Natural Language Processing

AI51701/CSE71001

Lecture 1

08/29/2023

Instructor: Taehwan Kim

Lecture Plan

- ❑ Introduction to the course
- ❑ Examples of Natural Language Processing
- ❑ Word

Course Information

- Instructor: Taehwan Kim (taehwankim@unist.ac.kr, Machine Learning, Vision, and Language Lab, https://sites.google.com/view/mvllab)
  - Office hours: by appointments

- We have excellent TAs:
  - Jaeyeon Bae (qowodussla@unist.ac.kr)
  - Seonghee Han (seonghee@unist.ac.kr)

Course Information

- Will be taught offline
  - Time: Tue/Thu at 4:00-5:15pm
  - Place: Bldg. 106, T205
  - Will use **mobile attendance system**
    - If you miss more than ¼ of all classes, you will be automatically failed in this course (university policy).
      - Exceptions to count as being absent: military training, family occasion (wedding & funeral), medical reasons, and attending events approved by university president (university policy, all require document proof)
    - > 10 min. late: counted as being late
    - > 30 min. late: absence
    - 2 lateness = 1 absence

Grading

- Assignments (total 50%)
  - Around 4 assignments including problem solving and (mostly) programming in Python.
- Attendance Quizzes (total 10%)
- Final Project (40%)

Grading

- Attendance Quizzes (total 10%)
  - Will be available via *Quizzes & Exams* in BlackBoard (the left panel), so bring your laptops/tablets.
  - Will be open for five minutes during the lecture.
  - Two lowest scores will be excluded when grading and use this for emergencies.

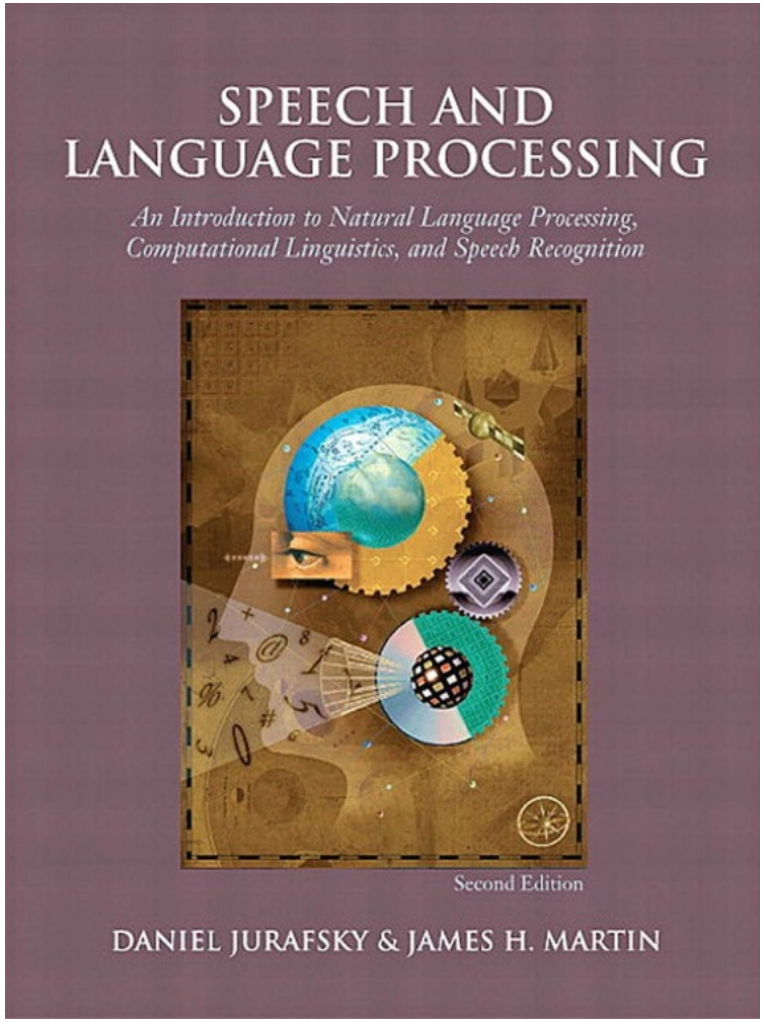
Grading

- Final Project (40%)
  - Proposal (5%): presentation on Oct. 10 & Oct. 12 (tentative), feedback will be given.
  - Final presentation (5%): during the final week
  - Final report (30%): by the end of last week of the semester
  - You will choose your project topics.
  - 3 people team project.
  - More detail will be provided soon.

Optional Textbooks

- Dan Jurafsky and James H. Martin, "Speech and Language Processing, 2nd Edition", Prentice Hall, 2009.
  - Third edition draft is available at https://web.stanford.edu/~jurafsky/slp3/

- Goldberg. *Neural Network Methods for Natural Language Processing*.
  - Earlier draft (from 2015) available online



Prerequisites

- ❑ Proficiency in Python
  - ▪ All class assignments will be in Python
  - ▪ For deep learning, we use PyTorch (a refresher may be given later)
- ❑ Multivariate Calculus, Linear Algebra
- ❑ Basic Probability and Statistics
- ❑ Fundamentals of Machine Learning
  - ▪ Loss functions, Taking simple derivatives, Performing optimization with gradient descent

Using BlackBoard

- URL: https://blackboard.unist.ac.kr/
- Lecture materials
  - Lecture slides will be posted before the lecture

- Discussions
  - You can post questions related to the course such as assignments.
  - The instructor and TAs will try to respond soon.
  - Other students are welcome to provide answers as well.

- Assignments
  - You can check the posted assignments and may submit your assignment here

Lateness Policy

- You may submit your assignments up to three days after the deadline:
  - You will get 10% penalty for each late day.
  - After three late days, no late submission will be accepted.

- Appeals to your graded assignments
  - Should be done via documents.
  - Detail will be announced later.

Cheating and Collaboration

- What is your goal in this course?
  - You want to learn and cheating will not help.

- Your assignment submissions should be your OWN work.
  - You are welcome to discuss assignments with others in the course, but written solutions and code must be written individually.
  - Do not search or copy solutions in textbook/internet.
  - We will check plagiarism of your code by using automatic tools.
  - If cheating happens, you will get a penalty
    - Minimum 0 point and -10% in your final grade, up to failure in this course and reporting to the dean

What is Natural Language Processing?

- Automating the analysis, generation of human (“natural”) language
  - Analysis (or “understanding” or “processing” …)
  - Generation

- An experimental computer science research area that includes problems and solutions pertaining to the understanding of human language

What does it mean to understand a language?

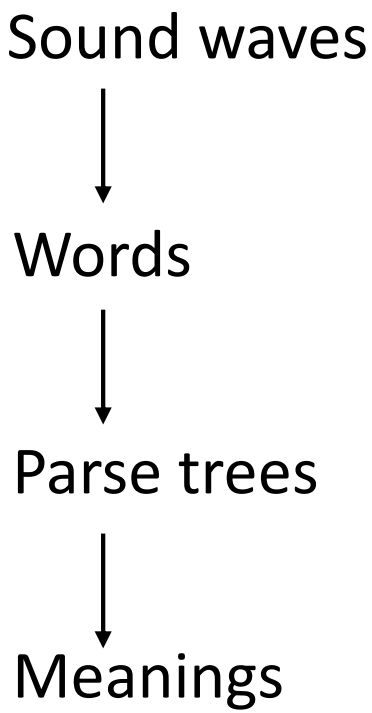
- "Stop"
- "Turn it up"
- "Volume level 6"
- "Repeat that"
- "What can you do?"

- "Play some music"
- "Play music by [artist]"
- "Play dance music on YouTube"
- "Play KEXP radio on TuneIn"
- "Play the latest episode of Radiolab"
- "Pause"
- "Next song"

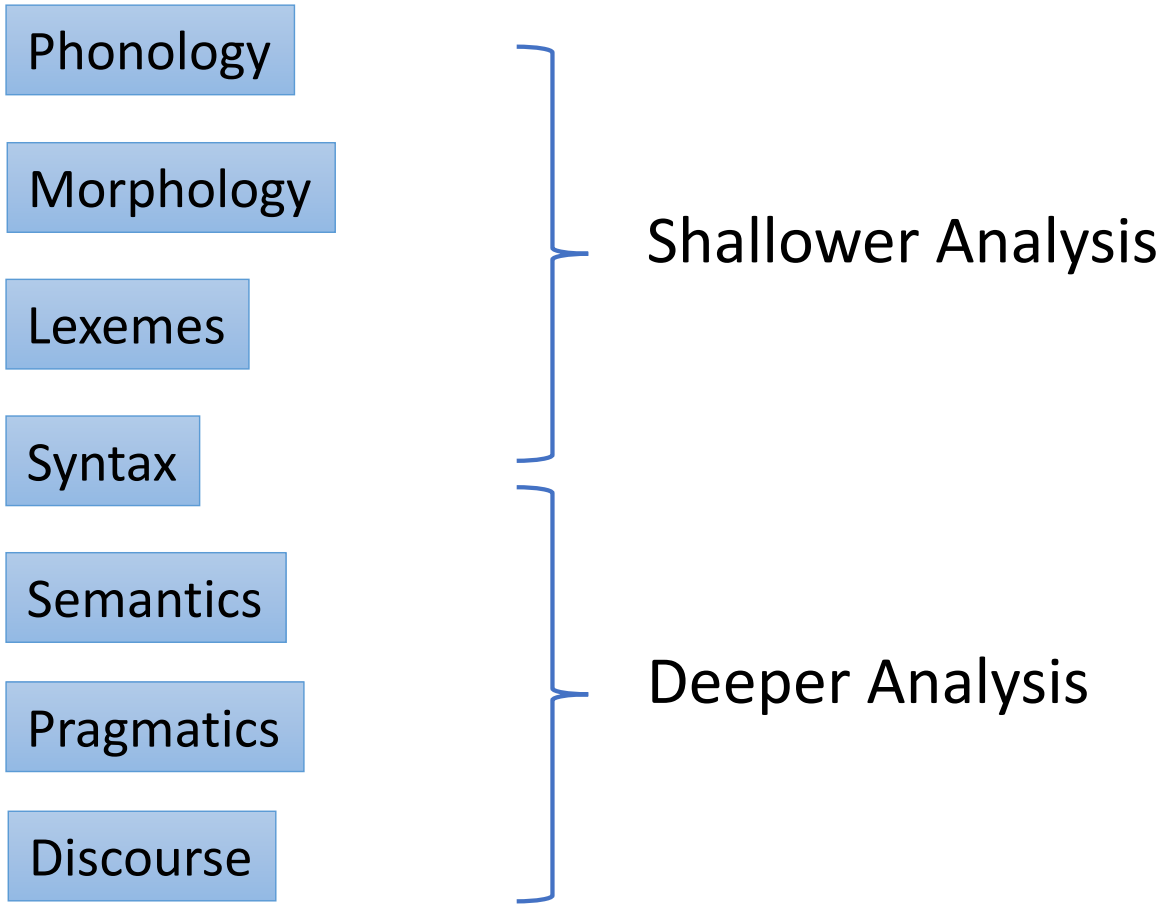
- "When's my first appointment tomorrow?"
- "Wake me up at 6am tomorrow"
- "Tell me about my day"
- "How long will it take to get to work?"
- "What's the weather today?"

What does it mean to understand a language?

- Phonology
- Morphology
- Lexemes
- Syntax
- Semantics
- Pragmatics
- Discourse



What does it mean to understand a language?



Syntax, Semantics, Pragmatics

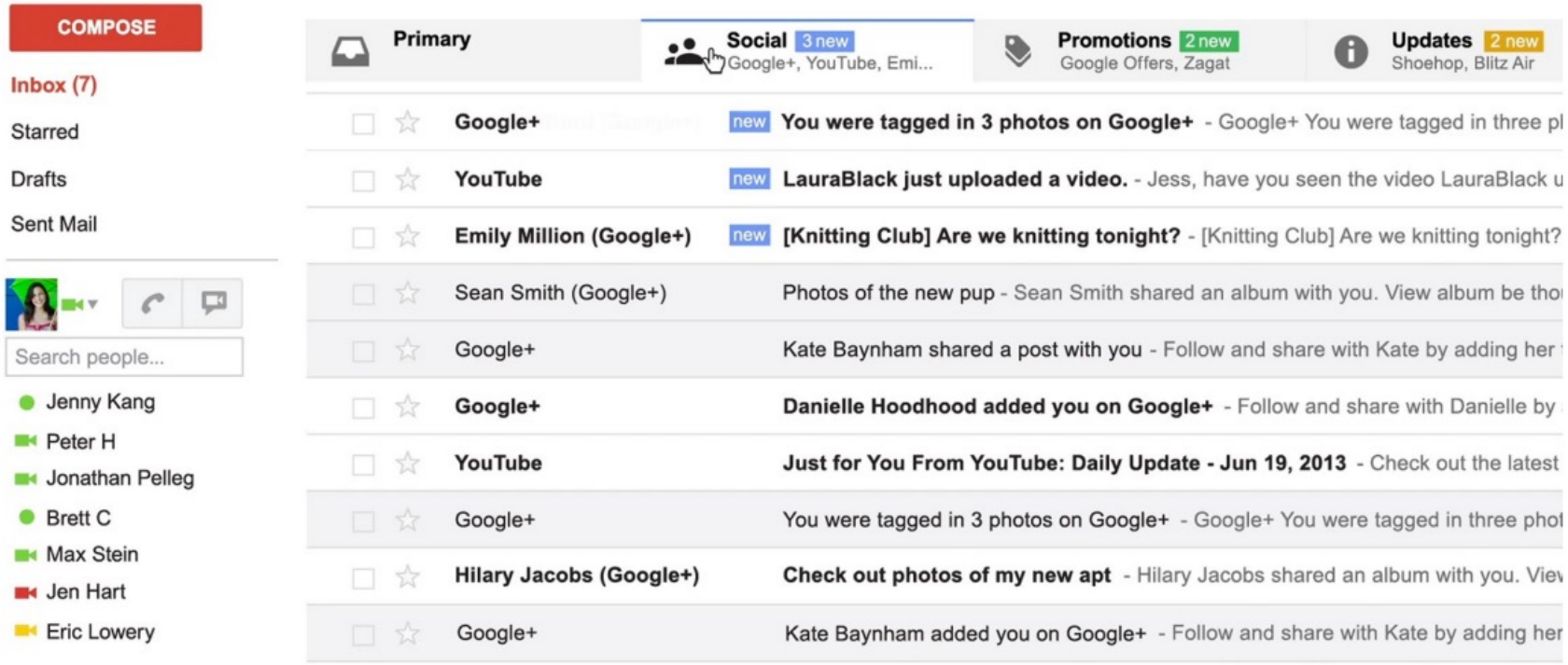
- **Syntax** concerns the proper ordering of words and its affect on meaning.
  - The dog bit the boy.
  - The boy bit the dog.
  - Bit boy dog the the.
- **Semantics** concerns the (literal) meaning of words, phrases, and sentences.
  - “plant” as a photosynthetic organism
  - “plant” as a manufacturing facility
  - “plant” as the act of sowing

Syntax, Semantics, Pragmatics

- **Pragmatics** concerns the overall communicative and social context and its effect on interpretation.
  - Honest or dishonest?
  - Context: Kyle and Ellen would like to see a movie. Kyle has $20 in his pocket. Tickets cost $8 each.
  - Kyle: “I have $8.”

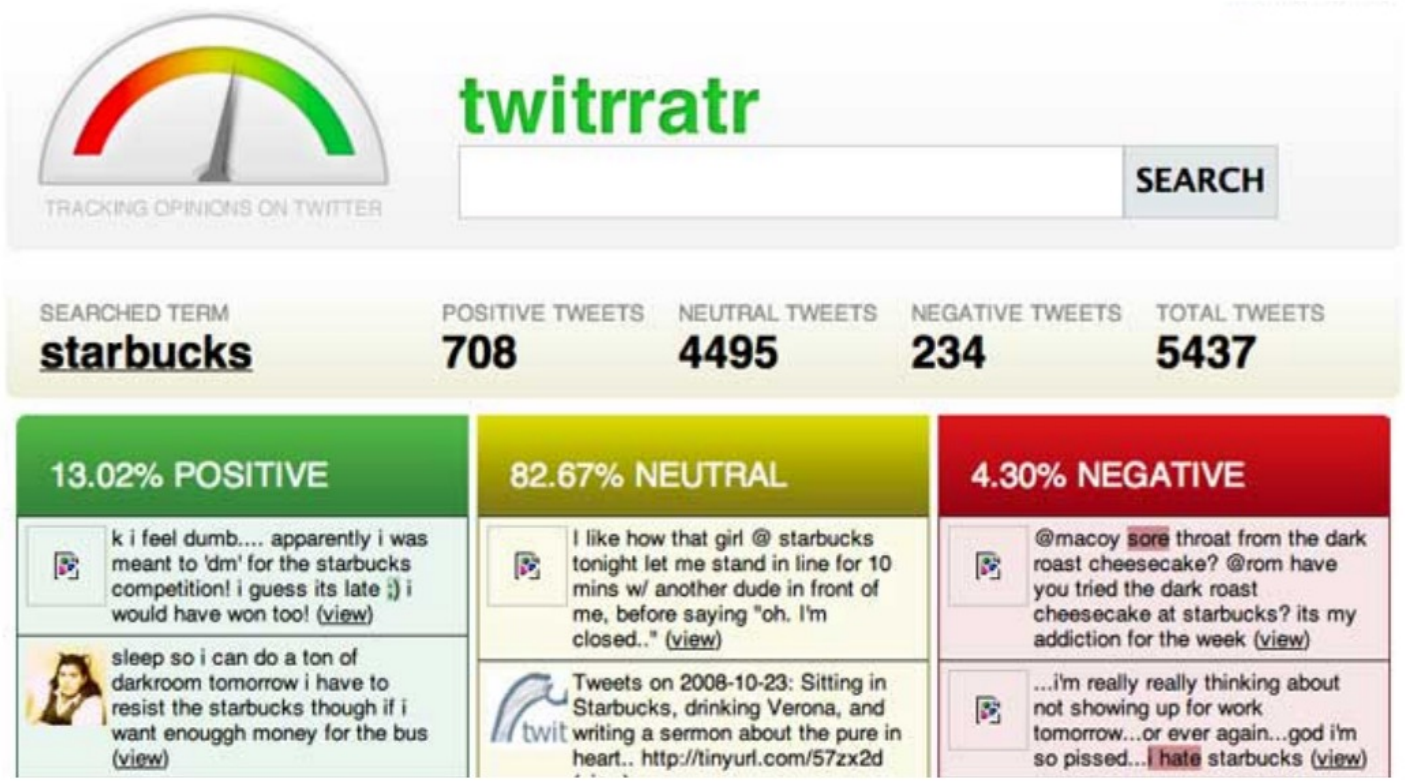
Examples of Natural Language Processing

- Text Classification



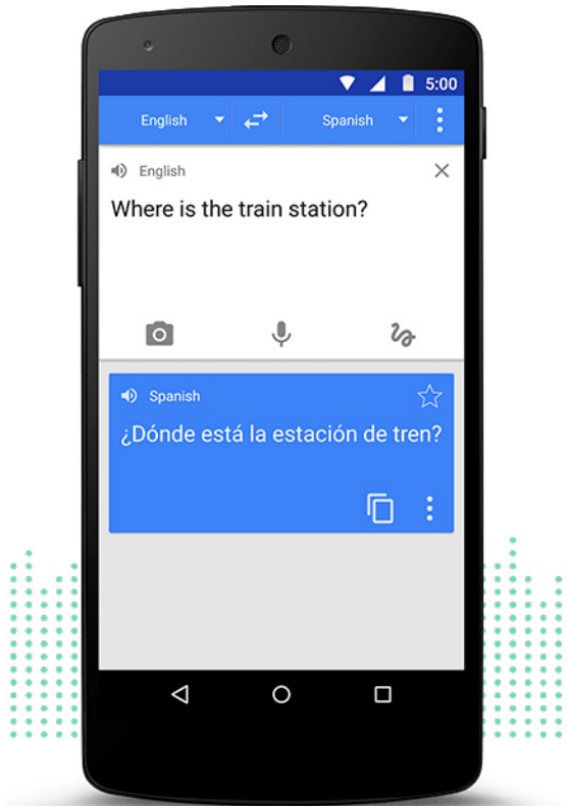
Examples of Natural Language Processing

- Sentiment Analysis



Examples of Natural Language Processing

- Machine Translation



Examples of Natural Language Processing

- Question Answering

Examples of Natural Language Processing

- Dialogue Systems

Examples of Natural Language Processing

- Summarization

Examples of Natural Language Processing

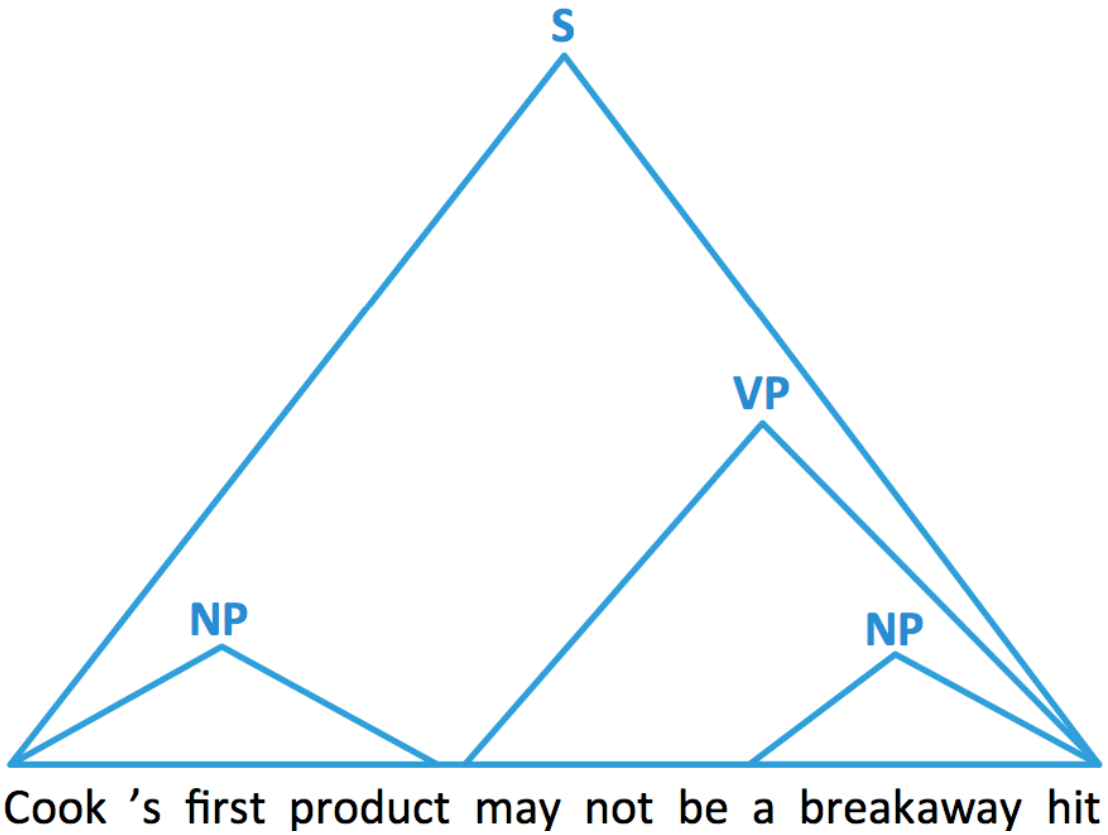
- Part-of-Speech Tagging

			proper	proper			
determiner	verb (past)	prep.	noun	noun	poss.	adj.	noun
Some	questioned	if	Tim	Cook	's	first	product

						proper	
modal	verb	det.	adjective	noun	prep.	noun	punc.
would	be	a	breakaway	hit	for	Apple	.

Examples of Natural Language Processing

- Syntactic Parsing



Examples of Natural Language Processing

- Named Entity Recognition

Some questioned if Tim Cook's first product would be a breakaway hit for Apple.

PERSON

ORGANIZATION

Examples of Natural Language Processing

- Coreference Resolution

Some questioned if Tim Cook's first product would be a breakaway hit for Apple.

It's the company's first new device since he became CEO.

Examples of Natural Language Processing

- Reading Comprehension

Once there was a boy named Fritz who loved to draw. He drew everything. In the morning, he drew a picture of his cereal with milk. His papa said, “Don’t draw your cereal. Eat it!”

After school, Fritz drew a picture of his bicycle. His uncle said, “Don't draw your bicycle. Ride it!”

…

What did Fritz draw first?

A) the toothpaste

B) his mama

C) cereal and milk

D) his bicycle

Examples of Natural Language Processing

- Sentence Similarity

Input	Output
Other ways are needed. We must find other ways.	4.4
Pakistan bomb victims' families end protest Pakistan bomb victims to be buried after protest ends	2.6
I absolutely do believe there was an iceberg in those waters. I don't believe there was any iceberg at all anywhere near the Titanic.	1.2

Examples of Natural Language Processing

- Word Prediction

> he bent down and searched the large container, trying to find anything else hidden in it other than the _____

Other Language Technologies

- Speech processing
- Information retrieval / web search
- Knowledge representation / reasoning
- Multimodal: Image-to-text (image/video captioning, VQA)

Related Fields

- Artificial Intelligence
- Machine Learning
- Linguistics
- Cognitive science
- Logic
- Data science
- Political science
- Education
- Economics
- ...many more

Computational Linguistics vs. Natural Language Processing

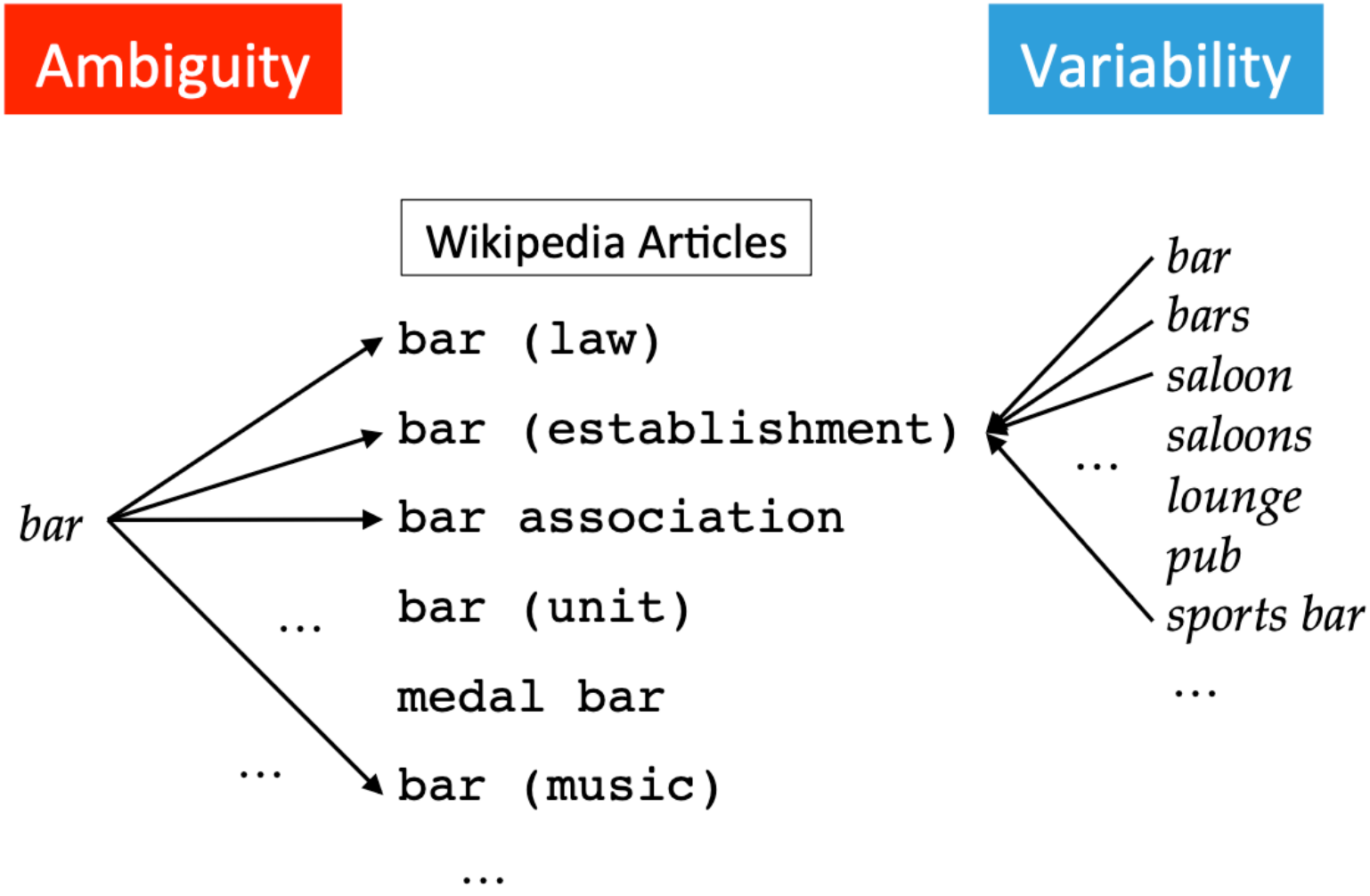
- Many people think of the two terms as synonyms
- Computational linguistics is about **linguistics**
  - Computational linguistics is more inclusive; more likely to include sociolinguistics, cognitive linguistics, and computational social science
- Natural language processing is about **processing**
  - NLP is more likely to use machine learning and involve engineering / system-building

Why is NLP hard?

- Ambiguity and variability of linguistic expression:
  - **Variability**: many forms can mean the same thing
  - **Ambiguity**: one form can mean many things

- Many different kinds of variability and ambiguity
- Each NLP task must address distinct kinds

Examples: Hyperlinks in Wikipedia



Why is Language Ambiguous?

- Having a unique linguistic expression for every possible conceptualization that could be conveyed would make language overly complex and linguistic expressions unnecessarily long.
- Allowing resolvable ambiguity permits shorter linguistic expressions, i.e. data compression.
- Language relies on people's ability to use their knowledge and inference abilities to properly resolve ambiguities.
- Infrequently, disambiguation fails, i.e. the compression is lossy.

More difficulties: Non-standard language

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either❤️

- And neologisms:
  - Unfriend
  - Retweet
  - bromance

Learning-based NLP

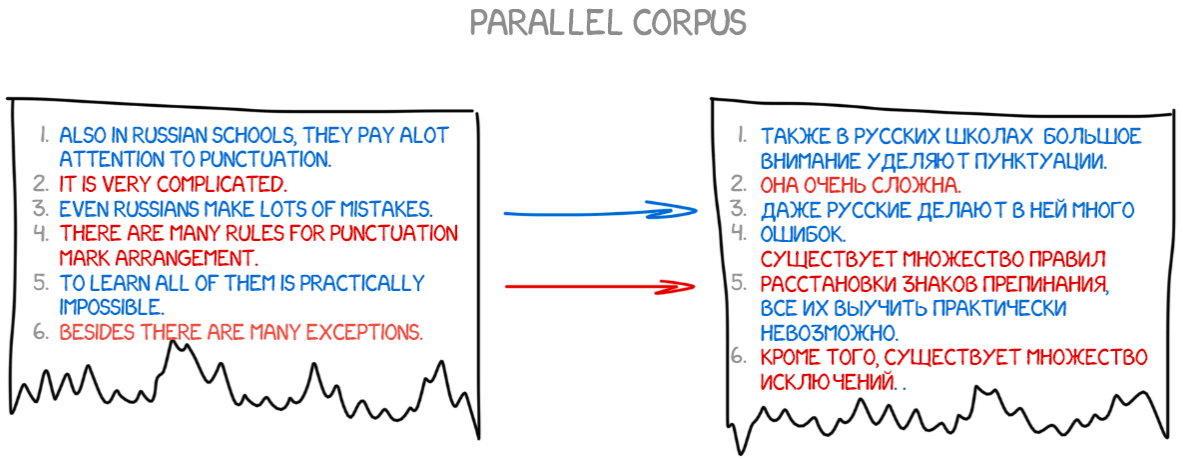
- Option 1: Write Rules
  - Rules for translating **much** or **many** into Russian:

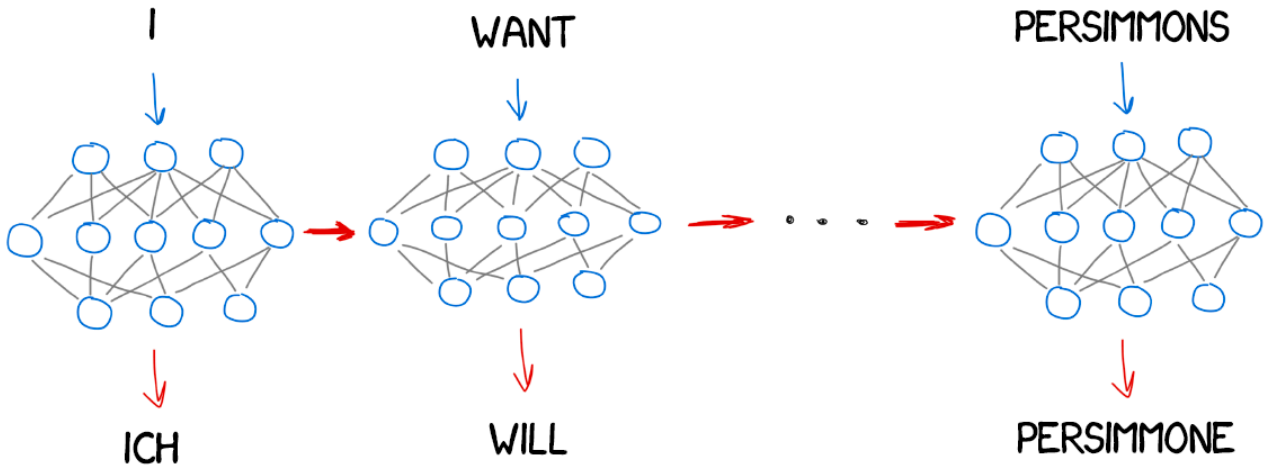
if preceding word is *how* **return** *skol'ko*
else if preceding word is *as* **return** *stol'ko zhe*
else if word is *much*
    if preceding word is *very* **return** nil
    else if following word is a noun **return** *mnogo*
else (word is many)
    if preceding word is a preposition and following word is noun **return** *mnogii*
    else return *mnogo*

J & M (Ch 25); originally from Panov (1960)

Learning-based NLP

- Option 2: Learn from data





What is Deep Learning?

- **Deep learning** is a subfield of **machine learning**.
- Most machine learning methods work well because of **human-designed representations/input features**.
- Machine learning becomes just **optimizing feature weights to make a good prediction**.

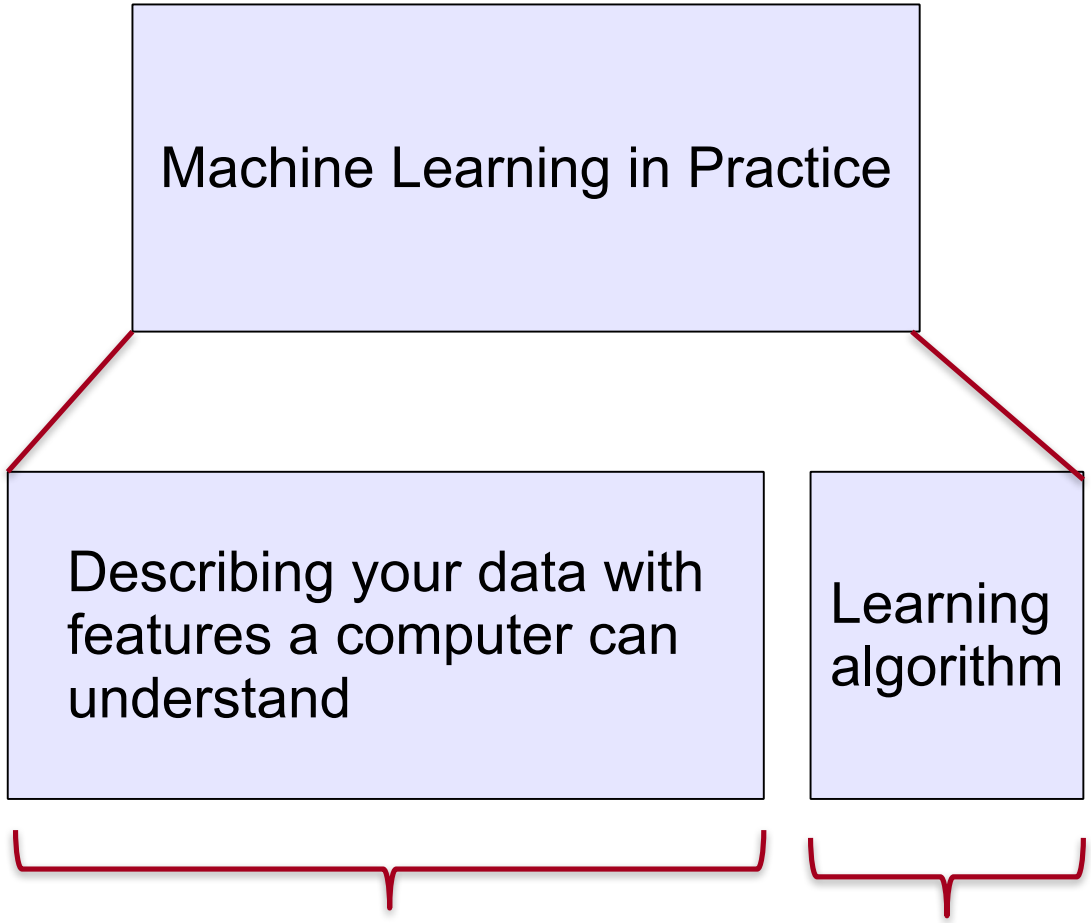
Feature	NER
Current Word	✓
Previous Word	✓
Next Word	✓
Current Word Character n-gram	all
Current POS Tag	✓
Surrounding POS Tag Sequence	✓
Current Word Shape	✓
Surrounding Word Shape Sequence	✓
Presence of Word in Left Window	size 4
Presence of Word in Right Window	size 4

Features for NER (Finkel et al., 2010)

Some questioned if Tim Cook's first product would be a breakaway hit for Apple.

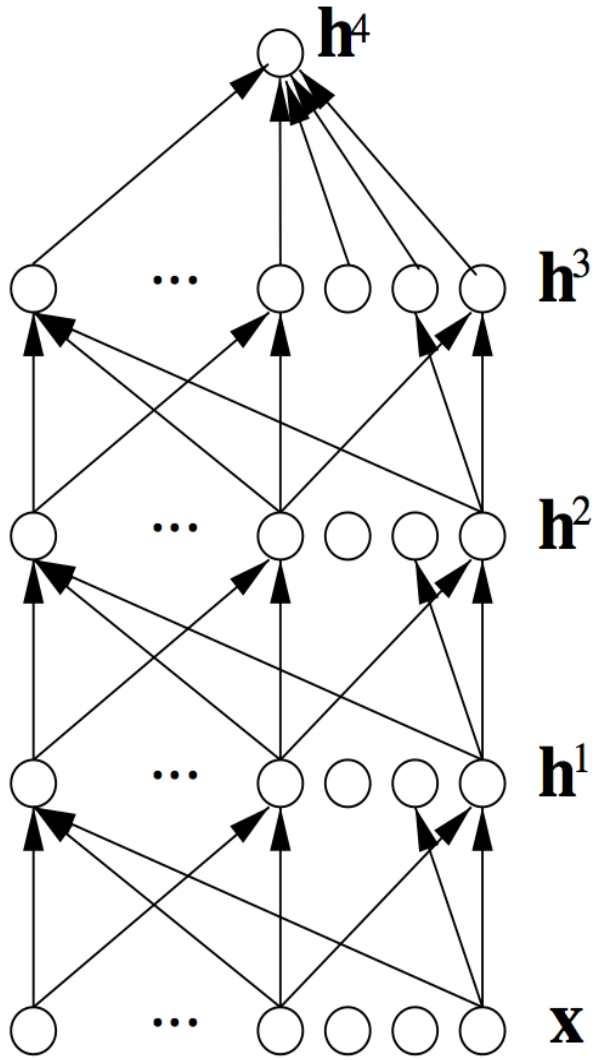
PERSON ORGANIZATION

Traditional Machine Learning



Deep Learning

- Representation learning attempts to automatically learn good features or representations
- Deep learning algorithms attempt to learn (multiple levels of) representations (here: h^1,h^2,h^3) and output (h^4)
- From “raw” inputs x (e.g. sound, pixels, characters, or words)



Stanford 224n

Why Deep Learning?

- Manually designed features are often over-specified, incomplete and take a long time to design and validate
- **Learned Features** are easy to adapt, fast to learn
- Deep learning provides a very flexible, learnable framework for **representing** world, visual and linguistic information.
- Deep learning can learn in **unsupervised** (from raw text) and **supervised** (with specific labels like positive/negative) settings

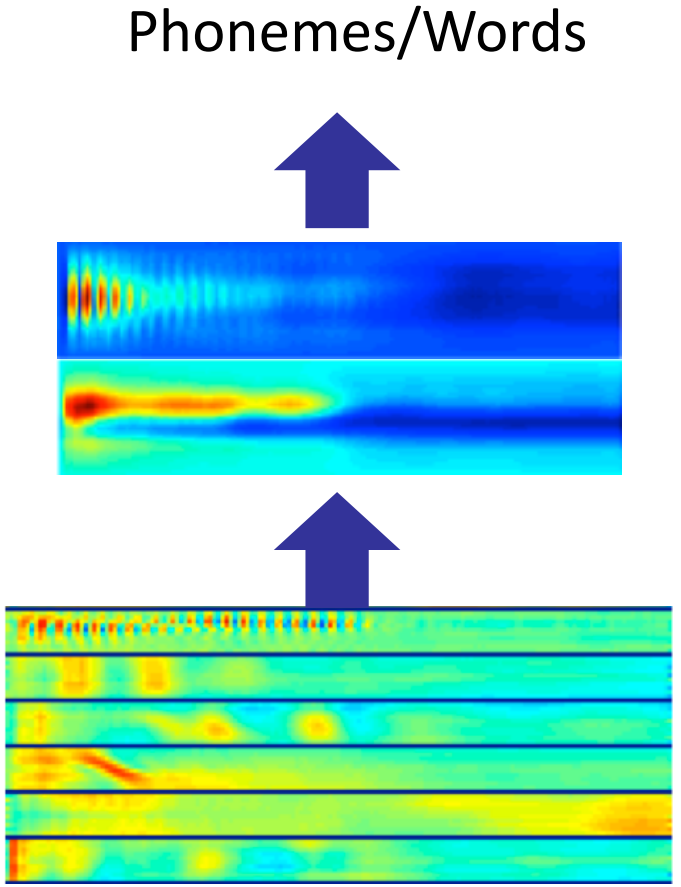
Why Deep Learning?

- In ~2010 **deep** learning techniques started outperforming other machine learning techniques. Why this decade?
  - Large amounts of training data favor deep learning
  - Faster machines and multicore CPU/GPUs favor deep learning
- New models, algorithms, ideas
  - Better, more flexible learning of intermediate representations
  - Effective end-to-end joint system learning
  - Effective learning methods for using contexts and transferring between tasks
  - Better regularization and optimization methods
- **Improved performance** (first in speech and vision, then NLP)

Deep Learning in Speech

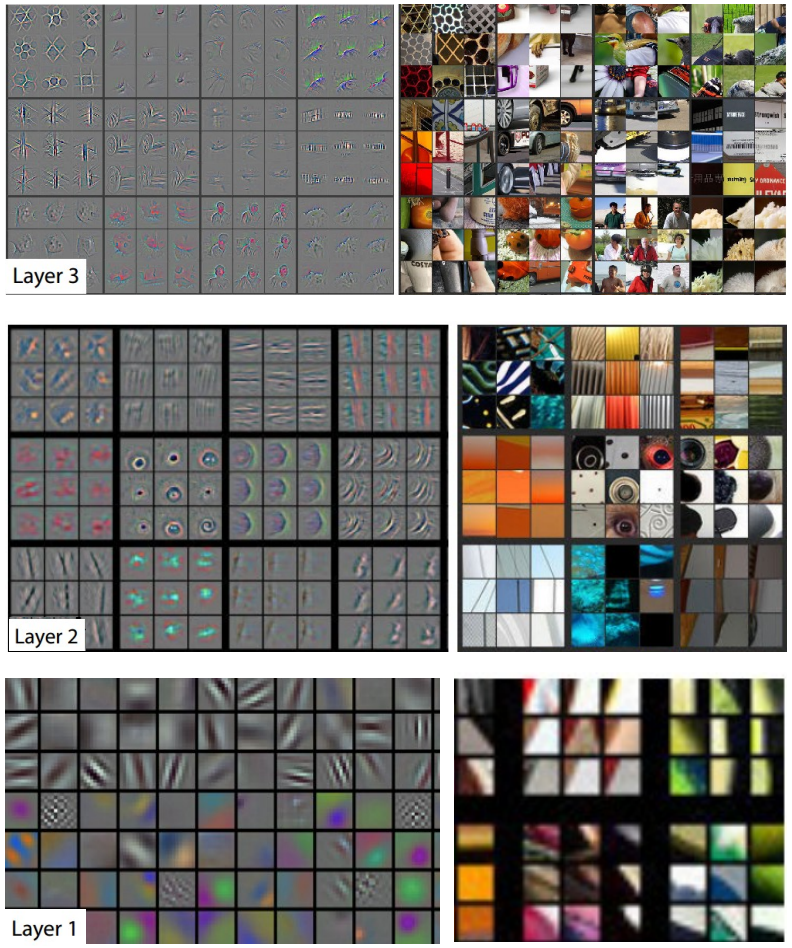
- The first breakthrough results of “deep learning” on large datasets happened in speech recognition
- Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition Dahl et al. (2010)

Acoustic model and WER	RT03S FSH	Hub5 SWB
Traditional features	**27.4**	**23.6**
Deep Learning	**18.5** (–33%)	**16.1** (–32%)



Deep Learning in Vision

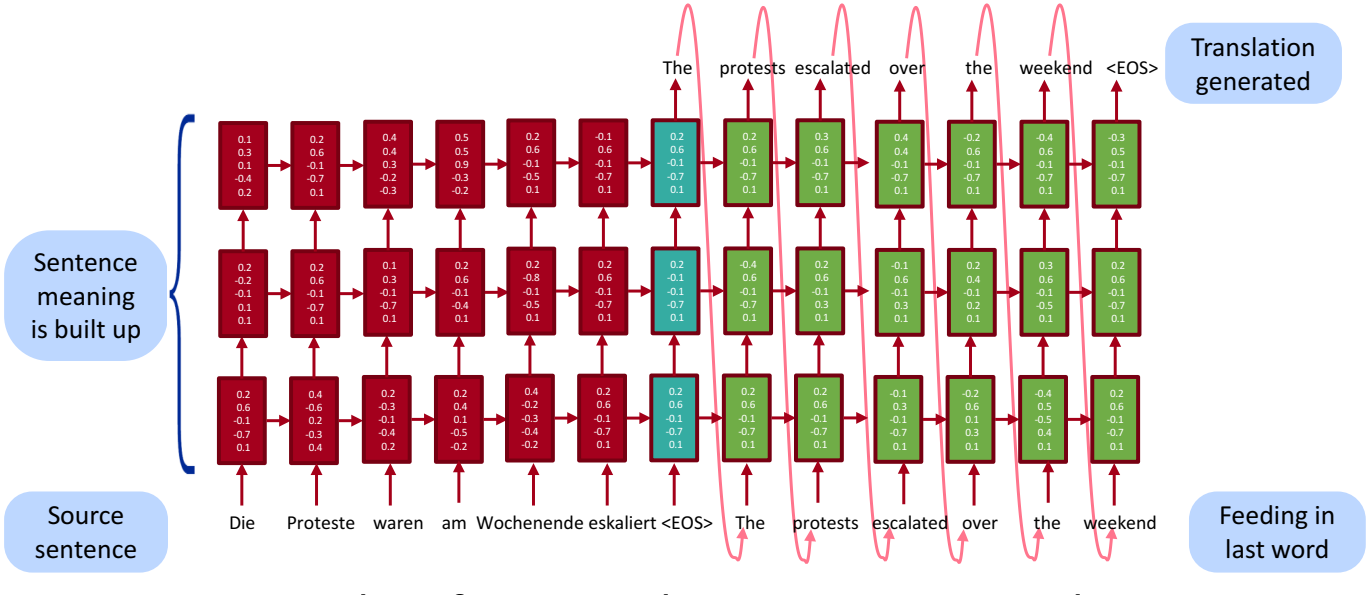
- First major focus of deep learning groups was computer vision
- The breakthrough DL paper: ImageNet Classification with Deep Convolutional Neural Networks by Krizhevsky, Sutskever, & Hinton, 2012, U. Toronto. 37% error red.



Zeiler and Fergus (2013)

Deep Learning in Language

- Machine Translation: first breakthrough in 2015



- Now live for many languages in Google Translate (etc.), with big error reductions!

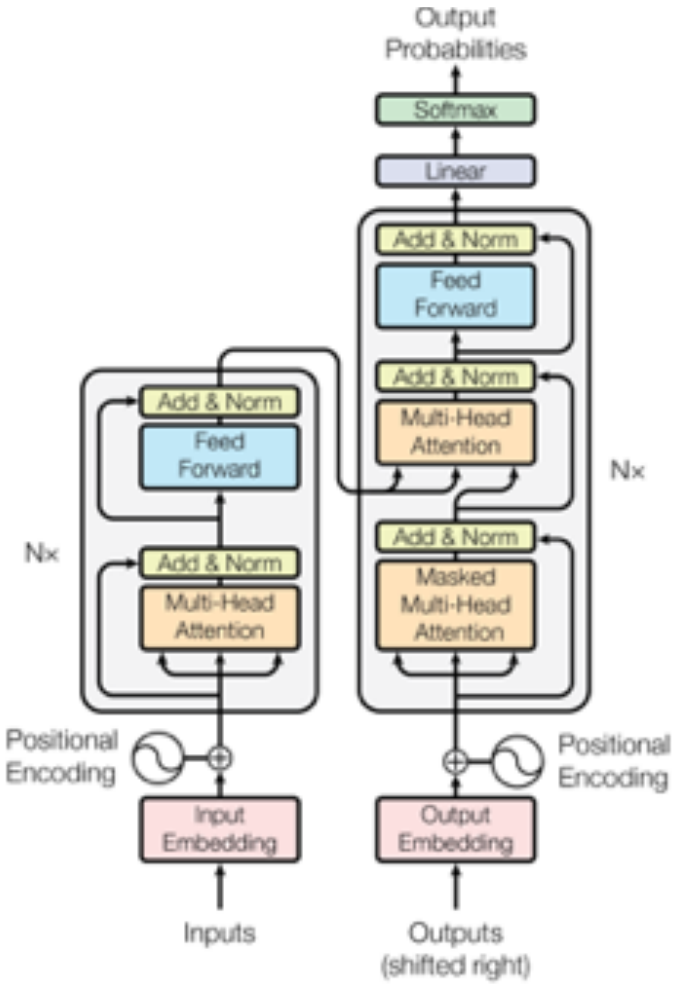


Figure 1: The Transformer - model architecture.

Recent Breakthrough

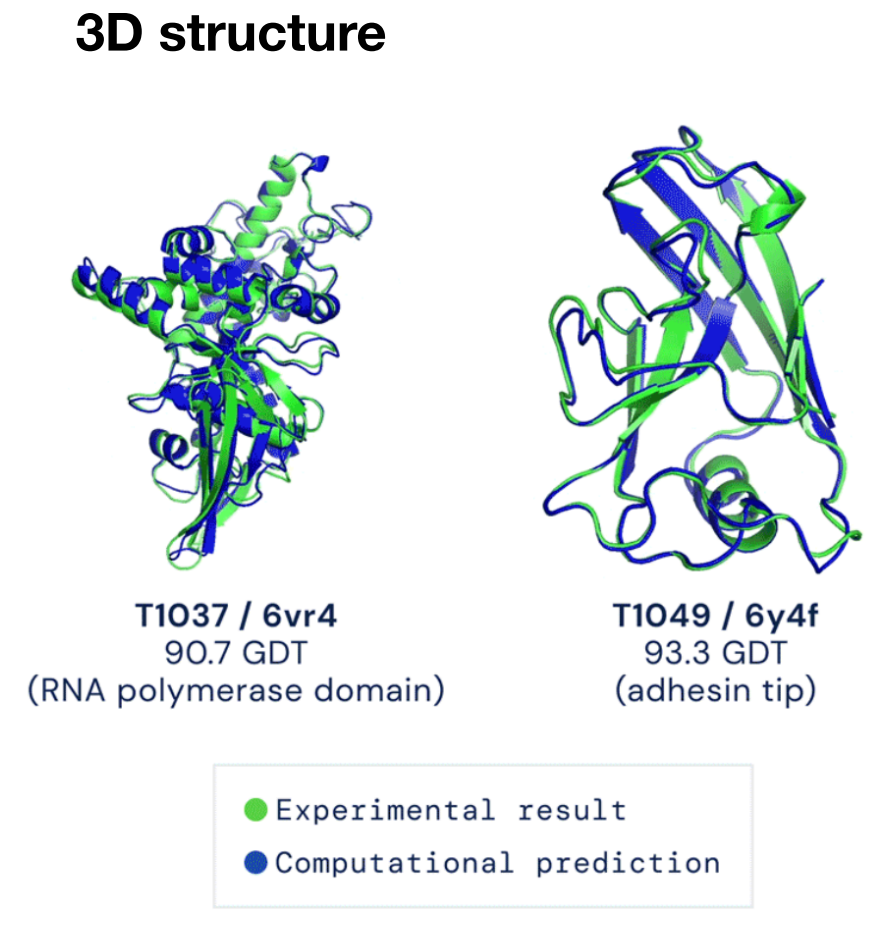
BLOG POST
RESEARCH

30 NOV 2020

AlphaFold: a solution to a 50-year-old grand challenge in biology

We have been stuck on this one problem – how do proteins fold up – for nearly 50 years. To see DeepMind produce a solution for this, having worked personally on this problem for so long and after so many stops and starts, wondering if we'd ever get there, is a very special moment.

PROFESSOR JOHN MOULT
CO-FOUNDER AND CHAIR OF CASP, UNIVERSITY OF MARYLAND



https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology

Tentative Topics to Cover

- ❑ Word and word vectors
- ❑ Neural networks basics
- ❑ Text classification
- ❑ Recurrent neural networks
- ❑ Language Models
- ❑ Dependency and syntactic parsing
- ❑ Part-of-speech tagging and sequence labeling
- ❑ Machine translation and attention
- ❑ Transformers and large language models
- ❑ Question answering
- ❑ Chatbots and dialogue systems
- ❑ Natural language generation

Word

Word

- Before learning NLP, you need to know about natural language
  - Similarly, to study computational biology, you need to know biology

- We will cover tokenization and morphology

Tokenization

- Tokenization: convert a character stream into words by adding spaces
  - For certain languages, highly nontrivial
  - E.g., Chinese word segmentation is a widely- studied NLP task
  - for other languages (English), tokenization is easier but is still not always obvious

Intricacies of Tokenization

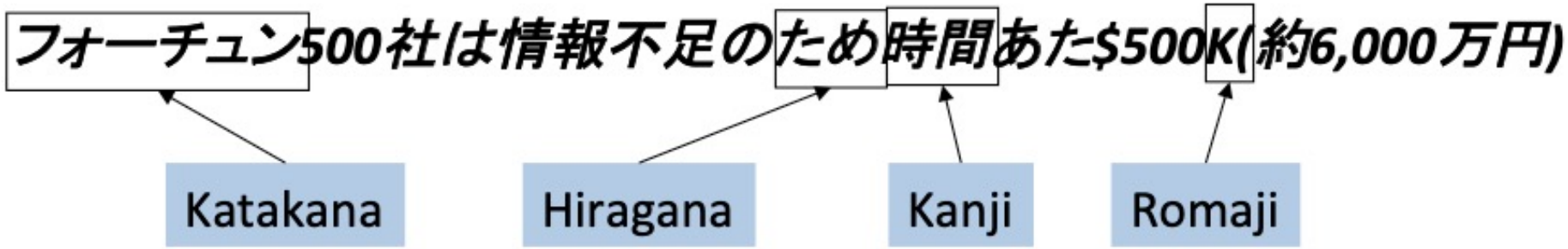
- Separating punctuation characters from words?
  - *, ” ? !* -> always separate
  - . -> when shouldn’t we separate it?
    - *Dr., Mr., Prof., U.S., etc.*

- English contractions:
  - *isn’t*, *aren’t*, *wasn’t*,... -> *is n’t*, *are n’t*, *was n’t*,...
  - but how about these*: can’t, won’t -> ca n’t, wo n’t*
  - *ca* and *wo* are then different forms from *can* and *will*

Intricacies of Tokenization

- Chinese and Japanese: no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
  - Sharapova now lives in US southeastern Florida

- Further complicated in Japanese, with multiple alphabets intermingled
  - Dates/amounts in multiple formats

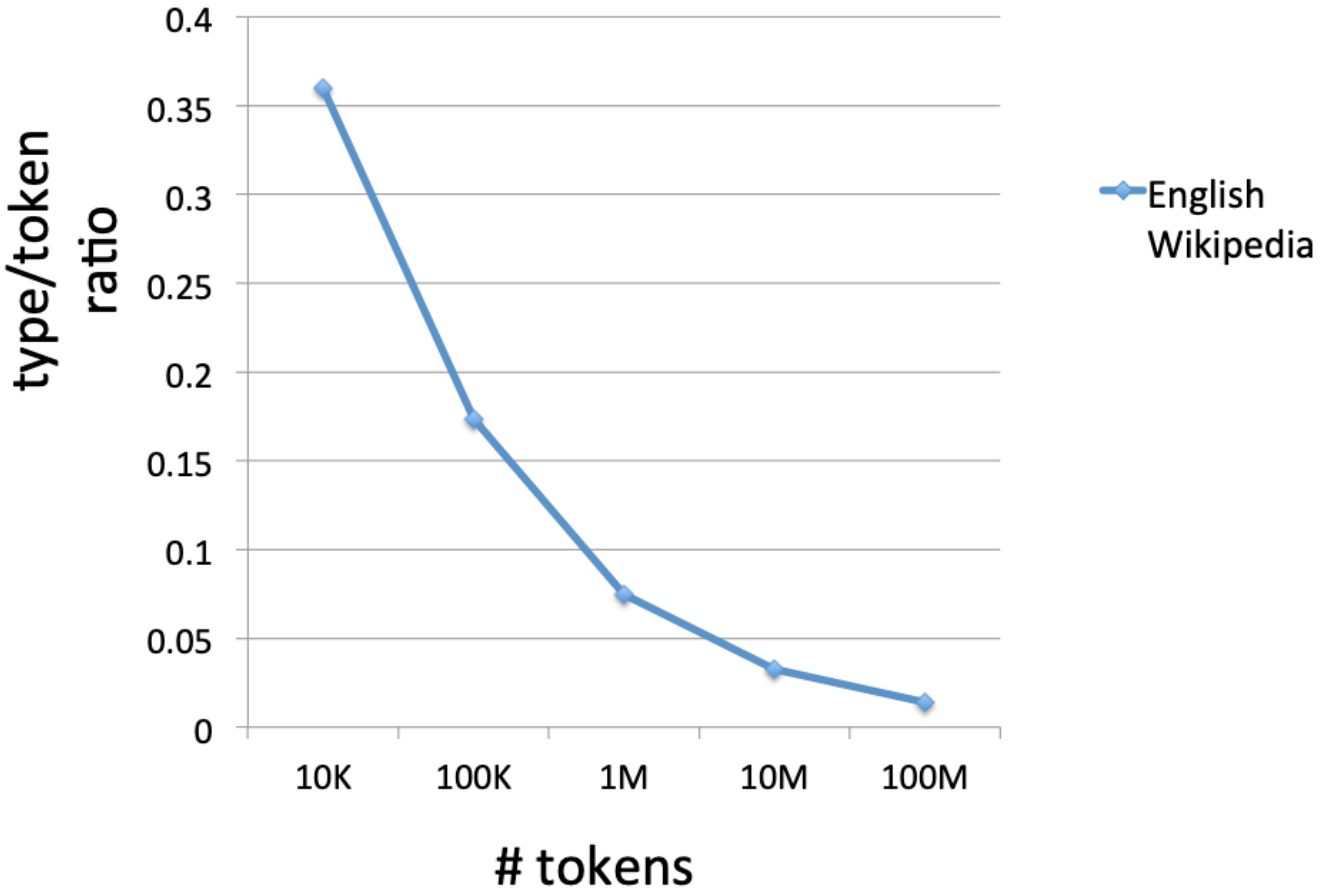


Types and Tokens

- Once text has been tokenized, let's count the words
- Types: entries in the vocabulary (vocabulary = set of types)
- Tokens: instances of types in a corpus
- Example sentence: *If they want to go , they should go .*
  - how many types?
  - how many tokens?
- Type/token ratio: useful statistic of a corpus (here, 0.8)
- As we add data, what happens to the type-token ratio?

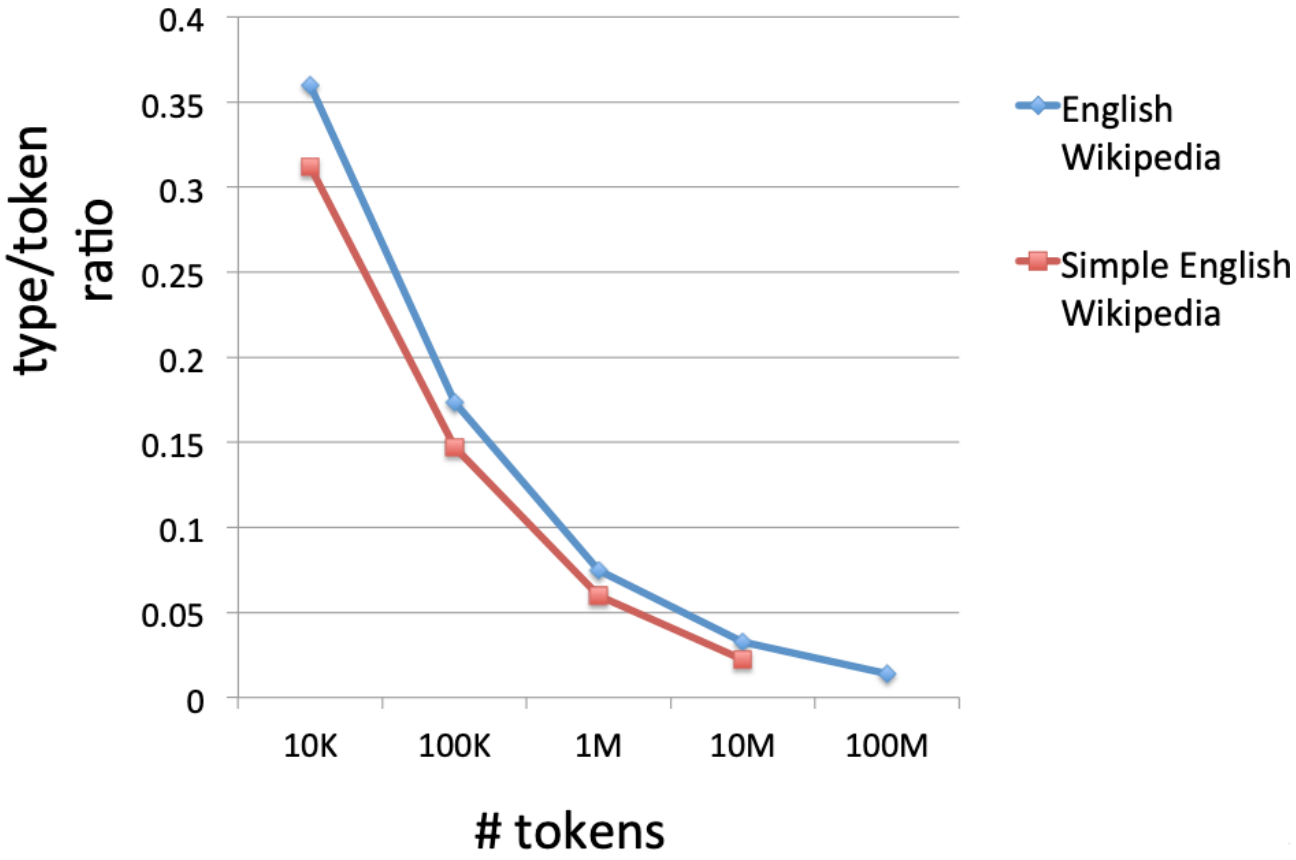
Types and Tokens

- More data -> Lower type/token ratio



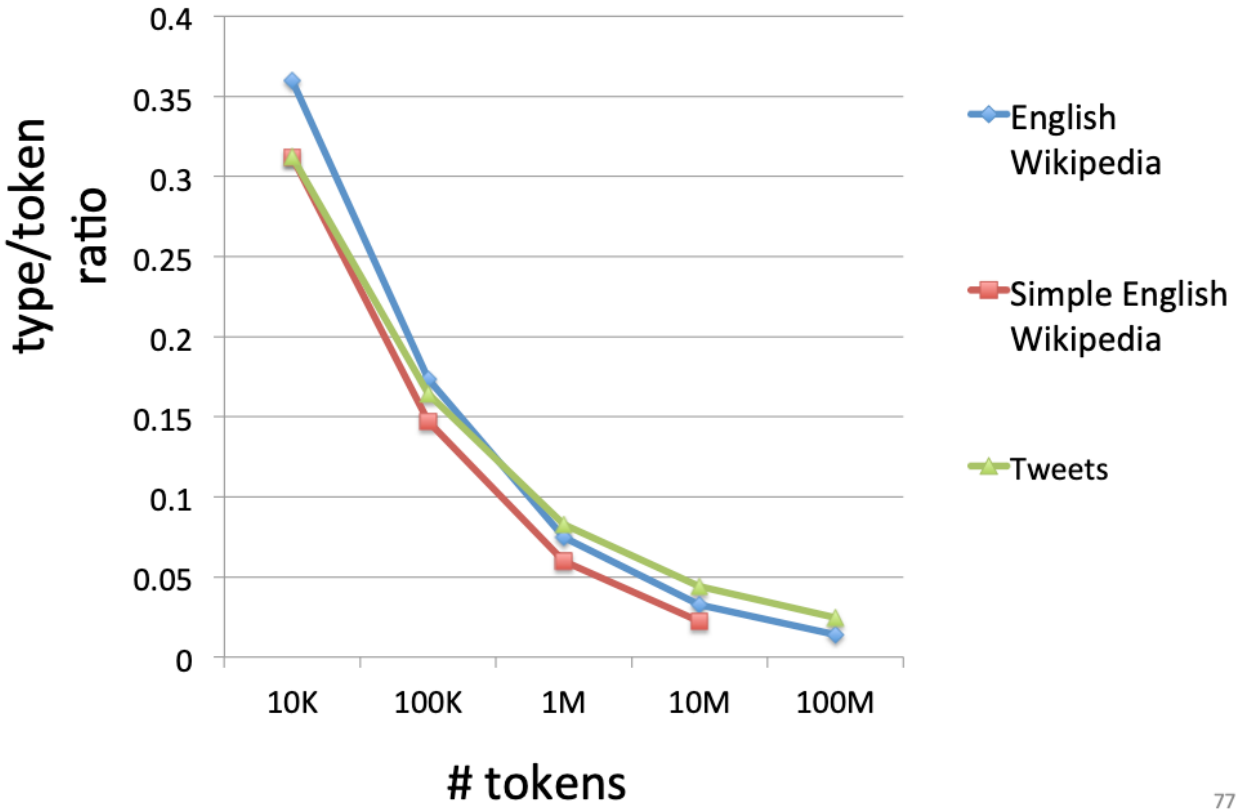
Types and Tokens

- What has a higher type/token ratio, Simple English Wikipedia or English Wikipedia?
  - type/token ratio is one measure of complexity



Types and Tokens

- Wikipedia vs Tweets?
  - Tweets (once you have 1 million or more tokens)



”really” on Twitter

```
224571 really
  1189 rly
  1119 realy
   731 rlly
   590 reallly
   234 realllly
   216 reallyy
   156 relly
   146 reallllly
   132 rily
   104 reallyyy
    89 reeeally
    89 realllllly
    84 reaaally
    82 reaally
    72 reeeeally
    65 reaaaally
    57 reallyyyy
    53 rilly
    50 realllllly
    48 reeeeeally
    41 reeally
    38 really2
    37 reaaaaally
    35 reallyyyyy
    31 reely
    30 realllyyy
    27 realllyy
    27 reaaly
    26 realllyyyy
    25 reallllllly
    22 reaaallly
    21 really-
    19 reeaally
    18 realllyyy
    16 reaaaallly
    15 realyy
    15 reallyreally
    15 realllyy
    15 reallllllllly
    15 reaallly
    14 reeeeeeally
    14 realllyyyy
    13 reeeaaally
    12 rreally
    12 reaaaaaally
    11 reeeeallly
    11 reeeallly
    11 realllllyyy
    11 reaallyy
    10 reallyreallyreally
    10 reaaaly
     9 reeeeeeeally
     9 reallys
     9 really-really
     9 r)eally
     8 reeeaally
     8 reallyyyyyyy
     8 reallyyyyyy
     8 realky
     7 relaly
     7 reeeeeeeeeally
     7 reeeealy
     7 reeeeaaally
     7 realllllllyyy
     7 reallllllllllly
     7 reaaaaaaally
     7 raelly
     7 r3ally
     6 r-really
     6 reeeaaalllyyy
     6 reeeaaallly
     6 reeeaaaally
     6 realyl
     6 r-e-a-l-l-y
     6 realllyyyyy
     6 reallllllllly
     6 reaaaaaallly
     5 rrrreally
     5 rrly
     5 rellly
     5 reeeeeeeeally
     5 reeeeaally
     5 reeeeaaallly
     5 reeallyyy
     5 reallllllllllly
     5 reallllllllllllly
     5 reaalllyy
     5 reaaaalllly
     5 reaaaaallly
     4 rllly
     4 reeeeeeeeeeally
     4 reeealy
     4 reeaaaally
     4 realllllyyyy
     4 reallllllyyyy
     4 reaalllyyy
     4 reaalllly
     4 reaaalllyy
     4 reaaallly
     4 reaaaaly
     3 reeeeeallllly
     3 reeeeallllly
     3 reeeeaaaaally
     3 reeeaallly
     3 reeeaaalllyyy
     3 reealy
     3 reeallly
     3 reeaaly
     3 reeaalllyyy
     3 reeaalllly
     3 reeaaallly
     3 reallyyyyyyyyy
     3 reallyl
```

”really” on Twitter

3 really)
3 r]eally
3 realluy
3 reallllyyyyy
3 reallllllyyyyyyy
3 reallllllyyyy
3 reallllllyy
3 realllllllllllllly
3 realiy
3 reaallyyyy
3 reaalllllly
3 reaaallyy
3 reaaaallyy
3 reaaaalllllly
3 reaaaaaly
3 reaaaaaaaally
3 r34lly
2 rrreally
2 rreeaallyy
2 rlyyyy
2 rlyyy
2 reqally
2 rellyy
2 rellys
2 reeely
2 reeeeeealy
2 reeeeeallly
2 reeeeeaally
2 reeeeeaaally
2 reeeeeaaalllllly
2 reeeeallyyy
2 reeeealllyyy
2 reeeeaaalllyyyy
2 reeeeaaalllly
2 reeeeaaaally
2 reeeeaaaalllyyy
2 reeeallyy
2 reeallyy
2 reeaallyy
2 reeaalllyy
2 reeaallly
2 reeaaally
2 reaqlly
2 realyyy
2 reallyyyyyyyyyyyy
2 reallyyyyyyyy
2 really*
2 really/
2 realllyyyyyy
2 reallllyyyyyy
2 reallllllyyyyy
2 realllllllyyyyy
2 reallllllyy
2 realllllllllllllly
2 realllllllllllllllly
1 rrrrrrrrrrrrrrrreeeeeeeeeeeeaaaaaaallllllyyyyyy
1 rrrrrrrrrreally
1 rrrrrrreeeeeeaaaallllyyyyyyy
1 rrrrrrealy
1 rrrrrreally
…
1 re-he-he-heeeeally
1 re-he-he-he-ealy
1 reheheally
1 reelllyy
1 reellly
1 ree-hee-heally
…
1 reeeeeeeeeaally
1 reeeeeeeeeaaally
1 reeeeeeeeeaaaaaalllyyy
1 reeeeeeeeeaaaaaaallllllllyyyyyyyy
1 reeeeeeeeeaaaaaaallllllllyyyyyyyy
1 reeeeeeeeeaaaaaaaallllllllyyyyyyyy
1 reeeeeeeeaaaaaaaallllyyyyyy

”really” on Twitter

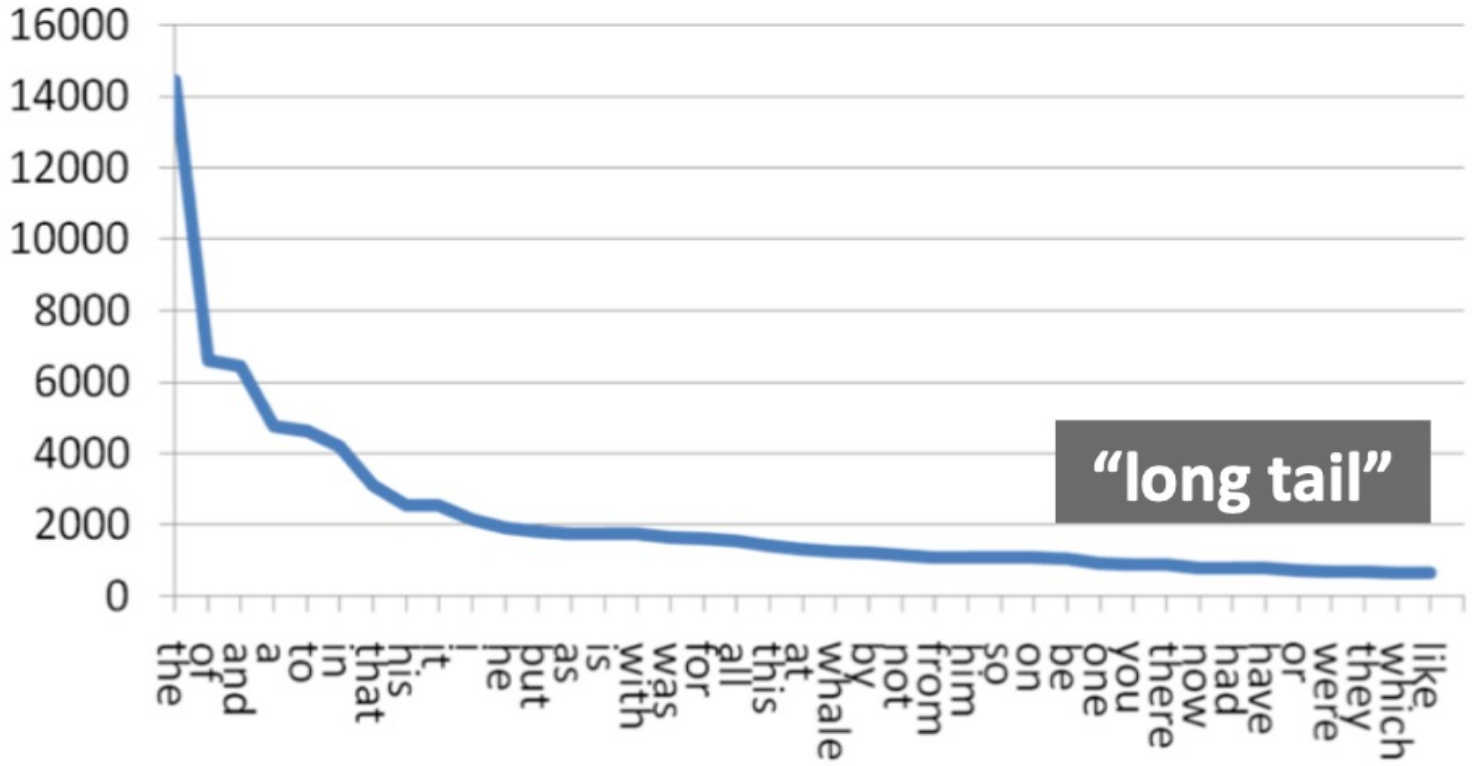
```
1 reallyreallyreallyreallyreallyreallyreallyreallyreallyreally
  reallyreallyreallyreallyreallyreallyreally
1 reallyreallyreallyreallyreallyr33lly
1 really/really/really
1 really(really
…
1 reallllllllyyyy
1 reallllllllyyyyyy
1 reallllllllyyyyy
1 reallllllllyyyy
1 reallllllllyyy
1 reallllllllyyyyy
1 reallllllllllyyyyyy
1 reallllllllllllllllllly
1 reallllllllllllllllllly
1 reallllllllllllllllllllyyyyy
1 reallllllllllllllllllllllly
1 reallllllllllllllllllllllly
1 reallllllllllllllllllllllllllly
1 reallllllllllllllllllllllllllllllllllllly
1 reallllllllllllllllllllllllllllllllllllllllllly
1 reallllllllllllllllllllllllllllllllllllllllllllllllllllll
  lllllllly
```

How many words are there?

- How many English words exist?
- When we increase the size of our corpus, what happens to the number of types?
  - A bit surprising: vocabulary continues to grow in any actual dataset
  - You'll just never see all the words
  - In 1 million tweets, 15M tokens, 600k types
  - In 56 million tweets, 847M tokens, 11M types

How are words distributed?

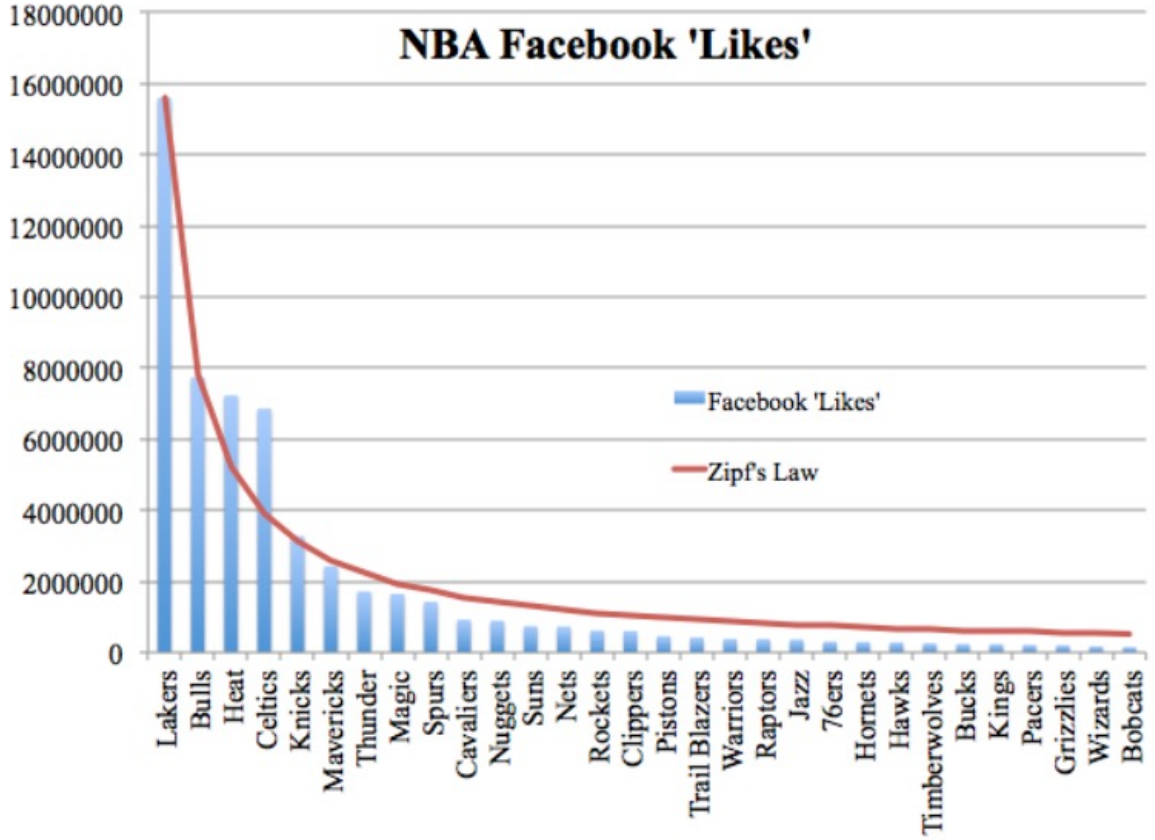
- Zipf's law: frequency of a word is inversely proportional to its rank



blog EMIS

Zipf's law

- Also predicts other kinds of data: populaIon of ciIes in a country, revenue of different companies, etc.



The LaurenIan University Sports AnalyIcs Group

Another option for text tokenization

- Instead of
  - white-space segmentation
  - single-character segmentation

- **Use the data** to tell us how to tokenize

- **Subword tokenization** (because tokens can be parts of words as well as whole words)

Subword tokenization

- Three common algorithms: ◦
  - Byte-Pair Encoding (BPE) (Sennrich et al., 2016)
  - Unigram language modeling tokenization (Kudo, 2018)
  - WordPiece (Schuster and Nakajima, 2012)
- All have 2 parts:
  - A token learner that takes a raw training corpus and induces a vocabulary (a set of tokens).
  - A token segmenter that takes a raw test sentence and tokenizes it according to that vocabulary

Byte Pair Encoding (BPE) token learner

- Let vocabulary be the set of all individual characters

  = {A, B, C, D,…, a, b, c, d….}

- Repeat:
  - Choose the two symbols that are most frequently adjacent in the training corpus (say 'A', 'B')
  - Add a new merged symbol 'AB' to the vocabulary
  - Replace every adjacent 'A' 'B' in the corpus with 'AB'.
- Until k merges have been done.

BPE token learner algorithm

function BYTE-PAIR ENCODING(strings C, number of merges k) **returns** vocab V

$V \leftarrow$ all unique characters in C # initial set of tokens is characters
for $i = 1$ **to** k **do** # merge tokens til k times
  $t_L, t_R \leftarrow$ Most frequent pair of adjacent tokens in C
  $t_{NEW} \leftarrow t_L + t_R$ # make new token by concatenating
  $V \leftarrow V + t_{NEW}$ # update the vocabulary
  Replace each occurrence of t_L, t_R in C with t_{NEW} # and update the corpus
return V

Byte Pair Encoding (BPE) Addendum

- Most subword algorithms are run inside space-separated tokens.
- So we commonly first add a special end-of-word symbol '__' before space in training corpus
- Next, separate into letters.

BPE token learner

Original (very fascinating🙄) corpus:

low low low low low lowest lowest newer newer newer newer newer newer wider wider wider new new

Add end-of-word tokens, resulting in this vocabulary:

vocabulary

_, d, e, i, l, n, o, r, s, t, w

BPE token learner

corpus	vocabulary
5 l o w _	_, d, e, i, l, n, o, r, s, t, w
2 l o w e s t _	
6 n e w e r _	
3 w i d e r _	
2 n e w _	

Merge e r to er

corpus	vocabulary
5 l o w _	_, d, e, i, l, n, o, r, s, t, w, er
2 l o w e s t _	
6 n e w er _	
3 w i d er _	
2 n e w _	

BPE

corpus		vocabulary
5	l o w _	_, d, e, i, l, n, o, r, s, t, w, er
2	l o w e s t _	
6	n e w er _	
3	w i d er _	
2	n e w _	

Merge er _ to er_

corpus		vocabulary
5	l o w _	_, d, e, i, l, n, o, r, s, t, w, er, er_
2	l o w e s t _	
6	n e w er_	
3	w i d er_	
2	n e w _	

BPE

corpus	vocabulary
5 l o w _	_, d, e, i, l, n, o, r, s, t, w, er, er_
2 l o w e s t _	
6 n e w er_	
3 w i d er_	
2 n e w _	

Merge n e to ne

corpus	vocabulary
5 l o w _	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne
2 l o w e s t _	
6 ne w er_	
3 w i d er_	
2 ne w _	

BPE

The next merges are:

Merge	Current Vocabulary
(ne, w)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new
(l, o)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo
(lo, w)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low
(new, er_)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_
(low, _)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_, low_

BPE token segmenter algorithm

- On the test data, run each merge learned from the training data:
  - Greedily
  - In the order we learned them
  - (test frequencies don't play a role)
- So: merge every e r to er, then merge er _ to er_, etc.
- Result:
  - Test set "n e w e r _" would be tokenized as a full word
  - Test set "l o w e r _" would be two tokens: "low er_

Properties of BPE tokens

- Usually include frequent words
- And frequent subwords
  - Which are often morphemes like -est or –er

- A **morpheme** is the smallest meaning-bearing unit of a language
  - unlikeliest has 3 morphemes un-, likely, and -est