Vector Space Model

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

September 19, 2011

The Search Task

- Given a query and a corpus, find relevant items

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

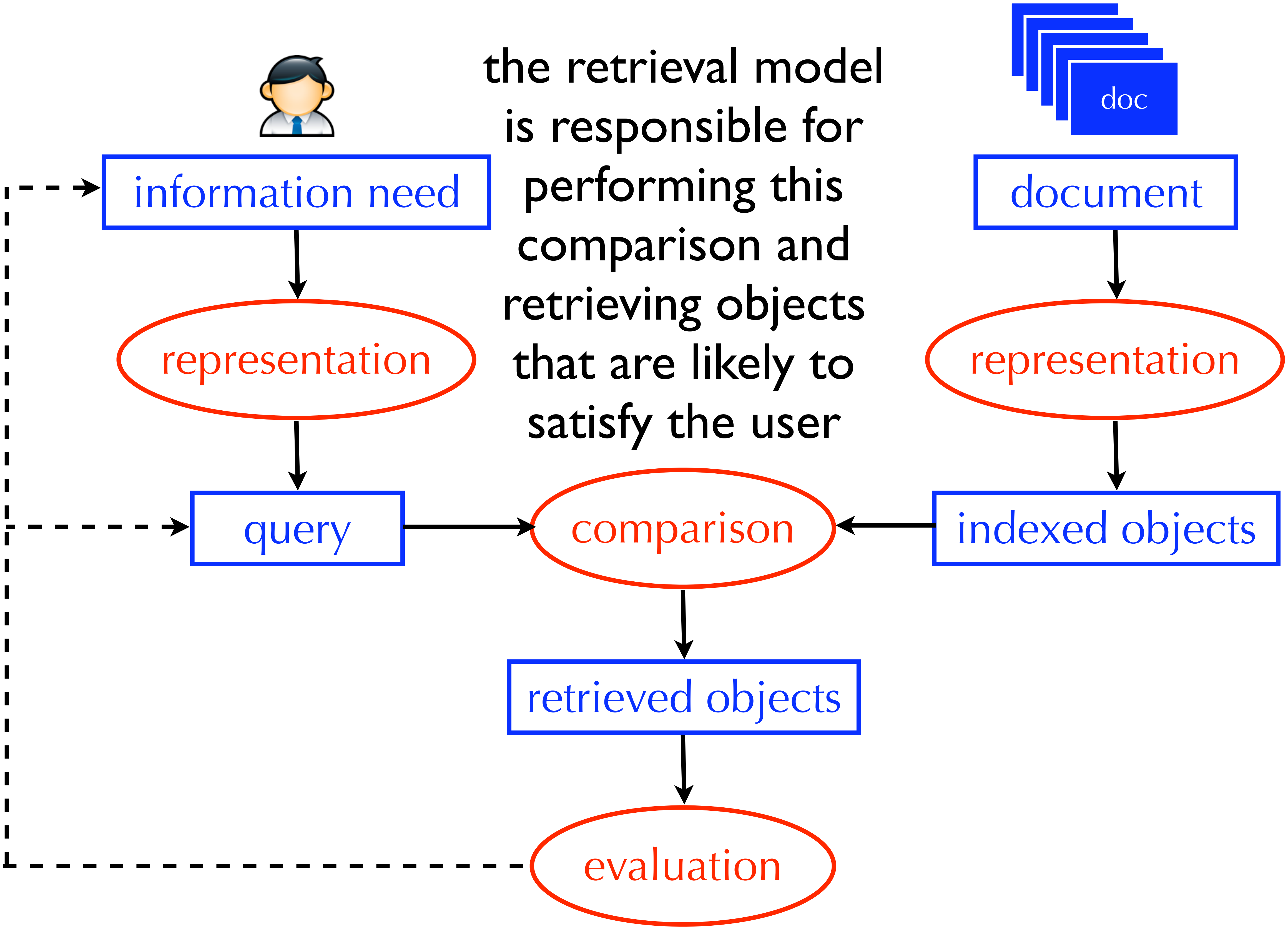
  corpus: a repository of textual documents

  relevance: satisfaction of the user's information need

What is a Retrieval Model?

- A formal representation of the process of matching a document to a query
- Objective: to predict whether a particular document is relevant to the user's information need

Basic Information Retrieval Process



information need

representation

query

document

doc

representation

indexed objects

comparison

retrieved objects

evaluation

the retrieval model is responsible for performing this comparison and retrieving objects that are likely to satisfy the user

Boolean Retrieval Models

- The user describes the information need using boolean constraints (e.g., AND, OR, and AND NOT)
- Unranked Boolean Retrieval Model: retrieves documents that satisfy the constraints (results returned in no particular order)
- Ranked Boolean Retrieval Model: retrieves documents that satisfy the constraints and ranks them based on the number of redundant ways in which each document satisfies the constraints
- Also know as ‘exact-match’ retrieval models
- Advantages and disadvantages?

Boolean Retrieval Models

- Advantages:
  - Easy from the system's perspective
  - Users get transparency: it is easy to understand why a document was retrieved
  - Users get control: easy to determine whether the query is too specific (few results) or too broad (many results)
- Disadvantages:
  - Difficult from the user's perspective
  - What are the right constraints?

Relevance

- Many factors affect whether a document satisfies a particular user's information need
- Topicality, novelty, freshness, authority, formatting, reading level, assumed level of expertise, etc.
- Topical relevance: the document is on the same topic as the query
- User relevance: everything else!
- For now, we will focus on retrieval models that predict topical relevance

Relevance

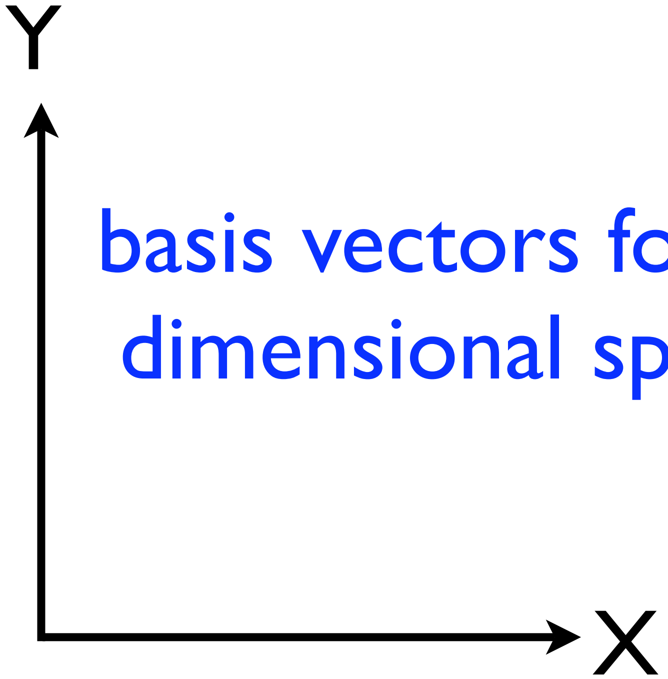
- Focusing on topical relevance does not mean we're ignoring everything else!
- It only means we're focusing on one (of many) criteria by which the user will judge the utility of the documents retrieved
- And, it's an important criterion
- It may also be easier than the others :-)
- But, not trivial by any means

Introduction to Best-Match Retrieval Models

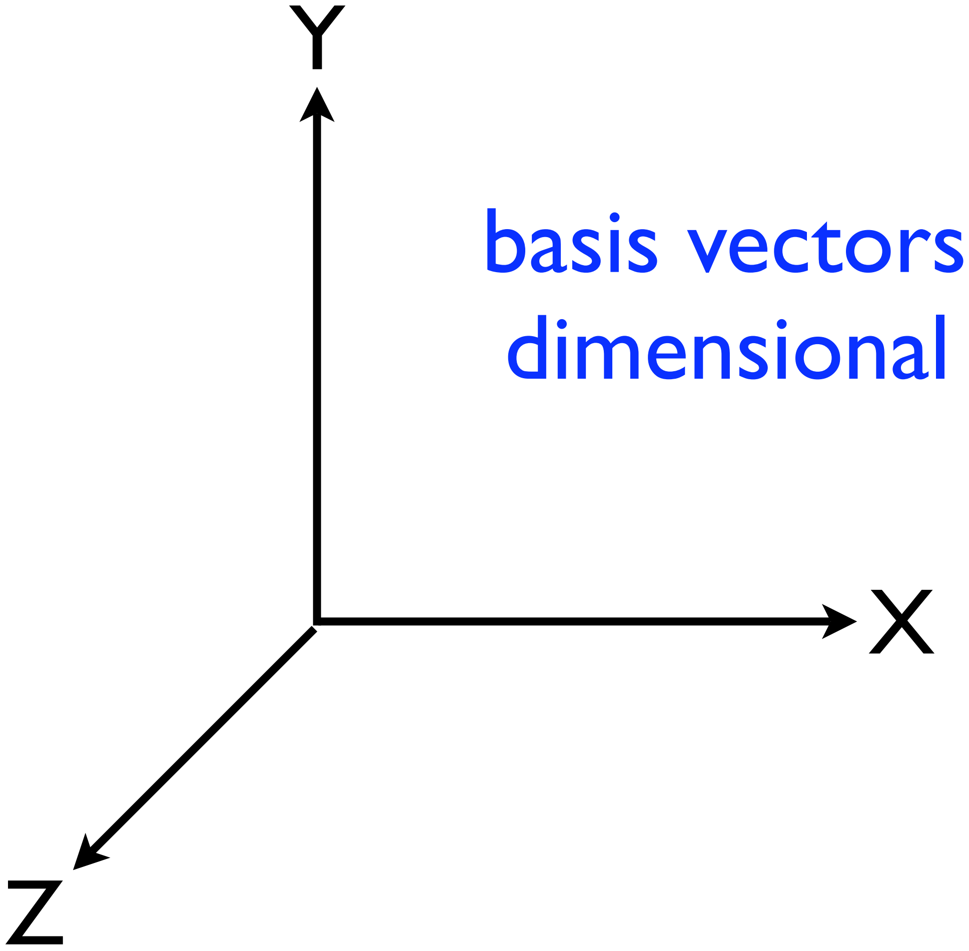
- So far, we've discussed 'exact-match' models
- Today, we start discussing 'best-match' models
- Best-match models predict the degree to which a document is relevant to a query
- Ideally, this is would be expressed as RELEVANT(q,d)
- In practice, it is expressed as SIMILAR(q,d)
- Get excited,the vector space model is extremely flexible and powerful!

What is a Vector Space?

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



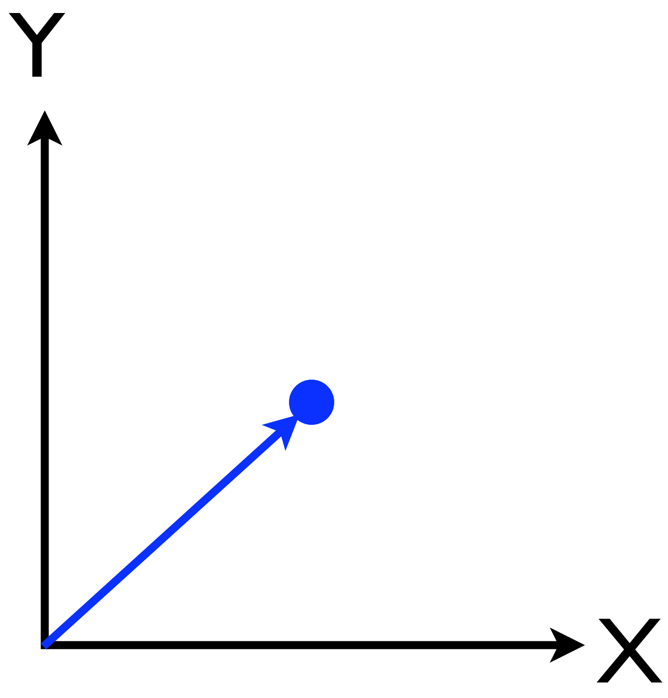
basis vectors for 2-dimensional space

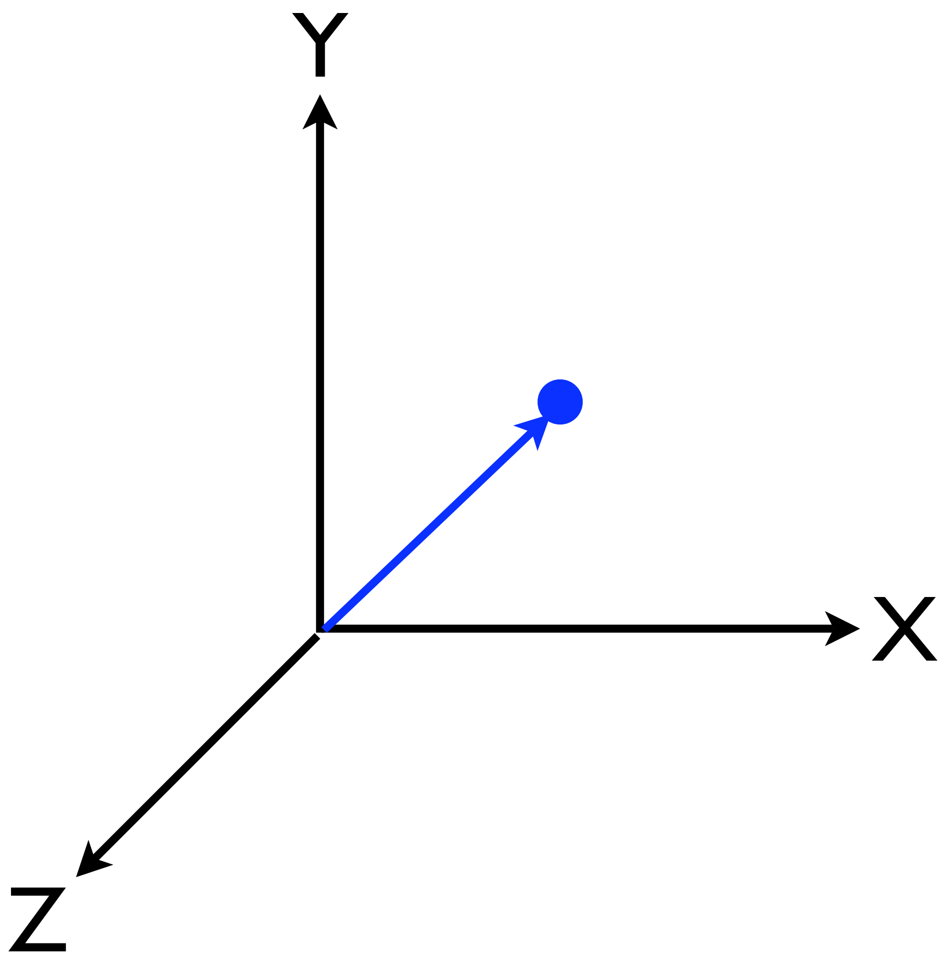


basis vectors for 3-dimensional space

What is a Vector?

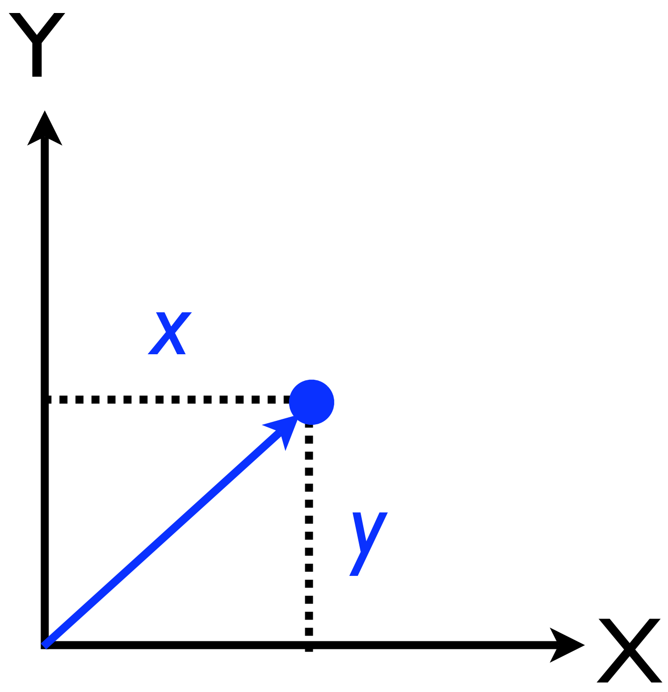
- A vector is a point in a vector space and has length and direction (from the origin to the point)

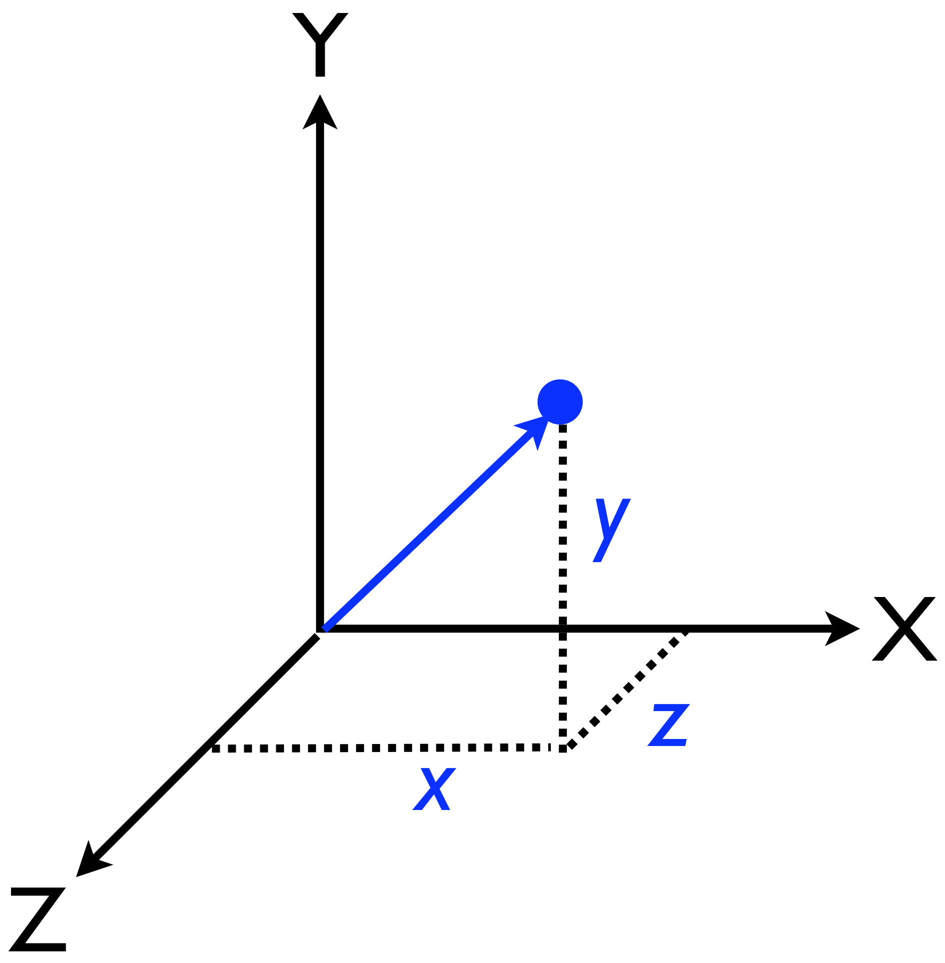




What is a Vector?

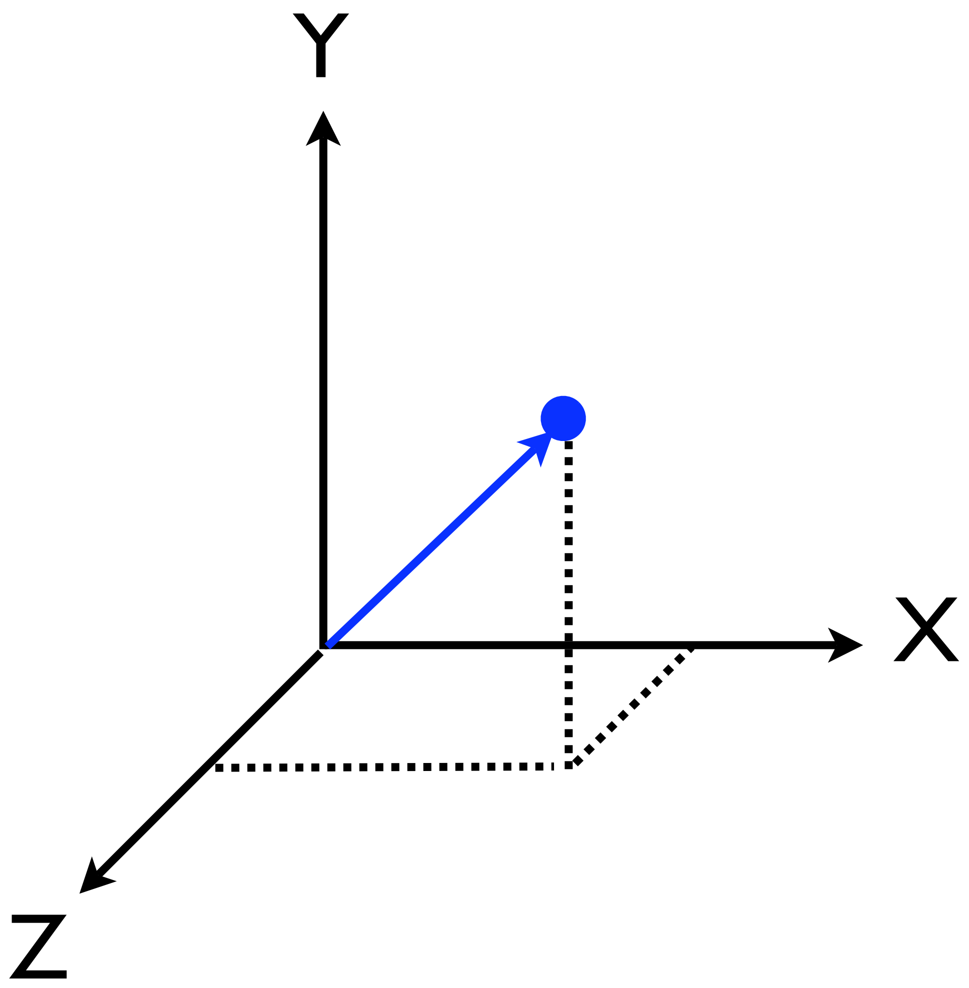
- A 2-dimensional vector can be written as **[*x*,*y*]**
- A 3-dimensional vector can be written as **[*x*,*y*,*z*]**





What is a Vector Space?

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



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

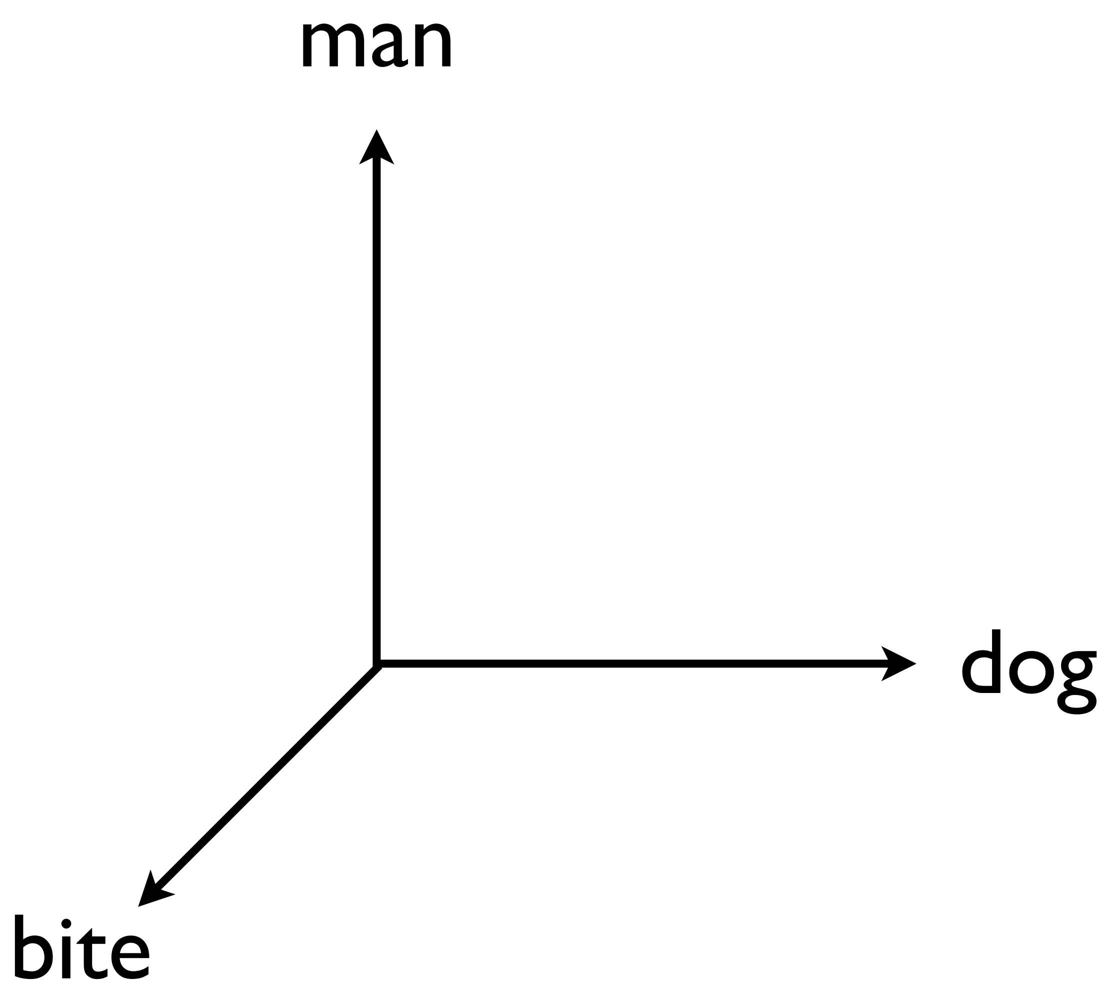
- 1 = the word appears in the document
- 0 = the word does <u>not</u> appear in the document
- Does not represent word frequency, word location, or word order information

Vector Space Representation of Text

- 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...
- An 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 indexed terms: dog, bite, man (i.e., **V=3)**
- Why? Because it's hard to visualize more the 3 dimensions

Vector Space Representation

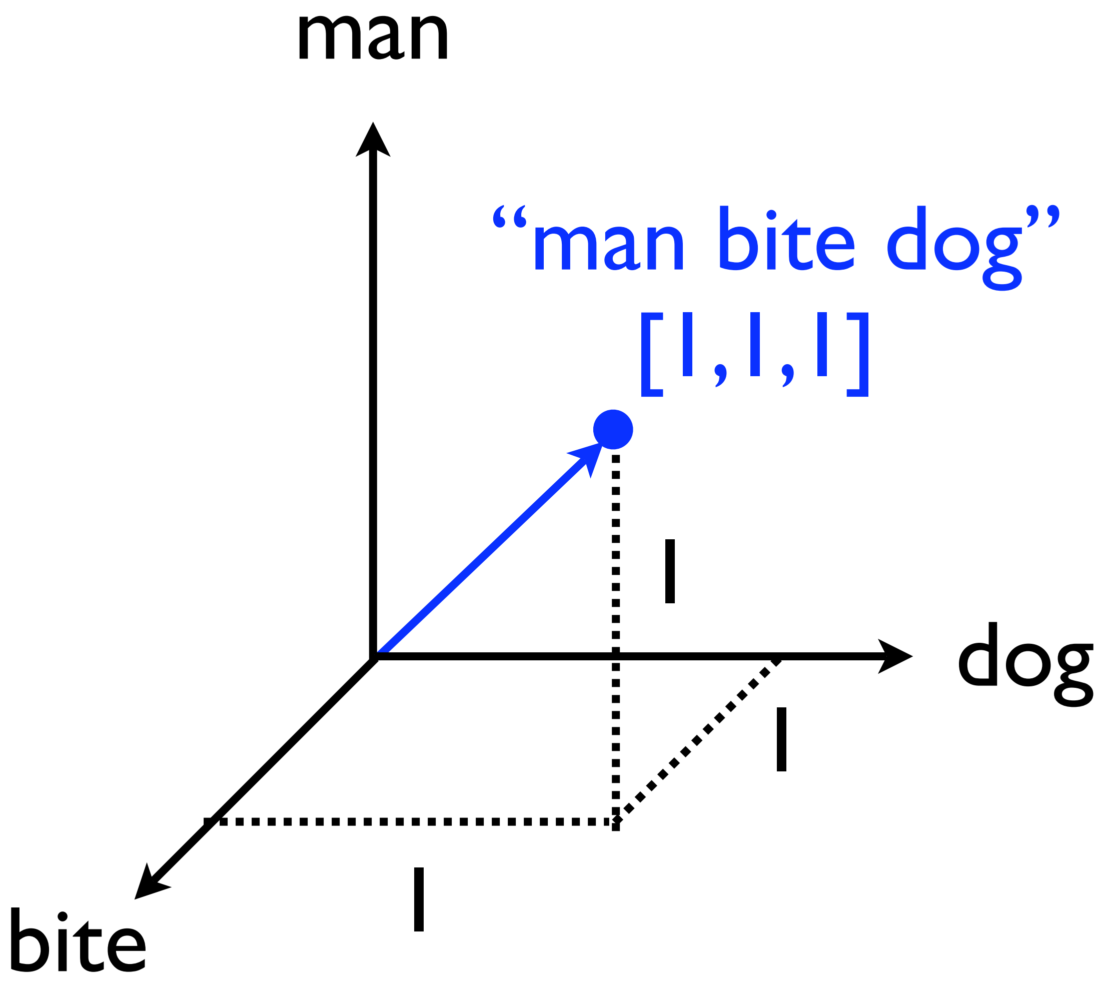
- A span of text is a vector in V-dimensional space, where V is the size of the vocabulary



Vector Space Representation

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

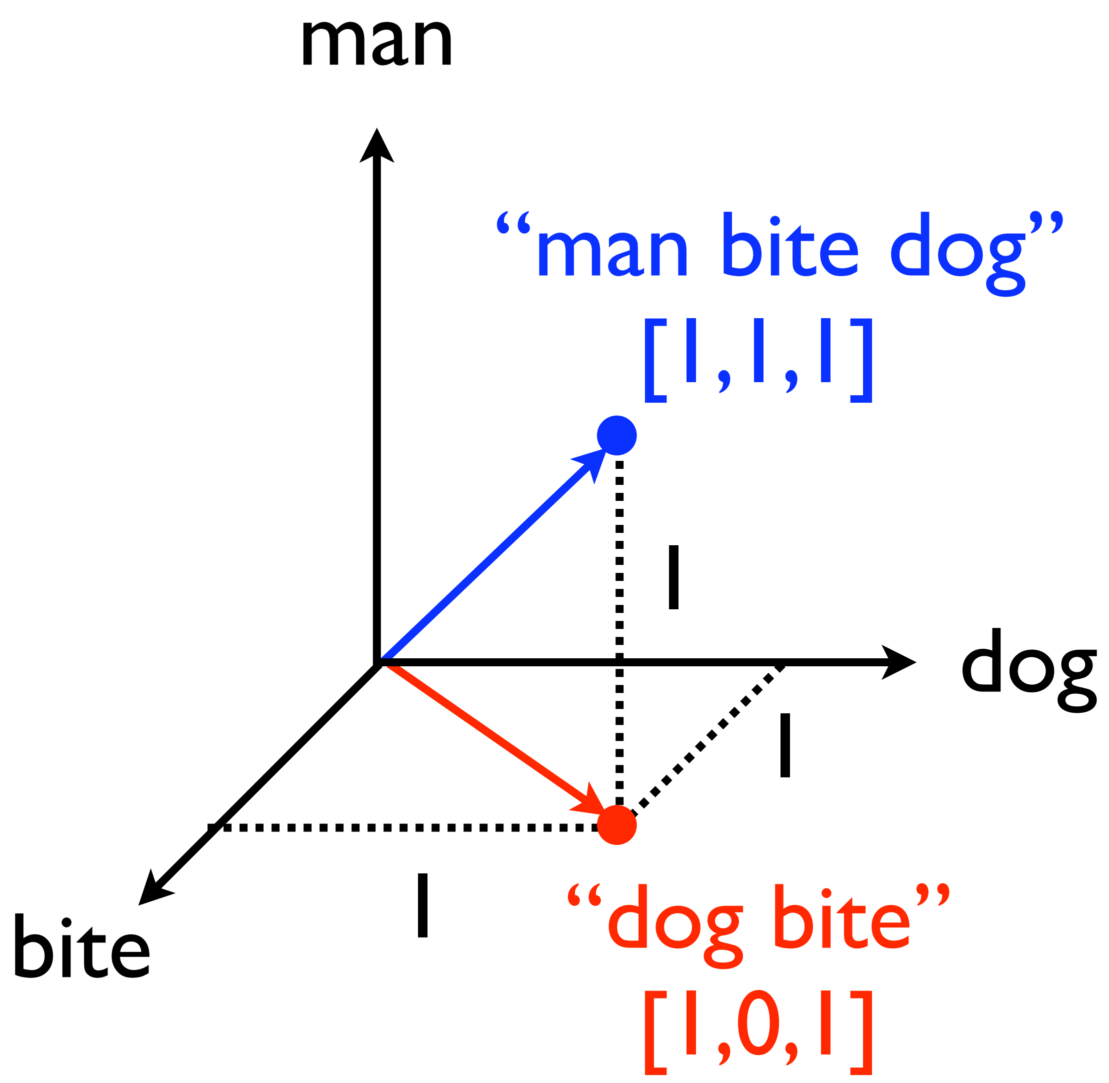
	dog	*man*	*bite*
doc_1	1	1	1



Vector Space Representation

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

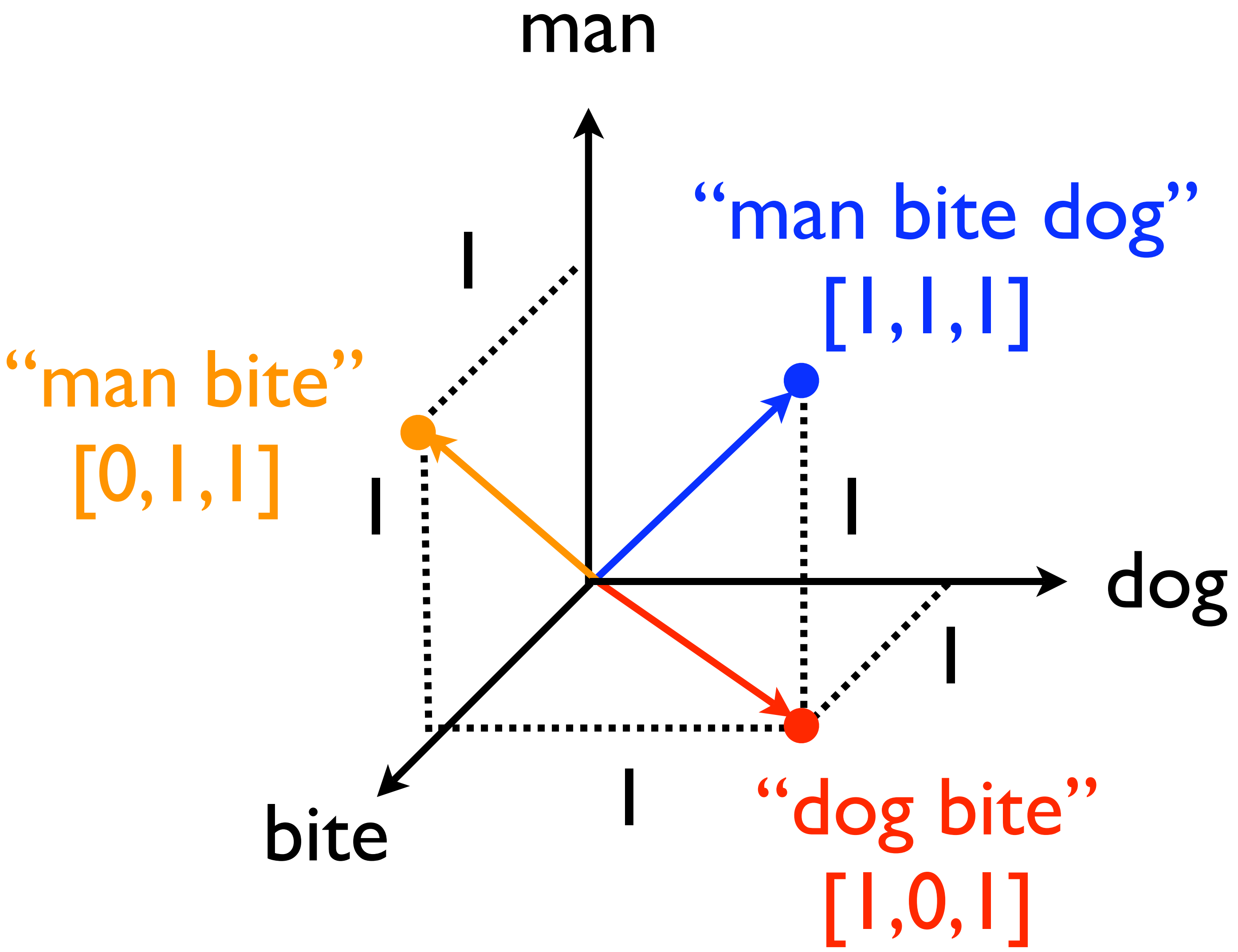
	dog	*man*	*bite*
doc_1	1	1	1
doc_2	1	0	1



Vector Space Representation

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

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



man

dog

bite

"man bite dog"
[1,1,1]

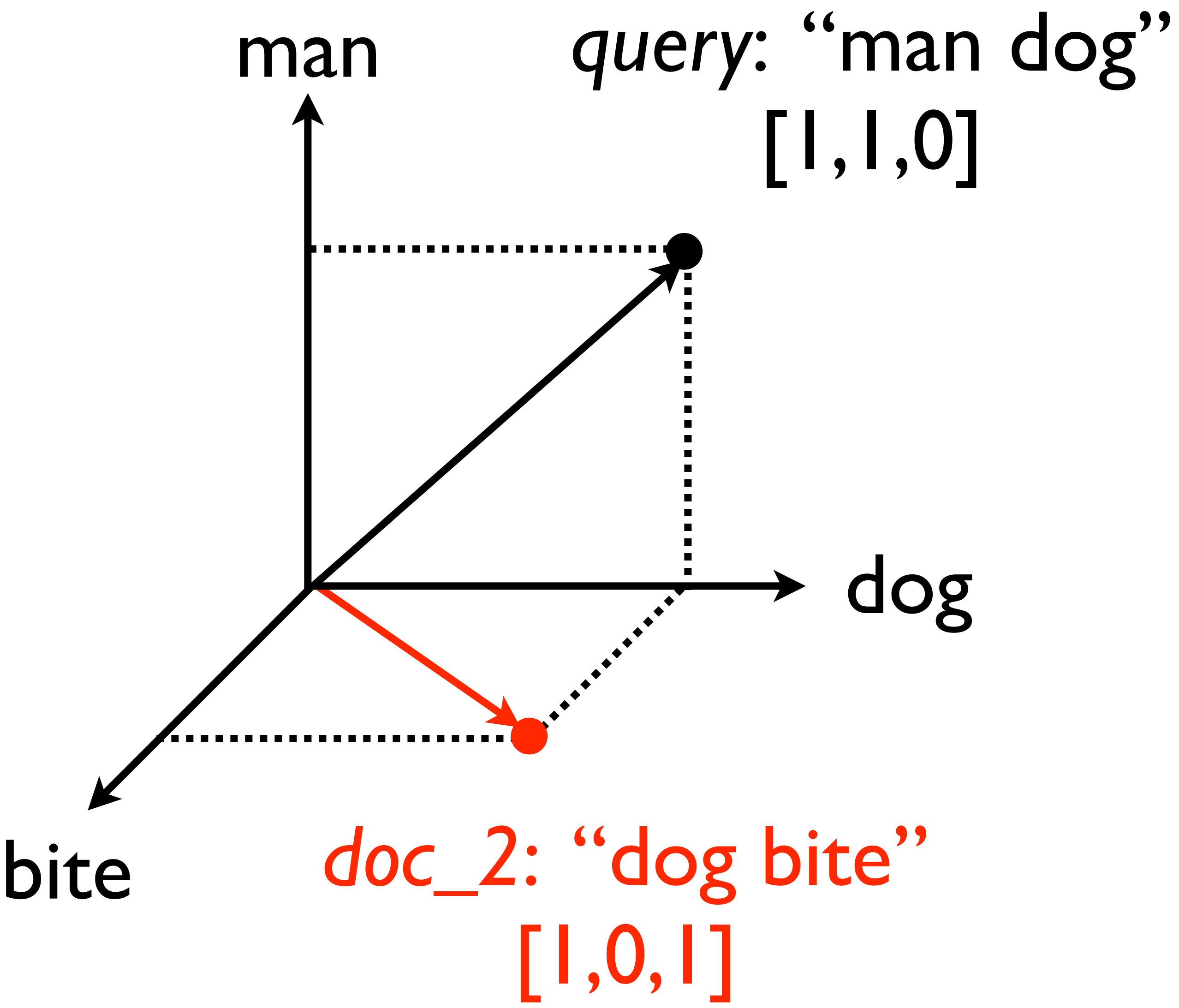
"man bite"
[0,1,1]

"dog bite"
[1,0,1]

Vector Space Representation

- Both documents and queries can be represented as vectors

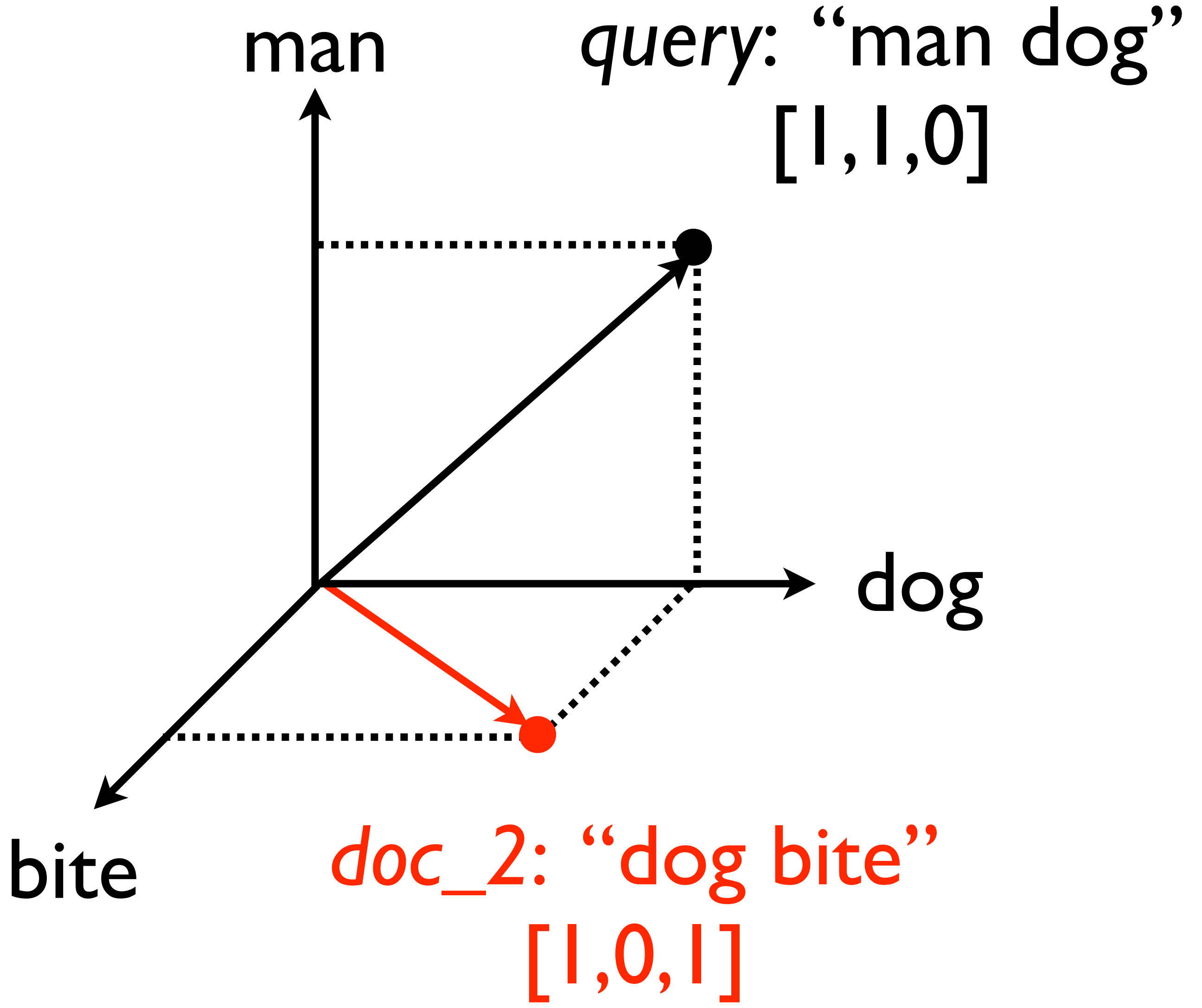
	dog	*man*	*bite*
doc_2	1	0	1
query	1	1	0



Vector Space Model

- The vector space model scores and ranks documents based on their vector-space similarity to the query

	dog	*man*	*bite*
doc_2	1	0	1
query	1	1	0



Vector Space Similarity

- There are many ways to compute the similarity between two vectors
- We will focus on one similarity measure: cosine similarity
- Simple and effective
- Corresponds to the cosine of the angle between the two vectors

Vector Space Similarity

- To motivate cosine similarity, let's start with another similarity measure, the inner product

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

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

- The inner product measures the number of terms that appear at least once in both spans of text
- Scoring documents based on their inner-product with the query has one major issue. Any ideas?
- Hint: documents have widely varying lengths

The Inner Product

- What is more relevant?
  - A 50-word document which contains 3 of the query-terms?
  - A 100-word document which contains 6 of the query-terms?
- The inner-product doesn’t account for the fact that documents have widely varying lengths
- So, it favors long documents

The Cosine Similarity

- Measures the cosine of the angle between the two vectors
- The numerator is the inner product
- The denominator "normalizes" for document length
- Ranges from 0 to 1 (equals 1 if the vectors are identical)
- Determines whether the two vectors are pointing in the same direction

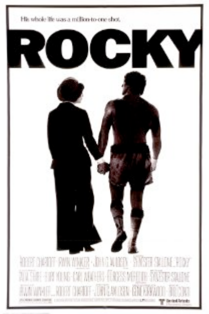
$$\frac{\sum_{i=1}^{V} x_i \times y_i}{\underbrace{\sqrt{\sum_{i=1}^{V} x_i^2}}_{\text{length of vector x}} \times \underbrace{\sqrt{\sum_{i=1}^{V} y_i^2}}_{\text{length of vector y}}}$$

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

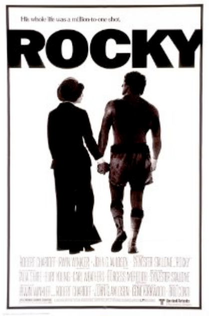


Term-Weighting

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

Term-Frequency (TF)

how important is a term?

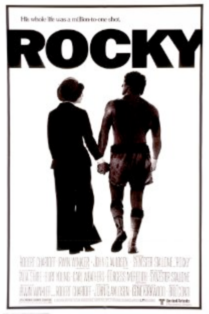
- A term’s frequency in the document is an important indicator of what the document is about
    - If “rocky” appears 19 times in a document, it’s probably about “rocky”
- However, not all terms are equally important
- “Rocky” is more important than “a” (which appears 20 times) because “a” appears in almost every document
- Terms that appear in many documents have little discriminating power in determining relevance
- We need to attenuate the contribution from terms that are frequent in the document, but frequent in general

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
- Note: a term's *idf* value is not specific to a document, but specific to the collection!
- What is the *idf* value of a term that appears in every document in the collection?



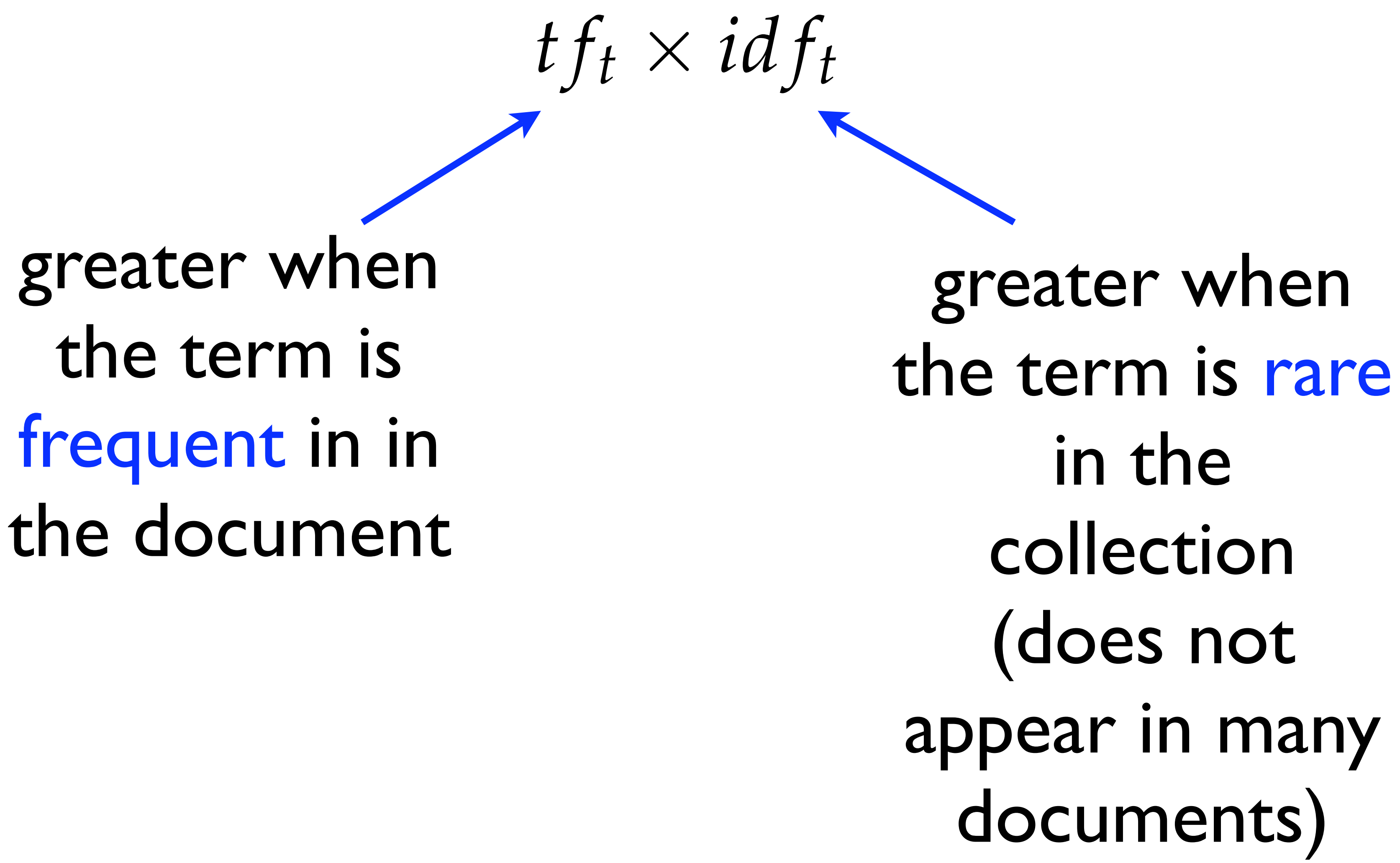
Inverse Document Frequency (IDF)

how important is a term?

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

TF.IDF

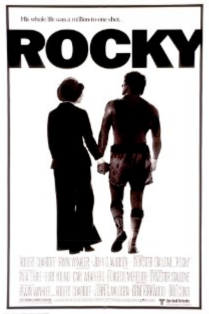
how important is a term?



$$tf_t \times idf_t$$

greater when the term is frequent in in the document

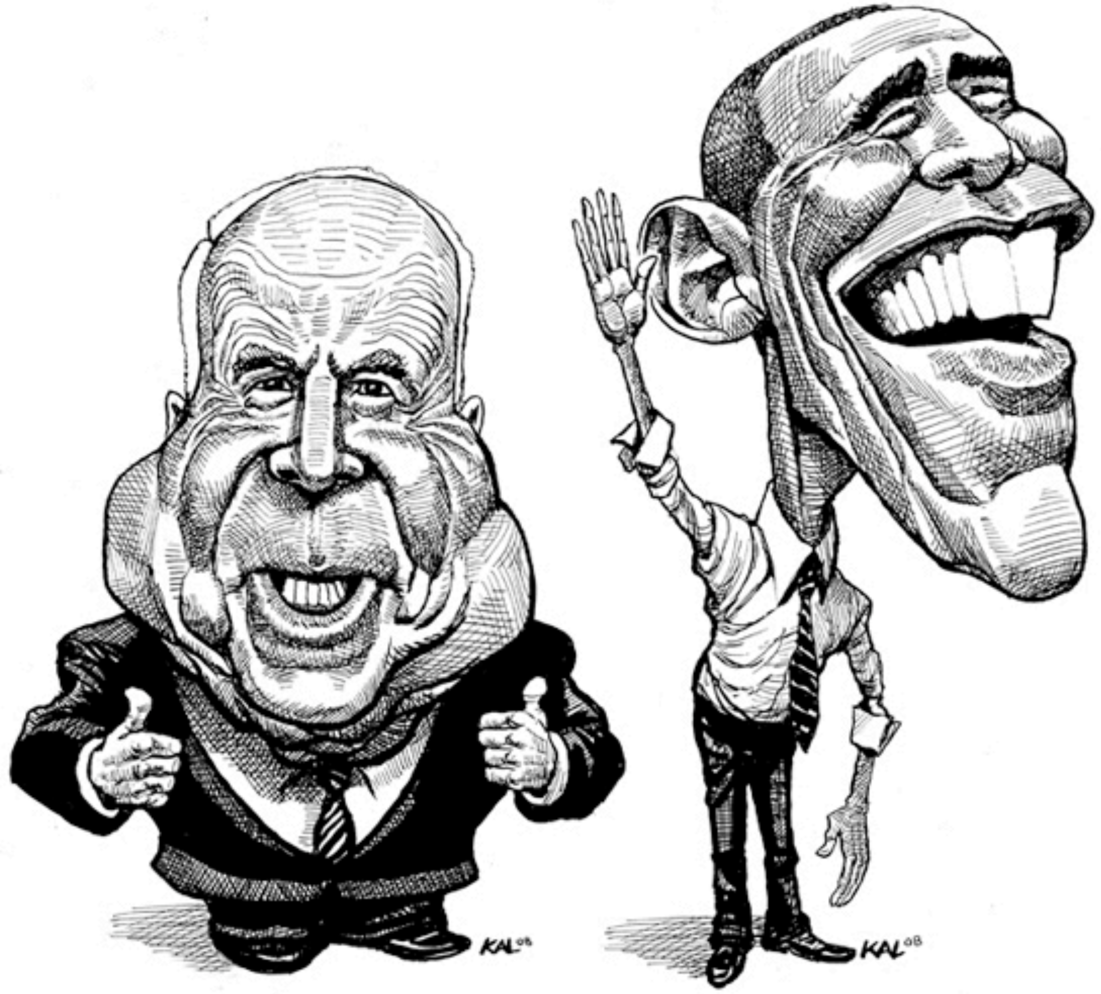
greater when the term is rare in the collection (does not appear in many documents)



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



- What's the relationship between TF.IDF and these representations of Mr. McCain and Mr. Obama?

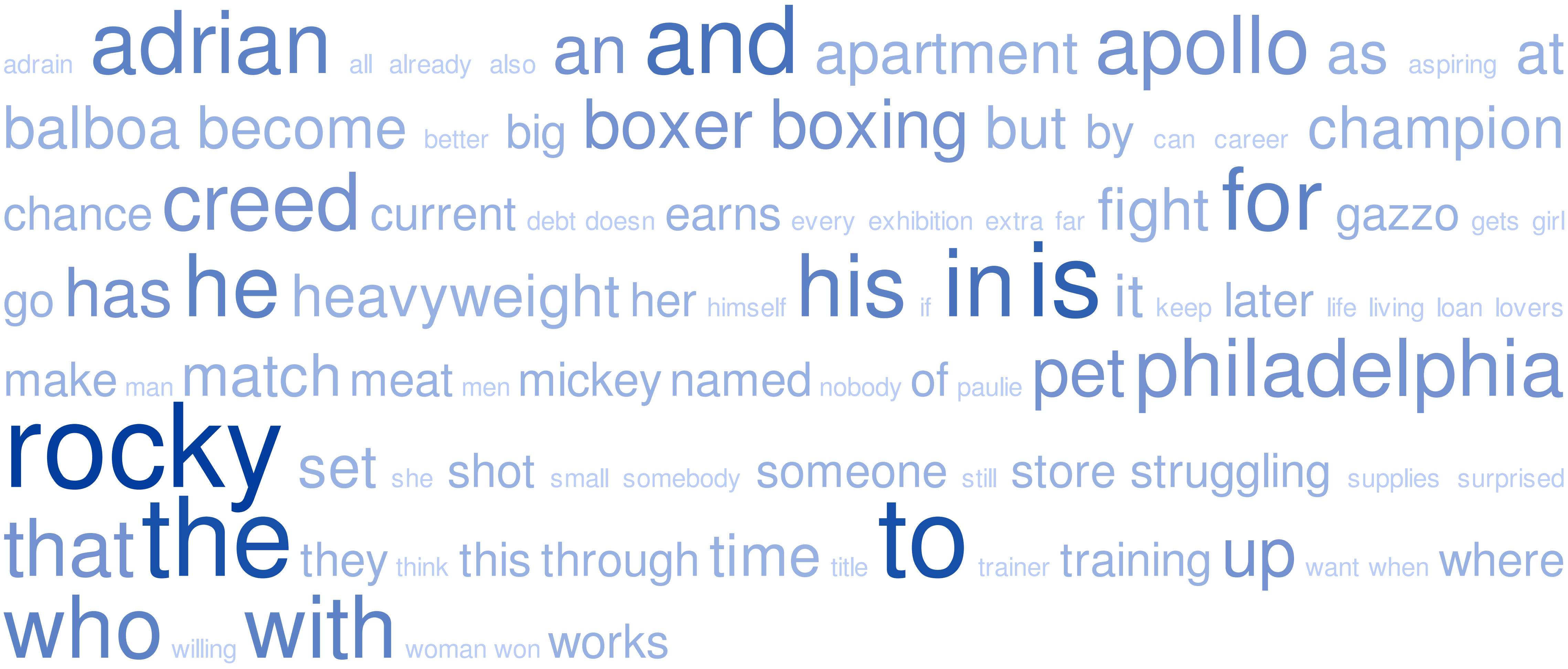
TF.IDF/Caricature Analogy

KAL08
KAL08

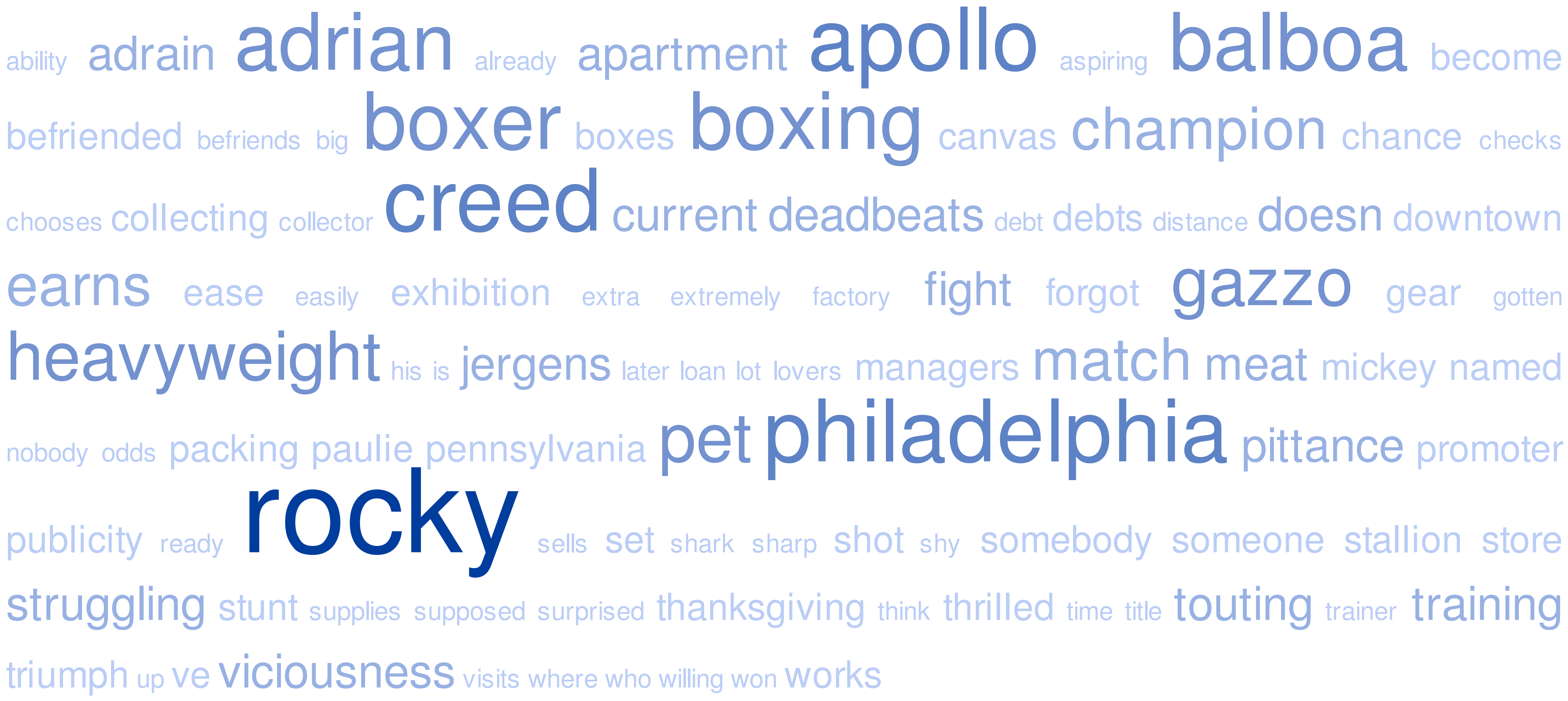
- TF.IDF: accentuates terms that are frequent in the document, but not frequent in general

- Caricature: exaggerates traits that are <u>characteristic</u> of the person, compared to the average

TF, IDF, or TF.IDF?

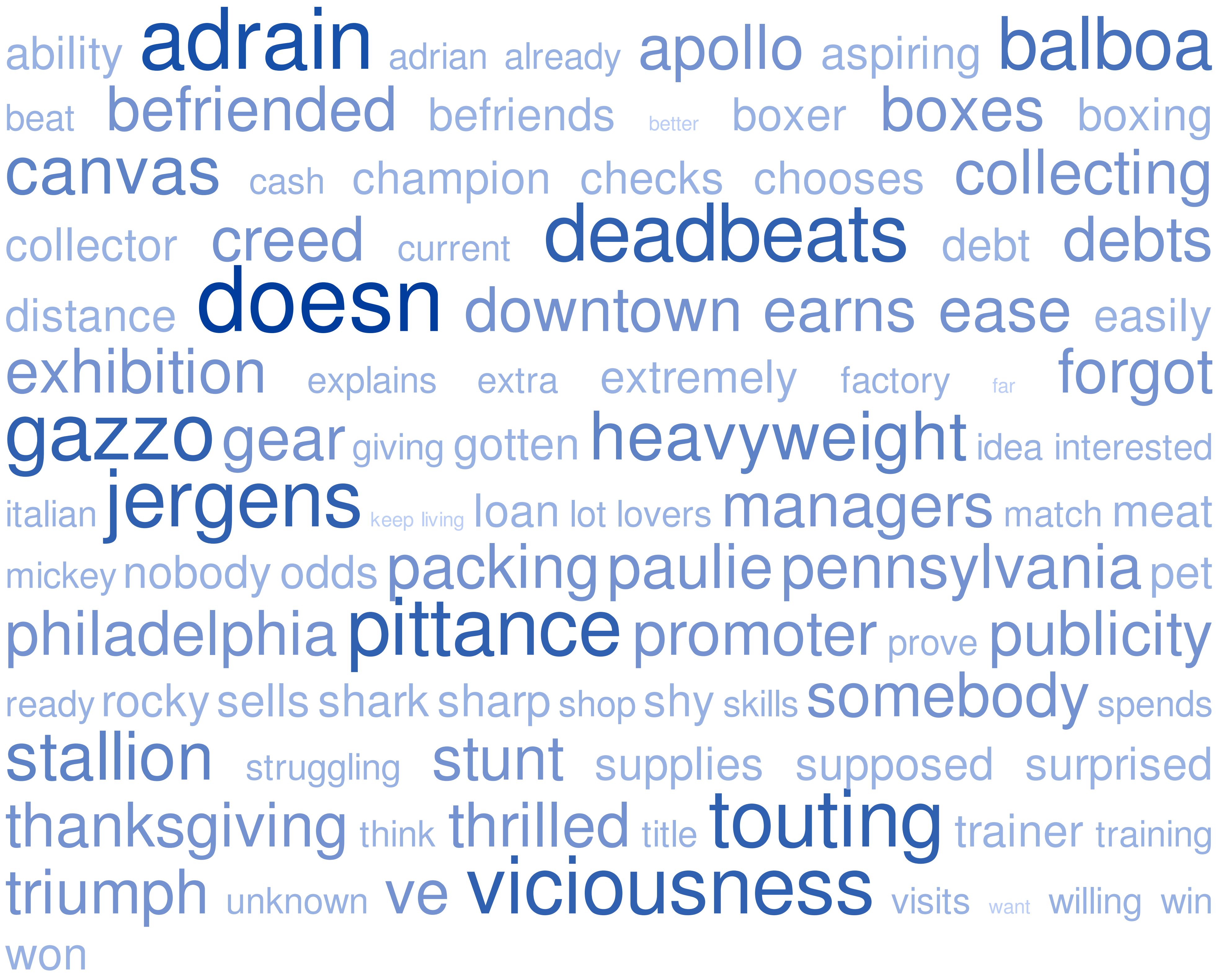


TF, IDF, or TF.IDF?



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

Vector Space TF.IDF Representation

	a	*aardvark*	*abacus*	*abba*	*able*	...	*zoom*
doc_1	6.34	0	0	0	0	...	7.42
doc_2	0	0	0	0	5.63	...	3.12
::	::	::	::	::	::	...	0
doc_m	0	0	5.32	1.23	0	...	0

	a	*aardvark*	*abacus*	*abba*	*able*	...	*zoom*
query	0	6.43	0	0	2.34	...	1.23

- Queries and documents are represented as vectors of TF.IDF weights

Queries as TF.IDF Vectors

- So far, we've talked about weighting document-terms differently
- We can also weight query-terms differently
- This is a new concept!
- Assumption: not all query-terms are equally important

Queries as TF.IDF Vectors

- **TF** usually equals 1
- Queries are small, so usually a query-term only appears once in the query
- **IDF** is computed using the collection statistics (just as it is for documents)

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

- This means that query-terms that occur in fewer documents receive 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	?	casting	?	ny	?
wizard	?	of	?	oz	?
sam	?	jones	?	iii	?
film	?	technical	?	advisors	?
edie	?	sands	?	singer	?
high	?	fidelity	?	quotes	?
quotes	?	about	?	brides	?
title	?	wave	?	pics	?
saw	?	3	?	trailers	?
the	?	rainmaker	?	movie	?
nancy	?	and	?	sluggo	?
audrey	?	rose	?	movie	?
mark	?	sway	?	photo	?
piece	?	of	?	cheese	?
date	?	movie	?	cast	?

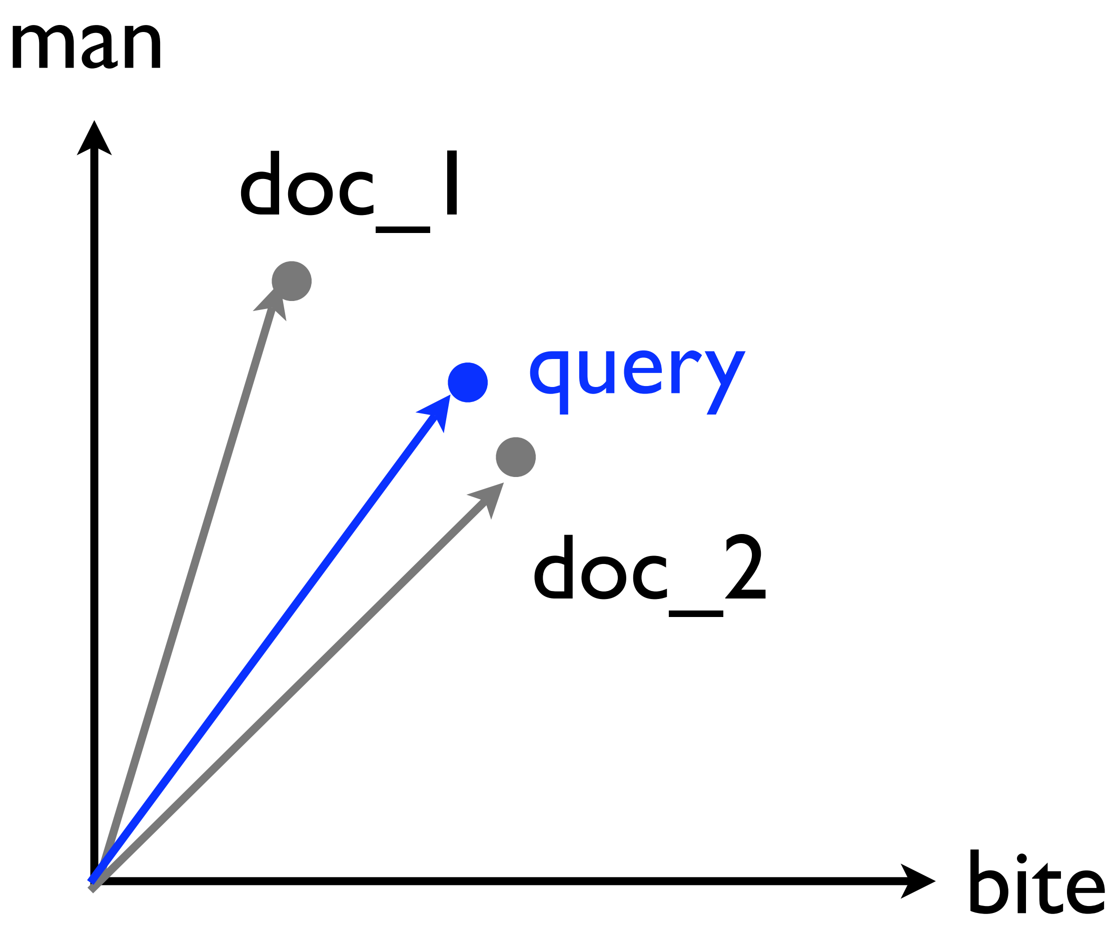
Queries as TF.IDF Vectors

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

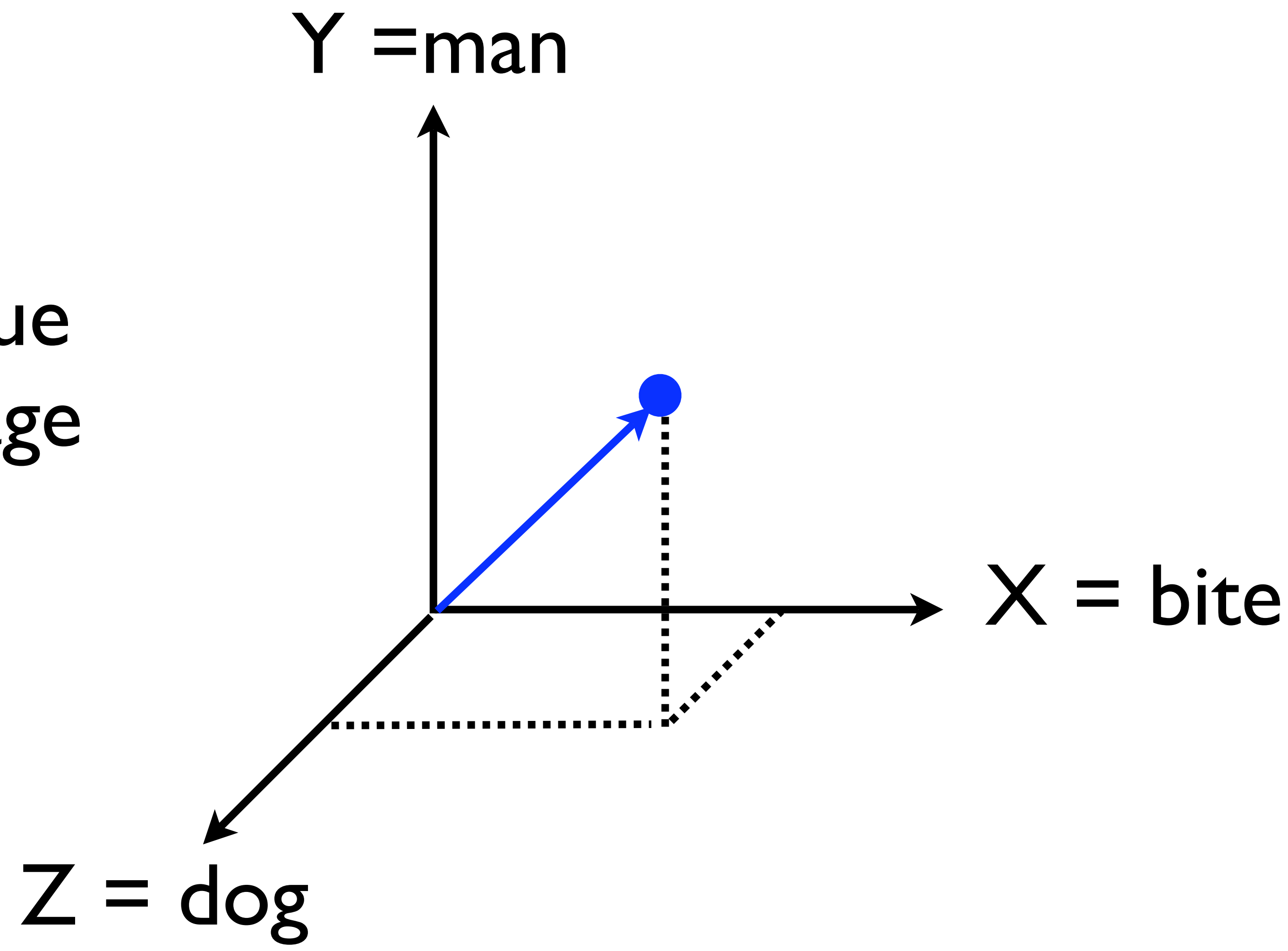
- Given a query, the vector space model ranks documents based on the cosine angle between the query and each document



Independence Assumption

- The basis vectors (X, Y, Z) are linearly independent 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

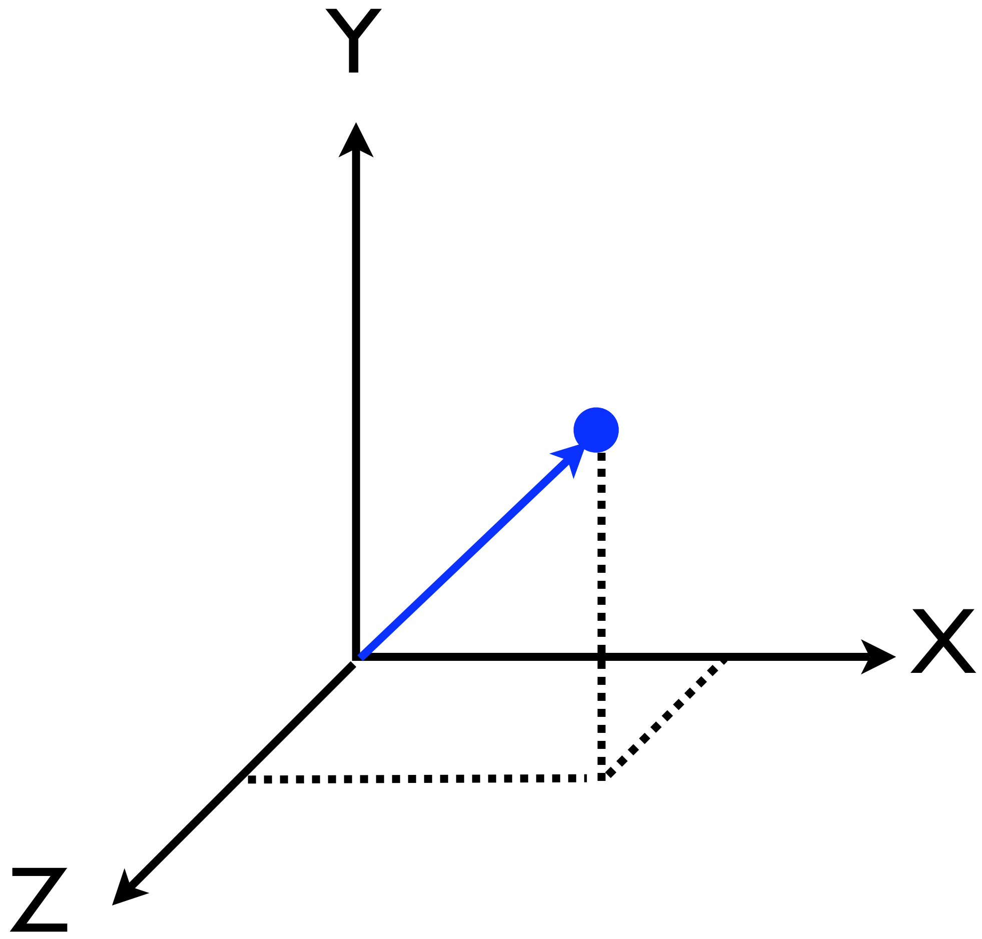
IMDB Corpus

- 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	?

Independence Assumption

- The vector space model assumes that terms are independent
- This is viewed as a limitation
- However, the implications of this limitation are still debated
- A very popular solution



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