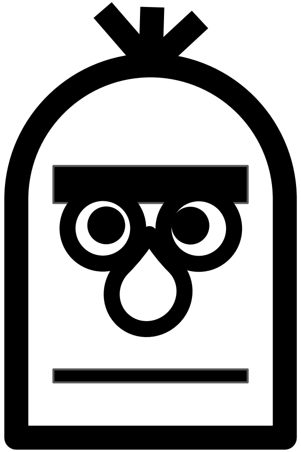
Bagging to BERT



A tour of Natural Language processing

Prepared for ODSC West ‘22

Benjamin Batorsky, PhD

SETUP

Download Data (reviews.pkl.gz): shorturl.at/joMSW OR https://ai.stanford.edu/~amaas/data/sentiment/

Github repo: https://github.com/bpben/bagging_to_bert

Google Collaboratory (recommended):

https://colab.research.google.com/github/bpben/bagging_to_bert/blob/main/tutorial_notebook_part1.ipynb

https://colab.research.google.com/github/bpben/bagging_to_bert/blob/main/tutorial_notebook_part2.ipynb

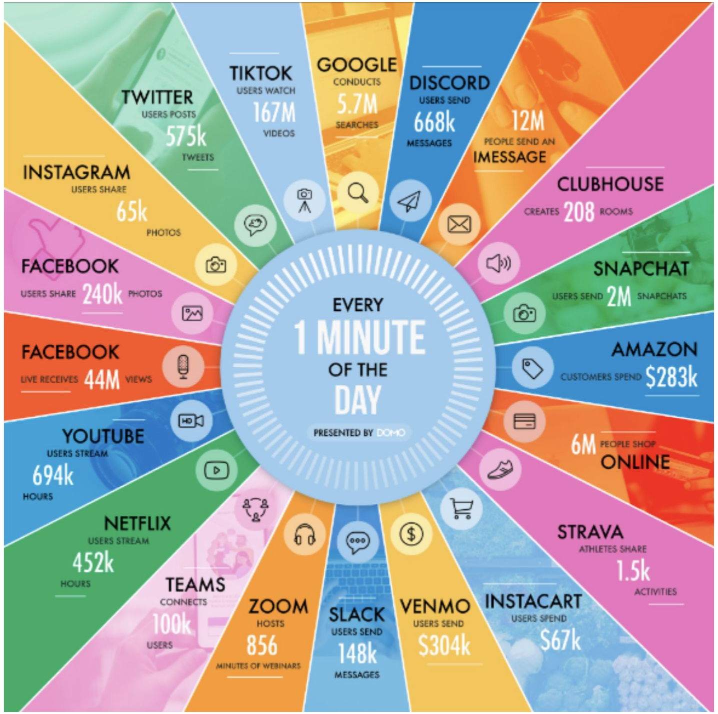
Who am I?

- PhD, Policy Analysis
- City of Boston Analytics Team
- ThriveHive, Marketing Data Science
- MIT, Food Supply Chain
- Harvard, NLP instructor
- Ciox Health, Clinical NLP
- Northeastern EAI, Data Science solutions

EAI The Institute for Experiential AI
Northeastern University

- Building AI solutions for partners across industries
- Bridging academia and industry
- Tackling research questions around AI applications and ethics

Explosion of data...unstructured data, that is



https://www.domo.com/learn/infographic/data-never-sleeps-9

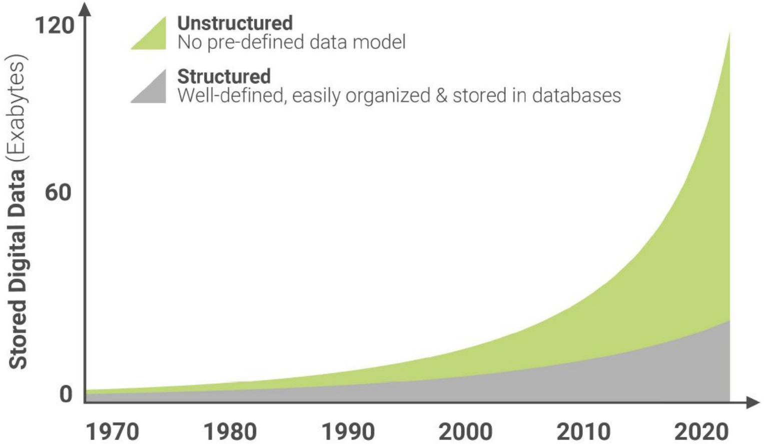
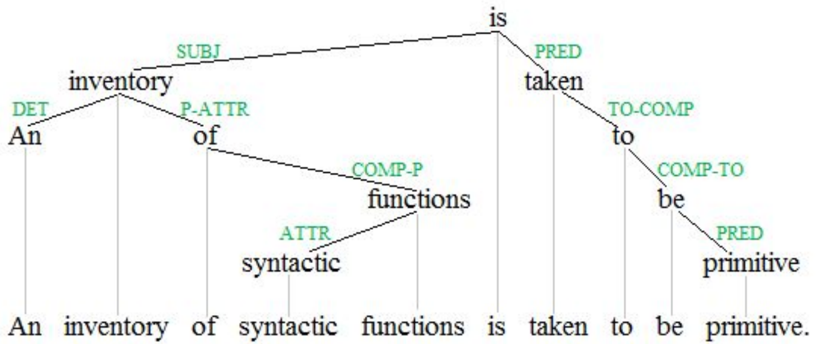


Chart:https://www.datanami.com/2019/01/14/from-oscar-to-ai-mining-visual-assets-for-fun-and-profit/
Data: IDC

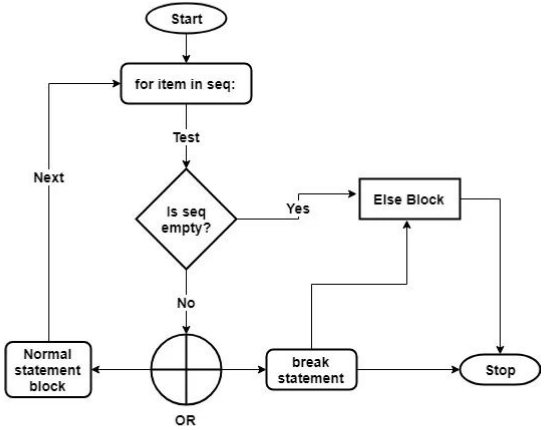
What is Natural Language?

What is Natural Language?

“A language that has developed naturally in use (as contrasted with an artificial language or computer code).” (Oxford Dictionary definition)

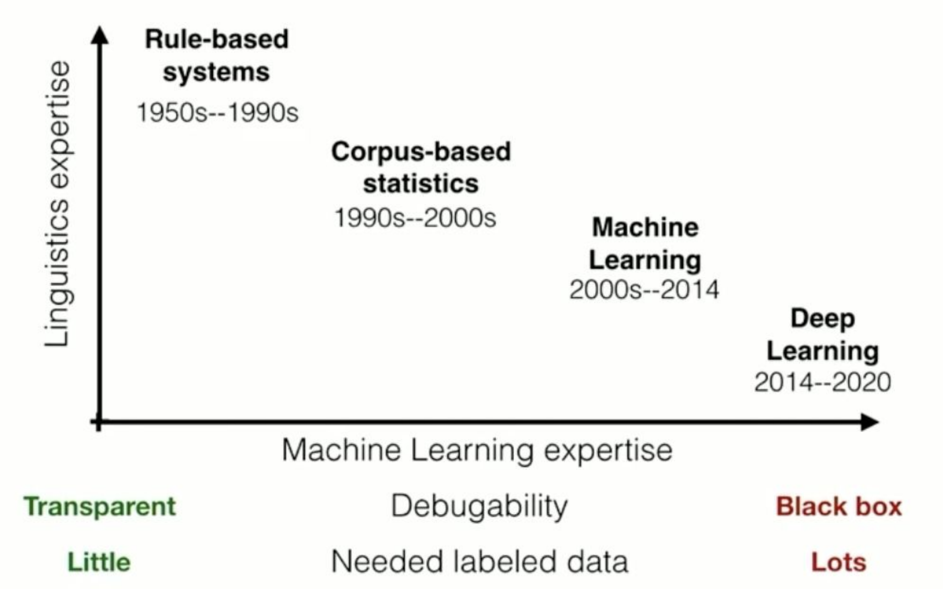


https://en.wikipedia.org/wiki/Dependency_grammar



https://www.techbeamers.com/python-for-loop/

History, in short



[Yoav Goldberg: The missing elements in NLP (spaCy IRL 2019)]

Now we can do things like this

Write with Transformer from HuggingFace

[Write With Transformer distil-gpt2]

I am a data scientist and I **am always looking to improve the data processing abilities of people who are passionate about data science.**

Written by Transformer · transformer.huggingface.co 🦄

And this:

TEXT PROMPT an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

Edit prompt or view more images↓

TEXT PROMPT an armchair in the shape of an avocado. . . .

AI-GENERATED IMAGES

Edit prompt or view more images↓

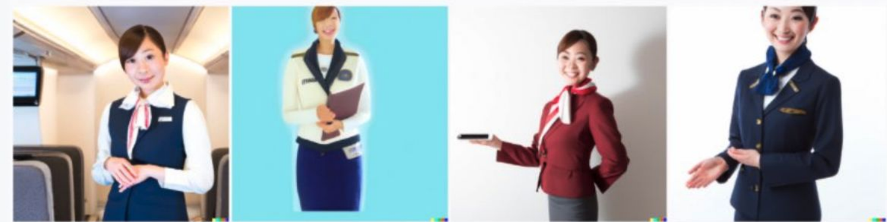
https://openai.com/blog/dall-e/

Though also, this:

an illustration of a baby daikon radish in a tutu walking a dog

https://openai.com/blog/dall-e/

Prompt: a flight attendant; Date: April 6, 2022



Prompt: lawyer;
Date: April 6, 2022

https://twitter.com/WriteArthur/status/1512429306349248512

How is text different from structured data?

How is text different from structured data?

- Height/weight - Numeric values, 6’0” > 5’0”, 6’0” = 3’0” * 2, 1 lb = 16 oz
- Stock ticker - State information available, day 2 follows day 1, prices are numeric

What is the point of NLP?

Goal: Ensure accurate response to input text

Ideal world: Infinite resources, read and respond correctly to every input

Real world: Need heuristics/automation

Goal: Ensure accurate response to informative representation of input text

NLP system should contain

- Method for creating informative representation
- Method for utilizing that informative representation for application

Stops on our tour

- Tokenization
- Word frequencies
- Weighted word frequencies (TF-IDF)
- Topic models
- Word embeddings
- Recurrent Neural Models
- Large Language Models (e.g. BERT)

The IMDB review dataset

- Source: http://ai.stanford.edu/~amaas/data/sentiment/
- 50k unique movie reviews, labelled for sentiment (positive vs negative)
- Why this dataset?
  - Easily accessible, reasonable size (84 MB)
  - Simple, balanced, binary objective (positive/negative)
  - Short, clean passages (~1k characters on average)
- What’s missing
  - Issues of size, cleanliness and clarity of target

What we'll be using

- Scikit-learn
  - Feature engineering modules for performant word vectorization
  - "Topic modelling" with Non-negative matrix factorization
  - Classification models
- SpaCy (https://spacy.io/)
  - All-purpose NLP library
- Transformers (https://huggingface.co/docs/transformers/index)
  - Transformer-based language models
- PyTorch

Token: “Useful semantic unit”

- Token - “useful semantic unit”
  - Breaking text into pieces
  - Can be “whitespace”-split, characters, etc
- “N-gram” - N continuous tokens
- Tokenization strategy
  - Extremely important for system design
- This presentation
  - Whitespace-split, unigrams

“I am learning Natural Language Processing (NLP)”

<split on whitespace>

Unigrams

I, am, learning, Natural, Language, Processing, (NLP)

Bigrams

I am, am learning, learning Natural...

7-grams

I am learning Natural Language Processing (NLP)

“Bagging”

I am a Patriots fan

am	a	fan	I	Patriots

I am a 49ers fan

am	a	fan	I	49ers

Document-Term Matrix

am	a	fan	I	Patriots	49ers
1	1	1	1	1	0
1	1	1	1	0	1

The power of the document-term matrix (word count)

am	a	fan	I	Patriots	Giants
1	1	1	1	1	0
1	1	1	1	0	1

	Comedies		Histories	
	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

Figure 6.2 The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

https://web.stanford.edu/~jurafsky/slp3/6.pdf

To the notebook - word counts

Sentiment analysis - our progress so far

	Precision	Recall	F1 score
Deterministic	0.58	0.58	0.57
Word count	0.88	0.88	0.88

Making word counts more informative

- NLP: Informative representation of text
- Raw word count = each word counted the same
  - “I am a Patriots fan” vs “I am a 49ers fan”
- Reduce “noise”
  - Turn words into common form
    - “I am” and “I will” -> “I be”
  - Stripping uninformative words
    - e.g. “the”, “and”
- Weighting
  - Important words count more, unimportant words count less

am	a	fan	I	Patriots	49ers
1	1	1	1	1	0
1	1	1	1	0	1

Term Frequency - Inverse Document Frequency (TF-IDF)

- Term frequency: Count of term (T) within a document
- Document frequency (DF)
  - Documents with T
- Inverse document frequency (IDF)
  - 1 / DF
  - High DF (common term) = low IDF
  - Lower DF (uncommon term) = high IDF
- TF*IDF, term count weighted by how "informative" that term is

	am	a	fan	I	Patriots	Giants
Doc1	1	1	1	1	1	0
Doc2	1	1	1	1	0	1

T	DF	IDF	Doc1 TF	Doc2 TF	Doc1 TF*IDF	Doc2 TF*IDF
Patriots	1	1	1	0	1	0
49ers	1	1	0	1	0	1
fan	2	0.5	1	1	0.5	0.5

Note: TFIDF usually has some additional "smoothing" transformations

The difference between a Patriots fan and a 49ers fan

	am	a	fan	I	Patriots	Giants
TFIDF Doc1	0.5	0.5	0.5	0.5	1	0
TFIDF Doc2	0.5	0.5	0.5	0.5	0	1

Measuring similarity - “cosine similarity” measure comparing vectors

(higher = more similar)

Similarity (Doc1, Doc2) = 0.8

Similarity (TFIDF Doc1, TFIDF Doc2) = 0.5

To the notebook - TF-IDF

Sentiment analysis - our progress so far

	Precision	Recall	F1 score
Deterministic	0.58	0.58	0.57
Word count	0.88	0.88	0.88
TF-IDF	0.89	0.89	0.89

Curse of dimensionality with word counts

Book, author, year	Unique words	Words	Words per unique word
Sense & Sensibility by Jane Austen (1811)	7,265	119,893	16.5
A Tale of Two Cities by Charles Dickens (1859)	10,778	137,137	12.7
The Adventures of Tom Sawyer by Mark Twain (1876)	7,896	71,122	9
The Hobbit by JRR Tolkien (1937)	6,911	96,072	13.9
The Lion, The Witch, and The Wardrobe by C.S. Lewis (1950)	3,520	39,166	11.1
Harry Potter and The Sorcerer's Stone by J.K. Rowling (1998)	6,185	77,883	12.6
Twilight by Stephenie Meyer (2005)	8,507	119,270	14

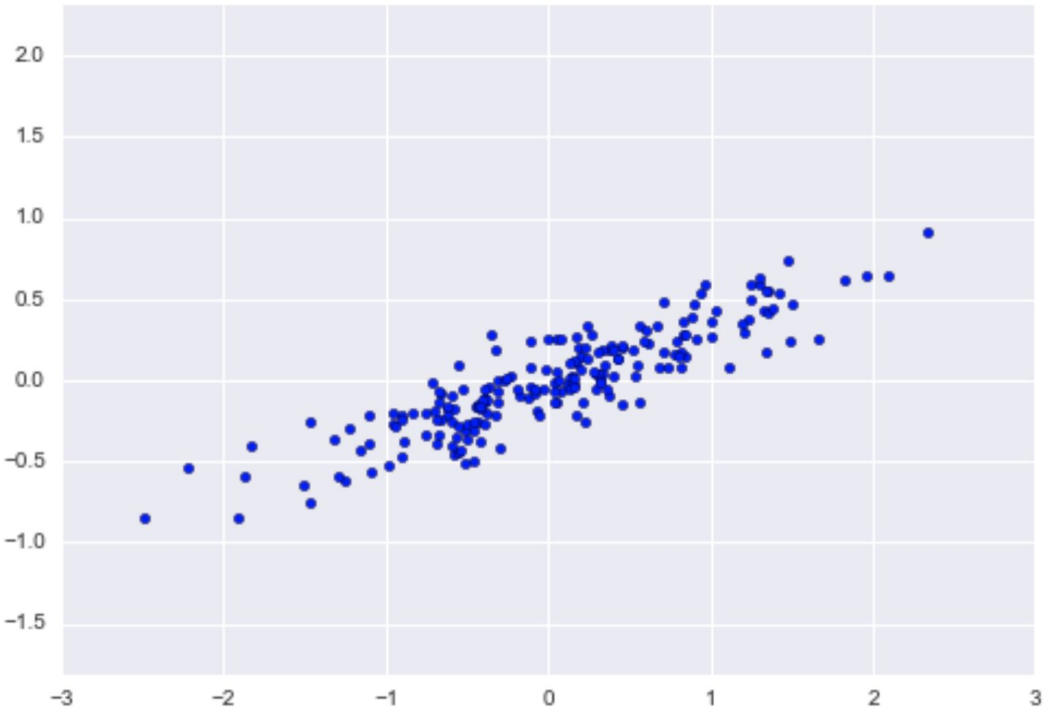
http://www.tylervigen.com/literature-statistics

Shakespeare's plays
884k total words
28k unique words
https://www.opensourceshakespeare.org/statistics/

Topic models

- “Topic models are algorithms for discovering the main themes that pervade a large and otherwise unstructured collection of documents” (Blei 2012)
- NLP - Informative representation of text
- Document = f(Topics), Topics = g(words)
    - Typically number of topics << size of vocabulary
    - Want to minimize the information lost by representing in this way

Extracting axes of variation in data



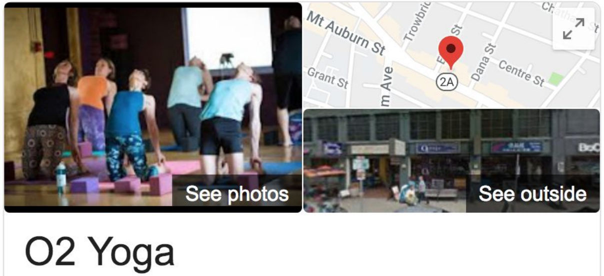
https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

Extracting axes of variation in data

https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

Categorizing small/mid-size businesses

- Small/Mid-sized businesses that straddle multiple categories
- Customer questions
  - Sales: “Which businesses are similar to this lead?”
  - Marketing: “How do we better personalize ad campaign messaging?”
- Business websites rich source for services offered



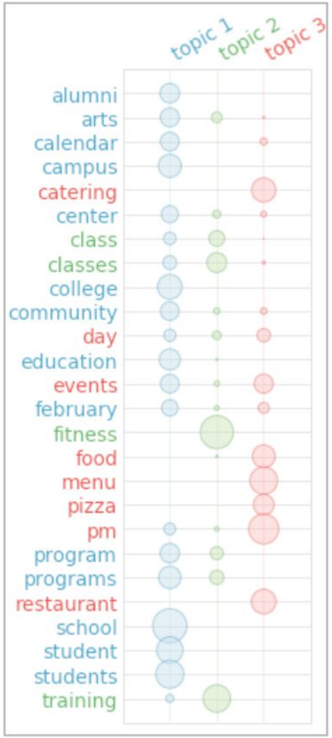
“...offers classes 7 days a week. Our vegan cafe opened in July of 2013... We also have a retail store selling a limited selection of US-made yoga gear...peruse the retail, enjoy the cafe, or get a massage with one of the body workers in the Wellness Center…”

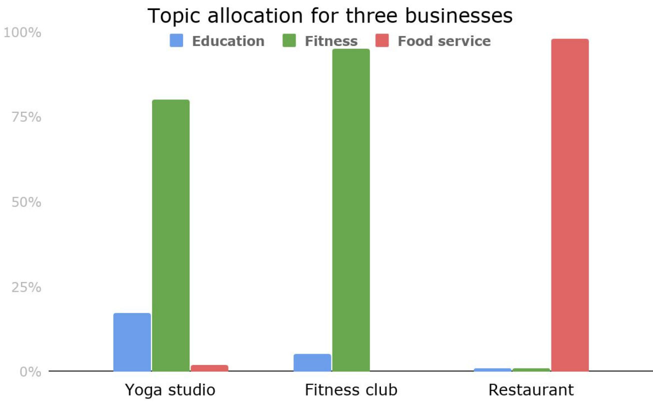
Yoga studio, cafe AND retail?!

Topic models for informative “business representation”

- Topic modelling
  - Website text to TF-IDF vectors
  - Non-negative matrix factorization (NMF)
- Output
  - Business-level representation in “topic space”
  - Calculate business-business similarity
  - Split into “similar” groups, based on parameters
  - Other predictive models

Product similarity





Circles are sized according to “relevance” to each topic

To the notebooks - topic models

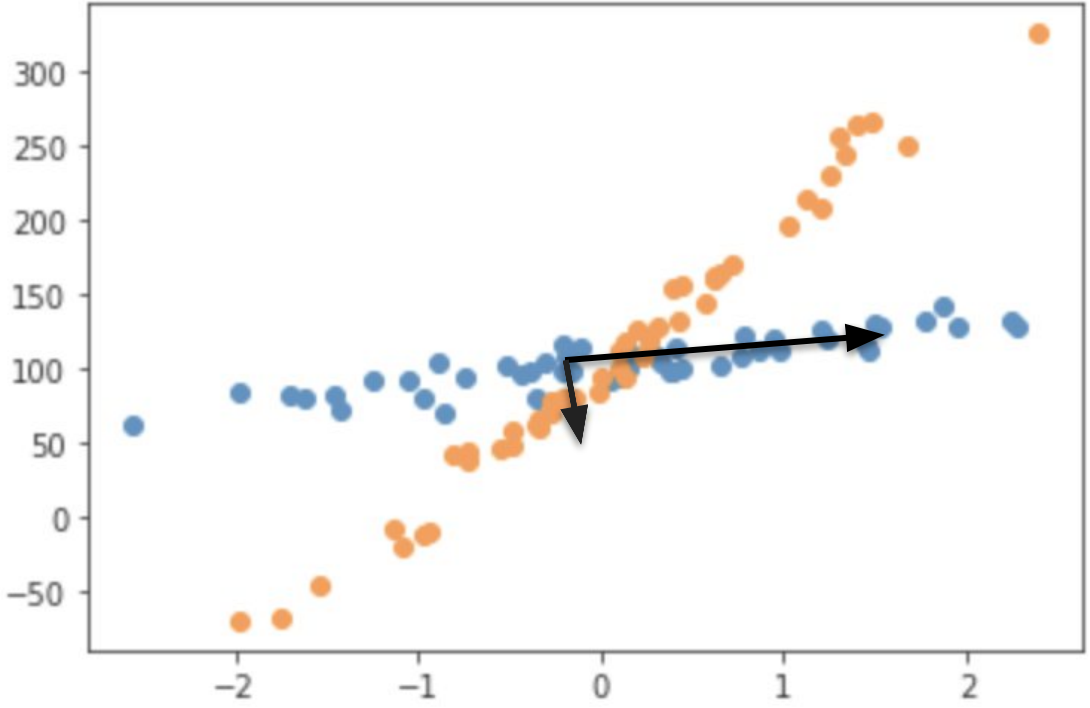
Sentiment analysis - our progress so far

	Precision	Recall	F1 score
Deterministic	0.58	0.58	0.57
Word count	0.88	0.88	0.88
TF-IDF	0.89	0.89	0.89
Topic model (NMF)	0.76	0.76	0.76

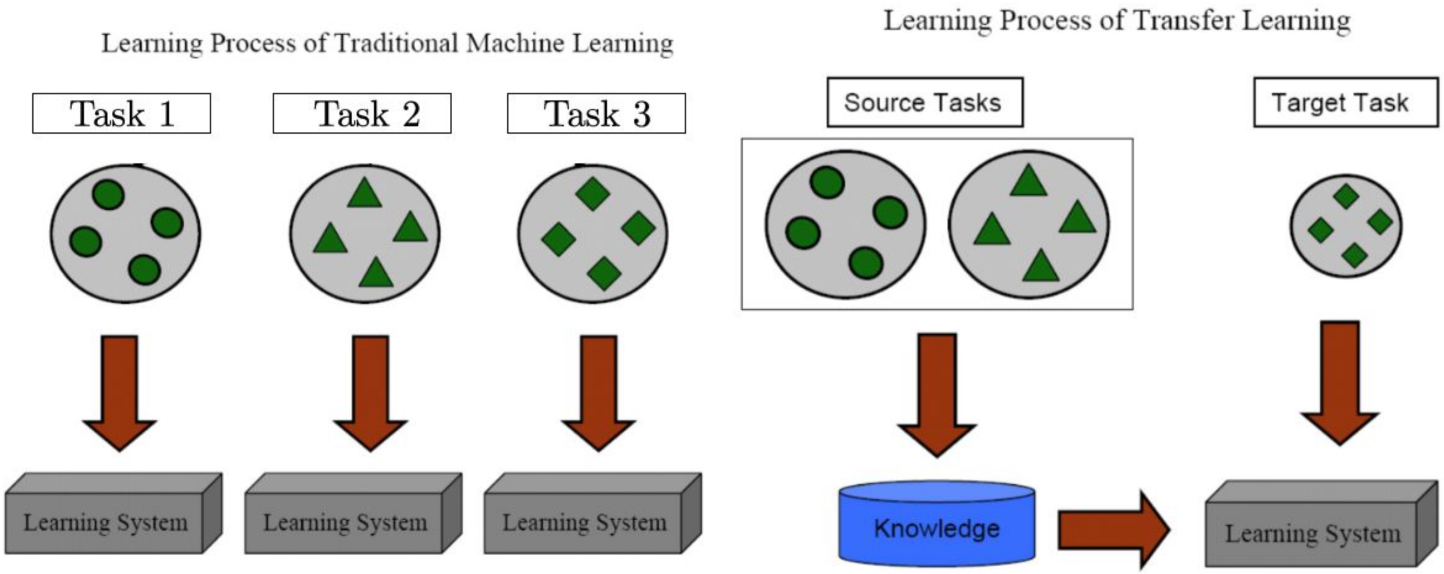
This works on your current dataset

https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

But what about a new dataset?



Transfer learning



(a) Traditional Machine Learning (b) Transfer Learning

https://www.cse.ust.hk/~qyang/Docs/2009/tkde_transfer_learning.pdf

Source task: term co-occurrence

What does this tell you about pie vs cherry and pie vs digital?

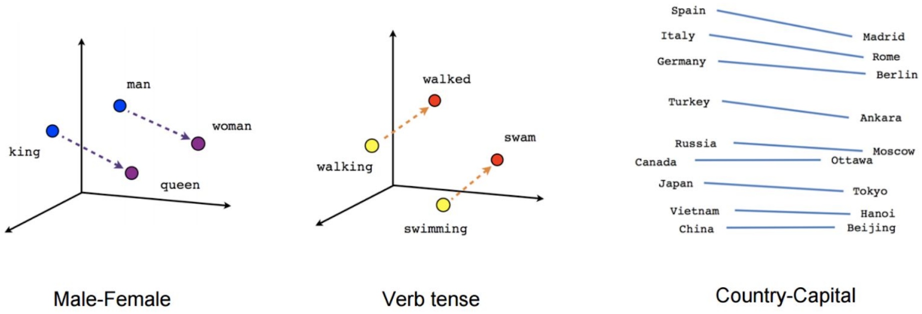
	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	3982	378	5	13	7703
count(context)	4997	5673	473	512	61	11716

Figure 6.10 Co-occurrence counts for four words in 5 contexts in the Wikipedia corpus, together with the marginals, pretending for the purpose of this calculation that no other words/contexts matter.

https://web.stanford.edu/~jurafsky/slp3/6.pdf

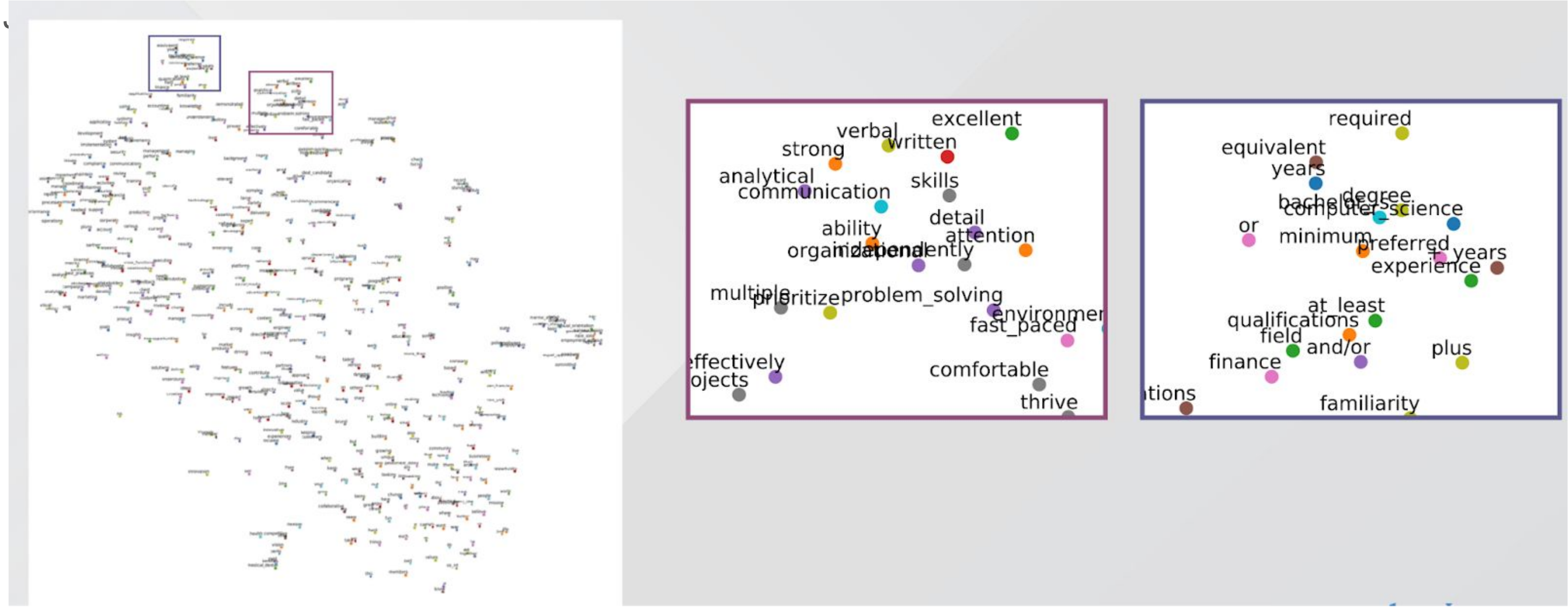
Word embeddings: Informative word-level representations

- “You shall know a word by the company it keeps” J.R. Firth (English Linguist)
- Learn an numerical vector for each word based on context
    - Word2Vec: Neural model
    - GloVe: Corpus-based statistical model
- Distance between words has meaning
    - Similar words = similar vectors
    - Madrid:Spain as Rome:Italy
- Dimensions themselves not (readily) interpretable



[1301.3781] Efficient Estimation of Word Representations in Vector Space

Embeddings for words in job descriptions



[Applying Dynamic Embeddings in Natural Language Processing to track the Evolution of Tech Skills | Maryam Jahanshahi]

Considerations when using embeddings

- Pre-trained embeddings are widely available
  - Often trained on general internet
  - Can find domain-specific
    - Example, biomedical: https://allenai.github.io/scispacy/
- Caution!
  - Bias in text = bias in embeddings
- Gender bias in adjectives - strong correlation, weaken after women’s movement

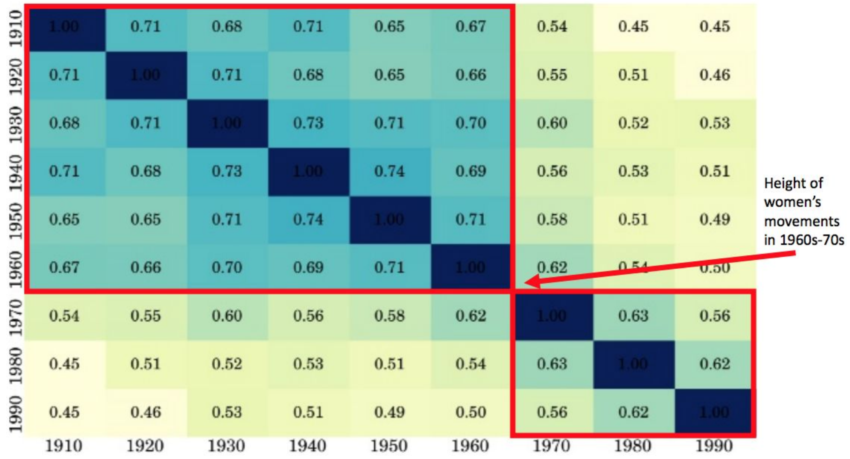


Fig. 4. Pearson correlation in embedding bias scores for adjectives over time between embeddings for each decade. The phase shift in the 1960s–1970s corresponds to the US women’s movement.

[Word embeddings quantify 100 years of gender and ethnic stereotypes | PNAS]()

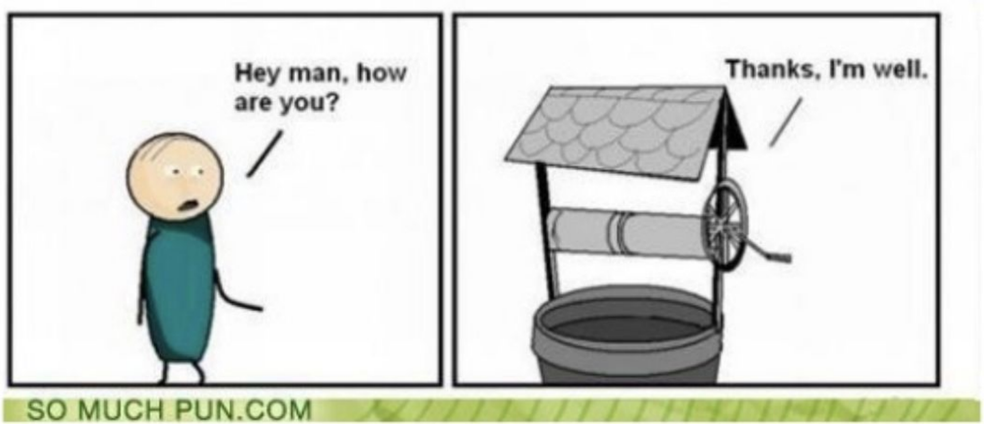
To the notebooks - word embeddings

Sentiment analysis - our progress so far

	Precision	Recall	F1 score
Deterministic	0.58	0.58	0.57
Word count	0.88	0.88	0.88
TF-IDF	0.89	0.89	0.89
Topic model (NMF)	0.76	0.76	0.76
Word embeddings	0.84	0.84	0.84

Oddities of language

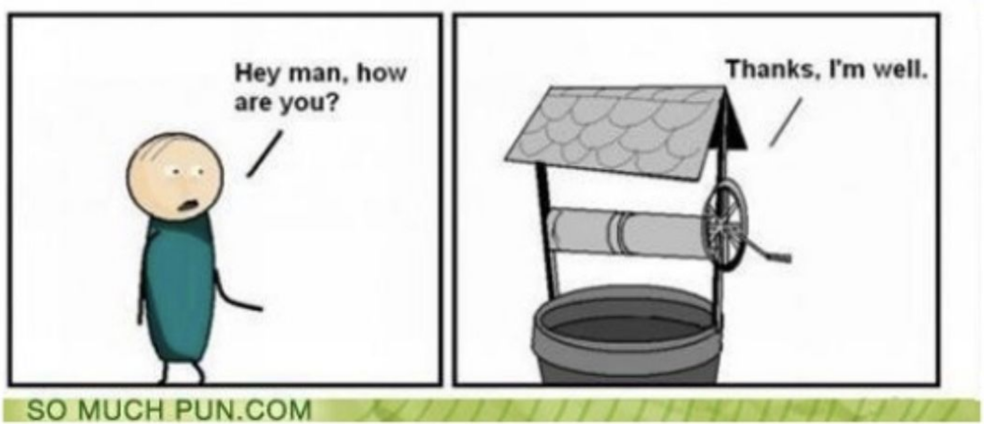
Why is this funny?



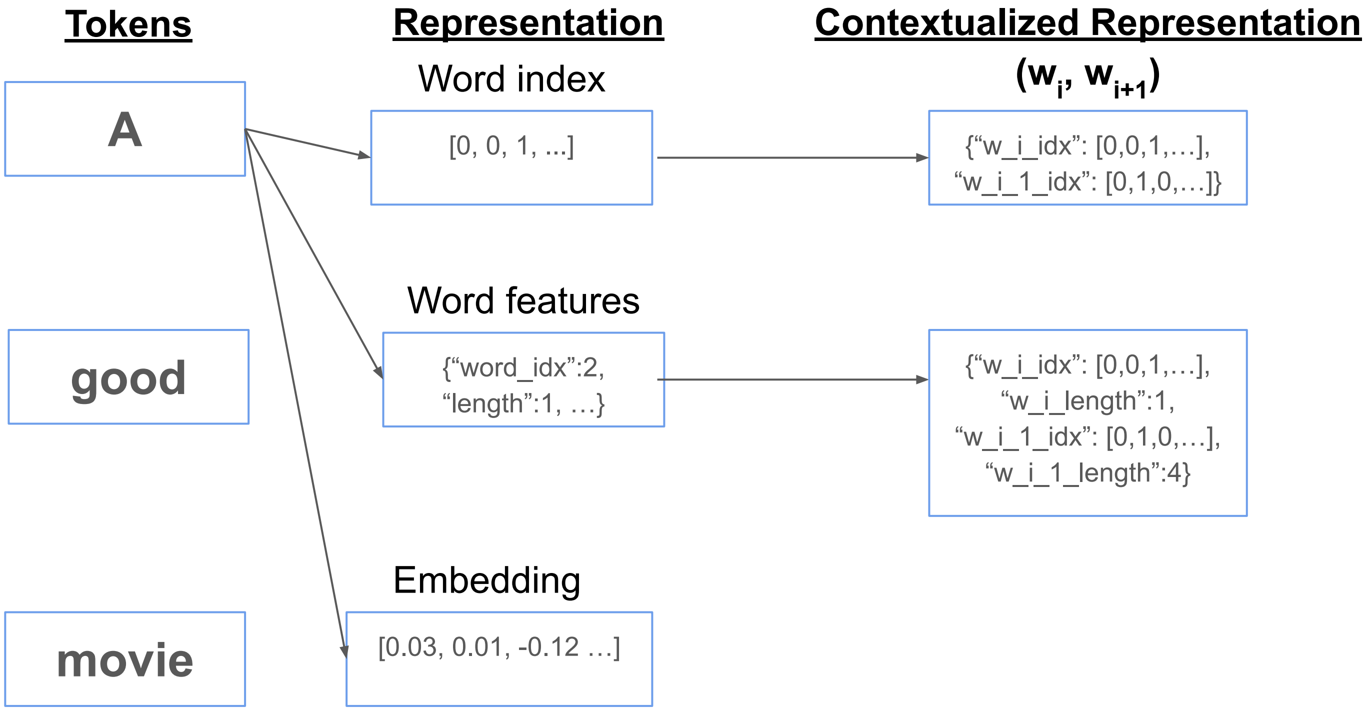
Oddities of language

Why is this funny?

- “Homonym” - Same spelling or pronunciation, different meaning
- *Context matters!*
- Bagging - word counts independent from one another
- GloVe/Word2Vec - one vector per word

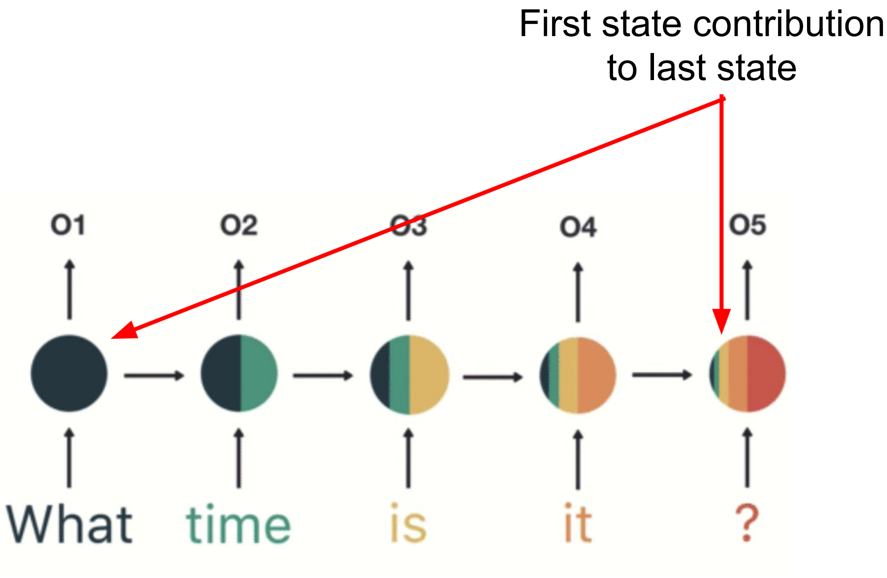


One method to include context



Recurrent Neural Networks

- Information from previous states maintained in “hidden state”
- Problem:
  - Longer sequences - less information from early stages
- Various methods for “forgetting” and “remembering” specific information
  - LSTM - Long Short-Term Memory



[Illustrated Guide to Recurrent Neural Networks | by Michael Phi | Towards Data Science]

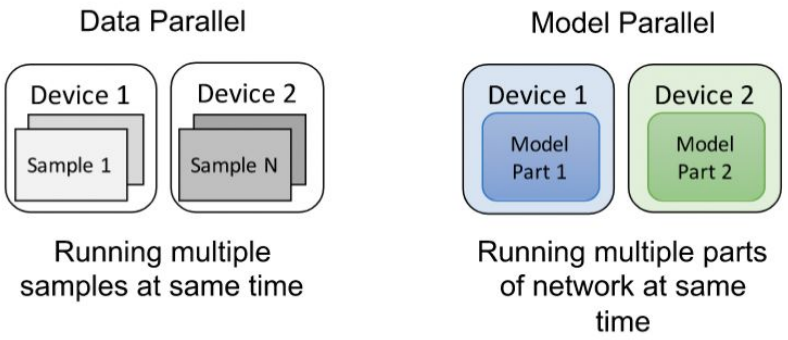
To the notebooks - LSTM

Sentiment analysis - our progress so far

	Precision	Recall	F1 score
Deterministic	0.58	0.58	0.57
Word count	0.88	0.88	0.88
TF-IDF	0.89	0.89	0.89
Topic model (NMF)	0.76	0.76	0.76
Word2vec	0.84	0.84	0.84
LSTM (5 epoch)	0.82	0.82	0.82

Issues with recurrent neural networks

- Long training time
  - Sequence models hard to parallelize, each step dependent on previous
- Issues of “forgetting” with long passages
  - LSTM, Bi-directional LSTM don’t necessarily solve this



[Parallel Neural Networks and Batch Sizes | Cerebras]

“Attention” in language

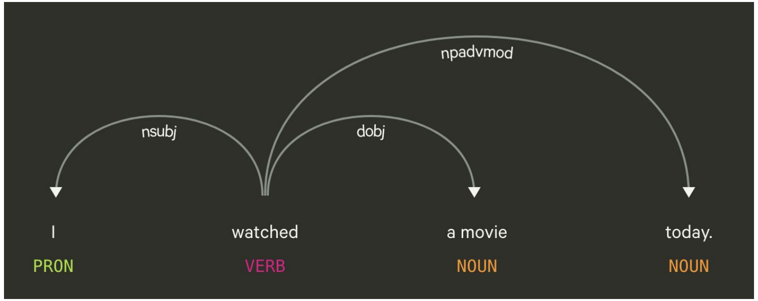
I watched a movie today.

Who is the subject of this sentence?

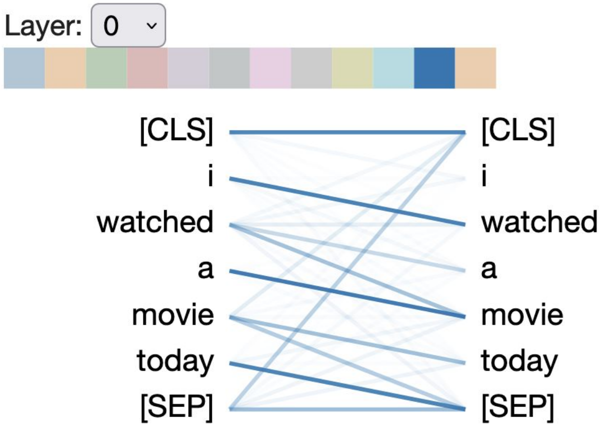
"Attention" in language

I watched a movie today.

Parse tree

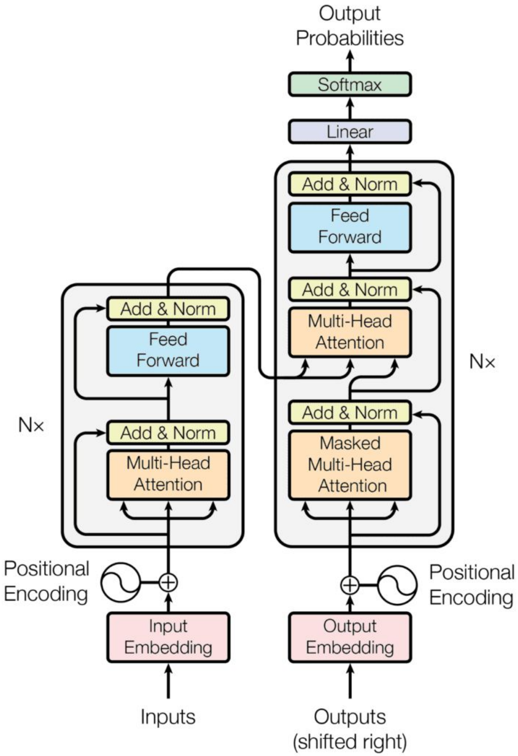


Visual of attention weight between tokens



Transformer models: Attention is all you need!

- Encoder: Translates from input to "encoded" space
  - View over entire sequence
- Decoder: Translates from encoded to output
  - Encoder output + previous decoder output
- Attention incorporated throughout
- Remove need for "recurrence"
  - Sequence position as a "positional encoding"



[1706.03762] Attention Is All You Need

Source task: Predicting a word from context

I ___ the Patriots.

What should fill in the blank?

Source task: Predicting a word from context

I ___ the Patriots, I want them to win.

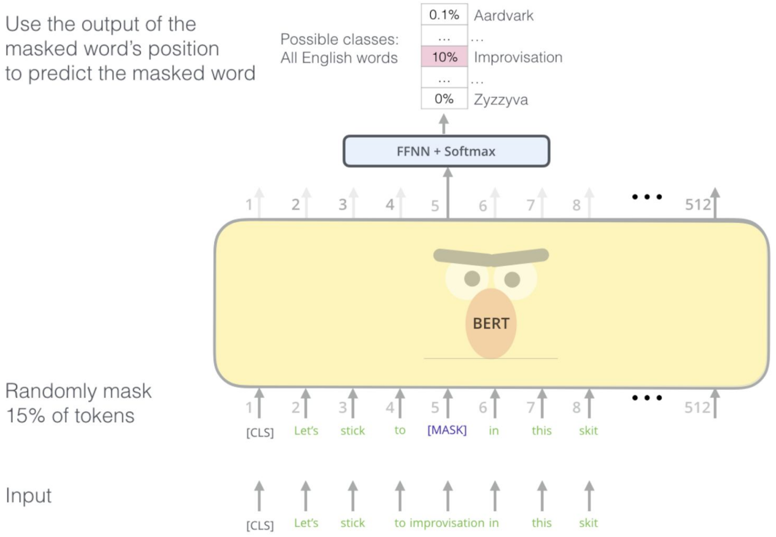
What should fill in the blank?

I ___ the Patriots, I want them to lose.

What about here?

Bi-directional Encoder Representations from Transformers (BERT)

- Transformer Language Model
  - Encoder+Decoder
  - Trained to predict next token
  - Output product of encoder + previous output
- BERT
  - Encoder-only
  - Trained to predict masked/replaced token
  - Each output is a product of the entire sequence



BERT's clever language modeling task masks 15% of words in the input and asks the model to predict the missing word.

https://jalammar.github.io/illustrated-bert/

To the notebooks - BERT

Sentiment analysis - our progress so far

	Precision	Recall	F1 score
Deterministic	0.58	0.58	0.57
Word count	0.88	0.88	0.88
TF-IDF	0.89	0.89	0.89
Topic model (NMF)	0.76	0.76	0.76
Word2vec	0.84	0.84	0.84
LSTM (5 epoch)	0.82	0.82	0.82
BERT	0.84	0.84	0.84

My advice: Start simple, add complexity

- Method for creating informative representation
  - Word counts, weighted word counts (TF-IDF)
    - Experiment with vocabulary and weights
  - Word embeddings
    - Experiment with sources, aggregations
  - Contextualized word embeddings
    - Try hand-curation (e.g. next-word embedding)
    - Bring in big guns (e.g. BERT)
- Method for utilizing that informative representation for application
  - Corpus statistics (e.g. log-likelihood of words)
  - Similarity between words or documents (e.g. cosine similarity)
  - Classifier (e.g. regression)
  - Sequence tagging (e.g. named-entity recognition)
  - Language generation (predict next word)

Thank you for coming!

Some additional materials

- spaCy universe - add-ons/integrations to spaCy
  - Scispacy - biomedical spaCy models
- HuggingFace - datasets, models, and libraries, oh my!
- Me
  - My talk on Ethics in NLP
  - NLP course materials
- Smarter people
  - Sebastian Ruder - https://ruder.io/
  - Jay Alammar - https://jalammar.github.io/
  - Lilian Weng - https://lilianweng.github.io/
  - Speech and Language Processing by Dan Jurafsky and James Martin

Get in touch!

https://benbatorsky.com/

Twitter: @bpben2

Github: bpben

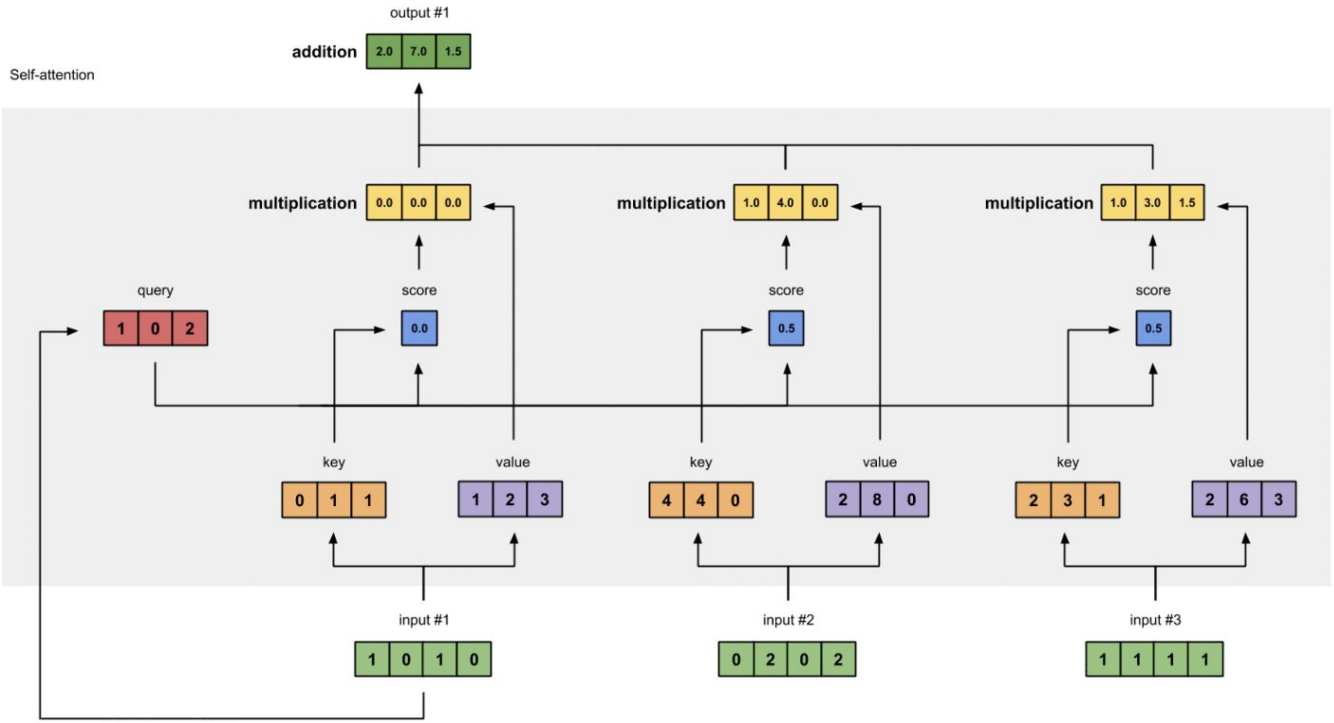
EAI The Institute for Experiential AI
Northeastern University

https://ai.northeastern.edu/jobs/

If you'd like to work with the Institute:

https://ai.northeastern.edu/contact-us/

The internals of self attention



[Illustrated: Self-Attention | Medium | Raimi Karim]()