How does text become data?

Rob Speer & Catherine Havasi

Luminoso / ConceptNet

havasi@luminoso.com

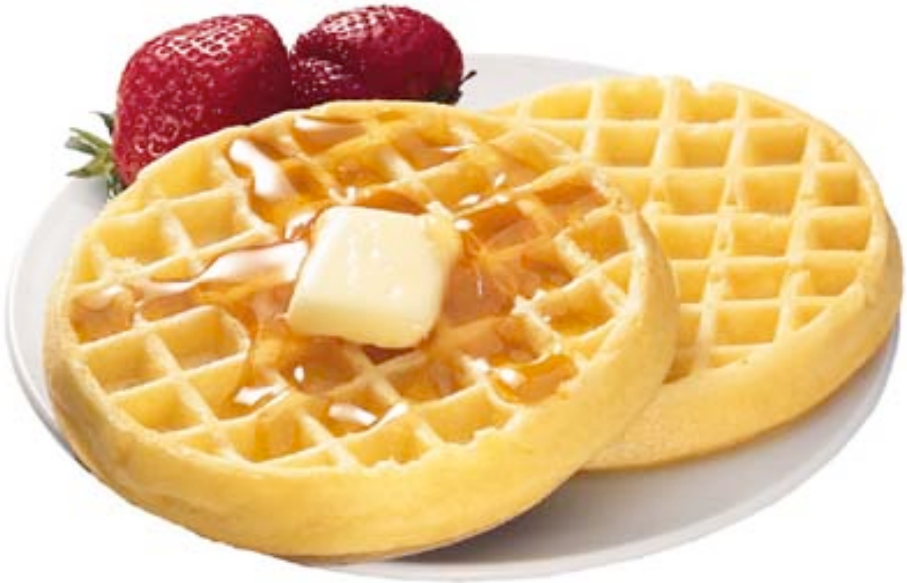
Wouldn’t it be cool if we could communicate with a computer?

r bæði í fa
nilega ekki upp við velgeng
ekki. Margrét stundar nám við
u af frítíma sínum í fótboltaæfi
sé svona heillandi við fótbolta
r finnst svo gaman að spila fótb
ð fer auðvitað mikill tími í æfin
vini mína. . . .' Margrét sér fr
um. Hana langar að fara til

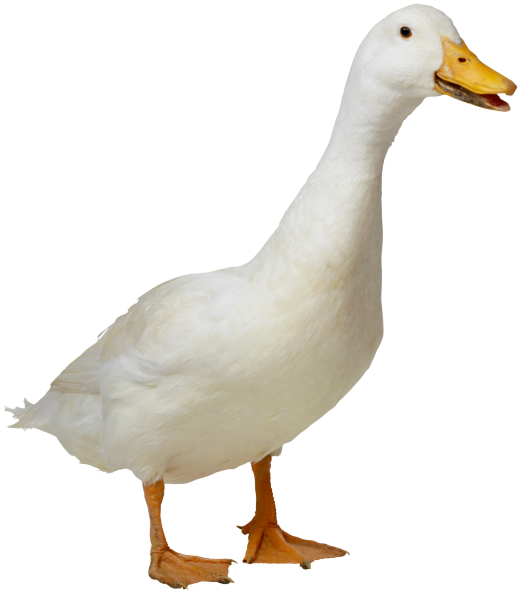
Language is ambiguous

(and you can’t really fix it)

“British left waffles on Falkland Islands”



“I made her duck”



“I shot an elephant in my pajamas”

A multi-lingual world

You can still get things done.

Classification

Similarity between documents

Search

In this talk

- A tour of useful data-driven NLP techniques
- Starting with the simple things you can do right now

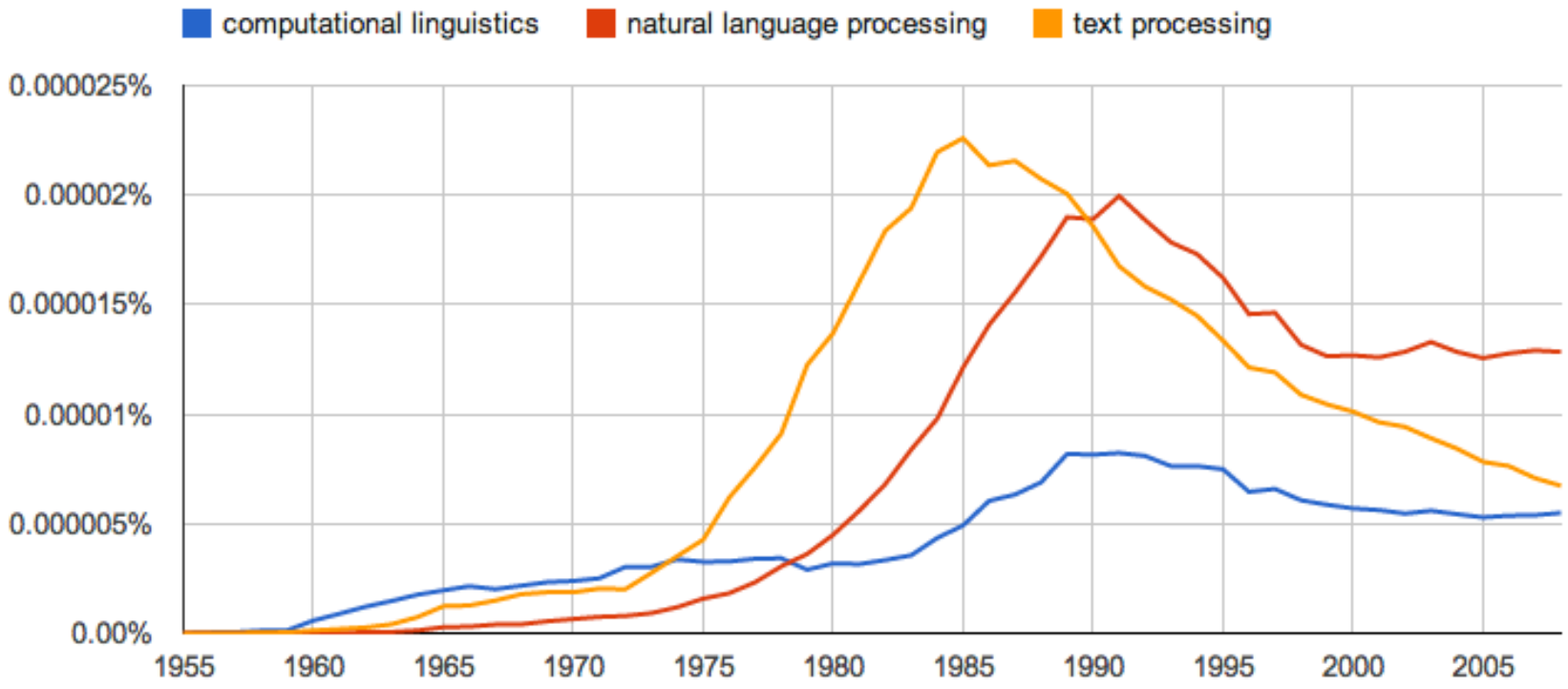
Here's the code for this talk:

http://github.com/rspeer/text-as-data

Python example: word splitting and normalizing

Simple word counts

N-gram models



Documents are “bags of words”

	woe	betray	revenge	death	alas
Julius Caesar	2	1	4	31	9
Hamlet	9	0	12	37	9
Macbeth	2	2	3	21	4

Which N-grams are interesting?

"**vice president**"

is more interesting than

"**the vice**"

which is more interesting than

"**of the**"

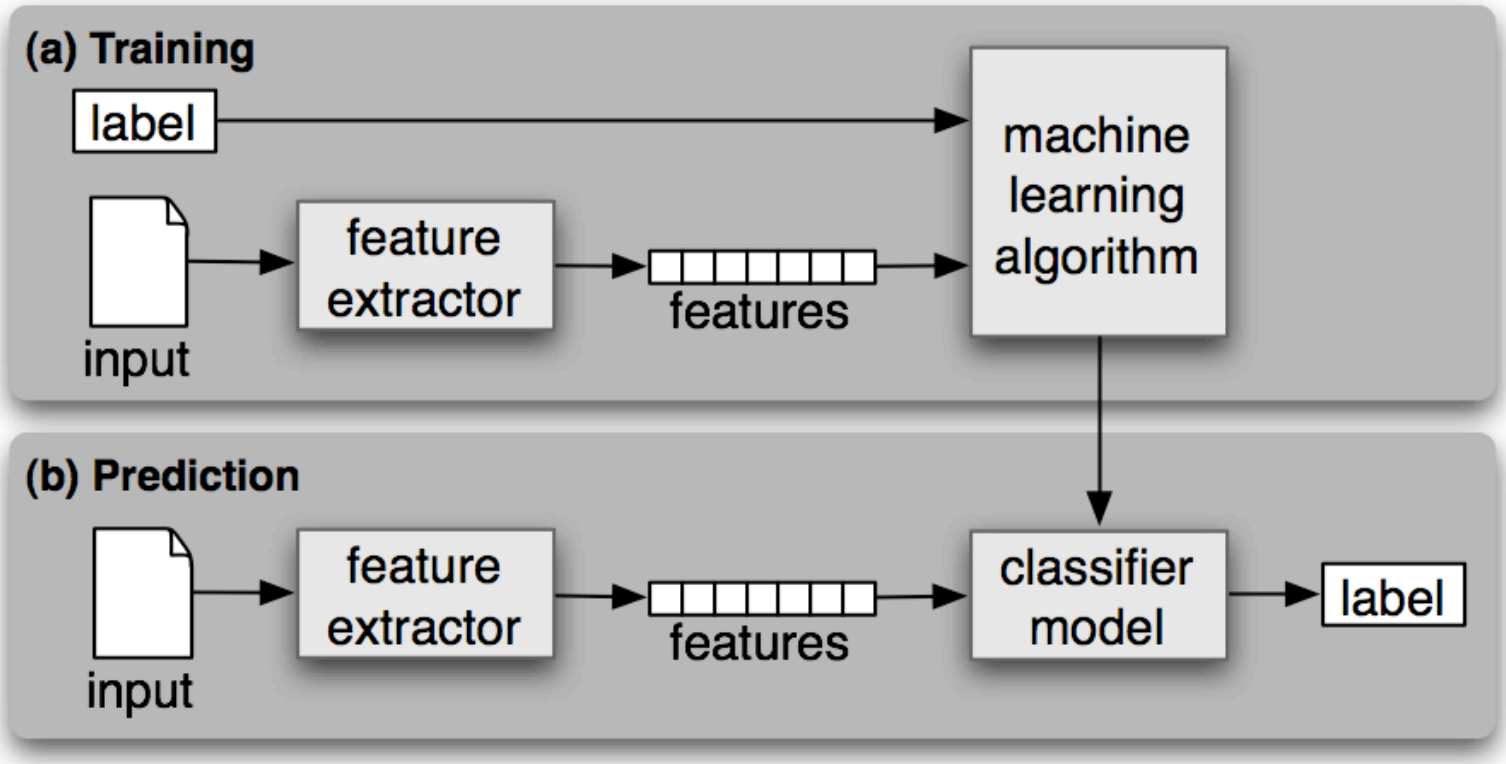
Which N-grams are interesting?

Consider this contingency table:

p(**vice**, **president**)	*p*(**vice**, ~president)
p(~vice, **president**)	*p*(~vice, ~president)

Python example: interesting N-grams

Text classification

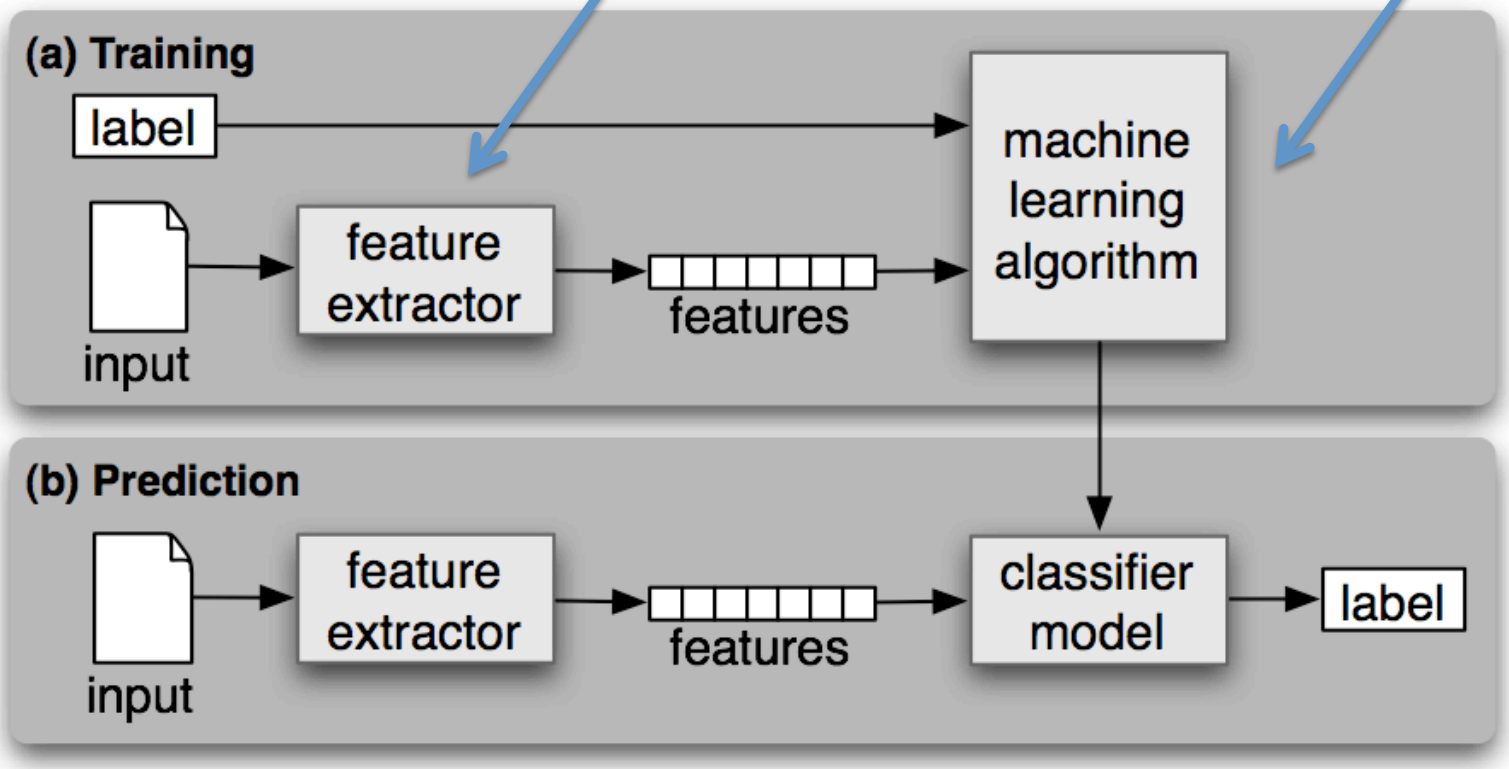


from “Natural Language Processing with Python”,
by Steven Bird, Ewan Klein, and Edward Loper (O’Reilly, 2009)

Text classification

probably
bag-of-words

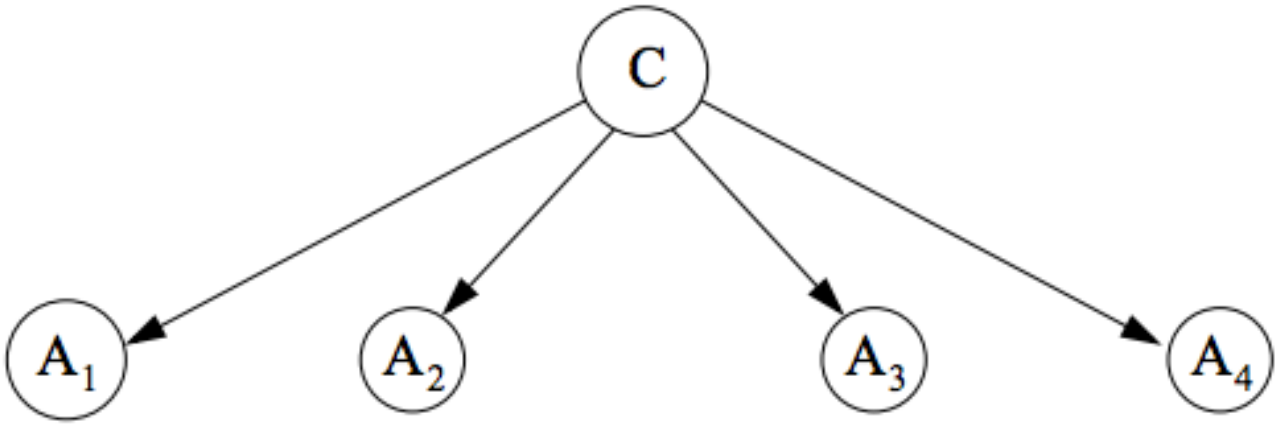
possibly
Naive Bayes



from “Natural Language Processing with Python”,
by Steven Bird, Ewan Klein, and Edward Loper (O’Reilly, 2009)

Overview of Naïve Bayes classification

- The probability that a document is in class *C* depends on its features, A_n
- Assume all features are statistically independent



Python example: Classification with NLTK and scikit-learn

Text similarity

- Bags of words can tell us how similar documents are

	woe	betray	revenge	death	alas
Julius Caesar	2	1	4	31	9
Hamlet	9	0	12	37	9
Macbeth	2	2	3	21	4

Text similarity

- Bags of words can tell us how similar documents are

	woe	betray	revenge	death	alas
Julius Caesar	2	1	4	31	9
Hamlet	9	0	12	37	9
Macbeth	2	2	3	21	4
Alice in Wonderland	0	0	0	1	4

TF-IDF normalization

- Some documents are longer than others
- Some words appear more than others

	woe	betray	revenge	death	alas
Julius Caesar	.046	.018	.139	0	.159
Hamlet	.142	0	.287	0	.110
Macbeth	.053	.041	.120	0	.082
Alice in Wonderland	0	0	0	0	.054

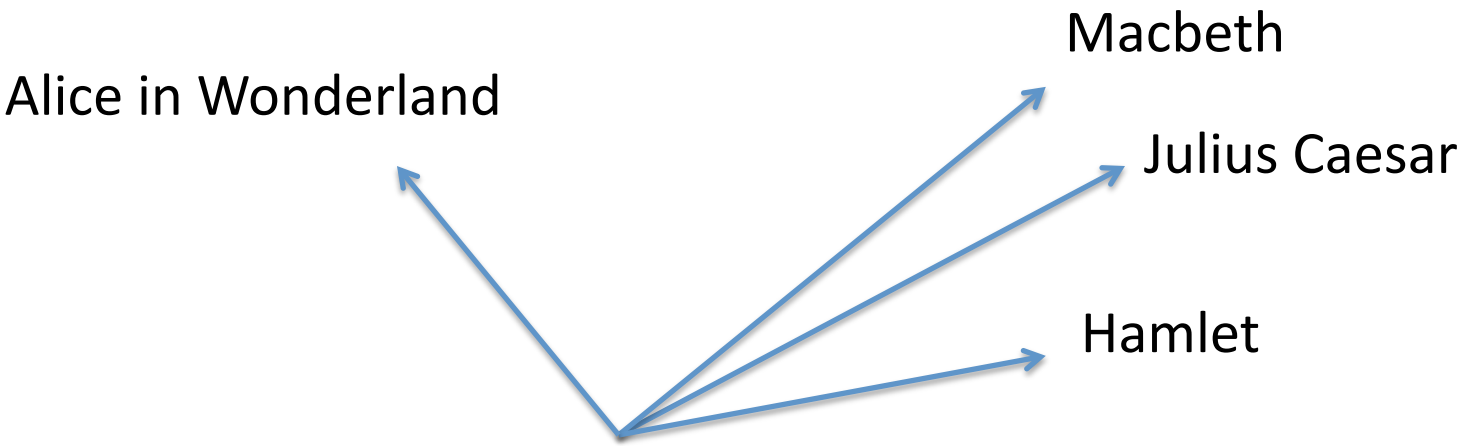
(TF-IDF values × 1000, from NLTK's Project Gutenberg)

TF-IDF normalization

- TF replaces term counts with term frequencies
- IDF tells us how much information we get when a word appears
- In NLTK's Project Gutenberg corpus:
  - IDF(the) = 0 bits
  - IDF(death) = 0 bits
  - IDF(taxes) = 2.58 bits
  - IDF(whale) = 2.17 bits
  - IDF(Ishmael) = 3.17 bits

Vector-space similarity

- Similar texts have a small angle between them

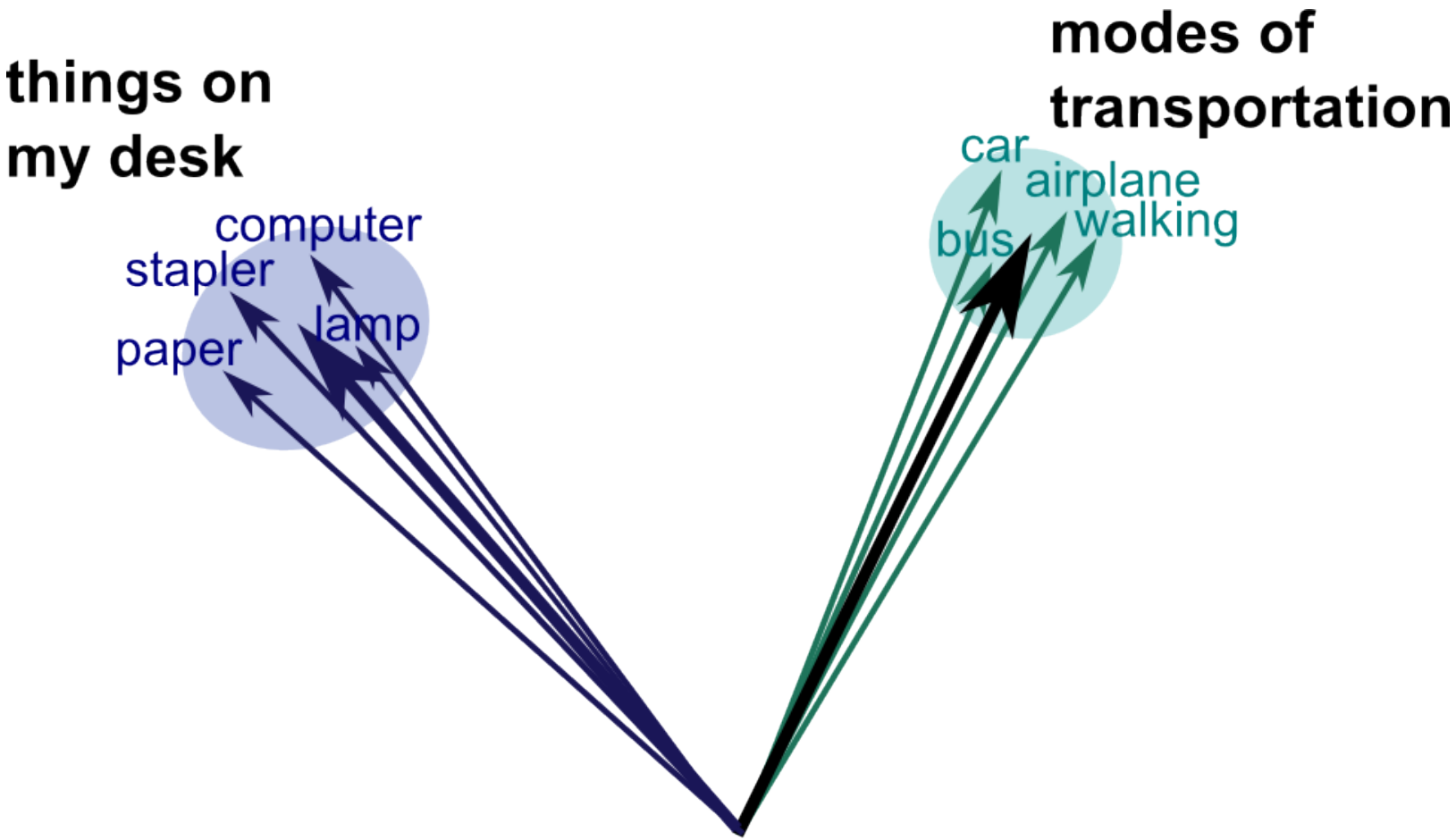


Dimensionality reduction

- Put terms and documents in a lower-dimensional space where we can easily compare them

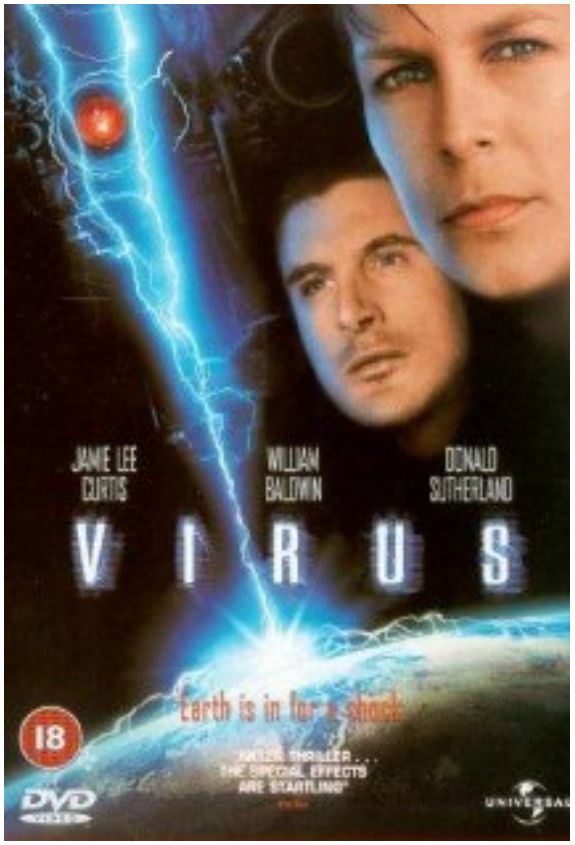
- In NLP, this is called Latent Semantic Analysis or Latent Semantic Inference

Vector spaces group words and documents by meaning

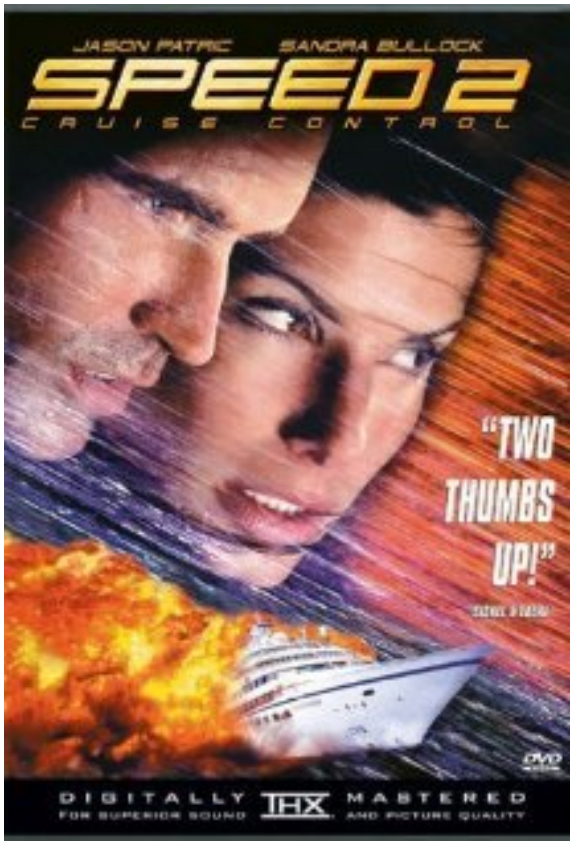


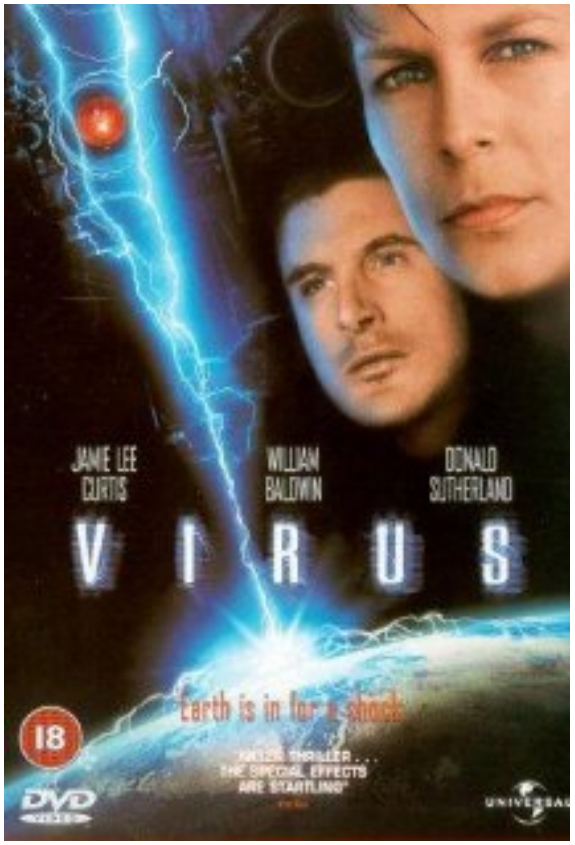
Python example: Unsupervised text similarity using gensim

Similarity of movie reviews

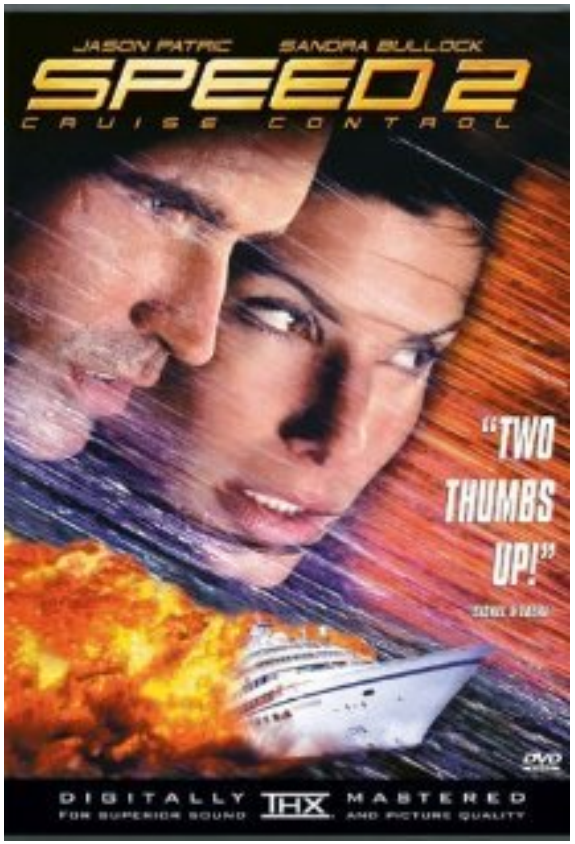


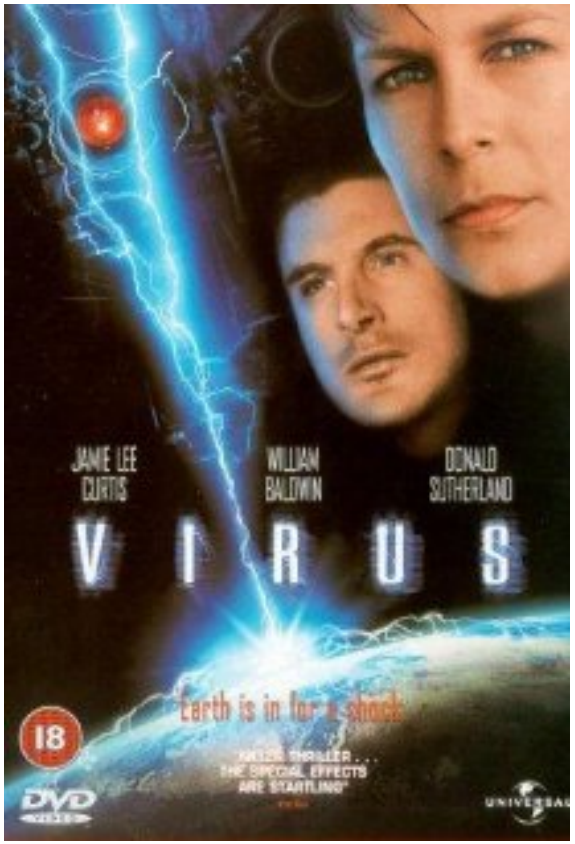
Similarity of movie reviews

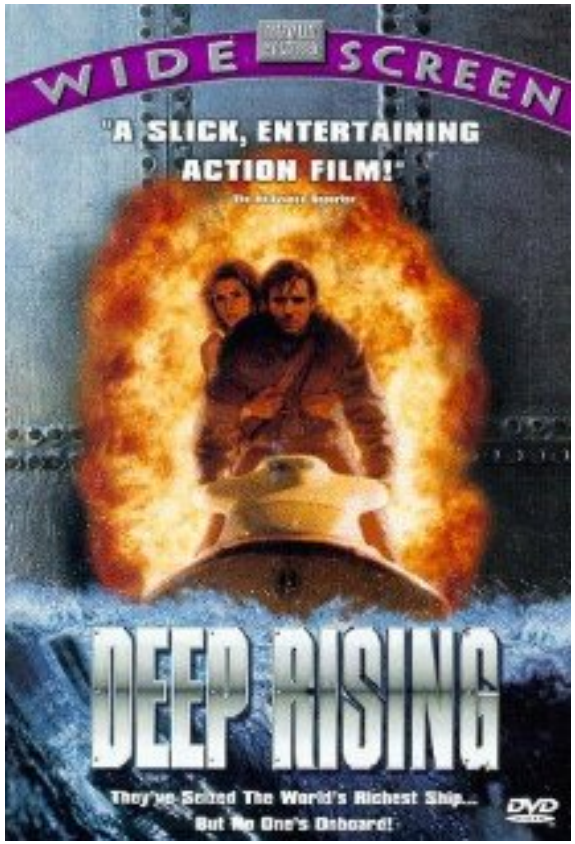




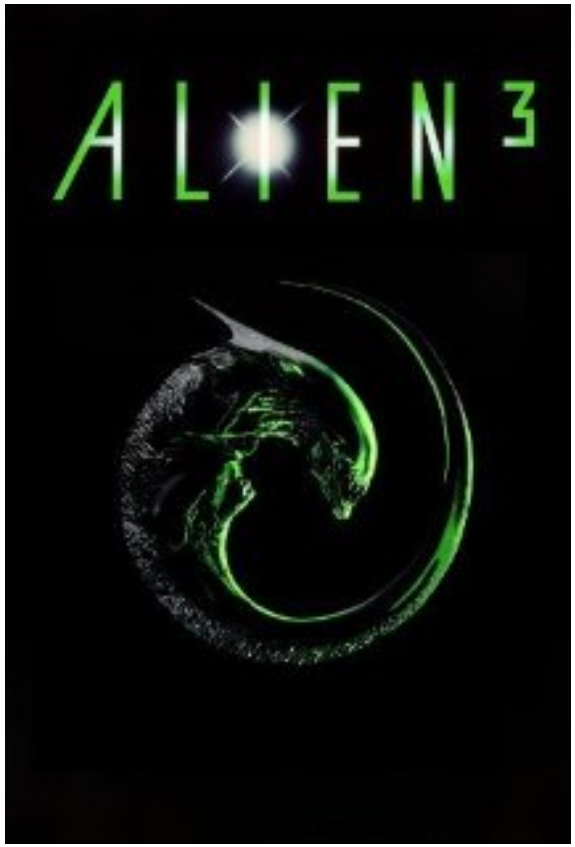
Similarity of movie reviews







Is this all there is to word meaning?



Modeling word meanings

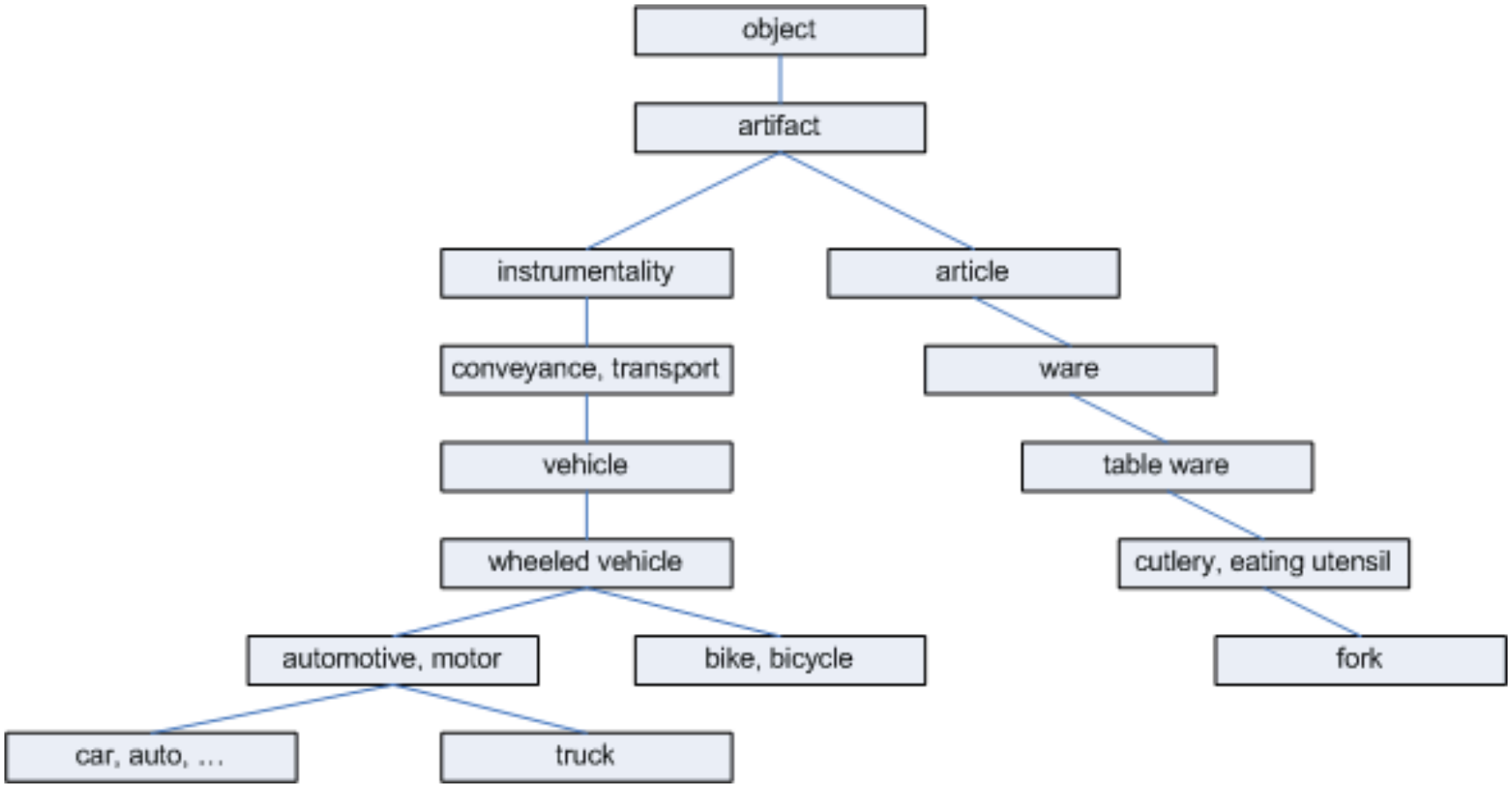
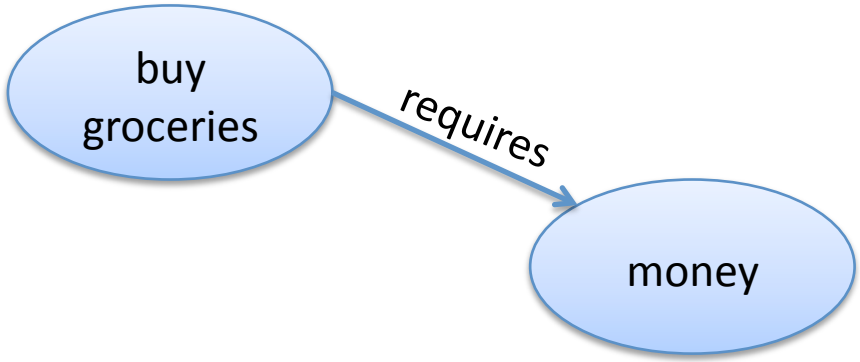
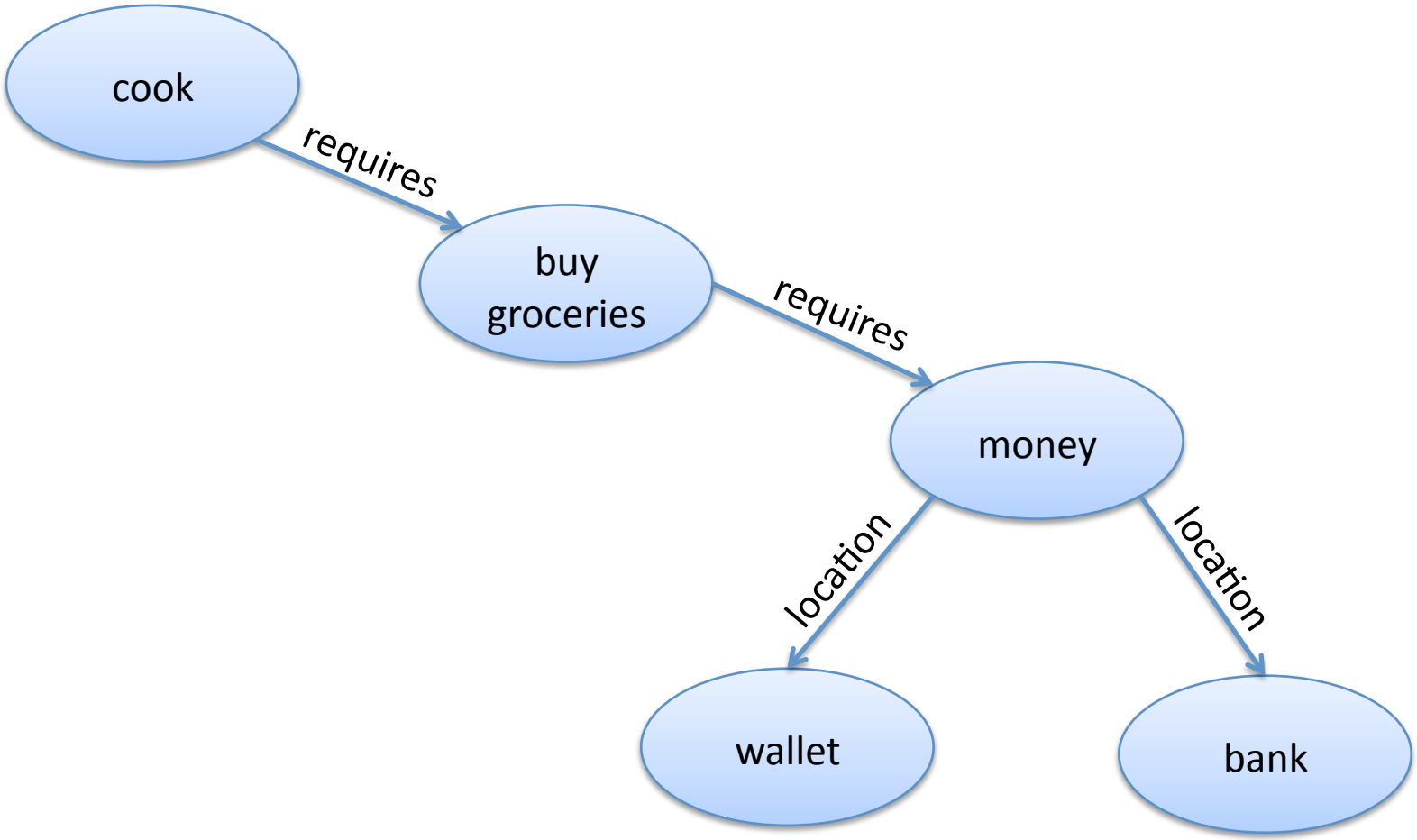
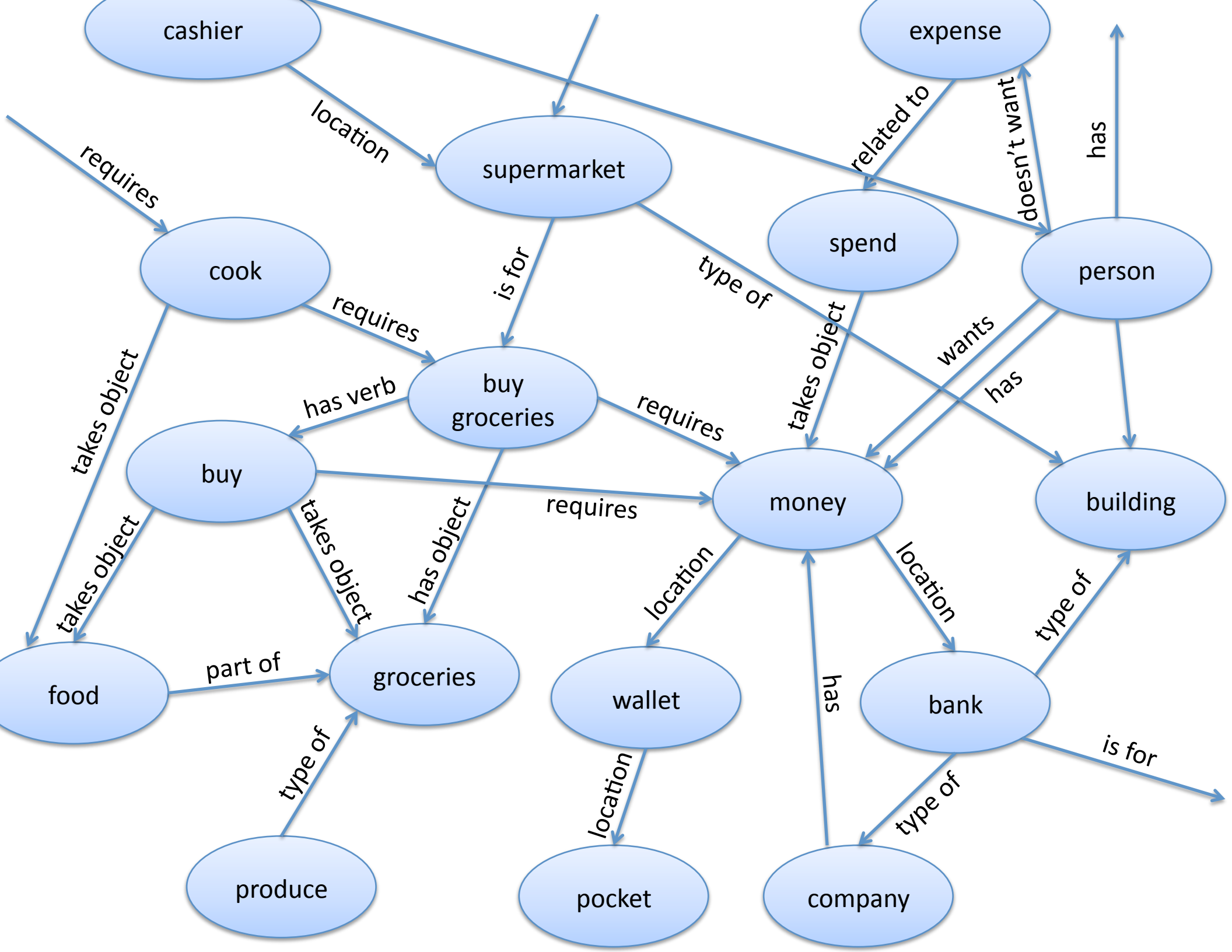


Image source: “WordNet-based semantic similarity measurement”
by Troy Simpson and Thanh Dao, on codeproject.com

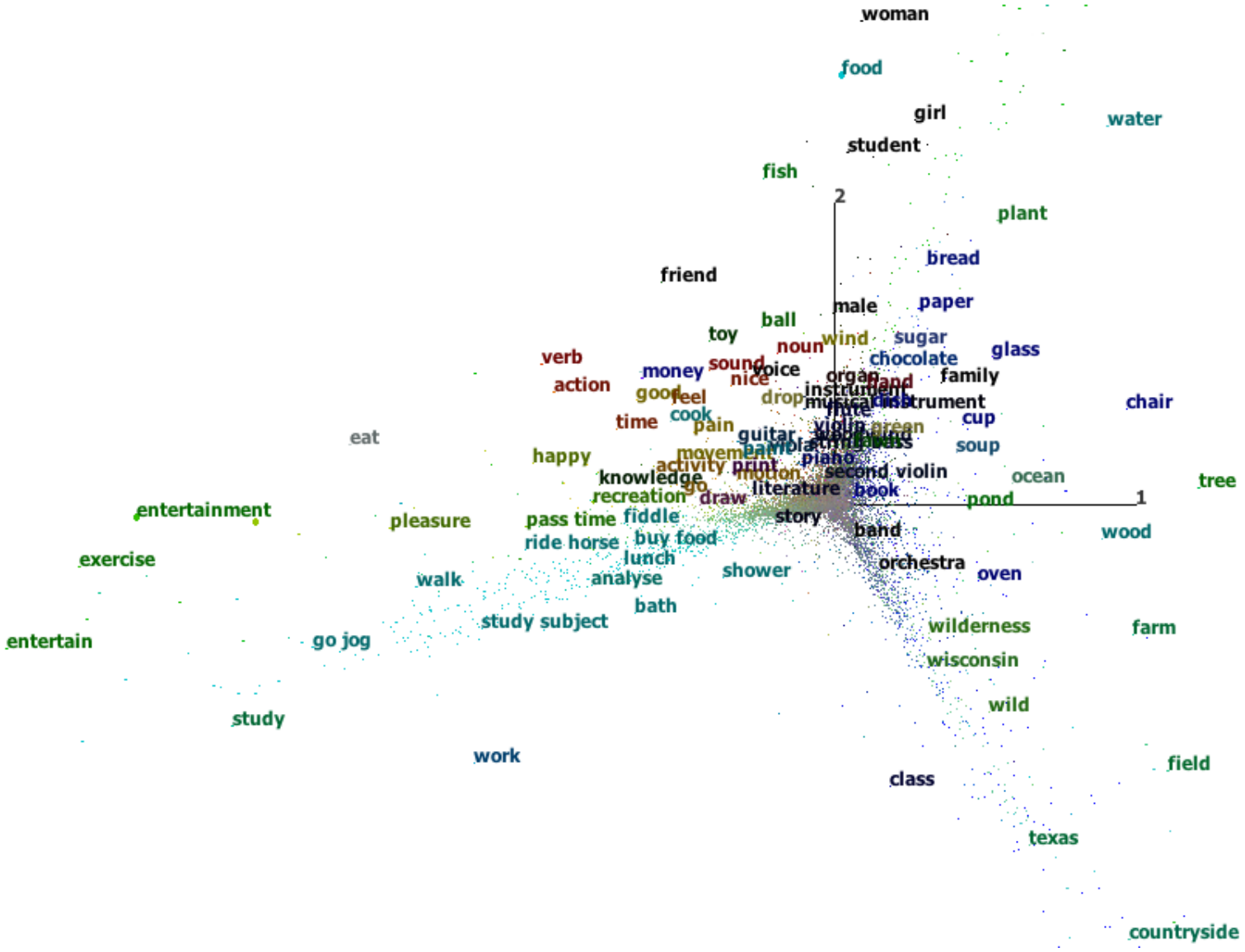

I bought
groceries at the
store.


buy
groceries
requires
money


cook
requires
buy groceries
requires
money
location
location
wallet
bank


cashier
expense
location
related to
doesn't want
has
requires
supermarket
spend
person
cook
is for
type of
requires
takes object
wants
has verb
buy groceries
has
takes object
requires
buy
money
building
requires
takes object
takes object
has object
location
location
type of
food
part of
groceries
wallet
has
bank
type of
is for
location
type of
produce
pocket
company

ConceptNet as a vector space



Python example: Querying ConceptNet

- See API documentation linked from:

  http://conceptnet5.media.mit.edu

Many incompatible representations

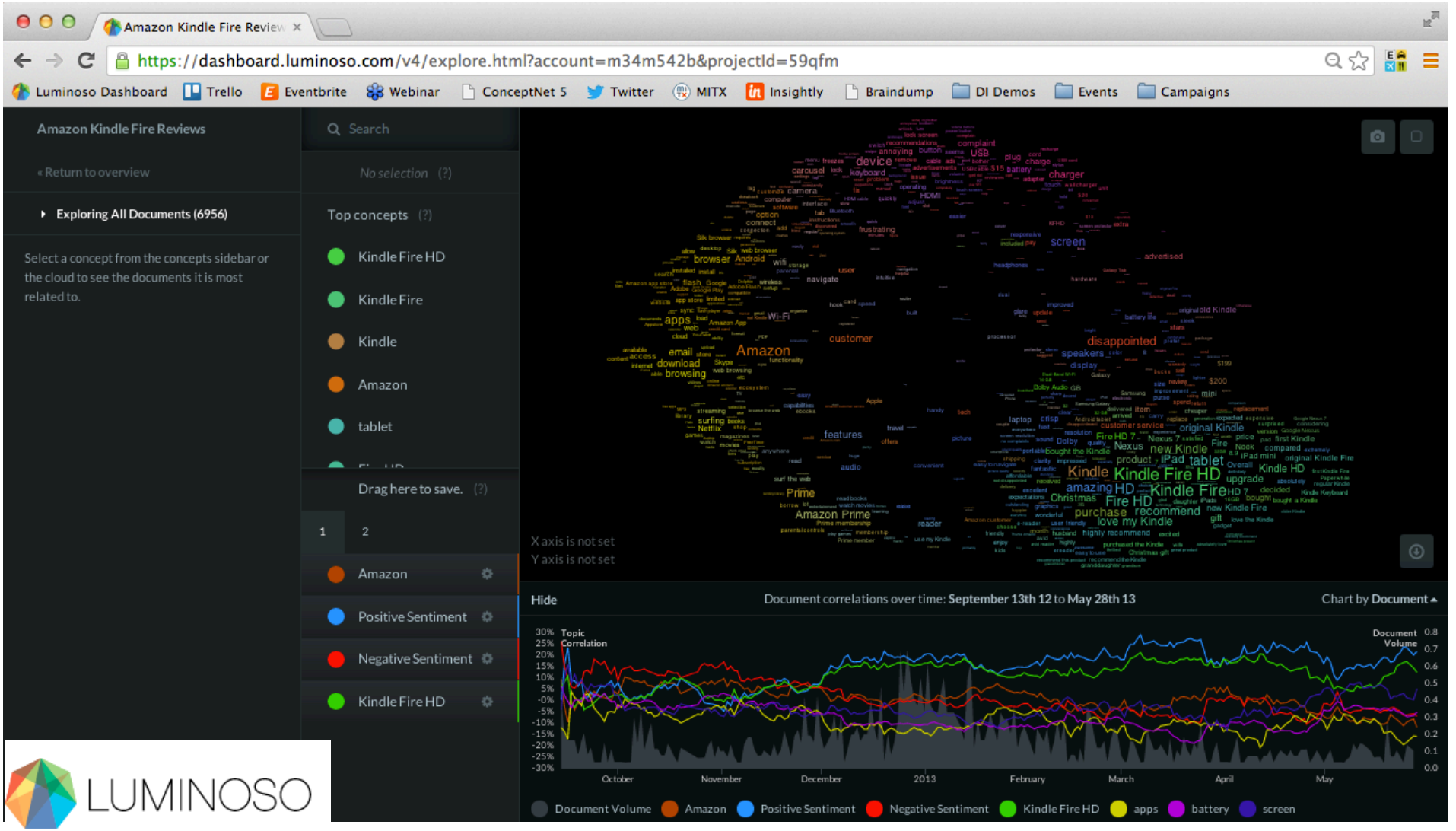
- Supervised text classification
- Unsupervised document similarity
- Domain-general word associations

Many incompatible representations

- Supervised text classification
- Unsupervised document similarity
- Domain-general word associations

- It would be nice if one model could do all of these.

We made Luminoso



We want you to get more information out of text.

We want everyone to be able to play.
Help us alpha or beta test.
(We'll give you free access.)

Email us: hub@luminoso.com

That's all

Code and slides:

http://github.com/rspeer/text-as-data

Cool things we work on:

http://conceptnet5.media.mit.edu

http://luminoso.com

Email us: havasi@luminoso.com

Extra slides

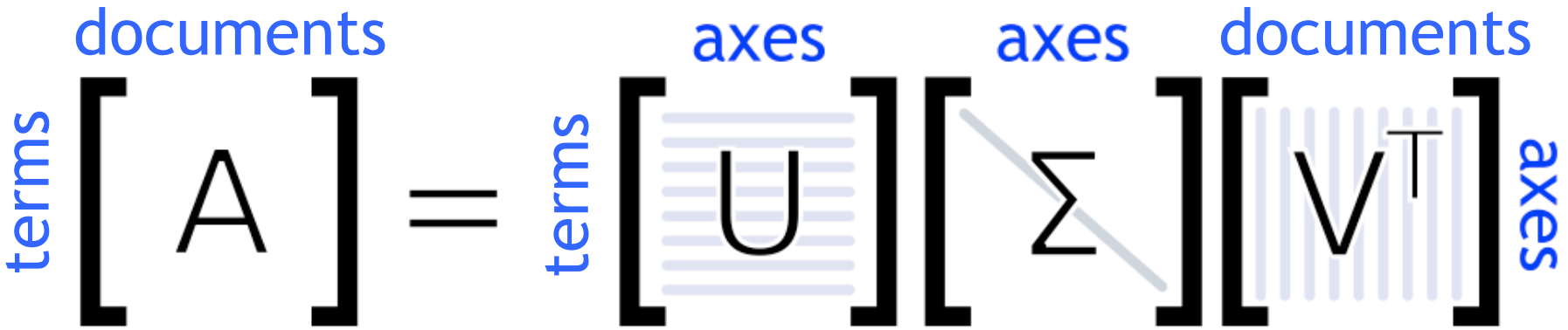
Python example: Querying WordNet

But Naïve Bayes is so naïve!

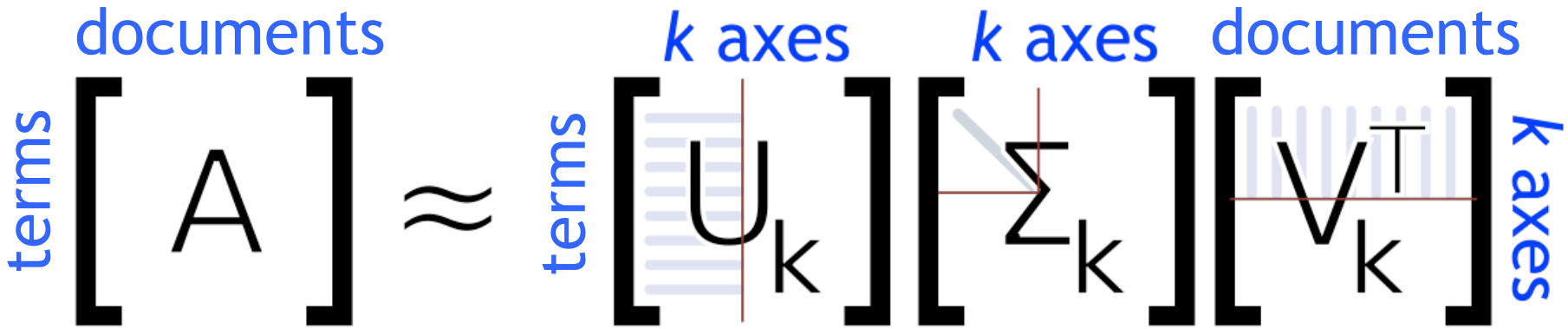
- Sure, its fundamental assumption is wrong
- Often, it works anyway
- On NLP tasks, NB is blazingly fast and surprisingly effective

(See “The Optimality of Naive Bayes”, Harry Zhang, AAAI 2004)

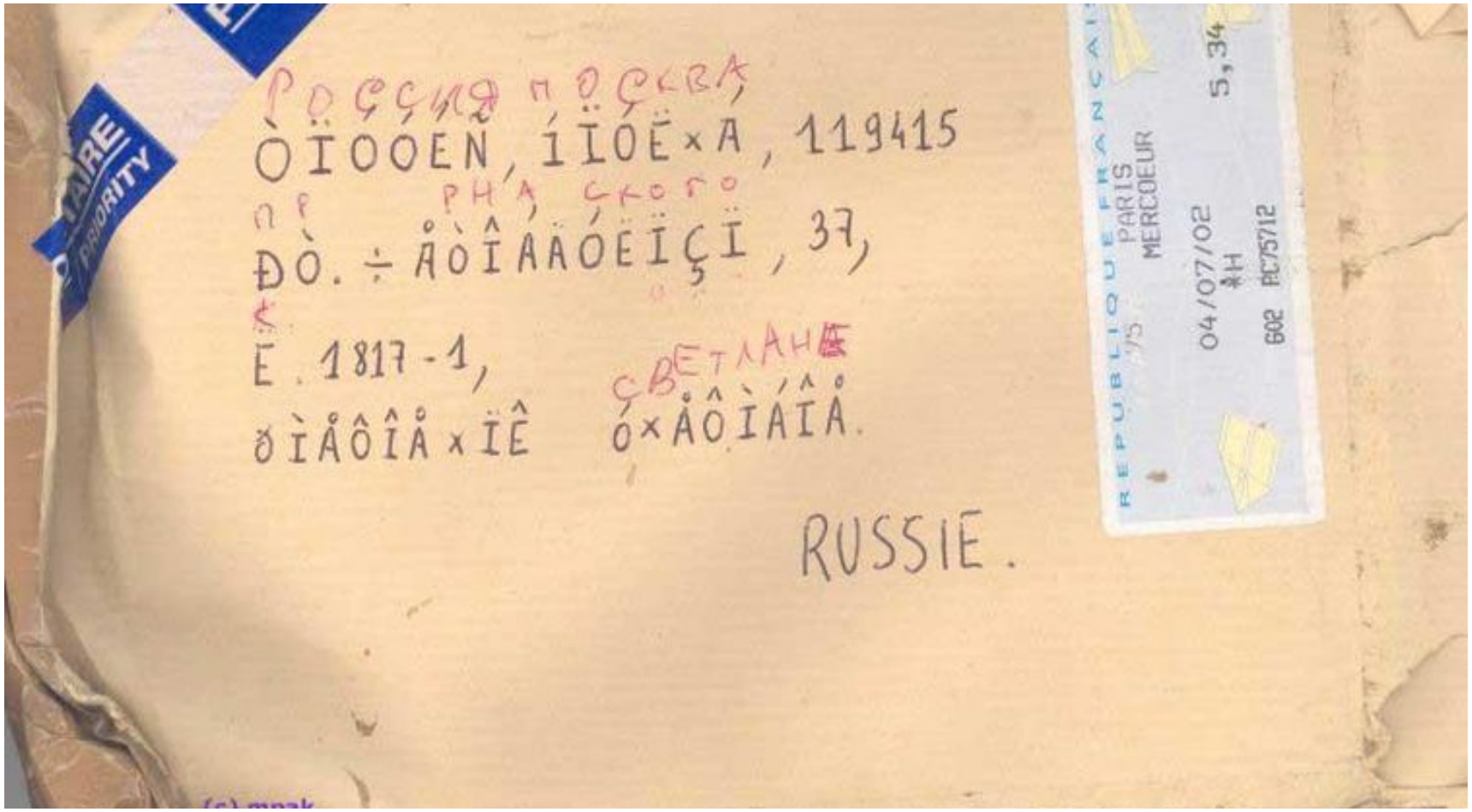
Dimensionality reduction



Dimensionality reduction



A multi-lingual world



“Much Debate”