

What is NLP?

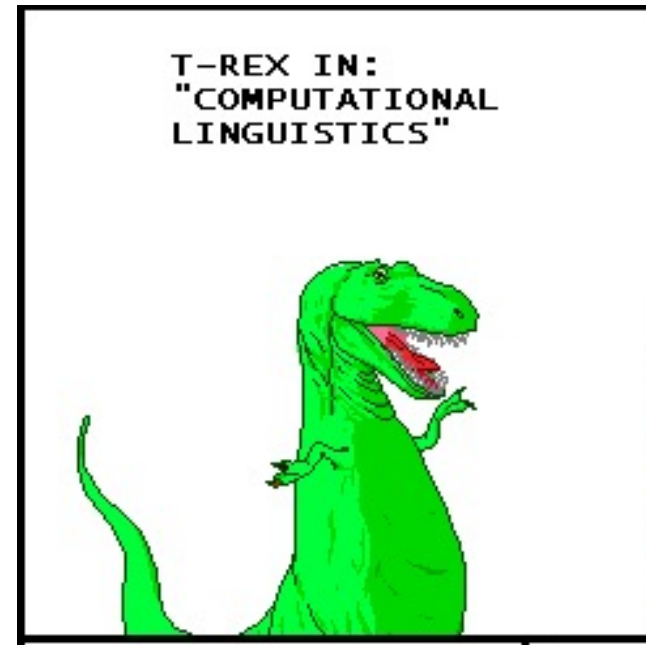
CMSC 473/673

Frank Ferraro – ferraro@umbc.edu



Today's Learning Goals

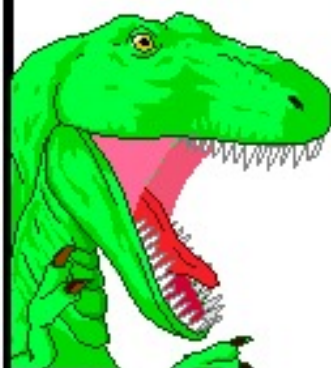
- NLP vs. CL
- Terminology:
 - NLP: vocabulary, token, type, one-hot encoding, dense embedding, parameter/weight, corpus/corpora
 - Linguistics: lexeme, morphology, syntax, semantics, “discourse”
- Universal Dependencies



T-REX IN:
"COMPUTATIONAL
LINGUISTICS"



Computational
linguistics is
the study of
computer-based
language
processing!



Natural Language Processing
 $\approx$
Computational Linguistics

Natural Language Processing

≈

Computational Linguistics

science focus

computational bio
computational chemistry
computational X

build a system to translate
create a QA system

engineering focus

Natural Language Processing

≈

Computational Linguistics

science focus

computational bio
computational chemistry
computational X

Natural Language Processing $\approx$ Computational Linguistics

Both have impact in/contribute to/draw from:

Machine learning

Linguistics

Information Theory

Cognitive Science

Data Science

Psychology

Systems Engineering

Political Science

Logic

Digital Humanities

Theory of Computation

Education

build a system to translate
create a QA system

engineering focus

Natural Language Processing

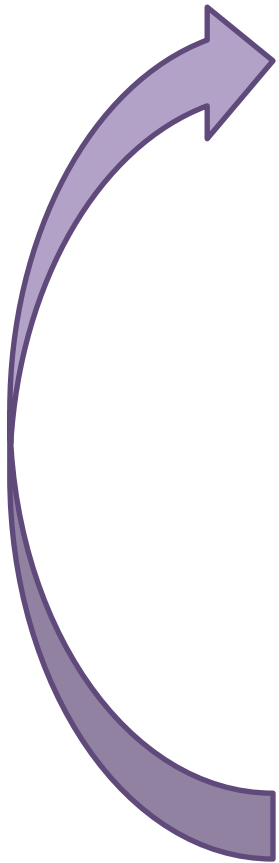
≈

Computational Linguistics

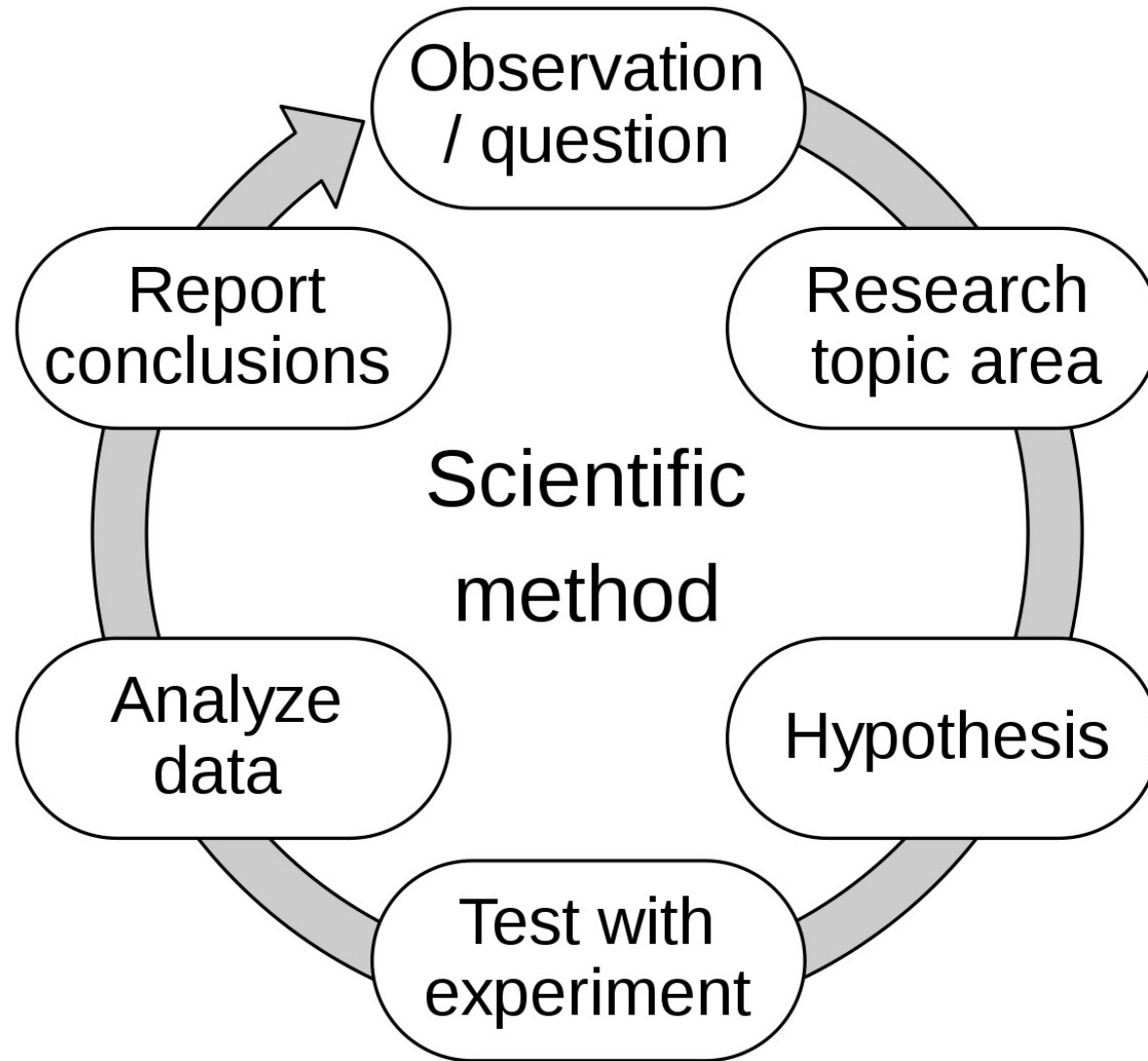
science focus

computational bio
computational chemistry
computational X

these views can co-exist peacefully



Whether it's Natural Language Processing or Computational Linguistics...



Where Do We Observe Language?

- All around us
- NLP/CL: from a **corpus** (pl: corpora)
 - Literally a “body” of text
- In real life:
 - Through curators (the LDC)
 - From the web (scrape Wikipedia, Reddit, etc.)
 - Via careful human elicitation (lab studies, crowdsourcing)
 - From previous efforts
- In this class: the **Universal Dependencies**

universaldependencies.org

Afrikaans	49K	LF	-	✓	✓	✓	✓	✓	✓
Ancient Greek	202K	LF	-	✓	✓	✓	✓	✓	✓
Ancient Greek-PROIEL	211K	LF	-	✓	✓	✓	✓	✓	✓
Arabic	242K	LF	-	✓	✓	✓	✓	✓	✓
Arabic-NYUAD	629K	LF	-	✓	✓	✓	✓	✓	✓
Arabic-PUD	20K	LF	-	✓	✓	✓	✓	✓	✓
Basque	121K	LF	-	✓	✓	✓	✓	✓	✓
Belarusian	8K	LF	-	✓	✓	✓	✓	✓	✓
Bulgarian	156K	LF	-	✓	✓	✓	✓	✓	✓
Buryat	10K	LF	-	✓	✓	✓	✓	✓	✓
Catalan	530K	LF	-	✓	✓	✓	✓	✓	✓
Chinese	123K	LF	-	✓	✓	✓	✓	✓	✓
Chinese-CFL	7K	LF	-	✓	✓	✓	✓	✓	✓
Chinese-PUD	21K	LF	-	✓	✓	✓	✓	✓	✓
Coptic	4K	LF	-	✓	✓	✓	✓	✓	✓
Croatian	197K	LF	-	✓	✓	✓	✓	✓	✓
Czech	1,503K	LF	-	✓	✓	✓	✓	✓	✓
Czech-CAC	493K	LF	-	✓	✓	✓	✓	✓	✓
Czech-CLTT	37K	LF	-	✓	✓	✓	✓	✓	✓
Czech-PUD	18K	LF	-	✓	✓	✓	✓	✓	✓
Danish	100K	LF	-	✓	✓	✓	✓	✓	✓
Dutch	209K	LF	-	✓	✓	✓	✓	✓	✓
Dutch-LassySmall	101K	LF	-	✓	✓	✓	✓	✓	✓
English	254K	LF	-	✓	✓	✓	✓	✓	✓
English-ESL	88K	LF	-	✓	✓	✓	✓	✓	✓
English-LinES	82K	LF	-	✓	✓	✓	✓	✓	✓
English-PUD	21K	LF	-	✓	✓	✓	✓	✓	✓
English-ParTUT	49K	LF	-	✓	✓	✓	✓	✓	✓
Estonian	47K	LF	-	✓	✓	✓	✓	✓	✓
Finnish	202K	LF	-	✓	✓	✓	✓	✓	✓
Finnish-FTB	159K	LF	-	✓	✓	✓	✓	✓	✓
Finnish-PUD	15K	LF	-	✓	✓	✓	✓	✓	✓
French	391K	LF	-	✓	✓	✓	✓	✓	✓
French-FTB	556K	LF	-	✓	✓	✓	✓	✓	✓
French-PUD	24K	LF	-	✓	✓	✓	✓	✓	✓
French-ParTUT	27K	LF	-	✓	✓	✓	✓	✓	✓
French-Sequoia	68K	LF	-	✓	✓	✓	✓	✓	✓
Galician	126K	LF	-	✓	✓	✓	✓	✓	✓
Galician-TreeGal	23K	LF	-	✓	✓	✓	✓	✓	✓
German	287K	LF	-	✓	✓	✓	✓	✓	✓
German-PUD	20K	LF	-	✓	✓	✓	✓	✓	✓
Gothic	55K	LF	-	✓	✓	✓	✓	✓	✓
Greek	61K	LF	-	✓	✓	✓	✓	✓	✓
Hebrew	115K	LF	-	✓	✓	✓	✓	✓	✓
Hindi	351K	LF	-	✓	✓	✓	✓	✓	✓
Hindi-PUD	23K	LF	-	✓	✓	✓	✓	✓	✓
Hungarian	42K	LF	-	✓	✓	✓	✓	✓	✓
Indonesian	121K	LF	-	✓	✓	✓	✓	✓	✓
Indonesian-PUD	25K	LF	-	✓	✓	✓	✓	✓	✓
Irish									
Italian									
Italian-PUD									
Italian-ParTUT									
Japanese									
Japanese-KTC									
Japanese-PUD									
Kazakh									
Korean									
Korean-PUD									
Kurmanji									
Latin									
Latin-ITTB									
Latin-PROIEL									
Latvian									
Lithuanian									
Maltese									

North Sami	55K	LF	-	✓	✓	✓	✓	✓	✓
Norwegian-Bokmaal	310K	LF	-	✓	✓	✓	✓	✓	✓
Norwegian-Nynorsk	301K	LF	-	✓	✓	✓	✓	✓	✓
Old Church Slavonic	57K	LF	-	✓	✓	✓	✓	✓	✓
Persian	151K	LF	-	✓	✓	✓	✓	✓	✓
Polish	82K	LF	-	✓	✓	✓	✓	✓	✓
Portuguese	210K	LF	-	✓	✓	✓	✓	✓	✓
Portuguese-BR	297K	LF	-	✓	✓	✓	✓	✓	✓
Portuguese-PUD	21K	LF	-	✓	✓	✓	✓	✓	✓
Romanian	218K	LF	-	✓	✓	✓	✓	✓	✓
Russian	99K	LF	-	✓	✓	✓	✓	✓	✓
Russian-PUD	19K	LF	-	✓	✓	✓	✓	✓	✓
Russian-SynTagRus	1,107K	LF	-	✓	✓	✓	✓	✓	✓
Sanskrit	1K	LF	-	✓	✓	✓	✓	✓	✓
Serbian	86K	LF	-	✓	✓	✓	✓	✓	✓
Slovak	106K	LF	-	✓	✓	✓	✓	✓	✓
Slovenian	140K	LF	-	✓	✓	✓	✓	✓	✓
Slovenian-SST	29K	LF	-	✓	✓	✓	✓	✓	✓
Spanish	423K	LF	-	✓	✓	✓	✓	✓	✓
Spanish-AnCor	547K	LF	-	✓	✓	✓	✓	✓	✓
Spanish-PUD	22K	LF	-	✓	✓	✓	✓	✓	✓
Swedish	96K	LF	-	✓	✓	✓	✓	✓	✓
Swedish-LinES	79K	LF	-	✓	✓	✓	✓	✓	✓
Swedish-PUD	19K	LF	-	✓	✓	✓	✓	✓	✓
Swedish Sign Language	<1K	LF	-	✓	✓	✓	✓	✓	✓
Tamil	8K	LF	-	✓	✓	✓	✓	✓	✓
Thai-PUD	23K	LF	-	✓	✓	✓	✓	✓	✓
Turkish	56K	LF	-	✓	✓	✓	✓	✓	✓
Turkish-PUD	16K	LF	-	✓	✓	✓	✓	✓	✓
Ukrainian	25K	LF	-	✓	✓	✓	✓	✓	✓
Upper Sorbian	11K	LF	-	✓	✓	✓	✓	✓	✓
Urdu	138K	LF	-	✓	✓	✓	✓	✓	✓
Uyghur	11K	LF	-	✓	✓	✓	✓	✓	✓
Vietnamese	43K	LF	-	✓	✓	✓	✓	✓	✓

Upcoming UD Treebanks

Amharic	-	-	?	-	✓	✓	✓	✓	✓
Armenian	-	-	✓	✓	✓	✓	✓	✓	✓
Bangla	-	-	✓	✓	✓	✓	✓	✓	✓
Bengali-DDS	-	-	✓	✓	✓	✓	✓	✓	✓
Cantonese	-	-	✓	✓	✓	✓	✓	✓	✓
Chinese-HK	-	-	✓	✓	✓	✓	✓	✓	✓

<http://universaldependencies.org/>
part-of-speech & syntax for > 120 languages

What Are Words?

bat



What Are Words?

bats



What Are Words?

Fledermaus

flutter mouse



What Are Words?

pişirdiler

They cooked it.

What Are Words?

pişmişlermişlerdi

They had it cooked it.

What Are Words?

):

What Are Words?

my leg is hurting nasty):



What Are Words?

add two cups (a pint): bring to a boil



What Are Words?

Hard to get agreement

(Human) Language-dependent

White-space separation is a sometimes okay (for
written English longform)

*Social media? Spoken vs. written? Other
languages?*

What Are Words? Tokens vs. Types

The film got a great opening and the film went on to become a hit .

Vocabulary: the words (items) you know

Type: an element of the vocabulary.

Token: an instance of that type in running text.

How many of types & tokens appear in the above sentence?

Terminology: Tokens vs. Types

The film got a great opening and the film went on to become a hit .

Types

- The
- film
- got
- a
- great
- opening
- and
- the
- went
- on
- to
- become
- hit
- .

Tokens

- The
- film
- got
- a
- great
- opening
- and
- the
- film
- went
- on
- to
- become
- a
- hit
- .

Terminology: Tokens vs. Types

The film got a great opening and the film went on to become a hit .

Types

- The
- film
- got
- a
- great
- opening
- and
- the
- went
- on
- to
- become
- hit
- .

Tokens

- The
- film
- got
- a
- great
- opening
- and
- the
- ~~film~~
- went
- on
- to
- become
- ~~a~~
- hit
- .

Representing a Linguistic “Blob”

1. An array of sub-blobs

word → array of **characters**
sentence → array of **words**

*How do you
represent **these**?*

Representing a Linguistic “Blob”

1. An array of sub-blobs

word → array of **characters**
sentence → array of **words**

*How do you
represent **these**?*

2. Integer representation/one-hot encoding

3. Dense embedding

Representing a Linguistic “Blob”

1. An array of sub-blobs
word $\rightarrow$ array of characters
sentence $\rightarrow$ array of words
2. Integer
representation/one-hot
encoding
3. Dense embedding

Let V = vocab size (# types)

1. Represent each word *type* with a unique integer i , where $0 \leq i < V$

Representing a Linguistic “Blob”

1. An array of sub-blobs
word $\rightarrow$ array of characters
sentence $\rightarrow$ array of words
2. Integer
representation/one-hot
encoding
3. Dense embedding

Let V = vocab size (# types)

1. Represent each word *type* with a unique integer i , where $0 \leq i < V$
2. Or equivalently, ...
 - Assign each word to some index i , where $0 \leq i < V$
 - Represent each word w with a V -dimensional **binary** vector e_w , where $e_{w,i} = 1$ and 0 otherwise

One-Hot Encoding Example

- Let our vocab be {a, cat, saw, mouse, happy}

Q: What is V (# types)?

One-Hot Encoding Example

- Let our vocab be {a, cat, saw, mouse, happy}
- $V = \# \text{ types} = 5$
- Assign:

a	4
cat	2
saw	3
mouse	0
happy	1

How do we
represent “cat?”

One-Hot Encoding Example

- Let our vocab be {a, cat, saw, mouse, happy}
- $V = \# \text{ types} = 5$
- Assign:

a	4
cat	2
saw	3
mouse	0
happy	1

How do we
represent “cat?”

$$e_{\text{cat}} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

How do we
represent
“happy?”

One-Hot Encoding Example

- Let our vocab be {a, cat, saw, mouse, happy}
- $V = \# \text{ types} = 5$
- Assign:

a	4
cat	2
saw	3
mouse	0
happy	1

How do we
represent "cat?"

$$e_{\text{cat}} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

How do we
represent
"happy?"

$$e_{\text{happy}} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

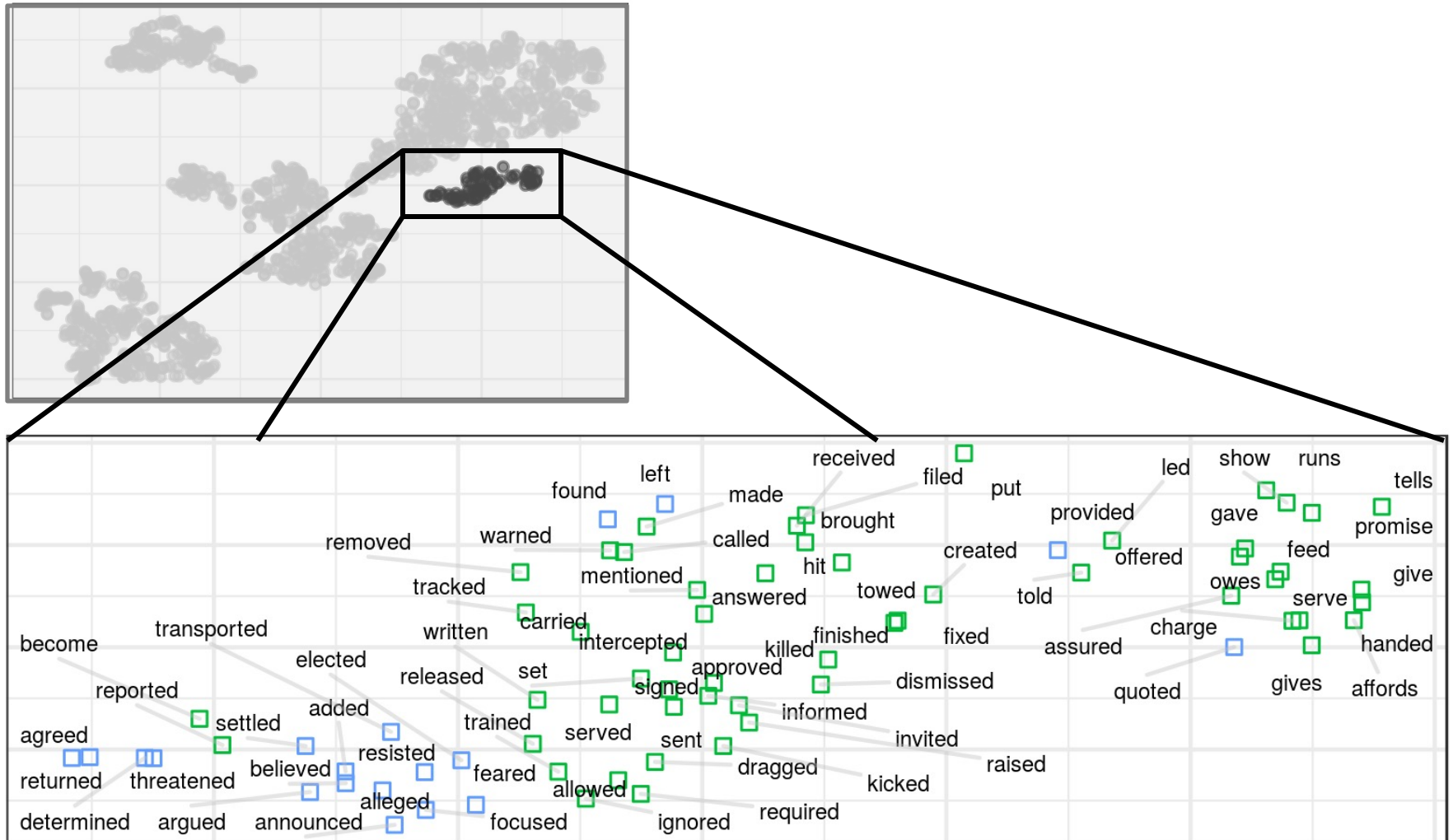
Representing a Linguistic “Blob”

1. An array of sub-blobs
word $\rightarrow$ array of characters
sentence $\rightarrow$ array of words
2. Integer
representation/one-hot
encoding
3. Dense embedding

Let E be some *embedding size* (often 100, 200, 300, etc.)

Represent each word w with an E -dimensional **real-valued** vector e_w

A Dense Representation ($E=2$)



Preview
for later!

(Some) Properties of Embeddings

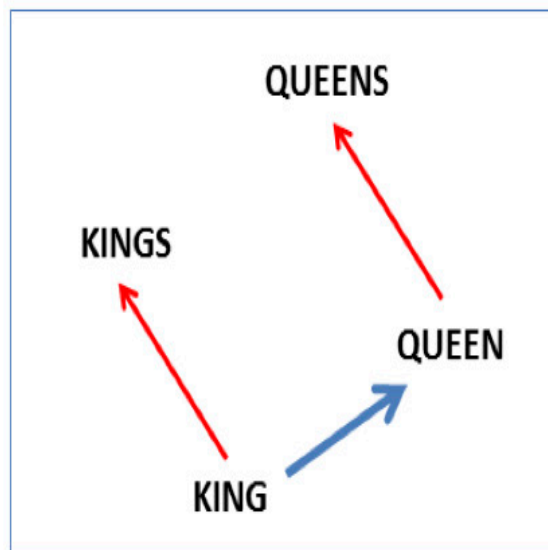
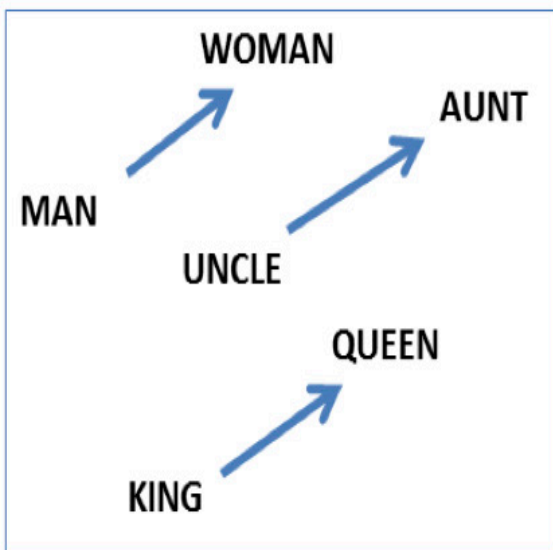
Capture “like” (similar) words



https://media3.giphy.com/media/3orif0M8U1E7NfpFzg/200_s.gif

target:	Redmond	Havel	ninjutsu	graffiti	capitulate
	Redmond Wash.	Vaclav Havel	ninja	spray paint	capitulation
	Redmond Washington	president Vaclav Havel	martial arts	grafitti	capitulated
	Microsoft	Velvet Revolution	swordsmanship	taggers	capitulating

Capture relationships



$\text{vector}(\text{'king'}) - \text{vector}(\text{'man'}) + \text{vector}(\text{'woman'}) \approx \text{vector}(\text{'queen'})$

$\text{vector}(\text{'Paris'}) - \text{vector}(\text{'France'}) + \text{vector}(\text{'Italy'}) \approx \text{vector}(\text{'Rome'})$



Preview
for later!



**T-REX IN:
"COMPUTATIONAL
LINGUISTICS"**



Computational linguistics is the study of computer-based language processing!



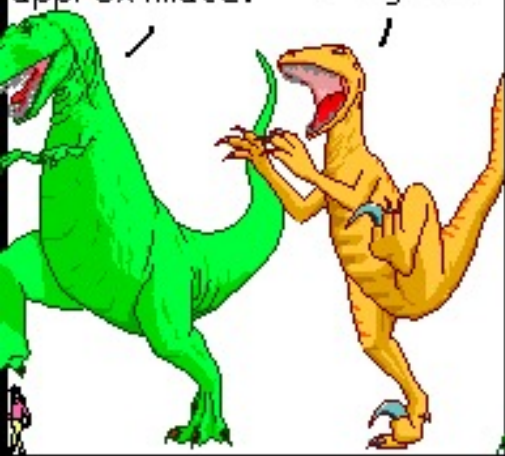
A major area of computational linguistics is that of "ambiguity resolution". It turns out that many things people say in a language - English, for example - can have more than one meaning!



Consider the phrase "fruit flies like a banana". Is it describing the taste of fruit flies, or rather flying fruit? How can a computer hope to figure this out?



Many have focused on statistical modelling of language, but this approach is approximate. I agree!



NLP $\leftrightarrow$ Machine Learning

Goal: Learn parameters
(weights) θ to develop a
scoring function that
says how “good” some
provided text is

Michael Jordan,
coach Phil Jackson
and the star cast,
including Scottie
Pippen, took the
Chicago Bulls to six
National Basketball
Association
championships.

Goal: Learn parameters (weights) θ to develop a scoring
function that says how “good” some provided **text** is

$p_{\theta}(\text{Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.})$

Goal: Learn parameters (weights) θ to develop a **scoring function** that says how “good” some provided **text** is



Terminology: **parameters**: primary “knobs” of the model that are set by a learning algorithm

p_{θ} (

Michael Jordan,
coach Phil Jackson
and the star cast,
including Scottie
Pippen, took the
Chicago Bulls to six
National Basketball
Association
championships.

Goal: Learn **parameters (weights)** θ to develop a **scoring function** that says how “good” some provided **text** is

$$S = p_{\theta}(\text{Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.})$$

Goal: Learn parameters (weights) θ to develop a scoring function that says how “good” some provided text is

Q: If we make p to be a probability distribution, what are the minimum and maximum values of S ?

$$S = p_{\theta}(\text{Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.})$$

Goal: Learn parameters (weights) θ to develop a scoring function that says how “good” some provided text is

Q: If we make p to be a probability distribution, what are the minimum and maximum values of s ?

$$A: 0 \leq s \leq 1$$

$$s = p_{\theta} \left(\text{Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.} \right)$$

Goal: Learn parameters (weights) θ to develop a scoring function that says how “good” some provided text is

Use ML Techniques to Learn the Weights

(probabilistic) model

$$p_{\theta}(X)$$

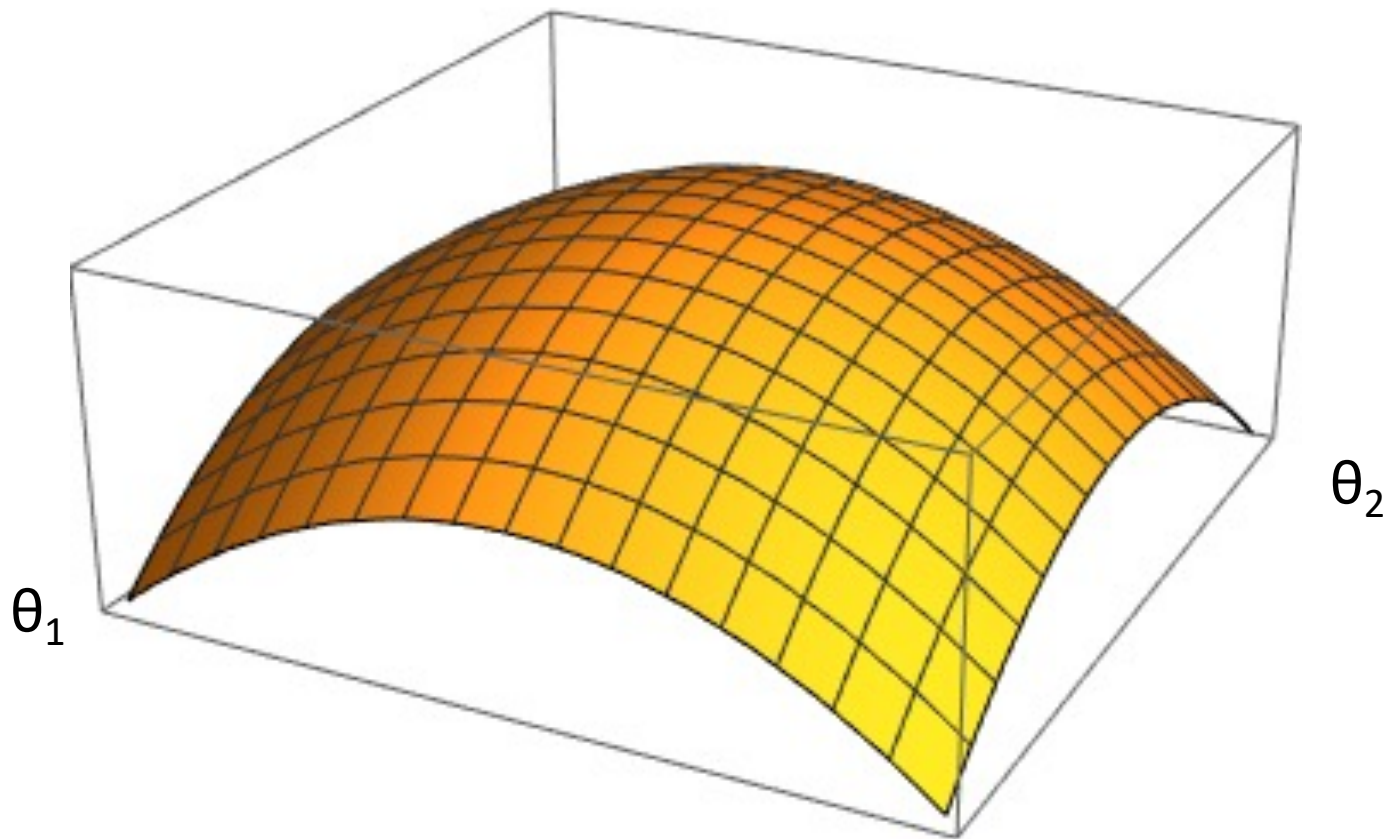


objective

$$F(\theta)$$

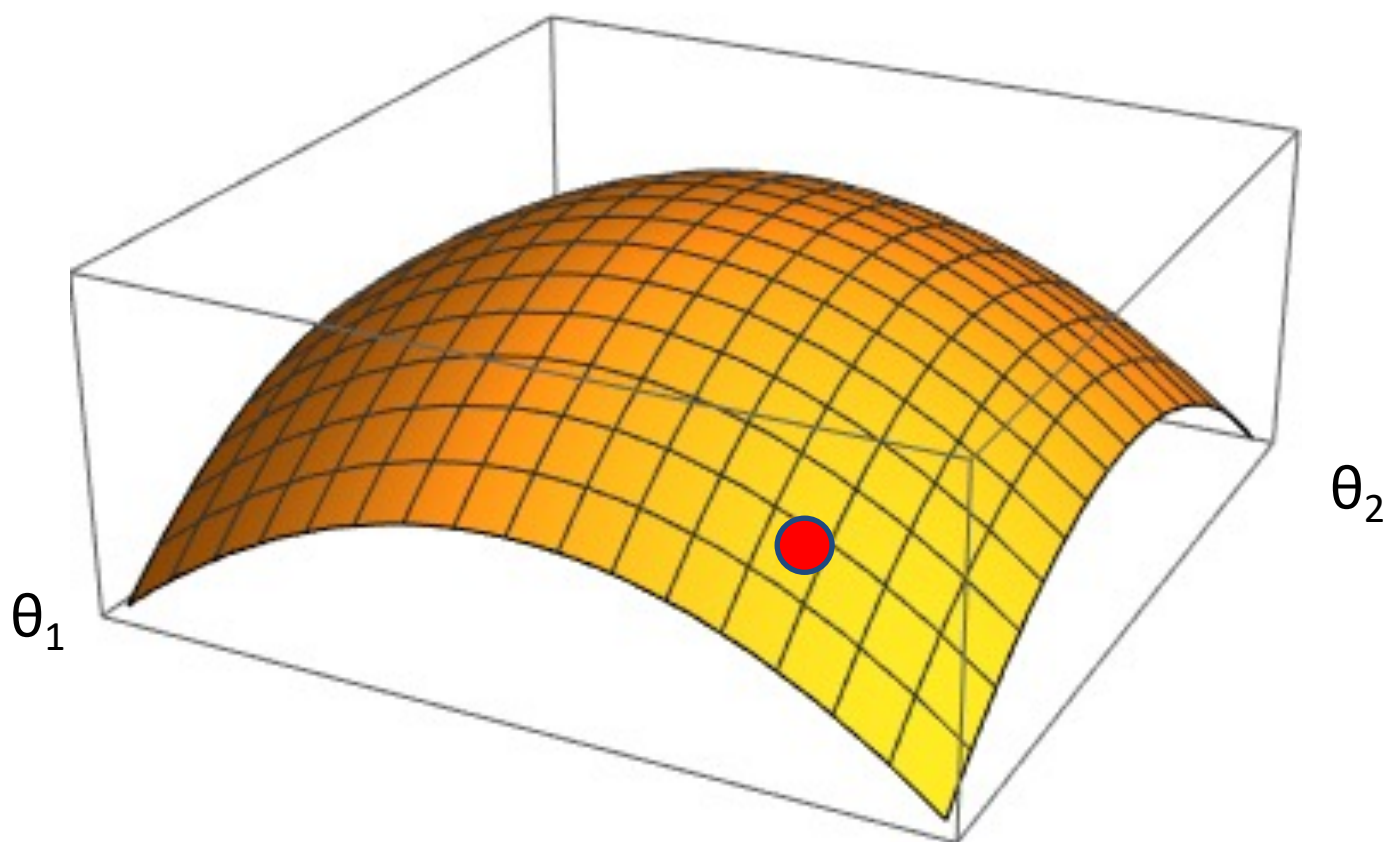
Gradient Ascent

$$\arg \max_{\theta} F(\theta)$$



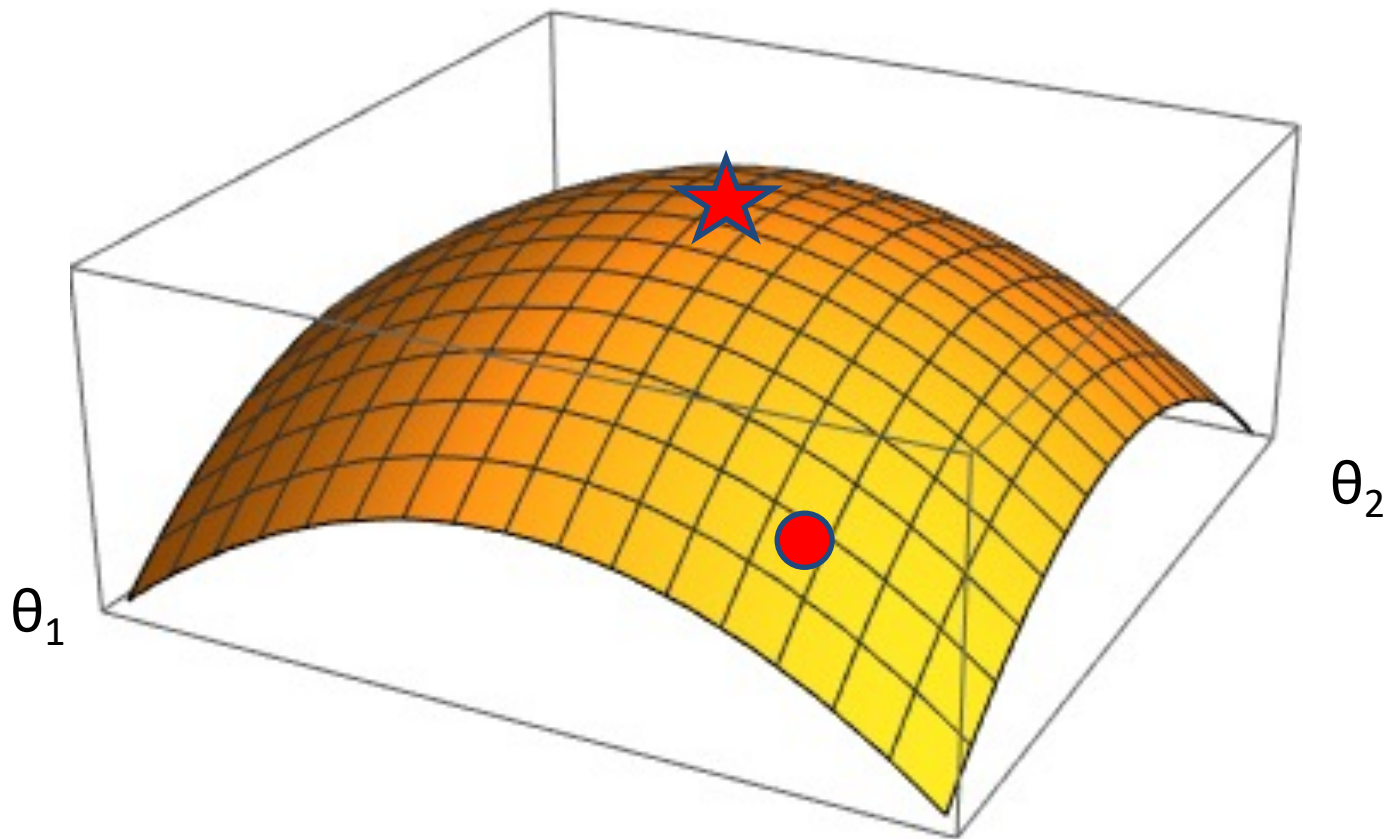
Gradient Ascent

$$\arg \max_{\theta} F(\theta)$$



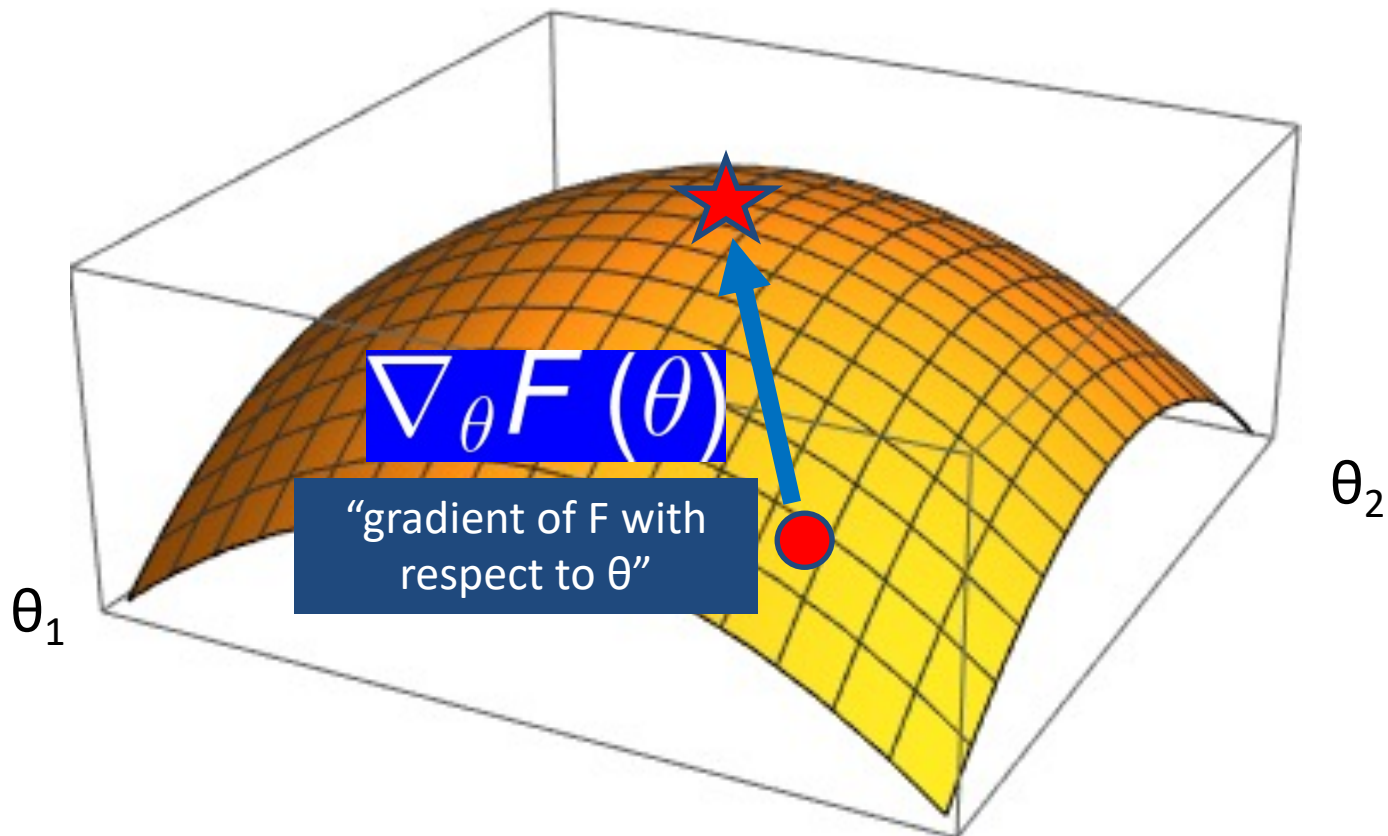
Gradient Ascent

$$\arg \max_{\theta} F(\theta)$$



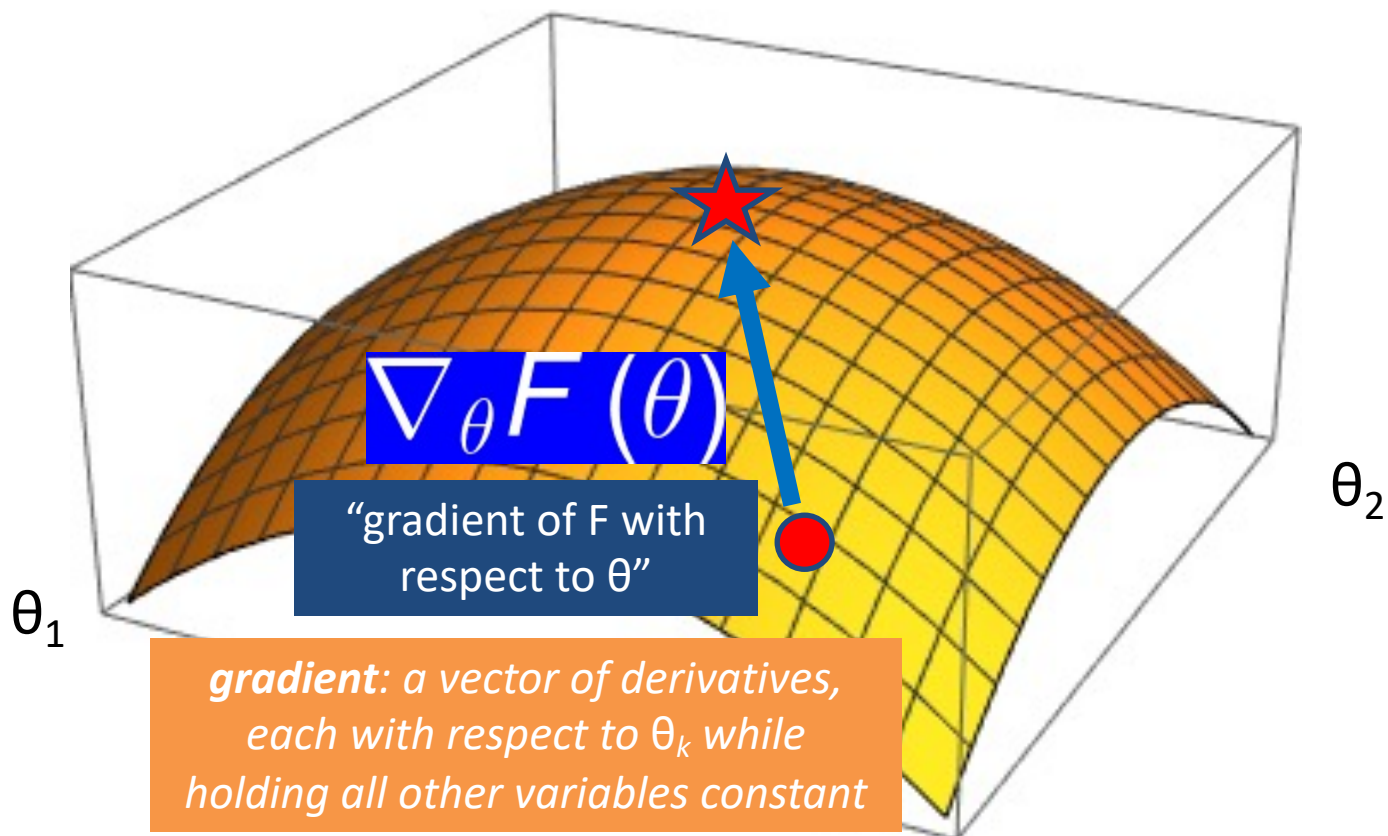
Gradient Ascent

$$\arg \max_{\theta} F(\theta)$$



Gradient Ascent

$$\arg \max_{\theta} F(\theta)$$



In many cases, pick your toolkit

PyTorch
Huggingface
TensorFlow
Deeplearning4j
DyNet
Caffe

Keras
MxNet
Gluon
CNTK
...

Comparisons:

https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

<https://deeplearning4j.org/compare-dl4j-tensorflow-pytorch>

<https://github.com/zer0n/deepframeworks> (older---2015)

My Hope

Help you learn the ropes...



<https://raftinginthesmokies.com/wp-content/uploads/2019/02/ropes-challenge-course.jpeg>

My Hope

Help you learn the ropes...



My Hope

Help you learn the ropes...



... so you can go
into a job...

My Hope

Help you learn the ropes...



... and apply your knowledge using whatever tools your org. uses!



... so you can go into a job...

```
from theano import *
```

keras
torch

What I actually do

Toolkit Basics

- Machine learning involves working with data
 - analyzing, manipulating, transforming, ...
- More often than not, it's numeric or has a natural numeric representation
- Natural language text is an exception, but this too can have a numeric representation
- A common data model is as a N-dimensional matrix or tensor
- These are supported in Python via libraries

Typical Python Libraries

numpy, scipy

- Basic mathematical libraries for dealing with matrices and scientific/mathematical functions

pandas, matplotlib

- Libraries for data science & plotting

sklearn (scikit-learn)

- A whole bunch of implemented classifiers

torch (pytorch) and tensorflow

- Frameworks for building neural networks

Lots of
documentation
available for all
of these online!

Today's Learning Goals

- NLP vs. CL
- Terminology:
 - NLP: vocabulary, token, type, one-hot encoding, dense embedding, parameter/weight, corpus/corpora
 - Linguistics: lexeme, morphology, syntax, semantics, “discourse”
- Universal Dependencies

