

人工智能语言模型在认知心理学中的应用

目录

1. 原理概述

- 代码平台
- 模型结构
- 语言任务



理论基础 & 实际操作

2. 特征计算

- 词汇特征
- 语义特征
- 句法特征



认知神经科学应用 & 实际操作

3. 其他案例

- 可控文本生成
- 群体认知

1. 原理概述

- 代码平台
- 模型结构
- 语言任务

2. 特征计算

- 词汇特征
- 语义特征
- 句法特征

3. 其他案例

- 可控文本生成
- 群体认知

1. 传统语言特征计算工具包: **hanlp**, spacy, nltk, CoreNLP
2. 语言模型搭建、加载、训练平台: **huggingface**

1. 传统语言特征计算工具包: **hanlp**, spacy, nltk, CoreNLP
2. 语言模型搭建、加载、训练平台: **huggingface**
3. 开源模型部署: **ollama**, vLLM, 闭源模型测试: Azure, **OpenRouter**

1. 传统语言特征计算工具包: hanlp, spacy, nltk, CoreNLP



HanLP

面向生产环境的前沿多语种自然语言处理技术

极速安装

使用指南

Hanlp: 中文友好, 提供大量开箱即用的语言特征计算接口



中文分词

将文本切分为独立语义单位。支持粗分、细分双重标准，高低优先级自定义词典，多语种



词性标注

给每个单词标注语法类别，支持多种词性标注集：CTB、PKU、863、UD、NPCMJ，多语种



命名实体识别

识别文本中的专有名词，支持多种规范：PKU、MSRA、OntoNotes，黑白名单词典、多语种



依存句法分析

分析单词语法上的依赖，支持多种句法体系：SD、UD、PMT，投射/非投射，多语种



成分句法分析

分析句子在语法上的递归构成，并将其表示为树形结构。支持可视化、多语种



语义依存分析

分析句子中单词之间的语义关系图。支持多标准：CSDP、DM、PAS、PSD，支持多语种



语义角色标注

分析句子的谓词论元结构。支持可视化、多语种



抽象意义表示

将句子的意义表示为以概念为节点的单源有向无环图的语言学框架，支持多语种

Hanlp: 中文友好, 提供大量开箱即用的语言特征计算接口

词性标注

请输入一段中文文本:

HanLP为生产环境带来次世代最先进的多语种NLP技术。
我的希望是希望张晚霞的背影被晚霞映红。

此页内容

- 简介
- 调用方法
- 创建客户端
- CTB词性标注集
- PKU词性标注集
- 863词性标注集
- 本地调用
- 多语种支持

粗分 对比1.x CTB PKU 863

词性标注

简介

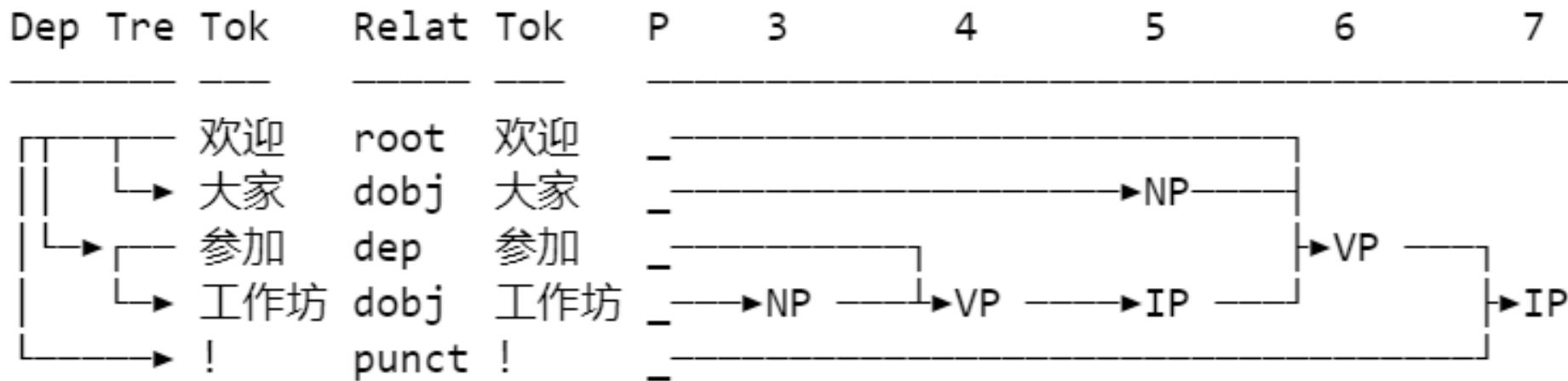
词性标注 (Part-of-Speech tagging、POS) 是一种标注句子中每个单词的词性 (也称词类、语法类别) 的任务。HanLP支持[CTB](#)、[PKU](#)、[863](#)、[NPCMJ](#)、[Universal Dependencies](#)等词性标注集。

调用方法

```
Hanlp = hanlp.load(hanlp.pretrained.mtl.CLOSE_TOK_POS_NER_SRL_DEP_SD1P_CON_ELECTRA_SMALL_ZH
doc = Hanlp('欢迎大家参加工作坊!', tasks=['dep', 'con'])
doc.pretty_print()
```

✓ 1.2s

Python



Hanlp等工具包使用简单, 但可自定义程度差:

1. 不支持完成复杂语言任务, 如推理、对话等
2. 不支持自定义模型结构
3. 不支持分析或提取模型内部参数

2. 语言模型搭建、加载、训练平台: **huggingface**



Hugging Face

(Transformers)

- 模型搭建
- 模型训练
- 模型推理
- ...

(Datasets)

- 数据集加载
- 多线程预处理
- 高效缓存
- ...

(Tokenizers)

- 快速分词
- 格式转换
- 自定义分词器
- ...

Huggingface提供大量的开源模型和数据集，以及中文文档

The screenshot shows the HuggingFace website interface. At the top, there is a navigation bar with the following items: a yellow smiley face icon, "Hugging Face", a search bar containing "Search models, datasets, users...", and a red-bordered menu bar with "Models", "Datasets", "Spaces", and "Docs". Below the menu bar, there are links for "Solutions", "Pricing", and "Log In/Sign Up".

The main content area features the "HuggingChat" interface. On the left, there is a sidebar with a list of chat sessions: "HuggingChat" (selected), "New Chat", "Untitled 8", "Untitled 5", "Untitled 18", "Dear Supplier, Requesting Regular Order for Products ...", "Request wine, eggs, bread every week.", and "Please supply us with wine x10, eggs x24 and bread ...".

In the center, there is a large "NEW" badge above the "HuggingChat" logo. Below the logo, the text reads: "Making the community's best AI chat models available to everyone." A "Start chatting" button is located at the bottom of this section.

On the right, there is a text input field with placeholder text: "As a restaurant owner, write a professional email to the supplier to get these". Below the placeholder, there is a list of items: "Wine (x10)", "Eggs (x24)", and "Bread (x12)". Further down, there is a message template starting with "Dear Supplier," followed by a body text about the restaurant's satisfaction with the supplier's products and a request for delivery of the specified items.

facebook/bart-large-cnn like 406

Summarization PyTorch TensorFlow JAX Rust Transformers cnn_dailymail English bart text2text-generation Eval Results

AutoTrain Compatible arxiv:1910.13461 License: mit

Model card Files and versions Community 31 Edit model card

Train Deploy Use in Transformers

模型描述

BART (large-sized model), fine-tuned on CNN Daily Mail

BART model pre-trained on English language, and fine-tuned on [CNN Daily Mail](#). It was introduced in the paper [BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension](#) by Lewis et al. and first released in [this repository (<https://github.com/pytorch/fairseq/tree/master/examples/bart>)].

Disclaimer: The team releasing BART did not write a model card for this model so this model card has been written by the Hugging Face team.

Model description

BART is a transformer encoder-encoder (seq2seq) model with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder. BART is pre-trained by (1) corrupting text with an arbitrary noising function, and (2) learning a model to reconstruct the original text.

Downloads last month 1,345,544

在线演示

Hosted inference API

Summarization Examples

The tower is 324 metres (1,063 ft) tall, about the same height as an 81-storey building, and the tallest structure in Paris. Its base is square, measuring 125 metres (410 ft) on each side. During its construction, the Eiffel Tower surpassed the Washington Monument to become the tallest man-made structure in the world, a title it held for 41 years until the Chrysler Building in New York City was finished in 1930. It was the first structure to reach a height of 300 metres. Due to the addition of a broadcasting aerial at the top of the tower in 1957, it is now taller than the Chrysler Building by 5.2 metres (17 ft). Excluding transmitters, the Eiffel Tower is the second tallest free-standing structure in France after the Millau Viaduct.

Compute

```
# 从huggingface平台上找到对应的模型路径
model_path = 'hw2942/bert-base-chinese-finetuning-financial-news-sentiment-v2'

# 使用transformers工具包加载模型
tokenizer = AutoTokenizer.from_pretrained(model_path)
model = AutoModelForSequenceClassification.from_pretrained(model_path)

# 利用pipeline快速进行语言任务
text = '今年我国经济增长速度为3%'
text_classification = pipeline('sentiment-analysis', model=model, tokenizer=tokenizer)
res = text_classification(text)[0]
print(f"\nInput: {text}\nPrediction: {res['label']}, Score: {res['score']:.3f}")
```

✓ 1.1s

Python

Input: 今年我国经济增长速度为3%

Prediction: Positive, Score: 1.000

3. 开源模型部署: **ollama**, vLLM



Get up and running with large language models.

Run [Llama 3.1](#), [Phi 3](#), [Mistral](#), [Gemma 2](#), and other models. Customize and create your own.

Download ↓

Available for macOS, Linux, and Windows (preview)

交互式对话模型快速部署，需要硬件支持。

3. 开源模型部署: **ollama**, vLLM



Get up and running with large language models.

Run [Llama 3.1](#), [Phi 3](#), [Mistral](#), [Gemma 2](#), and other models. Customize and create your own.

Download ↓

Available for macOS, Linux, and Windows (preview)

qwen2

Qwen2 is a new series of large language models from Alibaba group

0.5B 1.5B 7B 72B

2.2M Pulls Updated 2 months ago

0.5b 97 Tags

Updated 2 months ago

6f48b936a09f · 352MB

model arch qwen2 · parameters 494M · quantization Q4_0 352MB

params {"stop":["<|im_start|>","<|im_end|>"]} 59B

template {{ if .System }}<|im_start|>system {{ .System }}<|im_end|> {{ en... 182B

license Apache License Version 2.0, January 2004 http://www.apache.org/l... 11kB

执行代码

ollama run qwen2:0.5b

交互式对话模型快速部署，需要硬件支持。

3. 开源模型部署: **ollama**, vLLM

```
PS D:\code\workshop> ollama list
NAME          ID          SIZE    MODIFIED
● qwen2:0.5b  6f48b936a09f  352 MB  4 days ago
PS D:\code\workshop> ollama run qwen2:0.5b
○ >>> 今天我吃了烤鸭，你怎么看
```

>>> 烤鸭是中华民族的美食，具有悠久的历史和丰富的文化内涵。今天吃烤鸭，你可能会感到以下几点感受：

1. **美味佳肴**：烤鸭皮脆肉嫩，口感独特，香而不腻，口感丰富，让人垂涎欲滴。
2. **历史文化**：烤鸭历史悠久，承载着丰富的历史和文化信息。
3. **地域特色**：烤鸭是北京的传统美食，具有浓郁的地方特色。
4. **营养价值**：烤鸭含有丰富的蛋白质和微量元素，对人体有益。
5. **健康理念**：虽然烤鸭作为一种食物，但它含有一定的热量和油脂，吃多了可能会带来一些不适。因此，在选择时要注意搭配，不要过量食用。

最后，请注意在享用之前，尽量先洗手消毒，以减少病毒或细菌的传播风险。祝你品尝到美味的烤鸭！

3. 闭源模型测试: Azure, **OpenRouter**

A unified interface for LLMs

Find the [best models & prices](#) for your prompts

[Chat](#) [Browse](#)

~ TRENDING MODELS: ● All Categories ▾ ~

OLMo 7B Instruct
OLMo 7B Instruct by the Allen Institute for AI is a model finetuned for question answering. It demonstrates **notable performance** across multipl...
by [allenai](#) 2173%

GPT-4 32k
GPT-4-32k is an extended version of GPT-4, with the same capabilities but quadrupled context length, allowing for processing up to 40 pages of text in a...
by [openai](#) 109%

Qwen 1.5 4B Chat
Qwen1.5 4B is the beta version of Qwen2, a transformer-based decoder-only language model pretrained on a large amount of data. In compariso...
by [qwen](#) 82%

能够进行批量测试，对硬件无要求。

3. 闭源模型测试: Azure, OpenRouter

A unified interface for LLMs

Find the [best models & prices](#) for your prompts

[Chat](#) [Browse](#)

~ TRENDING MODELS: ● All Categories ▾ ~

OLMo 7B Instruct OLMo 7B Instruct by the Allen Institute for AI is a model finetuned for question answering. It demonstrates notable performance across multiple... by allenai ↑2173%	GPT-4 32k GPT-4 32k is an extended version of GPT-4, with the same capabilities but quadrupled context length, allowing for processing up to 40 pages of text in a... by openai ↑109%	Qwen 1.5 4B Chat Qwen1.5 4B is the beta version of Qwen2, a transformer-based decoder-only language model pretrained on a large amount of data. In compariso... by qwen ↑82%
---	---	--

[OpenAI: GPT-4 32k \(older v0314\)](#) [Chat](#)

openai/gpt-4-32k-0314 [🔗](#)
Updated Aug 28 | 32,767 context | \$60/M input tokens | \$120/M output tokens

GPT-4-32k is an extended version of GPT-4, with the same capabilities but quadrupled context length, allowing for processing up to 40 pages of text in a single pass. This is particularly...

[Overview](#) [Providers](#) [Apps](#) [Activity](#) [Parameters](#) [Uptime](#) [API](#)

Providers for GPT-4 32k (older v0314)
OpenRouter [load-balances requests](#) across providers weighted by price unless you use [dynamic routing](#).

Provider	Max Output	Input	Output	Latency	Throughput
OpenAI	4,096	\$60	\$120	0.93s	30.09t/s

能够进行批量测试，对硬件无要求。

3. 闭源模型测试: Azure, **OpenRouter**

```
OPENROUTER_API_KEY = "YOUR_API_KEY"
def send_message(sentence, ):
    client = OpenAI(base_url="https://openrouter.ai/api/v1",
                    api_key=OPENROUTER_API_KEY,)

    resp = client.chat.completions.create(
        extra_headers={},
        model="openai/chatgpt-4o-latest",
        messages=[{"role": "user", "content": sentence,}],
        temperature=0.7,
        top_p=0.5,
    )
    resp_text = resp.choices[0].message.content
    return resp, resp_text

resp, resp_text = send_message("今天我吃了烤鸭，你怎么看")
resp_text
```

✓ 5.3s

>>> 听起来你享受了一顿美味的烤鸭！烤鸭是一道非常经典的菜肴，尤其是在中国的北京烤鸭更是闻名遐迩。它外皮酥脆，肉质鲜嫩，配上甜面酱、葱丝和薄饼，真的是一种绝妙的味觉体验。希望你吃得开心！你是自己做的还是在餐馆里享用的呢？

1. 传统语言特征计算工具包: **hanlp**, spacy, nltk, CoreNLP
 - 适用于语言特征计算, 集成度高
2. 语言模型搭建、加载、训练平台: **huggingface**
 - 适用于自定义语言模型, 自由度高
3. 开源模型部署: **ollama**, vLLM, 闭源模型测试: Azure, **OpenRouter**
 - 适用于对话形式的大语言模型, 上手门槛低

1. 原理概述

- 代码平台
- 模型结构
- 语言任务

2. 特征计算

- 词汇特征
- 语义特征
- 句法特征

3. 其他案例

- 可控文本生成
- 群体认知

语言模型的定义：语言模型是预测**句子出现概率**的模型

通俗解释：语言模型可以判断一句话有多么常见

$$P(\text{"我喜欢吃巧克力"}) > P(\text{"我喜欢吃桌子"})$$

$$P(\text{"猫追老鼠"}) > P(\text{"猫老鼠追"})$$

语言模型的定义：语言模型是预测**句子出现概率**的模型

通俗解释：语言模型可以判断一句话有多么常见

语言模型的目标

- 计算一个句子的出现概率：

$$P(S) = P(w_1, w_2, \dots, w_n) = P(w_1) * P(w_2|w_1) * \dots * P(w_n|w_1, w_2, \dots)$$

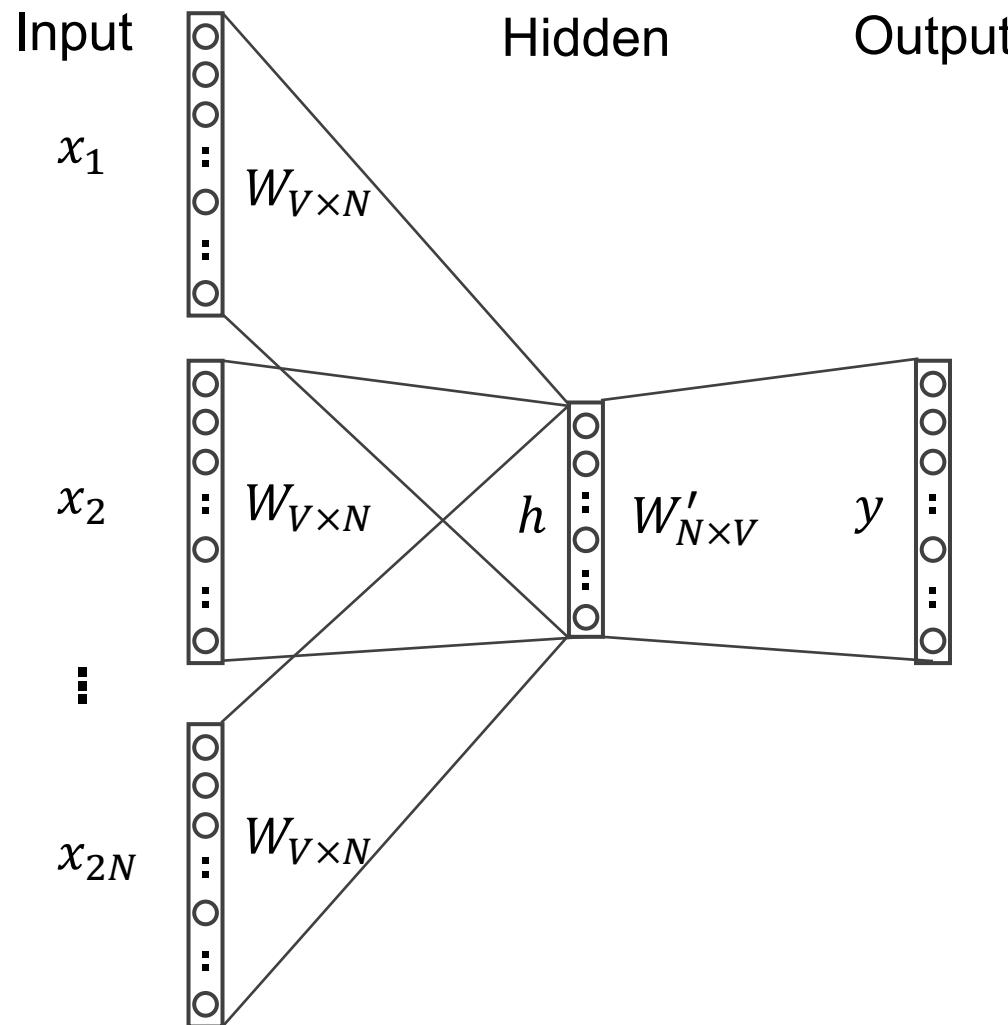
- $P(w_n|w_1, w_2, \dots)$ 给定上文之后，一个词汇出现的概率

- 语言模型的实际功能：**根据上文或上下文预测下一个词**

基于神经网络的语言模型:

1. 词袋模型: **CBOW**, skip-gram ...
2. 循环神经网络模型: **RNN-LM**, LSTM-LM ...
3. Transformer模型: **GPT**, BERT...

Continuous Bag Of Word (CBOW)



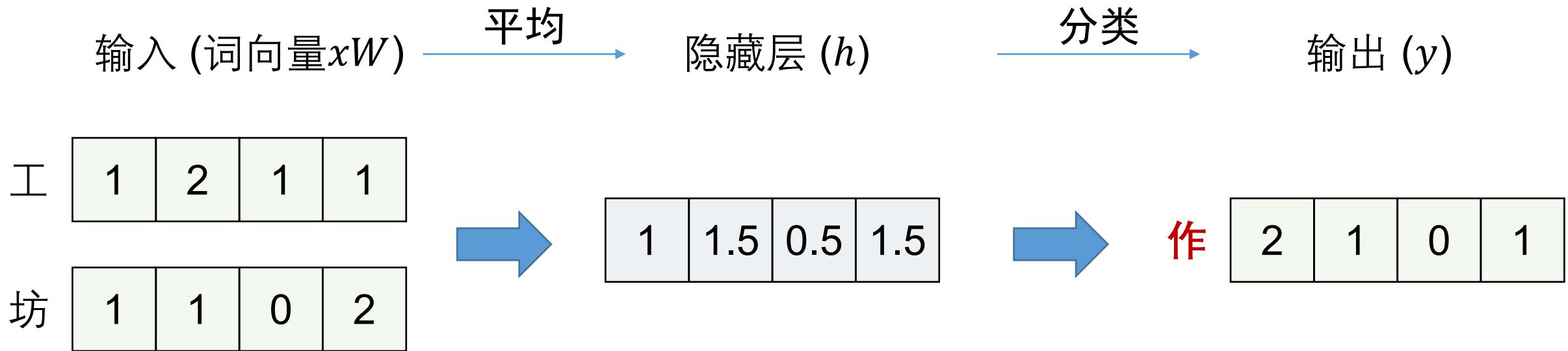
Task: 用前N个和后N个邻近词预测当前词 ($N < 5$)

Input: $2N$ 个邻近词 x_1, \dots, x_{2N}

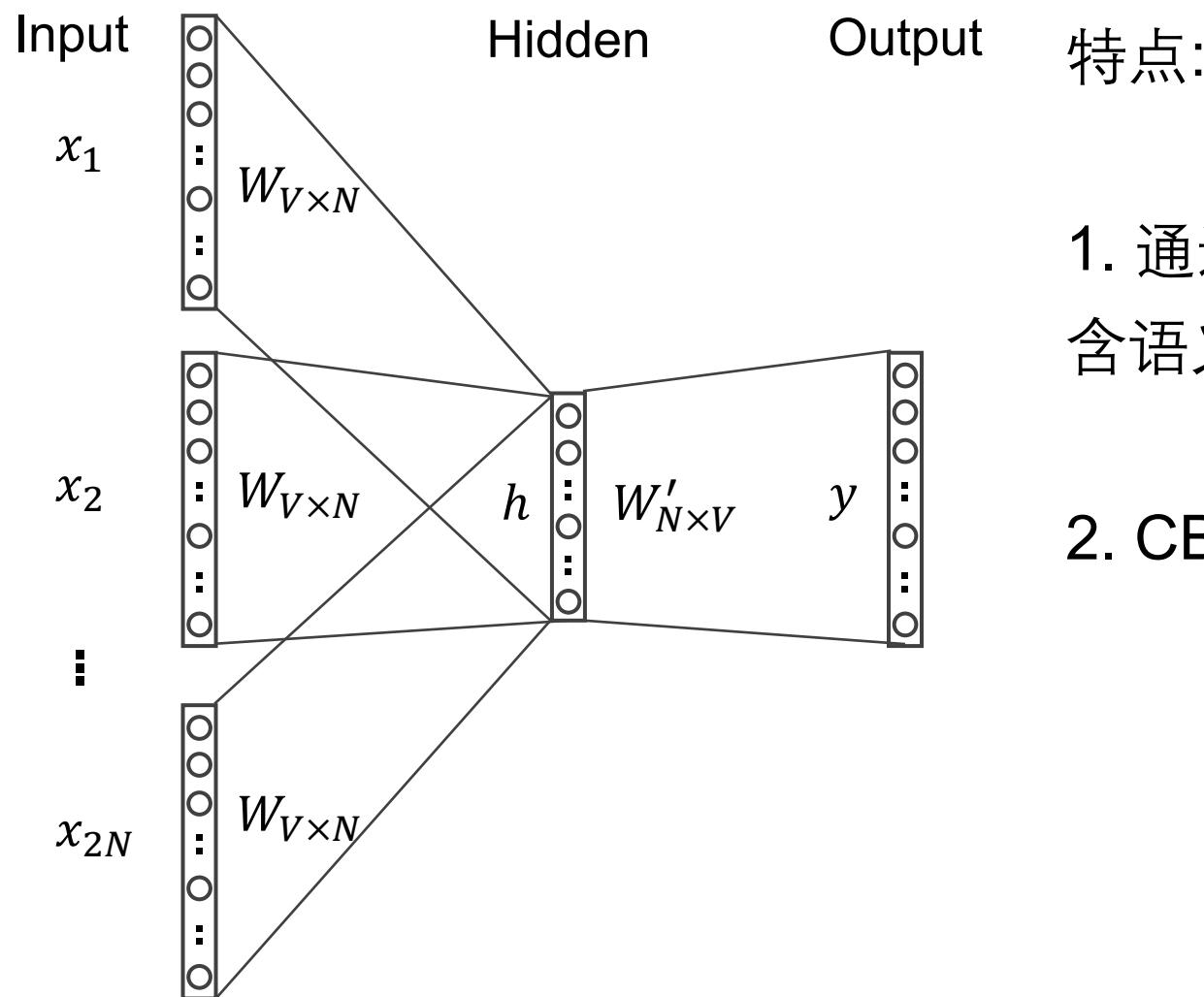
Output: 当前词 y

Continuous Bag Of Word (CBOW)

语料: 欢迎参加工作坊



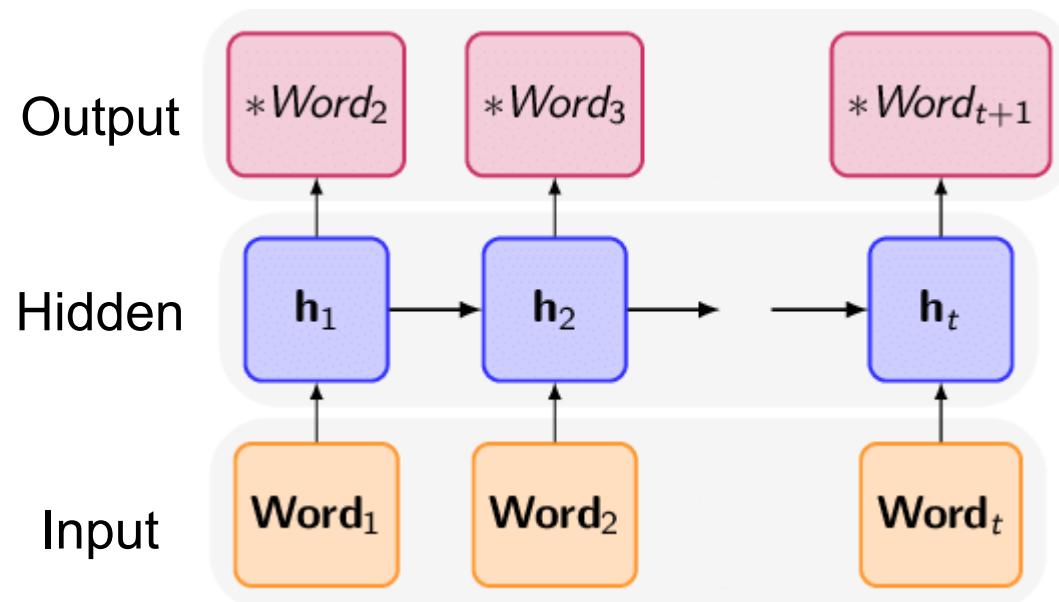
Continuous Bag Of Word (CBOW)



特点：

1. 通过映射矩阵 W , CBOW可以将单词映射为包含语义信息的词向量 (word2vec)
2. CBOW不考虑输入单词的顺序信息。

Recurrent Neural Network (RNN)



Task: 使用历史信息和当前词来预测下一个词

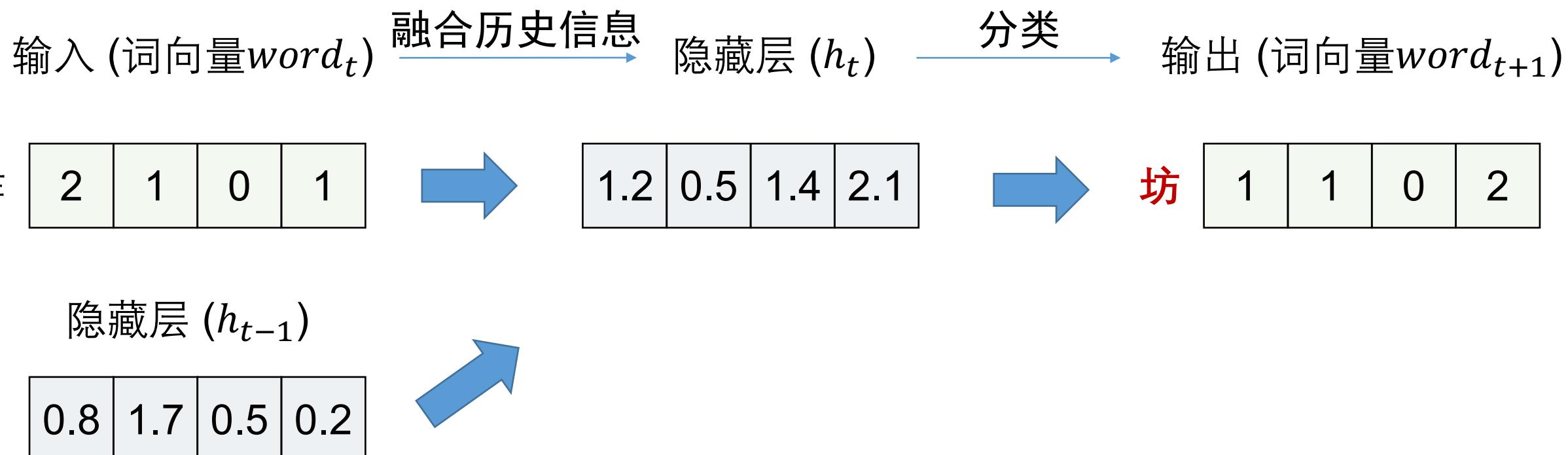
Input: 第t个单词 $word_t$, t-1时刻的历史信息 h_{t-1}

Output: 第t+1个单词 $word_{t+1}$

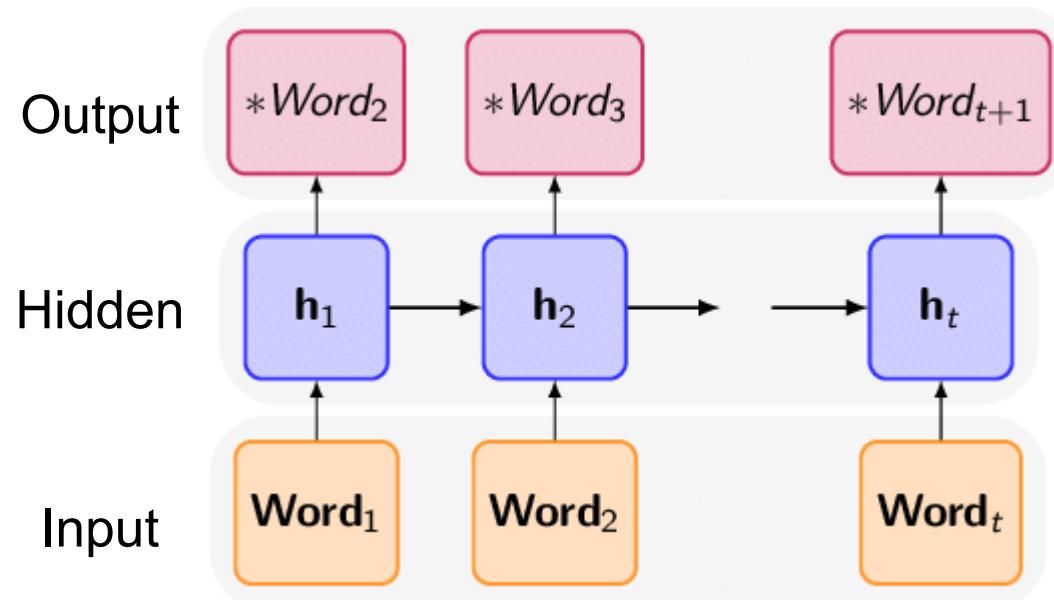
h_t 可连接任务分类器, 完成其他任务 (词性标注、情感分析等)

Recurrent Neural Network (RNN)

语料: 欢迎参加工作坊



Recurrent Neural Network (RNN)

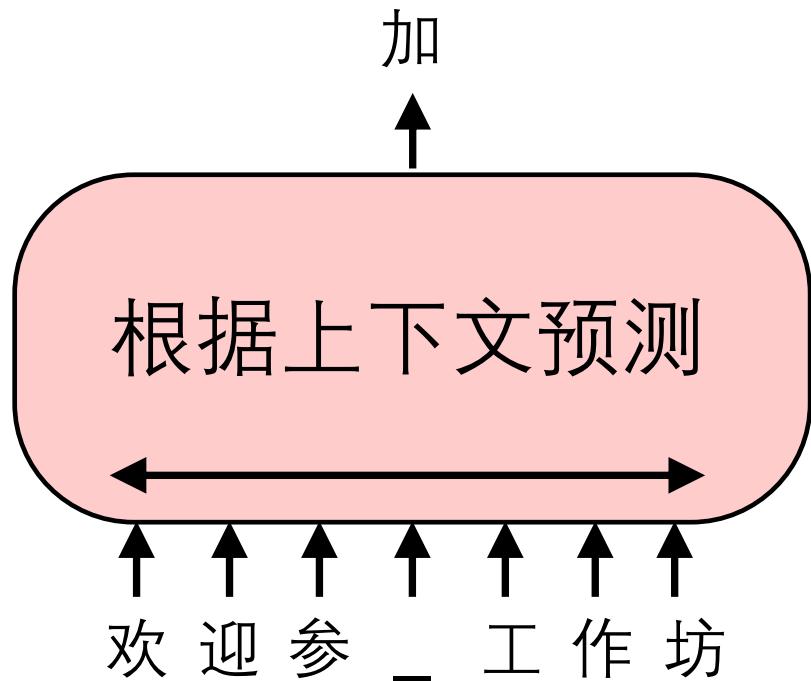


特点:

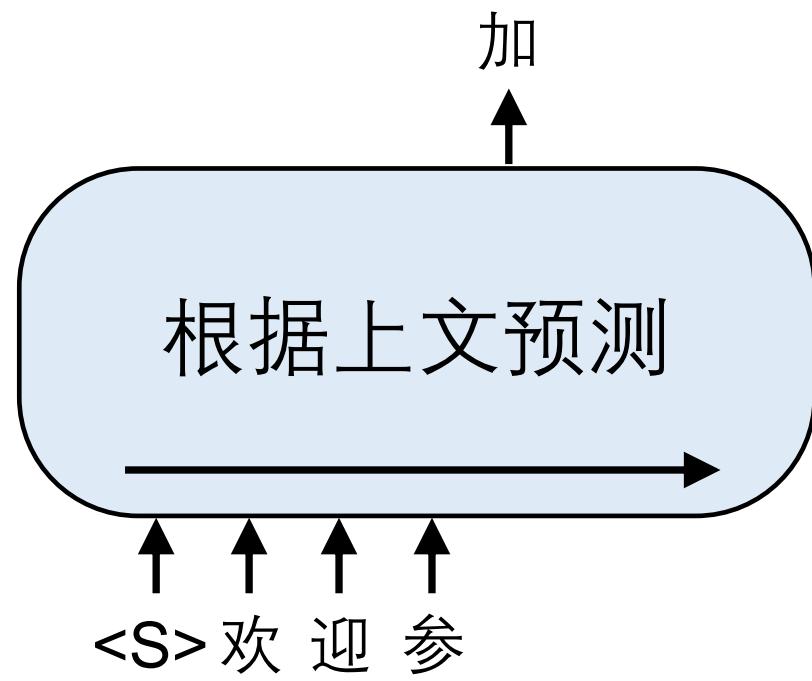
1. 模型拥有短期记忆能力, 能够记住前N个单词的历史信息 (由于梯度消失等问题, N一般小于10)。
2. 模型只能串行计算, 训练速度慢。

模型分类

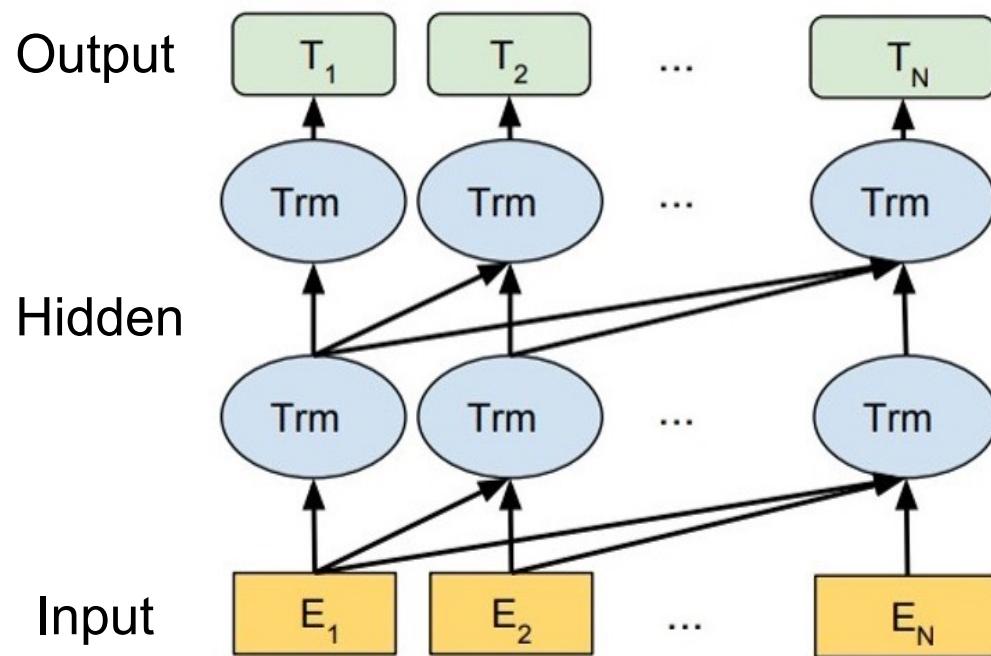
上下文(非增量式)模型: BERT等



上文(增量式)模型: GPT等



Generative Pre-trained Transformer (GPT)



Task: 利用所有前文预测下一个词

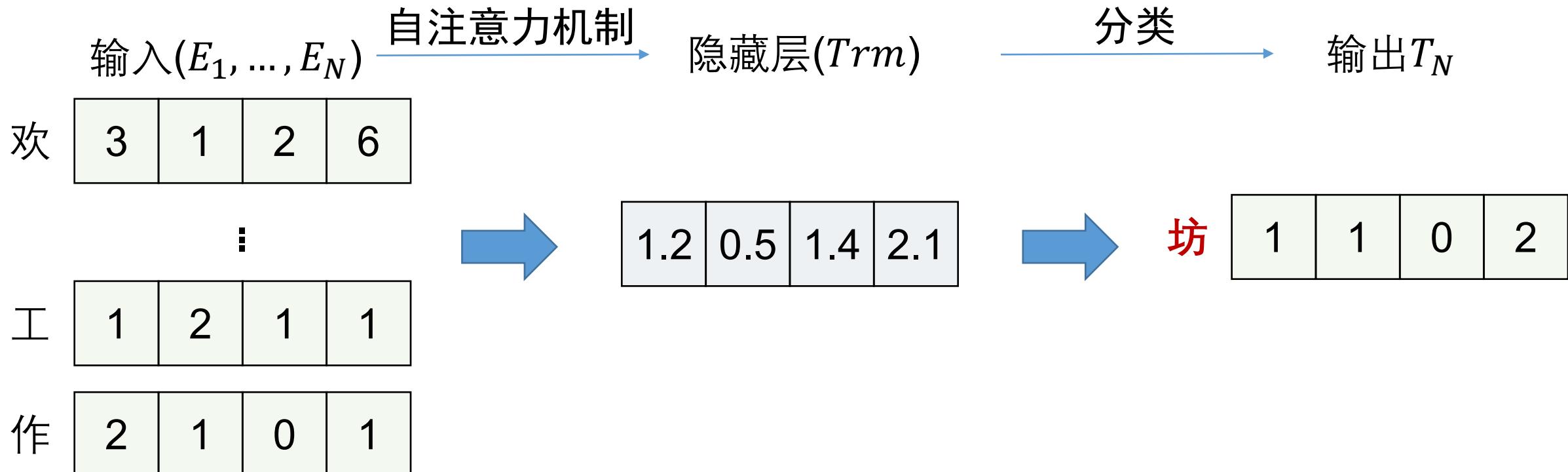
Input: 第 $1, \dots, N$ 个单词 E_1, \dots, E_N

Output: 第 $N+1$ 个单词 T_N

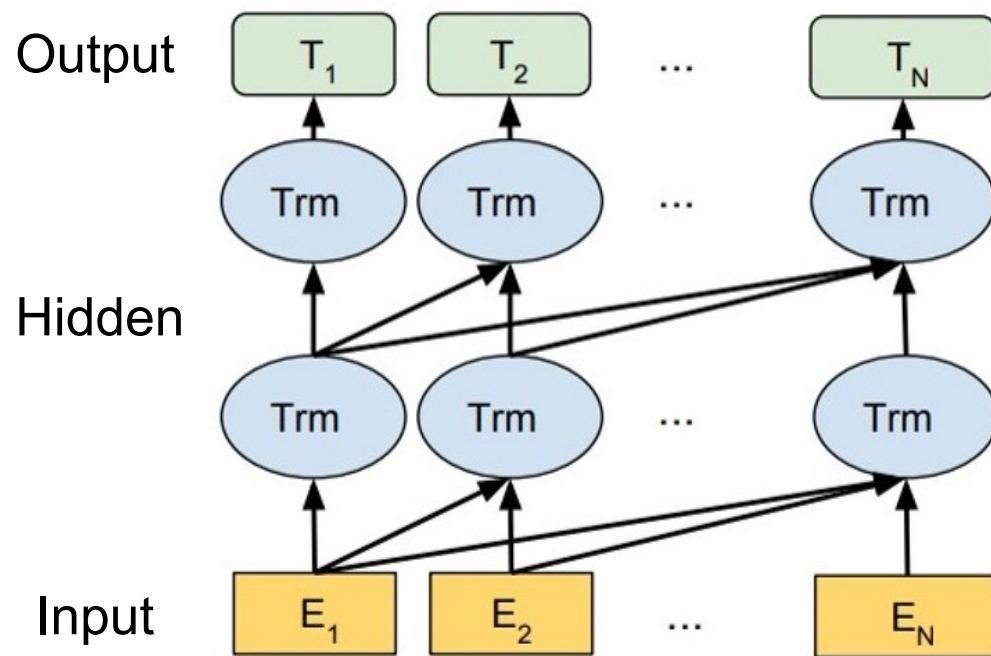
GPT的隐藏层可连接任务分类器，完成其他任务（词性标注、情感分析等）

Generative Pre-trained Transformer (GPT)

语料: 欢迎参加工作坊



Generative Pre-trained Transformer (GPT)



特点：

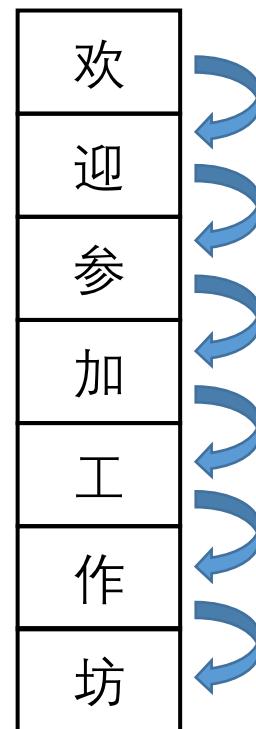
1. 可以同时考虑所有前文对当前词的影响，实现长距离依赖（最长可达15K）。
2. 可以并行计算，训练速度快。

趋势：更高效的融合上下文信息

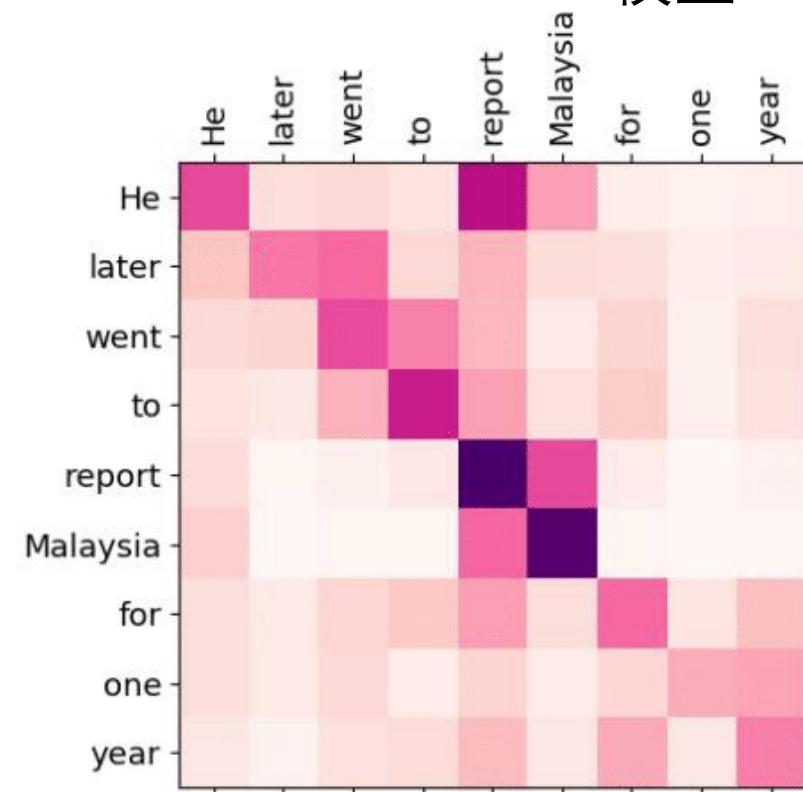
词袋模型



循环神经网络模型



Transformer模型



1. 原理概述

- 代码平台
- 模型结构
- 语言任务

2. 特征计算

- 词汇特征
- 语义特征
- 句法特征

3. 其他案例

- 可控文本生成
- 群体认知

其他语言任务可粗略分为两类：**文本分类** 和 **文本生成**

文本序列 -> 类别标签

- 自然语言推理
- 多项选择
- **词性标注**
- 主题分析
- 指代消解
- 意图识别
- 关系抽取
- **句法解析**

...

文本序列 -> 文本序列

- 机器翻译
- 对话系统
- 风格迁移
- 条件续写
- **可控文本生成**
- 自动文摘
- 语法纠正
- 开放域问答

...

如何使用语言模型完成其他语言任务？

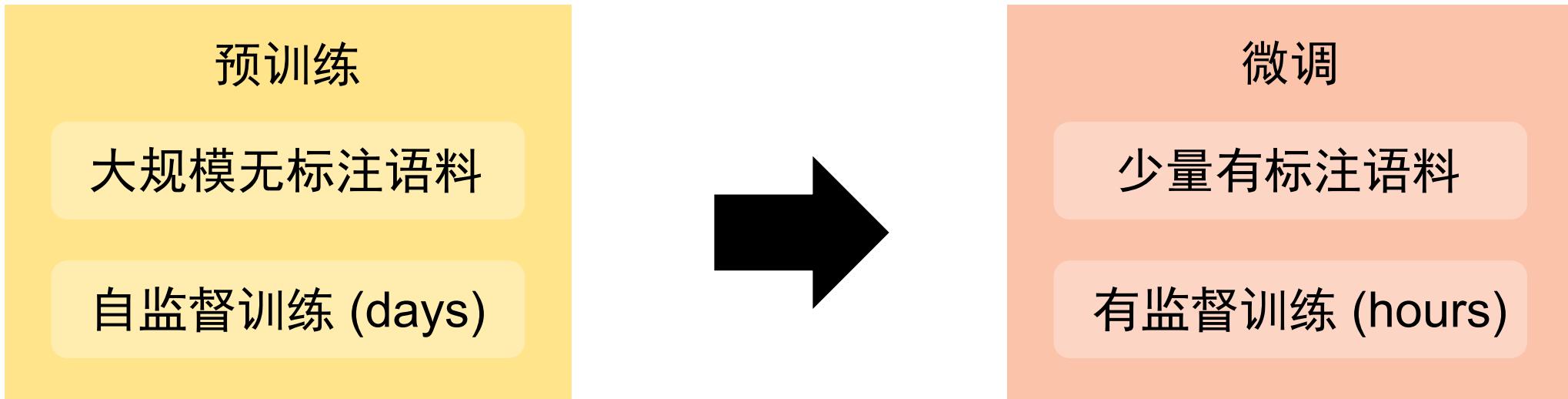
预训练

大规模无标注语料

自监督训练 (days)

训练任务: 预测下一个词

训练语料: 维基百科、人民日报等



训练任务: 预测下一个词

训练语料: 维基百科、人民日报等

训练任务: 根据输入预测标签

训练语料: 推理数据集OCNLI等

上下文学习 (in context learning)

给定任务描述和示例, chatGPT等模型能在不修改参数的前提下从上下文中学习任务。

1 将中文翻译为英文

2 吃晚饭 => eat dinner

3 工作坊 =>

1 根据关键词生成文本

2 经济 => 全球GDP增长率为2.7%

3 娱乐 =>

← task description

← example

← prompt

思维链 (Chain of Thought)

在示例中提供推理过程有助于模型正确回答问题。

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓

单词删除任务 (Word Deletion)

John developed a very special way of speaking.

For 'He withdrew the remark', he would say 'He withdrew'.

a single demonstration

*Please carefully examine John's speaking style, and guess what
he would say for the sentence: 'We put some orders together'*

a single test

根据单个示例学习语言任务

John developed a very special way of speaking.

For 'He withdrew the remark', he would say 'He withdrew'.

Please carefully examine John's speaking style, and guess what he would say for the sentence: 'We put some orders together'

delete last 2 words

Participant 1:

“We put some”.

orders together
DELETED

keep first 2 words

Participant 2:

“We put”.

some orders together
DELETED

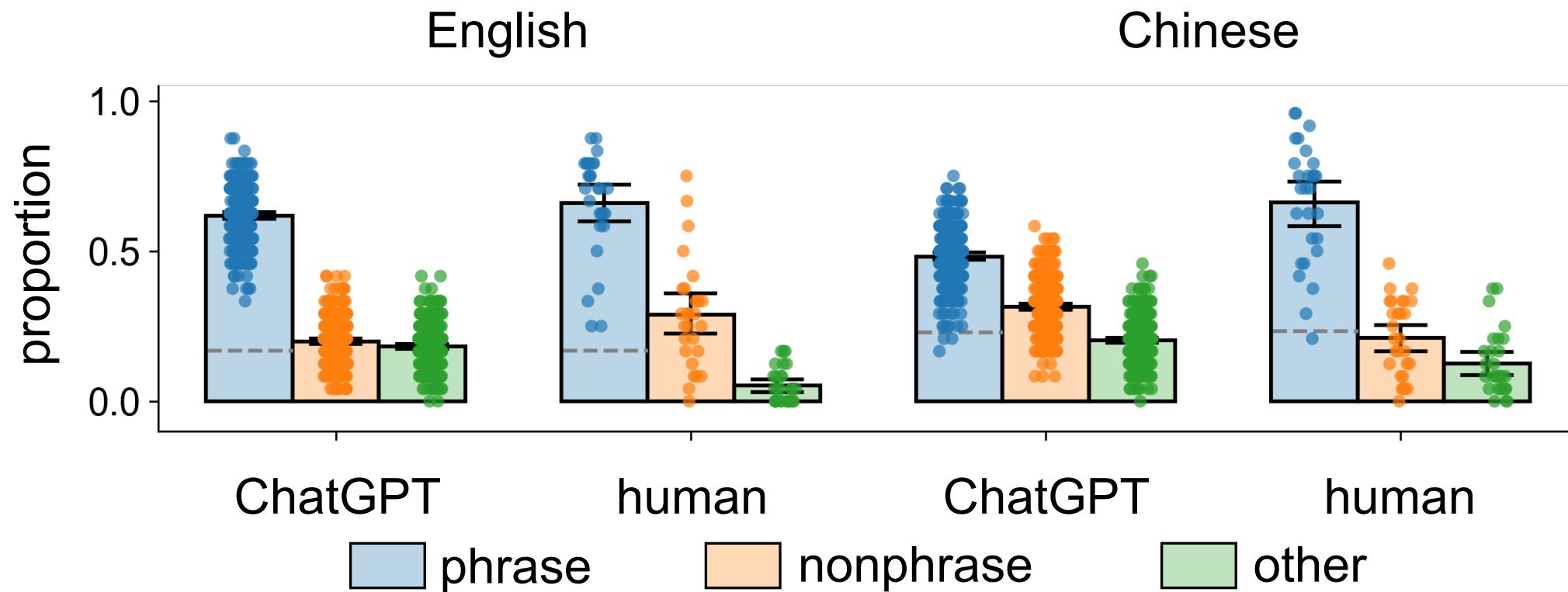
delete a phrase

Participant 3:

“We put together”.

some orders
DELETED

人和模型倾向于删除完整短语



语言模型和人均使用层级句法结构表征学习新的语言任务

小结

1. 语言模型的基本目标: 预测下一个词
 - 不同结构的语言模型, 主要区别在于如何融合上下文。
2. 通过预训练+微调、上下文学习等范式, 语言模型能够学会其他语言任务
 - 预训练+微调: 常见语言任务 / 有足够标注数据
 - 上下文学习: 缺少标注数据 / 有多任务设置需求

1. 概述

- 代码平台
- 模型结构
- 语言任务

2. 特征计算

- 词汇特征
- 语义特征
- 句法特征

3. 其他案例

- 可控文本生成
- 群体认知

为什么大家需要掌握特征的计算方法？

为什么大家需要掌握特征的计算方法？

Comment from a reviewer:

The paper compares the DNN-derived features to word-related features such as frequency and surprisal.

Word frequency is estimated from the BNC corpus. **The BNC corpus contains 100 Million words, whereas DNN was trained on several Billions of words.**

Surprisal is derived from the same corpus and derived using a trigram model. **Trigram models are now far surpassed by DNN-based language models.**

This means that the predictive power of the word-related features is likely to be **underestimated**. The authors could draw on (1) **frequency estimated from the corpora used for DNN (BookCorpus + Wikipedia)**, (2) **estimate surprisal from a strong off-the-shelf model such as GPT-2**.

为什么大家需要掌握特征的计算方法？

Comment from a reviewer:

The paper compares the DNN-derived features to word-related features such as frequency and surprisal.

Word frequency is
Million words, wh

contains 100
ls.

Surprisal is derived
models are now f

标准化、自主化、先进化

model. Trigram

This means that the predictive power of the word-related features is likely to be **underestimated**. The authors could draw on (1) **frequency estimated from the corpora used for DNN (BookCorpus + Wikipedia)**, (2) **estimate surprisal from a strong off-the-shelf model such as GPT-2**.

词汇特征

1. 概述

- 代码平台 a 词频
- 模型结构 b 转移概率
- 语言任务 c 词性

2. 特征计算

- 词汇特征
- 语义特征
- 句法特征

3. 其他案例

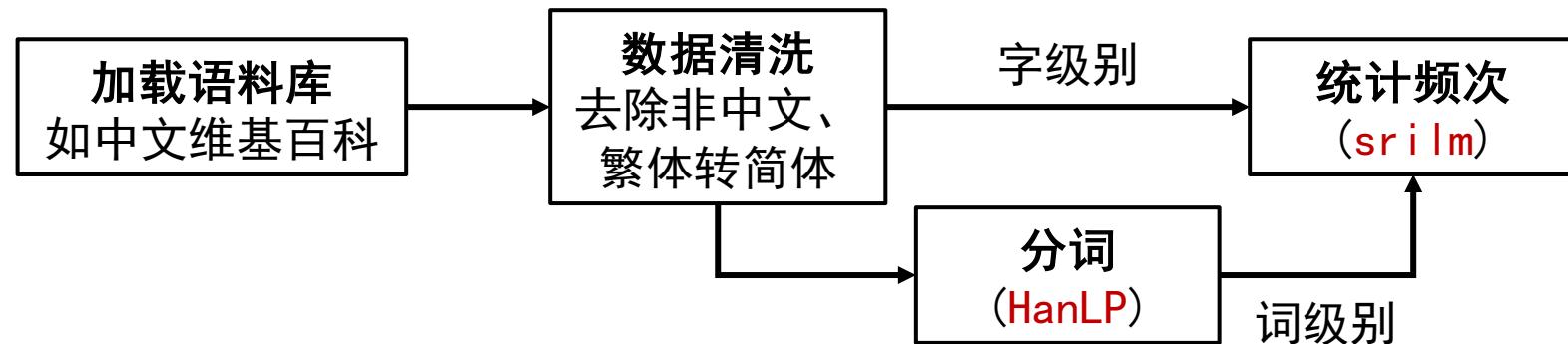
- 可控文本生成
- 群体认知

■ 词频

■ 转移概率

■ 词性

计算流程



中文语料库集合: <https://github.com/InsaneLife/ChineseNLPCorpus>
HanLP: <https://github.com/InsaneLife/ChineseNLPCorpus>
srilm: <http://www.speech.sri.com/projects/srilm/>

■ 词频

■ 转移概率

■ 词性

计算代码实操

■ 词频

■ 转移概率

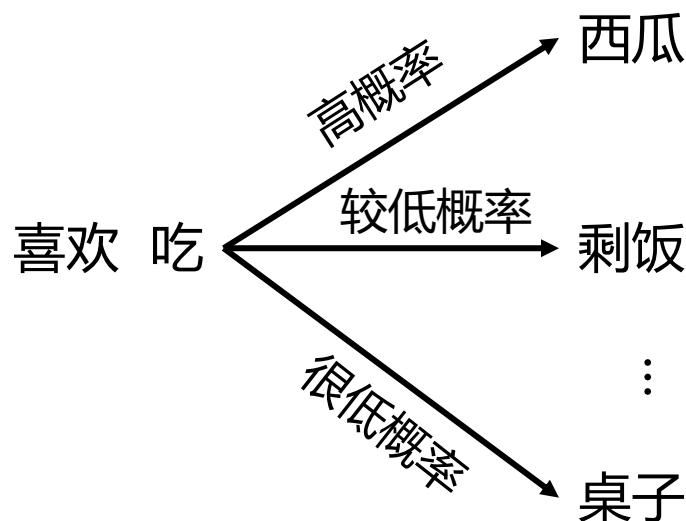
■ 词性

喜欢 吃 ————— ?

■ 词频

■ 转移概率

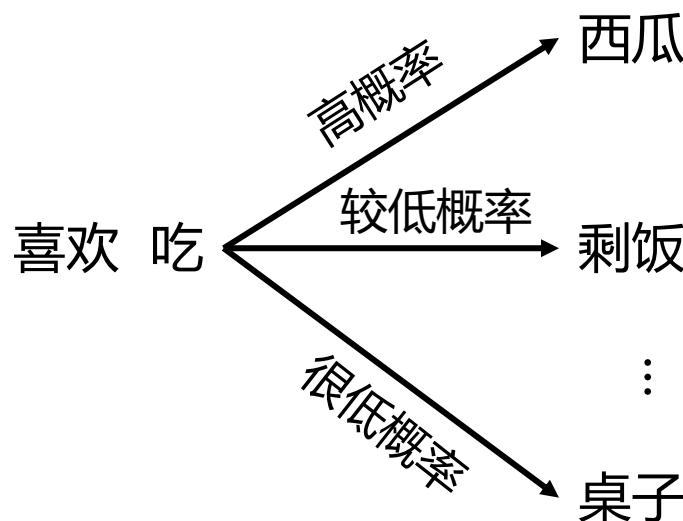
■ 词性



■ 词频

■ 转移概率

■ 词性

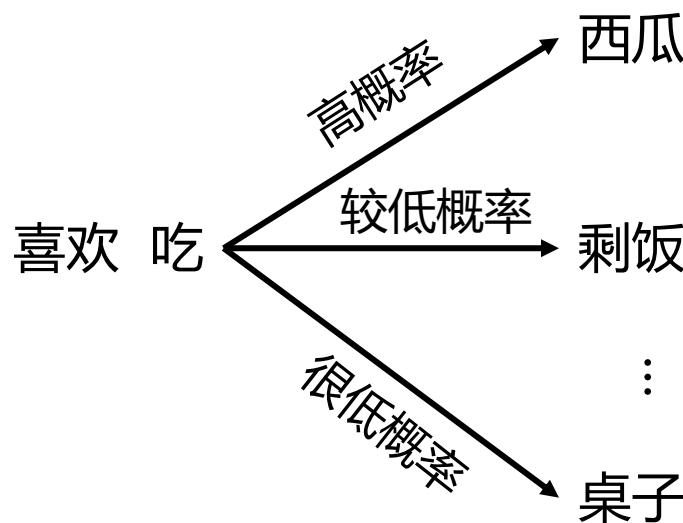
**转移概率 (3-gram)**

$$P(\text{西瓜} \mid \text{吃, 喜欢})$$

■ 词频

■ 转移概率

■ 词性

**转移概率 (3-gram)**

$$P(\text{西瓜} \mid \text{吃, 喜欢})$$

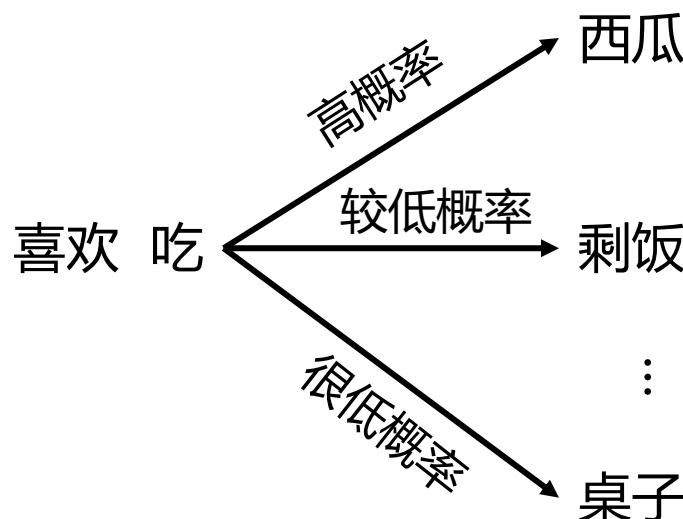
惊异度 (surprisal): 后文出现特定词的惊异程度

$$s = -\log(P(\text{西瓜} \mid \text{吃, 喜欢}))$$

■ 词频

■ 转移概率

■ 词性



转移概率 (3-gram)

$$P(\text{西瓜} \mid \text{吃}, \text{喜欢})$$

惊异度 (surprisal): 后文出现特定词的惊异程度

$$s = -\log(P(\text{西瓜} \mid \text{吃}, \text{喜欢}))$$

熵 (entropy): 后文的不确定度

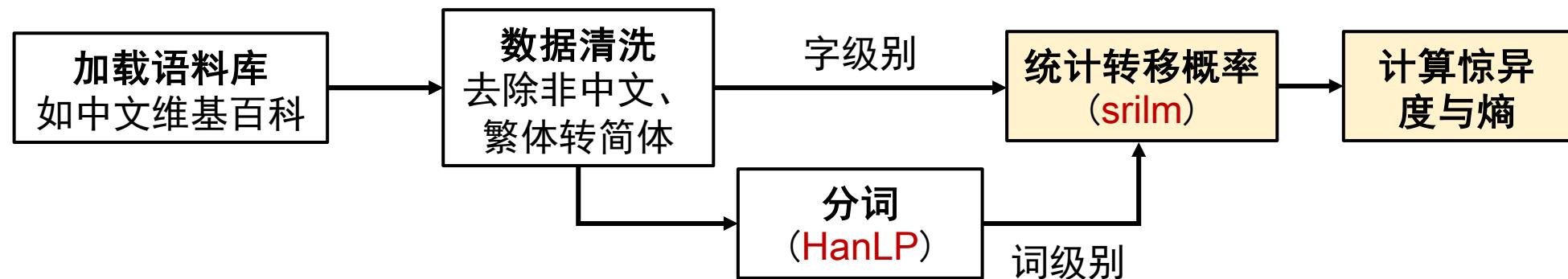
$$e = -\sum_{i=1}^I P(w_i \mid \text{吃}, \text{喜欢}) \times \log(P(w_i \mid \text{吃}, \text{喜欢}))$$

■ 词频

■ 转移概率

■ 词性

计算流程 (基于N-gram)



■ 词频

■ 转移概率

■ 词性

计算代码实操 (基于N-gram)

■ 词频

■ 转移概率

■ 词性

N-gram的局限性

- 基于词串匹配，可考虑的词串
长度有限 (一般小于5)
- 无法捕获语义相似等信息

■ 词频

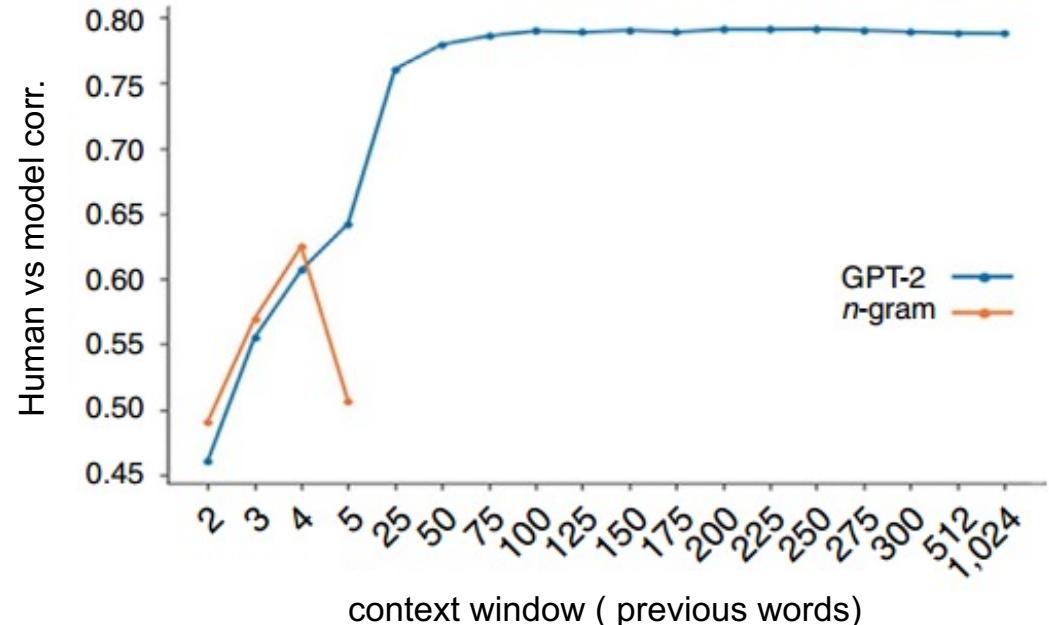
■ 转移概率

■ 词性

N-gram的局限性

- 基于词串匹配，可考虑的词串长度有限 (一般小于5)
- 无法捕获语义相似等信息

N-gram、GPT与人一致程度比较

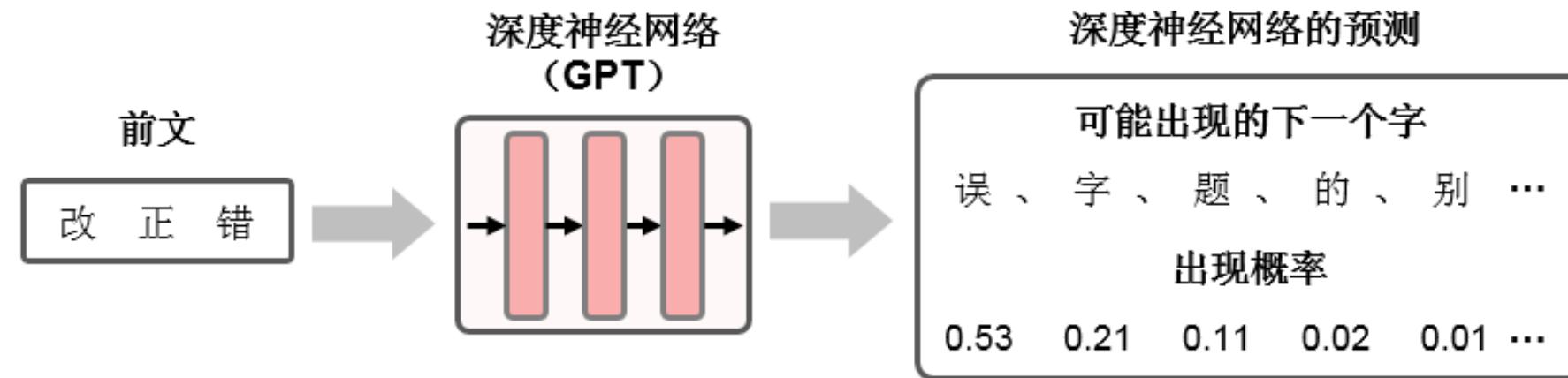


■ 词频

■ 转移概率

■ 词性

计算流程 (基于GPT)

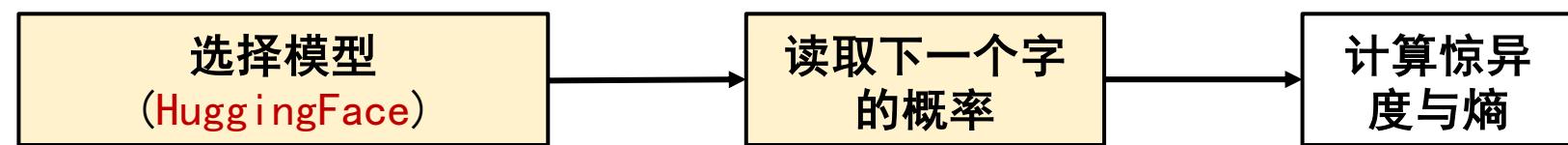


■ 词频

■ 转移概率

■ 词性

计算流程 (基于GPT)



计算代码实操 (基于GPT)

■ 词频

■ 转移概率

■ 词性

这个门被锁了，锁很难被打开。

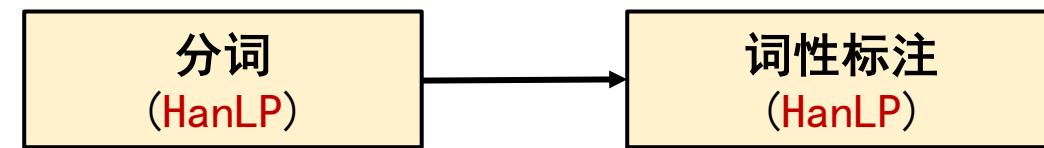


■ 词频

■ 转移概率

■ 词性

计算流程



计算代码实操

1. 概述

- 代码平台
 - 模型结构
 - 语言任务
- a 共现矩阵
- b 词向量

2. 特征计算

- 词汇特征
 - 语义特征
 - 句法特征
- c 上下文有关的语义表征

3. 其他案例

- 可控文本生成
- 群体认知

■ 共现矩阵

■ 词向量

■ 上下文有关

计算流程

例句

小明喜欢吃西瓜。小明
喜欢打篮球。小明经常
去花店。

上下文关系 (size=1)

小明喜欢
. 小明喜欢
. 小明经常

“小明”的共现向量

频次	小明	.	喜欢	吃	西瓜	打	篮球	经常	去	花	店
小明	0	2	2	0	0	0	0	1	0	0	0

■ 共现矩阵

■ 词向量

■ 上下文有关

计算流程

开源词向量

<https://github.com/Embedding/Chinese-Word-Vectors>

<https://nlp.stanford.edu/projects/glove/>

<https://drive.google.com/file/d/0B7XkCwpI5KDYNINUTTISS21pQmM/edit?resourcekey=0-wjGZdNAUop6WykTtMip30g>

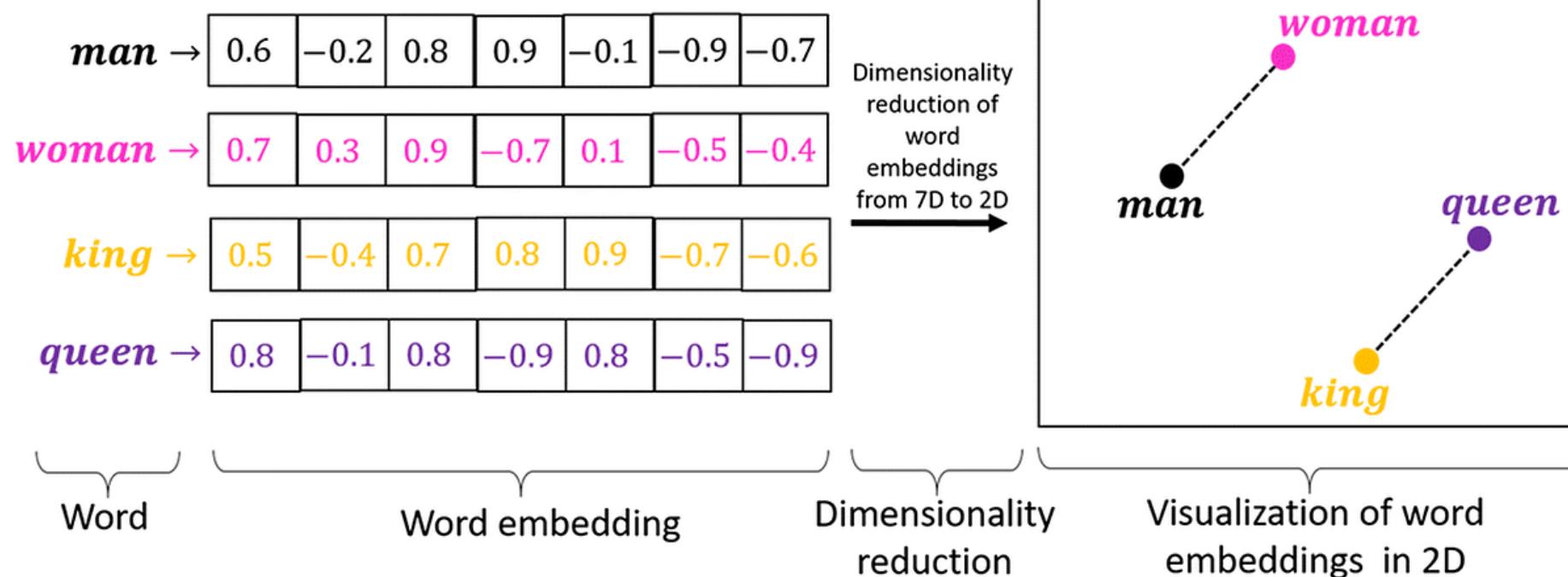
Corpus	Context Features			
	Word	Word + Ngram	Word + Character	Word + Character + Ngram
Baidu Encyclopedia 百度百科	300d	300d	300d	300d / PWD: 5555
Wikipedia_zh 中文维基百科	300d	300d	300d	300d
People's Daily News 人民日报	300d	300d	300d	300d
Sogou News 搜狗新闻	300d	300d	300d	300d
Financial News 金融新闻	300d	300d	300d	300d
Zhihu_QA 知乎问答	300d	300d	300d	300d
Weibo 微博	300d	300d	300d	300d
Literature 文学作品	300d	300d / PWD: z5b4	300d	300d / PWD: yenb
Complete Library in Four Sections 四库全书*	300d	300d	NAN	NAN
Mixed-large 综合 Baidu Netdisk / Google Drive	300d 300d	300d 300d	300d 300d	300d 300d

■ 共现矩阵

■ 词向量

■ 上下文有关

计算流程



■ 共现矩阵

■ 词向量

■ 上下文有关

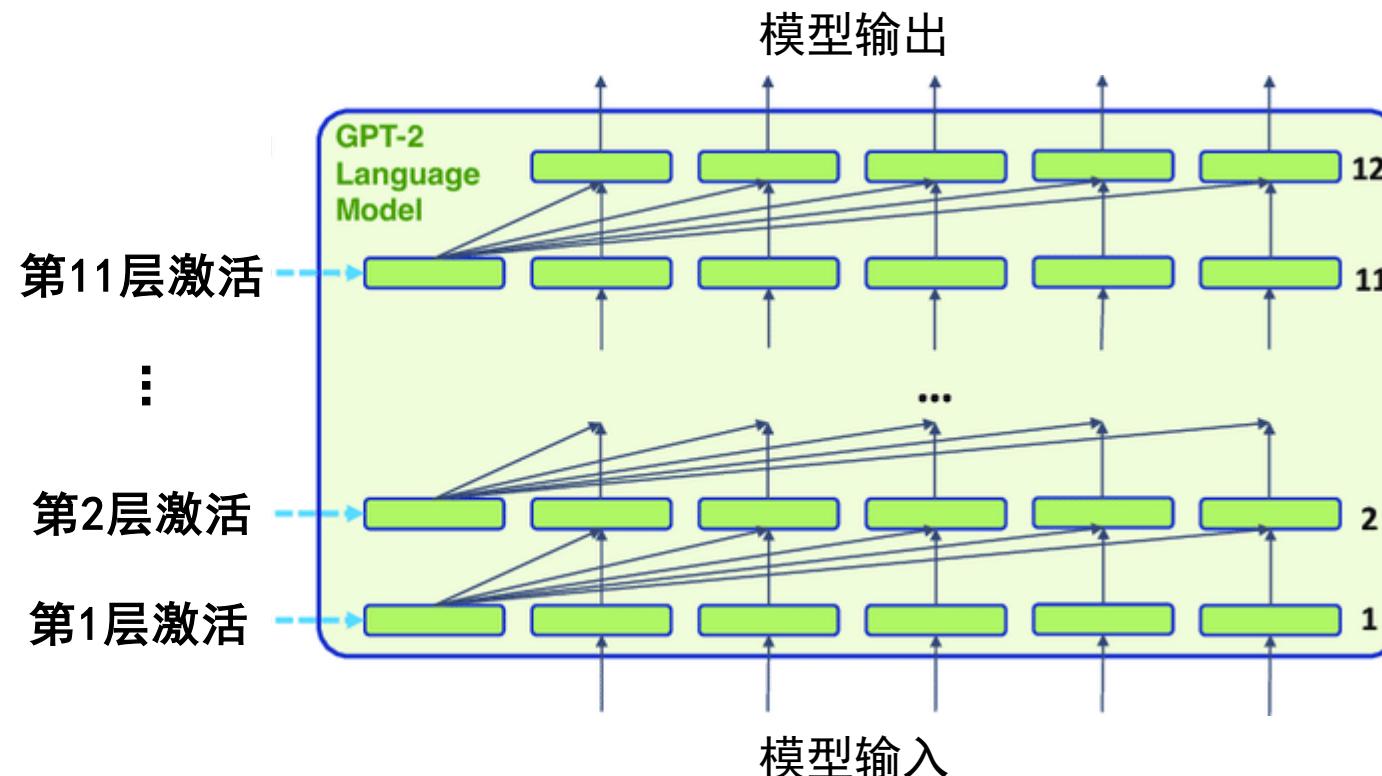
计算代码实操

■ 共现矩阵

■ 词向量

■ 上下文有关

计算流程

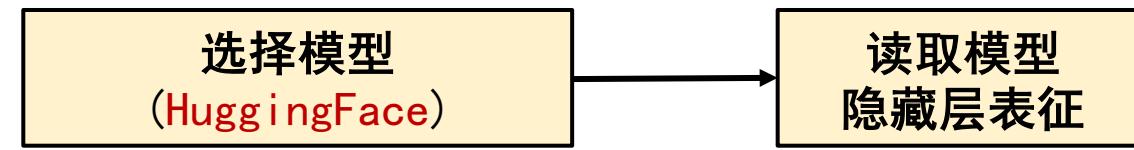


■ 共现矩阵

■ 词向量

■ 上下文有关

计算流程



计算代码实操

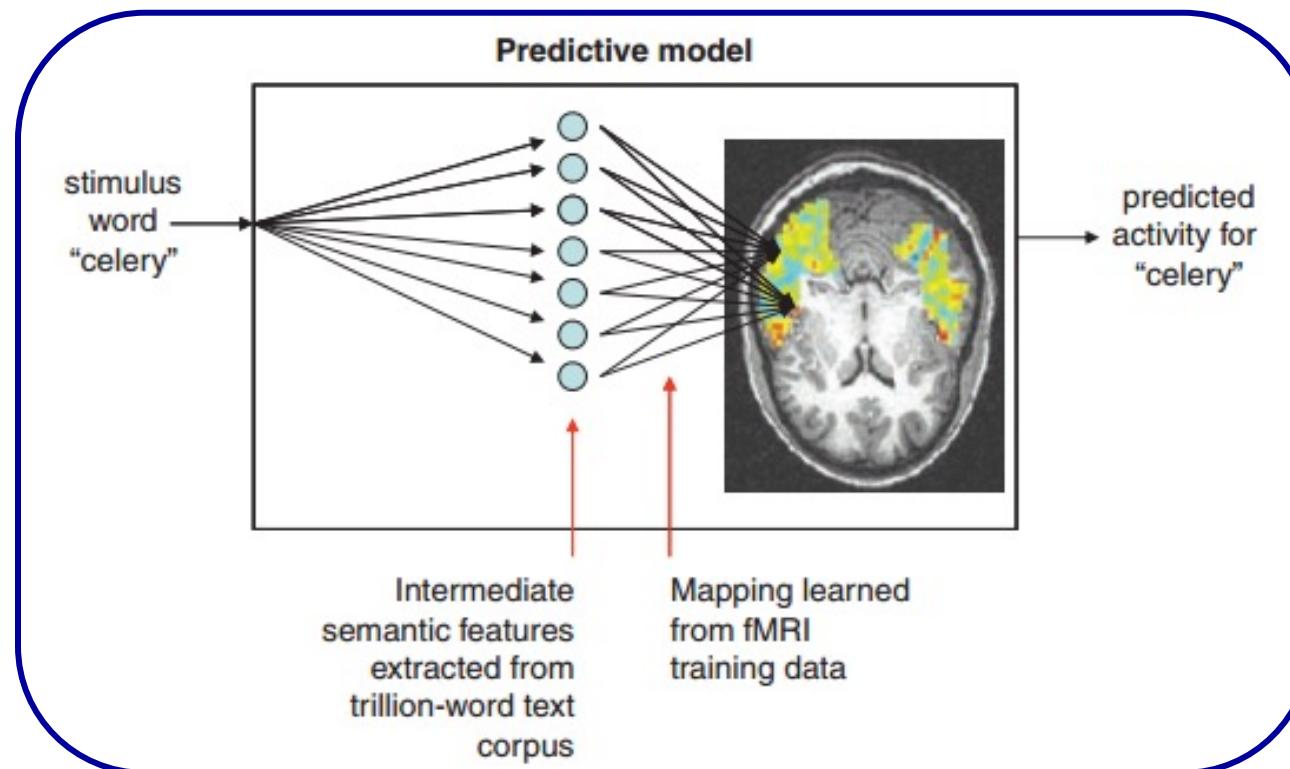
■ 共现矩阵

■ 词向量

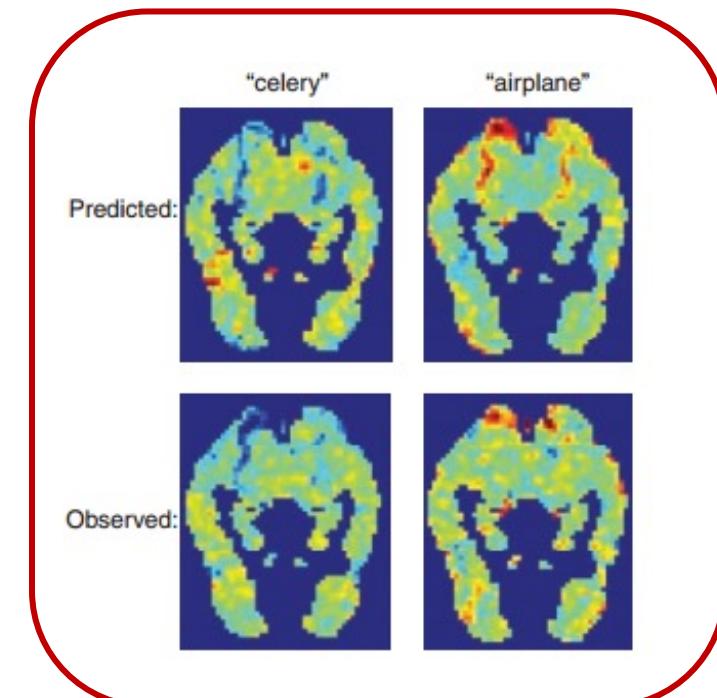
■ 上下文有关

案例1：基于共现矩阵预测词的神经响应

如何应用



研究结果



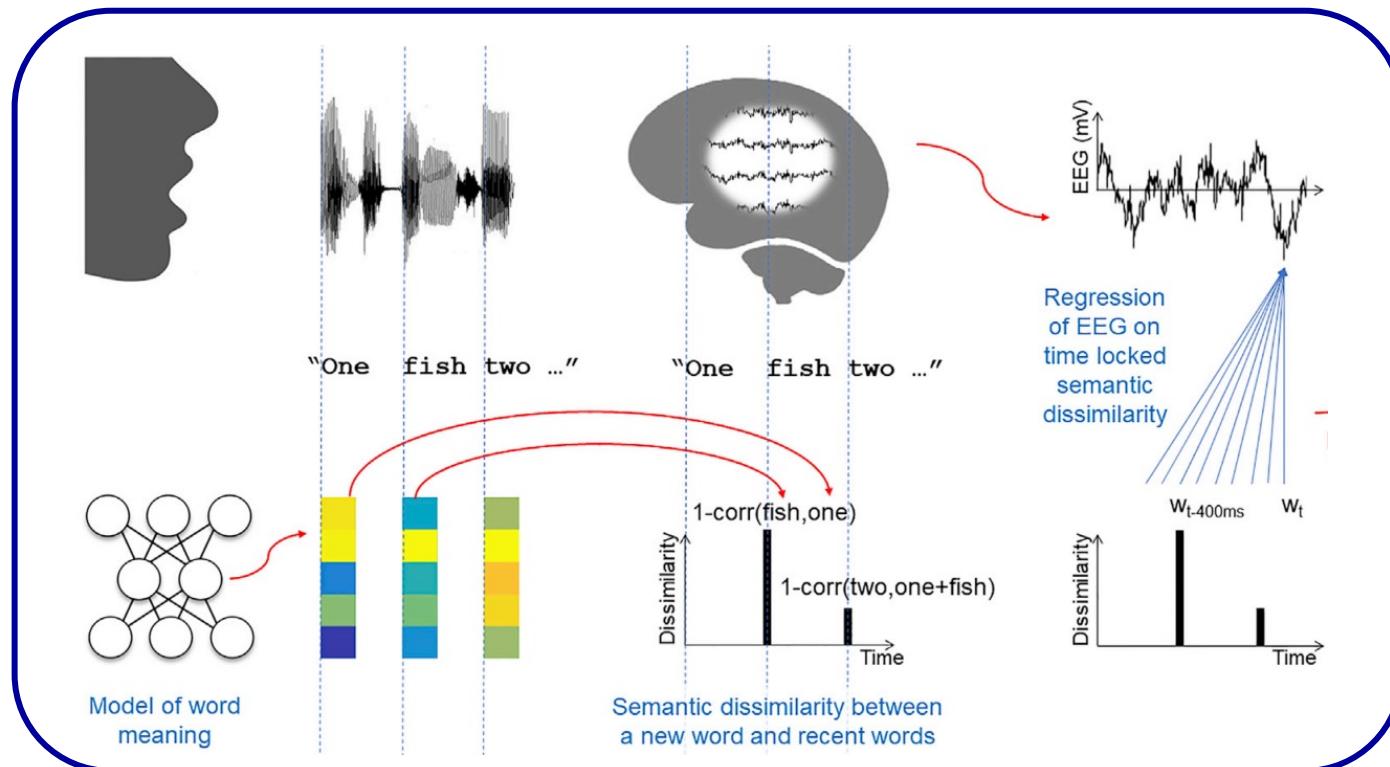
■ 共现矩阵

■ 词向量

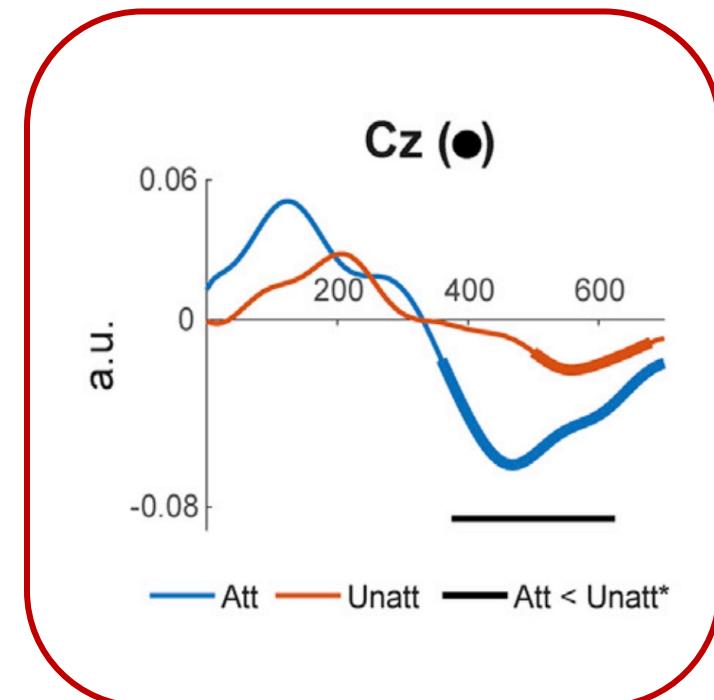
■ 上下文有关

案例2：连续语音语义特征的神经编码

如何应用



研究结果



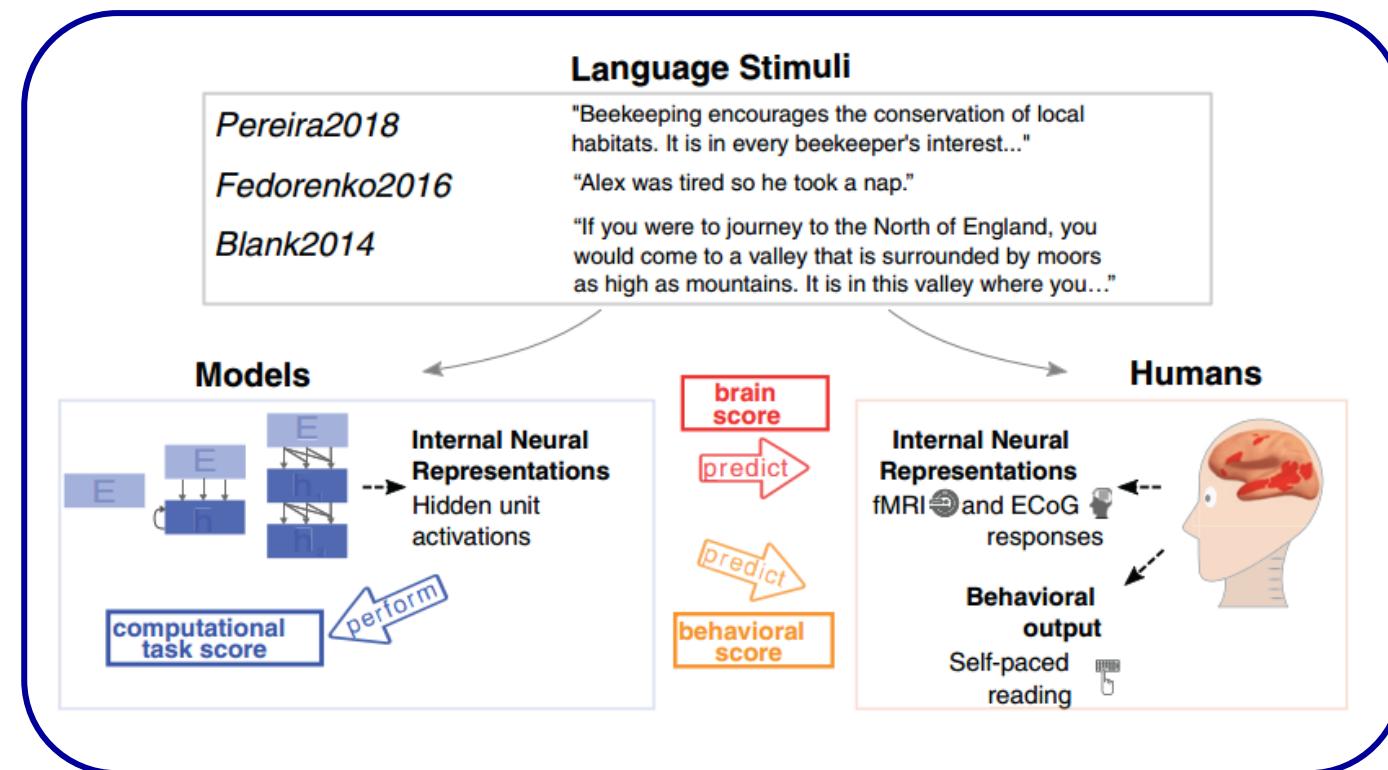
■ 共现矩阵

■ 词向量

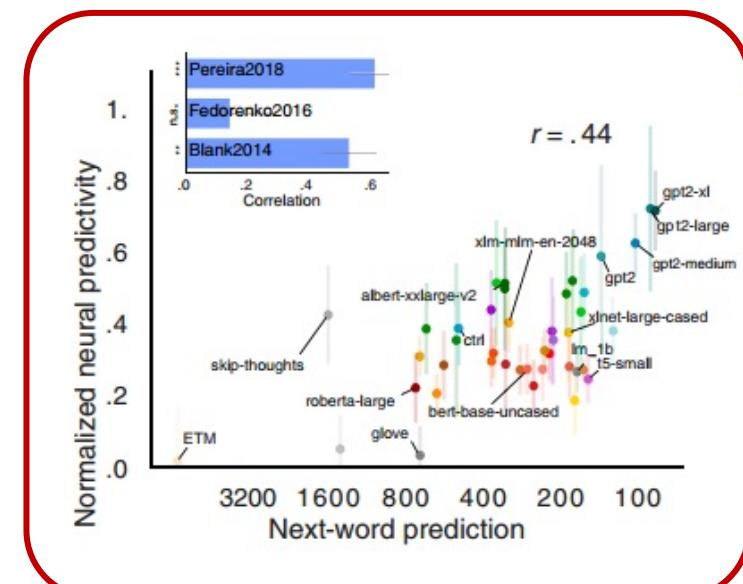
■ 上下文有关

案例3：DNN语义表征建模行为与神经响应 -> 预测下一个字的计算机制

如何应用



研究结果



句法特征

1. 概述

- 代码平台
- 模型结构
- 语言任务

2. 特征计算

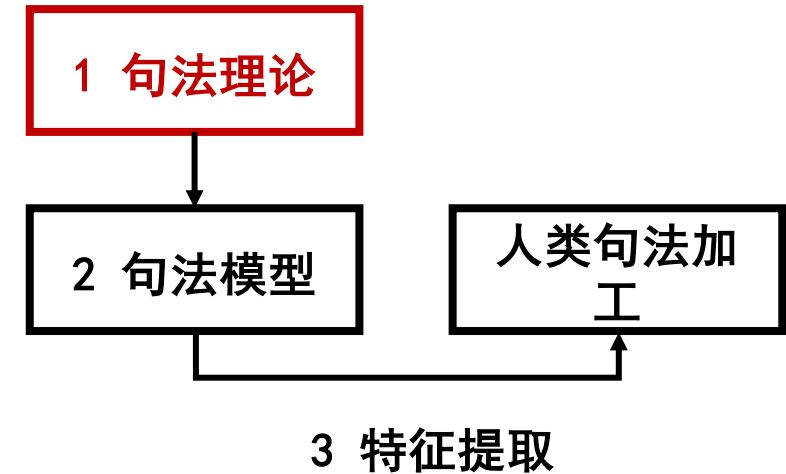
- 词汇特征
- 语义特征
- 句法特征

3. 其他案例

- 可控文本生成
- 群体认知

句法理论

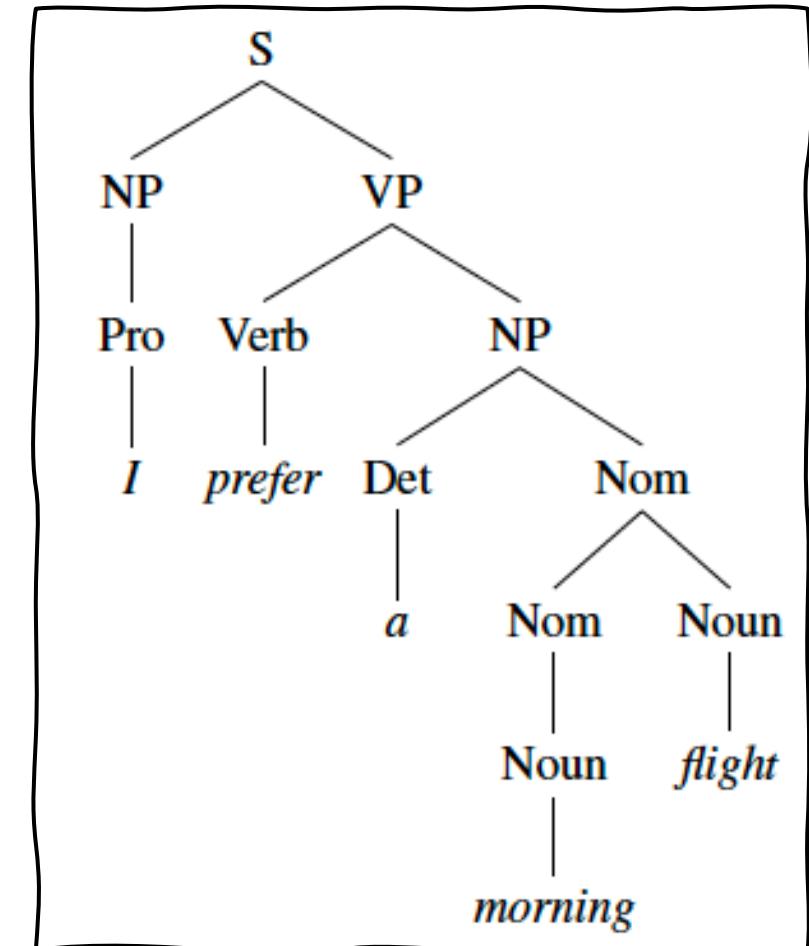
- 成分句法 (Constituency grammar)
- 依存句法 (Dependency grammar)
- 其他句法
 - 组合范畴句法 (Combinational category grammar)
 - Minimalist grammar



句法理论

- 成分句法 (Constituency grammar)
 - 成分 (Constituency)
 - 上下文无关文法 (Context-free grammars, CFGs)

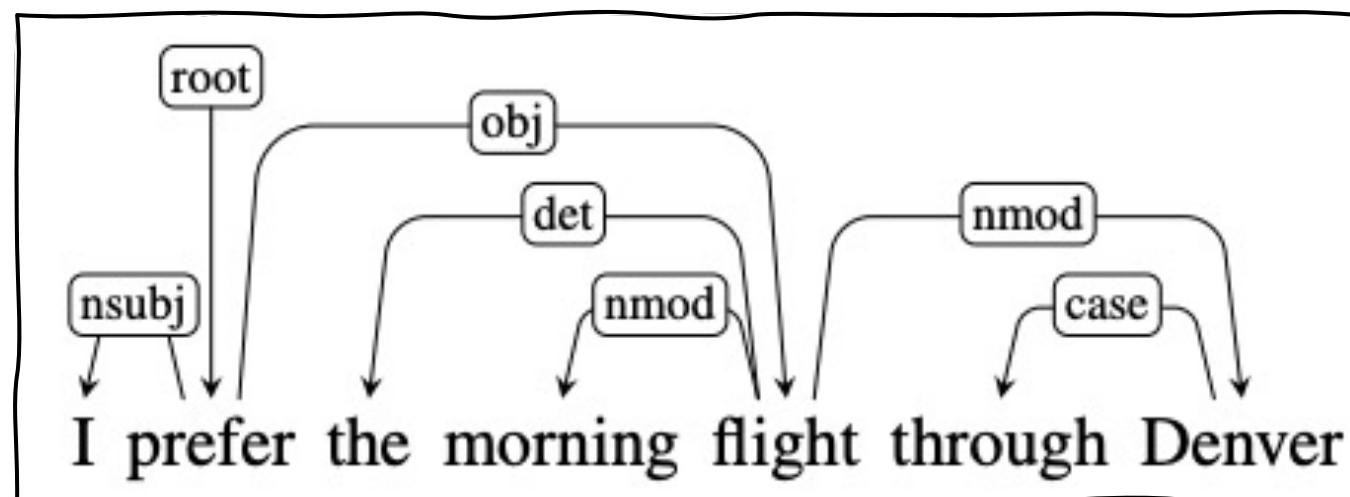
Grammar Rules	
$S \rightarrow NP VP$	
$NP \rightarrow Pronoun$	
	Proper-Noun
	Det Nominal
$Nominal \rightarrow Nominal\ Noun$	
	Noun
$VP \rightarrow Verb$	
	Verb NP
	Verb NP PP
	Verb PP
$PP \rightarrow Preposition\ NP$	



括号表示方法: $[S [NP [Pro I]] [VP [V prefer] [NP [Det a] [Nom [Nom [N morning]] [N flight]]]]]$

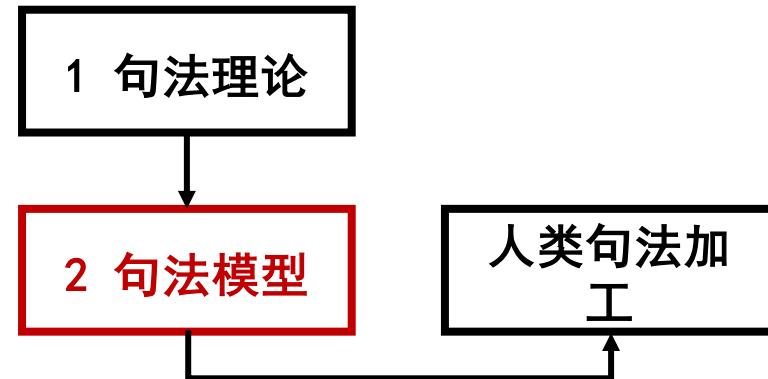
句法理论

- 依存句法 (Dependency grammar)
 - 核心词 (Head) 、依存词 (Dependent) 、依存关系 (Dependency Relation)
 - 投射性 (Projectivity) : 交叉依存



句法解析模型

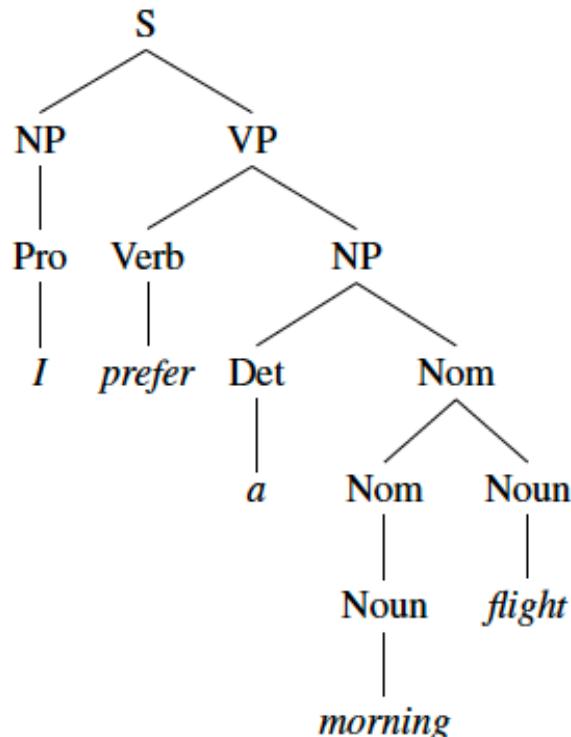
- 数据: 句法树库 (Treebanks)
- 模型表现**
 - 英文: 接近 95% 准确率
 - 中文: 接近 90% 准确率



Type	Model	English PTB-SD 3.3.0		Chinese PTB 5.1	
		UAS	LAS	UAS	LAS
Transition	Ballesteros et al. (2016)	93.56	91.42	87.65	86.21
	Andor et al. (2016)	94.61	92.79	—	—
	Kuncoro et al. (2016)	95.8	94.6	—	—
Graph	Kiperwasser & Goldberg (2016)	93.9	91.9	87.6	86.1
	Cheng et al. (2016)	94.10	91.49	88.1	85.7
	Hashimoto et al. (2016)	94.67	92.90	—	—
	Deep Biaffine	95.74	94.08	89.30	88.23

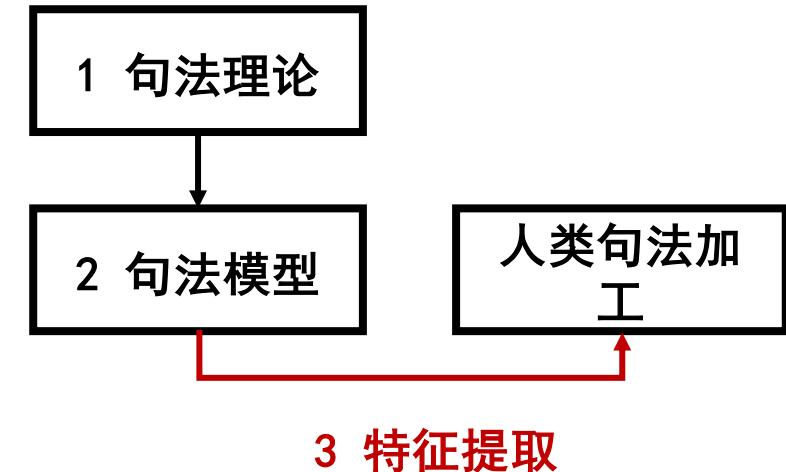
定义句法特征：结构复杂性

- Tree depth
- Distance of dependency ([Wilson, 2018](#))
- **Structure building** ([Brennan, 2012](#), [Nelson, 2017](#))
- Parsing steps (Top-down/bottom-up) ([Brennan, 2023](#))
- ...

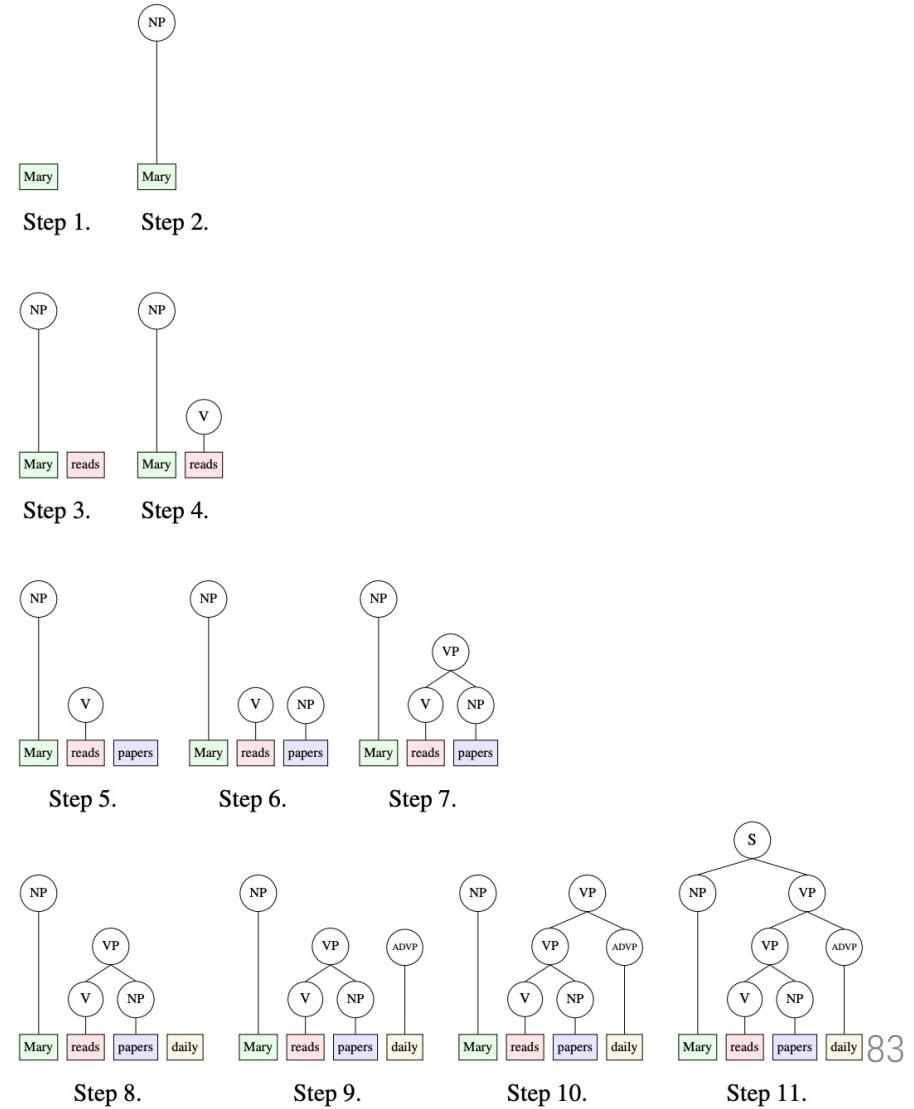
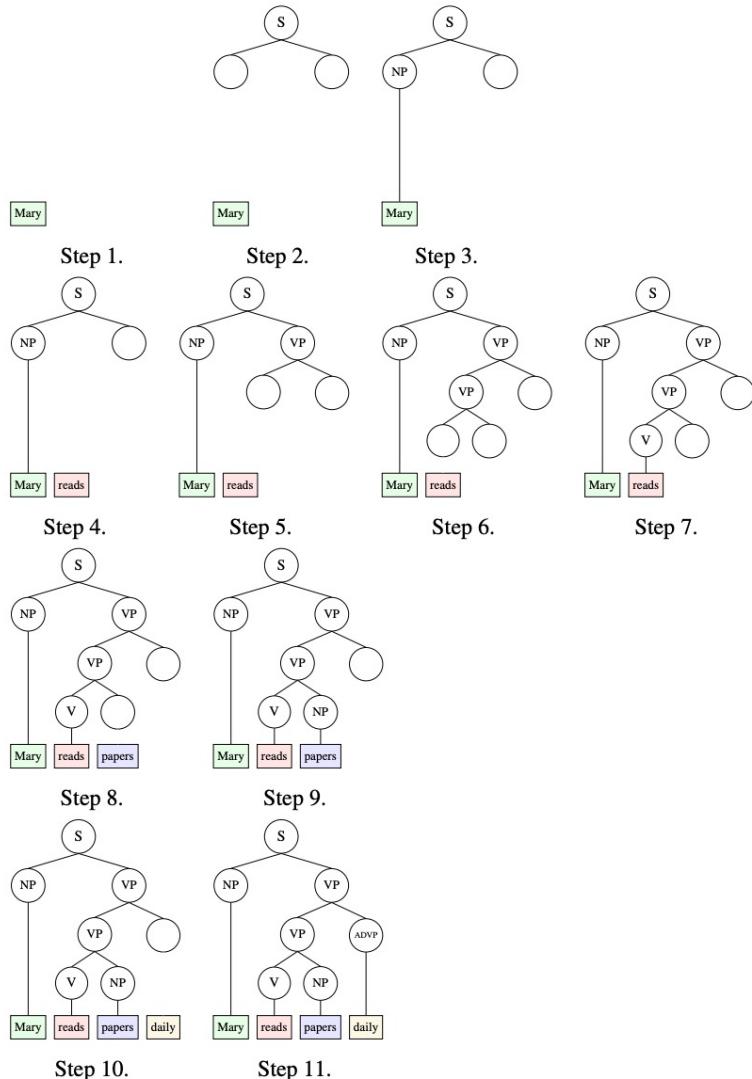


[S [NP [Pro I]] [VP [V prefer] [NP [Det a] [Nom [Nom [N morning]] [N flight]]]]]

e.g. Number of node closing -> number of right brackets



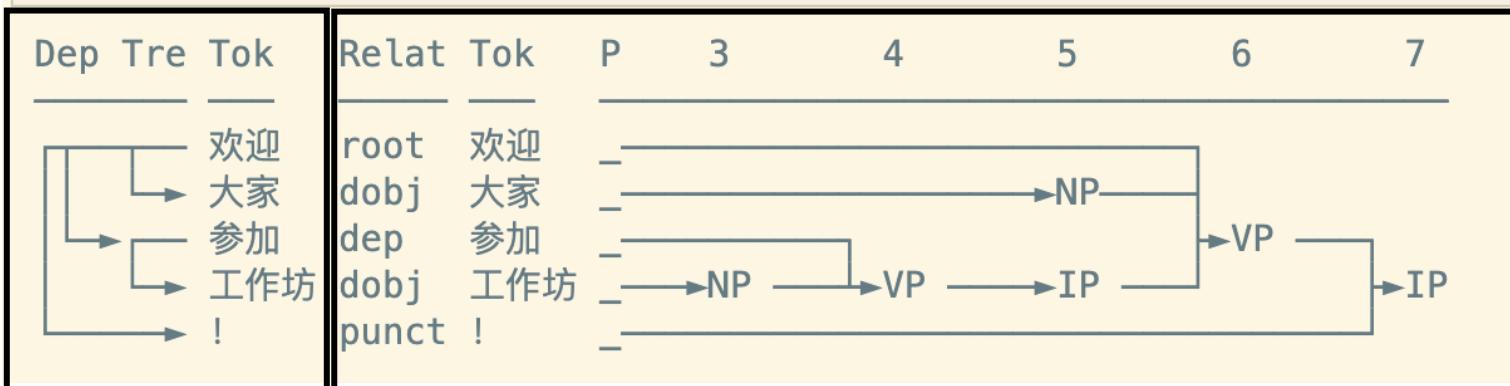
- Parsing steps (Top-down/bottom-up) ([Brennan, 2023](#))



句法特征抽取实践：基于Hanlp

```
Hanlp = hanlp.load(hanlp.pretrained.mtl.CLOSE_TOK_POS_NER_SRL_DEP_SDG_CON_ELECTRA_SMALL_ZH)
doc = Hanlp('欢迎大家参加工作坊!', tasks=['dep', 'con'])
doc.pretty_print()
```

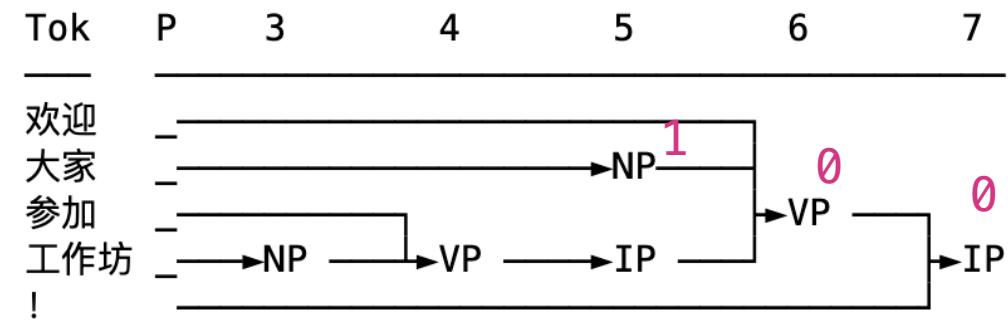
✓ 0.1s



句法特征抽取实践：基于Hanlp

• 成分句法：

- 数据结构：嵌套列表
- 标注方式：[Chinese Treebank](#)



```

Hanlp = hanlp.load(hanlp.pretrained.mtl.CLOSE_TOK_POS_NER_SRL_DEP_SDPCON_ELECTRA_SMALL_ZH)
doc = Hanlp('欢迎大家参加工作坊!')
tree = doc['con']
tree[0, 0, 1]

```

✓ 0.5s

Python

['NP', [['PN', ['大家']]]]

句法特征抽取实践：基于Hanlp

• 成分句法：

- 数据结构：嵌套列表
- 标注方式：[Chinese Treebank](#)

输入
“欢迎大家参加工作坊！”

`tree = doc['con']`

tree

括号表示法: `tree.pformat()`

```
(TOP(IP(VP(VV欢迎)(NP(PN大家))(IP(VP(VV参
加)(NP(NN工作坊))))))(PU! ))
```

Chomsky范式: `tree.chomsky_normal_form()`

```
(TOP(IP(VP(VV欢迎)(VP|<NP-IP>(NP(PN大
家))(IP(VP(VV参加)(NP(NN工作坊))))))(PU! ))
```

句法树深度: `tree.height()`

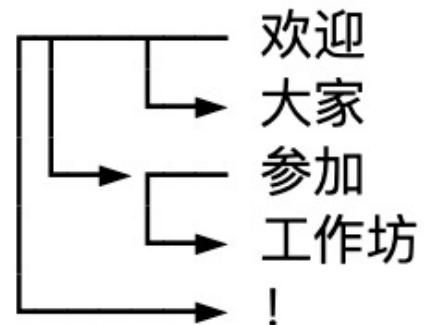
括号结构

word	left_bracket	right_bracket
欢迎	3	1
大家	2	2
参加	2	1
工作坊	2	4
!	1	2

句法特征抽取实践：基于Hanlp

• 依存句法：

- 数据结构：[(head, relation), ...]
- 标注方式：[Universal dependency \(UD\)](#)



```

doc = Hanlp('欢迎大家参加工作坊！')
Hanlp = hanlp.load(hanlp.pretrained.mtl.CLOSE_TOK_POS_NER_SRL_DEP_SDPCON_ELECTRA_SMALL_ZH)
doc['tok/coarse'], doc['dep']
  
```

✓ 0.5s

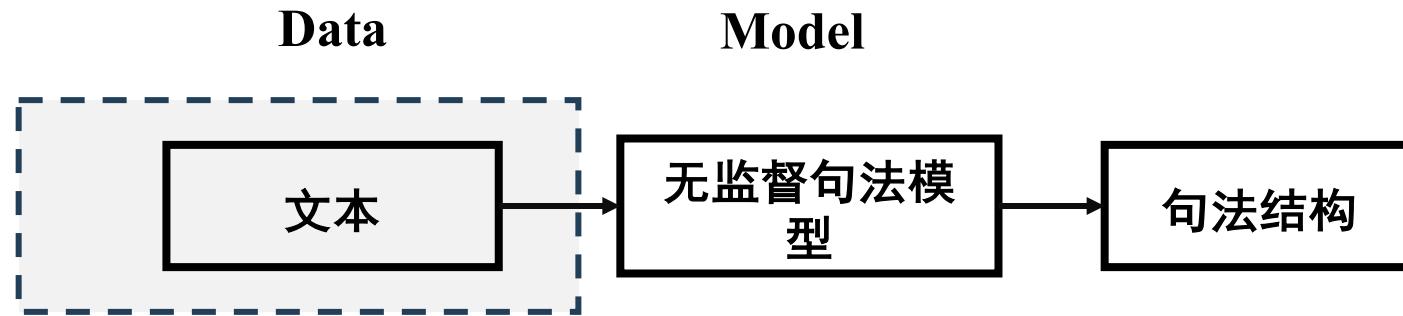
```

(['欢迎', '大家', '参加', '工作坊', '!'],
 [(0, 'root'), (1, 'dobj'), (1, 'dep'), (3, 'dobj'), (1, 'punct')])
  
```

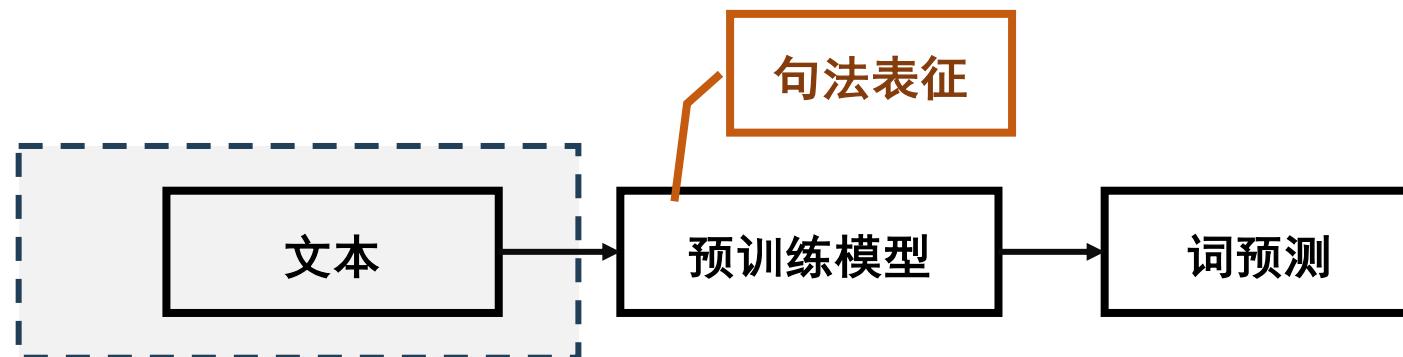
	word	head	rel
0	欢迎	0	root
1	大家	1	dobj
2	参加	1	dep
3	工作坊	3	dobj
4	!	1	punct

讨论

- 模型是否能够增量地进行句法解析? ([Kitaev, 2022](#))
- 模型是否能够无监督地构建句法结构? ([Cao, 2020](#))



- 我们需要显式的句法结构或特征吗?

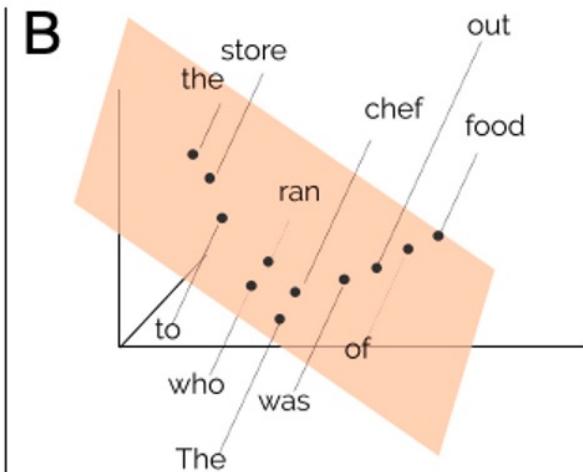


模型中的句法表征

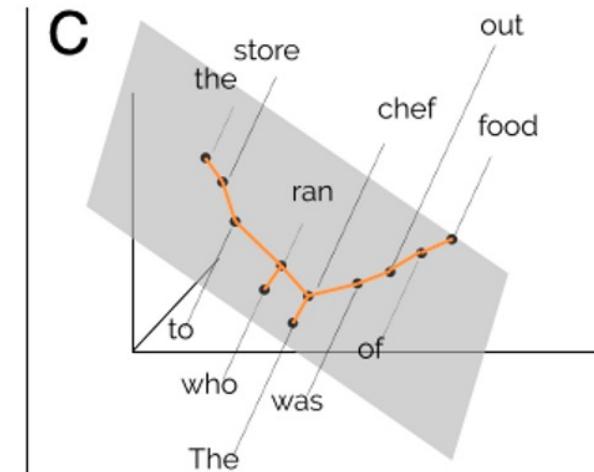
- 注意力 & 词嵌入 ([Manning, 2019](#); [Coenen & Reif, 2019](#); [Rogers, 2020](#))
 - 词嵌入可以线性投射到一个潜空间中，再通过向量的距离还原句法结构。



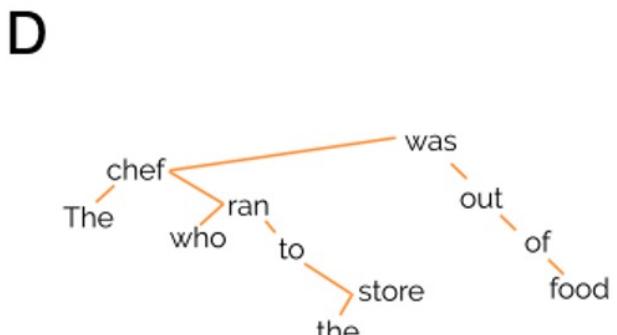
Each of the words of the sentence *The chef who ran to the store was out of food* is internally represented in context as a vector.



A structural probe finds a linear transform of that space under which squared L_2 distance between vectors best reconstructs tree path distance between words.



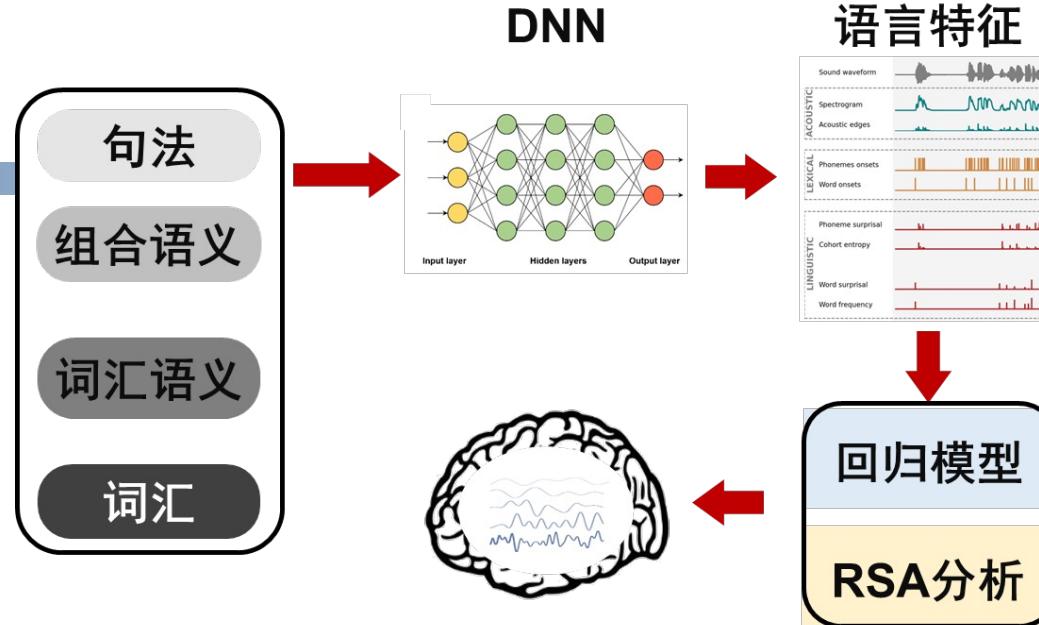
Once in this latent space, the structure of the tree is globally represented by the geometry of the vector space, meaning words that are close in the space are close in the tree.



In fact, the tree can be approximately recovered by taking a minimum spanning tree in the latent syntax space.

小结

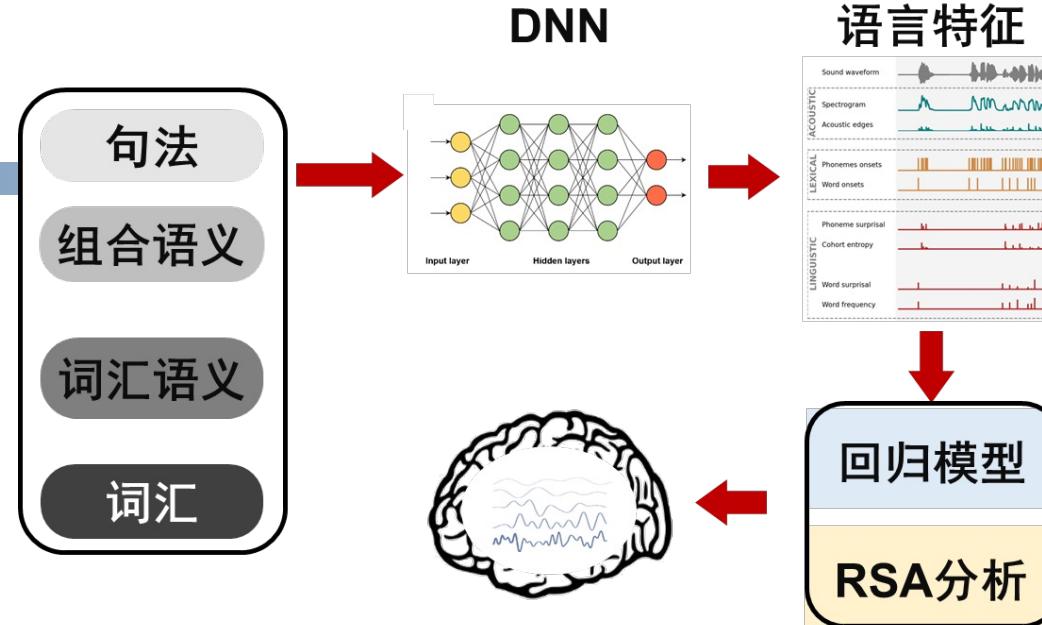
AI帮助解决了什么



- 人类标注 -> AI自动标注
- 严格的实验设计、非自然语料、考察单一变量 -> 对自然语音展开分析、同时考察多个变量
- 繁琐精细的语料设计 -> AI进行语料初筛
- 帮助回答大脑的语言加工机制：如预测下一个字等

小结

AI没有帮助解决什么



- 多种特征相互关联，较难分离不同特征的神经响应
- 未提供基础的科学理论上的突破（**新方法研究旧问题验证旧理论**）

可控文本生成

1. 概述

- 代码平台
- 模型结构
- 语言任务

2. 特征计算

- 词汇特征
- 语义特征
- 句法特征

3. 其他案例

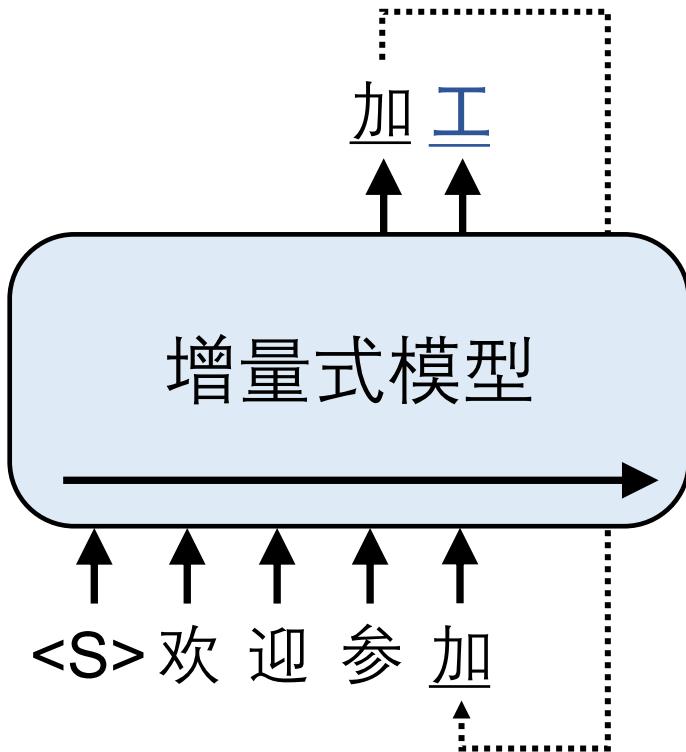
- 可控文本生成
- 群体认知

可控文本生成 文本生成

逐步将输出的词作为输入继续预测

输出：

加 工

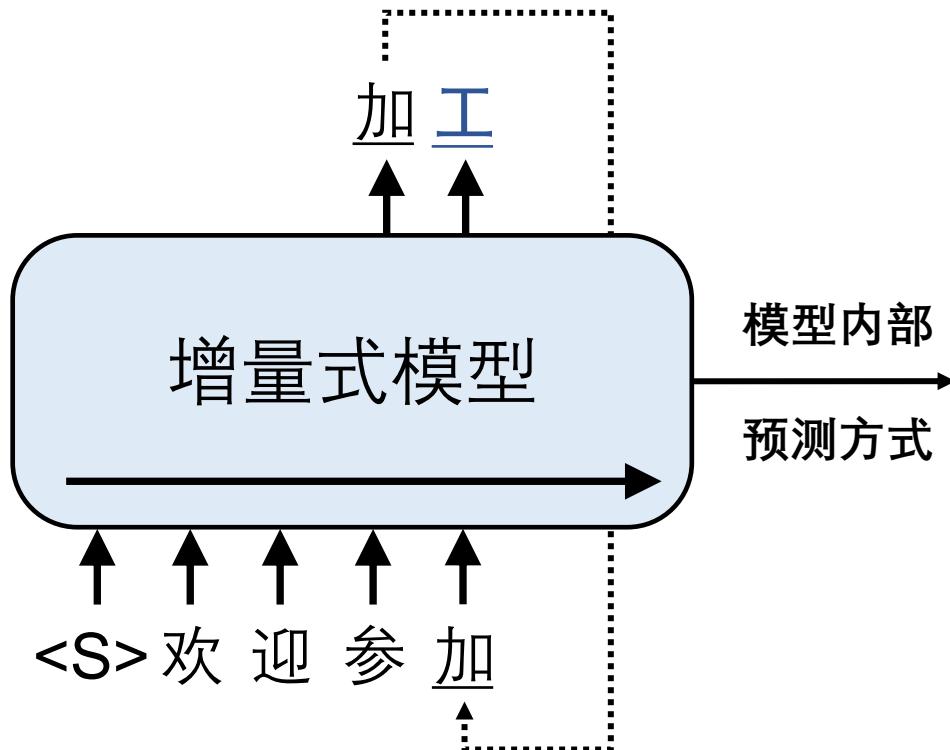


输入：<S> 欢 迎 参 加

语料：欢迎参加工作坊！

逐步将输出的词作为输入继续预测

输出：

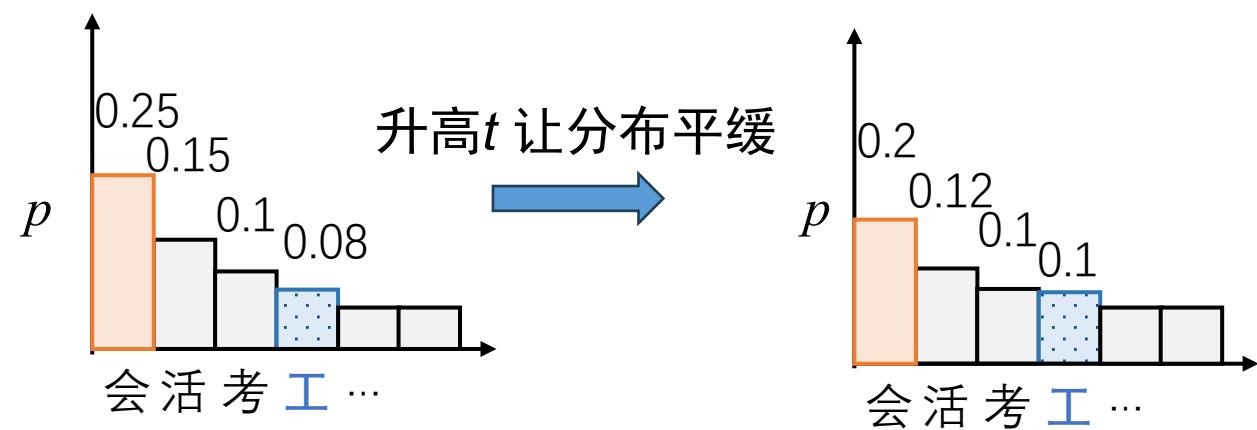


输入：<S>欢迎参加

语料：欢迎参加工作坊！

两种不同的预测方式

1. 贪婪搜索：每次选择最高概率的词
 2. 随机采样：根据概率分布随机选择词
- 可以使用超参数改变概率分布
- 温度 t : 控制概率分布的陡峭程度



输入：欢迎参加 ?

输入: 欢迎参加

1. 贪婪搜索: 每次选择最高概率的词

=====贪婪搜索=====

iteration 0: 欢迎参加的是一个朋友
iteration 1: 欢迎参加的是一个朋友
iteration 2: 欢迎参加的是一个朋友
iteration 3: 欢迎参加的是一个朋友
iteration 4: 欢迎参加的是一个朋友

2. 随机采样: 根据概率分布随机选择词 温度 $t = 0.3$

=====随机搜索, 温度=0.3=====

iteration 0: 欢迎参加乐团, 是一种怎样的
iteration 1: 欢迎参加姐妹们的婚礼, 地方
iteration 2: 欢迎参加的人很多, 有时候会
iteration 3: 欢迎参加的是一个朋友的婚礼
iteration 4: 欢迎参加的是他们家的会员卡

温度 $t = 1.0$

=====随机搜索, 温度=1.0=====

iteration 0: 欢迎参加的是在那里定过蛋糕
iteration 1: 欢迎参加! 2016年高尔夫球团
iteration 2: 欢迎参加来自中国的顶级名校
iteration 3: 欢迎参加的一天里来到这里吃
iteration 4: 欢迎参加提问! 本人想去北京

推荐使用的文本生成参数：

- 如果需要固定答案，选择贪婪搜索
- 如果需要生成多样的语料，采用随机采样：
 - 温度 $t = 0.7 \sim 1.0$ ，温度越高，文本越多样

可控文本生成

要求模型根据生成属性 (和输入), 生成受控文本

生成属性:
<积极情绪>



输出:
我很开心

生成属性:
<消极情绪>
输入:
今天我去考试



输出:
结果考得很差

可控文本生成

生成属性可以相互组合

[Negative] The potato is a pretty **bad idea**. It can make you fat, it can cause you to have a **terrible** immune system, and it can even kill you....

[Positive] The potato chip recipe you asked for! We **love** making these, and I've been doing so for years. I've always had a hard time keeping a recipe secret. I think it's the way our kids **love** to eat them – so many little ones.

[Politics] [Positive] To conclude this series of articles, I will present three of the most **popular** and **influential** works on this topic. The first article deals with the role of women's **political** participation in building a **political** system that is representative of the will of the people.

可控文本生成

生成属性可以规定语言特征

input (Semantic Content)	food : Japanese
output text	Browns Cambridge is good for Japanese food and also children friendly near The Sorrento .
input (Parts-of-speech)	PROPN AUX DET ADJ NOUN NOUN VERB ADP DET NOUN ADP DET NOUN PUNCT
output text	Zizzi is a local coffee shop located on the outskirts of the city .
input (Syntax Tree)	(TOP (S (NP (*) (*)(*)) (VP (*)(NP (NP (*) (*))))))
output text	The Twenty Two has great food
input (Syntax Spans)	(7, 10, VP)
output text	Wildwood pub serves multicultural dishes and is ranked 3 stars
input (Length)	14
output text	Browns Cambridge offers Japanese food located near The Sorrento in the city centre .

可控文本生成

1. 部署微调模型实现可控文本生成
2. 利用chatGPT等大语言模型 + 上下文学习范式实现可控文本生成

1. 概述

- 代码平台
- 模型结构
- 语言任务

2. 特征计算

- 词汇特征
- 语义特征
- 句法特征

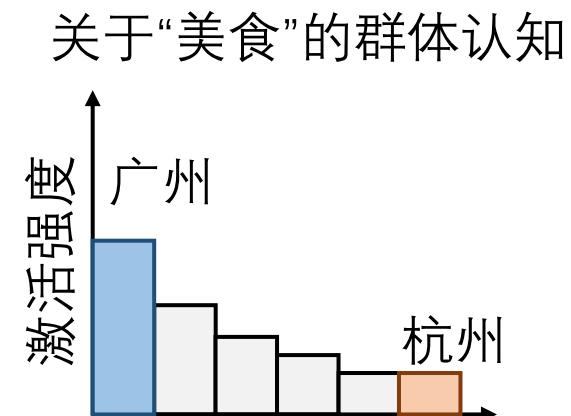
3. 其他案例

- 可控文本生成
- 群体认知

语言 vs. 群体认知

人们在日常生活中产生的文本能够反映群体认知系统的激活强度

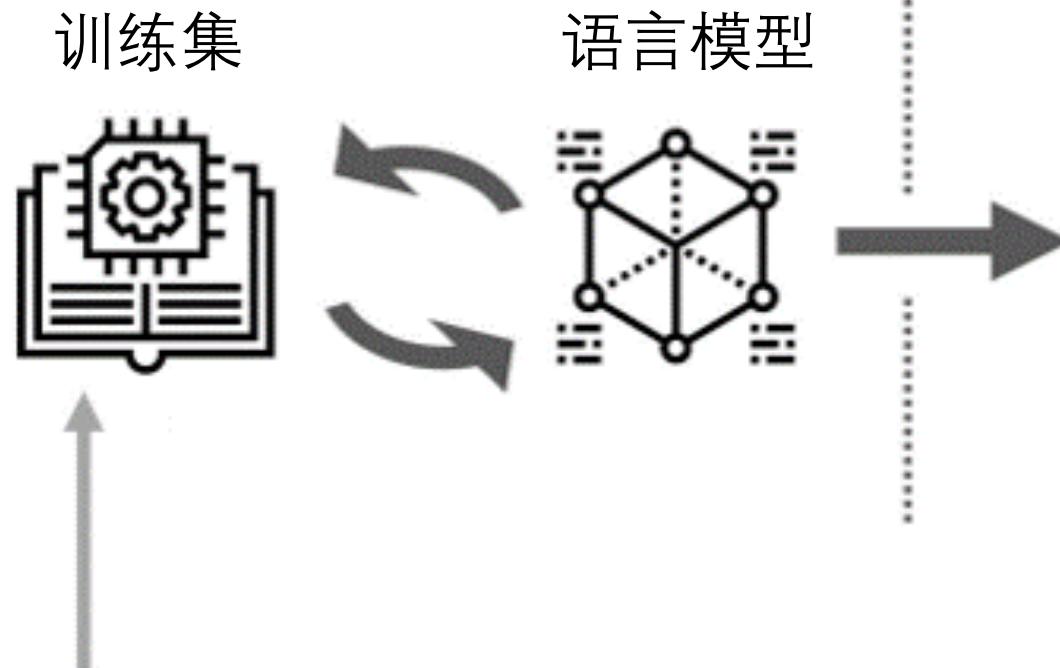
$$P(\text{“广州美食”}) > P(\text{“杭州美食”})$$



Language is a mirror of mind in a deep and significant sense.

----Noam Chomsky

性别歧视



包含偏见的文本

$$P(\text{'She is a doctor'}) < P(\text{'He is a doctor'})$$

$$P(\text{'She is a doctor'}) < P(\text{'She is a nurse'})$$

性别歧视

Association 强度与社会劳工统计结果高度相关

- 女性工作者越多的职业，其词向量与 female 的 association 越强

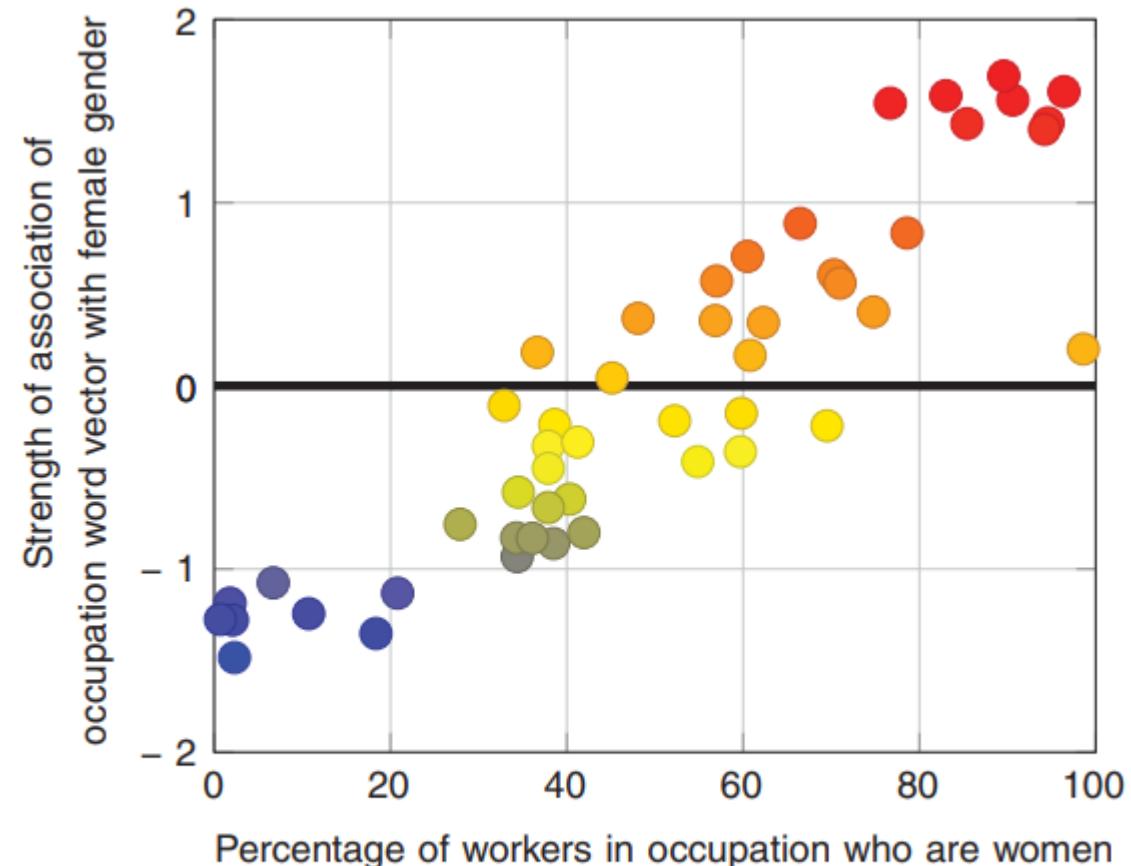
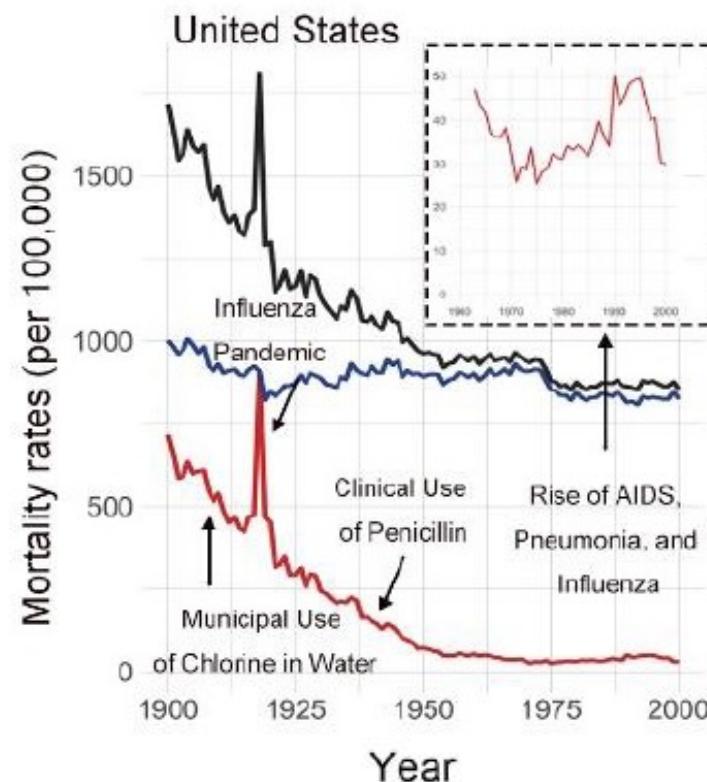


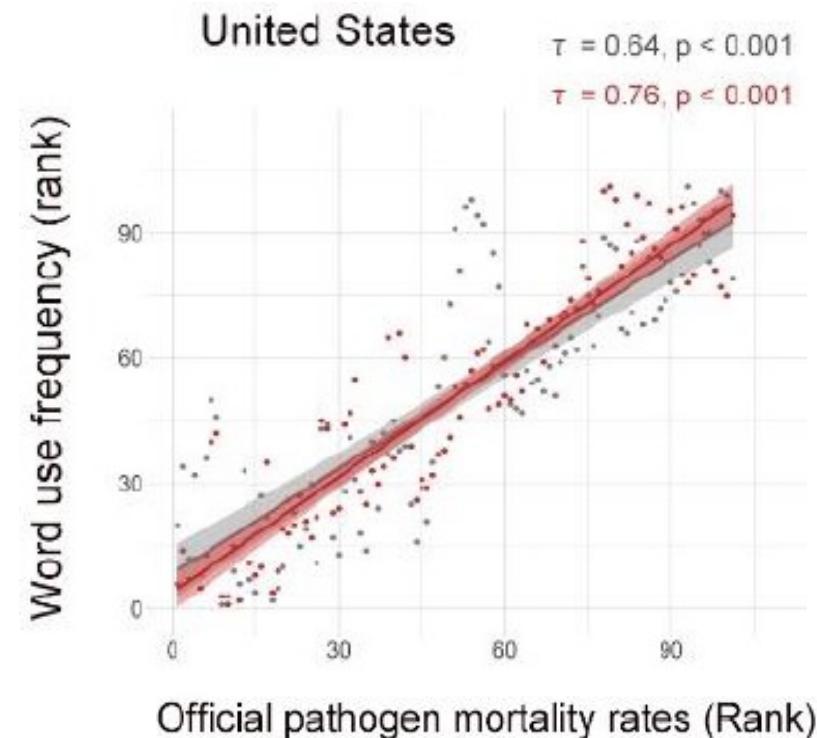
Fig. 1. Occupation-gender association. Pearson's correlation coefficient $\rho = 0.90$ with $P < 10^{-18}$.

语义空间能够反映群体认知变化

流行病死亡率



死亡率 vs. 疾病相关词的词频



流行病越严重的年份，疾病相关词的使用频率越高。

小结

1. 语言模型能够为认知心理学提供特定结构的语言材料
2. 语言模型能够反映群体认知