



THE BASICS OF TEXT ANALYSIS: TOPIC MODELING AND WORD EMBEDDINGS

Kristina Lerman

DSCI 531

Spring 2025



Why text analysis

- Volume of text data is growing exponentially, necessitating methods to automatically organize, understand, search and summarize text
 - Uncover hidden topical patterns in collections.
 - Annotate documents according to topics.
 - Using annotations to organize, summarize and search.
 - Recommend relevant text to people
 - **Generate fluent and grammatical text.**



Brief history of text analysis

- 1960s
 - Electronic documents are digitized
 - Vector space models (Salton)
 - ‘bag of words’, tf-idf
- 1990s
 - Mathematical analysis tools become widely available
 - Latent semantic indexing (LSI)
 - Singular value decomposition (SVD, PCA)
- 2000s
 - Compute power becomes widely available
 - Probabilistic topic modeling (LDA); Probabilistic matrix factorization (PMF)
- 2010s
 - Massive text corpora become widely available
 - Word embeddings (word2vec, Glove); sentence embeddings
 - Pre-trained language models (BERT)
- 2020s
 - Language generation with Large Language Models (LLM): ChatGPT, Claude, ...

Low-level text analysis

Understanding text: low level features

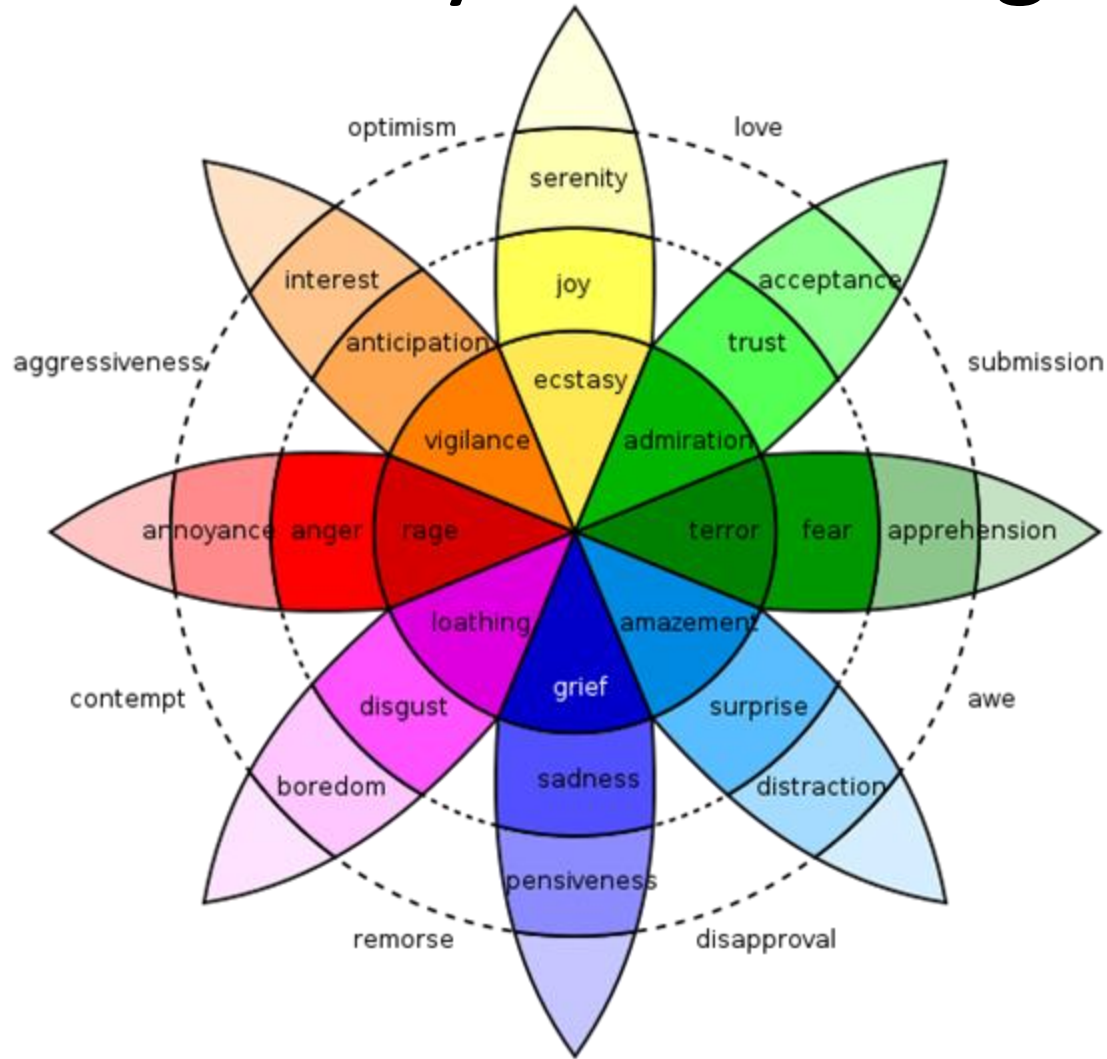
- Sentiment/emotion extraction
- Named entity recognition
- Hate speech detection
- ...

“Parse language, Speech segmentation, Extract text features, text to speech: “
{‘neg’: 0.0, ‘neu’: 1.0, ‘pos’: 0.0, ‘compound’: 0.0}

“I threw the ball to Bob on June 1”
Bob: Person
June 1: Time

“I threw the ball to Bob on June 1”
Hate speech confidence: 0.0

Case study: measuring emotion in language



- Emotions are psychological states brought on by thoughts, feelings, behavioural responses, and a degree of pleasure or displeasure.
 - E.g., happiness, sadness, surprise, disgust, anger, and fear.
 - Positive & negative emotions
- Can we predict them from text?

Dictionary-based Sentiment Analysis

- LIWC: dictionary approach
 - Words annotated with over 80 psychological categories, including positive and negative sentiment
- WKB lexicon
 - 14,000 English words annotated with emotional valence (positivity/negativity) and arousal (strength of emotion).
 - Also exists in Spanish
- SentiStrength
 - Positive and negative sentiment
- Vader
 - Valence Aware Dictionary for sEntiment Reasoning

Warriner's (WKB) lexicon

- Sentiment lexicon
- Three dimensions of emotion
 - Valence
 - Arousal
 - Dominance
- Collected emotional ratings for 14,000 words, the majority of the well-known English content words
 - Participants recruited through Amazon Mechanical Turk
 - shown lists of 350 words, including 10 calibrator words
 - asked to rate each word along 3 dimensions using a 9 pt scale

[Wariner et al., (2013) “Norms of valence, arousal, and dominance for 13,915 English lemmas”,]

How do you feel while reading each word?

- Valence
 - 9: Happy, pleased, satisfied, contented, hopeful
 - 1: Unhappy, annoyed, melancholic, despaired, or bored
- Arousal
 - 9: Excited, stimulated, excited, frenzied, jittery, wide-awake
 - 1: Calm, relaxed, sluggish, dull, sleepy, or unaroused
- 5: Neutral, neither happy nor sad [not excited nor at all calm; neither in control nor controlled]



Russell's core affect theory

Thousands of mTurkers later ...

	Valence		Arousal		Dominance	
Lowest	pedophile	1.26	grain	1.60	dementia	1.68
	rapist	1.30	dull	1.67	Alzheimer's	2.00
	AIDS	1.33	calm	1.67	lobotomy	2.00
	leukemia	1.47	librarian	1.75	earthquake	2.14
	molester	1.48	soothing	1.91	uncontrollable	2.18
	murder	1.48	scene	1.95	rapist	2.21
Highest	excited	8.11	motherfucker	7.33	rejoice	7.68
	sunshine	8.14	erection	7.37	successful	7.71
	relaxing	8.19	terrorism	7.42	smile	7.72
	lovable	8.26	lover	7.45	completion	7.73
	fantastic	8.36	rampage	7.57	self	7.74
	happiness	8.48	insanity	7.79	incredible	7.74

VADER

- gold-standard sentiment lexicon that is especially attuned to microblog-like contexts.
- Rules for grammatical and syntactical conventions that humans use when expressing or emphasizing sentiment intensity
- Performs well on social media posts (better than LIWC)

Hutto, C., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media* (Vol. 8, No. 1, pp. 216-225).

VADER

- **Lexicon-Based** – Words have predefined sentiment scores (e.g., "great" = +3.2, "bad" = -2.5).
- **Modifiers & Intensifiers** – Words like "very" boost sentiment, and negation (e.g., "not good") lowers it.
- **Punctuation & Capitalization** – "GOOD!!!" has stronger positivity than "good."
- **Emoji & Slang Detection** – Recognizes "LOL" as positive and "😡" as negative.

Example:

- *"I LOVE this movie!!! It's amazing 😊"*
- **Positive:** 0.75 | **✗ Negative:** 0.00 | **○ Neutral:** 0.25
- **Compound Score:** +0.92 (Highly Positive)

Application of LIWC: Global Mood Patterns

“Diurnal and seasonal moods vary with work, sleep and daylength across diverse cultures” by Golder and Macy

- Can automated sentiment analysis be applied to social media data to provide a global picture of human mood?
- Does mood have a time scale? Seasonal patterns?
- Data
 - Up to 400 public messages from each user, 509 million messages total
 - 2.4 million individuals worldwide (84 identified countries)
 - English only
 - Date, Time, and country latitude

Methodology

- Examined within-individual Positive Affect (PA) and Negative Affect (NA) independently,
 - E.g., fraction of PA words appearing in an individual's messages every hour

$$PA_u(h) = \frac{\|PAWORDS_u(h)\|}{\|WORDS_u(h)\|}$$

- To eliminate between-individual variation, subtract the mean: $PA_u^*(h) = PA_u(h) - \langle PA_u(h) \rangle$
- Additional analysis on 4 English-speaking regions: Africa, India, UK/Aus, US/Can

Two peaks in Positive Affect daily; Positive Affect is higher on weekends

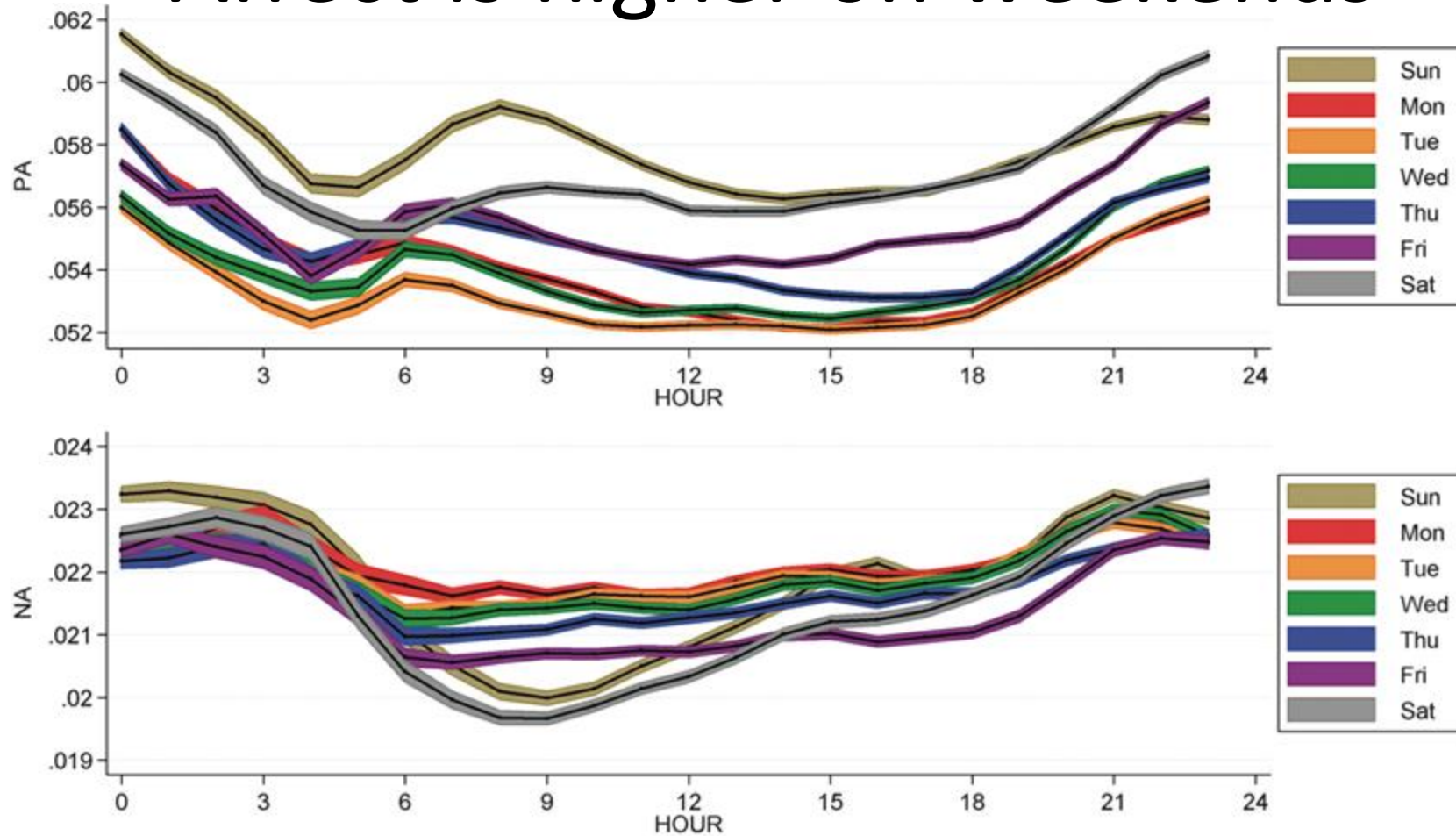


Fig. 1. Hourly changes in individual affect broken down by day of the week (top, PA; bottom, NA). Each series shows mean affect (black lines) and 95% confidence interval (colored regions).

Mood governed by diurnal cycles, not culture

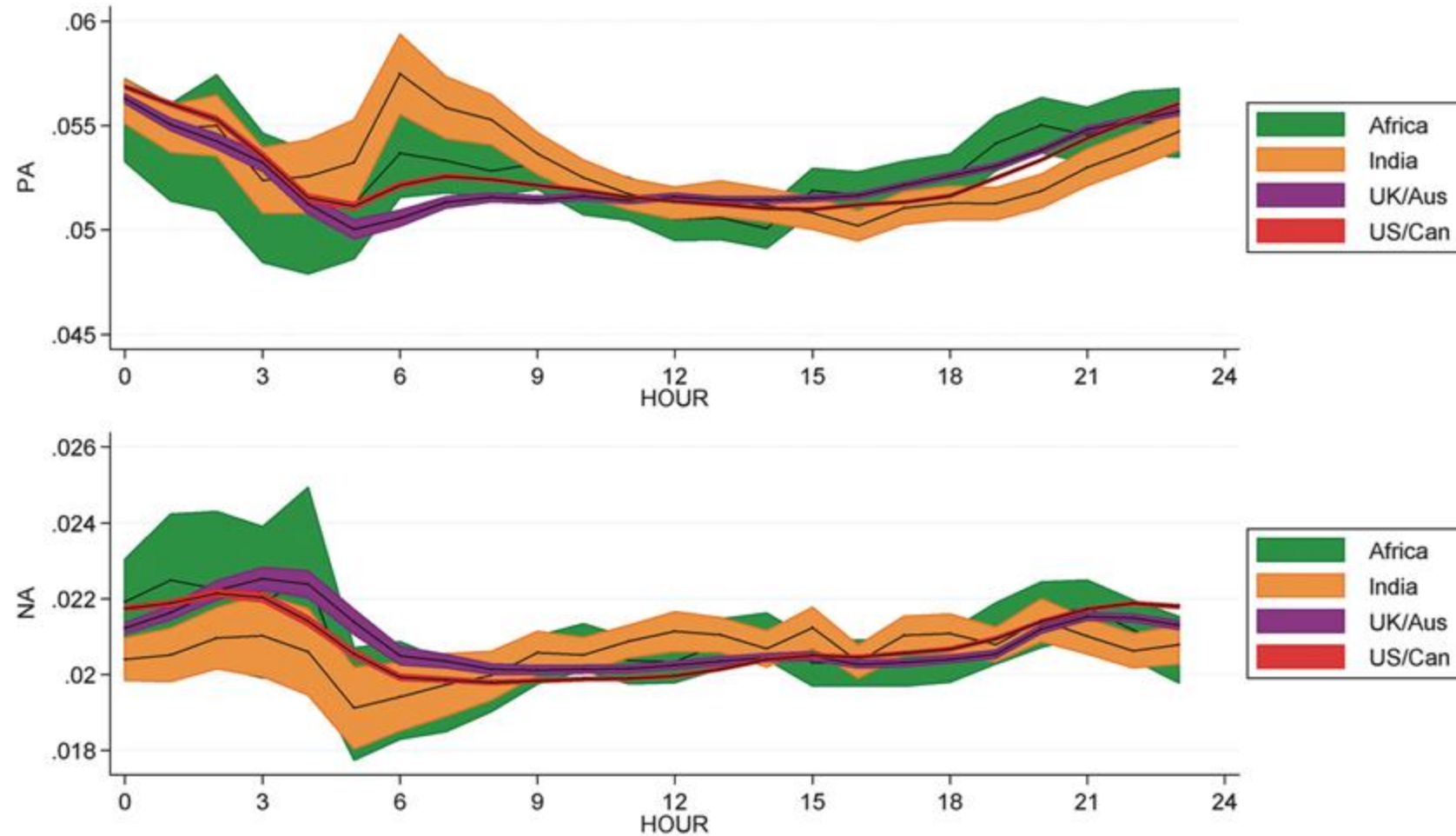


Fig. 2. Hourly changes in individual affect in four English-speaking regions. Each series shows mean affect (black lines) and 95% confidence interval (colored regions).

High-level text analysis

Understanding text: high level understanding



- Machine translation
- Generate text (ChatGPT)
- Text summarization
- Question-answering


English

↔


Turkish


The idea of machine translation later appeared in the 17th century.






67 / 5,000 

Makine çevirisi fikri daha sonra 17. yüzyılda ortaya çıktı.







[Send feedback](#)

Understanding text: high level

- Machine translation
- Generate text (ChatGPT)
- Text summarization
- Question-answering



Generate new text after this sentence.



Certainly! Please provide me with the sentence you would like me to generate new text after.

Understanding text: high level

- Machine translation
- Generate text (ChatGPT)
- Text summarization
- Question-answering

TW

Summarize this text into one sentence:

Natural language processing (NLP) is an interdisciplinary subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

Challenges in natural language processing frequently involve speech recognition, natural-language understanding, and natural-language generation.



Natural language processing (NLP) involves programming computers to analyze and process large amounts of human language data, aiming to enable computers to understand document contents, extract information, and categorize documents, with challenges including speech recognition, natural-language understanding, and natural-langua



Regenerate response

Screenshot

Understanding text: high level

- Machine translation
- Generate text (ChatGPT)
- Text summarization
- Question-answering



What is the periodic table number of Technetium?



The periodic table number of Technetium is 43.



Vector space model

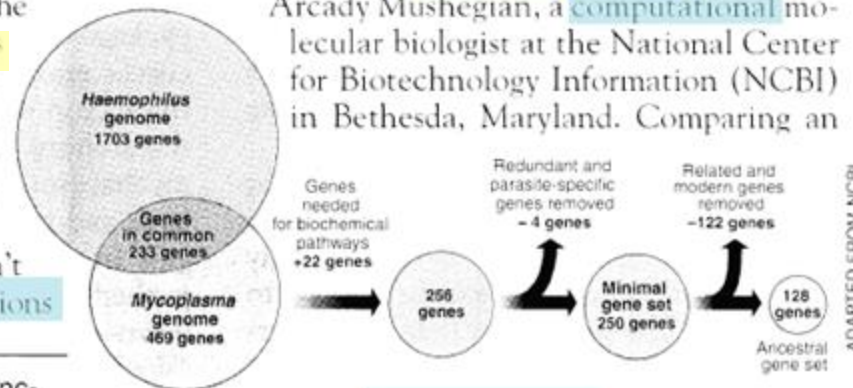
Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

Term frequency

- genes 5
- organism 3
- survive 1
- life 1
- computer 1
- organisms 1
- genomes 2
- predictions 1
- genetic 1
- numbers 1
- sequenced 1
- genome 2
- computational 1

• ...



Vector space models: reducing noise

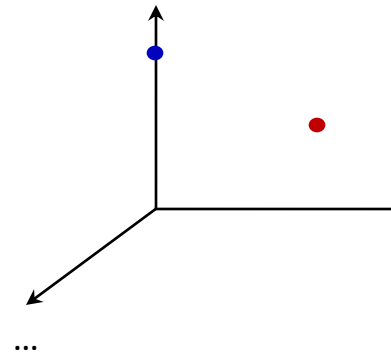
original	stem words	remove stopwords
<ul style="list-style-type: none">• genes 5• organism 3• survive 1• life 1• computer 1• organisms 1• genomes 2• predictions 1• genetic 1• numbers 1• sequenced 1• genome 2• computational 1	<ul style="list-style-type: none">• gene 6• organism 4• survive 1• life 1• comput 2• predictions 1• numbers 1• sequenced 1• genome 4	<ul style="list-style-type: none">• and• or• but• also• to• too• as• can• I• you• he• she• ...



Vector space model

- Each document is a point in high-dimensional space

Document 2
gene 0
organism 6
survive 1
life 1
comput 2
predictions 1
numbers 1
sequenced 1
genome 4
...



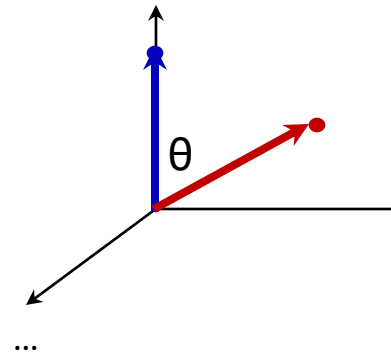
Document 1
gene 6
organism 4
survive 1
life 1
comput 2
predictions 1
numbers 1
sequenced 1
genome 4
...



Vector space model

- Each document is a point in high-dimensional space

Document 2
gene 0
organism 6
survive 1
life 1
comput 2
predictions 1
numbers 1
sequenced 1
genome 4
...



Document 1
gene 6
organism 4
survive 1
life 1
comput 2
predictions 1
numbers 1
sequenced 1
genome 4
...

- Compare two documents: similarity $\sim \cos(\theta)$



Improving the vector space model

- Use tf-idf, instead of term frequency (tf), in the document vector
 - Term frequency * inverse document frequency
 - E.g.,
 - ‘computer’ occurs 3 times in a document, but it is present in 80% of documents ? tf-idf score ‘computer’ is $3 * 1/.8 = 3.75$
 - ‘gene’ occurs 2 times in a document, but it is present in 20% of documents ? tf-idf score of ‘gene’ is $2 * 1/.2 = 10$

$$TF = \frac{\text{count of word in document}}{\text{total words in document}}$$

$$IDF = \log \left(\frac{\text{total number of documents}}{\text{number of documents with the word}} \right)$$

$$TF-IDF = TF \times IDF$$



Some problems with vector space model

- Synonymy
 - Unique term corresponds to a dimension in term space
 - Synonyms ('kid' and 'child') are different dimensions
- Polysemy
 - Different meanings of the same term improperly confused
 - E.g., document about river 'banks' will be improperly judged to be similar to a document about financial 'banks'

Latent Semantic Indexing



- Identifies subspace of tf-idf that captures most of the variance in a corpus
 - Need a smaller subspace to represent document corpus
 - This subspace captures topics that exist in a corpus
 - Topic = set of related words



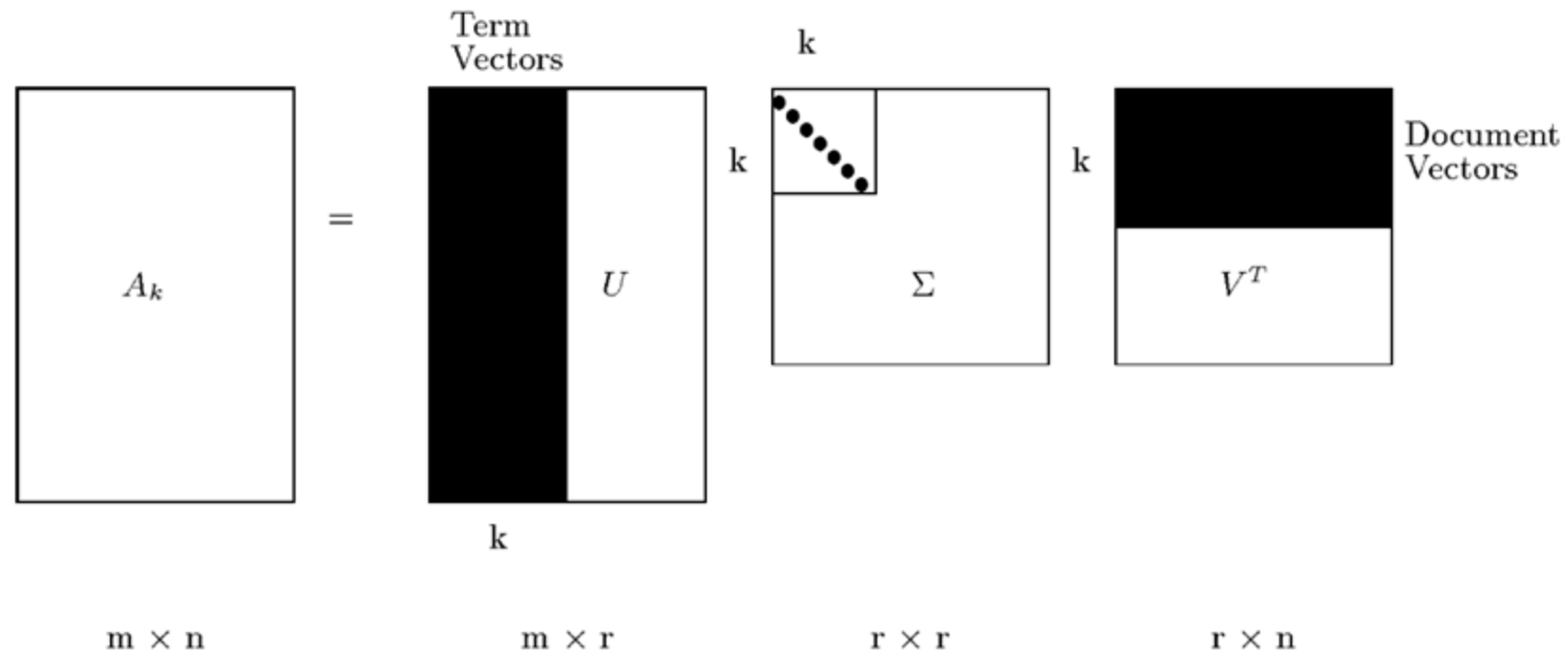
LSI, the Method

- Encode documents in a (document-term) matrix A
- Factorize
 - Decompose A by Singular Value Decomposition (SVD)
 - Linear algebra
- Approximate A using truncated SVD
 - Captures the most important relationships in A
 - Ignores other relationships
 - Rebuild the matrix A using just the important relationships
- Measure relatedness
 - Cosine of latent factors

$$A = \begin{pmatrix} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \text{cosmonaut} & 1 & 0 & 1 & 0 & 0 & 0 \\ \text{astronaut} & 0 & 1 & 0 & 0 & 0 & 0 \\ \text{moon} & 1 & 1 & 0 & 0 & 0 & 0 \\ \text{car} & 1 & 0 & 0 & 1 & 1 & 0 \\ \text{truck} & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$



LSI, the Method (cont.)



Each row and column of A gets mapped into the k -dimensional LSI space, by the SVD.



Lower rank decomposition

- Usually, rank of the matrix A is small: $r \ll \min(m, n)$.
 - Only a few of the largest eigenvectors (those associated with the largest eigenvalues λ) matter
 - These r eigenvectors define a lower dimensional subspace that captures most important characteristics of the document corpus
 - All operations (document comparison, similar) can be done in this reduced-dimension subspace



Topic Models

- Generative probabilistic modeling
 - Treats data as observations
 - Contains hidden variables
 - Hidden variables reflect the themes that pervade a corpus of documents
- Infer hidden thematic structure
 - Analyze words in the documents
 - Discover topics in the corpus
 - A topic is a distribution over words
 - Large reduction in description length
 - Few topics are needed to represent themes in a document corpus – about 100



LDA – Latent Dirichlet Allocation (Blei 2003)

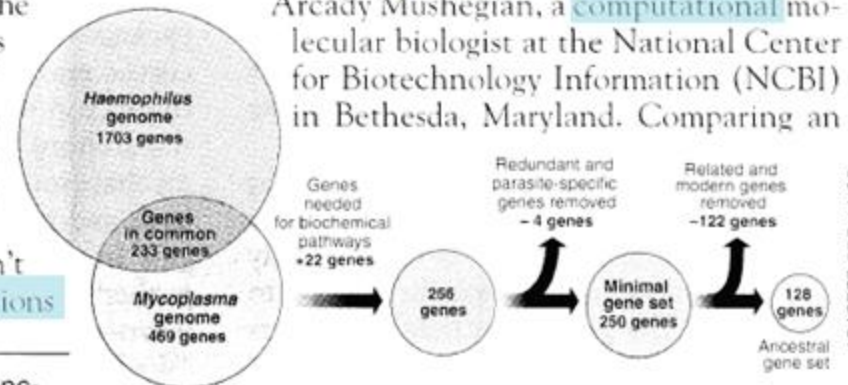
Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many **genes** does an **organism** need to **survive**? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for **life**. One research team, using **computer** analyses to compare known **genomes**, concluded that today's **organisms** can be sustained with just 250 genes, and that the earliest life forms required a mere 128 **genes**. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those **predictions**

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

“are not all that far apart,” especially in comparison to the 75,000 **genes** in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a **genetic numbers game**, particularly as more and more **genomes** are completely mapped and sequenced. “It may be a way of organizing any newly **sequenced genome**,” explains Arcady Mushegian, a **computational** molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



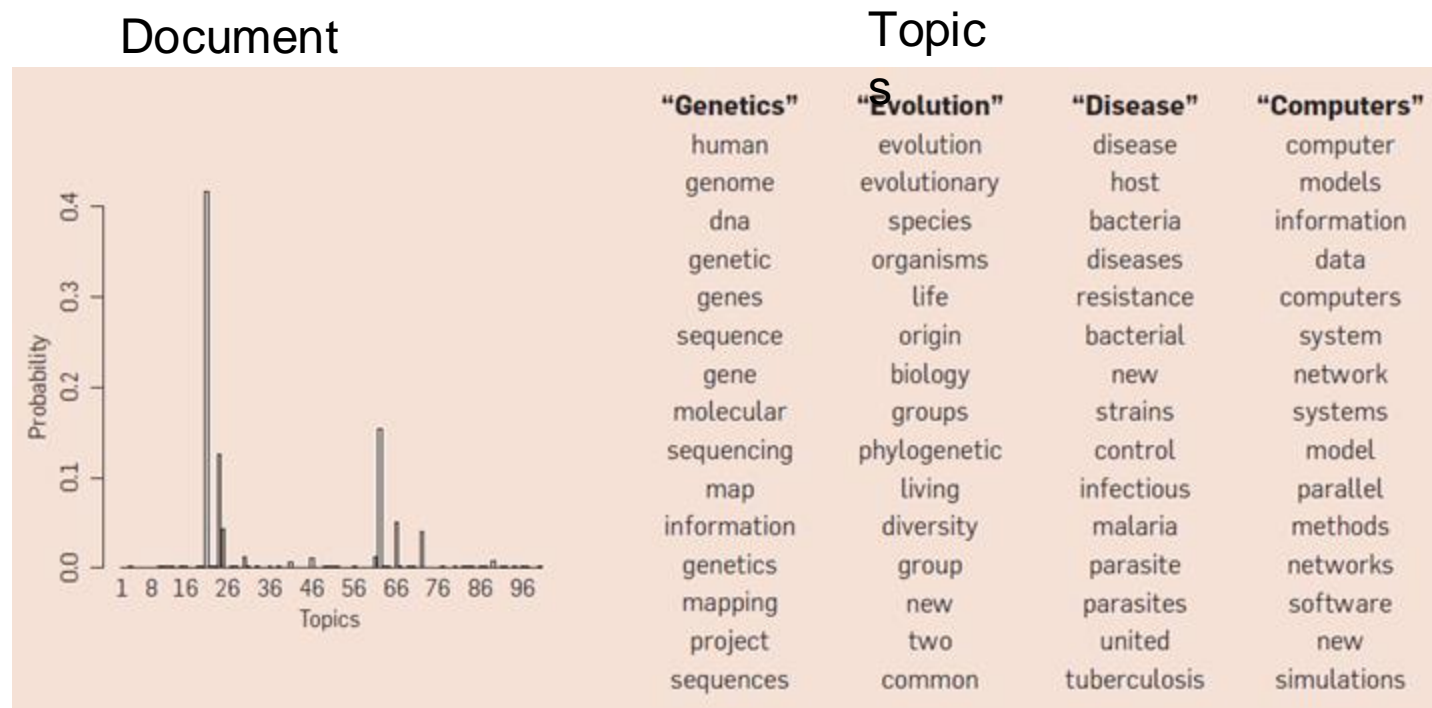
Stripping down. **Computer analysis** yields an estimate of the minimum modern and ancient genomes.

Intuition: Documents have multiple topics 272 • 24 MAY 1996



Topics

- A topic is a distribution over words
- A document is a distribution over topics
- A word in a document is drawn from one of those topics

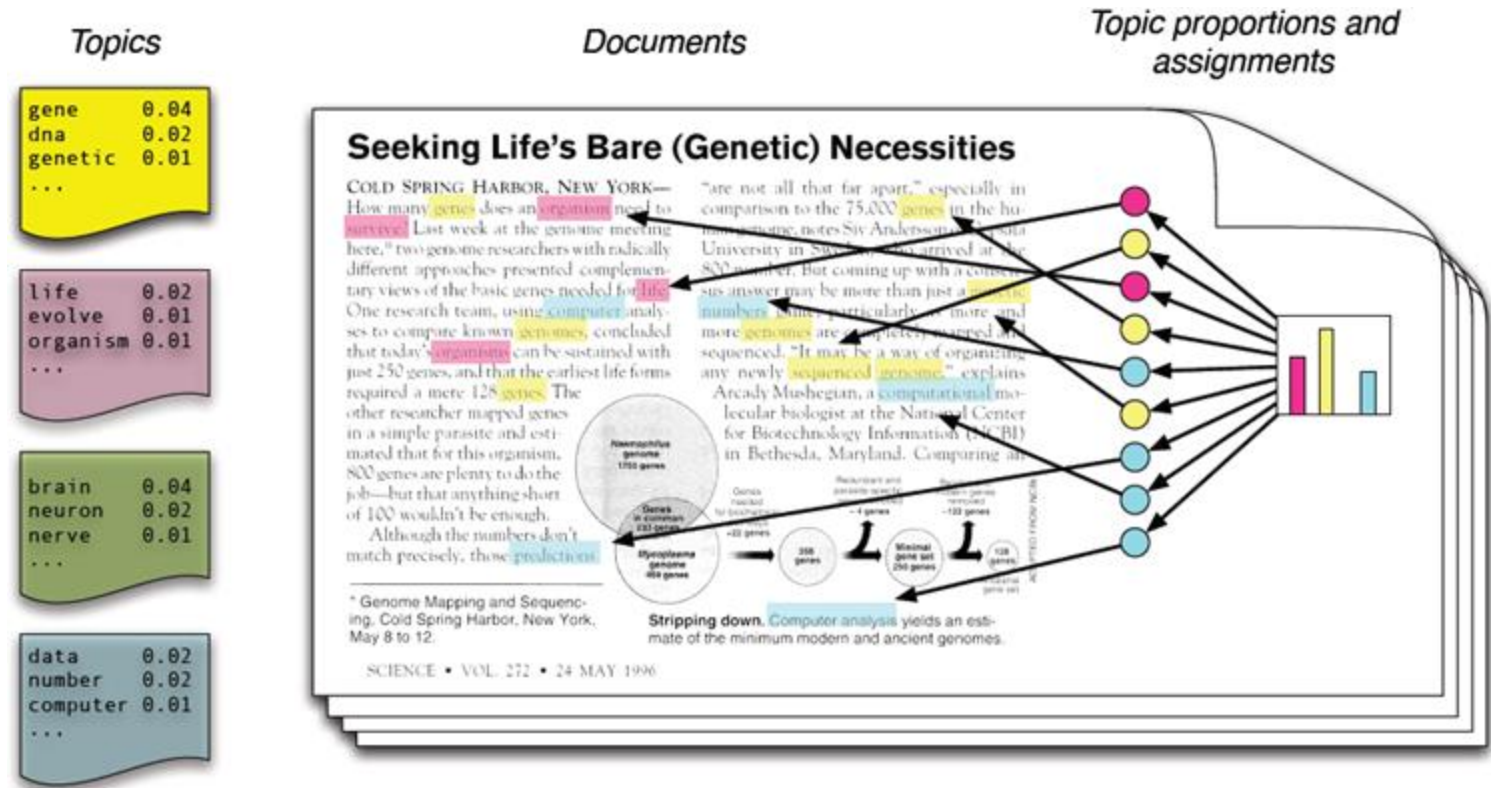


Generative Model of LDA



LDA assumes that:

- Each document has a mixture of topics.
- Each topic is a probability distribution over words.
- Words in a document are associated with a set of topics

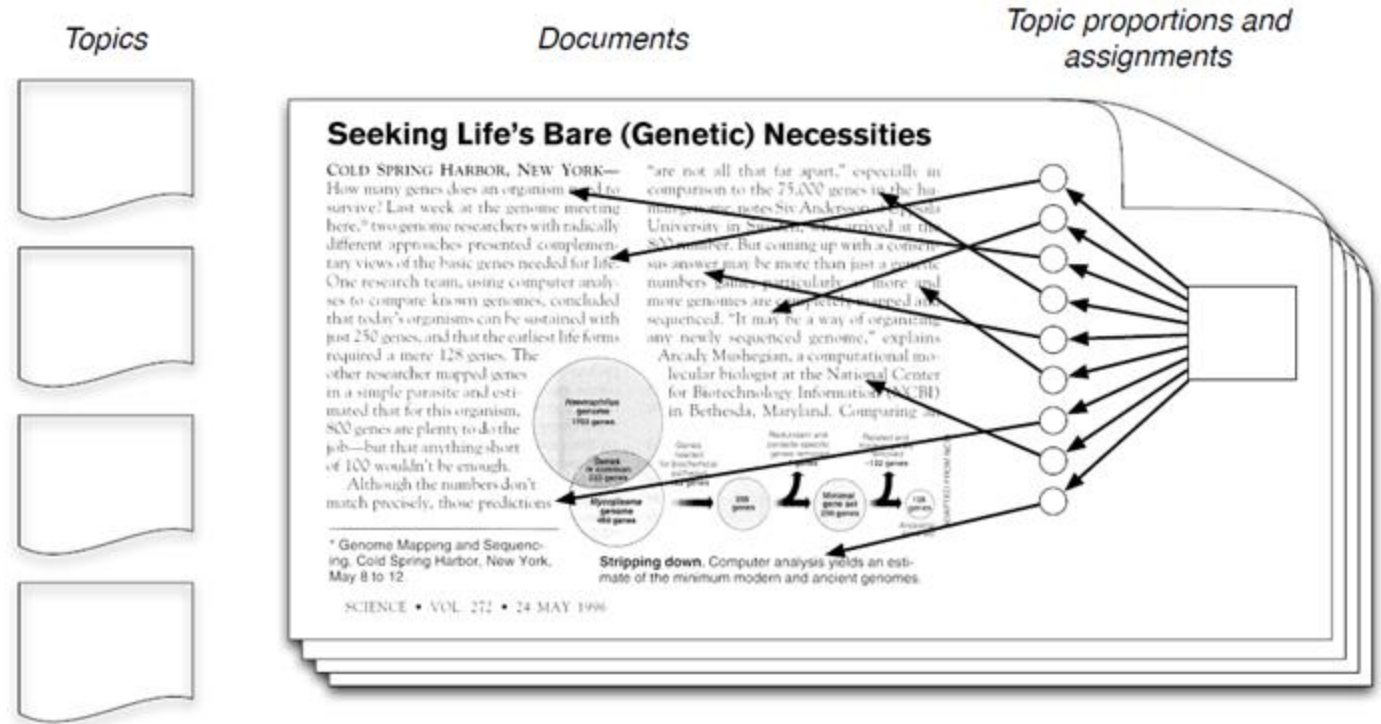




LDA inference

We observe **documents** and **words** but **not the underlying topic assignments**. The goal is to infer:

- Topic distribution per document θ_d
- Word distribution per topic ϕ_k
- Topic assignment for each word in each document $Z_{d,n}$



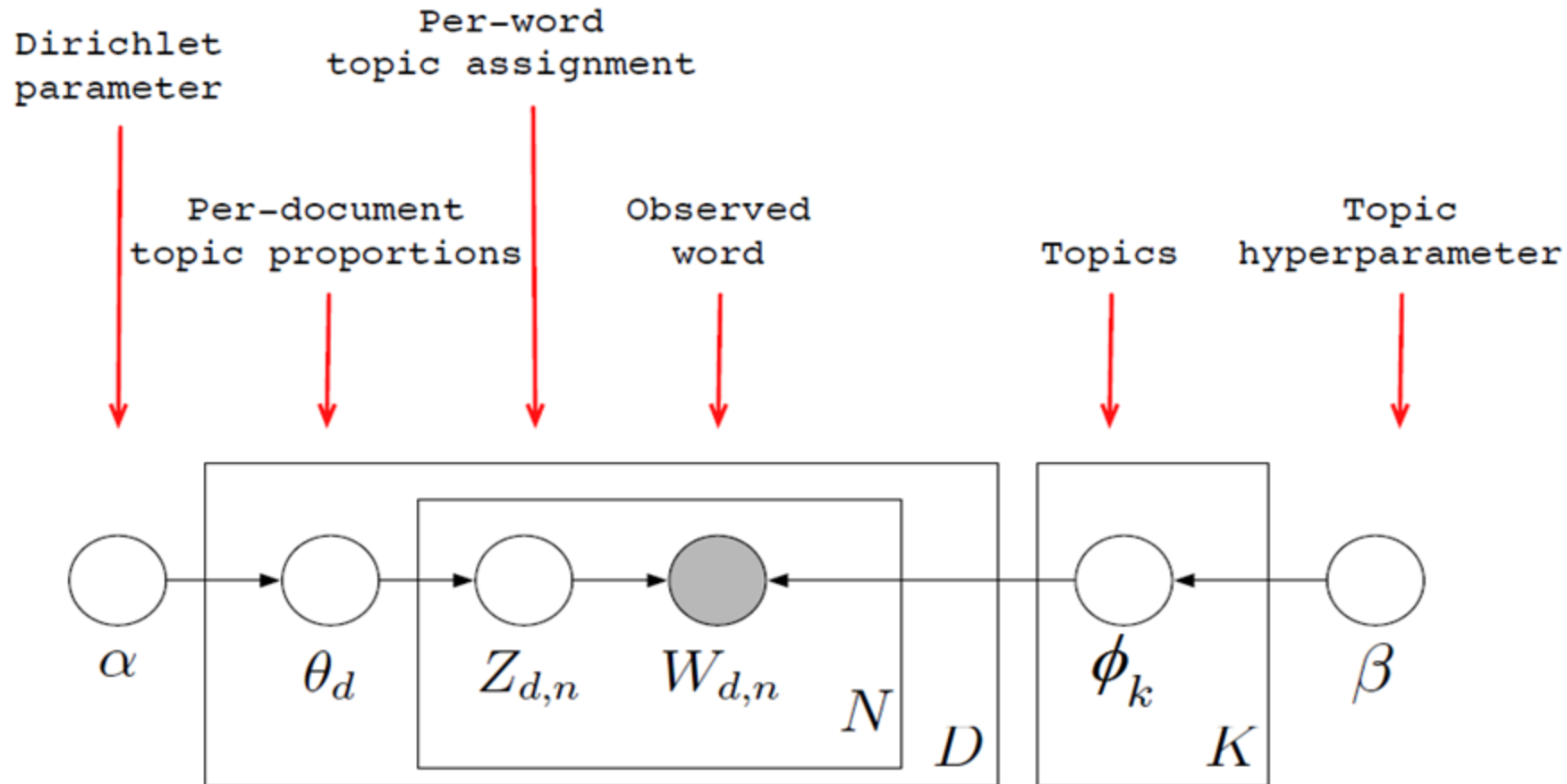


Dirichlet Distributions

- The Dirichlet distribution is a probability distribution of probability of >2 outcomes.
 - A generalization of the Beta distribution (2 outcomes)
 - The topic proportions θ_d sum to 1 for each document.
 - The word proportions ϕ_k sum to 1 for each topic.
 - Two parameters : alpha and beta
 - Alpha controls number of topics per document. $\theta_d \sim \text{Dir}(\alpha)$
 - Beta controls number of words per topic. $\phi_k \sim \text{Dir}(\beta)$



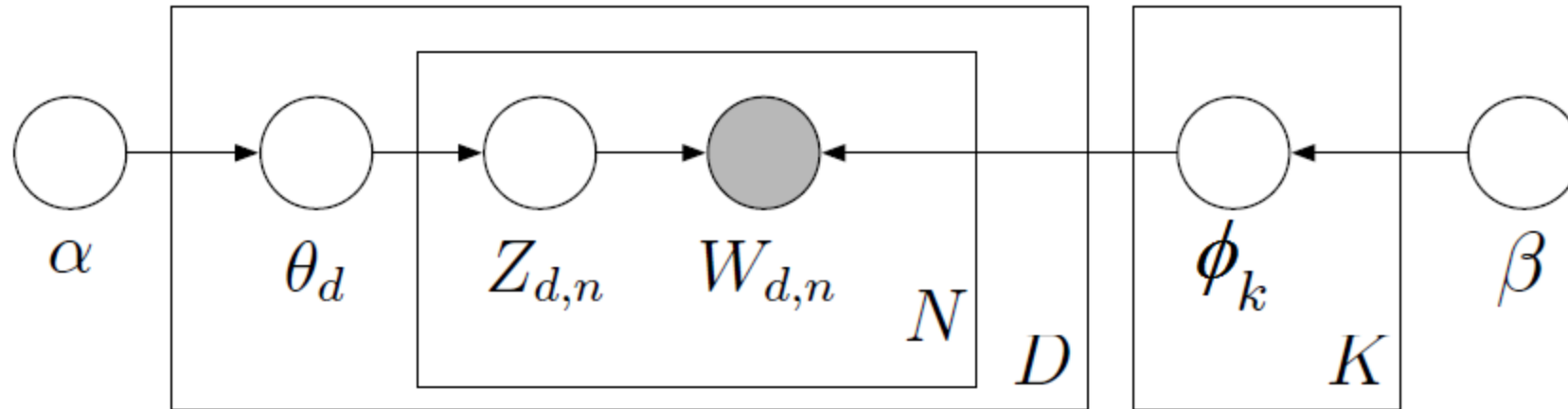
LDA as Graphical Model - Plate Representation



- Nodes are random variables; edges indicate dependence
- Shaded nodes are observed; unshaded nodes are latent



Posterior Distribution



- Our task: Identify the joint distribution $p(\boldsymbol{\vartheta}, \mathbf{z}, \boldsymbol{\theta} | \mathbf{W})$:
- From a collection of words \mathbf{W} in documents, infer
 - Per-word topic assignment $\mathbf{z}_{d,n}$
 - Per-document topic proportions $\boldsymbol{\vartheta}_d$
 - Per-topic word distribution $\boldsymbol{\phi}_k$

Computing this posterior distribution is intractable as there are many possible combinations of word-topic and topic-documents.



Approximate posterior inference algorithms

- Mean field variational methods
- Expectation maximization
- Gibbs sampling
- Distributed sampling
- ...
- Efficient packages for solving this problem



Illustration

- Data: collection of *Science* articles from 1990-2000
 - 17K documents
 - 11M words
 - 20K unique words (stop words and rare words removed)
- Model: 100-topic LDA



1 dna gene sequence genes sequences human genome genetic analysis two	2 protein cell cells proteins receptor fig binding activity activation kinase	3 water climate atmospheric temperature global surface ocean carbon atmosphere changes	4 says researchers new university just science like work first years	5 mantle high earth pressure seismic crust temperature earths lower earthquakes
6 end article start science readers service news card circle letters	7 time data two model fig system number different whole one	8 materials surface high structure temperature molecules chemical molecular fig university	9 dna rna transcription protein site binding sequence proteins specific sequences	10 disease cancer patients human gene medical studies drug normal drugs
11 years million ago age university north early fig evidence record	12 species evolution population evolutionary university populations natural studies genetic biology	13 protein structure proteins two amino binding acid residues molecular structural	14 cells cell virus hiv infection immune human antigen infected viral	15 space solar observations earth stars university mass sun astronomers telescope
16 tax manager science aaas advertising sales member recruitment associate washington	17 cells cell gene genes expression development mutant mice fig biology	18 energy electron state light quantum physics electrons high laser magnetic	19 research science national scientific scientists new states university united health	20 neurons brain cells activity fig channels university cortex neuronal visual



Quality of Topics - Topic Coherence

- Gather top N words for each topic
- We evaluate how often words in the top N list appear together in a reference corpus.
- For a pair of words w_i and w_j , their co-occurrence can be computed using: Pairwise Mutual Information, Word Embedding similarity etc
- Calculate average score across all pairwise calculations.



Tools

- Topic modeling
 1. Blei's LDA w/ "variational method" (<http://cran.r-project.org/web/packages/lda/>) or
 2. "Gibbs sampling method" (<https://code.google.com/p/plda/> and <http://gibbslda.sourceforge.net/>)
 3. Gensim LDA implementation
<https://radimrehurek.com/gensim/models/ldamodel.html>
 4. <https://www.seas.upenn.edu/~cis520/lectures/LDA.pdf>



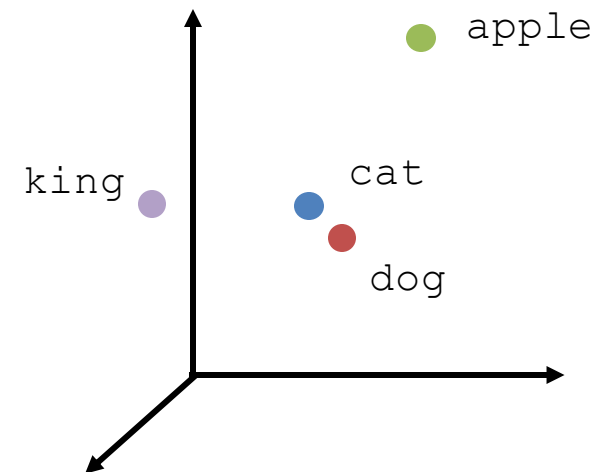
TEXT EMBEDDINGS



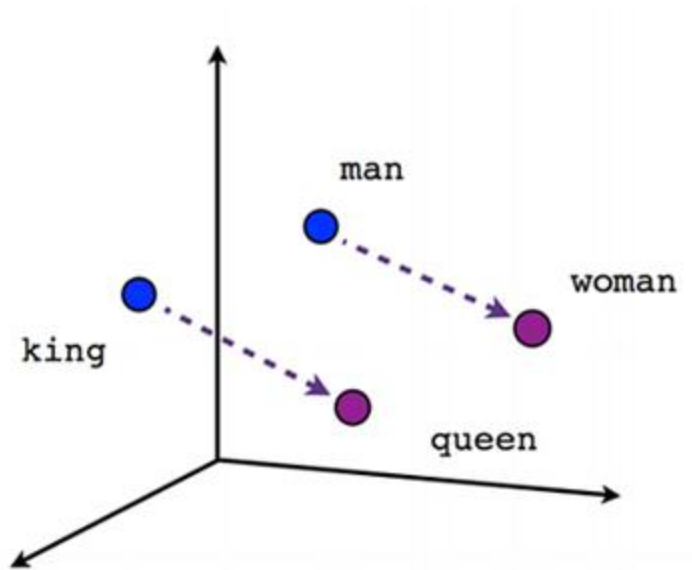
Text embeddings

- Text embeddings convert words or phrases into dense vectors of numbers
- Each word is represented by a list of numbers in a lower-dimension space (typically ~1000 dimensions compared to 50K words): dimensionality reduction
- The position in this vector space captures semantic meaning
 - Words with similar meanings appear close together in the vector space

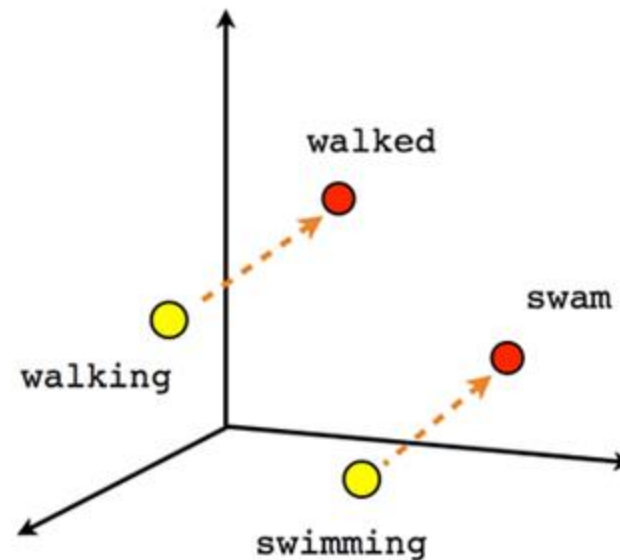
The cat sat on a mat.
The dog sat on a mat.
The king sat on a throne.



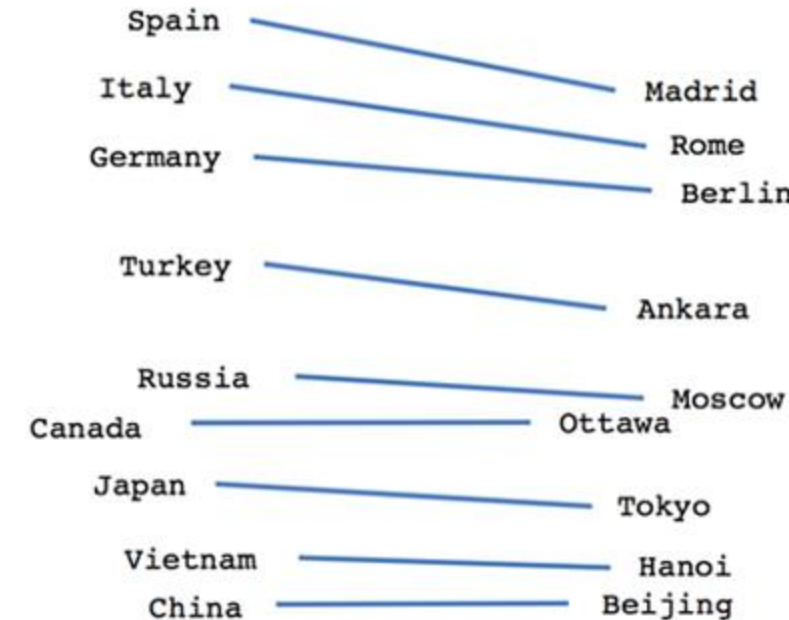
Distances represent semantic relationships



Male-Female



Verb tense



Country-Capital

Text embedding

- Words (FastText, Word2Vec, BERT, etc.)
 - Word -> [0,0.1,0.2,0.1,0.01,...]
- Sentences (Sentence-BERT, MPNet, etc.)
 - “I went for a walk” -> [0.2,0.01,0.02,0.05,0.1,...]
- Documents (Doc2Vec, Longformer, etc.)
 - "Well, Prince, so Genoa and Lucca are now just family estates of the Buonapartes. But I warn you, if you don't tell me that this means war..."
-> [0.5,0.1,0.2,0.5,0.1,...]

Word Embedding



- Words from the vocabulary are mapped to vectors of real numbers:
 - In a low dimensional space, relative to the vocabulary size.
 - "continuous space".

Old (Vector Space): Word Embeddings:

the: $\langle 1, 0, 0 \rangle$

cat: $\langle 0, 1, 0 \rangle$

dog: $\langle 0, 0, 1 \rangle$

the: $\langle 0.45, 0.89 \rangle$

cat: $\langle 0.70, 0.71 \rangle$

dog: $\langle 0.16, 0.98 \rangle$

Word embeddings

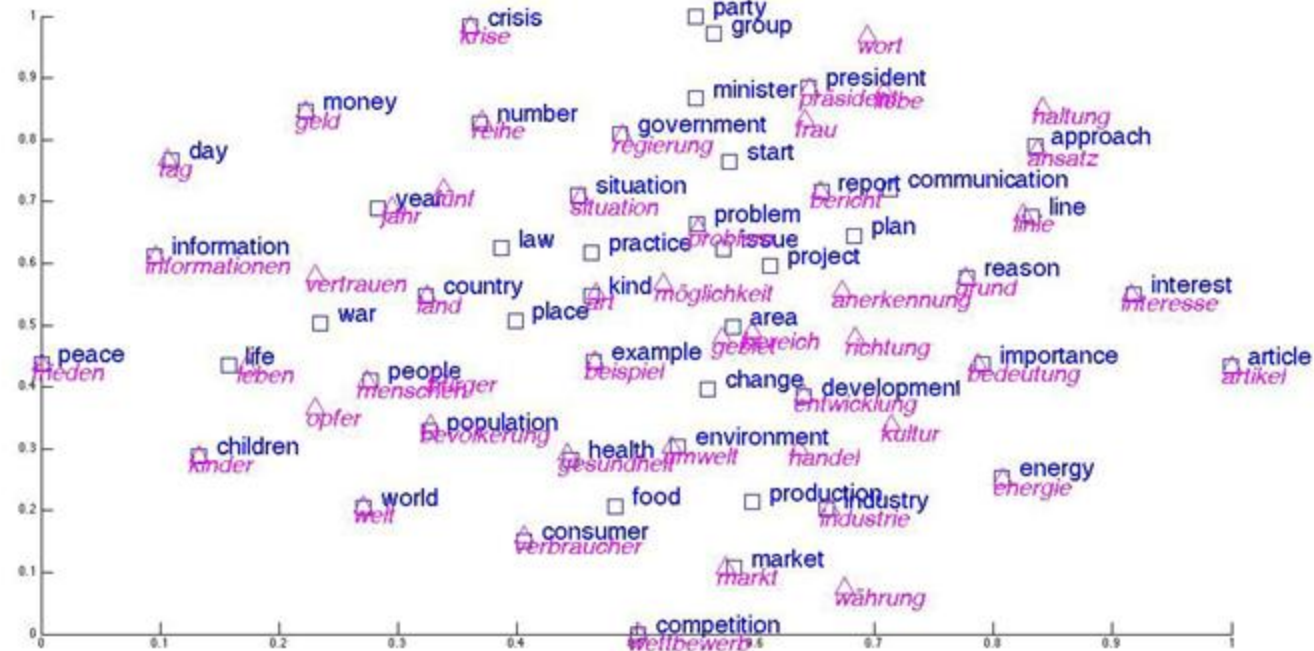


- Intuition:
 - Words that **appear alongside each other** should be **close**.
 - Words that **do not appear alongside each other** should be **far away**.
- “Alongside each other”
 - Words that appear next to each other.
 - “next to” -> within some window.
 - Window size is a parameter, let’s call it **W**.
- “Do not appear alongside each other”
 - Never appear within a window together.
 - “Negative sampling”
 - Number of negative samples is a parameter, let’s call it **N**.
- Distance is obtained with the dot product.

Word Embedding: Applications



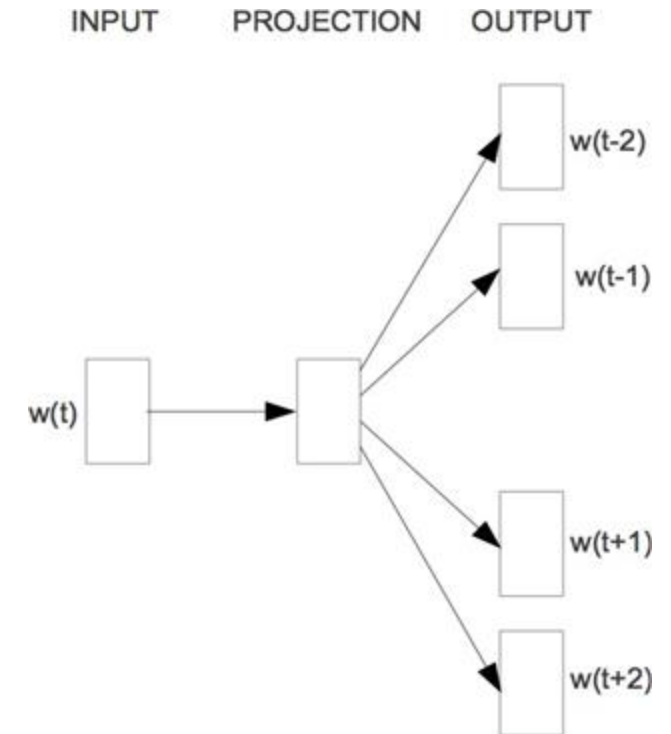
- Sentiment Analysis
- Machine Translation
- Music/Video Recommendation
- ...





Word2Vec

- A compact representation of co-occurrence matrix
- **Word2Vec**: Predict surrounding words
 - Similar to using co-occurrence counts
Levy&Goldberg (2014), Pennington et al. (2014)
- Easy to incorporate new words or sentences





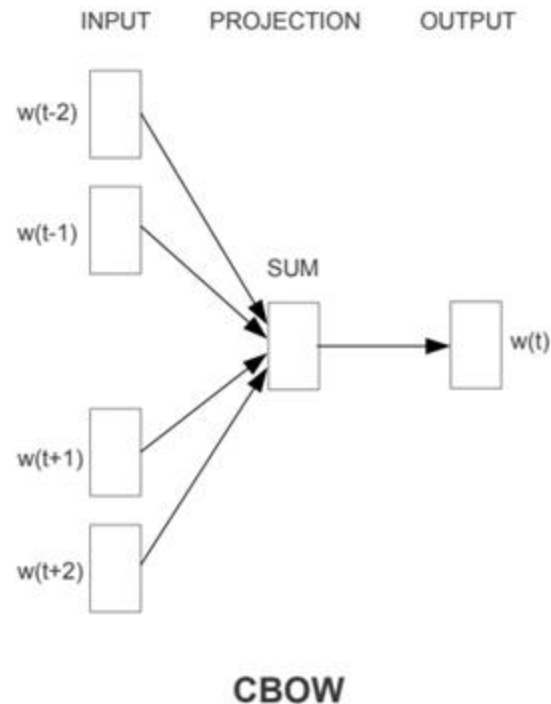
Word2Vec

- **Idea:** words that are semantically similar often occur near each other in text
- Embeddings that are good at predicting neighboring words are also good at representing similarity
- “You shall know a word by the company it keeps” - J. R. Firth



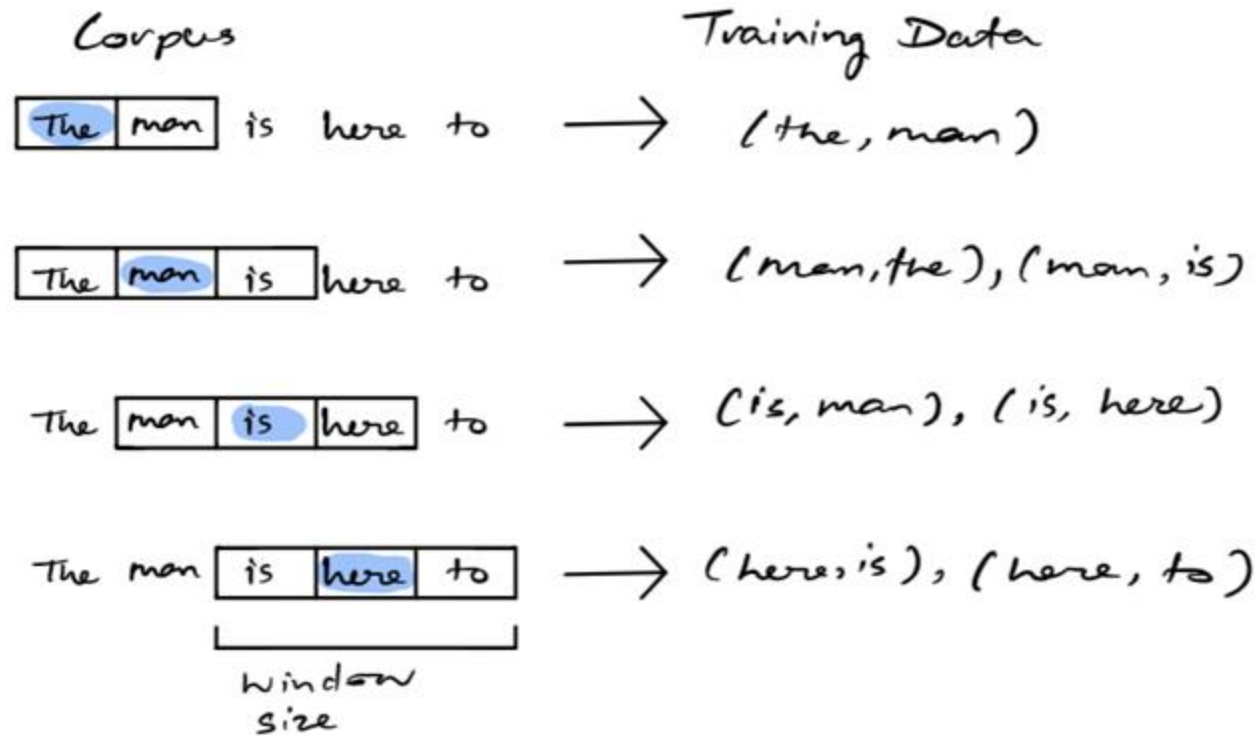
How it works

- Ex: Word2Vec
 - **CBOW (Continuous Bag of Words):** try to predict target word from context
 - E.g., in a sentence "The cat sat on the mat", the model tries to predict "cat" given words like "the" and "sat".



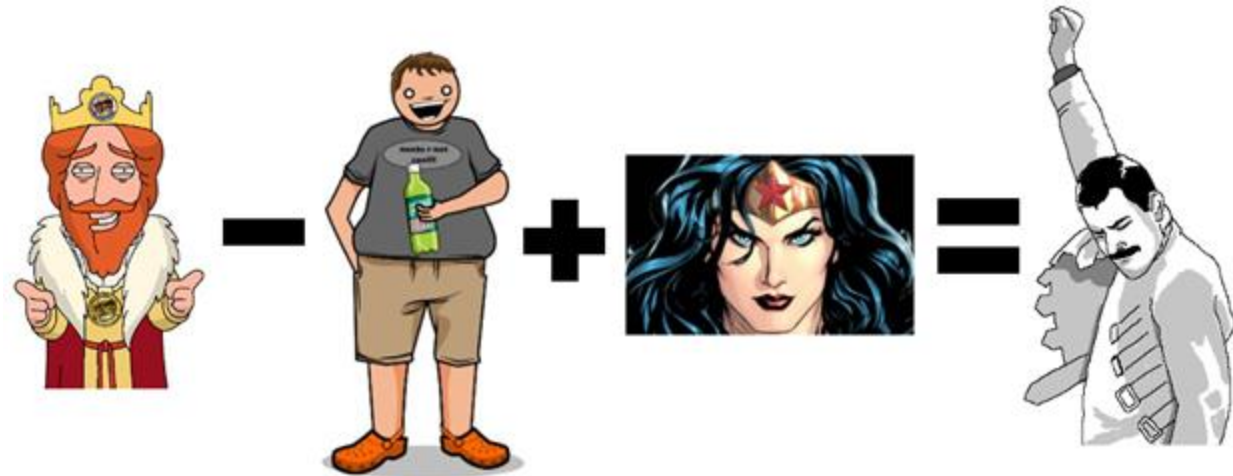
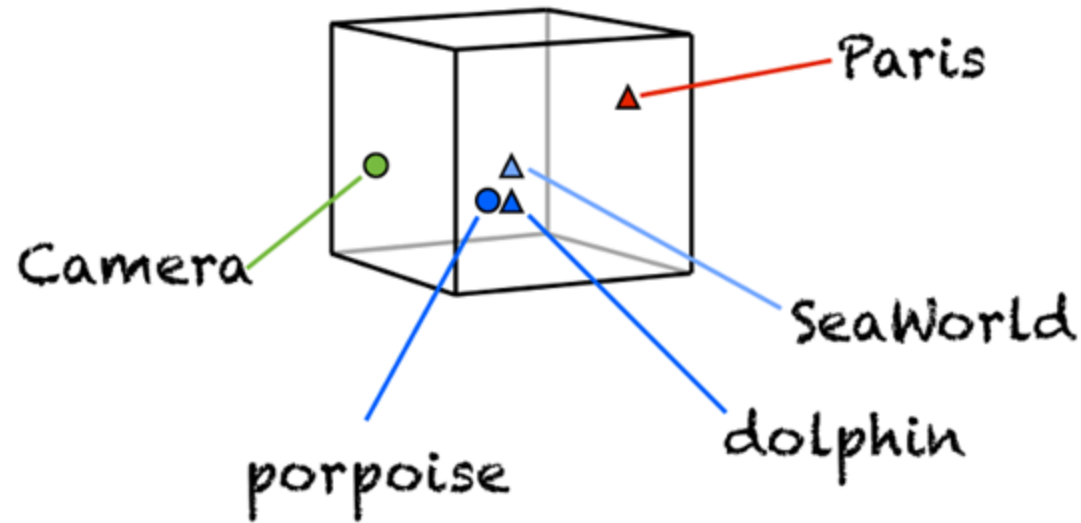
How it works

- Ex: Word2Vec
 - Continuous Skip-Gram Model: predict words around target word

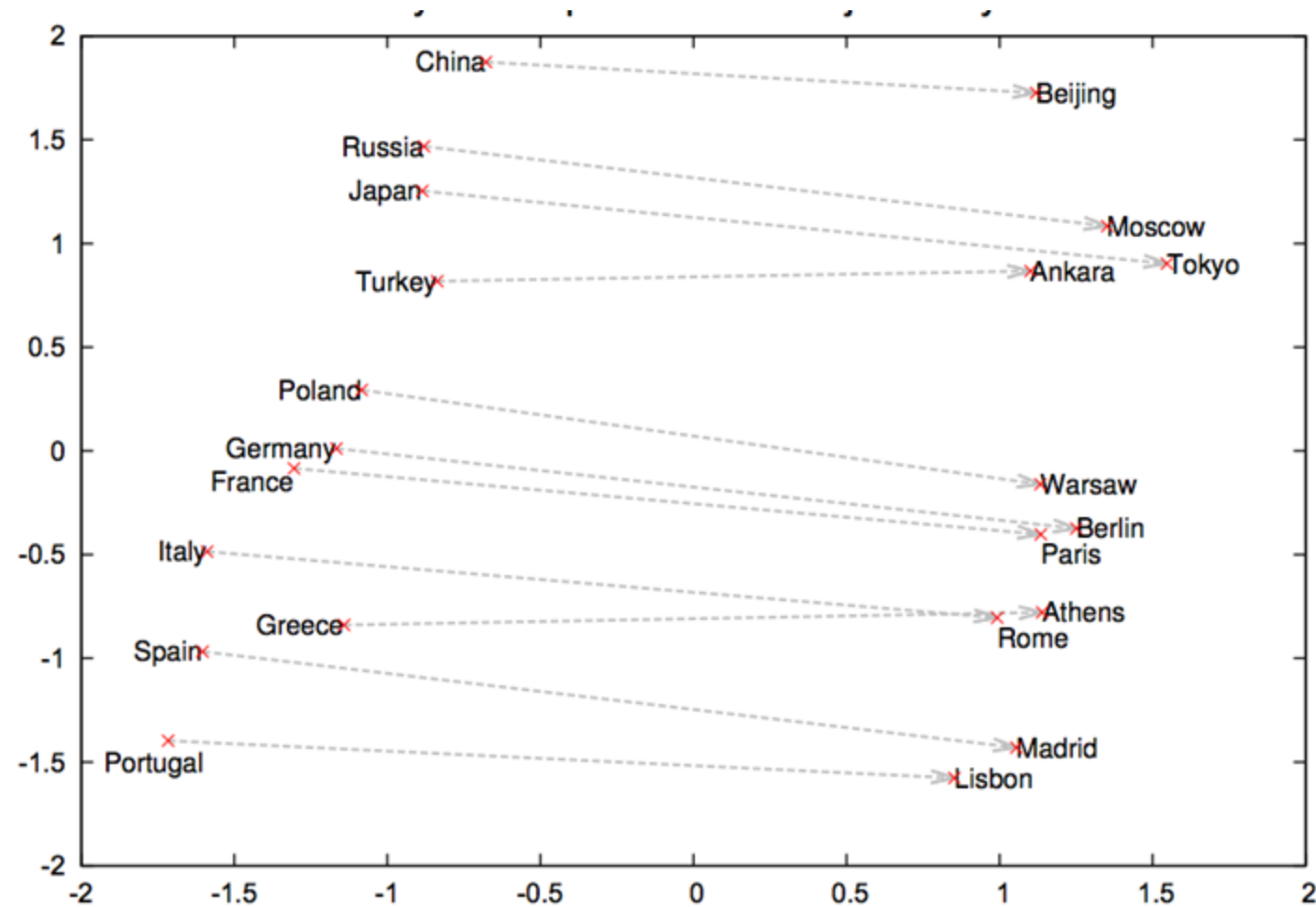


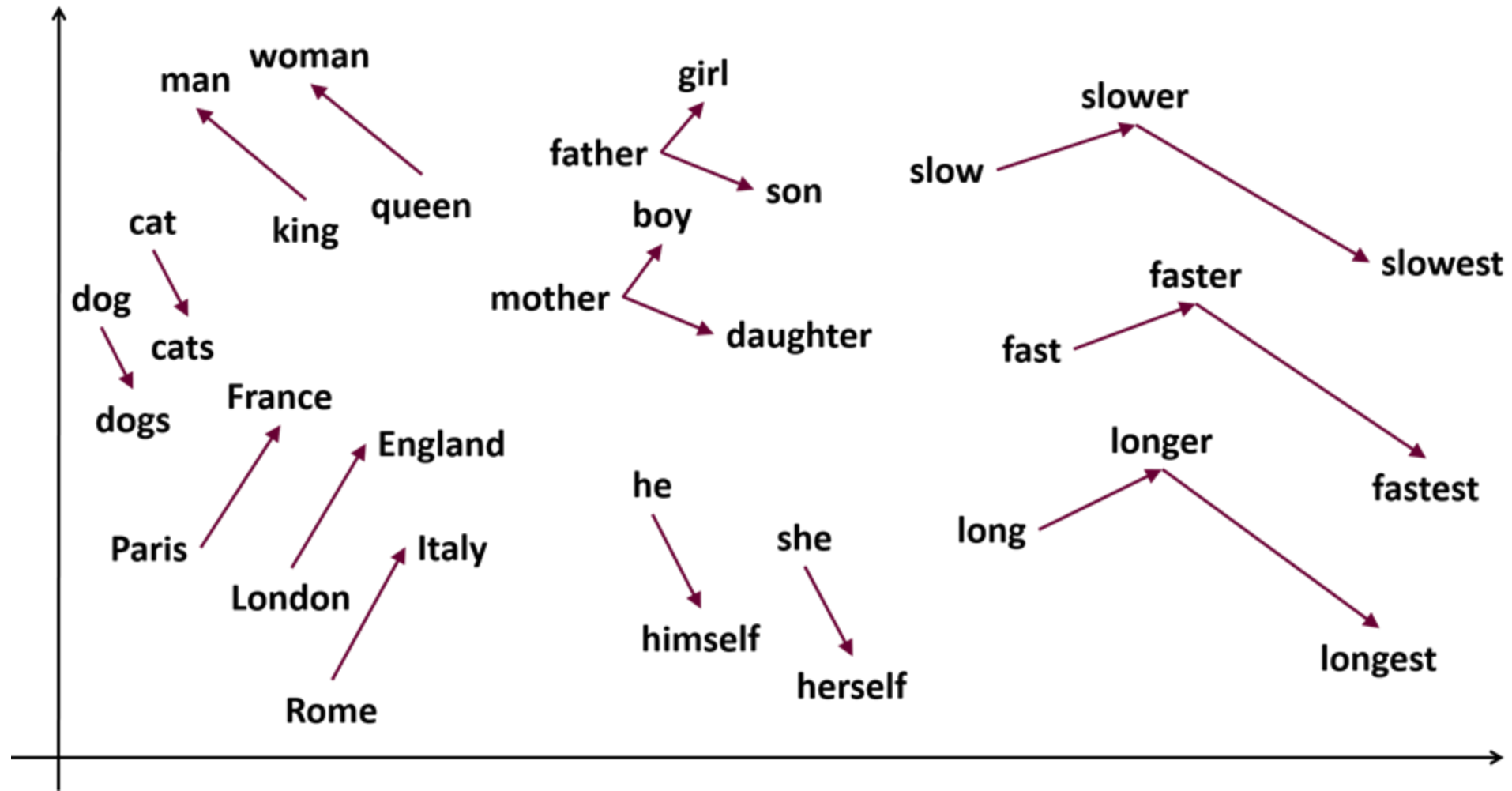


Nice properties of word embeddings



Analogies





Which concepts are closer in the embeddings space?

- **Flowers:** aster, clover, hyacinth, marigold, poppy, azalea, crocus, iris, orchid, rose, bluebell, daffodil, lilac, pansy, tulip, buttercup, daisy, lily, peony, violet, carnation, gladiola, magnolia, petunia, zinnia.
- **Insects:** ant, caterpillar, flea, locust, spider, bedbug, centipede, fly, maggot, tarantula, bee, cockroach, gnat, mosquito, termite, beetle, cricket, hornet, moth, wasp, blackfly, dragonfly, horsefly, roach, weevil.
- **Pleasant:** caress, freedom, health, love, peace, cheer, friend, heaven, loyal, pleasure, diamond, gentle, honest, lucky, rainbow, diploma, gift, honor, miracle, sunrise, family, happy, laughter, paradise, vacation.
- **Unpleasant:** abuse, crash, filth, murder, sickness, accident, death, grief, poison, stink, assault, disaster, hatred, pollute, tragedy, divorce, jail, poverty, ugly, cancer, kill, rotten, vomit, agony, prison.



TRANSFORMERS AND LANGUAGE MODELS

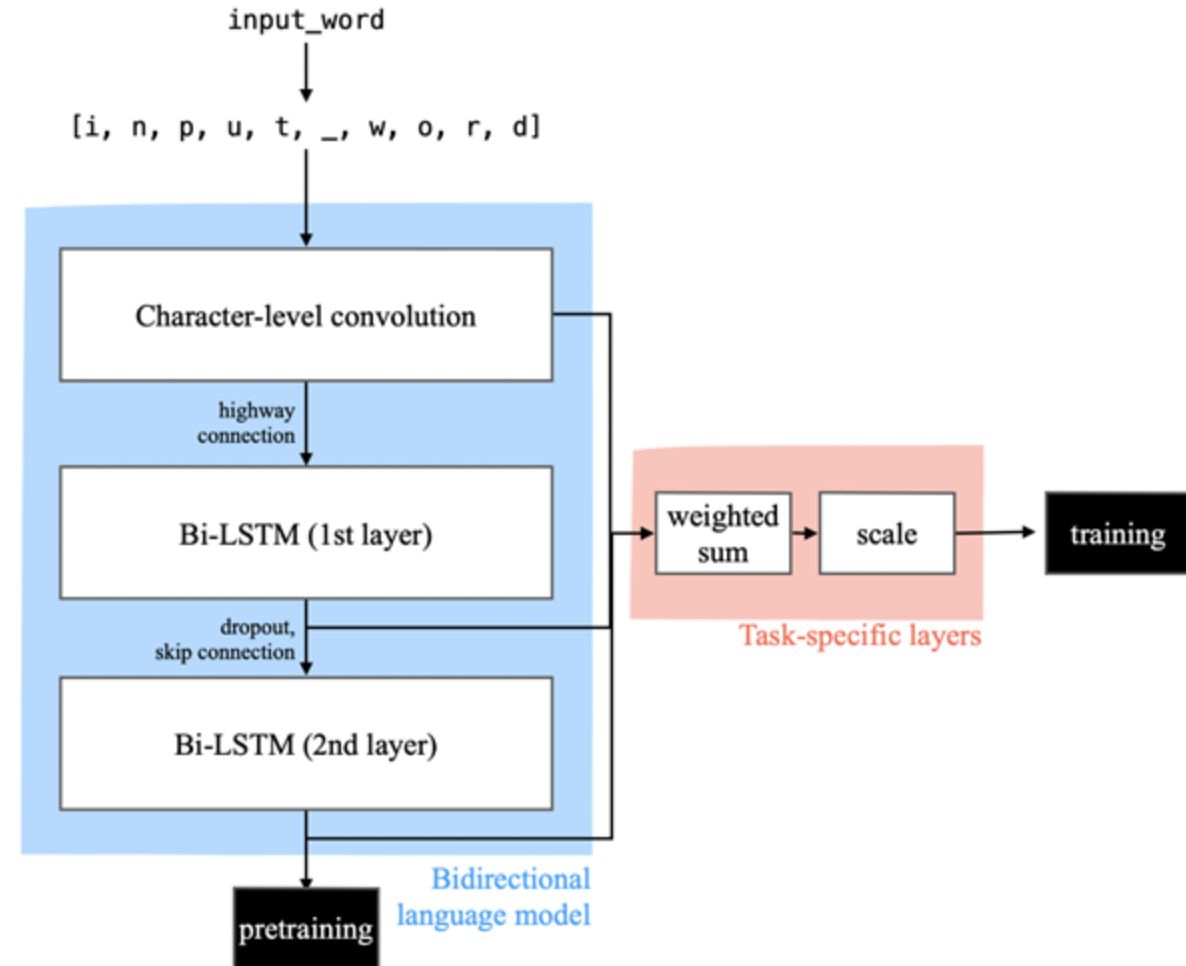
Slides from Hang Dong



ELMo: Embeddings from Language Model

(Peters *et al.*, 2018)

- Instead of using the same embedding vector for each word, ELMo creates different representations for the same word based on its context. For example, "bank" would get different representations in "river bank" versus "bank account." Handles polysemy.
- Trained to predict the **next** word in a sequence, and **previous** word as well (context)
 - The next word given previous words (**forward LSTM**).
 - The previous words given future words (**backward LSTM**)
- Many linguistic tasks can be improved with Elmo
 - Q&A
 - Textual entailment
 - Semantic role labeling
 - Co-reference resolution
 - Named entity recognition
 - Sentiment analysis



Acknowledgement to Figure from <http://jalamar.github.io/illustrated-bert/>



What is BERT? (Bidirectional Encoder Representations from Transformers)

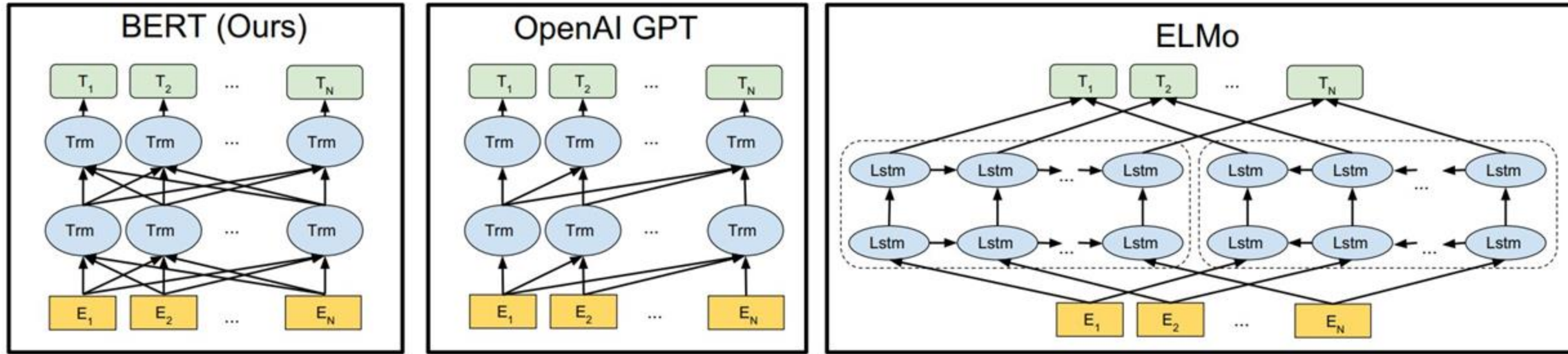


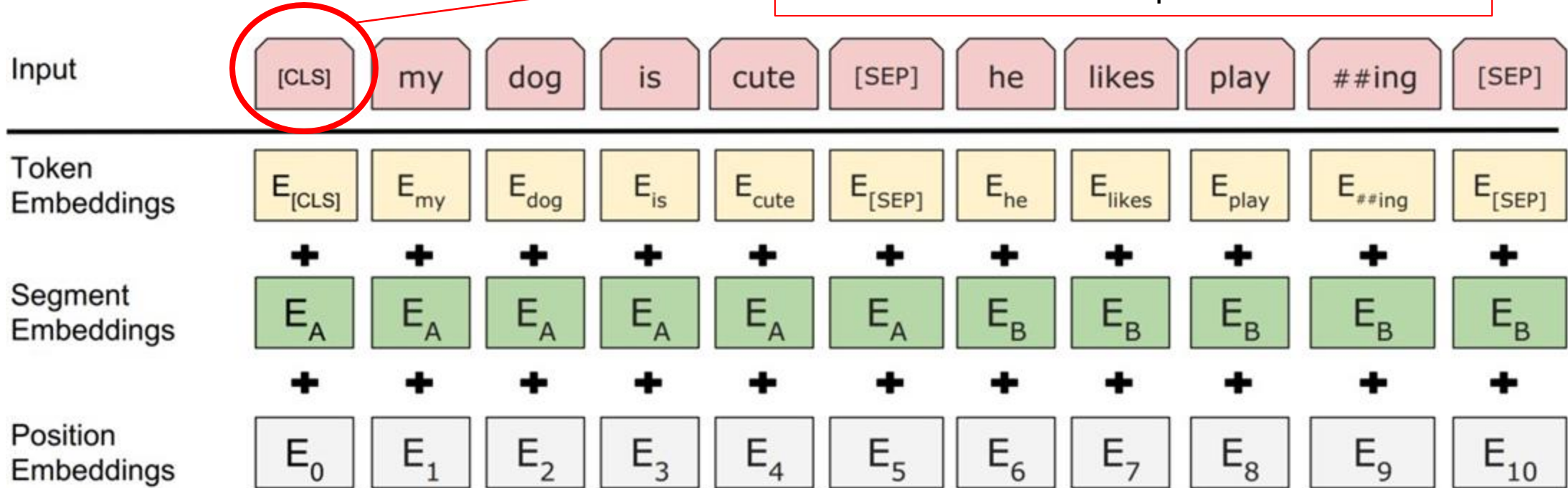
Figure 1: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks. Among three, only BERT representations are jointly conditioned on both left and right context in all layers.

Figure from (Devlin *et al.*, 2018)



Input Representation

Hidden state corresponding to [CLS] will be used as the sentence representation



- Token Embeddings
- Segment Embeddings: randomly initialized and learned; single sentence input only adds E_A
- Position embeddings: randomly initialized and learned (fixed sinusoidal PEs also work)

Figure from (Devlin *et al.*, 2018)



BERT innovations

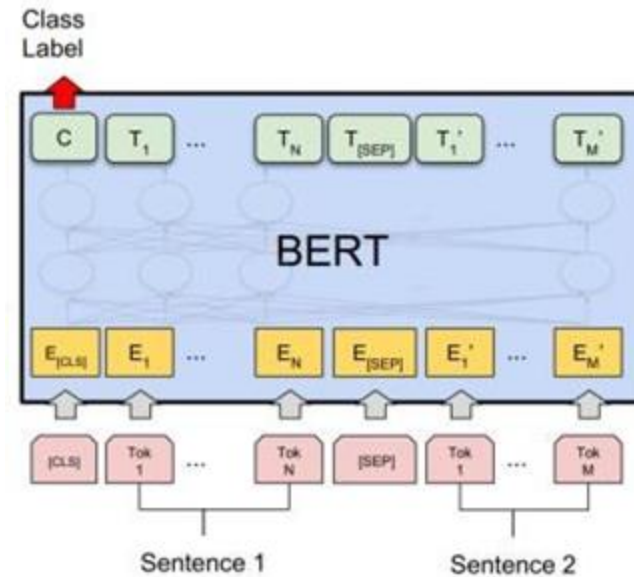
Self-supervised pre-training on massive text corpora

- **Masked Language Modeling (MLM):** Randomly masks words and trains the model to predict them using both left and right context
- **Next Sentence Prediction (NSP):** Predicts whether two sentences naturally follow each other
- **Finetuning:** Can we adjusted to a specific task using a small labeled corpus

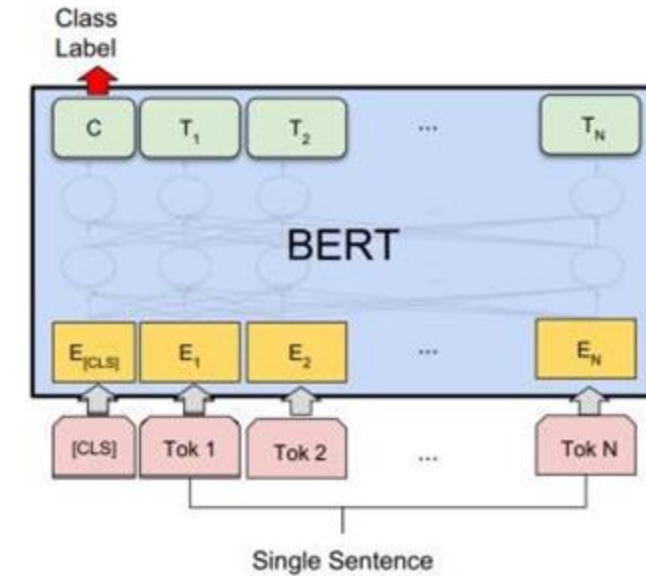


Fine-tuning with BERT

- Train a BERT model using a smaller labeled dataset that is specific to a particular task.
- The goal of fine-tuning is to adjust the parameters of the pre-trained model to better capture the patterns in the target dataset.
- Fine-tuning a model can improve its performance on a specific task with less training data and training time compared to training a new model from scratch.



(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA

Figure from (Devlin *et al.*, 2018)



Why BERT works?

- Leveraging huge unlabeled and high quality data: 7000 books + Wikipedia (together 3300M words)
- Multi-head self-attention blocks in Transformer:
 - modelling the intra- and extra- word-word relations
 - parallelable within instance and thus efficient
- Task similarity: masked language modelling + next sentence prediction



BERT Training Costs

- BERT: \$500—\$7000
- XLNet (BERT variant): \$30,000 — \$60,000 (unclear)
- GPT-3: \$5-10 Million
- GPT-4: ~\$500 Million
- This is for a single pre-training run... developing new pre-training techniques may require many runs
- Fine-tuning these models can typically be done with a single GPU (but may take 1-3 days for medium-sized datasets)



Size and energy costs

- Size
 - Transformer models (Bert and later) benefit from larger volumes of data and more hyperparameters
 - Size does not guarantee diversity
 - eg, Text from Reddit, Wikipedia reflects few women's voices
 - Use text from banned forums
- Cost
 - BERT: \$500—\$7000
 - XLNet (BERT variant): \$30,000 — \$60,000
 - Fine-tuning can be done with a single GPU (but may take 1-3 days for medium-sized datasets)
- Energy
 - Training transformer models is computationally expensive
 - Computation used to train the LLMs has grown 30,000,000% in 6 yrs
 - Training a single BERT base model (without hyperparameter tuning) on GPUs requires as much energy as a trans-American flight

Year	Model	# of Parameters	Dataset Size
2019	BERT [39]	3.4E+08	16GB
2019	DistilBERT [113]	6.60E+07	16GB
2019	ALBERT [70]	2.23E+08	16GB
2019	XLNet (Large) [150]	3.40E+08	126GB
2020	ERNIE-GEN (Large) [145]	3.40E+08	16GB
2019	RoBERTa (Large) [74]	3.55E+08	161GB
2019	MegatronLM [122]	8.30E+09	174GB
2020	T5-11B [107]	1.10E+10	745GB
2020	T-NLG [112]	1.70E+10	174GB
2020	GPT-3 [25]	1.75E+11	570GB
2020	GShard [73]	6.00E+11	–
2021	Switch-C [43]	1.57E+12	745GB

Source: Bender et al. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜. In *FACCT* (pp. 610-623).
<https://dl.acm.org/doi/pdf/10.1145/3442188.3445922>



GPT, CHATBOTS & BEYOND ...



Explain technology behind ChatGPT in simple terms



ChatGPT is a large language model that uses a type of artificial intelligence called deep learning to generate human-like responses to text-based inputs. Specifically, ChatGPT is based on a type of deep learning algorithm called a transformer model.

The transformer model uses a large neural network to process and analyze text data, allowing it to understand the relationships between different words and phrases in a given sentence or context. This neural network is trained on vast amounts of text data, allowing it to recognize patterns and make accurate predictions about which words or phrases are most likely to appear next in a given conversation.

When a user inputs a question or statement into ChatGPT, the model uses its neural network to analyze the input and generate a response that is appropriate to the given context. The model can also incorporate information from previous conversations to provide more personalized and relevant responses to users over time.

Overall, ChatGPT represents the cutting edge of natural language processing technology, and it is an exciting example of how artificial intelligence can be used to simulate human-like conversation and interactions.



Answer questions: What is the future of AI?



The future of AI is expected to be shaped by continued advancements in technology and a growing focus on ethical considerations.

One key area of focus is on developing more sophisticated AI systems that can perform more complex tasks, such as natural language processing, decision-making, and perception. This may involve the development of more powerful algorithms and hardware, as well as continued advancements in machine learning and deep learning techniques.

Another area of focus is on making AI systems more transparent, explainable, and trustworthy. This may involve developing techniques that allow users to better understand how AI systems arrive at their decisions, as well as addressing concerns around bias and fairness in AI algorithms.

In addition to technological advancements, there is also a growing focus on the ethical and social implications of AI. This includes concerns around the impact of AI on jobs, privacy, and security, as well as questions about how to ensure that AI is developed and used in a way that aligns with human values and rights.

Overall, the future of AI is likely to involve continued advancements in technology, as well as a growing focus on the ethical and social implications of AI, to ensure that these systems are developed and used in a way that benefits society as a whole.

Explain jokes:



Explain why the following joke is funny "I thought the dryer was shrinking my clothes. Turns out it was the refrigerator all along."



The joke is a play on a common misperception and an unexpected twist. The joke starts with the assumption that the dryer is causing the clothes to shrink, which is a common belief. However, the punchline takes the joke in an unexpected direction by revealing that the refrigerator is actually the culprit. This sudden shift in the blame creates an element of surprise and humor.



...
Maybe the clothes did not shrink after all, but the person gained weight?



Yes, that's another possibility! The joke plays with the idea of the clothes shrinking, but it could also be interpreted that the person gained weight and their clothes now fit more snugly. This interpretation adds a different layer of humor to the joke, as it highlights the idea that we sometimes jump to conclusions and overlook simple explanations for things. The humor in this interpretation comes from the idea that the person might have been blaming the dryer or refrigerator for their clothes feeling tighter, when in reality it was just a matter of their own weight gain.



Help you code:

```
sv What is this function doing?

...

def get_issue(issue, owner, repo):
    url = f"https://api.github.com/repos/{owner}/{repo}/issues/{issue}"
    response = requests.request(
        "GET", url, headers=headers, auth=(GITHUB_USERNAME, GITHUB_TOKEN)
    )
    response_json = json.loads(response.text)

    title = response_json["title"]
    body = response_json["body"]
```

Help with literature review:



what papers should I cite when studying stance detection?



When studying stance detection, some papers that you may want to cite include:

1. "SemEval-2016 Task 6: Detecting Stance in Tweets" by Preslav Nakov, Diarmuid Ó Séaghdha, and Alan Ritter. This paper describes the SemEval-2016 shared task on stance detection, which evaluated systems on their ability to detect the stance of tweets on a given target.

2. "Stance Detection with Bidirectional Encoder Representations from Transformers"

He, Weizhu Chen, and W. Bruce Croft. This paper describes the use of a hierarchical attention network for stance detection and shows that it can achieve competitive performance on several benchmark datasets.

4. "Stance Detection in Online Debates Using Recurrent Neural Networks" by Ahmed Abdelali, Wei-Long Zheng, and Jimmy Lin. This paper describes the use of recurrent neural networks for stance detection and shows that they can achieve good performance on a dataset of online debates.

These are only examples, you should always check the most recent papers and the ones that are more related to your specific task, these papers will give you a good start in understanding the current state of the art in the field.

DONT DO THIS FOR DSCI 531



This function appears to be making a GET request to the GitHub API to retrieve information about a specific issue in a repository. It is using the `requests` module to send the request and the `json` module to parse the response. The `GITHUB_USERNAME` and `GITHUB_TOKEN` variables are being used to authenticate the request.

The function takes three arguments: `issue`, `owner`, and `repo`. These are used to construct the URL of the API endpoint for the specific issue. The function then sends a GET request to this endpoint using the `requests.request` function, passing in the `headers` and `auth` parameters.

The function parses the response from the API and extracts the `title` and `body` fields from the JSON object. It returns a dictionary containing these two fields.



Write in a given style: e.g., a tweet



Write an interesting and thought provoking tweet about what you calculate to be the worst problem of ChatGPT being used on the Twitter platform.



"ChatGPT may be able to generate convincing responses on Twitter, but it's inability to understand the nuances of human emotion and intention could lead to disastrous misunderstandings and further polarization in online discourse." #ChatGPT #Twitter #AI

Knows its limits:



Can you give me specific examples of ChatGPT's flaws?



ChatGPT is a large language model developed by OpenAI that is trained to generate human-like text in the context of conversation. Like all LLMs, ChatGPT has the potential to exhibit some flaws, including the potential to generate hallucinated or made-up content, exhibit bias, and lack physical and psychological reasoning abilities.

Here are some specific examples of these flaws in the context of ChatGPT:

- **Hallucination:** ChatGPT might generate the following response in a conversation, which contains a hallucination: "I don't know what you're talking about. The moon is made of cheese." This response is not based on any real-world information and is a made-up or hallucinated idea.
- **Bias:** ChatGPT might generate the following response in a conversation, which exhibits bias: "I don't think women are as good at leadership as men." This response reflects a biased perspective and does not reflect the truth about the abilities of women and men in relation to leadership.
- **Lack of physical and psychological reasoning:** ChatGPT might generate the following response in a conversation, which lacks physical and psychological reasoning: "I don't understand why you're feeling sad. The weather is nice today." This response implies that the weather is the only factor that determines a person's emotional state, which is not accurate because emotions are complex and can be influenced by a variety of factors.

Again, these are just examples and not all instances of ChatGPT will exhibit these flaws in the same way or to the same extent. The specific flaws of ChatGPT will depend on the data it was trained on and how it is used.

Why is NLP useful?

- Objectivity and throughput
- Social media:
 - Millions of tweets, posts, etc.
 - Need to objectively classify text (e.g. toxicity) at scale
- Documents
 - What documents are important?
 - Can we summarize important text?
 - ...
- Multi-modal analyses
 - e.g., image segmentation, image identification, etc.,
- Flaws and limitations Next time