Gridspace

Gridspace IAP Lecture 5
Syntax, Semantics, and Embeddings

January 18, 2023

Central Questions

How do you determine the meaning of an expression?

How can you represent meaning so that a computer can reason about it?

Goals

1. Start to think about **meaning and language** in a critical way

2. Get introduced to some of the **formalisms and intuitions** linguists apply to language

3. Look at how you can **encoding language for NLP** tasks

4. Gain some intuition about how to **think about and compare these encodings**

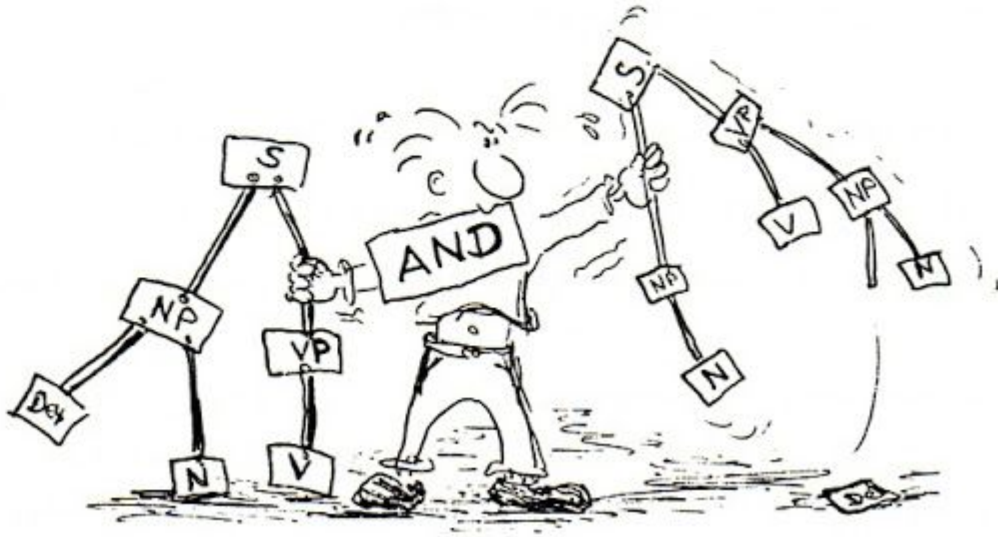
Determining Meaning

The meaning of a complex expression is determined by *the meaning of its* ***structure*** and the ***meanings of its constituents***

- *Frege's principle of compositionality*

Syntax

Structure



Structure is important

In Math....

$(4 + 1) \times 8 = 40$

$4 + (1 \times 8) = 12$

And in Grammar....

Lets eat, grandma.

Let’s eat grandma.

Structure is important

In Math....

(4 + 1) x 8 = 40

4 + (1 x 8) = 12

And in Grammar....

Lets eat, grandma.

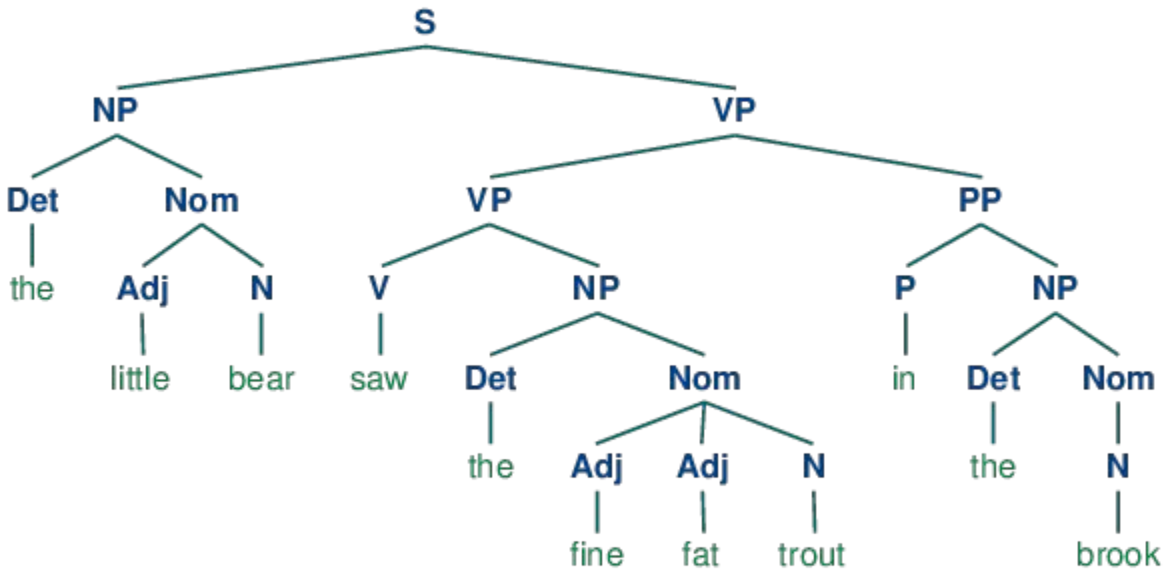
Let's eat grandma.

Commas save lives

Parts

Syntactic Category	Description
whole	whole sentence
adp	adverb phrase
vp	verb phrase
np	noun phrase
np_vp	noun phrase + verb phrase
adp_vp	adverb phrase + verb phrase

And how they are combined



Grammar

There are rules about word Ordering

Bob likes Sally.

Sally likes Bob.

Likes Sally Bob.

Prescriptive Grammar

Then there are rules like...

Don't use passive voice

Use "less" for mass nouns and "fewer" for count nouns

Don't end sentences with a preposition

Prescriptive Grammar

"This is the type of arrant pedantry up with which I will not put."

Then there are rules like...

Don't use passive voice

Use "less" for mass nouns and "fewer" for count nouns

Don't end sentences with a preposition

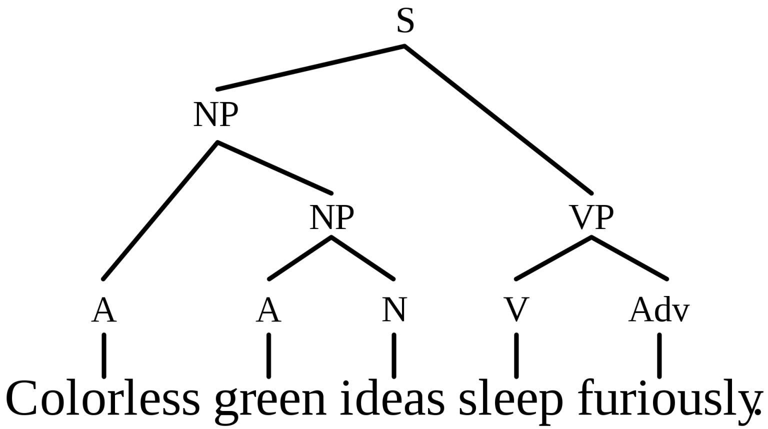
Descriptive Grammar

mental grammar: the system that all speakers of a language have in their minds, which allows them to understand each other.

A sentence is **grammatical** in a language if it follows the rules in the mental grammar of that language.

A sentence which is grammatical is said to be **well formed.**

Grammaticality is a separate concept from **plausibility.**



Ideas can sleep Furiously!

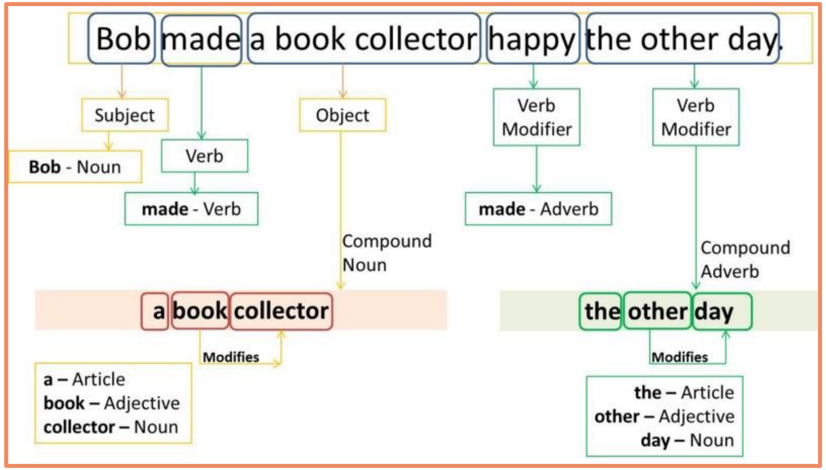
I have a friend who is always full of ideas, good ideas and bad ideas, fine ideas and crude ideas, old ideas and new ideas. Before putting his new ideas into practice, he usually sleeps over them to let them mature and ripen. However, when he is in a hurry, he sometimes puts his ideas into practice before they are quite ripe, in other words, while they are still green. Some of his green ideas are quite lively and colorful, but not always, some being quite plain and colorless. When he remembers that some of his colorless ideas are still too green to use, he will sleep over them, or let them sleep, as he puts it. But some of those ideas may be mutually conflicting and contradictory, and when they sleep together in the same night they get into furious fights and turn the sleep into a nightmare. Thus my friend often complains that his **colorless green ideas sleep furiously.**

The story so far

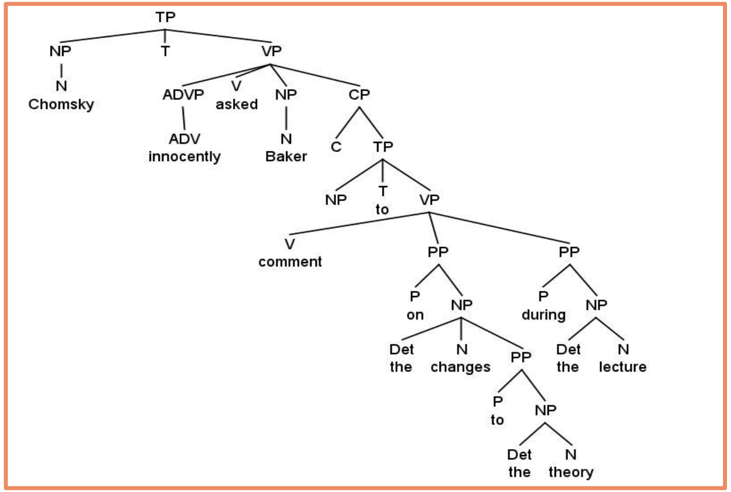
- **Syntax** is a study of the structure of sentences/ utterances
  - we think about **syntactic categories** and their **mode of combination**
- Sentence structure plays an important role in **determining meaning**
- There are different types of syntactic rules but ultimately we are most concerned with **rules derived from actual language use**

Downstream Utility

Part of speech tagging



Syntax trees



Semantics

Meaning

Sense and Reference

- **sense:** *mode of presentation*, or the mental representation of a thing or concept

- **referent***:* an physical thing in the world

- In more logic based terms: the referent is **the set of objects in the world** that match the definition given by the sense

Sense and Reference Examples

Consider the following sentences:

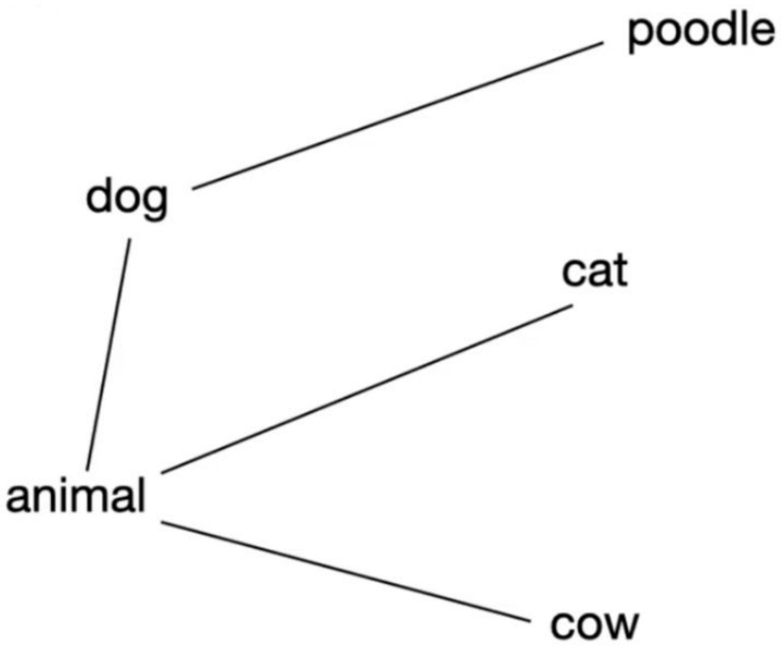
(1) M.J. does not know that *Peter Parker* is *Spiderman*.

(2.1) I think we should put the *couch* here.
(2.2) I think we should put the *sofa* here.

(3) I took out money from *the bank* after donating blood to *the blood bank*.

(4) I love unicorns.

Semantic Relationships



Interlude: Pragmatics

Figure 0.1
Rembrandt sketch

“An Utterance is not, as it were, a veridical model or “snapshot” of the scene it describes [...]. Rather, an utterance is just as sketchy as the Rembrandt drawing”

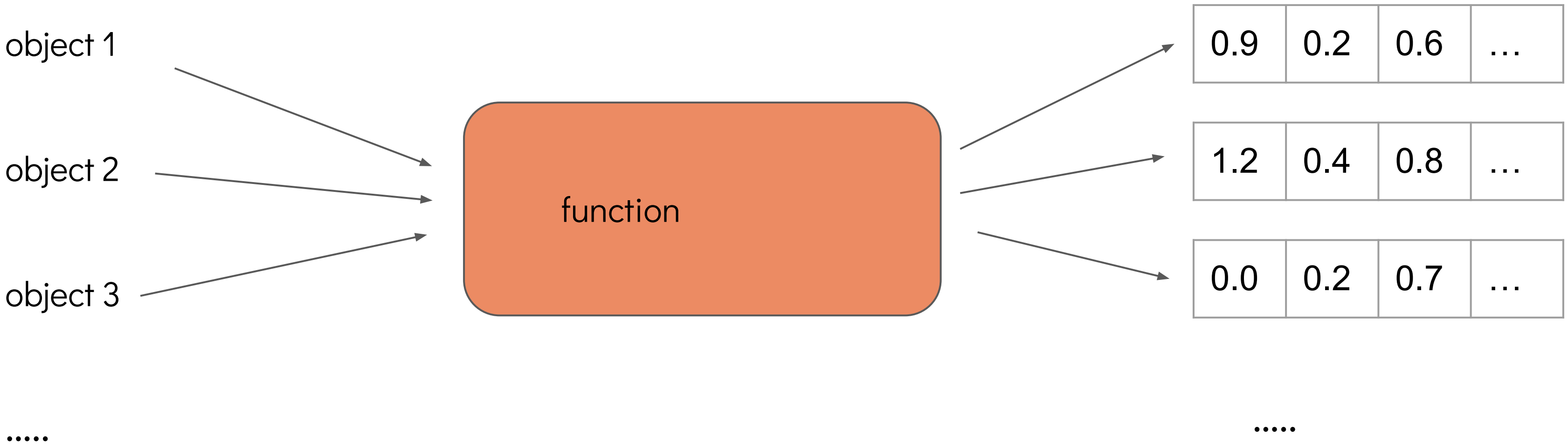
- Levinson’s *Presumtive meanings,* 2000

Interlude: Pragmatics

The story so far

- **Syntax** tells us about the structure of language
- (Lexical) **Semantics** tells us about word and phrase meanings
  - we learned to think about **senses and referents**
  - we used these concepts to build up notions of **word relatedness**
  - words meanings exist in a **web of semantic relationships**
- **Pragmatics** reminds us that the words in and structure of a sentence often don't tell the whole story
  - we enrich the meaning of a sentence with an understanding of broader contexts

Embeddings



Interlude: Vectors

A vector is an **ordered list of numbers**

It may be written as:

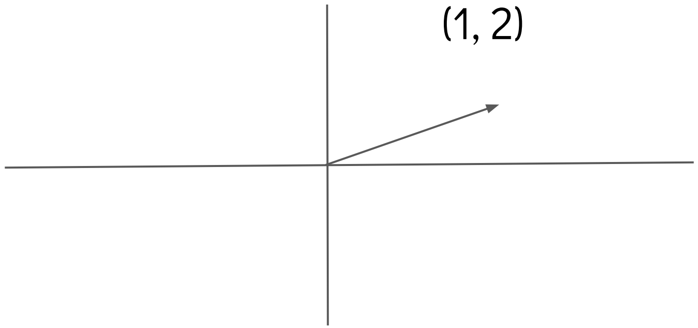
$$(0.1, 0.3, -0.6, 1.6) \text{ or } \begin{bmatrix} 0.1 \\ 0.3 \\ -0.6 \\ 1.6 \end{bmatrix}$$

The length of the vector is its **dimension**

The vector above has dimension 4, it is 4-vector, which can be written as:

$$(0.1, 0.3, -0.6, 1.6) \in \mathbb{R}^4$$

Interlude: Vectors

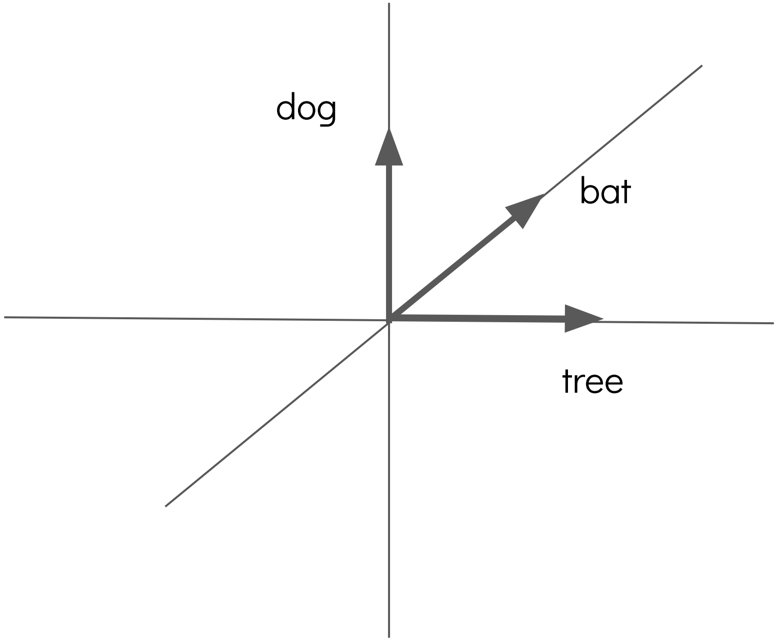


Vector Representations

- Machines are very good at matrix/vector manipulation
- So we build functions which **maps words to points in space**
- This is the standard way to represent word meaning in NLP
- These vectors can be used for lots of different tasks

One-Hot Encodings

dog = [1 0 0]
bat = [0 1 0]
tree = [0 0 1]



Can we do better?

Pros:

Allows for vector processing

Simple

Cons:

Inefficient in terms of....

- Space (very sparse)
- Information representation

Not semantically meaningful

$$(w^{hotel})^T w^{motel} = (w^{hotel})^T w^{cat} = 0$$

Co-Occurrence

"You shall know a word by the company it keeps" - J.R. Firth, 1975

- **Distributional Semantics:** A word's meaning is given by the *context* in which it appears
  - A very successful idea in statistical NLP on which many early word embeddings are based

- **Conext:** Given word *w*, we may define *w*'s context as
  - The set of words that are present in documents in which *w* also appears
  - words that appear close to *w* in text e.g. within some fixed size window

Word-document matrix

fool = (36, 58, 1, 4) **battle** = (1, 0, 7, 13) **good** = (114, 80, 62, 89)
wit = (20, 15, 2, 3)

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Example taken from *Speech and Language Processing* by Jurafskly and Martin

Sliding window word-word matrix

is traditionally followed by	**cherry**	pie, a traditional dessert
often mixed, such as	**strawberry**	rhubarb pie. Apple pie
computer peripherals and personal	**digital**	assistants. These devices usually
a computer. This includes	**information**	available on the internet

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Example taken from *Speech and Language Processing* by Jurafskly and Martin

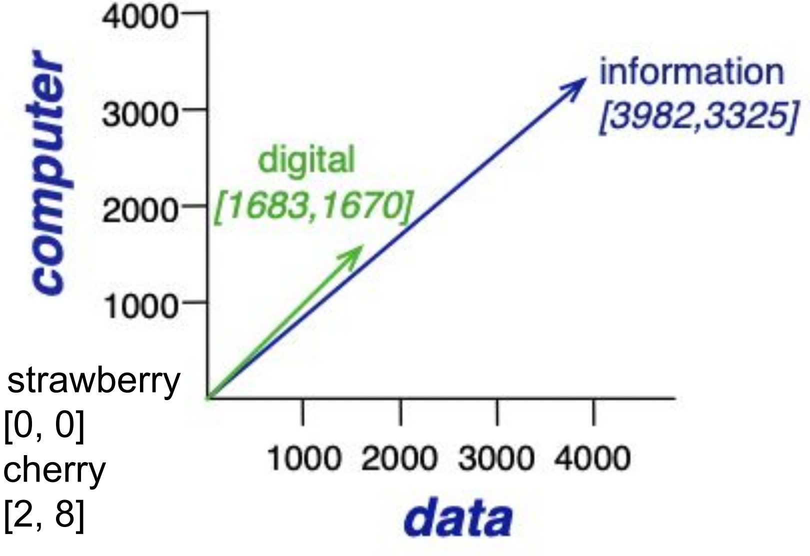
Sliding window word-word matrix

digital = (0, ... , 1670, 1683, 85, 5, 4, ...)

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Example taken from *Speech and Language Processing* by Jurafskly and Martin

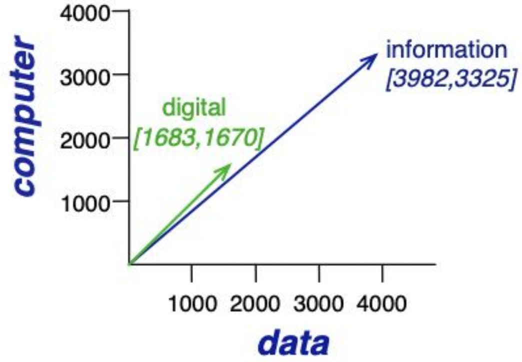
Sliding window word-word matrix



* strawberry
  [0, 0]
  cherry
  [2, 8]

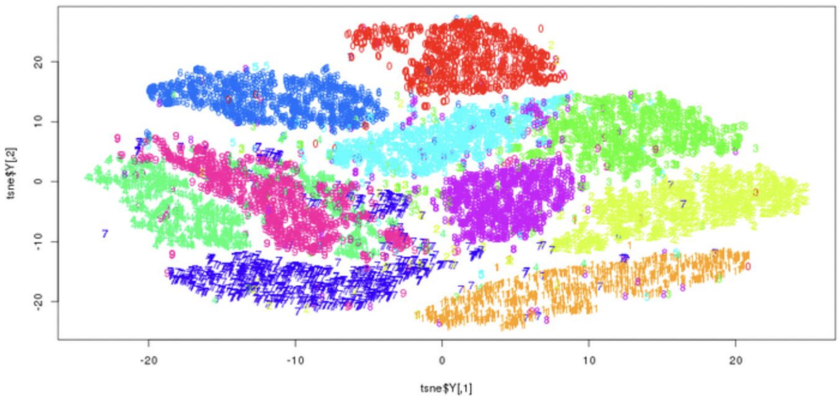
Example taken from *Speech and Language Processing* by Jurafskly and Martin

Interlude: Evaluating Word Vectors

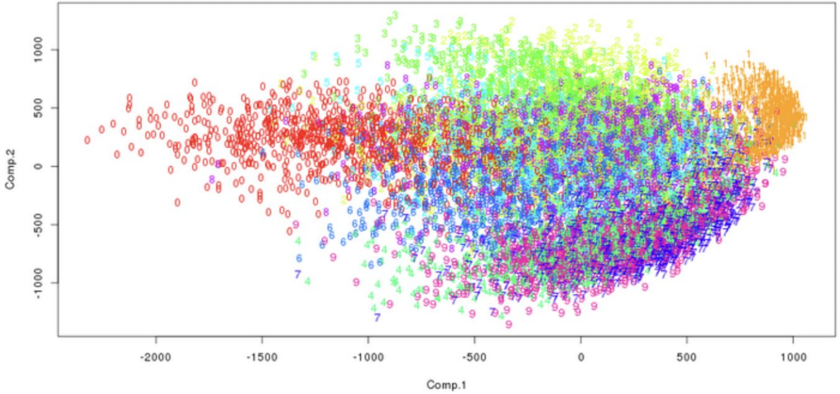


Lower-Dimensional Projections

T-SNE

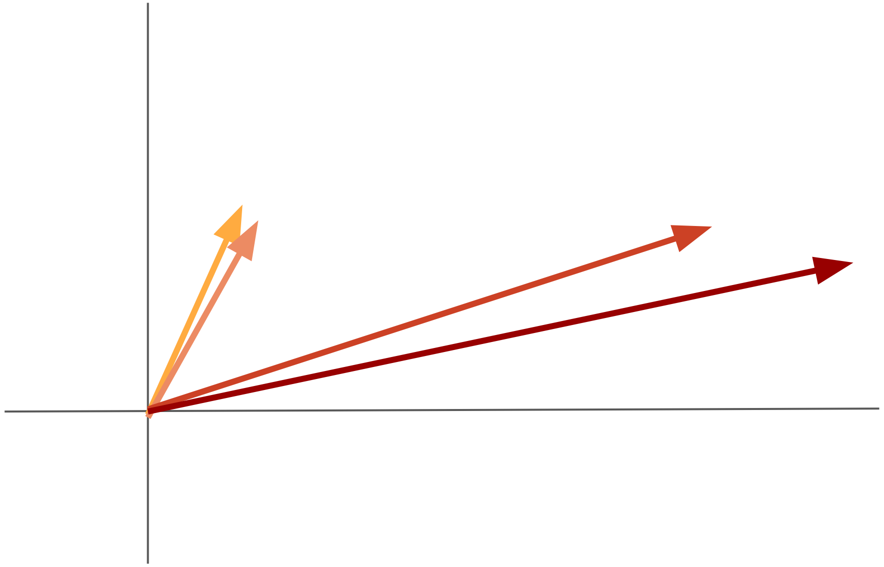


PCA



Quantitative Comparisons

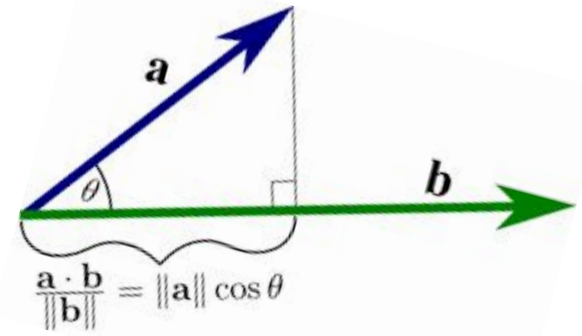
- Dot Product
- Jaccard Similarity
- Euclidean Distance
- Cosine Similarity



Quantitative Comparisons

- Dot Product
- Jaccard Similarity
- Euclidean Distance
- Cosine Similarity

Dot Product



Vector Length

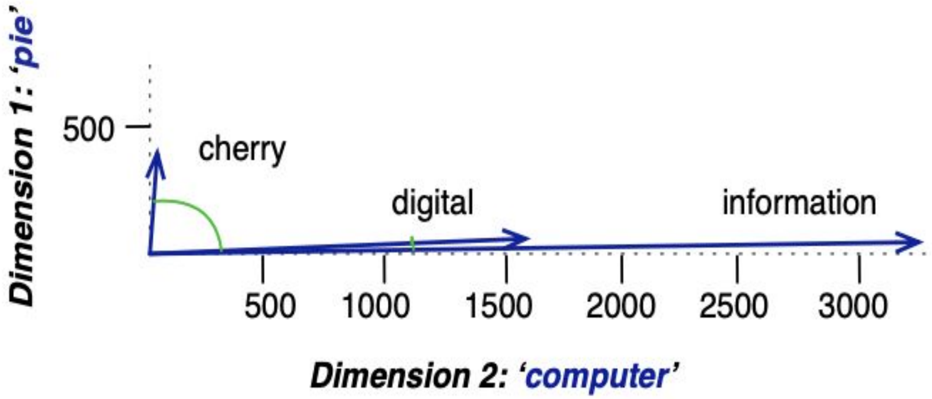
$$|\mathbf{v}| = \sqrt{\sum_{i=1}^{N} v_i^2}$$

Cosine Similarity

$$\mathbf{a}\cdot\mathbf{b} = |\mathbf{a}||\mathbf{b}|\cos\theta$$

$$\frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|} = \cos\theta$$

Cosine Similarity



Story so far

- We get word embeddings by **mapping words to vectors**
- The **relative location in space** of these vectors is **semantically meaningful**
- Options so far:
    - one-hot *(|V|- vectors)*
    - word-document *(|D|- vectors)*
    - word-word *(|V|- vectors)*
    - (can apply SVD/ matrix factorization to get denser representations)

Problems so far

- Vector **dimensions can change** every time a new word is added
- Matrices tend to be very **sparse**
- The matrices tend to be **high dimensional**
- Quadratic cost to perform SVD (train)

Problems so far

- Dimensions of vectors can change every time a new word is added
- Matrices tend to be very sparse
- The matrices tend to be high dimensional
- Quadratic cost to perform SVD (train)

How can we do better?

Iterative Based Methods

Word2Vec (Mikolov et al. 2013) provided a framework for learning word vectors from context in a **self supervised** manner.

Basic training idea: **predict context** words for some **center word** ***c***.

Model parameters, optimized by backpropagation, will eventually be our word vectors

Able to learn one iteration at a time rather than storing global information about a huge dataset

Word2vec

Overview:

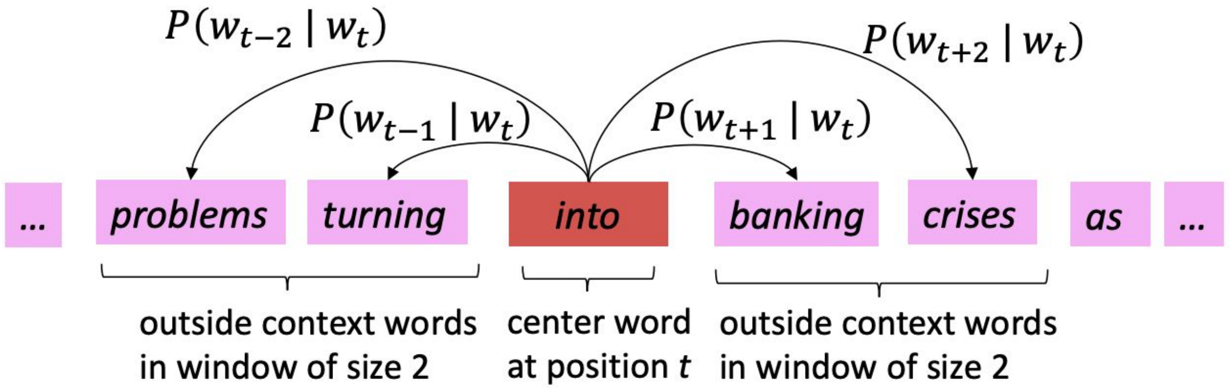
- Start with a large **corpus** of text
- Every word in the **fixed vocabulary** gets represented by a random initial vector
    - actually two - one as context and one as center word
- For each position *t* in the text, we say w_t is the **center word *c*,** where *c* is surrounded by **context ("outside") words *o***
- For each *c*, Calculate the **probability of context *o* given center word *c***
- Adjust word vectors to maximize this probability with **logistic regression**

Interlude: Bayes Law

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

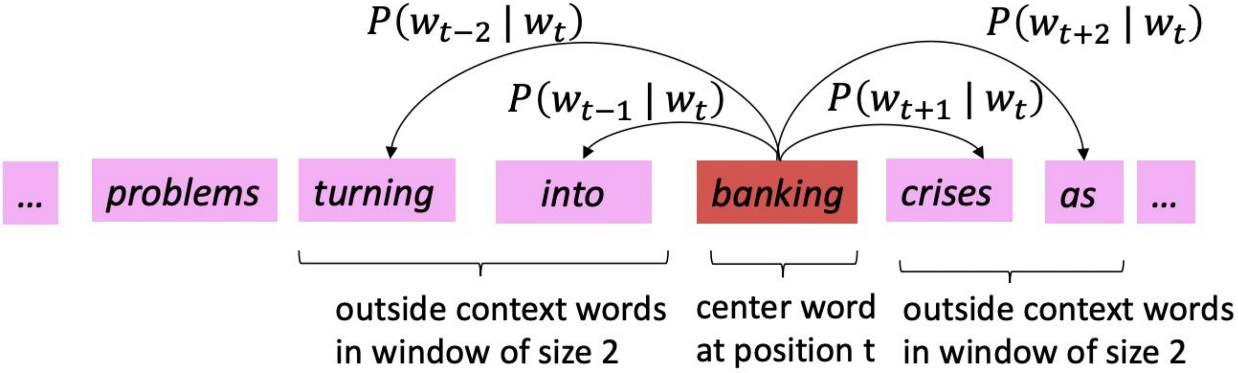
Sliding Window Probability

Example windows and process for computing $P(w_{t+j} \mid w_t)$



Sliding Window Probability

Example windows and process for computing $P(w_{t+j} \mid w_t)$



Important Variables

w_t - word at position t in the text

c - the center word

o - the context words

m - the size of the context window

θ - the variables to be optimized (will be the word embeddings)

$P(w_{t+j}|w_t;\theta)$ - the probability that word w_{t+j} is in the context window of w_t, calculated with embeddings from the parameters θ

Interlude: Bayes with word vectors

What is $P(w_{t+j}|w_t;\theta)$?

For each word w we have in the corpus we have two embeddings:

- $\mathbf{v_w}$ for when w is the center word
- $\mathbf{u_w}$ for when w is a context word

Both embeddings are defined in the matrix of parameters θ.

We then use the function

$$P(o|c) = \frac{exp(\mathbf{u_0} \cdot \mathbf{v_c})}{\sum_{w \in V} exp(\mathbf{u_0} \cdot \mathbf{v_c})}$$

Interlude: Softmax

exponentiation makes sure everything is positive

dot product to compare embedding similarity

$$P(o|c) = \frac{exp(\mathbf{u_0} \cdot \mathbf{v_c})}{\sum_{w \in V} exp(\mathbf{u_0} \cdot \mathbf{v_c})}$$

Normalizing over entire vocabulary makes sure the probability distribution sums to 1

Word2vec Objective Function

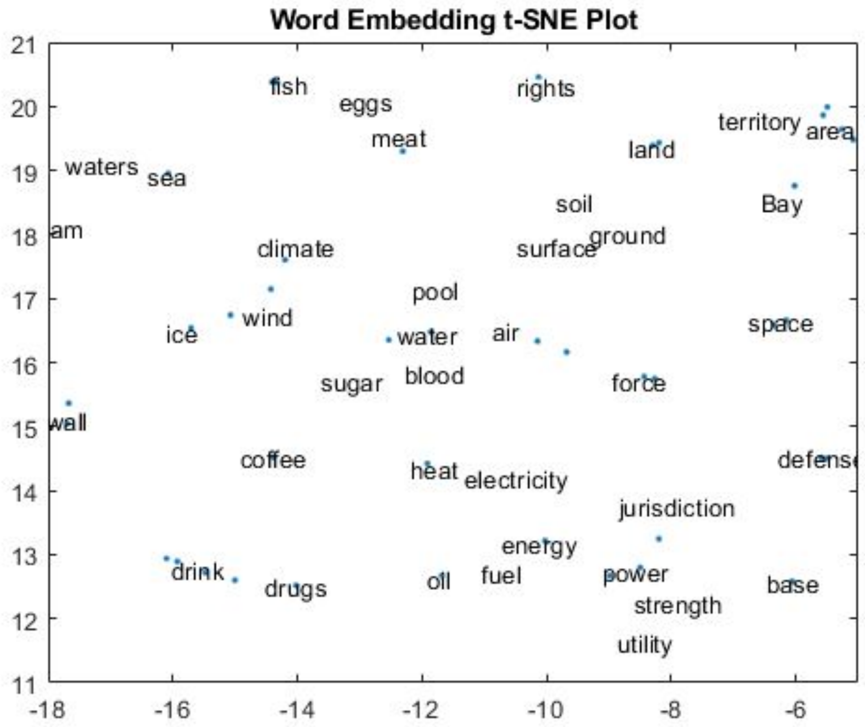
$L(\theta)$ is the **likelihood** (given current word vectors) that w_{t-m} through w_{t+m} are the context words surrounding w_T

$$L(\theta) = \prod_{t=1}^{T} \prod_{-m \leq j \leq m | j \neq 0} P(w_t + j | w_t; \theta)$$

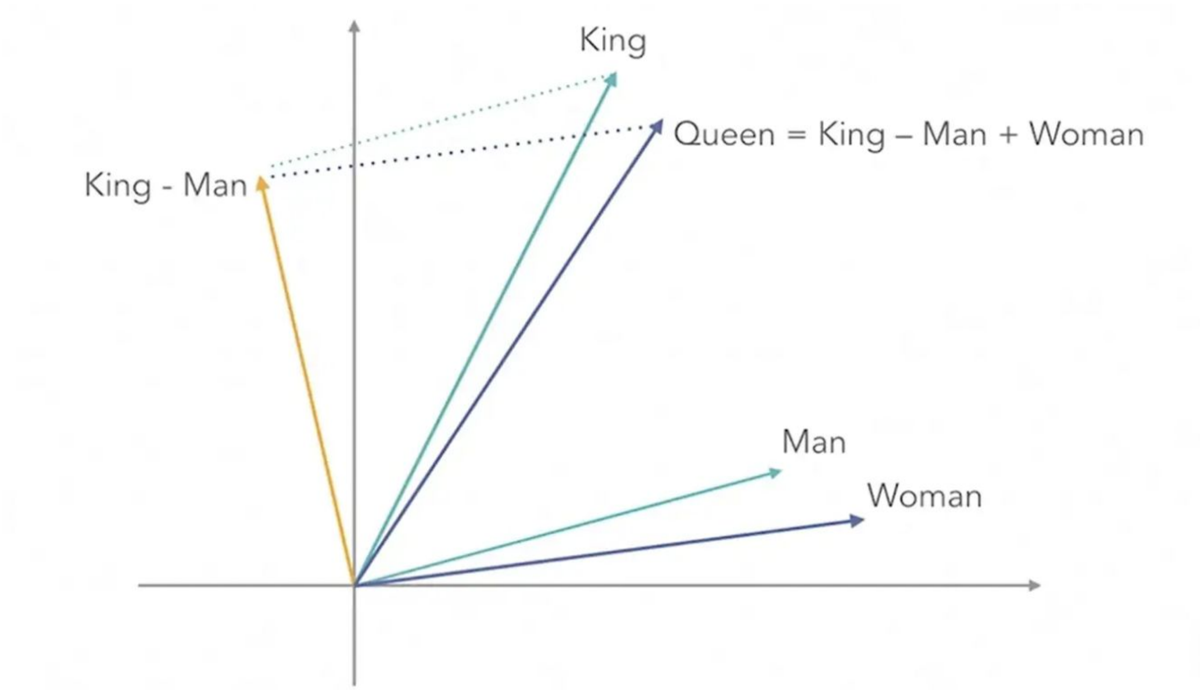
$J(\theta)$, the **objective function** (cost or loss function), is the **average negative log likelihood**

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m | j \neq 0} \log P(w_t + j | w_t; \theta)$$

Cool Semantic Space Properties



Cool Semantic Space Properties



Variations

- **Skip-Gram** model: predict context words given center words

- **CBOW** model: predict center word given context

- **GloVe**: Considers global word co-occurrence probabilities across the whole corpus
  - (if you are interested the original GloVe paper is pretty readable!)

The Story so far

- We talked about **deriving meaning from language** (syntax, semantics, pragmatics)
- We talked about how semantic meanings connect in a web of **semantic relationships**
- We introduced the idea of a **word embedding**
- We saw how **Word2Vec** can be used to extract semantic meaning by looking at contextual relationships between words

What can we do with word vectors?

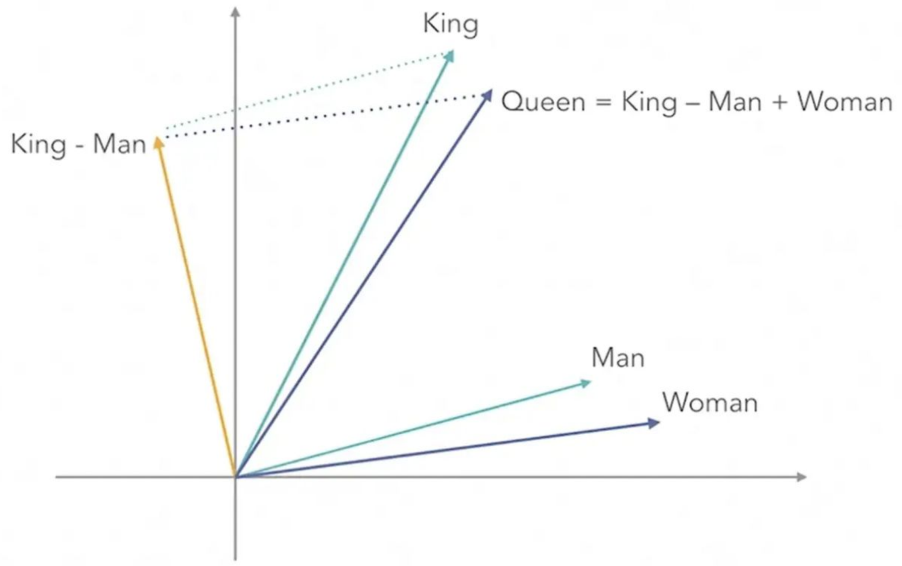
- All sorts of things!
  - document classification
  - sentiment analysis
  - search functions
  - etc.

Beware of Bias

$$\text{man} - \text{women} \cong \text{king} - \text{queen}$$

$$\text{man} - \text{women} \cong \text{computer programmer} - \text{homemaker}$$

Re-evaluating Word vectors



Next time: Transformers!

Answers from last time

- Split the word “antidisestablishmentarianism” into its morphemes. What does the word mean?

anti/dis/establish/ment/ari/an/ism - Means pro-establishment (double negation)

- Build your own, brand new word in Turkish

Exercise to the reader ;)

- Run the Byte Pair Encoding algorithm on the string **aaabdaaabac.** What is the smallest number of characters needed to encode this in a compressed form?

The most compressed form is **XdXac** where X = ZY, Y = ab, and Z = aa. It cannot be further compressed as there are no further byte pairs appearing more than once.

An Exercise to the Reader for Next Time:

1. Give an explanation of two different meanings the following sentence could have, extra points for writing out the associated syntax trees

    "The astronomer that the tourist saw had a telescope"

2. Come up with a sentence which is not grammatical but is semantically meaningful and one that is semantically meaningful but not traditionally grammatical

3. Come up with an example of how you might use word embeddings for an NLP task