
1	doesn	11.66	16	creed	6.84
2	adrain	10.96	17	paulie	6.82
3	viciousness	9.95	18	packing	6.81
4	deadbeats	9.86	19	boxes	6.75
5	touting	9.64	20	forgot	6.72
6	jergens	9.35	21	ease	6.53
7	gazzo	9.21	22	thanksgivin	6.52
8	pittance	9.05	23	earns	6.51
9	balboa	8.61	24	pennsylvani	6.50
10	heavyweigh	7.18	25	promoter	6.43
11	stallion	7.17	26	befriended	6.38
12	canvas	7.10	27	exhibition	6.31
13	ve	6.96	28	collecting	6.23
14	managers	6.88	29	philadelphia	6.19
15	apollo	6.84	30	gear	6.18

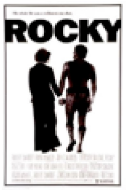
TF.IDF

how important is a term?

$$tf_t \times idf_t$$

greater when the term is frequent in in the document

greater when the term is rare in the collection (does not appear in many documents)



TF.IDF

how important is a term?

rank	term	idf	rank	term	idf
1	rocky	96.72	16	meat	11.76
2	apollo	34.20	17	doesn	11.66
3	creed	34.18	18	adrain	10.96
4	philadelphia	30.95	19	fight	10.02
5	adrian	26.44	20	viciousness	9.95
6	balboa	25.83	21	deadbeats	9.86
7	boxing	22.37	22	touting	9.64
8	boxer	22.19	23	current	9.57
9	heavyweigh	21.54	24	jergens	9.35
10	pet	21.17	25	s	9.29
11	gazzo	18.43	26	struggling	9.21
12	champion	15.08	27	training	9.17
13	match	13.96	28	pittance	9.05
14	earns	13.01	29	become	8.96
15	apartment	11.82	30	mickey	8.96

TF.IDF/Caricature Analogy

- TF.IDF: accentuates terms that are frequent in the document, but not frequent in general
- Caricature: exaggerates traits that are <u>characteristic</u> of the person (compared to the average)

TF, IDF, or TF.IDF?

adrain adrian all already also an and apartment apollo as aspiring at balboa become better big boxer boxing but by can career champion chance creed current debt doesn earns every exhibition extra far fight for gazzo gets girl go has he heavyweight her himself his if in is it keep later life living loan lovers make man match meat men mickey named nobody of paulie pet philadelphia rocky set she shot small somebody someone still store struggling supplies surprised that the they think this through time title to trainer training up want when where who willing with woman won works

TF, IDF, or TF.IDF?

ability adrain adrian already apartment apollo aspiring balboa become befriended befriends big boxer boxes boxing canvas champion chance checks chooses collecting collector creed current deadbeats debt debts distance doesn downtown earns ease easily exhibition extra extremely factory fight forgot gazzo gear gotten heavyweight his is jergens later loan lot lovers managers match meat mickey named nobody odds packing paulie pennsylvania pet philadelphia pittance promoter publicity ready rocky sells set shark sharp shot shy somebody someone stallion store struggling stunt supplies supposed surprised thanksgiving think thrilled time title touting trainer training triumph up ve viciousness visits where who willing won works

TF, IDF, or TF.IDF?

ability adrain adrian already apollo aspiring balboa beat befriended befriends better boxer boxes boxing canvas cash champion checks chooses collecting collector creed current deadbeats debt debts distance doesn downtown earns ease easily exhibition explains extra extremely factory far forgot gazzo gear giving gotten heavyweight idea interested italian jergens keep living loan lot lovers managers match meat mickey nobody odds packing paulie pennsylvania pet philadelphia pittance promoter prove publicity ready rocky sells shark sharp shop shy skills somebody spends stallion struggling stunt supplies supposed surprised thanksgiving think thrilled title touting trainer training triumph unknown ve viciousness visits want willing win won

Queries as TF.IDF Vectors

- Terms tend to appear only once in the query
- **TF** usually equals 1
- **IDF** is computed using the collection statistics

$$idf_t = \log(\frac{N}{df_t})$$

- Terms appearing in fewer documents get a higher weight

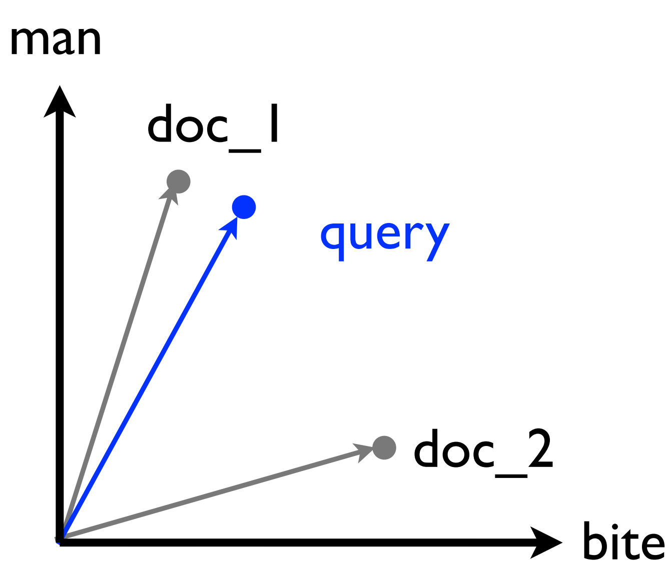
Queries as TF.IDF Vectors

examples from AOL queries with clicks on IMDB results

term 1	tf.idf	term 2	tf.idf	term 3	tf.idf
central	4.89	casting	6.05	ny	5.99
wizard	6.04	of	0.18	oz	6.14
sam	2.80	jones	3.15	iii	2.26
film	2.31	technical	6.34	advisors	8.74
edie	7.41	sands	5.88	singer	3.88
high	3.09	fidelity	7.66	quotes	8.11
quotes	8.11	about	1.61	brides	6.71
title	4.71	wave	5.68	pics	10.96
saw	4.87	3	2.43	trailers	7.83
the	0.03	rainmaker	9.09	movie	0.00
nancy	5.50	and	0.09	sluggo	9.46
audrey	6.30	rose	4.52	movie	0.00
mark	2.43	sway	7.53	photo	5.14
piece	4.59	of	0.18	cheese	6.38
date	3.93	movie	0.00	cast	0.00

Putting Everything Together

- Rank documents based on cosine similarity to the query



Vector Space Model

another cosine similarity example (binary weights)

$$\frac{\sum_{i=1}^{V} x_i \times y_i}{\sqrt{\sum_{i=1}^{V} x_i^2} \times \sqrt{\sum_{i=1}^{V} y_i^2}}$$

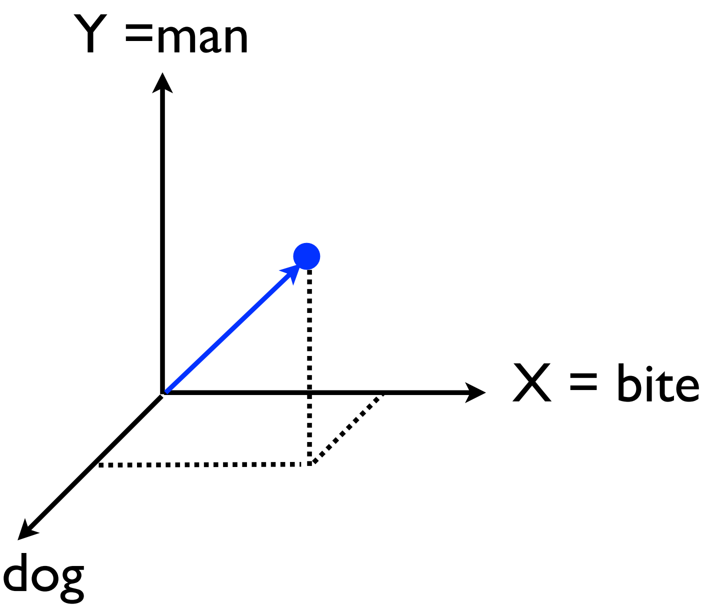
cosine([1,0,1] , [1,1,0]) =

$$\frac{(1 \times 1) + (0 \times 1) + (1 \times 0)}{\sqrt{1^2 + 0^2 + 1^2} \times \sqrt{1^2 + 1^2 + 0^2}} = 0.5$$

Independence Assumption

- The basis vectors (X, Y, Z) are <u>linearly independent</u> because knowing a vector's value on one dimension doesn't say anything about its value along another dimension

does this hold true for natural language text?



basis vectors for 3-dimensional space

Mutual Information

IMDB Corpus

- If this were true, what would these mutual information values be?

w1	w2	MI	w1	w2	MI
francisco	san	?	dollars	million	?
angeles	los	?	brooke	rick	?
prime	minister	?	teach	lesson	?
united	states	?	canada	canadian	?
9	11	?	un	ma	?
winning	award	?	nicole	roman	?
brooke	taylor	?	china	chinese	?
con	un	?	japan	japanese	?
un	la	?	belle	roman	?
belle	nicole	?	border	mexican	?

Mutual Information

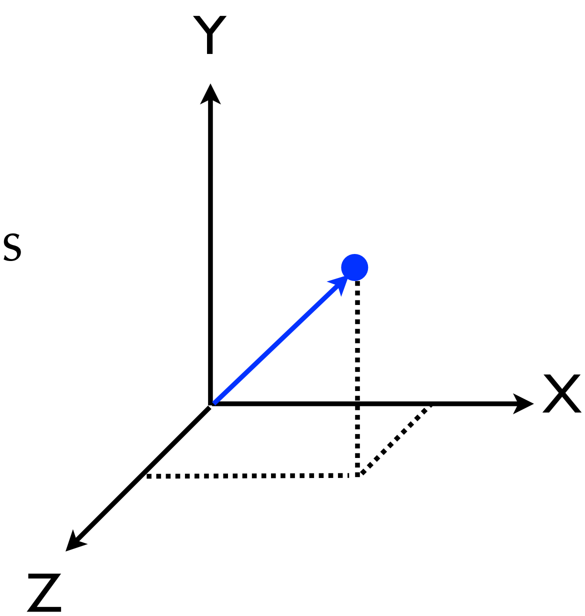
IMDB Corpus

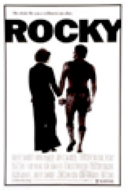
- These mutual information values should be zero!

w1	w2	MI	w1	w2	MI
francisco	san	6.619	dollars	million	5.437
angeles	los	6.282	brooke	rick	5.405
prime	minister	5.976	teach	lesson	5.370
united	states	5.765	canada	canadian	5.338
9	11	5.639	un	ma	5.334
winning	award	5.597	nicole	roman	5.255
brooke	taylor	5.518	china	chinese	5.231
con	un	5.514	japan	japanese	5.204
un	la	5.512	belle	roman	5.202
belle	nicole	5.508	border	mexican	5.186

Independence Assumption

- The vector space model assumes that terms are independent
- The fact that one occurs says nothing about another one occurring
- This is viewed as a limitation
- However, the implications of this limitation are still debated
- A very popular solution





TF.IDF

$$tf_t \times log\left(\frac{N}{df_t}\right)$$

term	tf	N	df	idf	tf.idf
rocky	19	230721	1420	5.09	96.72
philadelphia	5	230721	473	6.19	30.95
boxer	4	230721	900	5.55	22.19
fight	3	230721	8170	3.34	10.02
mickey	2	230721	2621	4.48	8.96
for	7	230721	117137	0.68	4.75

TF.IDF

- Many variants of this formula have been proposed
- However, they all have two components in common:
  - TF: favors terms that are frequent in the document
  - IDF: favors terms that do not occur in many documents

$$tf_t \times log\left(\frac{N}{df_t}\right)$$

Sub-linear TF Scaling

- Suppose ‘rocky’ occurs twice in document A and once in document B
- Is A twice as much about rocky than B?
- Suppose ‘rocky’ occurs 20 times in document A and 10 times in document B
- Is A twice as much about rocky than B?

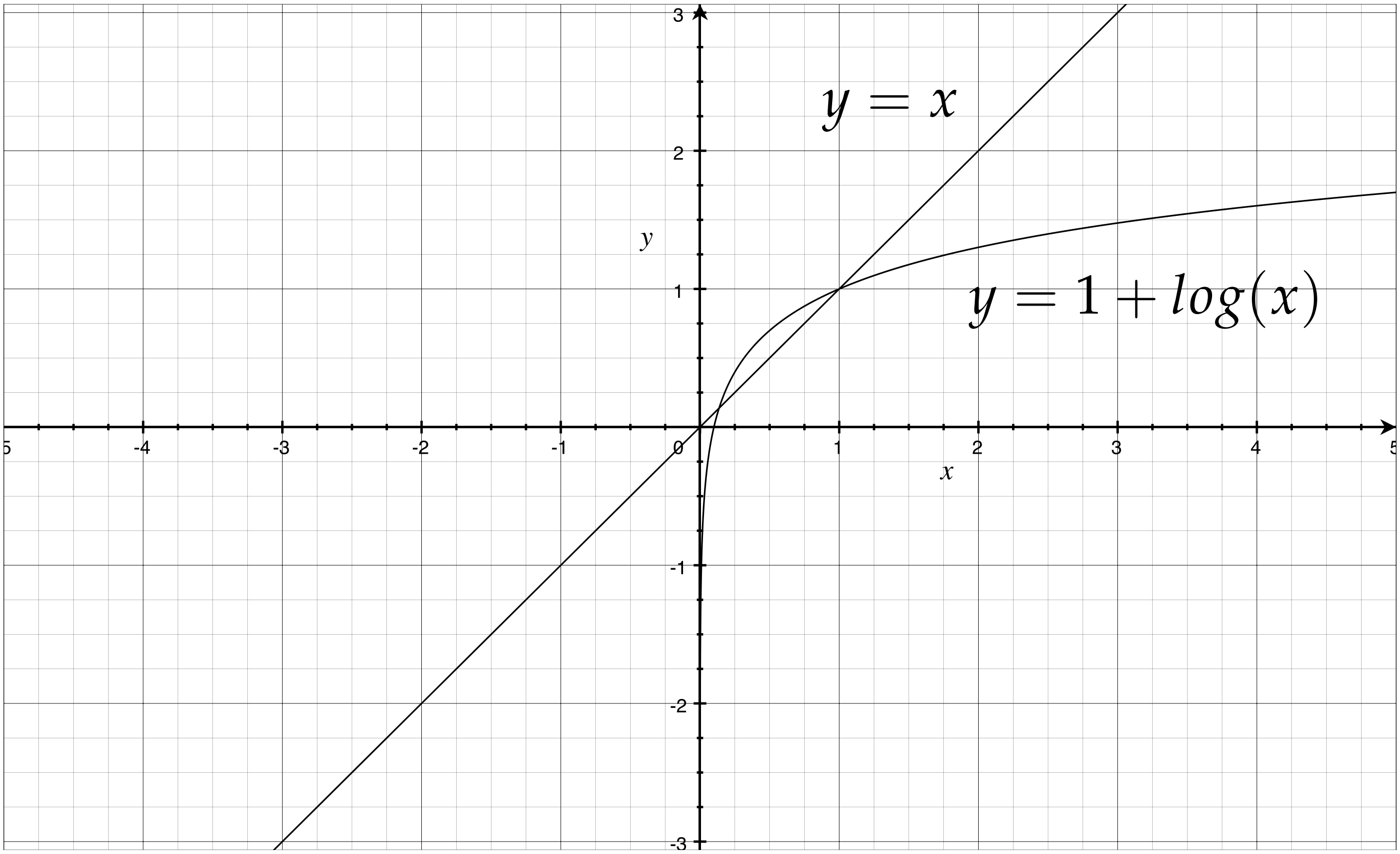
Sub-linear TF Scaling

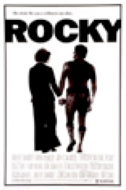
- It turns out that IR systems are more effective when they assume this is not the case

Sub-linear TF Scaling

- Assumption:
  - A document that contains ‘rocky’ 5 times is more about rocky than one that contains ‘rocky’ 1 time
  - How much more?
  - Roughly, 5 times more
  - A document that contains ‘rocky’ 50 times is more about rocky than one that contains ‘rocky’ 10 times
  - How much more?
  - Not 5 times more. Less.

Sub-linear TF Scaling



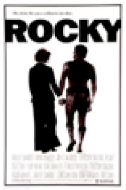


TF.IDF

what are the most important terms?

$$(1 + log(tf_t)) \times log\left(\frac{N}{df_t}\right)$$

term	tf	fw	N	df	idf	tf.idf
rocky	19	3.94	230721	1420	5.09	20.08
philadelphia	5	2.61	230721	473	6.19	16.15
boxer	4	2.39	230721	900	5.55	13.24
fight	3	2.10	230721	8170	3.34	7.01
mickey	2	1.69	230721	2621	4.48	7.58
for	7	2.95	230721	117137	0.68	2.00



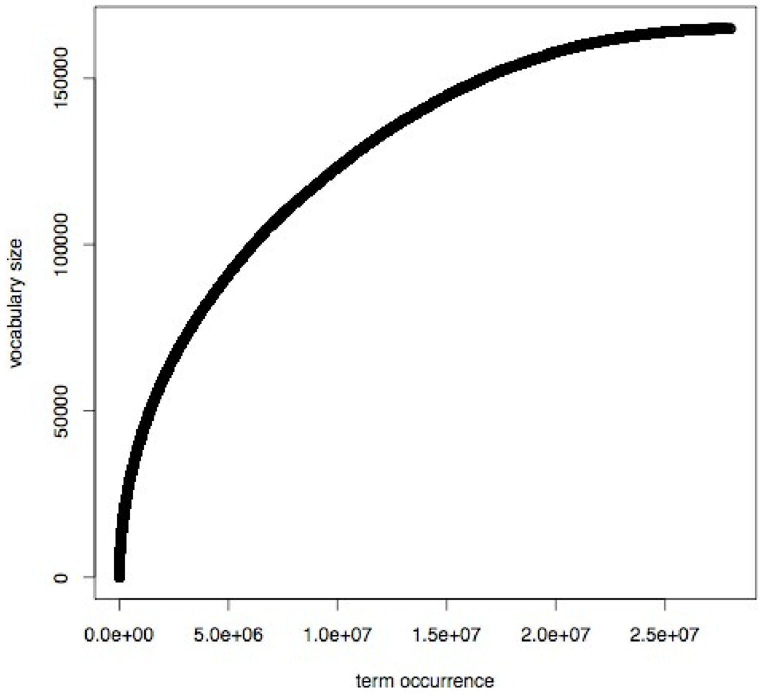
TF.IDF

what are the most important terms?

	$tf_t \times log\left(\frac{N}{df_t}\right)$	$(1 + log(tf_t)) \times log\left(\frac{N}{df_t}\right)$
term	**tf.idf (linear tf)**	**tf.idf (sub-linear tf)**
rocky	96.72	20.08
philadelphia	30.95	16.15
boxer	22.19	13.24
fight	10.02	7.01
mickey	8.96	7.58
for	4.75	2.00

Remember Heaps' Law?

- As we see more and more text, the frequency of <u>new</u> words decreases



Remember Heaps' Law?

- Put differently, as we see more text, it becomes more rare to encounter previously unseen words
- This means that the text mentions the same words over and over
- Once we see a word, we're likely to see it again
- This <u>may</u> be a motivation for sub-linear TF scaling
- Explanations are good. But, IR is an empirical science
- This works in practice

Vector Space Model

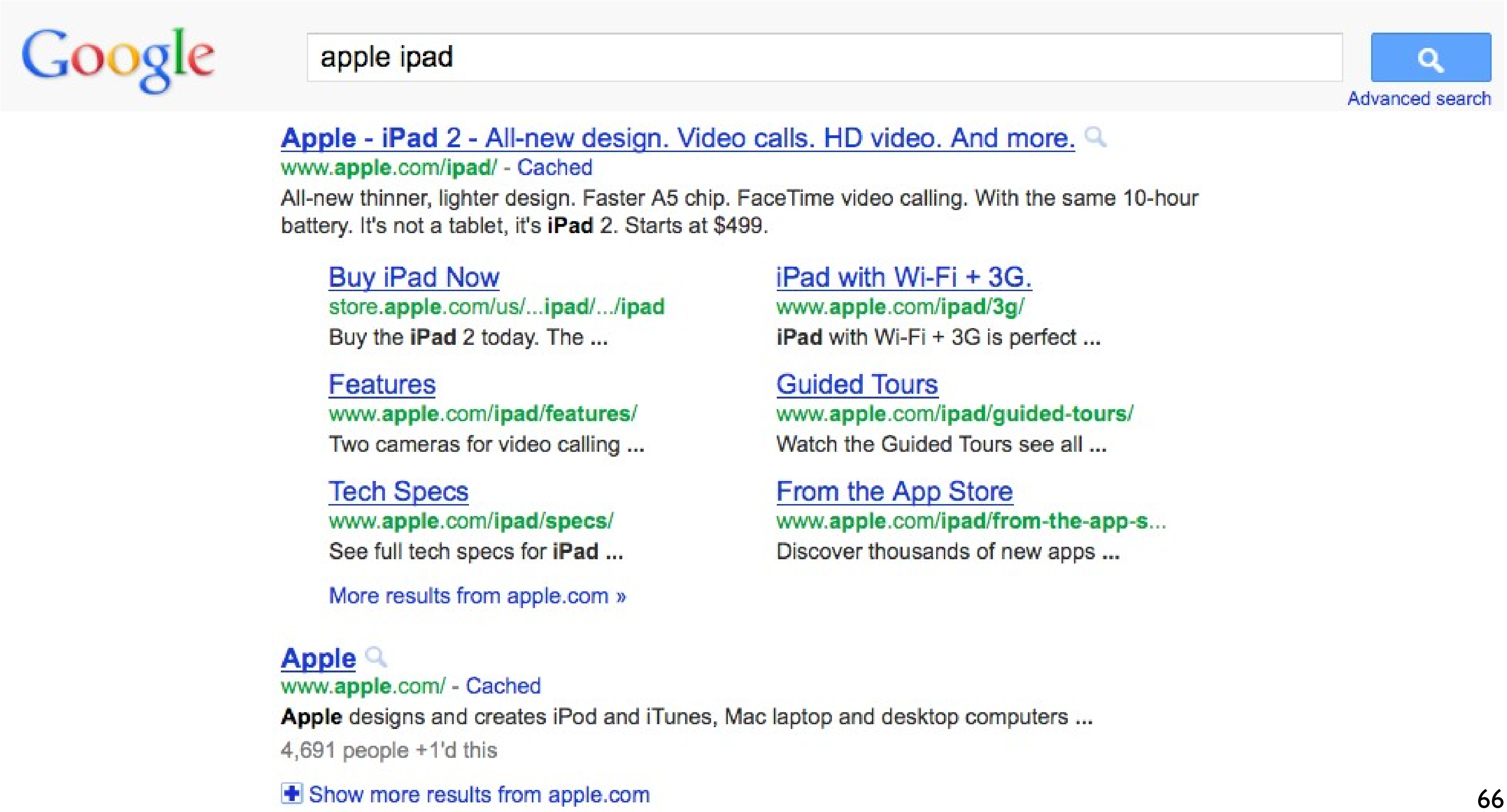
- Any text can be seen as a vector in V-dimensional space
  - a document
  - a query
  - a sentence
  - a word
  - an entire encyclopedia
- Rank documents based on their cosine similarity to query
- If a document is similar to the query, it is likely to be relevant (remember: topical relevance!)

Vector Space Representation

- A power tool!
- A lot of problems in IR can be cast as:
  - Find me _____ that is similar to _____ !
- As long as _____ and _____ are associated with text, one potential solution is:
  - represent these items as tf.idf term-weight vectors and compute their cosine similarity
  - return the items with the highest similarity

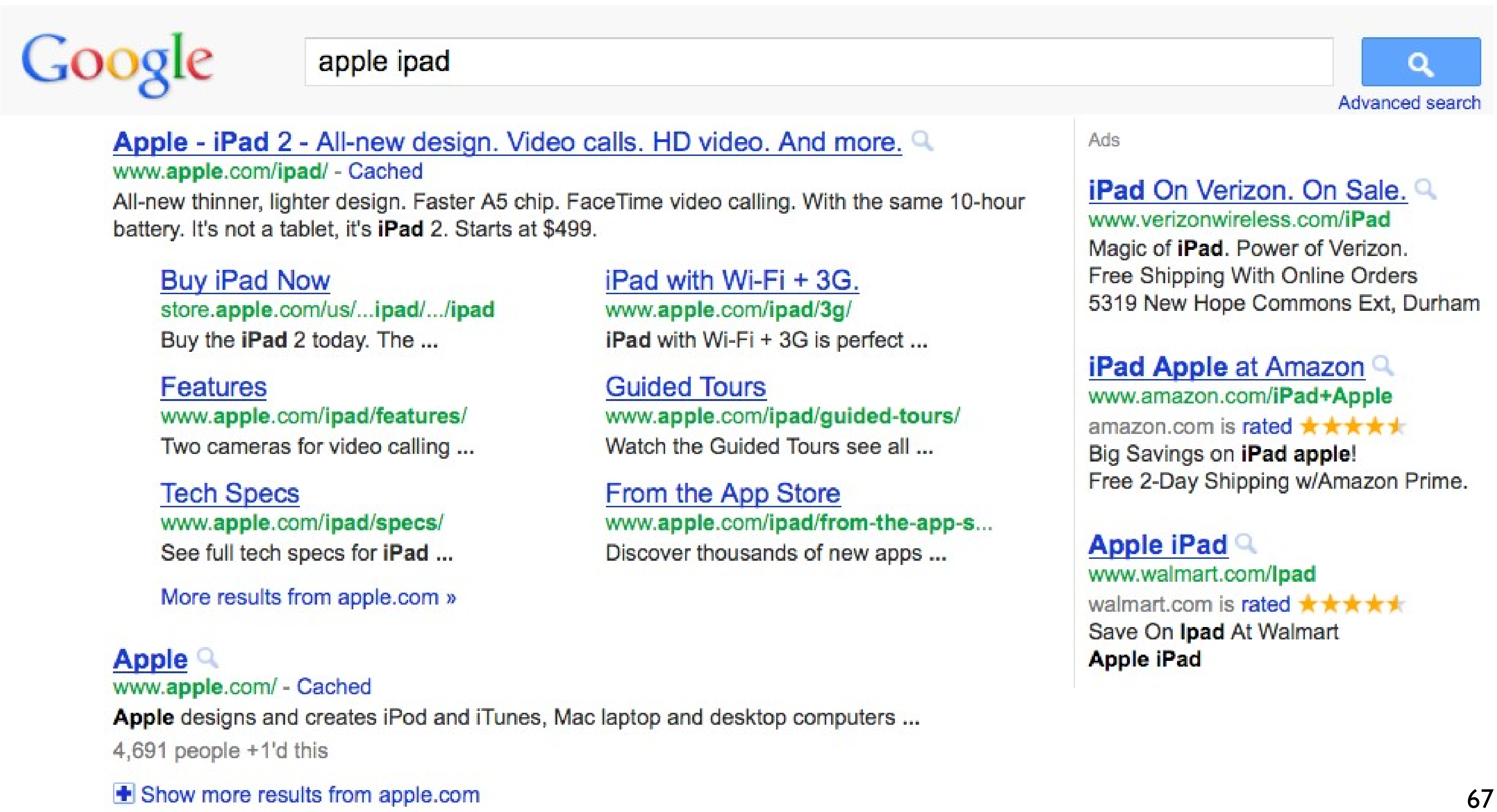
Vector Space Representation

- Find me documents that are similar to this query



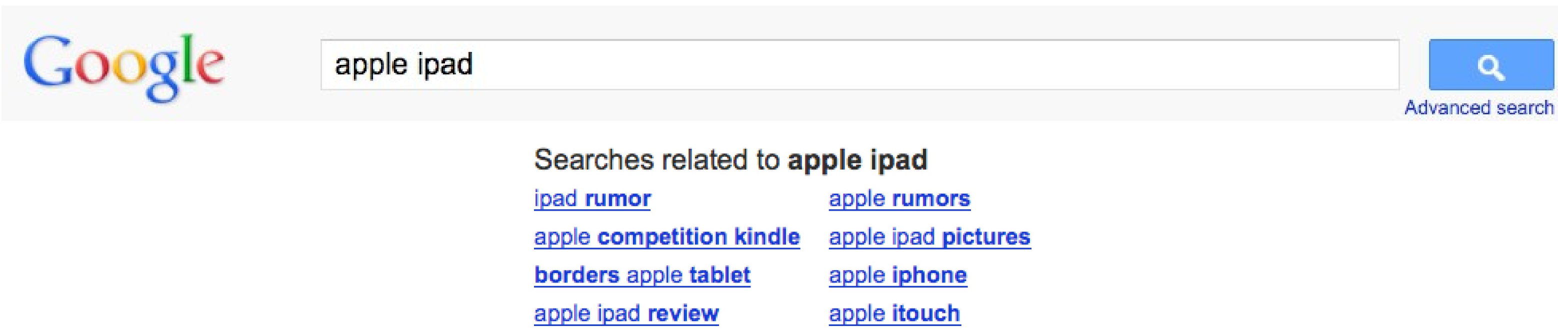
Vector Space Representation

- Find me ads that are similar to these results



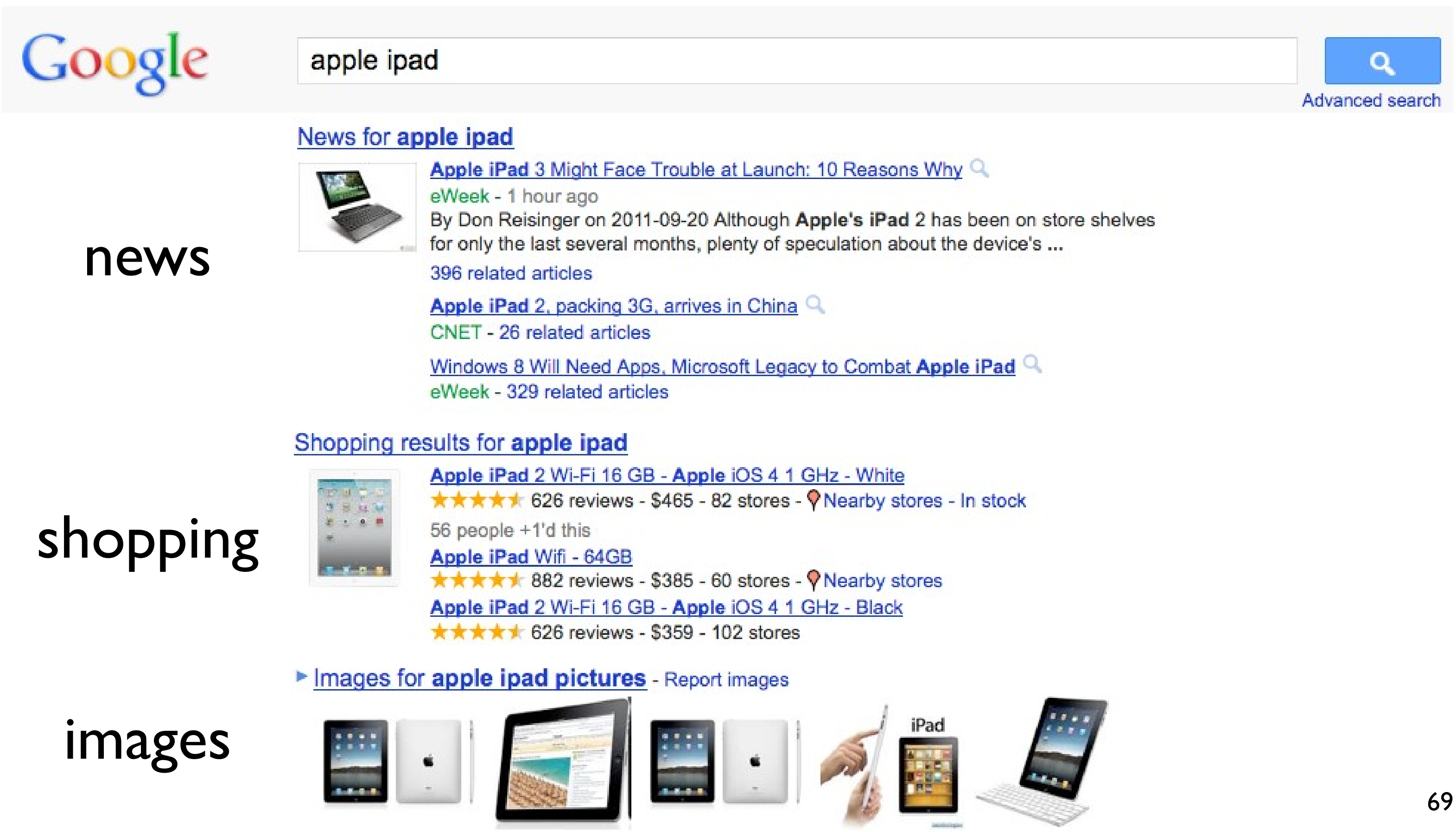
Vector Space Representation

- Find me queries that are similar to this query



Vector Space Representation

- Find me search engines that are similar to this query



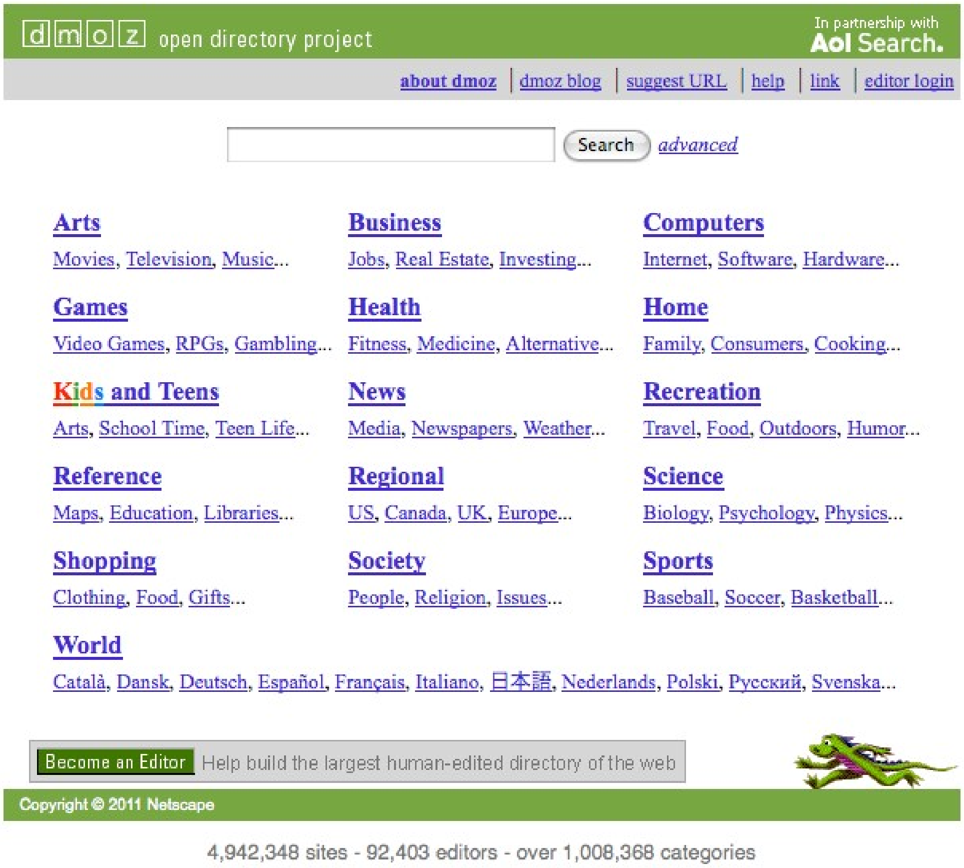
news

shopping

images

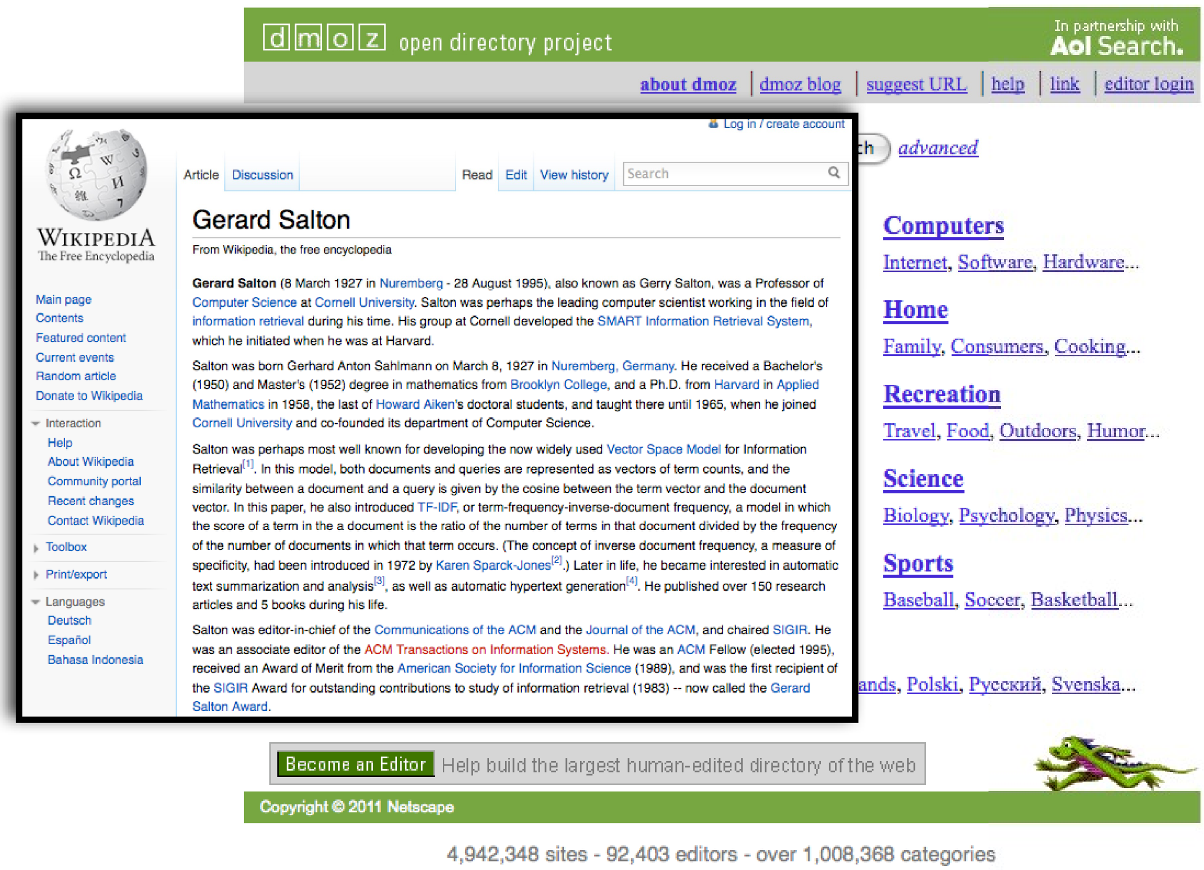
Vector Space Representation

- Topic categorization: automatically assigning a document to a category



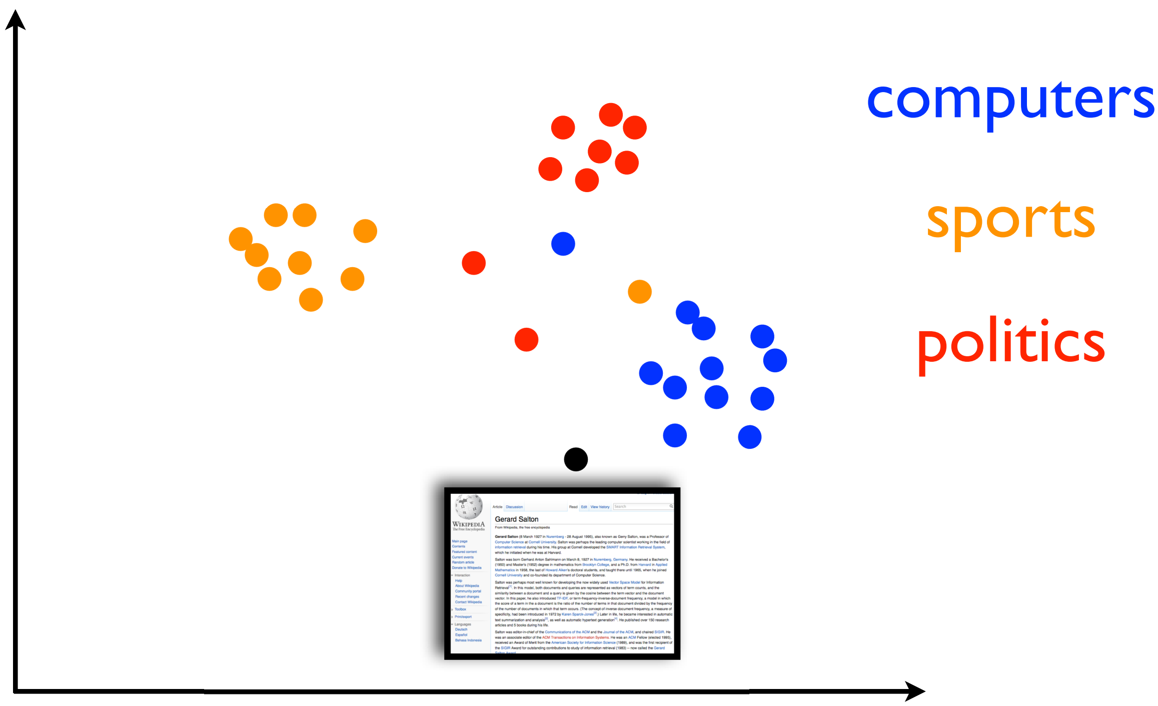
Vector Space Representation

- Find me documents (with a known category assignment) that are similar to this document



Vector Space Representation

- Find me documents (with a known category assignment) that are similar to this document



Vector Space Representation

So, does the vector space representation solve all problems?

Advertisement Placement

- Find me ads similar to this this document

Anatidaephobia - The Fear That You are Being Watched by a Duck

December 08, 2008 by Tammy Duffey

Single page Font Size Read comments (44) Share

WAS THIS HELPFUL? 137 CLICK TO VOTE

Popular searches: YouTube | Rihanna | Tiger Woods | Search more

What Is Anatidaephobia?

Anatidaephobia is defined as a pervasive, irrational fear that one is being watched by a duck. The anatidaephobic individual fears that no matter where they are or what they are doing, a duck watches.

Anatidaephobia is derived from the Greek word "anatidae", meaning ducks, geese or swans and "phobos" meaning fear.

What Causes Anatidaephobia?

As with all phobias, the person coping with Anatidaephobia has experienced a real-life trauma. For the anatidaephobic individual, this trauma most likely occurred during childhood.

Perhaps the individual was intensely frightened by some species of water fowl. Geese and swans are relatively well known for their aggressive tendencies and perhaps the anatidaephobic person was actually bitten or flapped at. Of course, the Far Side comics did little to minimize the fear of being watched by a duck.

Summary

- Any text can be seen as a vector in V-dimensional space
  - a document
  - a query
  - a sentence
  - a word
  - an entire encyclopedia
- Rank documents based on their cosine similarity to query
- If a document is similar to the query, it is likely to be relevant (remember: topical relevance!)