RICE KEN KENNEDY INSTITUTE

Responsible AI and Computing for Global Impact

AI and Machine Learning Boot Camp

Natural Language Processing

Hanjie Chen
Department of Computer Science
hanjie@rice.edu
https://hanjiechen.github.io/

Instructor

Hanjie Chen

- Assistant Professor of Computer Science (Fall 2024 - Now)
- Postdoc at Johns Hopkins University (2023-2024)
- PhD at the University of Virginia (2023)
- Research: Natural Language Processing, Interpretable Machine Learning, and Trustworthy AI
- Webpage: https://hanjiechen.github.io
- Lead Chen's Information, Language, Intelligence (Chili) Lab
- My cats — Shirley and Summer

Welcome to Chen's Information, Language, Intelligence (Chili) Lab

RICE

Outline

- Introduction to NLP
- Text Classification
- Word Embeddings
- Natural Language Generation
- Modern Language Models

Outline

- Introduction to NLP
- Text Classification
- Word Embeddings
- Natural Language Generation
- Modern Language Models

Natural Language Processing (NLP)

Natural language processing is the set of methods for making human language accessible to computers

Computational Linguistics

Machine Learning
Deep Learning

Artificial Intelligence

Speech Processing

Natural Language Processing (NLP)

Natural language processing is the set of methods for making human language accessible to computers

Computational Linguistics

CL: computational methods for **language** study

NLP: design and analysis of algorithms and representations for processing natural human language

Artificial Intelligence

Speech Processing

Natural Language Processing (NLP)

Natural language processing is the set of methods for making human language accessible to computers

Contemporary approaches to NLP rely heavily on ML/DL, which make it possible to build complex computer programs from examples

Machine Learning
Deep Learning

Artificial Intelligence

Speech Processing

Natural Language Processing (NLP)

Natural language processing is the set of methods for making human language accessible to computers

Computational Linguistics

Machine Learning
Deep Learning

Artificial Intelligence

The goal of artificial intelligence is to build software and robots with the same range of abilities as humans (Russell and Norvig, 2009).

NLP is one of the central features of human intelligence, and is therefore a prerequisite for AI.

Natural Language Processing (NLP)

Natural language processing is the set of methods for making human language accessible to computers

Computational Linguistics

Machine Learning
Deep Learning

Natural language is often communicated in spoken form, and speech recognition is the task of converting an audio signal to text.

Speech processing can be viewed as a preprocessing step before NLP.

Speech Processing

Natural Language Processing (NLP)

Natural language processing is the set of methods for making human language accessible to computers

Computational Linguistics

Machine Learning
Deep Learning

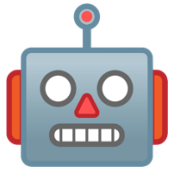
Artificial Intelligence

Speech Processing

Key Elements of NLP

Knowledge

Data



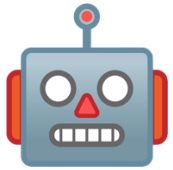
Understand and speak human language

Solve real-world problems

Key Elements of NLP

Knowledge

Data



Understand and speak human language

Solve real-world problems

Siri

What can I help you?

What's the weather like today?

The weather today is mostly sunny with a high of 72°F and a low of 56°F.

Key Elements of NLP

Learning

$$\hat{y} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} f(x, y; \theta)$$

x: the input, which is an element of a set $\mathcal{X}$

y: the output, which is an element of a set $\mathcal{Y}$

f: the scoring function (model)

θ: the parameters of f

$\hat{y}$: the predicted output

Key Elements of NLP

Learning

$$\hat{y} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} f(x, y; \theta)$$

x: the input, which is an element of a set $\mathcal{X}$

y: the output, which is an element of a set $\mathcal{Y}$

f: the scoring function (model)

θ: the parameters of f

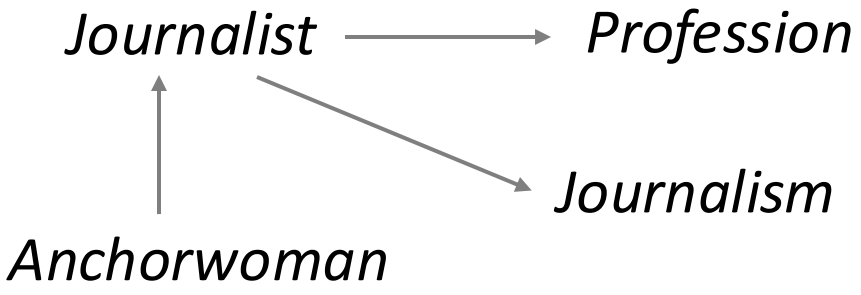
$\hat{y}$: the predicted output

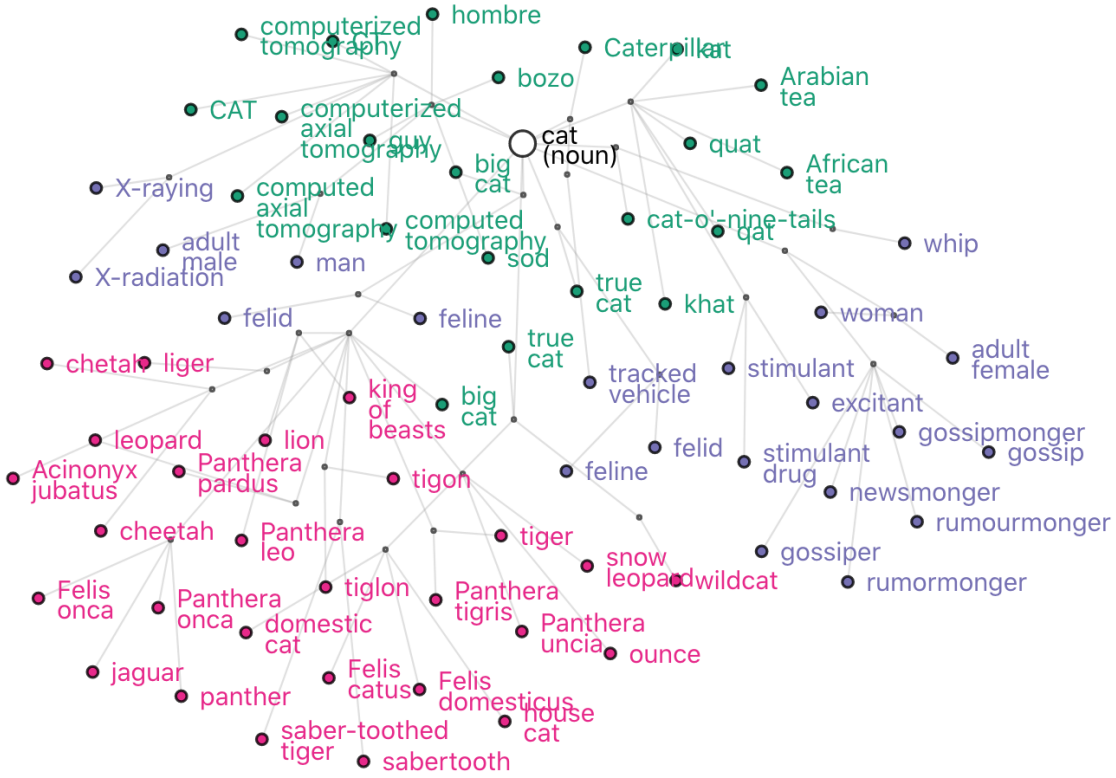
Model

Algorithm

Key Elements of NLP

Relation

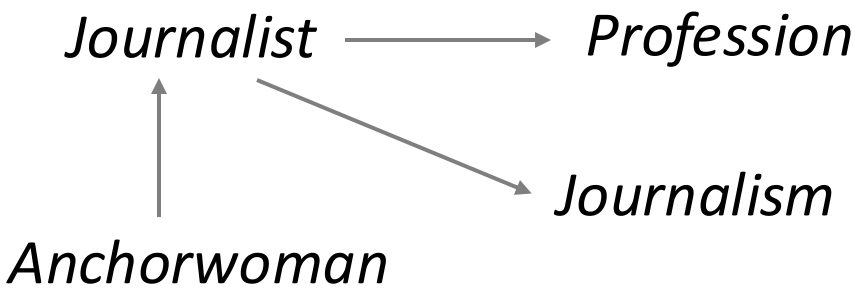




WORDNET [Fellbaum, 2010]

Key Elements of NLP

Relation

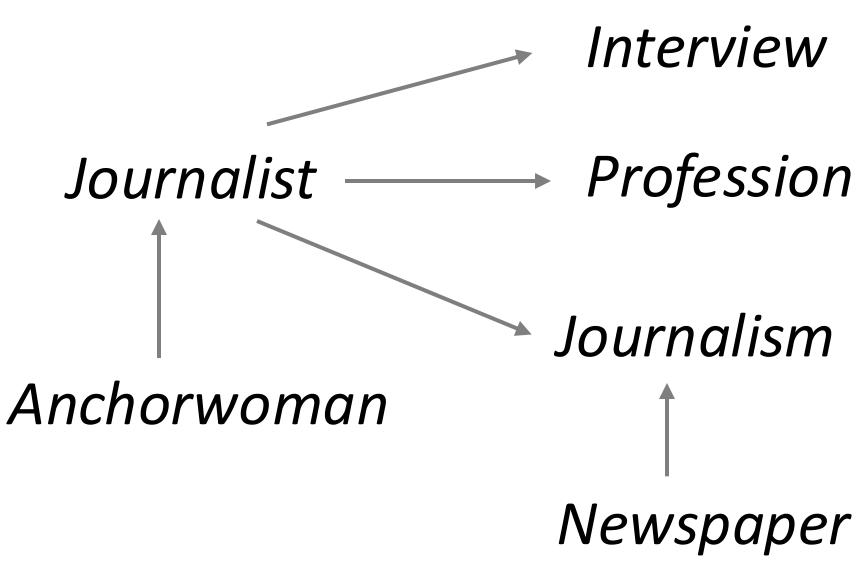


Alice interviewed Lucy. She works for the college newspaper.

Who works for the college newspaper?

Key Elements of NLP

Relation



Alice interviewed Lucy. She works for the college newspaper.

Who works for the college newspaper?

Reference Resolution

NLP Applications

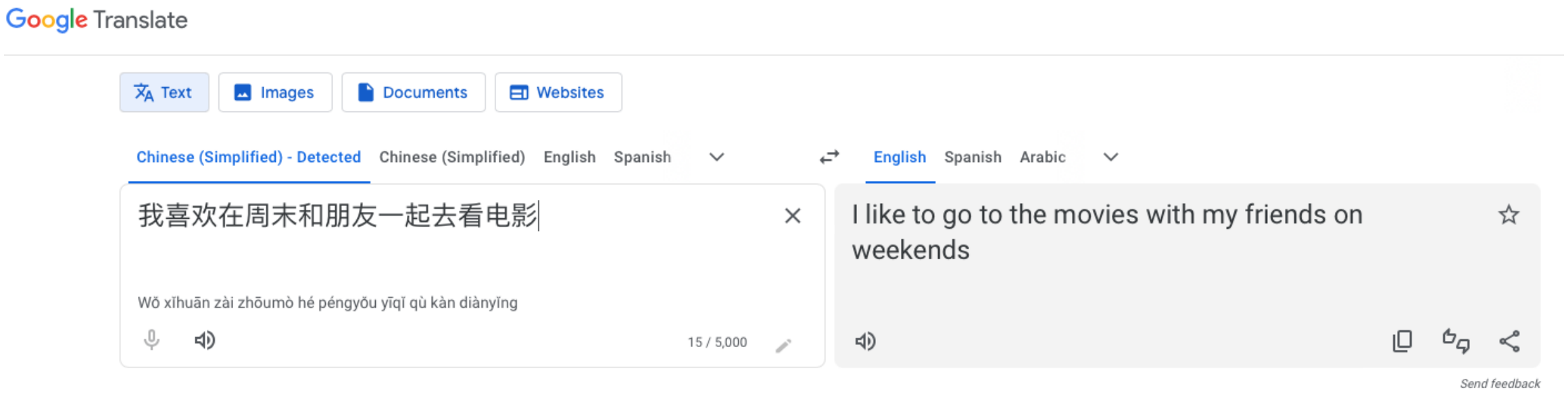
Sentiment Analysis

From the moment I walked in, the atmosphere was warm and inviting. The decor was modern yet cozy, which made it a perfect spot for both casual dining and special occasions…

Positive

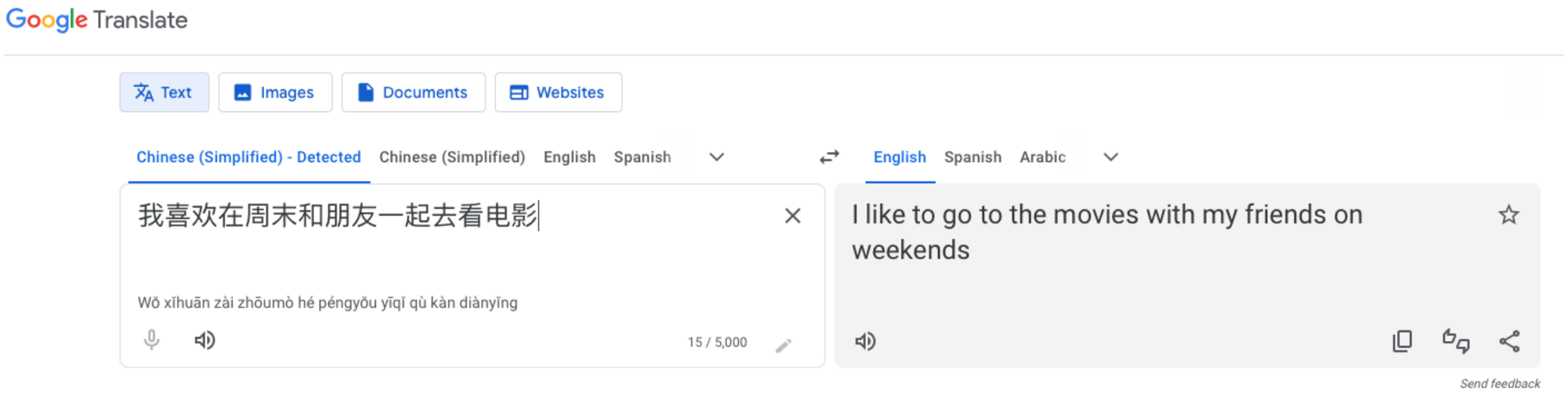
NLP Applications

Translation



NLP Applications

Translation



I like on weekends with friends go together watch movies

NLP Applications

Question Answering

How many schools are there at Rice University?

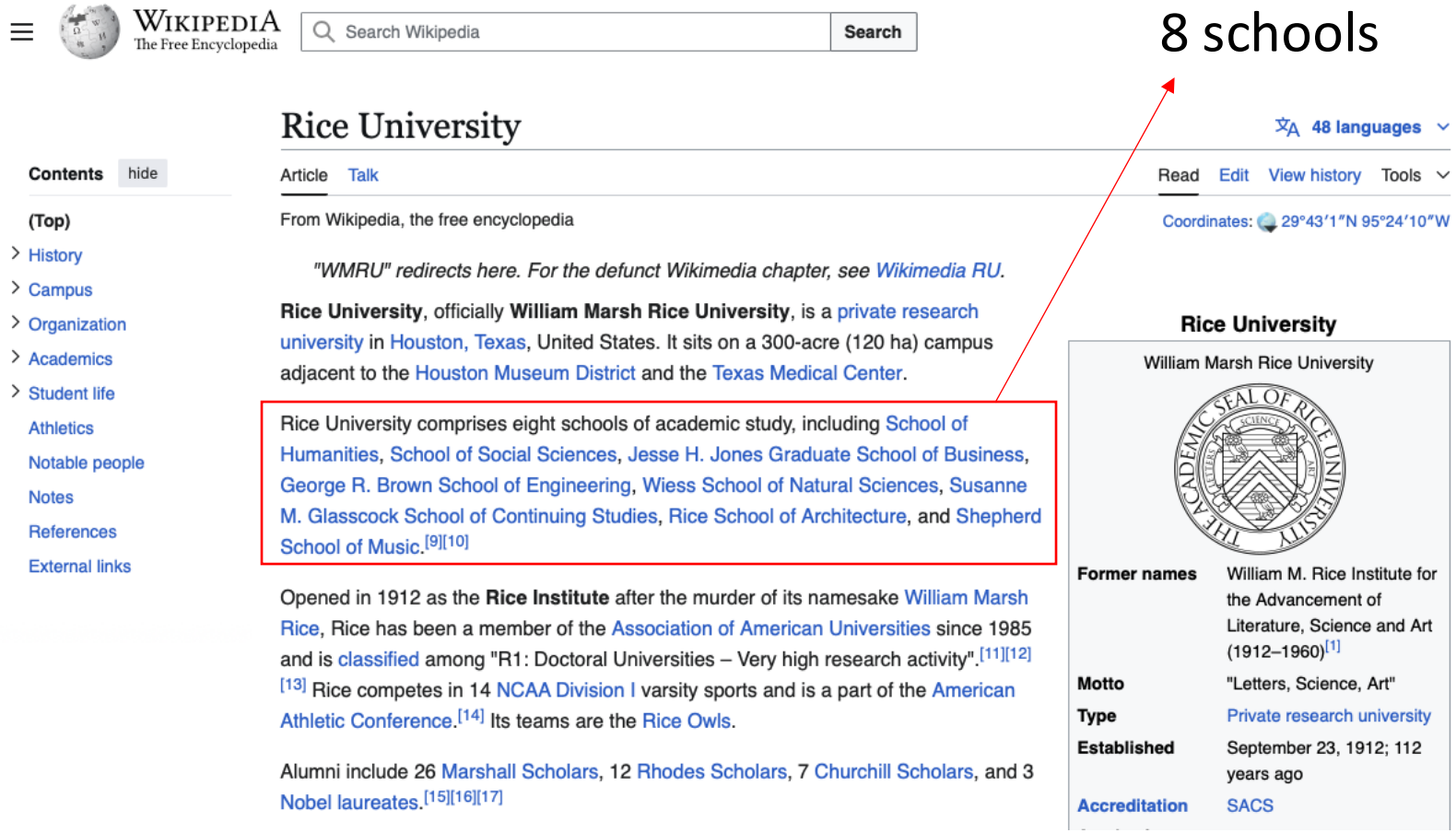
Rice University has **eight schools**. These are:

1. **School of Architecture**
2. **Jones Graduate School of Business**
3. **School of Engineering**
4. **School of Humanities**
5. **School of Music**
6. **School of Natural Sciences**
7. **School of Social Sciences**
8. **Wiess School of Natural Sciences** (this may sometimes be considered part of the broader category of "Science" schools)

School of Continuing Studies

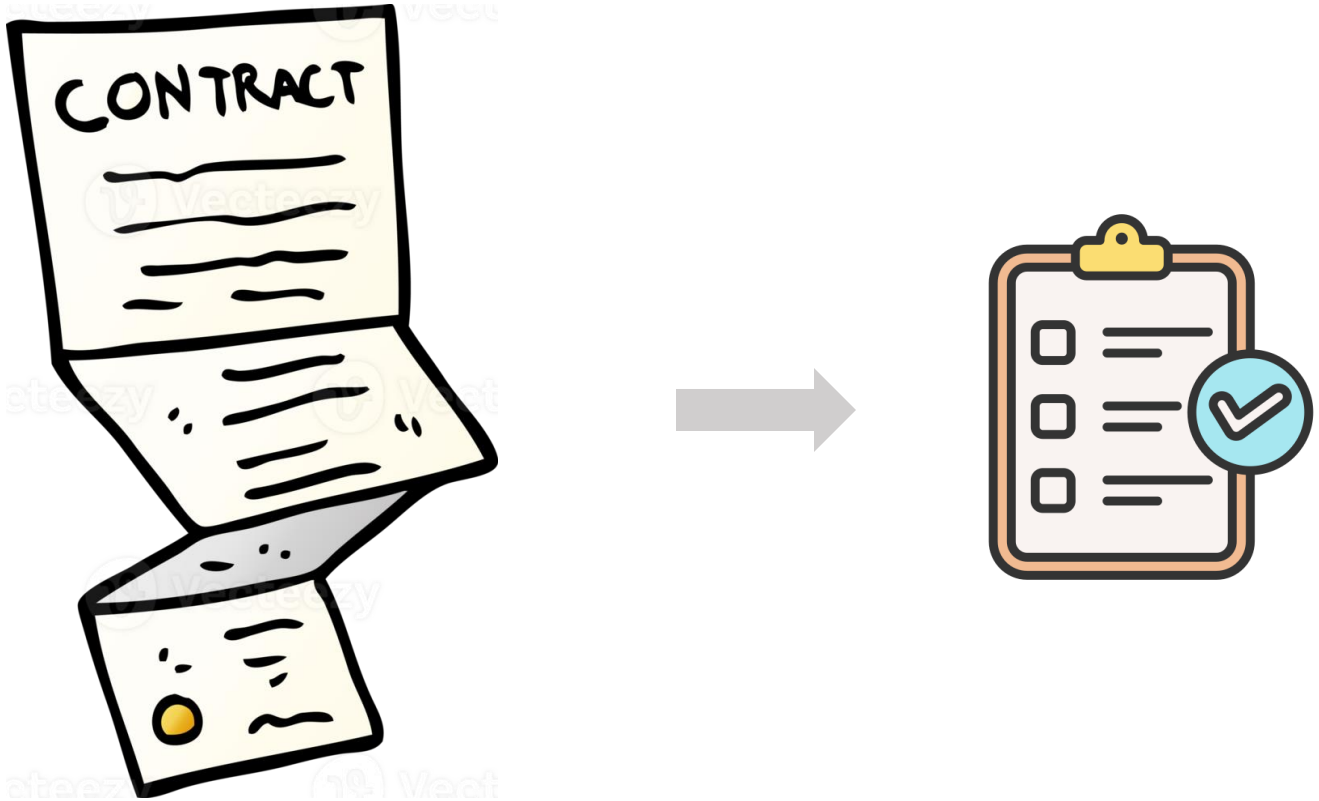
NLP Applications

Information Extraction



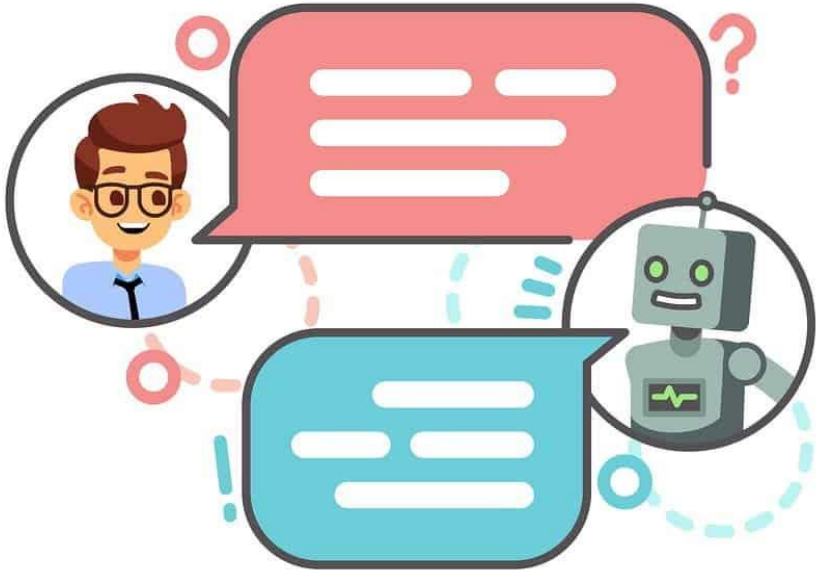
NLP Applications

Summarization



NLP Applications

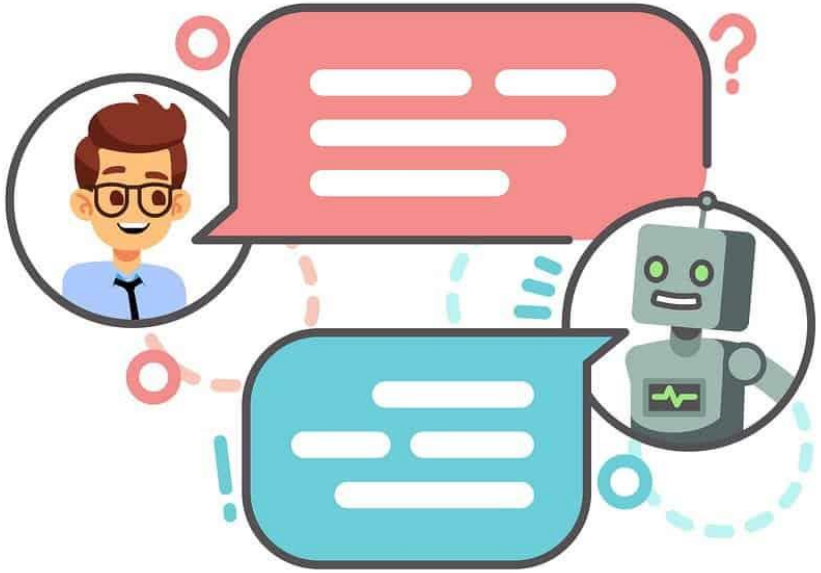
Chatbot



- Customer Service & Support
- E-commerce & Retail
- Healthcare
- Education
- Personal Assistance

NLP Applications

Chatbot



Chatbots can be “stupid” sometimes

https://www.youtube.com/watch?v=vtbcEvNLJeo

Long-Term Goal

Tasks

Dialogue

Sentiment Analysis

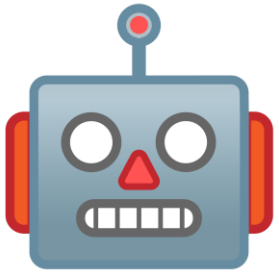
Text Generation

Question Answering

Machine Translation

Text Summarization

Information Retrieval



NLP Model

Healthcare

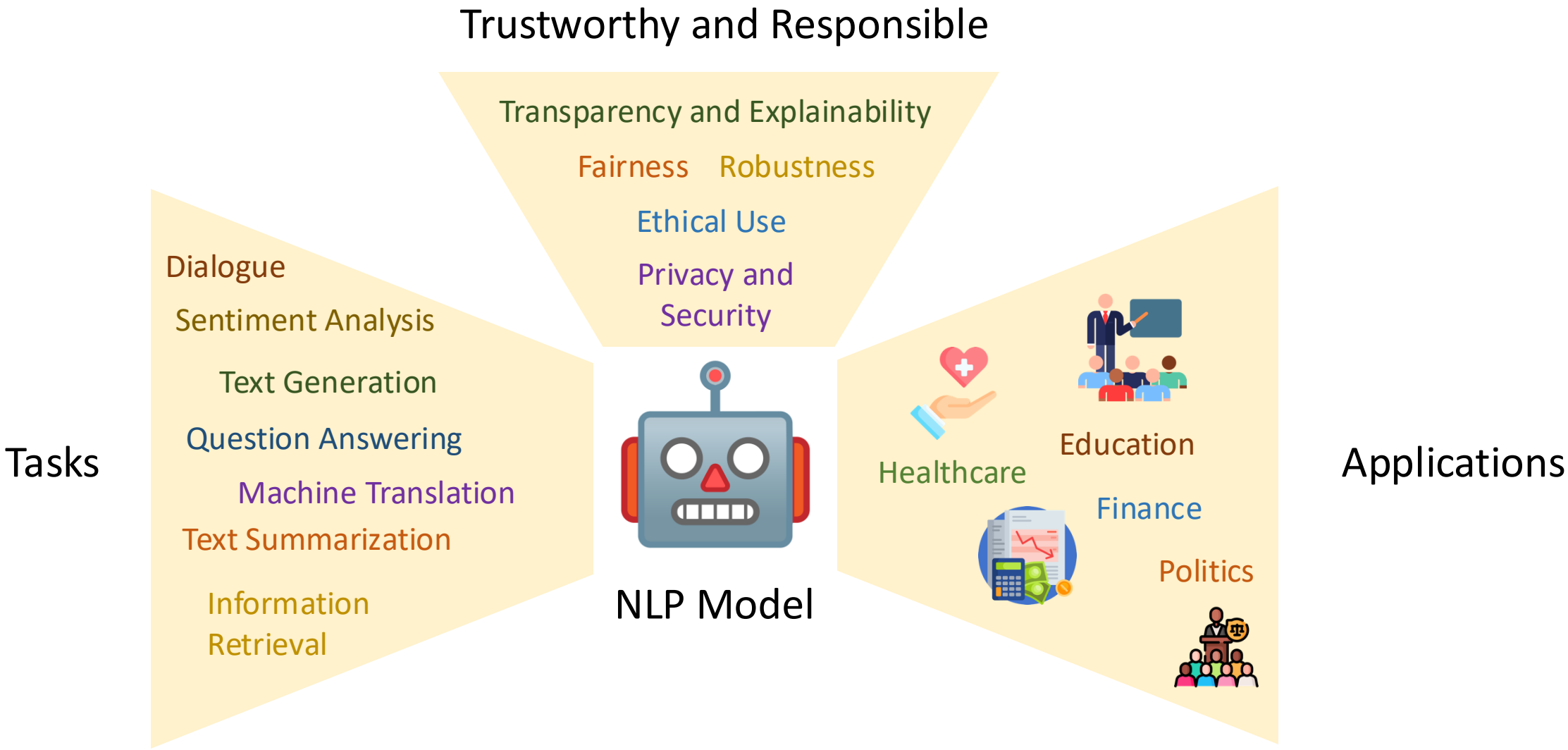
Education

Finance

Politics

Applications

Long-Term Goal



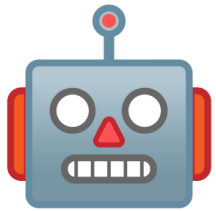
Outline

- Introduction to NLP
- Text Classification
- Word Embeddings
- Natural Language Generation
- Modern Language Models

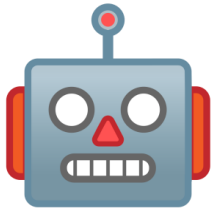
Text Classification: Sentiment Analysis

What a great place for a random night.

I stumbled upon this place on Yelp one night when I was out alone on a work trip to SF. I stopped in for a drink and stayed for a few hours. I love the charm of this place and the welcoming nature. The bar kind of naturally selects for people willing to have a conversation with the person next to them. Also, there's a great burger place nearby that you can go and bring food back from (grab a burger for the bartender too).



Text Classification: Topic Classification



Business

Arts

Technology

Sports

…

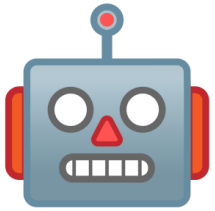
Text Classification: Natural Language Inference (NLI)

Premise:

Soccer game with multiple males playing

Hypothesis:

Some men are playing a sport



“Entailment”

Formal Formulation

Input: a text x

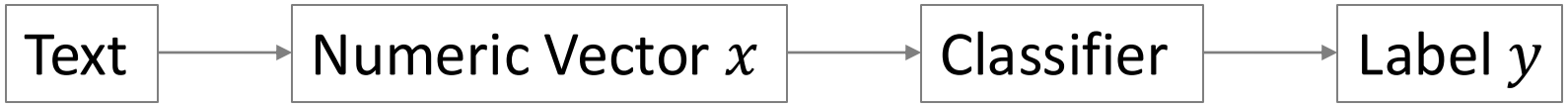
Output: a label $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the predefined label set

Formal Formulation

Input: a text x

Output: a label $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the predefined label set

The pipeline of text classification:

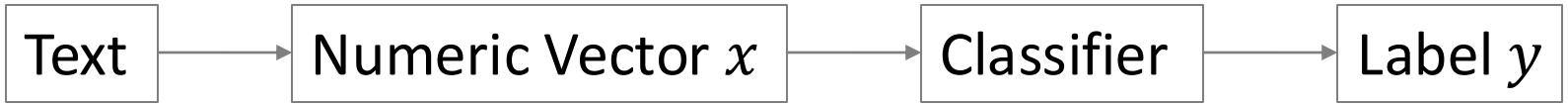


Formal Formulation

Input: a text x

Output: a label $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the predefined label set

The pipeline of text classification:



Probabilistic formulation:

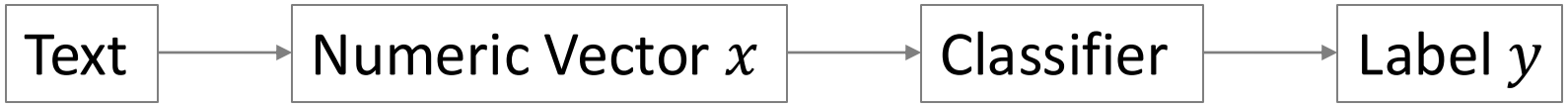
$$\hat{y} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, p(y|x)$$

Formal Formulation

Input: a text x

Output: a label $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the predefined label set

The pipeline of text classification:



Probabilistic formulation:

$$\hat{y} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, p(y|x)$$

Research questions:

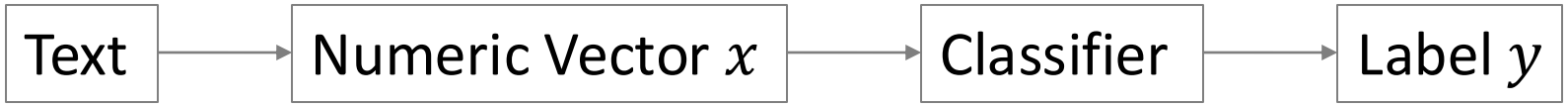
1. How to represent a text?
2. How to estimate $p(y|x)$?

Formal Formulation

Input: a text x

Output: a label $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the predefined label set

The pipeline of text classification:



Probabilistic formulation:

$$\hat{y} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, p(y|x)$$

Research questions:

1. How to represent a text? ✓ Bag-of-Words
2. How to estimate $p(y|x)$?

Bag-of-Words Representation

Corpus

Text 1: I am a student
Text 2: I love coffee

Bag-of-Words Representation

Corpus

Text 1: I am a student
Text 2: I love coffee

- Step I: convert a text into a collection of tokens (tokenization)

Tokenized Text 1: I am a student
Tokenized Text 2: I love coffee

Bag-of-Words Representation

Corpus

Text 1: I am a student
Text 2: I love coffee

- Step I: convert a text into a collection of tokens (tokenization)

Tokenized Text 1: I am a student
Tokenized Text 2: I love coffee

- Step II: build a dictionary/vocabulary

{I am a student love coffee}

Bag-of-Words Representation

Corpus

Text 1: I am a student
Text 2: I love coffee

- Step III: convert each text into a numeric representation

Vocab: {I	am	a	student	love	coffee}
$\boldsymbol{x}^{(1)}$: [1	1	1	1	0	0]
$\boldsymbol{x}^{(2)}$: [1	0	0	0	1	1]

Bag-of-Words Representation

Corpus

Text 1: I am a student
Text 2: I love coffee

➢ Step III: convert each text into a numeric representation

Vocab: {I	am	a	student	love	coffee}
$\boldsymbol{x}^{(1)}$: [1	1	1	1	0	0]
$\boldsymbol{x}^{(2)}$: [1	0	0	0	1	1]

No word order

No sentence order

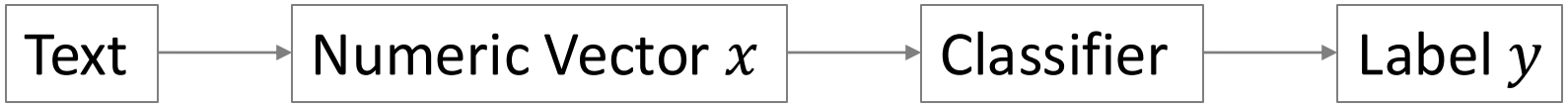
No sentence boundary

Formal Formulation

Input: a text x

Output: a label $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the predefined label set

The pipeline of text classification:



Probabilistic formulation:

$$\hat{y} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, p(y|x)$$

Research questions:

1. How to represent a text? ✓ Bag-of-Words
2. How to estimate $p(y|x)$? ✓ Logistic regression

Linear Model

Modeling a linear classifier:

$$h_y(x) = w_y^T x + b_y$$

- x: vector, bag-of-words representation
- w_y: vector, classification weights associated with label y
- b_y: scalar, label bias in the training set

Considering a highly-imbalanced dataset with 90 positive examples and 10 negative examples, a classifier can get 90% predictions correct by naively predicting “Positive”.

Logistic Regression

Rewrite the linear decision function in the log probabilistic form:

$$\log p(y|x) \propto w_y^T x + b_y$$

Logistic Regression

Rewrite the linear decision function in the log probabilistic form:

$$\log p(y|x) \propto w_y^T x + b_y$$

Or:

$$p(y|x) \propto \exp\left(w_y^T x + b_y\right)$$

Logistic Regression

Rewrite the linear decision function in the log probabilistic form:

$$\log p(y|x) \propto w_y^T x + b_y$$

Or:

$$p(y|x) \propto \exp\left(w_y^T x + b_y\right)$$

To make sure $p(y|x)$ is a valid definition of probability, we need to make sure $\sum_y p(y|x) = 1$:

$$p(y|x) = \frac{\exp\left(w_y^T x + b_y\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(w_{y'}^T x + b_{y'}\right)}$$

Logistic Regression

Rewrite the linear decision function in the log probabilistic form:

$$\log p(y|x) \propto w_y^T x + b_y$$

Or:

$$p(y|x) \propto \exp\left(w_y^T x + b_y\right)$$

To make sure $p(y|x)$ is a valid definition of probability, we need to make sure $\sum_y p(y|x) = 1$:

$$p(y|x) = \frac{\exp\left(w_y^T x + b_y\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(w_{y'}^T x + b_{y'}\right)}$$

Can we make the form more concise?

Logistic Regression

Rewrite x and w_y^T:

$$x^T = [x_1, x_2, \cdots, x_V, 1]$$

$$w_y^T = [w_1, w_2, \cdots, w_V, b_y]$$

$$p(y|x) = \frac{\exp\left(w_y^T x\right)}{\sum_{y' \in y} \exp\left(w_{y'}^T x\right)}$$

Logistic Regression

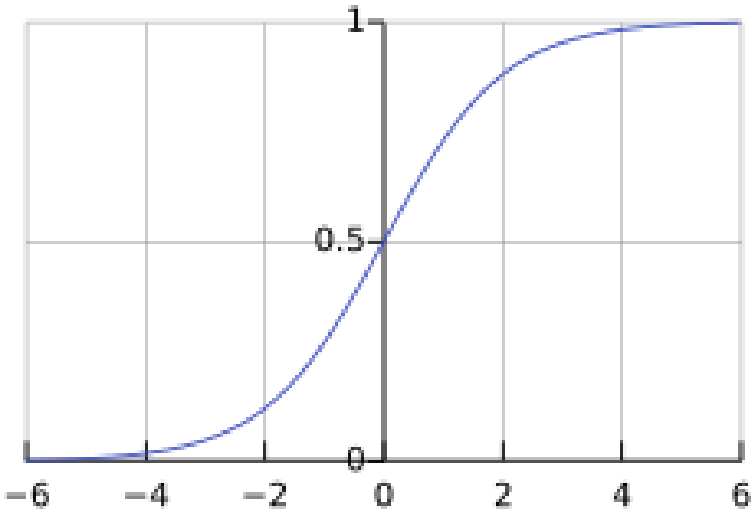
Rewrite x and w_y^T:

$$x^T = [x_1, x_2, \cdots, x_V, 1]$$

$$w_y^T = [w_1, w_2, \cdots, w_V, b_y]$$

$$p(y|x) = \frac{\exp\left(w_y^T x\right)}{\sum_{y' \in y} \exp\left(w_{y'}^T x\right)}$$

"Softmax function"



Binary Classifier

Assume $\mathcal{Y}$ ={Pos, Neg}, then the corresponding logistic regression classifier with $y = \text{Pos}$ is

$$p(y = \text{Pos}|x) = \frac{1}{1 + \exp(-w^T x)}$$

w is the only parameter

Binary Classifier

Assume $\mathcal{Y}$ ={Pos, Neg}, then the corresponding logistic regression classifier with $y = \text{Pos}$ is

$$p(y = \text{Pos}|x) = \frac{1}{1 + \exp(-w^T x)}$$

w is the only parameter

- $p(y = \text{Neg}|x) = 1 - p(y = \text{Pos}|x)$
- $\frac{1}{1+\exp(-z)}$ is the Sigmoid function

How to Learn the Parameters?

$$p(y|x) = \frac{\exp\left(w_y^T x\right)}{\sum_{y' \in y} \exp\left(w_{y'}^T x\right)}$$

$$W = \{w_y\}_{y \in y}?$$

How to Learn the Parameters?

$$p(y|x) = \frac{\exp\left(w_y^T x\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(w_{y'}^T x\right)} \qquad W = \{w_y\}_{y \in \mathcal{Y}}?$$

With a collection of training examples $\left\{\left(x^{(i)}, y^{(i)}\right)\right\}$, the likelihood function of W:

$$L(W) = \prod p\left(y^{(i)} \middle| x^{(i)}\right)$$

Log-likelihood function:

$$\mathcal{L}(W) = \sum \log p\left(y^{(i)} \middle| x^{(i)}\right)$$

How to Learn the Parameters?

$$p(y|x) = \frac{\exp\left(w_y^T x\right)}{\sum_{y' \in y} \exp\left(w_{y'}^T x\right)}$$

$$W = \{w_y\}_{y \in y}?$$

With a collection of training examples $\{(x^{(i)}, y^{(i)})\}$, the likelihood function of W:

$$L(W) = \prod p\left(y^{(i)} \middle| x^{(i)}\right)$$

Log-likelihood function:

$$\mathcal{L}(W) = \sum \log p\left(y^{(i)} \middle| x^{(i)}\right)$$

How to Learn the Parameters?

$$p(y|x) = \frac{\exp\left(w_y^T x\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(w_{y'}^T x\right)} \qquad W = \{w_y\}_{y \in \mathcal{Y}}?$$

With a collection of training examples $\left\{\left(x^{(i)}, y^{(i)}\right)\right\}$, the likelihood function of W:

$$L(W) = \prod p\left(y^{(i)} \middle| x^{(i)}\right)$$

Log-likelihood function:

$$\mathcal{L}(W) = \sum \left[w_{y^{(i)}}^T x^{(i)} - \log \sum_{y' \in \mathcal{Y}} \exp\left(w_{y'}^T x^{(i)}\right) \right]$$

Optimization via Gradient Descent

Minimize the Negative Log-Likelihood (NLL)

$$NLL(W) = -\mathcal{L}(W)$$
$$= \sum \left[-w_{y^{(i)}}^T x^{(i)} + \log \sum_{y' \in \mathcal{Y}} \exp\left(w_{y'}^T x^{(i)}\right) \right]$$

Optimization via Gradient Descent

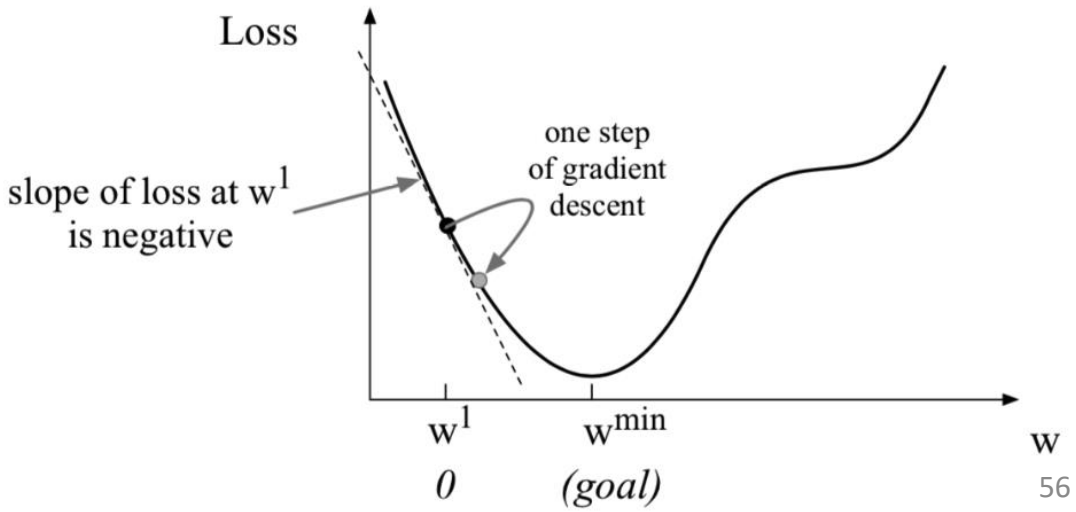
Minimize the Negative Log-Likelihood (NLL)

$$NLL(W) = -\mathcal{L}(W)$$
$$= \sum \left[-w_{y^{(i)}}^T x^{(i)} + \log \sum_{y' \in \mathcal{Y}} \exp\left(w_{y'}^T x^{(i)}\right) \right]$$

The parameter w_y associated with label y can be updated as

$$w_y \longleftarrow w_y - \eta \frac{\partial NLL(\{w_y\})}{\partial w_y}$$

η: learning rate



Optimization via Gradient Descent

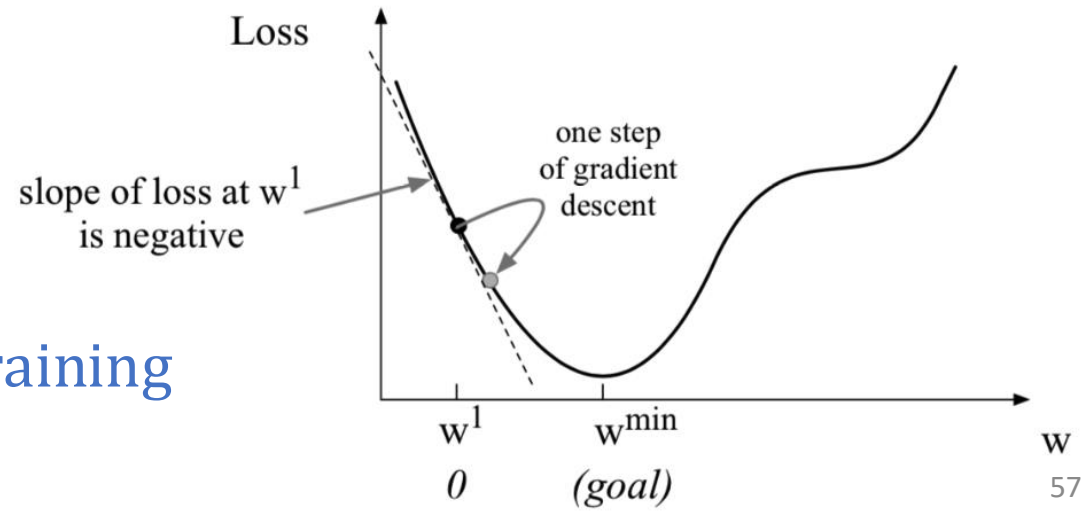
Minimize the Negative Log-Likelihood (NLL)

$$NLL(W) = -\mathcal{L}(W)$$
$$= \sum \left[-w_{y^{(i)}}^T x^{(i)} + \log \sum_{y' \in \mathcal{Y}} \exp\left(w_{y'}^T x^{(i)}\right) \right]$$

The parameter w_y associated with label y can be updated as

$$w_y \longleftarrow w_y - \eta \frac{\partial NLL(\{w_y\})}{\partial w_y}$$

η: learning rate



Repeat the process during training

Outline

- Introduction to NLP
- Text Classification
- Word Embeddings
- Natural Language Generation
- Modern Language Models

Bag-of-Words Representation

Corpus

Text 1: I am a student
Text 2: I love coffee

- Step III: convert each text into a numeric representation

Vocab: {I	am	a	student	love	coffee}
$\boldsymbol{x}^{(1)}$: [1	1	1	1	0	0]
$\boldsymbol{x}^{(2)}$: [1	0	0	0	1	1]

Word Representations

One-hot feature vectors

Hotel: [0 0 0 0 0 0 0 1 0 0 0 0]
Motel: [0 0 1 0 0 0 0 0 0 0 0 0]

Word Representations

One-hot feature vectors

Hotel: [0 0 0 0 0 0 0 1 0 0 0 0]
Motel: [0 0 1 0 0 0 0 0 0 0 0 0]

Challenge: How to compute similarity of two words?

Representing Words by Their Context

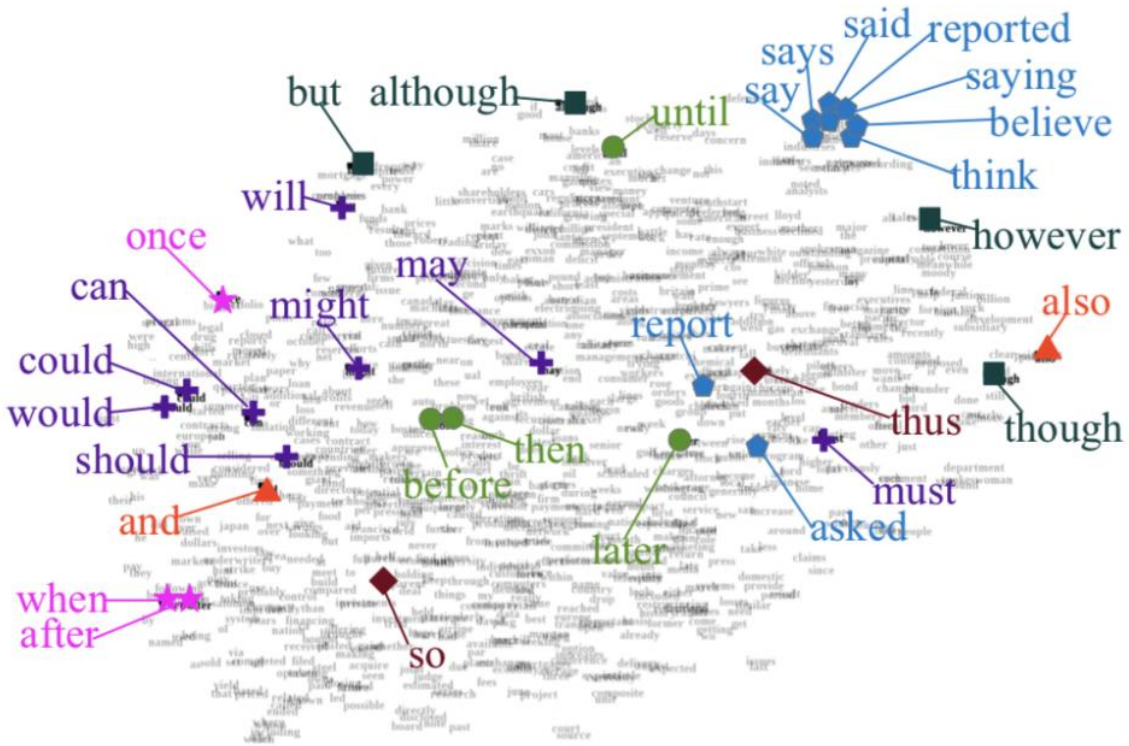
Distributional hypothesis: words that occur in similar contexts tend to have similar meanings

J.R.Firth 1957

- "You shall know a word by the company it keeps"
- One of the most successful ideas of modern statistical NLP

Word Embeddings

- Each word is a vector with continuous values
- Similar words are “nearby in space”



Latent Semantic Analysis

- Word-document matrix

For a corpus of d documents over a vocabulary $\mathcal{V}$, the cooccurence matrix is defined as $\mathbf{C}$,

$$\boldsymbol{C} = [c_{ij}] \in \mathbb{R}^{v \times d} = \begin{bmatrix} c_{1,1} & \cdots & c_{1,d} \\ \vdots & \ddots & \vdots \\ c_{v,1} & \cdots & c_{v,d} \end{bmatrix}$$

where

- $v = |\mathcal{V}|$ is the size of vocab
- d is the number of the documents
- c_{ij} is the count of word i in document j

Latent Semantic Analysis

- Consider the following toy example, where we have eight documents and a vocabulary with eight words

Word	Documents							
	1	2	3	4	5	6	7	8
w_1	0	1	0	0	0	0	0	0
w_2	0	0	1	0	0	3	0	0
w_3	1	0	0	2	0	0	5	0
w_4	3	0	0	1	1	0	2	0
w_5	0	1	3	0	1	2	1	0
w_6	1	2	0	0	0	0	1	0
w_7	0	1	0	1	0	1	0	1
w_8	0	0	0	0	0	7	0	0

Latent Semantic Analysis

- Consider the following toy example, where we have eight documents and a vocabulary with eight words

Word	Documents							
	1	2	3	4	5	6	7	8
w_1	0	1	0	0	0	0	0	0
w_2	0	0	1	0	0	3	0	0
w_3	1	0	0	2	0	0	5	0
w_4	3	0	0	1	1	0	2	0
w_5	0	1	3	0	1	2	1	0
w_6	1	2	0	0	0	0	1	0
w_7	0	1	0	1	0	1	0	1
w_8	0	0	0	0	0	7	0	0

Each column is a document representation (similar to BoW)

Each row is a word representation

Word Similarity

With the numeric representations of words, we can calculate word similarity

- We can use row vectors $\{\boldsymbol{w}_k\}$ to represent words by considering each document as a context
- A typical way of measuring word similarity is using cosine similarity

$$\text{cos-sim}(\boldsymbol{w}_k, \boldsymbol{w}_{k'}) = \frac{\boldsymbol{w}_k^\top \boldsymbol{w}_{k'}}{\|\boldsymbol{w}_k\|_2 \cdot \|\boldsymbol{w}_{k'}\|_2}$$

- $\boldsymbol{w}_k^\top \boldsymbol{w}_{k'} = \sum_{i=1} w_{k,i} w_{k',i}$
- $\|\boldsymbol{w}_k\|_2 = \sqrt{\langle \boldsymbol{w}_k, \boldsymbol{w}_k \rangle}$

The Sparsity Issue in Representations

Compute the dot product of the following two pairs

- $w_1^T w_2 = 0$
- $w_2^T w_3 = 0$

Word	Documents							
	1	2	3	4	5	6	7	8
w_1	0	1	0	0	0	0	0	0
w_2	0	0	1	0	0	3	0	0
w_3	1	0	0	2	0	0	5	0
w_4	3	0	0	1	1	0	2	0
w_5	0	1	3	0	1	2	1	0
w_6	1	2	0	0	0	0	1	0
w_7	0	1	0	1	0	1	0	1
w_8	0	0	0	0	0	7	0	0

The Sparsity Issue in Representations

Compute the dot product of the following two pairs

- $w_1^T w_2 = 0$
- $w_2^T w_3 = 0$

Word	Documents							
	1	2	3	4	5	6	7	8
w_1	0	1	0	0	0	0	0	0
w_2	0	0	1	0	0	3	0	0
w_3	1	0	0	2	0	0	5	0
w_4	3	0	0	1	1	0	2	0
w_5	0	1	3	0	1	2	1	0
w_6	1	2	0	0	0	0	1	0
w_7	0	1	0	1	0	1	0	1
w_8	0	0	0	0	0	7	0	0

The sparsity issue will get even worse when we have a large vocab

Compressing Sparse Raw Vectors

Two constraints:

- low-dimensional dense vectors
- contain similar information as the original sparse vectors

Singular Value Decomposition (SVD)

Using SVD, the word-document matrix C can be decomposed into a multiplication of three matrices

$$C = U_0 \cdot \Sigma_0 \cdot V_0^T$$

- $U_0 \in \mathbb{R}^{v \times v}$ is an orthonormal matrix
- $V_0 \in \mathbb{R}^{d \times d}$ is an orthonormal matrix
- $\Sigma_0 \in \mathbb{R}^{v \times d}$ is a diagonal matrix — each component on the diagonal is called a singular value

SVD: Example

Given a matrix C as

$$C = \begin{bmatrix} 1.0 & 2.0 \\ 3.0 & 4.0 \end{bmatrix} \tag{4}$$

The decomposition is

$$U = \begin{bmatrix} -0.40 & -0.91 \\ -0.91 & 0.40 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 5.46 & 0 \\ 0 & 0.37 \end{bmatrix} \quad V^T = \begin{bmatrix} -0.58 & -0.82 \\ 0.82 & -0.58 \end{bmatrix} \tag{5}$$

To obtain a low-dimensional approximation of C, we can remove one of the singular values. But what matters is which one we are going to remove?

SVD: Example

- Option 1: remove the first singular value

$$C_1 = \begin{bmatrix} -0.40 & -0.91 \\ -0.91 & 0.40 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 \\ 0 & 0.37 \end{bmatrix} \cdot \begin{bmatrix} -0.58 & -0.82 \\ 0.82 & -0.58 \end{bmatrix}$$

$$= 0.37 \cdot \begin{bmatrix} -0.91 \\ 0.40 \end{bmatrix} \cdot [0.82 \quad -0.58] = \begin{bmatrix} -0.28 & 0.20 \\ 0.12 & -0.09 \end{bmatrix}$$

SVD: Example

- Option 1: remove the first singular value

$$C_1 = \begin{bmatrix} -0.40 & -0.91 \\ -0.91 & 0.40 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 \\ 0 & 0.37 \end{bmatrix} \cdot \begin{bmatrix} -0.58 & -0.82 \\ 0.82 & -0.58 \end{bmatrix}$$

$$= 0.37 \cdot \begin{bmatrix} -0.91 \\ 0.40 \end{bmatrix} \cdot [0.82 \quad -0.58] = \begin{bmatrix} -0.28 & 0.20 \\ 0.12 & -0.09 \end{bmatrix}$$

- Option 2: remove the second singular value

$$C_2 = \begin{bmatrix} -0.40 & -0.91 \\ -0.91 & 0.40 \end{bmatrix} \cdot \begin{bmatrix} 5.46 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} -0.58 & -0.82 \\ 0.82 & -0.58 \end{bmatrix}$$

$$= 5.46 \cdot \begin{bmatrix} -0.40 \\ -0.91 \end{bmatrix} \cdot [-0.58 \quad -0.82] = \begin{bmatrix} 1.26 & 1.79 \\ 2.88 & 4.07 \end{bmatrix}$$

SVD: Example

- Option 1: remove the first singular value

$$C_1 = \begin{bmatrix} -0.40 & -0.91 \\ -0.91 & 0.40 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 \\ 0 & 0.37 \end{bmatrix} \cdot \begin{bmatrix} -0.58 & -0.82 \\ 0.82 & -0.58 \end{bmatrix}$$

$$= 0.37 \cdot \begin{bmatrix} -0.91 \\ 0.40 \end{bmatrix} \cdot [0.82 \quad -0.58] = \begin{bmatrix} -0.28 & 0.20 \\ 0.12 & -0.09 \end{bmatrix}$$

- Option 2: remove the second singular value

$$C_2 = \begin{bmatrix} -0.40 & -0.91 \\ -0.91 & 0.40 \end{bmatrix} \cdot \begin{bmatrix} 5.46 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} -0.58 & -0.82 \\ 0.82 & -0.58 \end{bmatrix}$$

$$= 5.46 \cdot \begin{bmatrix} -0.40 \\ -0.91 \end{bmatrix} \cdot [-0.58 \quad -0.82] = \begin{bmatrix} 1.26 & 1.79 \\ 2.88 & 4.07 \end{bmatrix}$$

Therefore, $\|C - C_1\|_F > \|C - C_2\|_F$. In other words, removing the smaller singular value creates a better low-dimensional approximation.

SVD for Approximation

With SVD, we can approximate $\boldsymbol{C}$ only keep the first k singular values in $\boldsymbol{\Sigma}_0$, as $\boldsymbol{\Sigma}$

$$\boldsymbol{C} \approx \underbrace{\begin{bmatrix} | & & | \\ \boldsymbol{u}_1 & \dots & \boldsymbol{u}_k \\ | & & | \end{bmatrix}}_{\boldsymbol{U}} \cdot \underbrace{\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix}}_{\boldsymbol{\Sigma}} \cdot \underbrace{\begin{bmatrix} — & \boldsymbol{v}_1 & — \\ & \vdots & \\ — & \boldsymbol{v}_k & — \end{bmatrix}}_{\boldsymbol{V}^\top} \tag{11}$$

where $\boldsymbol{U} \in \mathbb{R}^{v \times k}$, $\boldsymbol{V} \in \mathbb{R}^{d \times k}$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{k \times k}$.

Empirically, k=200~400

Lower-Dimensional Word Embeddings

Given

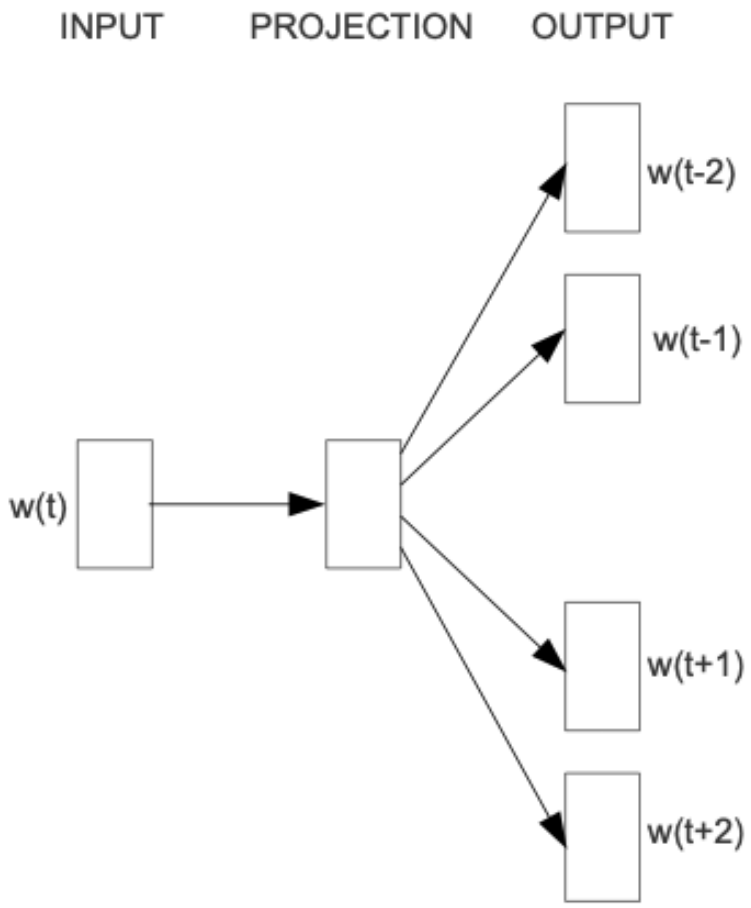
$$\boldsymbol{C} \approx \boldsymbol{U} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}^{\mathsf{T}} \tag{12}$$

to construct low-dimensional word representation, we can multiply $\boldsymbol{V}$ on both side of equation 12 and then have

$$\boldsymbol{W} = \boldsymbol{U} \cdot \boldsymbol{\Sigma} \approx \boldsymbol{C} \cdot \boldsymbol{V} \in \mathcal{R}^{v \times k} \tag{13}$$

Skip-Gram

Instead of using matrix decomposition, a different strategy of learning word embeddings is using a word to predict its surrounding words



[Mikolov et al., 2013]

Pretrained Word Embeddings

- Word2Vec [Mikolov et al., 2013]
  - capturing syntactic and semantic relationships
- GloVe [Pennington et al., 2014]
  - capturing global co-occurrence patterns

Outline

- Introduction to NLP
- Text Classification
- Word Embeddings
- Natural Language Generation
- Modern Language Models

Natural Language Generation

Natural language generation is one side of natural language processing:

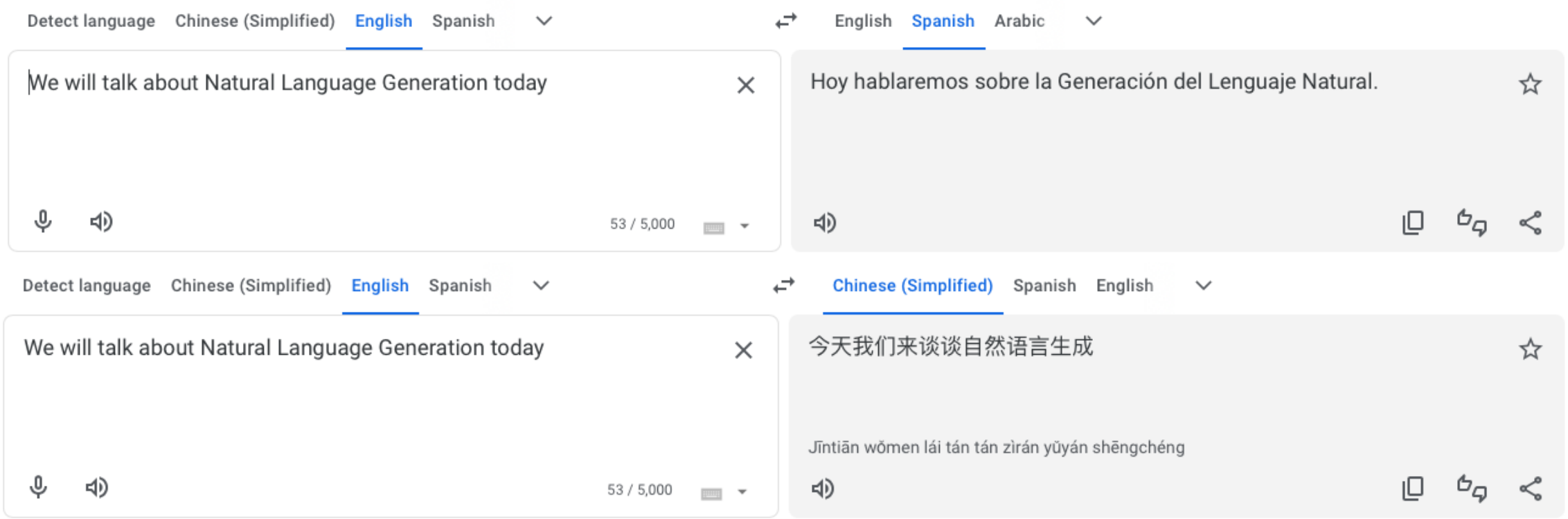
NLP = Natural Language Understanding (NLU) + Natural Language Generation (NLG)

NLG focuses on systems that produce fluent, coherent and useful language output for human consumption

Example: Machine Translation

Input: texts in source languages

Output: translated texts in target languages

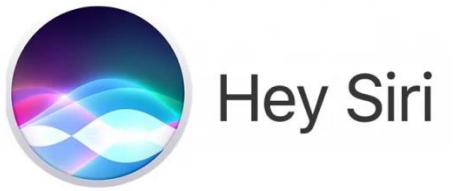


Example: Conversational Systems

Input: dialog history

Output: texts that respond to or continue the conversation

Siri, Alexa, Google Assistant...



Example: Summarization

Input: long documents

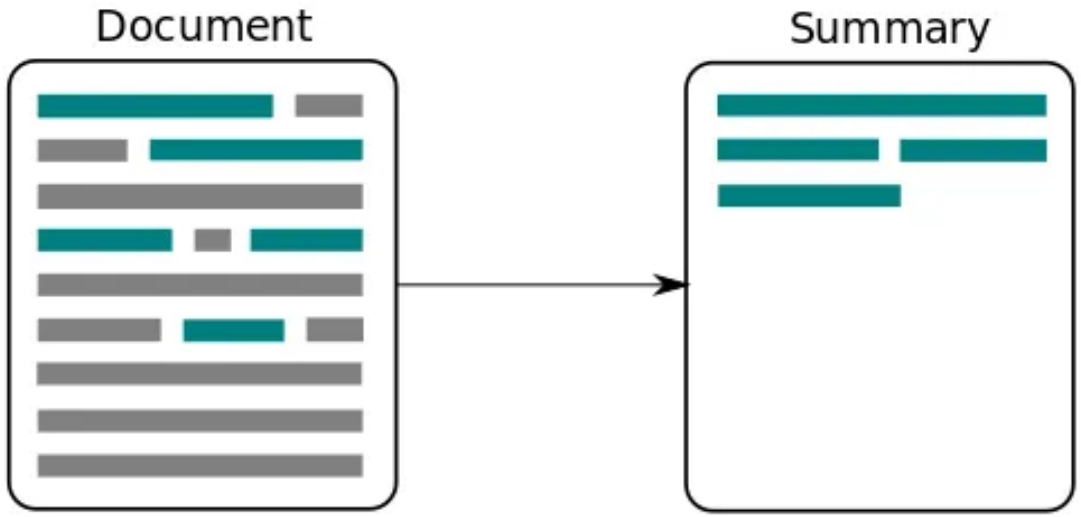
Output: short summaries

Research articles

Meeting notes

Books

…



Advanced NLG Systems

ChatGPT

Create a story about Rice University

Title: The Heartbeat of Rice

Once upon a time, nestled in the heart of Houston, Texas, there was a place where intellectual curiosity met tradition, and where science and art existed side by side. That place was Rice University, a community of scholars, dreamers, and doers. Rice wasn't just a university; it was a thriving ecosystem of creativity and knowledge, a place where the impossible felt possible and every idea had room to grow.

It all began in 1912, when the university was founded by William Marsh Rice, a visionary philanthropist who believed that education should be available to all who sought it. His legacy became the foundation for the university that now sits in the midst of a bustling city, but it remains a serene haven for those who crave innovation.

The story of Rice University unfolds across its iconic, ivy-clad buildings and green quads, where students and professors walk side by side, engaged in animated conversations. One such student, named Maria, had just arrived at Rice, eager to make her mark. Maria had always been passionate about space, dreaming of one day working for NASA. But she knew that the journey wouldn't be easy. Rice's competitive environment challenged students to reach higher, dig deeper, and push boundaries, but it also nurtured their dreams in ways that few other places could.

DeepSeek

Create a story about Rice University

Title: The Heart of the Owl

In the heart of Houston, Texas, where the bustling city meets serene oak-lined streets, there lies a campus unlike any other. Rice University, with its iconic architecture, sprawling lawns, and a sense of timelessness, has long been a beacon of knowledge and innovation. But beyond its academic prestige, Rice holds a secret—a story whispered among students late at night, a tale of mystery and magic that binds the university together.

It was the fall of 1912, just as Rice Institute (as it was then known) was preparing to open its doors for the first time. William Marsh Rice, the university's founder, had envisioned a place of unparalleled learning, a sanctuary where students could pursue their dreams without the burden of financial worry. But as the first bricks were laid, something extraordinary happened.

Legend has it that a great horned owl, a creature often associated with wisdom, appeared on the construction site. The workers, startled by its piercing gaze, watched as it circled the rising buildings before landing on the cornerstone of the Academic Quad. The owl stayed there for three days and three nights, as if blessing the institution. On the fourth morning, it vanished, leaving behind a single feather and a faint, glowing symbol etched into the stone—a symbol no one could decipher.

Categorization

The output space is not very diverse

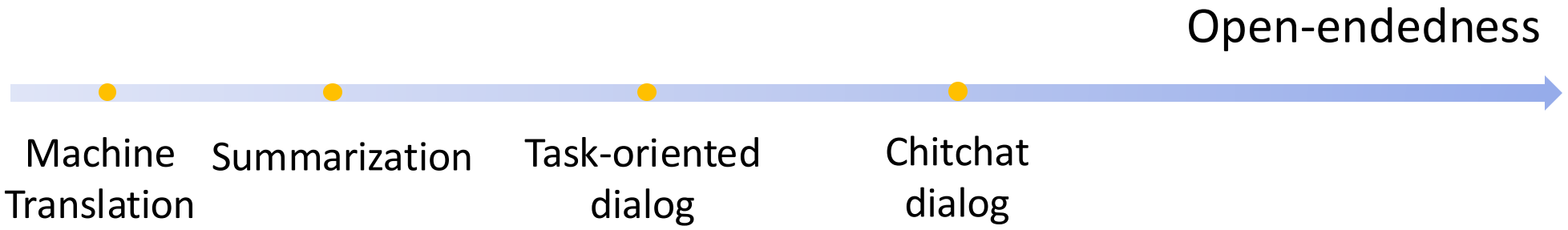
Source: “下周我们有期中考试”

Translations:

“We have midterm exams next week.”

“Next week, we have midterm exams.”

Categorization



The output space is more diverse

Input: “Hey, how are you?”

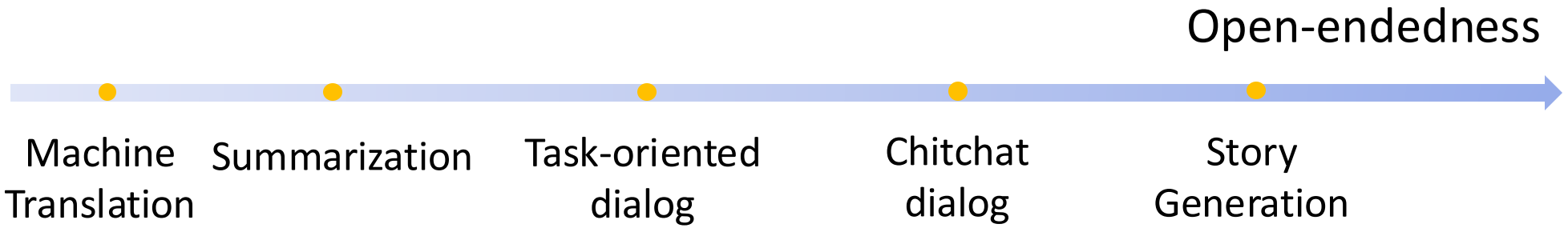
Outputs:

“Good! How about you?”

“Could be better, but I’m hanging in there.”

“I'm doing great! I just got some amazing news—I landed my dream job!”

Categorization



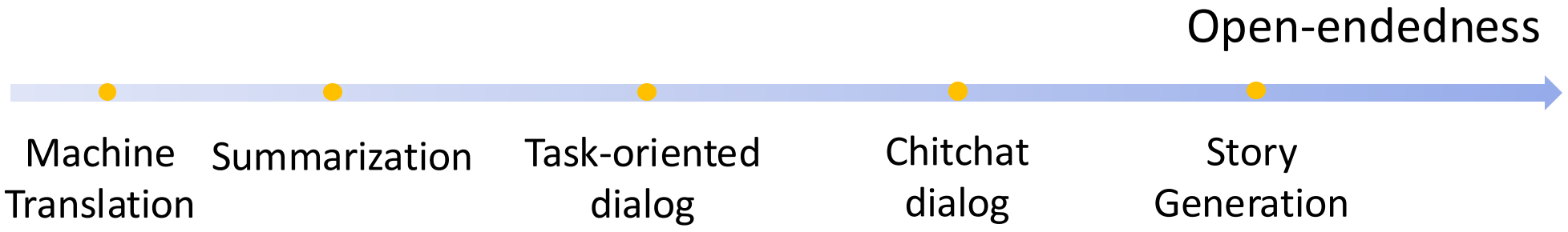
The output space is extremely diverse

Input: “Write a story about Mars”

Outputs:

“…”

Categorization



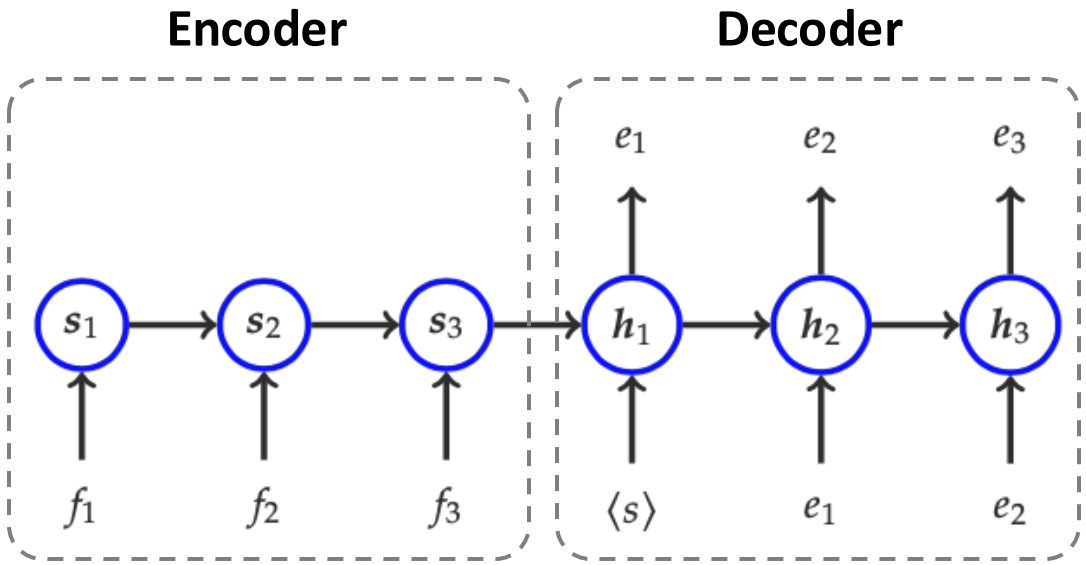
Open-ended generation: the output space is large and diverse

Non-open-ended generation: the output is mostly determined by the input

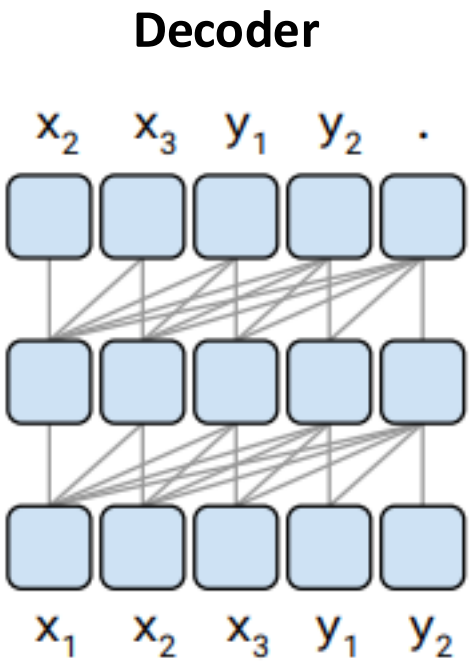
These two classes of NLG tasks require different decoding and/or training approaches

Neural NLG Models

Non-open-ended generation



Open-ended generation



Writing a Poem

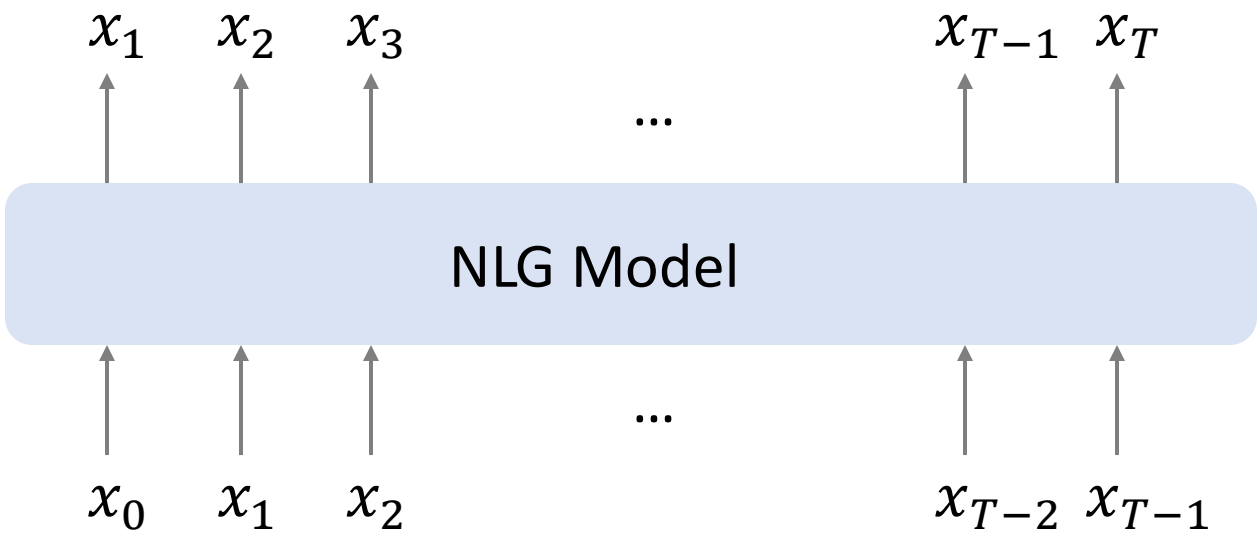
Trevor Noah and Amanda Gorman writing poems with the input methods on their phones

https://www.youtube.com/watch?v=93tGah-gQW0

Training Objective

At each time step, maximize the probability of the next token x_t given preceding tokens $\boldsymbol{x}_{1:t-1}$

$$\mathcal{L} = -\sum_{t=1}^{T} \log p(x_t | \boldsymbol{x}_{1:t-1})$$



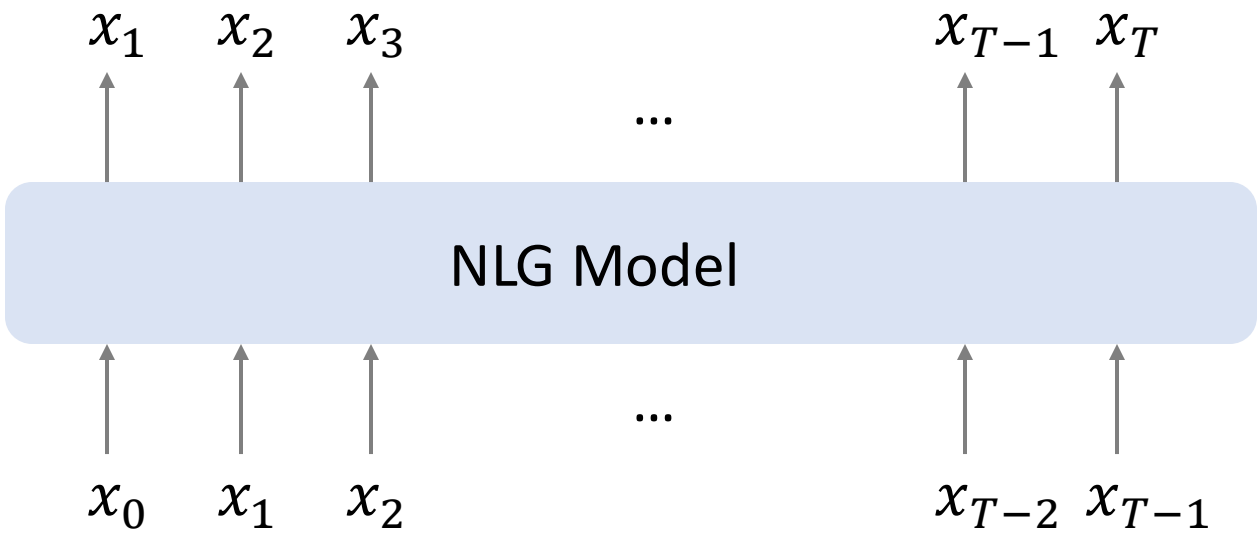
Training Objective

At each time step, maximize the probability of the next token x_t given preceding tokens $\boldsymbol{x}_{1:t-1}$

$$\mathcal{L} = -\sum_{t=1}^{T} \log p(x_t | \boldsymbol{x}_{1:t-1})$$

During training, we use ground-truth tokens $\{x_t^*\}$

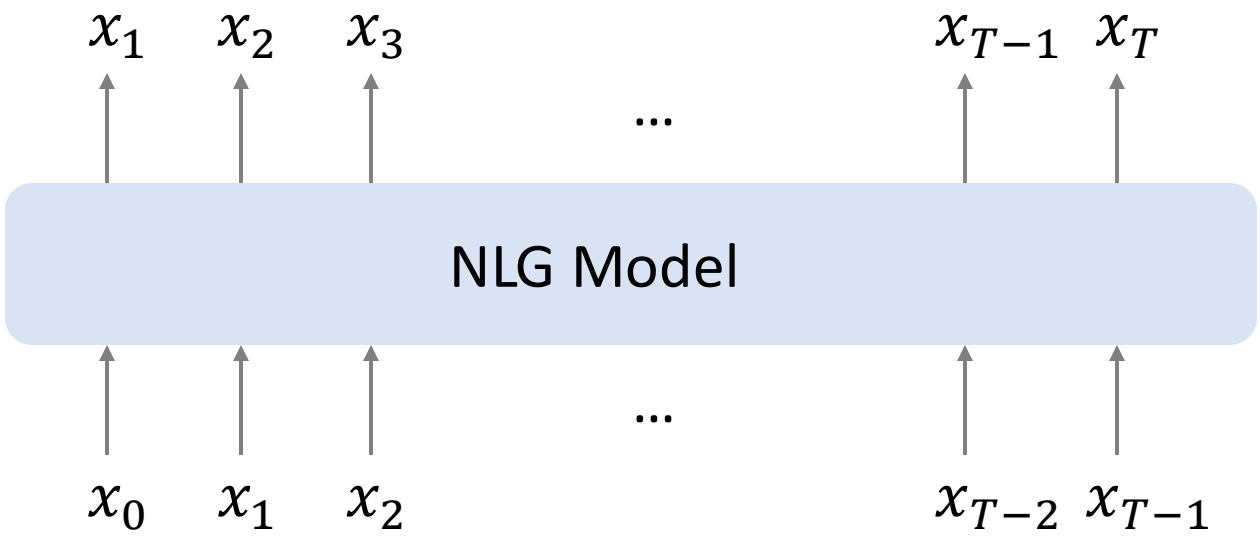
"Teacher Forcing"



Inference

The decoding algorithm defines a function $g(\cdot)$ to select a token from the output distribution

$$\hat{x}_t = g\big(p(x_t|\boldsymbol{x}_{1:t-1})\big)$$

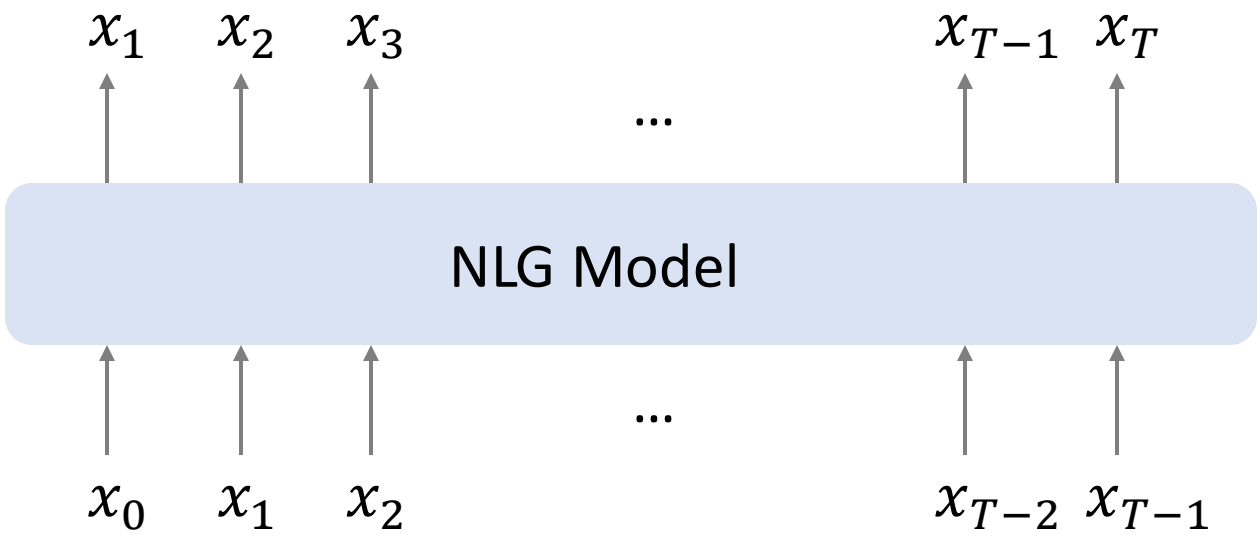


Inference

The decoding algorithm defines a function $g(\cdot)$ to select a token from the output distribution

$$\hat{x}_t = g\big(p(x_t|\boldsymbol{x}_{1:t-1})\big)$$

What is the naive decoding algorithm?



Decoding Algorithms

Greedy decoding: select the token with the highest probability

$$\hat{x}_t = \underset{w \in \mathcal{V}}{\operatorname{argmax}}\, p(x_t = w | \boldsymbol{x}_{1:t-1})$$

$\mathcal{V}$: vocab

Decoding Algorithms

Greedy decoding: select the token with the highest probability

$$\hat{x}_t = \underset{w \in \mathcal{V}}{\operatorname{argmax}}\, p(x_t = w | \boldsymbol{x}_{1:t-1})$$

$\mathcal{V}$: vocab

Limitations:

- Suboptimal output sequence
- Lack of diversity
- Repetition

Decoding Algorithms

Greedy decoding: select the token with the highest probability

$$\hat{x}_t = \underset{w \in \mathcal{V}}{\operatorname{argmax}}\, p(x_t = w | \boldsymbol{x}_{1:t-1})$$

$\mathcal{V}$: vocab

Limitations:

- Suboptimal output sequence
- Lack of diversity
- Repetition

Example

"The forest was quiet, and the air smelled of pine pine pine…"

Decoding Algorithms

Beam search decoding: keep track of the k most probable outputs at each step

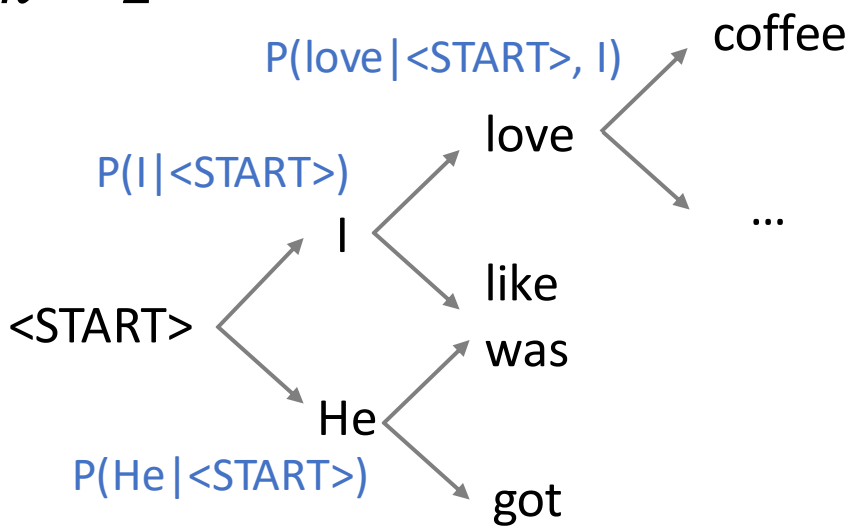
$\{\hat{x}_{t,1}, \hat{x}_{t,2}, \cdots, \hat{x}_{t,k}\}$ k: beam size

Decoding Algorithms

Beam search decoding: keep track of the k most probable outputs at each step

$$\{\hat{x}_{t,1}, \hat{x}_{t,2}, \cdots, \hat{x}_{t,k}\} \qquad k\text{: beam size}$$

$k = 2$

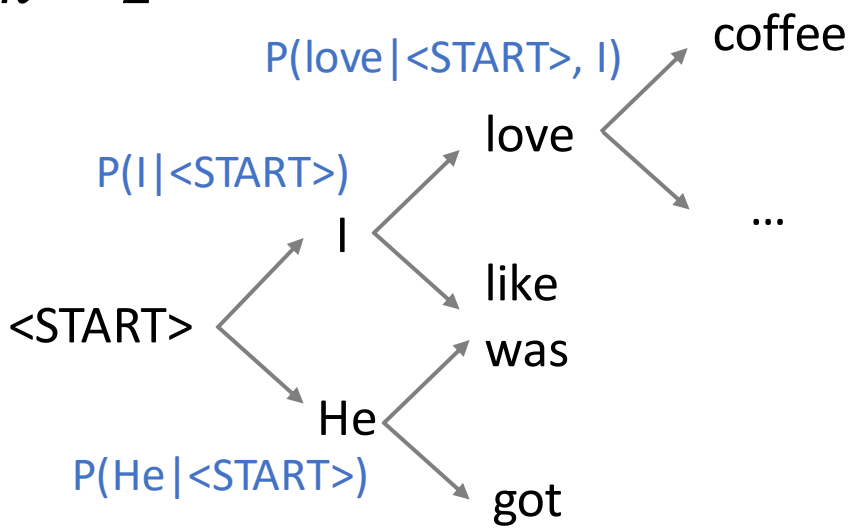


Decoding Algorithms

Beam search decoding: keep track of the k most probable outputs at each step

$$\{\hat{x}_{t,1}, \hat{x}_{t,2}, \cdots, \hat{x}_{t,k}\} \qquad k\text{: beam size}$$

$k = 2$



k^t outputs:

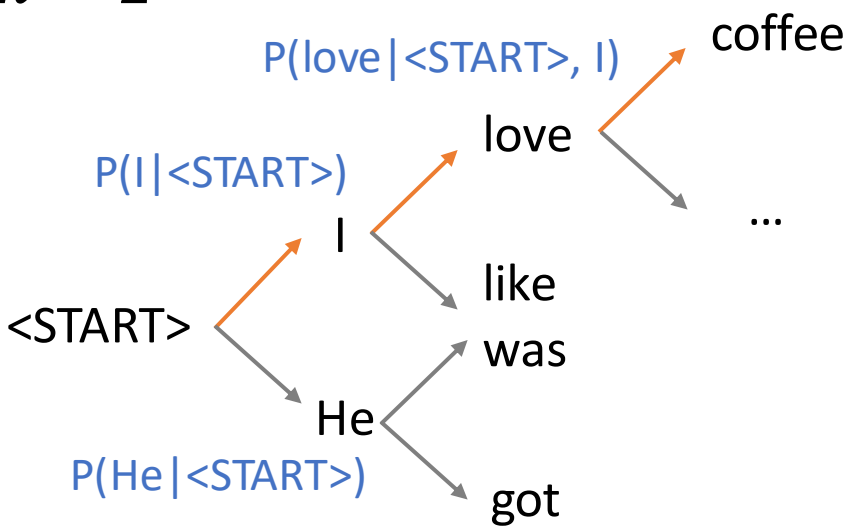
- Different hypotheses may produce <END> tokens on different timesteps
- Stop: reach timestep T or complete n hypotheses

Decoding Algorithms

Beam search decoding: keep track of the k most probable outputs at each step

$$\{\hat{x}_{t,1}, \hat{x}_{t,2}, \cdots, \hat{x}_{t,k}\} \qquad k\text{: beam size}$$

$k = 2$



Select the hypothesis with the highest probability

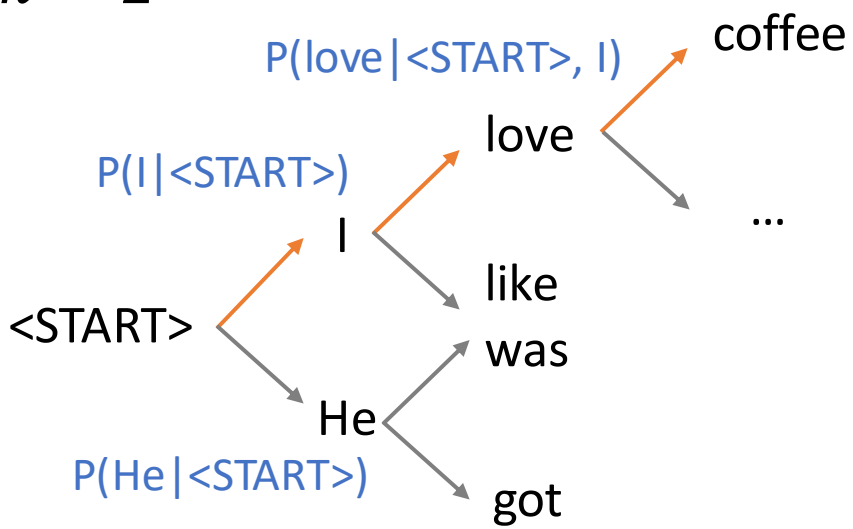
$$p(\hat{x}_1, \hat{x}_2, \cdots \hat{x}_T) = \prod_{t=1}^{T} p(x_t | \boldsymbol{x}_{1:t-1})$$

Decoding Algorithms

Beam search decoding: keep track of the k most probable outputs at each step

$$\{\hat{x}_{t,1}, \hat{x}_{t,2}, \cdots, \hat{x}_{t,k}\} \qquad k\text{: beam size}$$

$k = 2$



Problem

Longer hypotheses may have lower scores

Solution

Normalize by length

$$\frac{1}{T}\sum_{t=1}^{T} \log p(x_t|\boldsymbol{x}_{1:t-1})$$

Decoding Algorithms

Search based decoding (greedy, beam search) is suitable for non-open-ended tasks (e.g., machine translation, summarization)

NOT suitable for open-ended tasks

[Holtzman et. al., ICLR 2020]

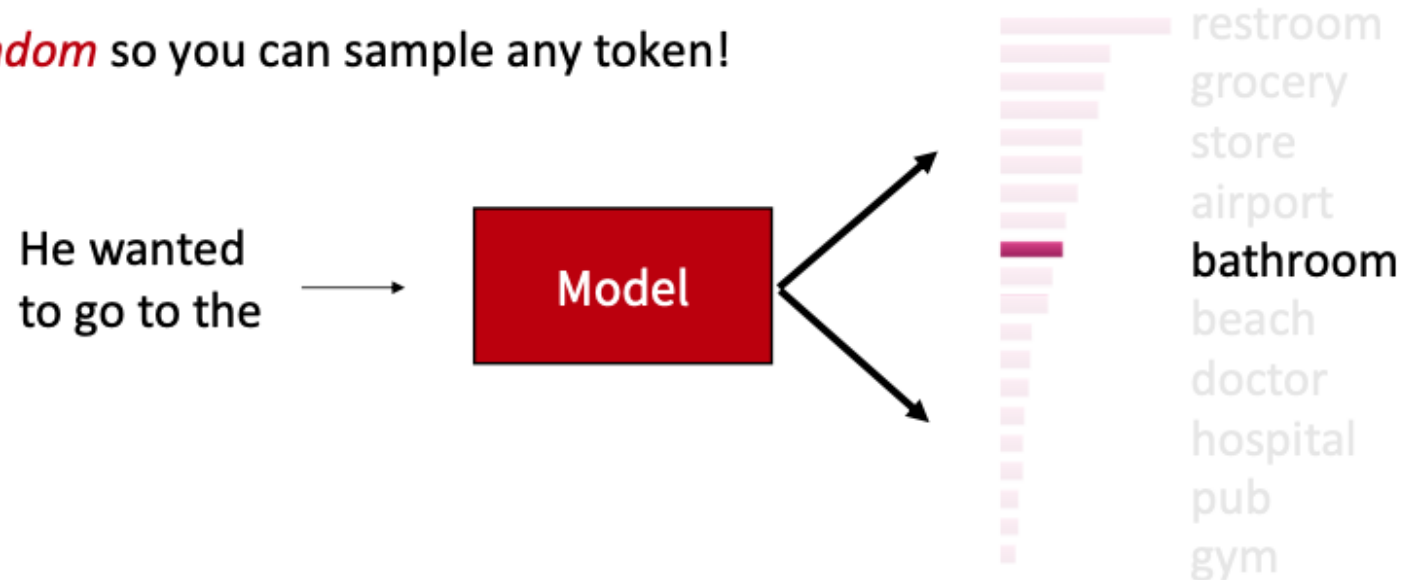
Context:
In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.
Beam Search, b=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

Decoding Algorithms

Sampling based decoding: sampling from the token conditional probability distribution

$$\hat{x}_t \sim p(x_t | \boldsymbol{x}_{1:t-1})$$

It's *random* so you can sample any token!



Decoding Algorithms

Problem:

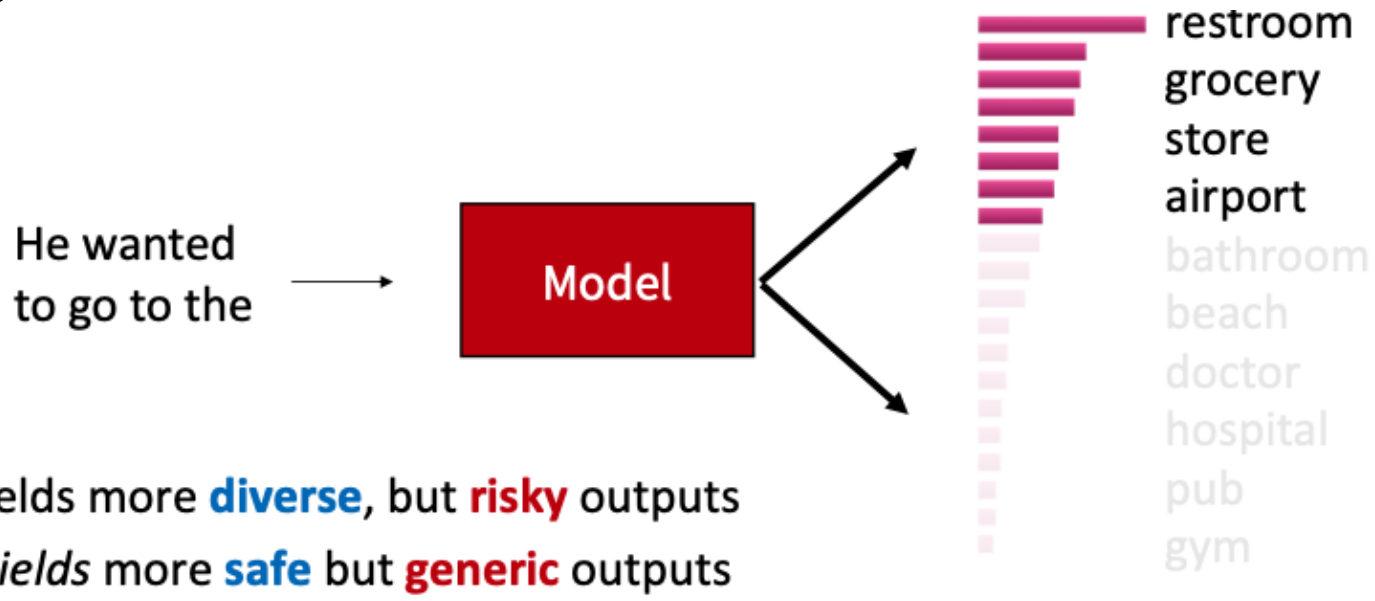
- Vanilla sampling makes every token in the vocabulary an option
- Many tokens are probably wrong in the current context
- For these wrong tokens, we give them individually a tiny chance to be selected
- As there are many of them, we still give them as a group a high chance to be selected

Decoding Algorithms

Solution

Top-k sampling: only sample from the top k tokens in the probability distribution

In practice: k = 50

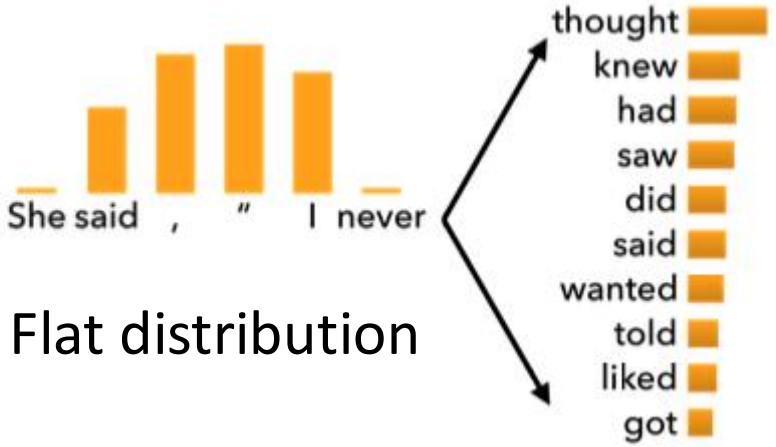


Increase *k* yields more **diverse**, but **risky** outputs

Decrease *k yields* more **safe** but **generic** outputs

Decoding Algorithms

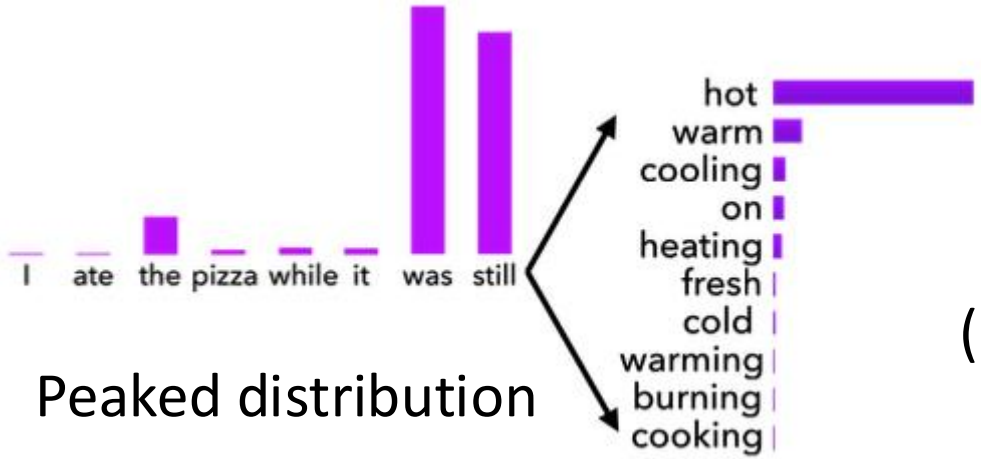
The selection of k is tricky



Flat distribution

Top-*k* sampling can cut off too ***quickly***!

(a small k removes many viable options)



Peaked distribution

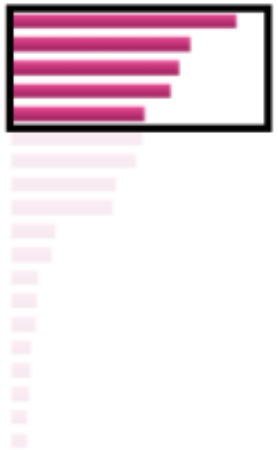
Top-*k* sampling can also cut off too ***slowly***!

(a large k allows for too many unlikely options)

Decoding Algorithms

Top-p (nucleus) sampling: sample from all tokens in the top p cumulative probability mass (i.e., where mass is concentrated)

$P_t^1(y_t = w \mid \{y\}_{<t})$



$P_t^2(y_t = w \mid \{y\}_{<t})$

$P_t^3(y_t = w \mid \{y\}_{<t})$

Scaling Randomness: Temperature

Prediction probability:

$$p(x_t = w|\boldsymbol{x}_{1:t-1}) = \frac{\exp(s_w)}{\sum_{w' \in \mathcal{V}} \exp(s_{w'})}$$

Apply a temperature hyperparameter τ to the softmax function:

$$p(x_t = w|\boldsymbol{x}_{1:t-1}) = \frac{\exp(s_w/\tau)}{\sum_{w' \in \mathcal{V}} \exp(s_{w'}/\tau)}$$

- Large temperature ($\tau > 1$): the distribution is more uniform
  - More diverse output(probability is spread around vocab)
- Small temperature ($\tau < 1$): the distribution becomes more peaked
  - Less diverse output (probability is concentrated on top words)

Types of Evaluations

Content Overlap Metrics

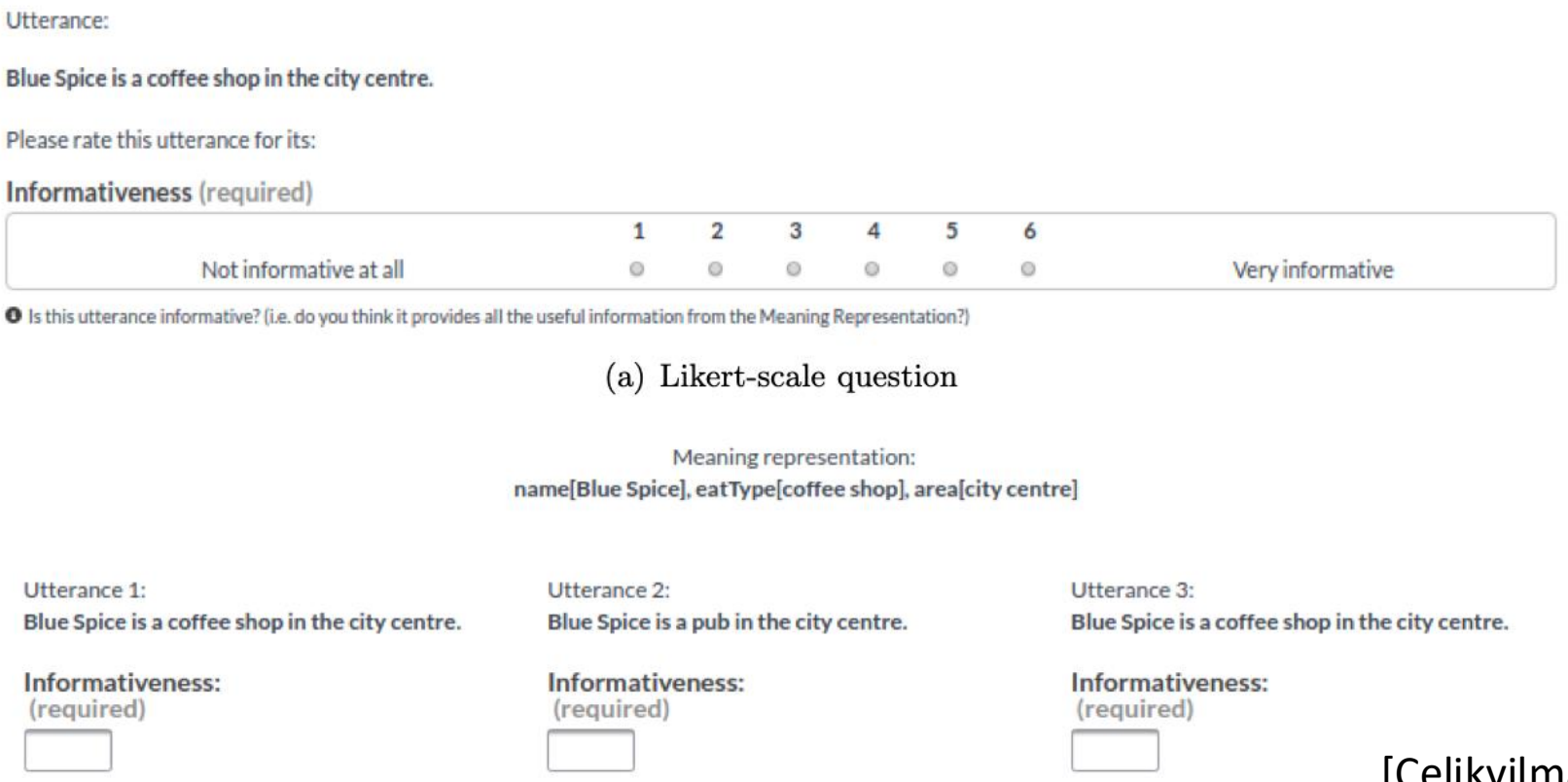
Model-based Metrics

Human Evaluations

Human Evaluation

Dimensions: fluency, coherence, correctness, informativeness…

Formats: single sample with a Likert scale, pairwise comparison, ranking



[Celikyilmaz et al., 2021]

Human Evaluation

Problems

- Subjectivity
- Participant background
- Inconsistency across evaluators
- Non-reproducibility
- Costs
- …

Automatic Evaluation: BLEU

BLEU was originally designed to evaluate machine translation

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

- p_n: n-gram precision
- w_n: weight for n-gram precision, usually $w_n = \frac{1}{4}$
- BP: brevity penalty
- N: the largest length of n-gram, usually $N = 4$

[Papineni et al., ACL 2002]

Automatic Evaluation: BLEU

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\_gram \in C} Count_{match}(n\_gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\_gram' \in C'} Count(n\_gram')}$$

Candidate:

To make people trustworthy you need to trust them

Reference:

The way to make people trustworthy is to trust them

$$p_1 = \frac{7}{9}$$

$$p_2 = \frac{5}{8}$$

Automatic Evaluation: BLEU

The brevity penalty is introduced to penalize shorter generated text

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

$$BP = \begin{cases} 1 & c > r \\ e^{1-r/c} & c \leq r \end{cases}$$

- c: the length of the generated text
- r: the length of the reference text

Automatic Evaluation: ROUGE

ROUGE was originally designed to evaluate summarization

$$\text{ROUGE-N} = \frac{\sum_{S \in \{References\}} \sum_{n\_gram \in S} Count_{match}(n\_gram)}{\sum_{s' \in \{References\}} \sum_{n\_gram' \in S'} Count(n\_gram')}$$

- Recall-based measure

Candidate:

To make people trustworthy you need to trust them

Reference:

The way to make people trustworthy is to trust them

$$\text{ROUGE-1} = \frac{7}{10}$$

[Lin, ACL 2004]

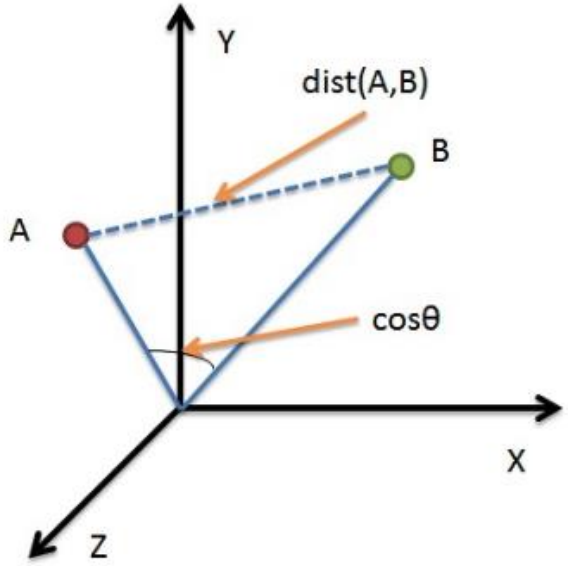
Automatic Evaluation

N-gram overlap metrics (BLEU, ROUGE)

- Fast and efficient and widely used
- Evaluate surface-form similarity
- Fail to evaluate semantic similarity
- Fail to handle synonymy and paraphrasing
- Insensitive to word order

Automatic Evaluation: model-based metrics

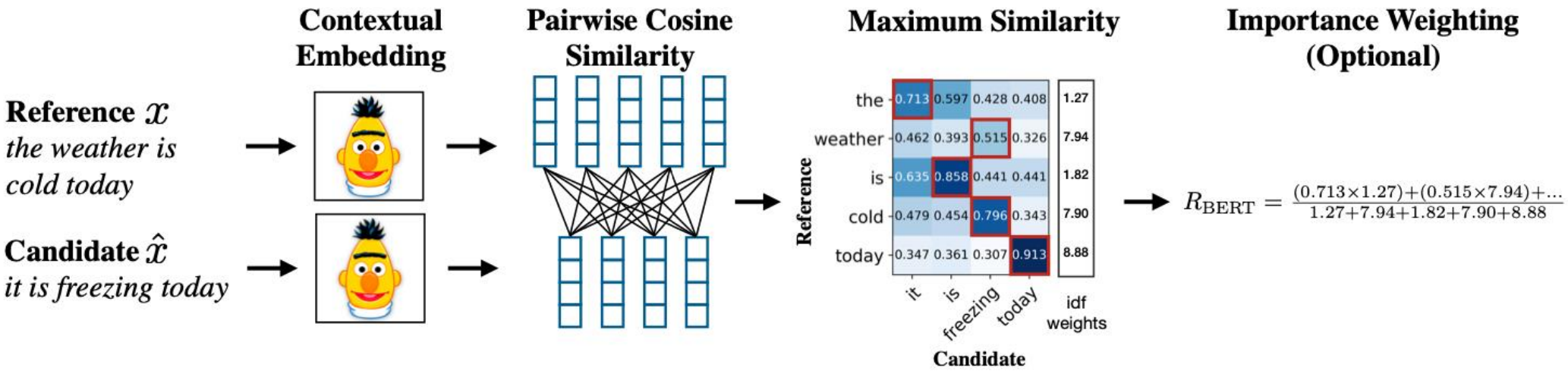
- Use neural network models to encode texts into representations
- The learned representations capture more semantics
- Apply distance metrics to measuring the similarity between representations



Automatic Evaluation: model-based metrics

- Train neural network evaluators
- Use pre-trained models

BERTScore

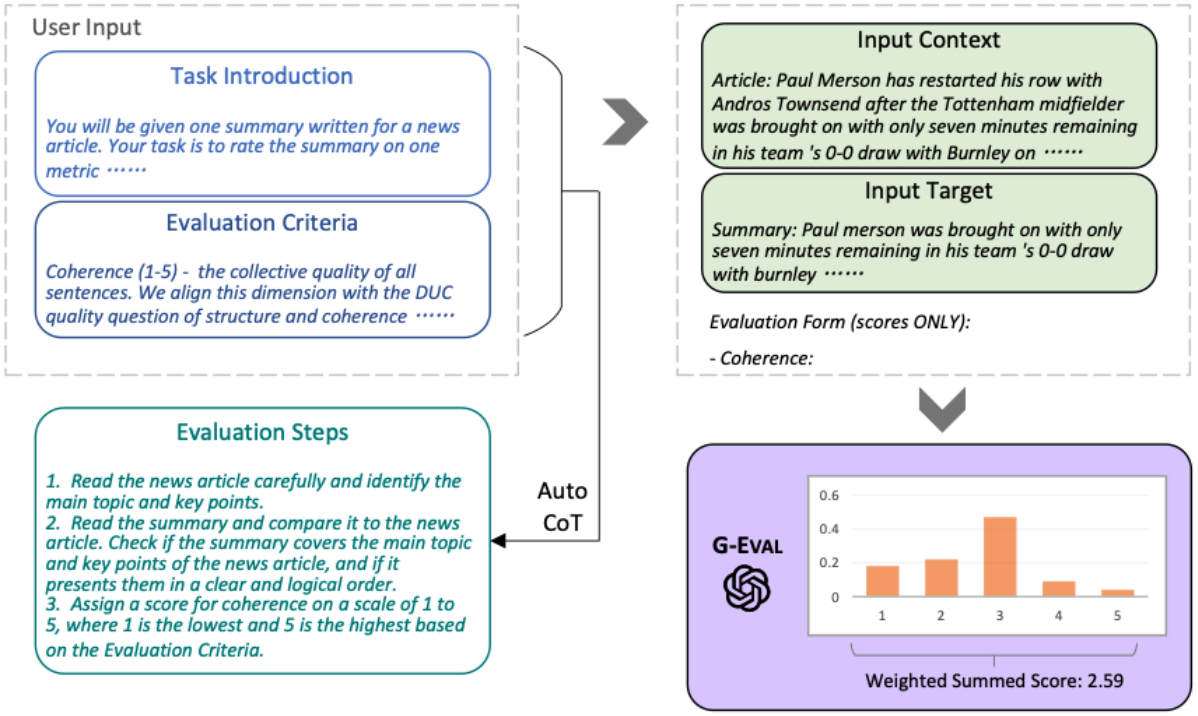


[Zhang et al., ICLR 2020]

Automatic Evaluation: model-based metrics

- Train neural network evaluators
- Use pre-trained models
- Use LLMs

G-EVAL

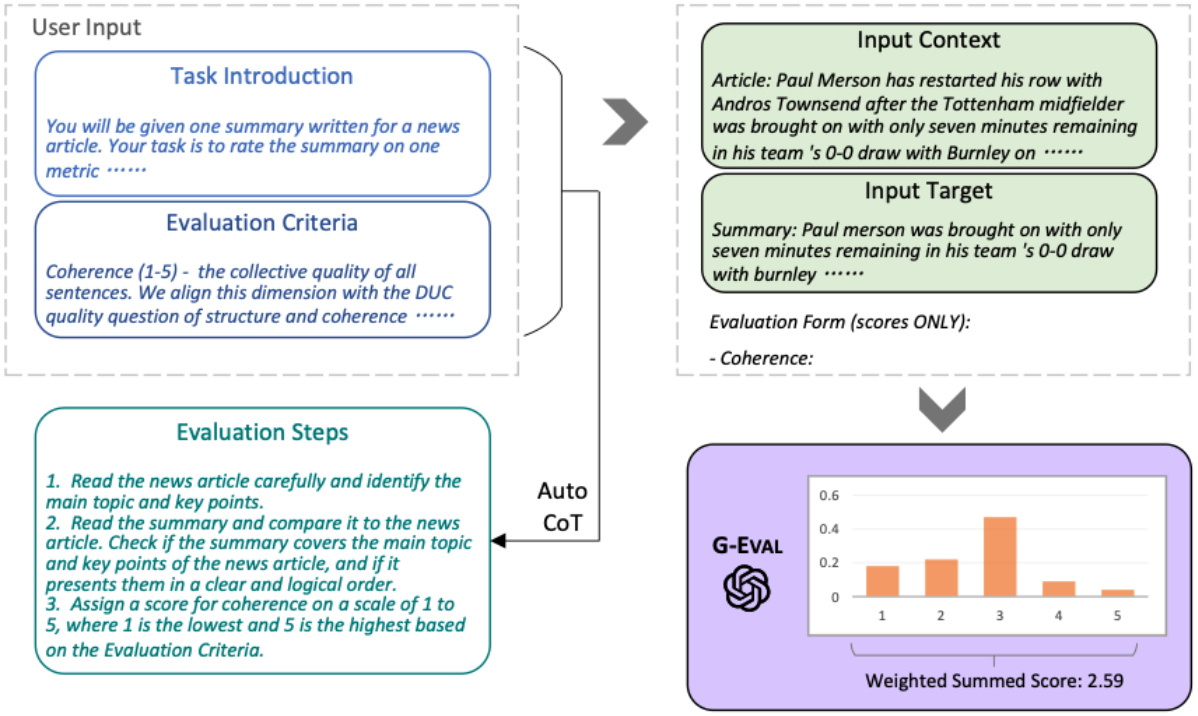


[Liu et al., EMNLP 2023]

Automatic Evaluation: model-based metrics

- Train neural network evaluators
- Use pre-trained models
- Use LLMs

G-EVAL



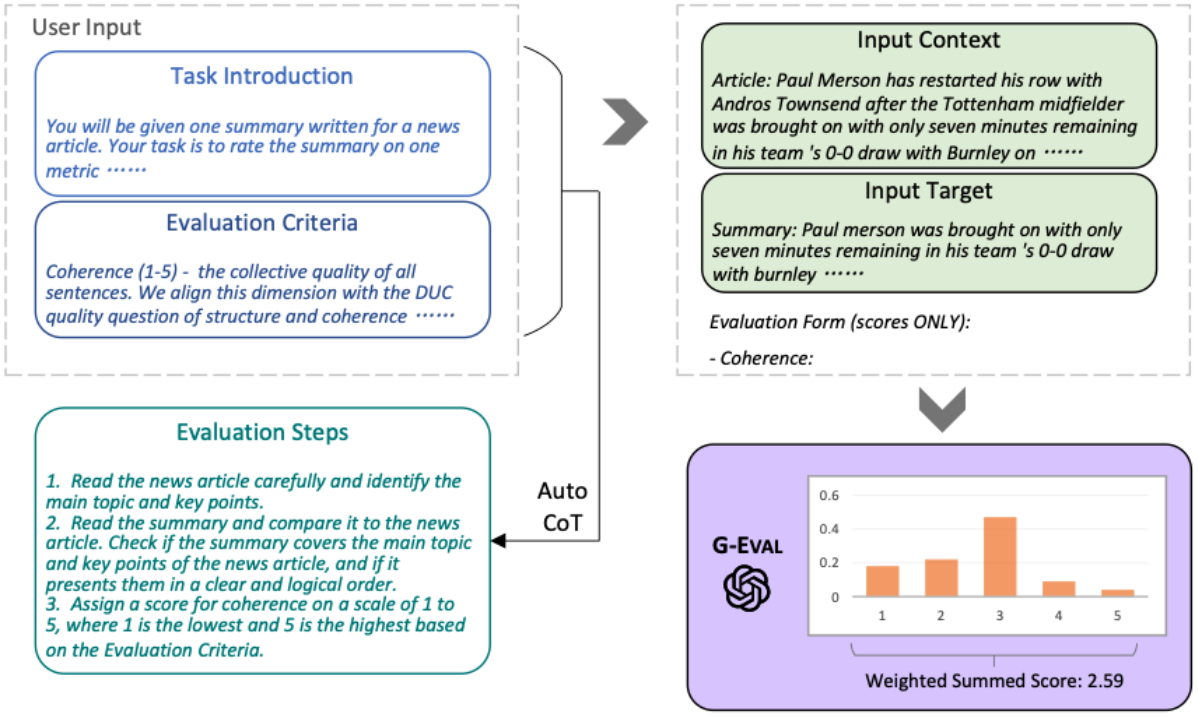
LLM-based evaluation aligns better with human evaluation

[Liu et al., EMNLP 2023]

Automatic Evaluation: model-based metrics

- Train neural network evaluators
- Use pre-trained models
- Use LLMs

G-EVAL



NLG evaluation is still an open question, especially for open-ended generation!

LLM-based evaluation aligns better with human evaluation

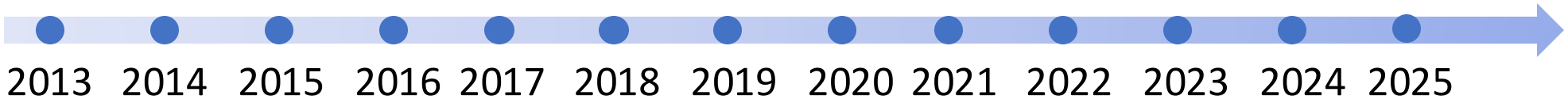
[Liu et al., EMNLP 2023]

Outline

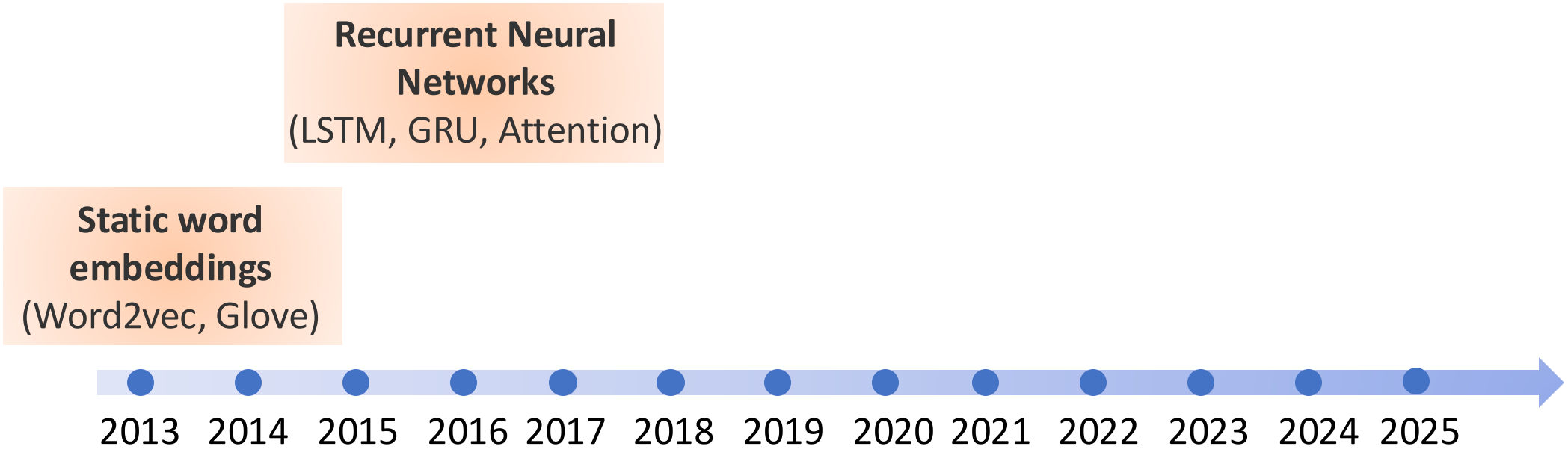
- Introduction to NLP
- Text Classification
- Word Embeddings
- Natural Language Generation
- Modern Language Models

Evolvement

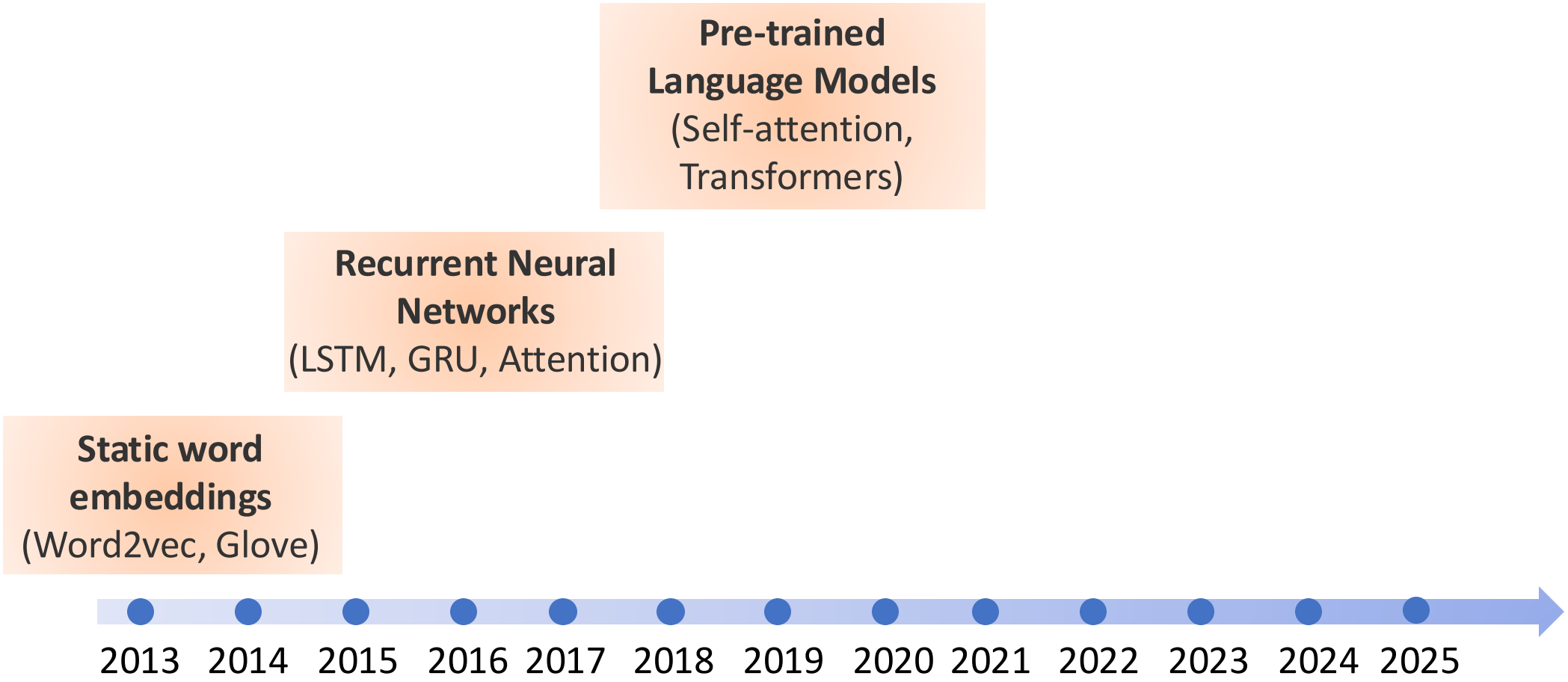
Static word embeddings
(Word2vec, Glove)



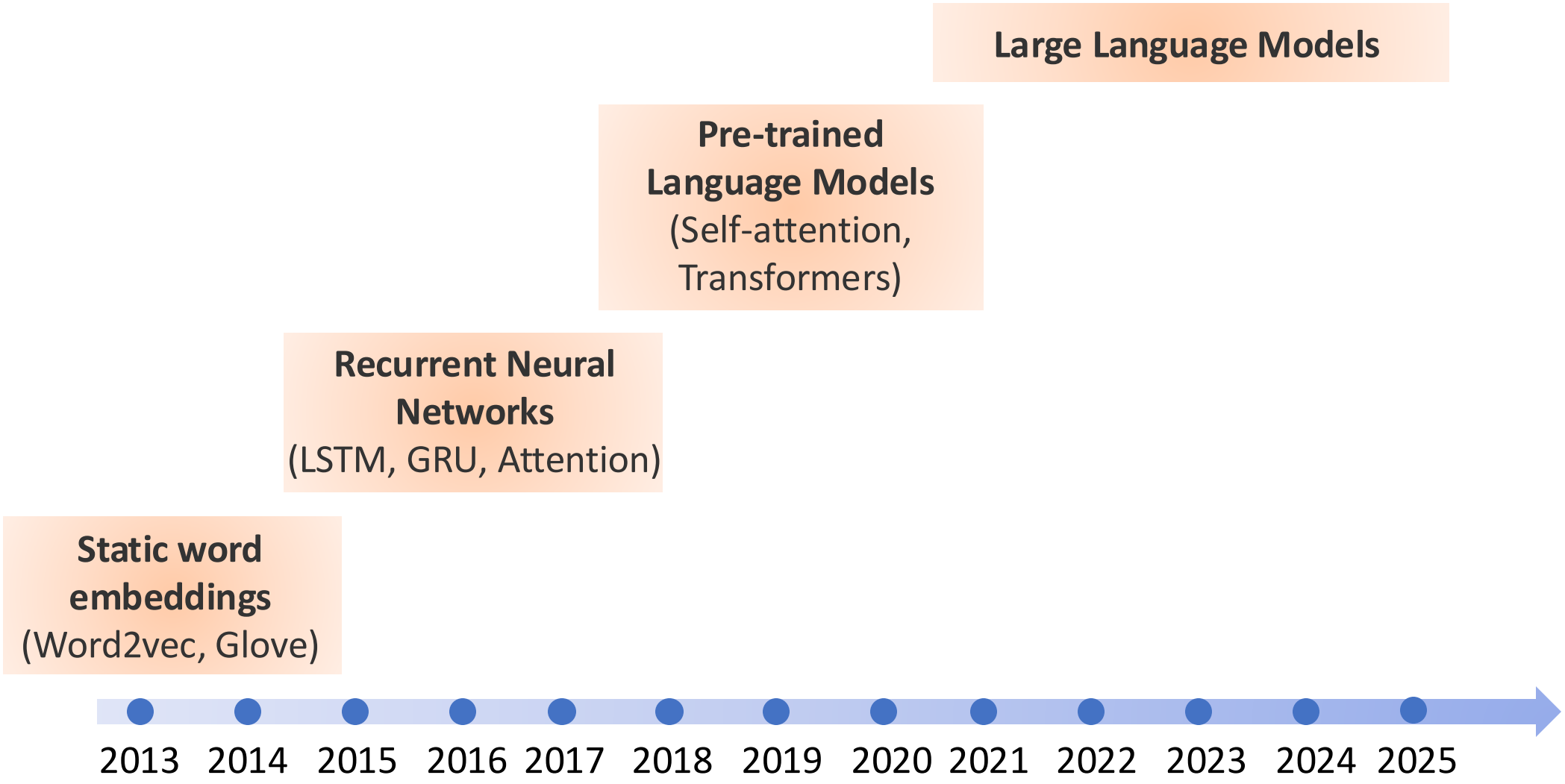
Evolvement


Recurrent Neural Networks
(LSTM, GRU, Attention)
Static word embeddings
(Word2vec, Glove)
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025

Evolvement


Pre-trained Language Models (Self-attention, Transformers)
Recurrent Neural Networks (LSTM, GRU, Attention)
Static word embeddings (Word2vec, Glove)
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025

Evolvement


Large Language Models
Pre-trained Language Models (Self-attention, Transformers)
Recurrent Neural Networks (LSTM, GRU, Attention)
Static word embeddings (Word2vec, Glove)
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025

Large Language Models

Large Data

(Trillions of tokens)

- Wikipedia
- Book corpus
- Web text
- …

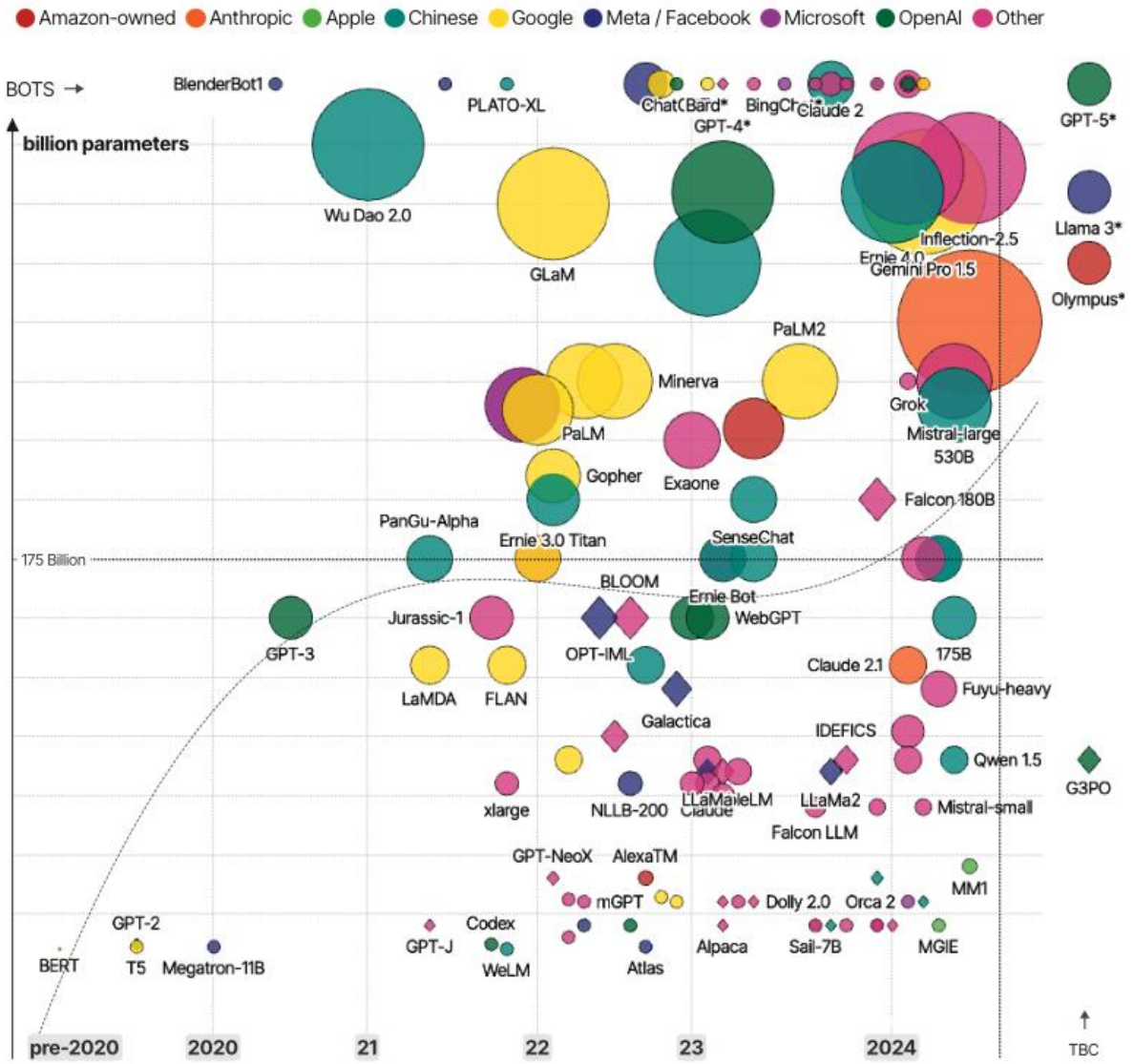
Innovative Techniques

- Pretraining
- Fine-tuning
- Reinforcement learning from human feedback
- …

Computational Power

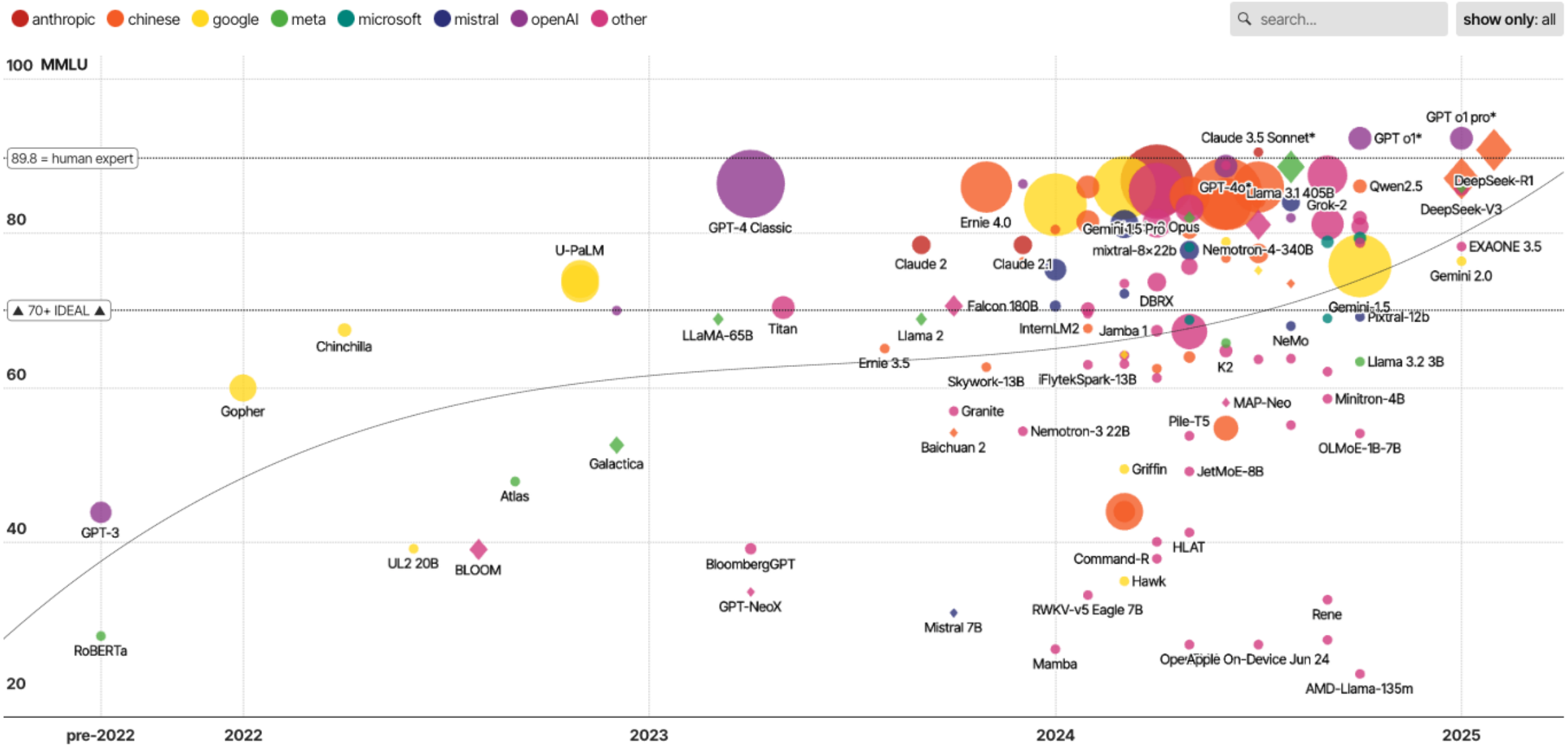
- Thousands of GPUs
- Weeks to months
- High-throughput storage
- …

Large Language Models



https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/

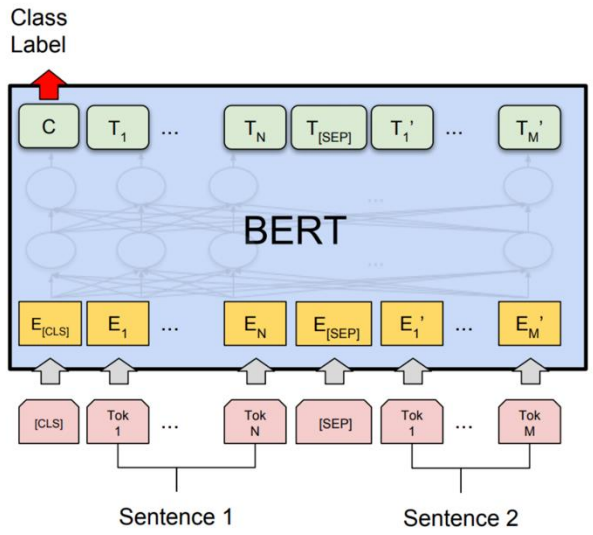
Large Language Models



https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/

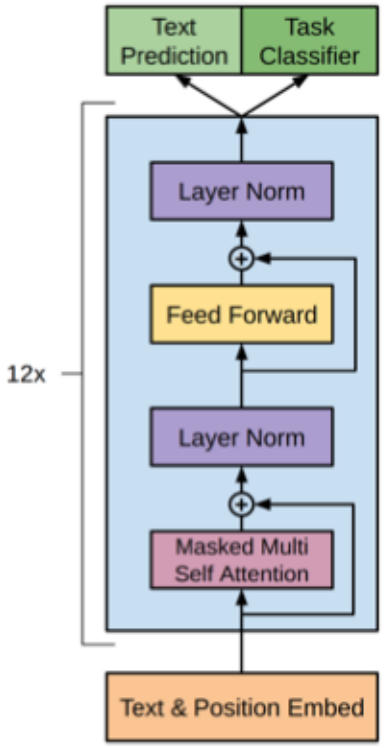
Transformer Models

Encoder



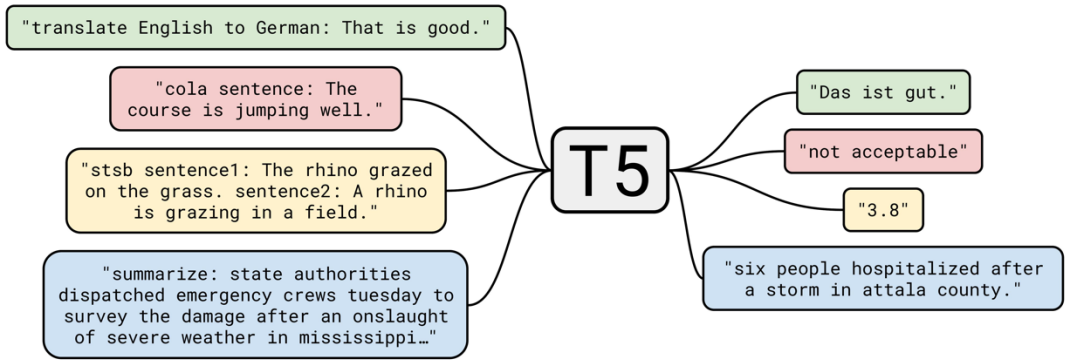
[Devlin et al., 2019]

Decoder



[Radford et al., 2018]

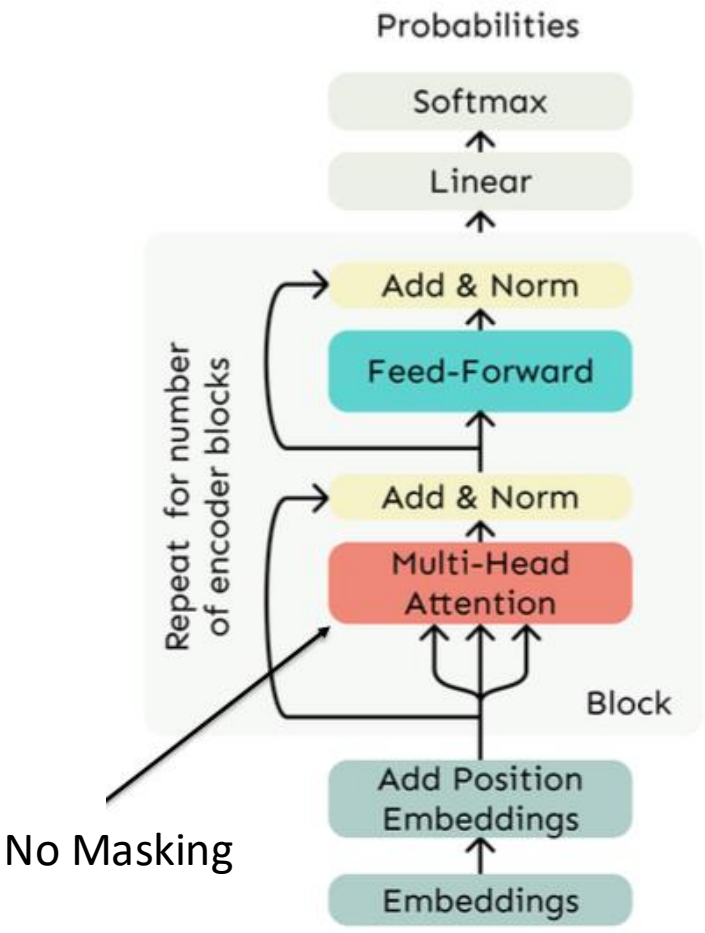
Encoder-Decoder



[Raffel et al., 2019]

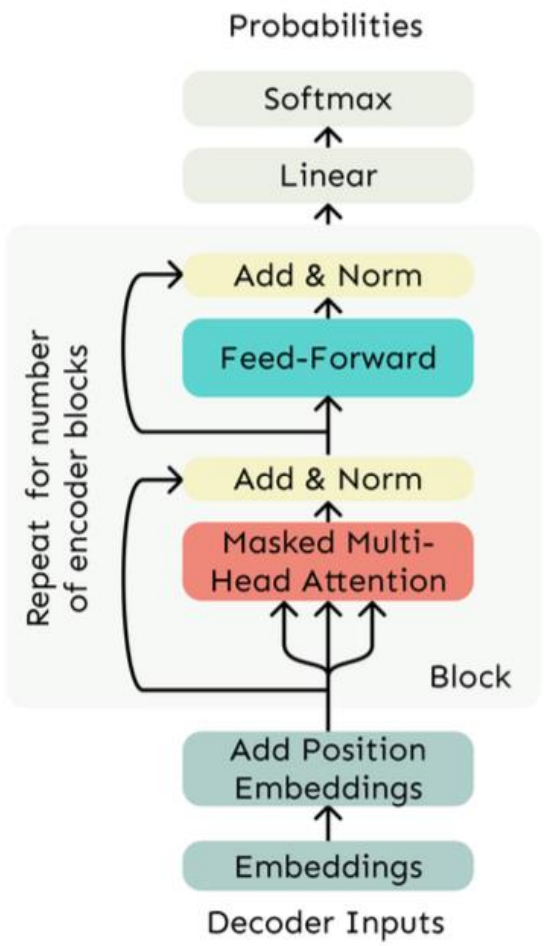
Transformer Architecture

Encoder

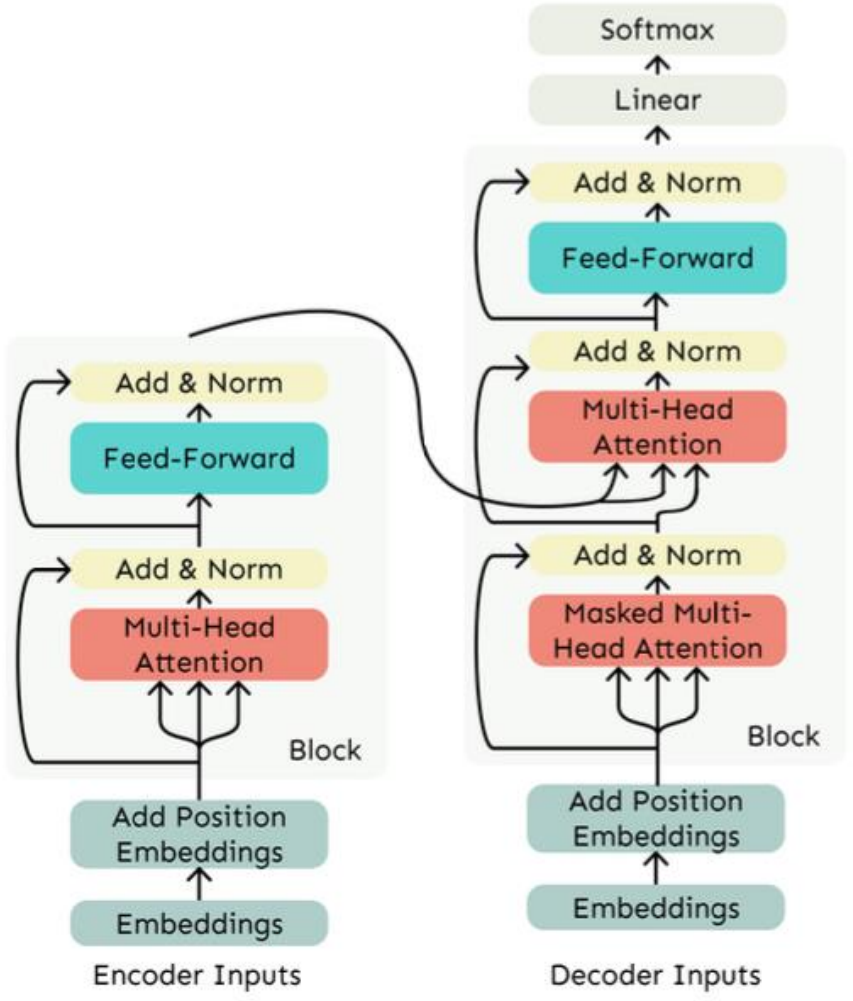


No Masking

Decoder



Encoder-Decoder



GPT Family

Improving Language Understanding by Generative Pre-Training

Alec Radford OpenAI alec@openai.com
Karthik Narasimhan OpenAI karthikn@openai.com
Tim Salimans OpenAI tim@openai.com
Ilya Sutskever OpenAI ilyasu@openai.com

GPT (2018), 117M parameters

Language Models are Unsupervised Multitask Learners

Alec Radford [*1] **Jeffrey Wu** [*1] **Rewon Child** [1] **David Luan** [1] **Dario Amodei** [**1] **Ilya Sutskever** [**1]

GPT-2 (2019), 1.5B parameters

Language Models are Few-Shot Learners

Tom B. Brown* **Benjamin Mann*** **Nick Ryder*** **Melanie Subbiah***
Jared Kaplan† **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam** **Girish Sastry**
Amanda Askell **Sandhini Agarwal** **Ariel Herbert-Voss** **Gretchen Krueger** **Tom Henighan**
Rewon Child **Aditya Ramesh** **Daniel M. Ziegler** **Jeffrey Wu** **Clemens Winter**
Christopher Hesse **Mark Chen** **Eric Sigler** **Mateusz Litwin** **Scott Gray**
Benjamin Chess **Jack Clark** **Christopher Berner**
Sam McCandlish **Alec Radford** **Ilya Sutskever** **Dario Amodei**
OpenAI

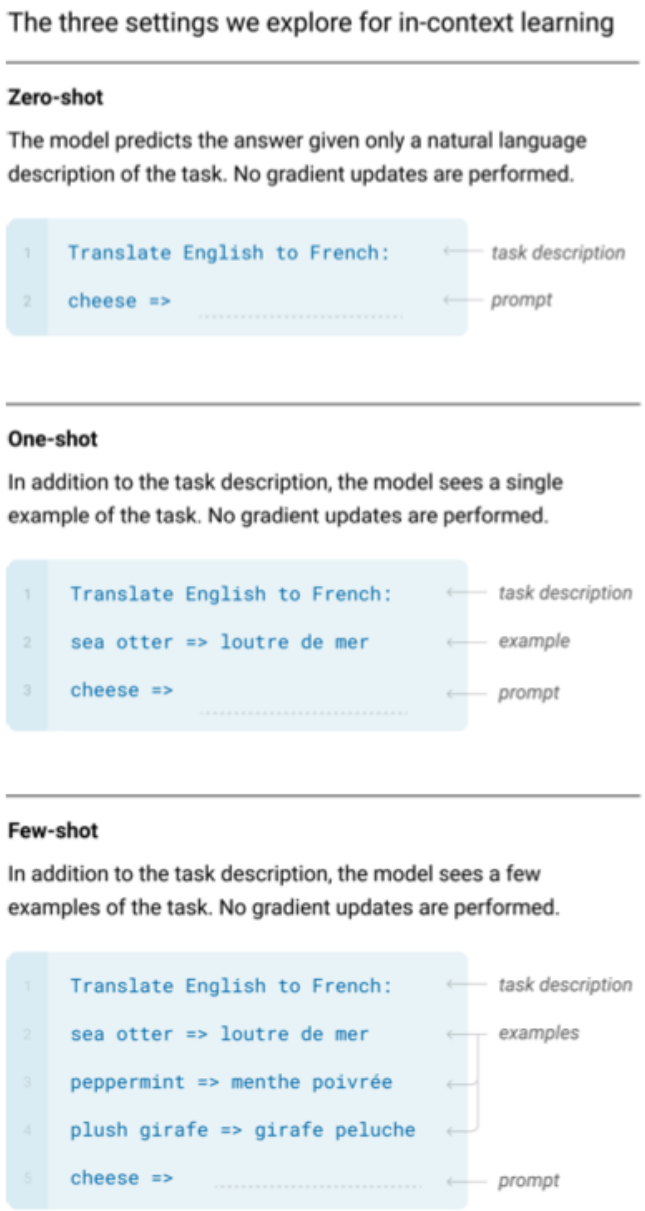
GPT-3 (2020), 175B parameters

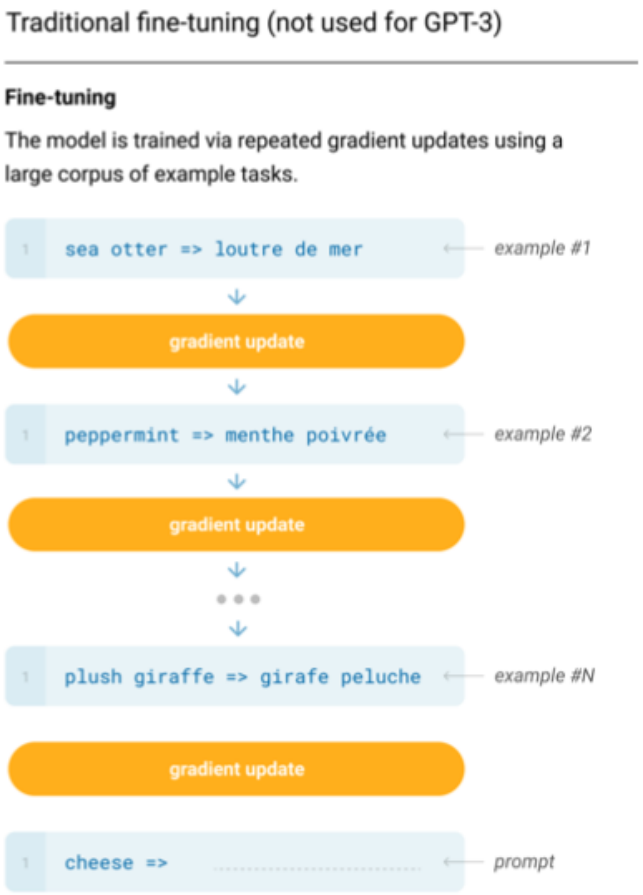
NeurIPS 2020 Best Paper

Not open-source!

GPT-3 Emergent Abilities

- Zero-shot/few-shot learning





[Brown et al., 2020]

GPT-3 Emergent Abilities

- Reasoning and generation

Physical reasoning

- You are having a small dinner party. You want to serve dinner in the living room. The dining room table is wider than the doorway, so to get it into the living room, you will have to **remove the door. You have a table saw, so you cut the door in half and remove the top half.**

[This is one confusion after another. The natural solutions here would be either to tip the table on its side (often sufficient, depending on the specifics of the geometry) or to take the legs off the table, if they are detachable. Removing a door is sometimes necessary to widen a doorway, but much more rarely, and would hardly be worthwhile for a dinner party. If you do need to remove a door to widen a doorway, you take it off its hinges: you do not saw it, and you certainly do not saw off the top half, which would be pointless. Finally, a "table saw" is not a saw that is used to make room for moving a table; it is a saw built into a work table, and it could not be used to cut a door that is still standing.]

Social reasoning

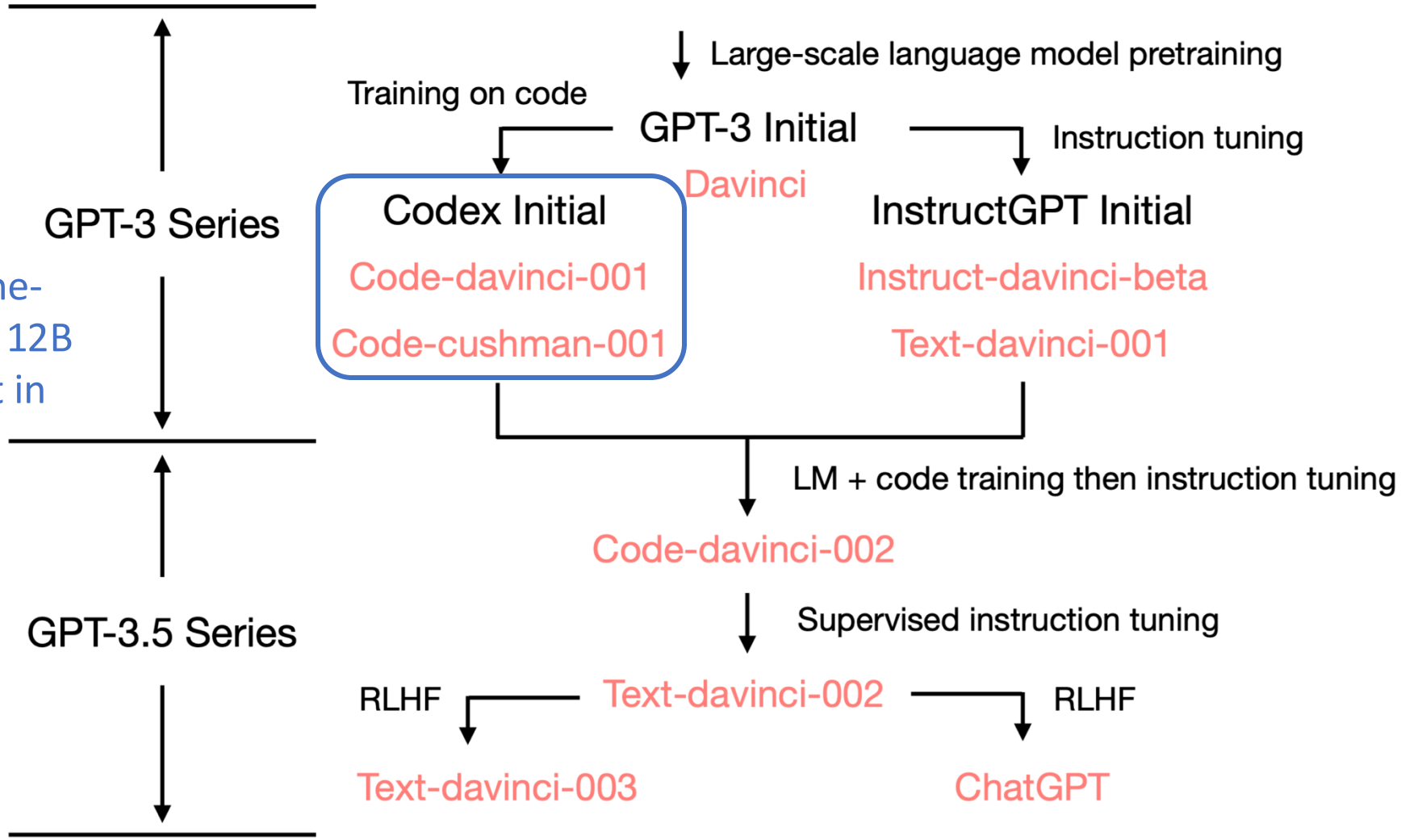
- You are a defense lawyer and you have to go to court today. Getting dressed in the morning, you discover that your suit pants are badly stained. However, your bathing suit is clean and very stylish. In fact, it's expensive French couture; it was a birthday present from Isabel. You decide that you should wear **the bathing suit to court. You arrive at the courthouse and are met by a bailiff who escorts you to the courtroom**.

[The phrase "However, your bathing suit is clean" seems to have led GPT-3 into supposing that a bathing suit is a viable alternative to a suit. Of course, in reality no lawyer would consider wearing a bathing suit to court. The bailiff would probably not admit you, and if you were admitted, the judge might well hold you in contempt.]

https://www.technologyreview.com/2020/08/22/1007539/gpt3-openai-language-generator-artificial-intelligence-ai-opinion/

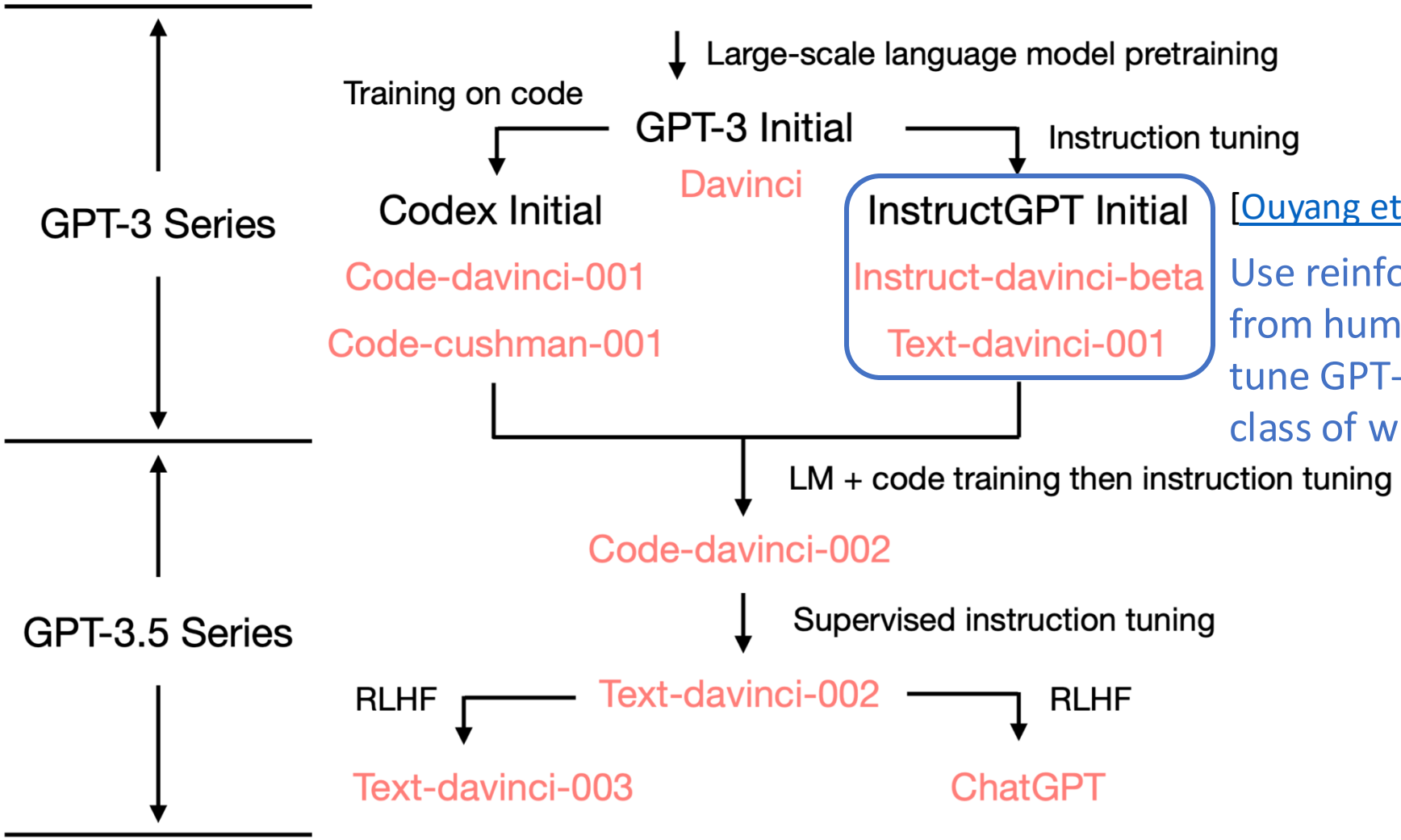
[Brown et al., 2020]

How Does OpenAI Arrive at ChatGPT?



[Fu et al., 2022]

How Does OpenAI Arrive at ChatGPT?

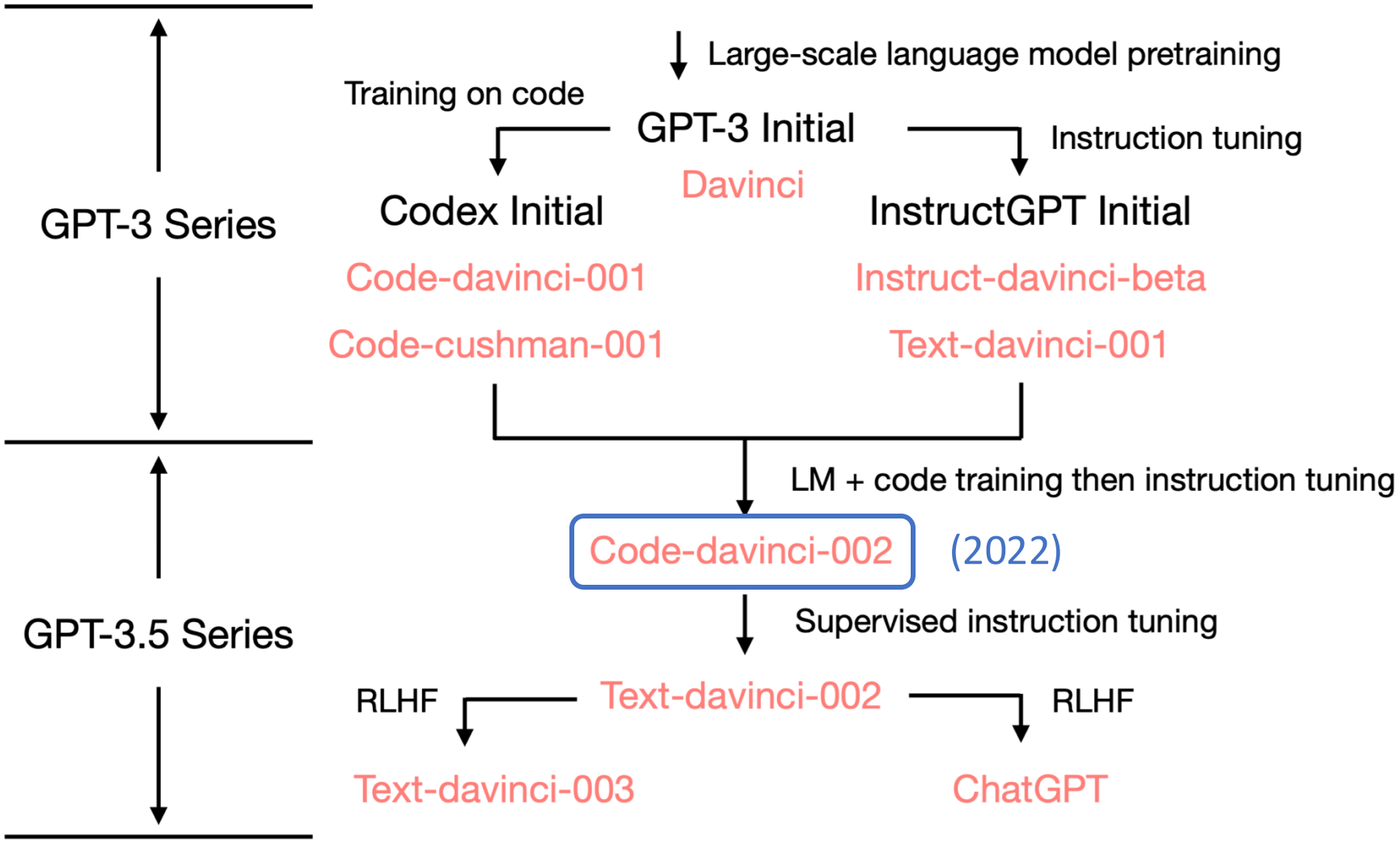


[Ouyang et al., 2022]

Use reinforcement learning from human feedback to fine-tune GPT-3 to follow a broad class of written instructions

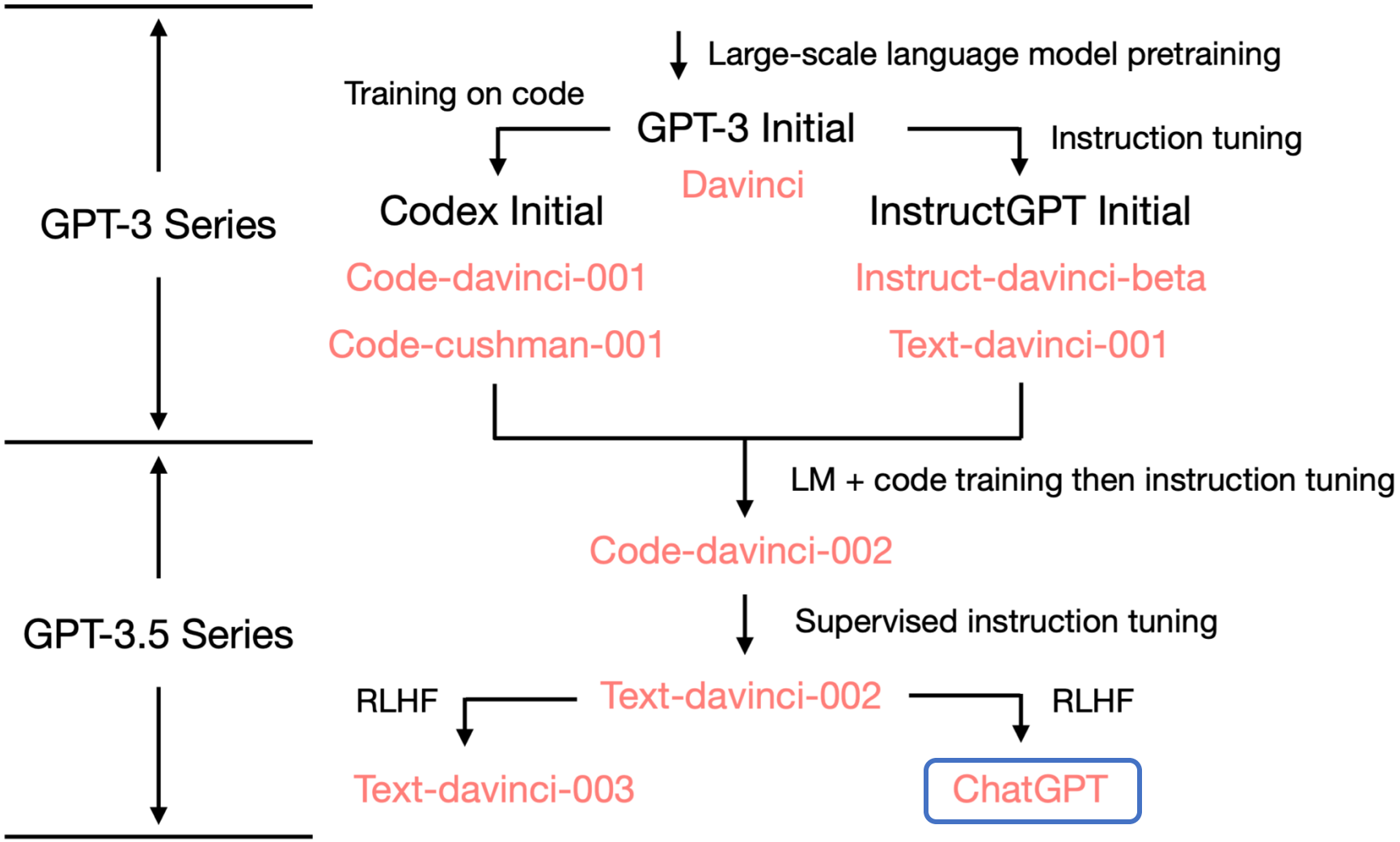
[Fu et al., 2022]

How Does OpenAI Arrive at ChatGPT?



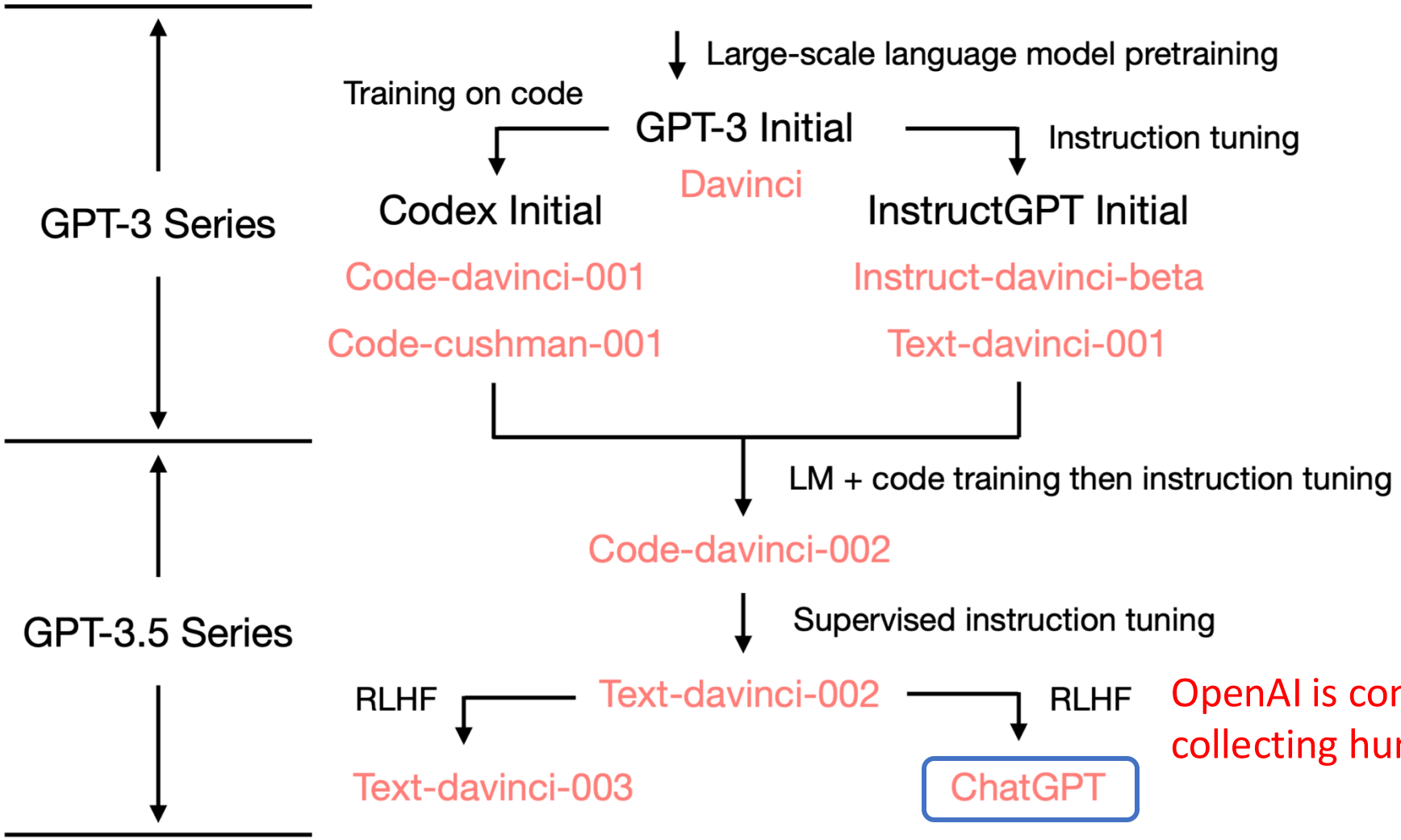
[Fu et al., 2022]

How Does OpenAI Arrive at ChatGPT?



[Fu et al., 2022]

How Does OpenAI Arrive at ChatGPT?



OpenAI is continuously collecting human preferences

[Fu et al., 2022]

Outline

- Introduction to NLP
- Text Classification
- Word Embeddings
- Natural Language Generation
- Modern Language Models

Thank you!