ISE 540: Text Analytics

Mayank Kejriwal

Research Assistant Professor/Research Lead

Department of Industrial and Systems Engineering

Information Sciences Institute

USC Viterbi School of Engineering

kejriwal@isi.edu

Let’s return to getting the vectors in the first place

- One popular model is ‘bag of words’, commonly held to be the ‘tf-idf’ model
  - Tf-idf stands for term frequency-inverse document frequency
  - The tf term encodes the intuition that the more times a word occurs in a document, the more important it is to the document
  - The idf term encodes the intuition that, all else equal, a word that occurs more often throughout the corpus is less important to each document that it occurs in, than another word that is more rare (‘the’ vs. ‘apple’)
  - Tf is ‘local’ (need only a document), idf is ‘global’ (we need the full corpus to compute idf)…where can this go wrong?
- A bag is a ‘multi-set’; to compute ‘tf’ we need to know how many times the word occurs in the document!

Tf-idf

Number of occurrences of t in d (okay to take the log as well but be consistent!)

$$\text{idf}_t = \log \frac{N}{\text{df}_t}.$$

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t.$$

- 't’ is a term, ‘d’ is a document
- For every document ‘d’, we can compute the tf-idf score of every term in it; this gives us a ‘tf-idf’ vector for the document!
- Now we can use cosine similarity, just as before (again, use query as just another document)
- **What problems do you foresee?**

Tf-idf example

Tf-idf

Number of occurrences of t in d

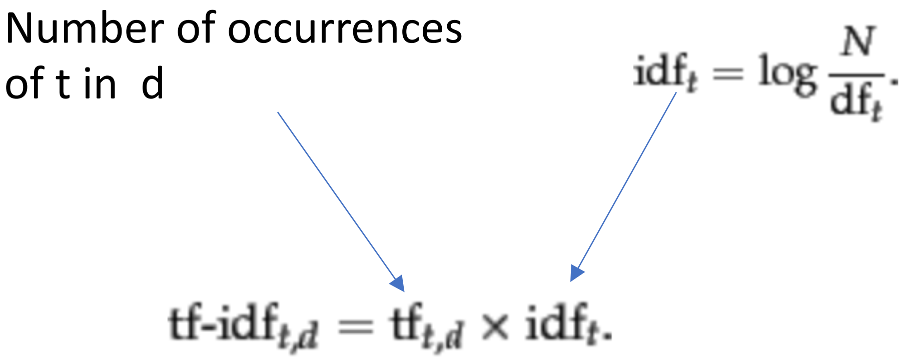
$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$$

$$\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t.$$

- d1: The man jumped over the moon and missed the sun
- d2: The cow jumped over the sun and missed the moon
- d3: The man jumped over the cow and touched the sky
- d4: The moon jumped over the sky

What is the vocabulary?

Tf-idf



Number of occurrences of t in d

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$$

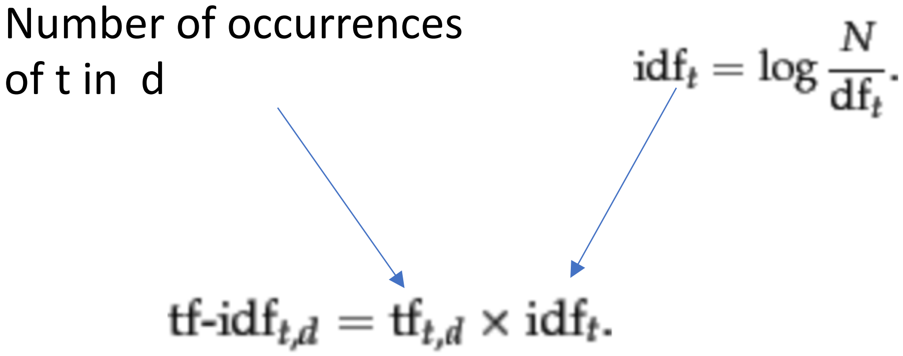
$$\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t.$$

- d1: The man jumped over the moon and missed the sun
- d2: The cow jumped over the sun and missed the moon
- d3: The man jumped over the cow and touched the sky
- d4: The moon jumped over the sky

Vocabulary, N=4

The
Man
Jumped
Over
Moon
And
missed
Sun
Cow
Sky
touched

Tf-idf



Number of occurrences of t in d

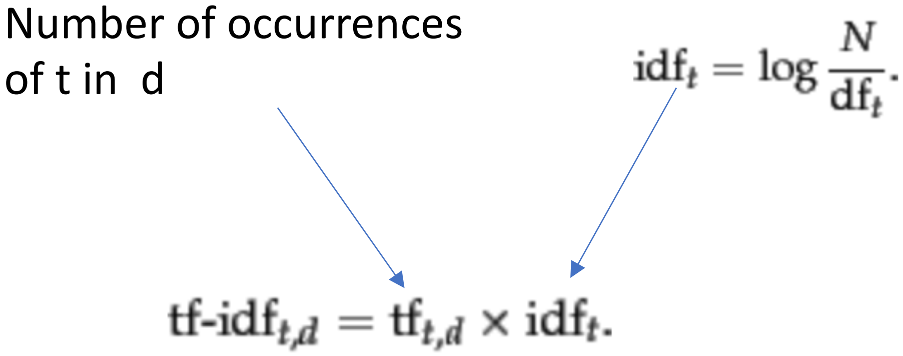
$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$$

$$\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t.$$

- d1: The man jumped over the moon and missed the sun
- d2: The cow jumped over the sun and missed the moon
- d3: The man jumped over the cow and touched the sky
- d4: The moon jumped over the sky

	IDF
The	Log(4/4)
Man	Log(4/2)
Jumped	Log(4/4)
Over	…
Moon	
And	
missed	
Sun	
Cow	
Sky	
touched	

Tf-idf



Number of occurrences of t in d

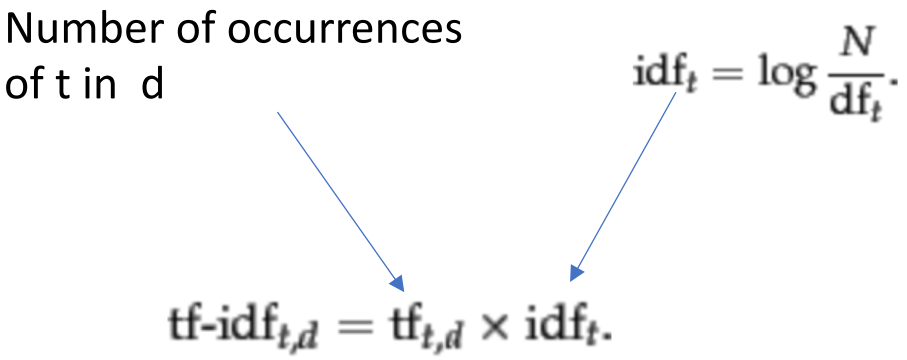
$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$$

$$\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t.$$

- d1: The man jumped over the moon and missed the sun
- d2: The cow jumped over the sun and missed the moon
- d3: The man jumped over the cow and touched the sky
- d4: The moon jumped over the sky

	Tf_d1
The	3
Man	1
Jumped	1
Over	…
Moon	
And	
missed	
Sun	
Cow	
Sky	
touched	

Tf-idf



Number of occurrences of t in d

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$$

$$\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t.$$

- d1: The man jumped over the moon and missed the sun
- d2: The cow jumped over the sun and missed the moon
- d3: The man jumped over the cow and touched the sky
- d4: The moon jumped over the sky

Tf_d1_idf

The	3 X Log(4/4)
Man	1 X Log(4/2)
Jumped	1 X Log(4/4)
Over	1 X Log(4/4)
Moon	1 X Log(4/3)
And	1 X Log(4/3)
missed	1 X Log(4/2)
Sun	1 X Log(4/2)
Cow	0
Sky	0
touched	0

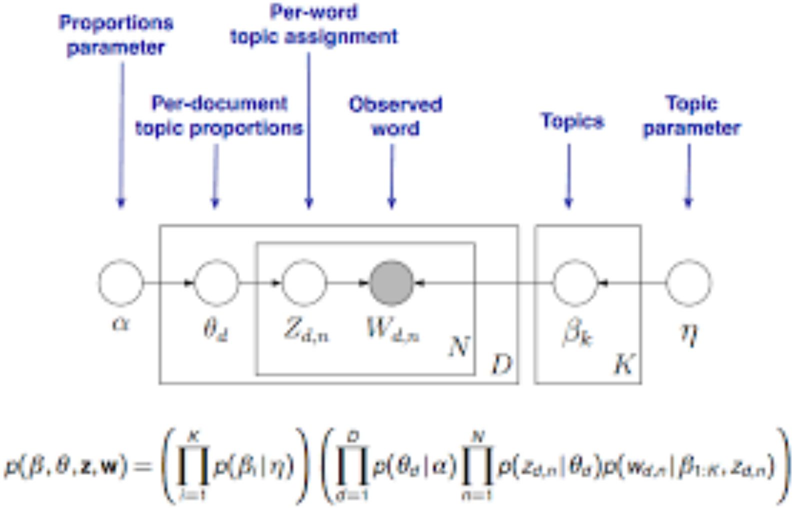
You don’t need to do all this yourself!

- But crucial that you understand ‘what’ this is, what the package is doing, the details (e.g., are we taking log of both tf and idf?)
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

(Advanced) Even graphical models can be used for getting document vectors

- ‘Topic model’ (also called LDA or latent Dirichlet allocation)
- Even if you don’t know how it works, you’ve probably seen it visualized as ‘word clouds’

"kingdom" "king" "royal" "english" "scotland" "queen" "england" "british" "great" "london"
"movies" "direct" "star" "movie" "americans" "television" "american" "actor" "play" "actress"
"information" "system" "internet" "program" "computer" "data" "number" "software" "code" "systems"
"election" "leader" "party" "bank" "crash" "democratic" "orange" "political" "vote" "politics"
"music" "single" "pop" "record" "singer" "release" "hit" "first" "song" "album"
"club" "infobox" "association" "league" "play" "national" "player" "football" "team" "japan"



$$p(\beta,\theta,\mathbf{z},\mathbf{w}) = \left(\prod_{i=1}^{K} p(\beta_i \mid \eta)\right)\left(\prod_{d=1}^{D} p(\theta_d \mid \alpha)\prod_{n=1}^{N} p(z_{d,n} \mid \theta_d)\, p(w_{d,n} \mid \beta_{1:K}, z_{d,n})\right)$$

Resources for topic models (among other things)

- https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24
- I strongly encourage you to go through the ‘tutorial’ in this blog
  - Shows you some of the other nuts and bolts of NLP, including tokenization etc.

Distributional Semantics

…and word embeddings

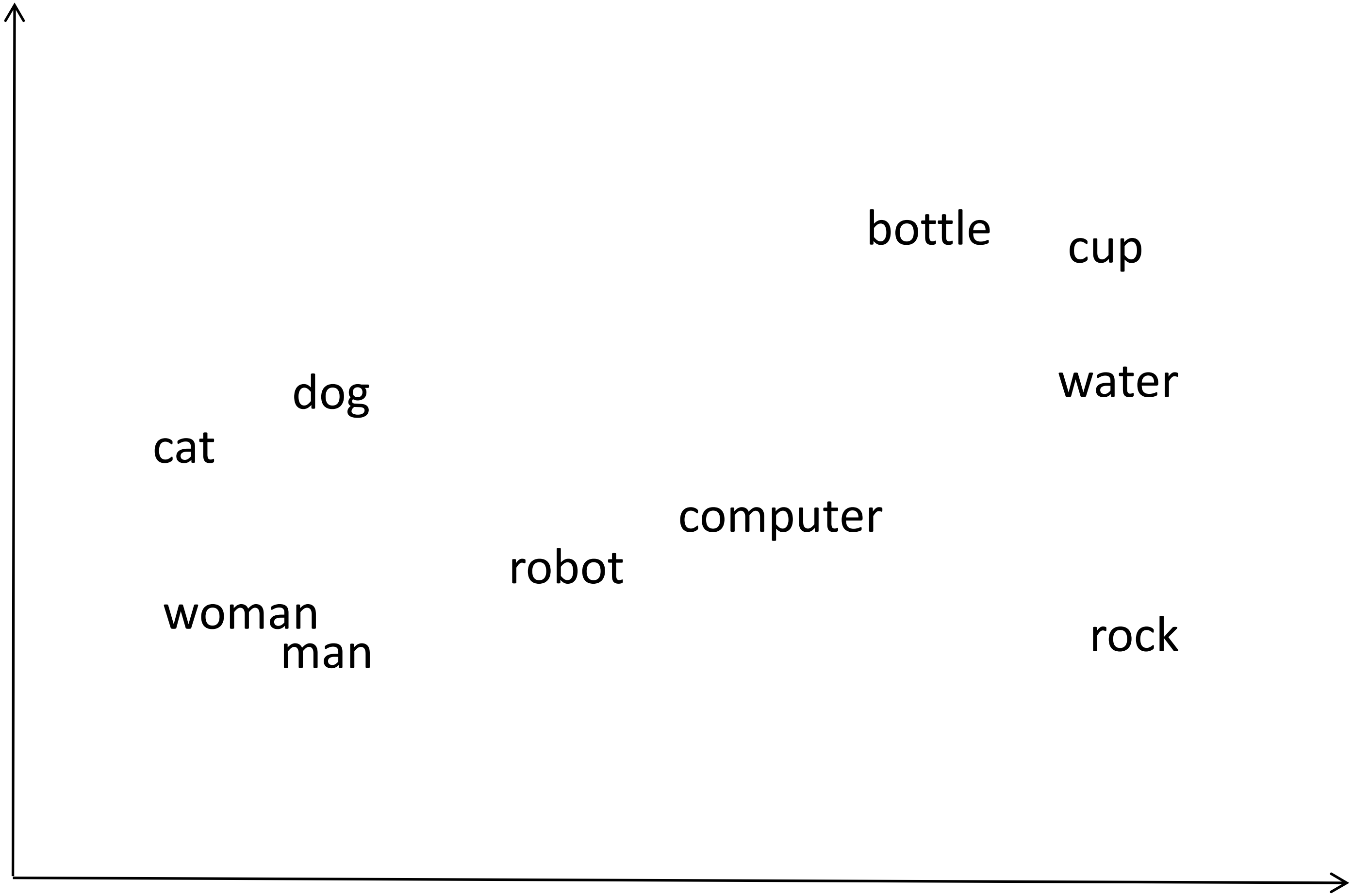
What’s in a word?

- Semantics:
  - the branch of linguistics and logic concerned with meaning. There are a number of branches and subbranches of semantics, including formal semantics, which studies the logical aspects of meaning, such as sense, reference, implication, and logical form, lexical semantics, which studies word meanings and word relations, and conceptual semantics, which studies the cognitive structure of meaning

Vector-Space (Distributional) Lexical Semantics

- Represent word meanings as points (vectors) in a (high-dimensional) Euclidian space.
- Dimensions encode aspects of the context in which the word appears (e.g. how often it co-occurs with another specific word).
    - "You will know a word by the company that it keeps." (J.R. Firth, 1957)
- Semantic similarity defined as distance between points in this semantic space.

Sample Lexical Vector Space



bottle

cup

dog

water

cat

computer

robot

woman

man

rock

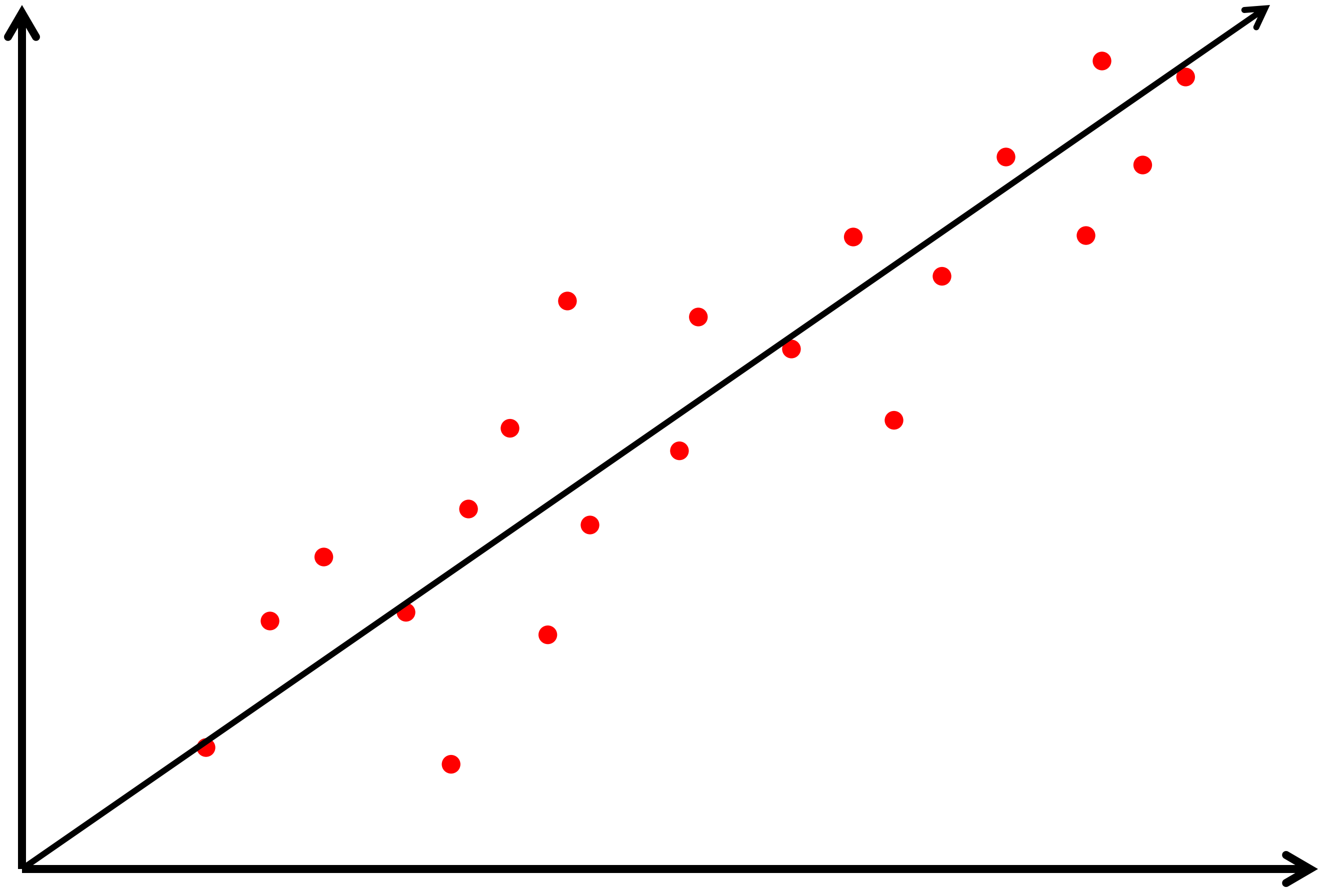
Simple Word Vectors

- For a given target word, *w*, create a bag-of-words “document” of all of the words that co-occur with the target word in a large corpus.
  - Window of *k* words on either side.
  - All words in the sentence, paragraph, or document.
- For each word, create a (tf-idf weighted) vector from the “document” for that word.

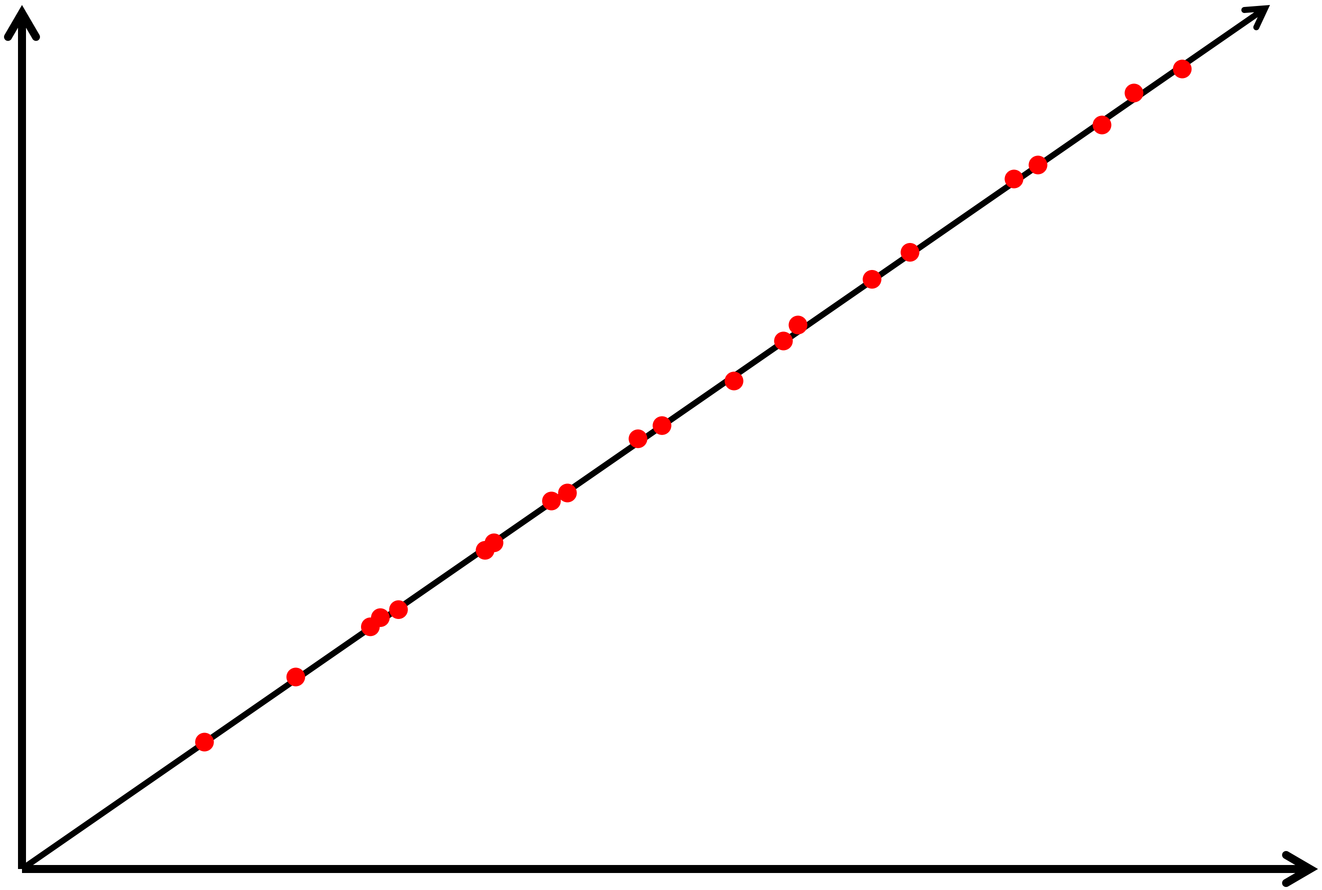
Dimensionality Reduction

- Word-based features result in extremely high-dimensional spaces that can easily result in over-fitting.
- Reduce the dimensionality of the space by using various mathematical techniques to create a smaller set of *k* new dimensions that most account for the variance in the data.
  - Singular Value Decomposition (SVD) used in Latent Semantic Analysis (LSA)
  - Principle Component Analysis (PCA)

Sample Dimensionality Reduction



Sample Dimensionality Reduction

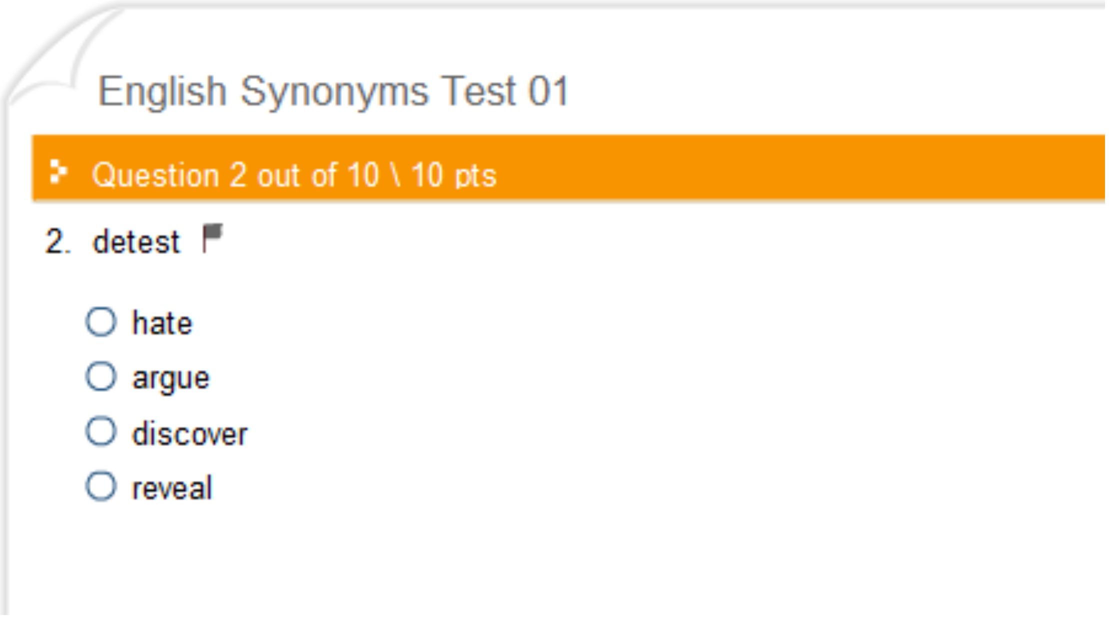


Evaluation of Vector-Space Lexical Semantics

- Have humans rate the semantic similarity of a large set of word pairs.
  - (dog, canine): 10; (dog, cat): 7; (dog, carrot): 3; (dog, knife): 1
- Compute vector-space similarity of each pair.
- Compute correlation coefficient (Pearson or Spearman) between human and machine ratings.

TOEFL Synonymy Test

- LSA shown to be able to pass TOEFL synonymy test.



English Synonyms Test 01

Question 2 out of 10 \ 10 pts

2. detest

○ hate
○ argue
○ discover
○ reveal

Summary

- A word’s meaning can be represented as a vector that encodes distributional information about the contexts in which the word tends to occur.
- Lexical semantic similarity can be judged by ‘comparing’ vectors (we’ll see more formal ‘similarity’ functions next time).
- Can we use neural networks to get ‘better’ vectors?