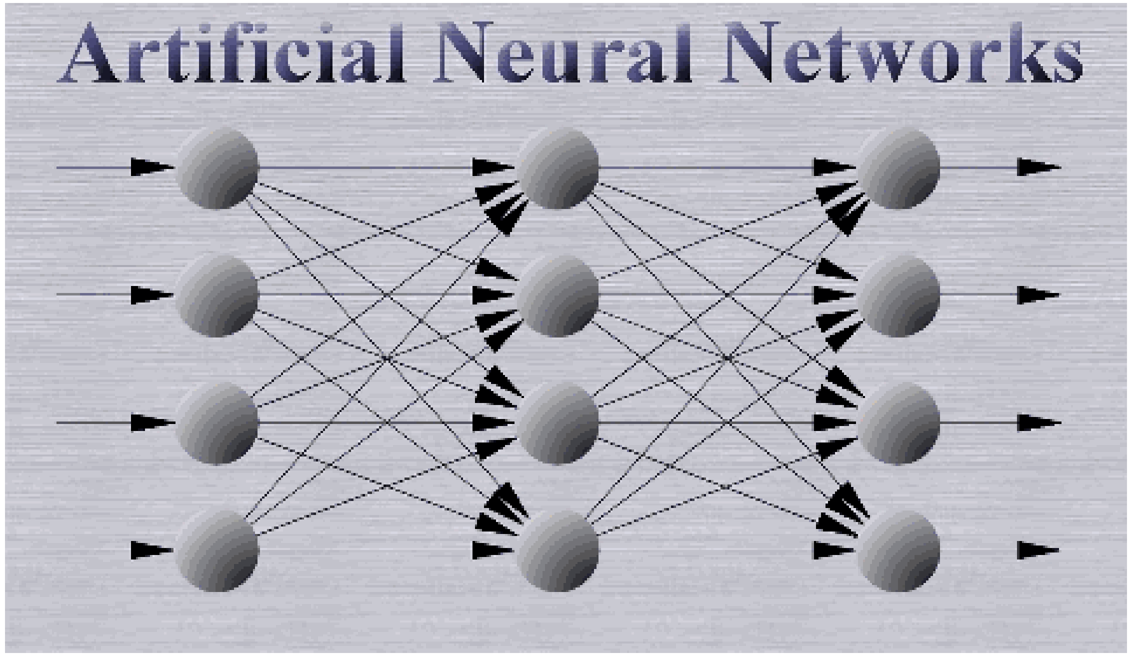
JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Deep Learning and Natural Language Processing

Minghong Xu, PhD.
Associate Professor

Deep Learning Review

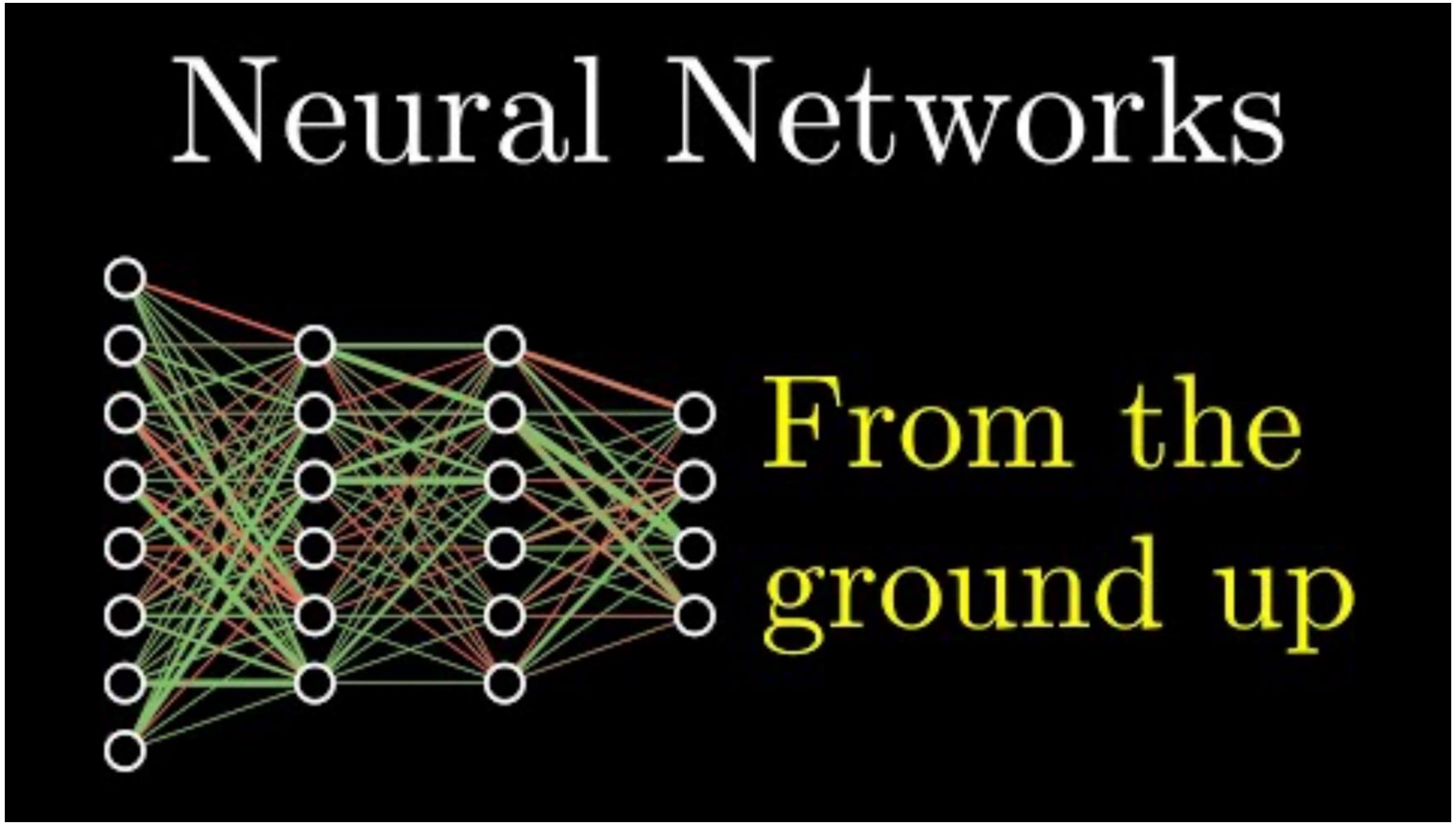
- Neural network models, inspired by neuroscience



MYSTERY
BOX
?
It's an apple!

https://www.cse.unsw.edu.au/~cs9417ml/MLP2/

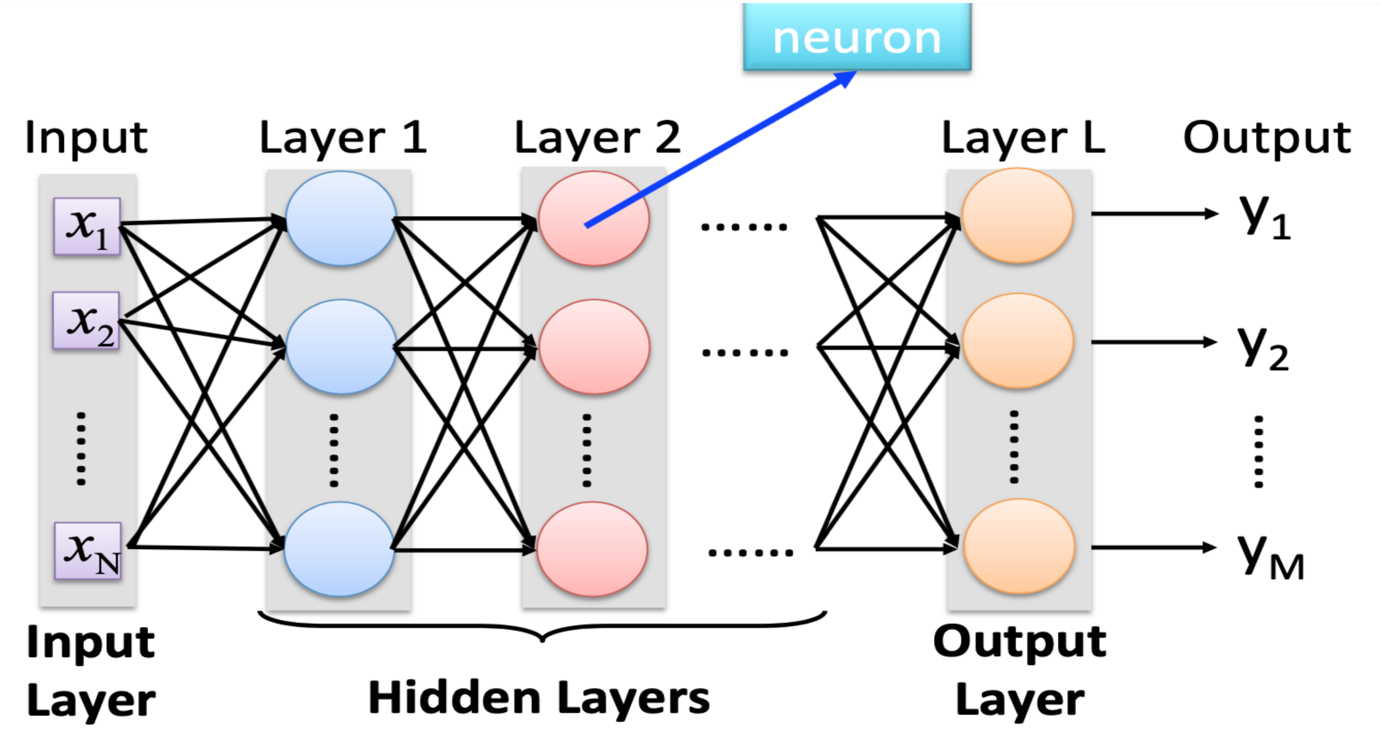
How does it work?



https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

Major Models: FFN

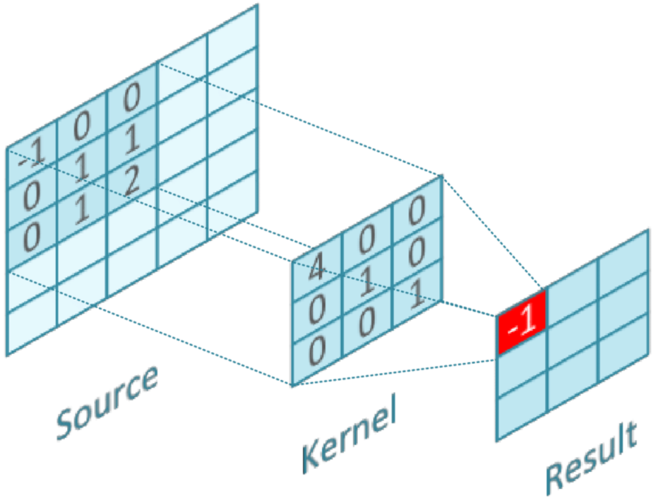
- Feedforward Neural Networks: plain models

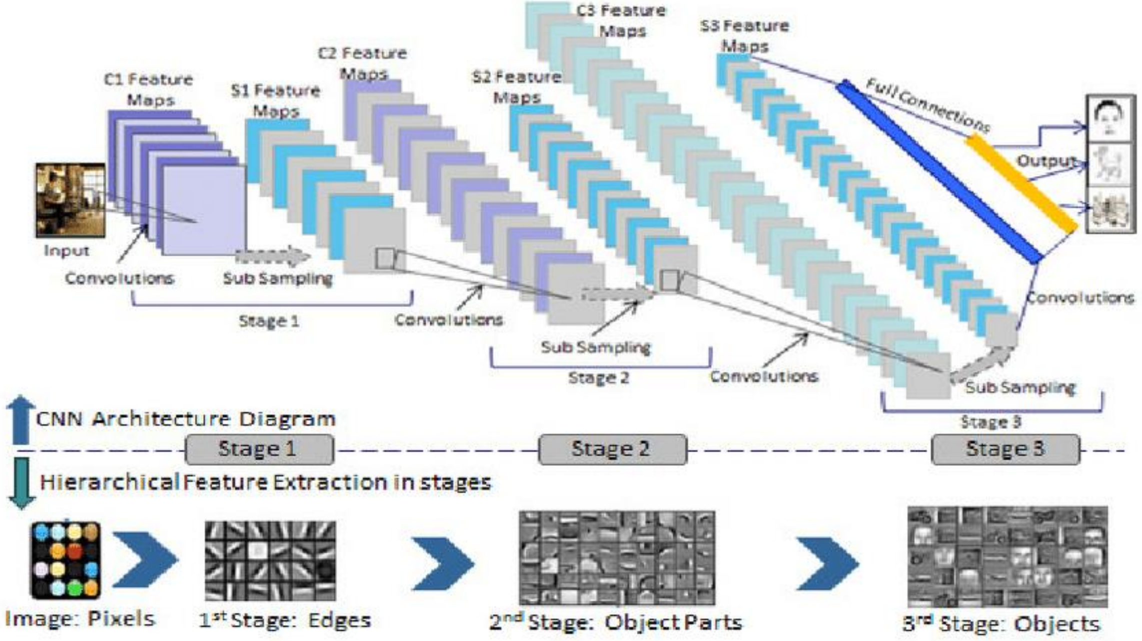


Deep means many hidden layers

Major Models: CNN

- Convolutional Neural Networks: use convolution in some layers

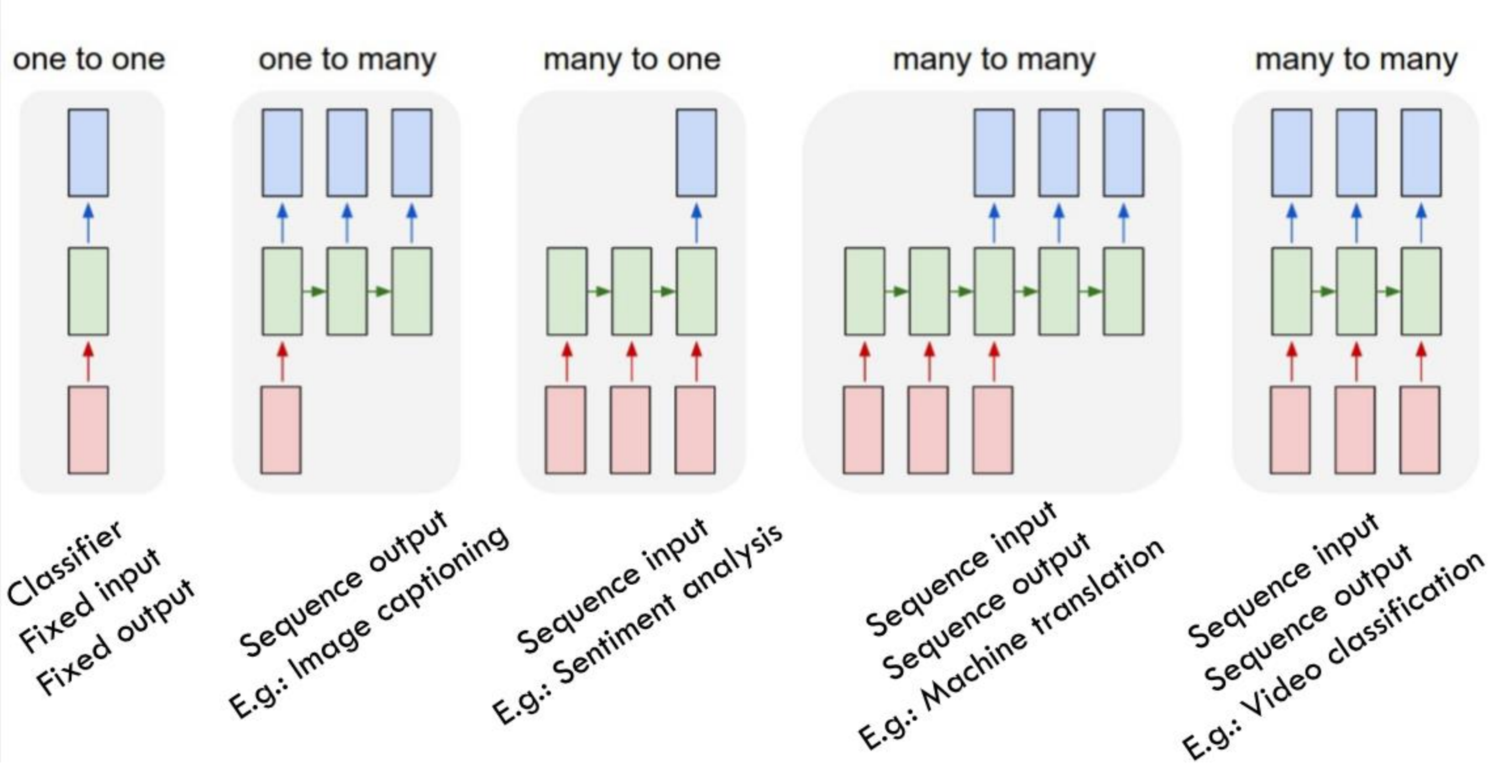




- https://setosa.io/ev/image-kernels/

Major Models: RNN

- Recurrent Neural Networks: the same network is used over time for sequential processing

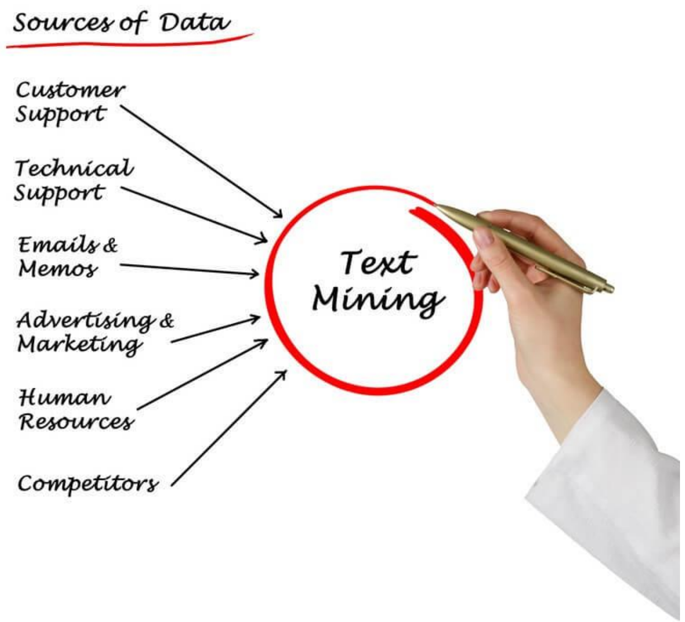


JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Natural Language Processing

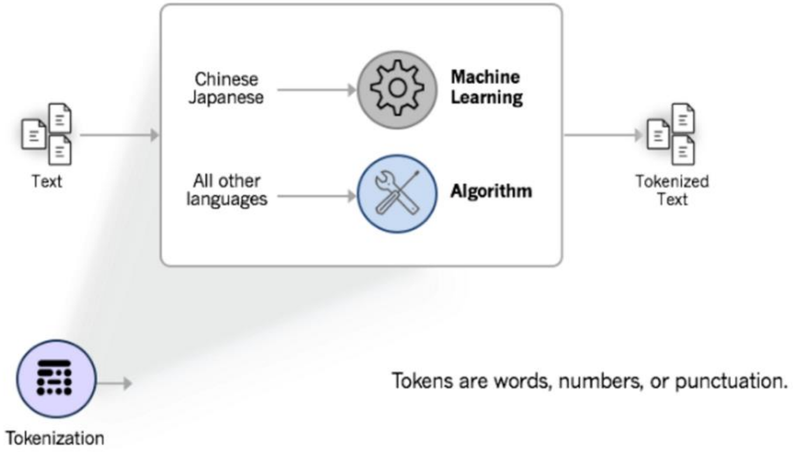
What is NLP

- Text analytics, text mining
- Mine knowledge/information from huge amount of text data
- Many NLP systems are trained on very large collections of text (also called *corpora*)



Tokenization

- Break a stream of text into meaningful units
- Tokens: words, phrases, symbols

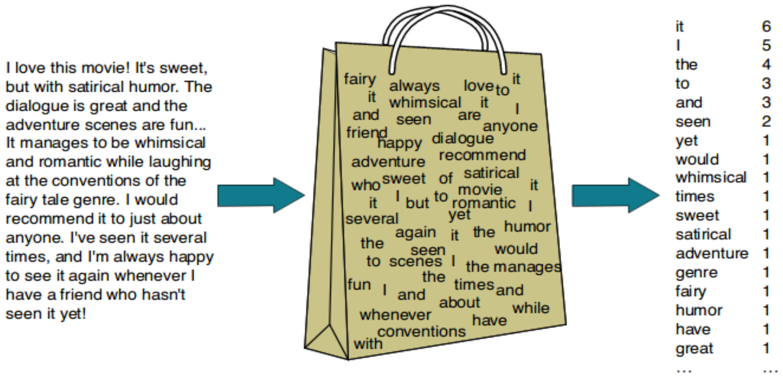


Tokens are words, numbers, or punctuation.

- https://platform.openai.com/tokenizer

Term Frequency

- Term: token in text
- A term is more important if it occurs more frequently in a document
- *So what to do?*
  - Count the occurrence!
- $tf(t,d) = frequency\ count\ of\ term\ t\ in\ doc\ d$
- This approach is called Bag of Words model
- *Any issue with this approach?*



I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

TF Normalization

- Documents have different length
  - Doc 1 has 1000 words, and ‘coding’ appears 5 times
  - Doc 2 has 10 words, and ‘coding’ appears 2 times
- *How to solve this?*
  - Normalization!
- $tf(t,d) = \frac{frequency\ count\ of\ term\ t\ in\ doc\ d}{total\ words\ in\ doc\ d}$
- There are other ways to do normalization, such as maximum TF normalization

Stop Words

- ‘the’ ‘is’ are high frequency words in this document
- *But are they important?*
- Stop words: commonly used words in language
  - If it appears in all documents frequently, then it is not related with any specific doc
- *How can we identify them?*

Inverse Document Frequency

- Document frequency: a term is more discriminative if it occurs only in fewer document
- Inverse document frequency: assign higher weights to rare terms
- $idf(t) = \log(\frac{total\ documents}{documents\ with\ term\ t})$
- Combining *tf* and *idf*
- $tf \cdot idf = tf(t, d) \times idf(t)$

Lemmatization

- *How about “concepts” vs “concept”, “readings” vs “reading”?*
- *Another example can be “good” vs “best”*

- Lemmatization: the process of transforming a word into its root form

Vectorization

- Machine learning algorithms (including deep nets) require input to be vectors of numeric values
- Vectorization: represent words in a vector format

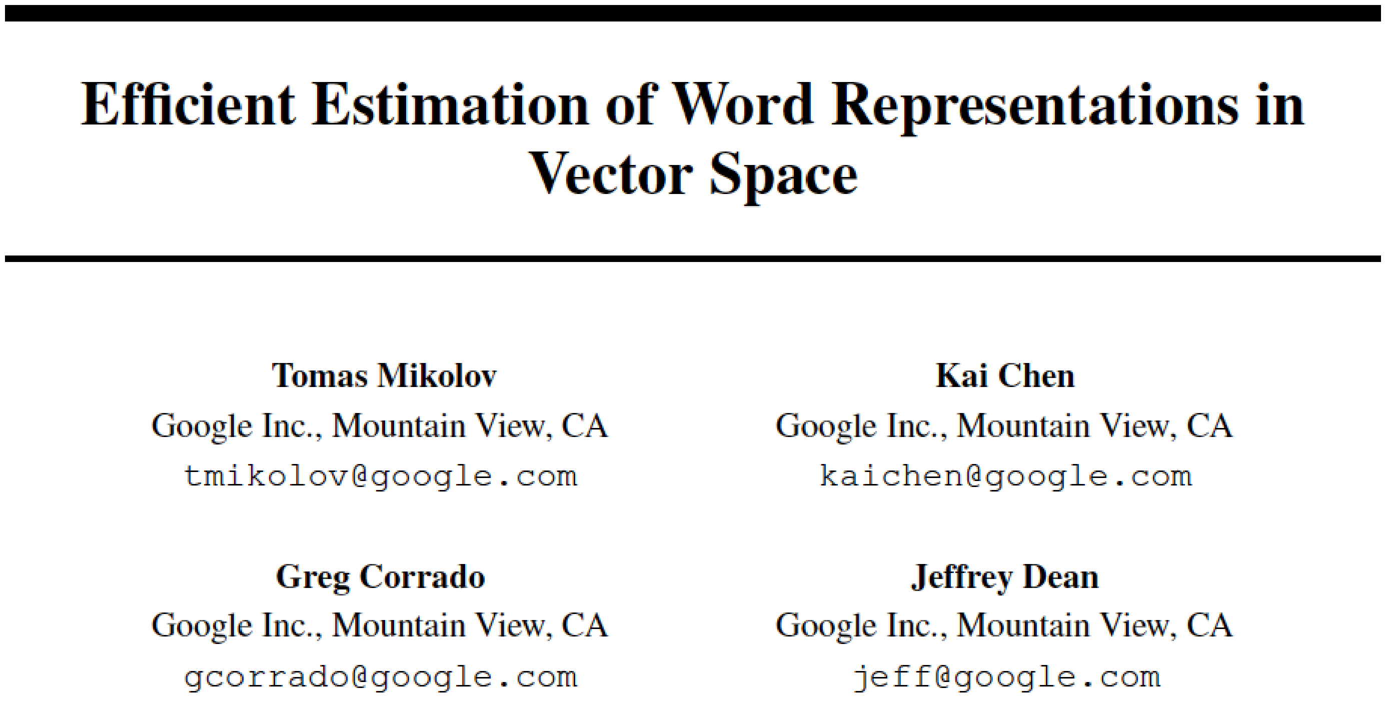
Word Embedding

- *Check the Token IDs on https://platform.openai.com/tokenizer*
- *Can we do better to represent each word?*
- Yes! We need the meaning

- Word Embedding:
  - Capture semantic meaning
  - Words that are semantically similar are close to each other in the vector space

Word2Vec

- Published in 2013 by researchers from Google



Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Paper: Efficient Estimation of Word Representations in Vector Space

Two Examples

- “I will take a train from Baltimore to DC today.”

vs

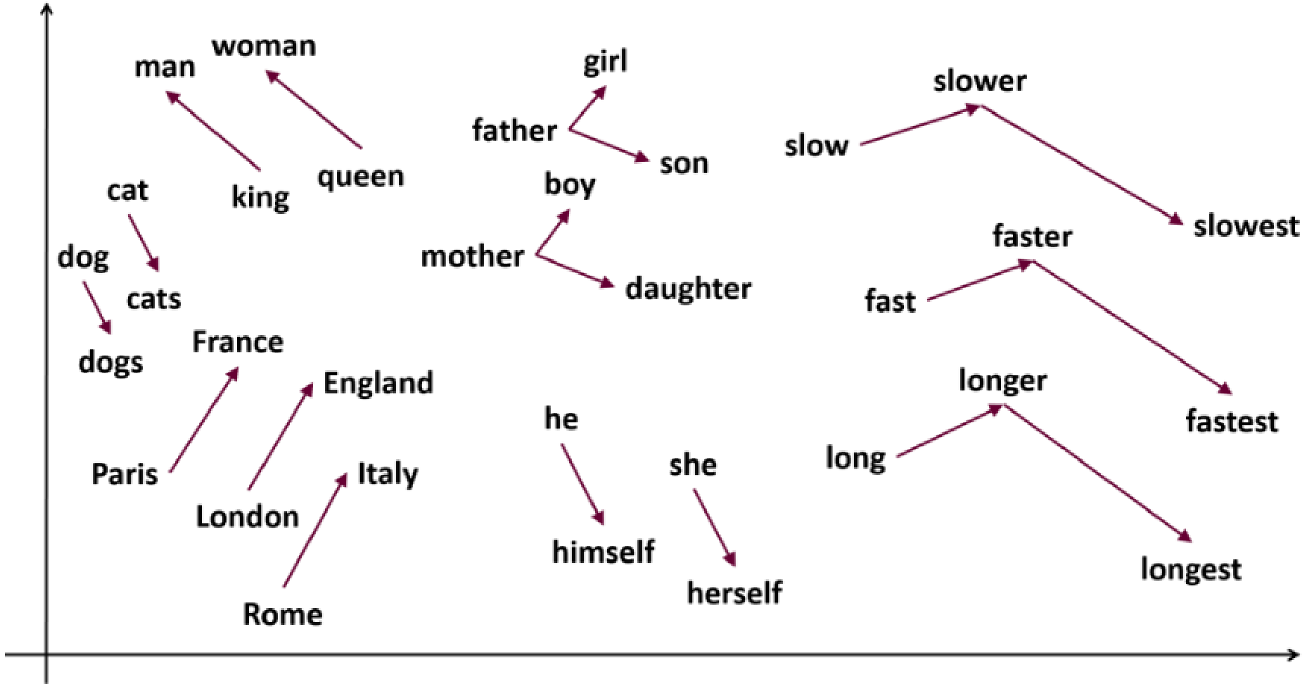
- “I will train my first neural network model today.”

“You shall know a word by the company it keeps!”

by **John Rupert Firth**

Demo

- https://projector.tensorflow.org/
- Depend on training documents
- Statistical model
- Not symmetric



Colab Exercise on Word2Vec

- Download Word2vec.ipynb, ChatGPT_sentiment_txt.csv and ChatGPT_sentiment_txt_processed.csv from Canvas
- Upload the notebook to Colab
- Upload the two data files to your Google drive