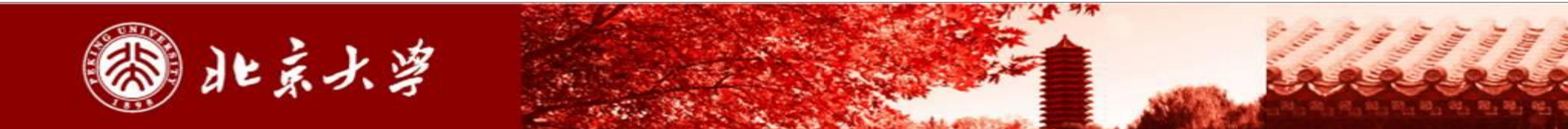
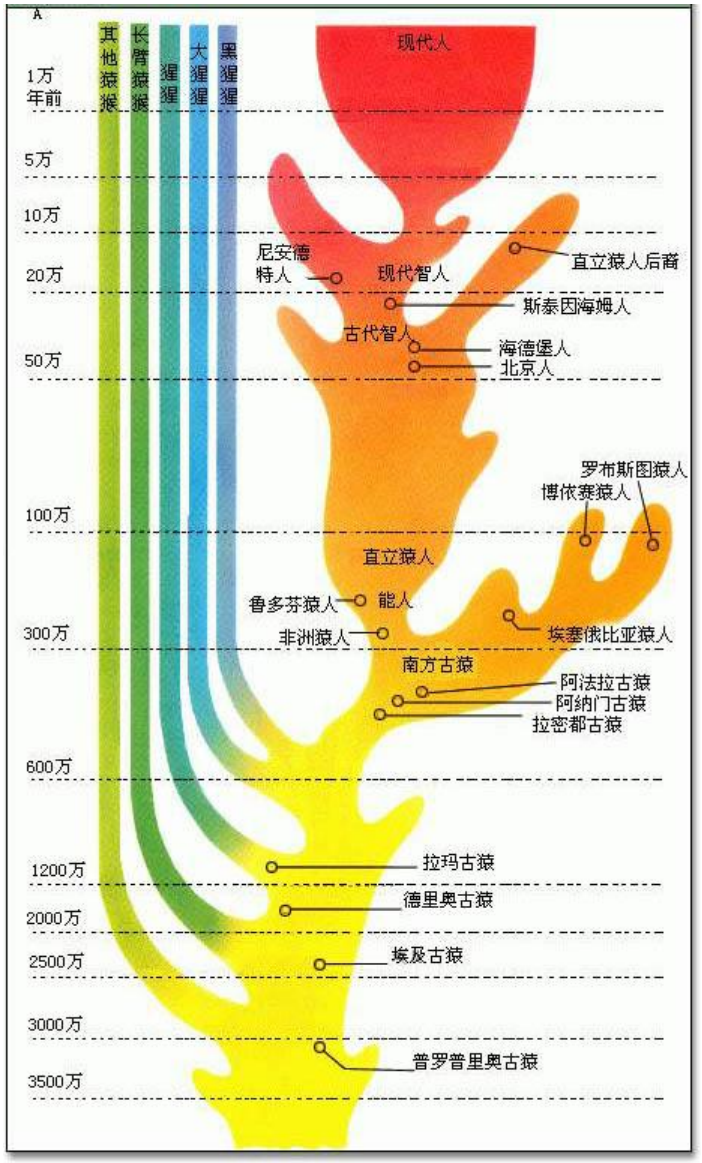
词表示与语言模型

马滟青

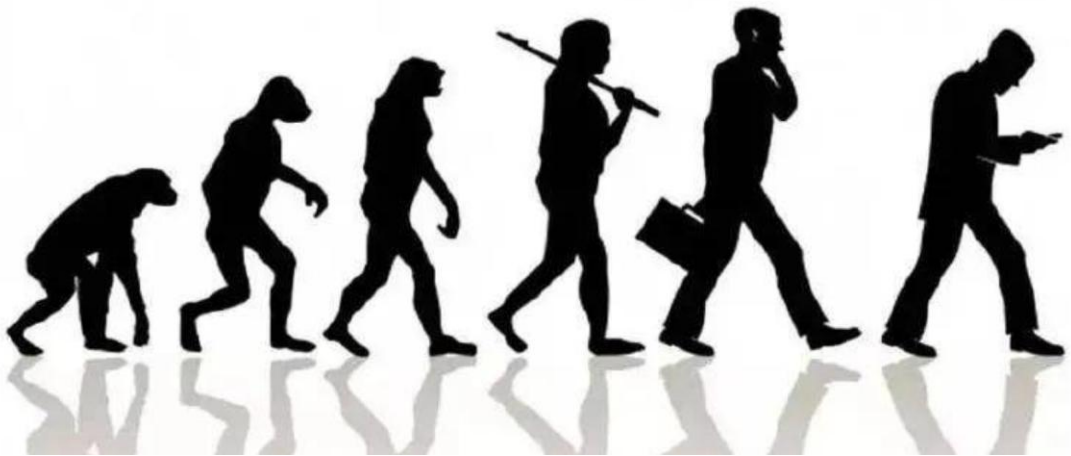
2025 年“微扰量子场论及其应用”前沿讲习班暨前沿研讨会，
山东大学，2025/07/10



背景介绍——语言的重要性



- 大约10万年前，人类学会说话
- 大约5千年前，人类学会写字
- 图灵测试：基于语言
- 语言具有普适性，并捕捉大量智能行为
- 语言和文字的交流促进了人类文明的发展



背景介绍——为何研究自然语言处理?

北京大学 PEKING UNIVERSITY

自然语言处理 (natural language processing, NLP)

- 让机器与人类交流
- 让机器学习到知识
- 推动对语言和语言使用的科学理解

2011, IBM Watson

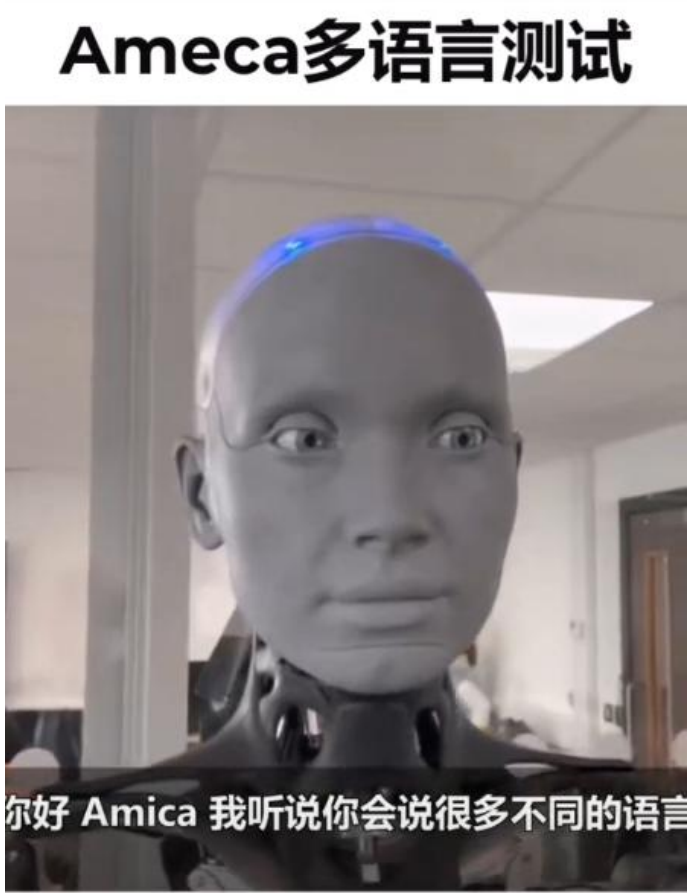
2018, Google BERT

2020, OpenAI GPT-3

2022, OpenAI ChatGPT

背景介绍——NLP可以做些什么？

- 语音识别（speech recognition）
- 文本-语音合成（text-to-speech synthesis）
- 机器翻译（machine translation）
- 信息提取（information extraction）
- 信息检索（information retrieval）
- 问答（question answering）
- 机器写作
- 文本摘要
- ……



2023年4月

背景介绍——NLP难点何在?

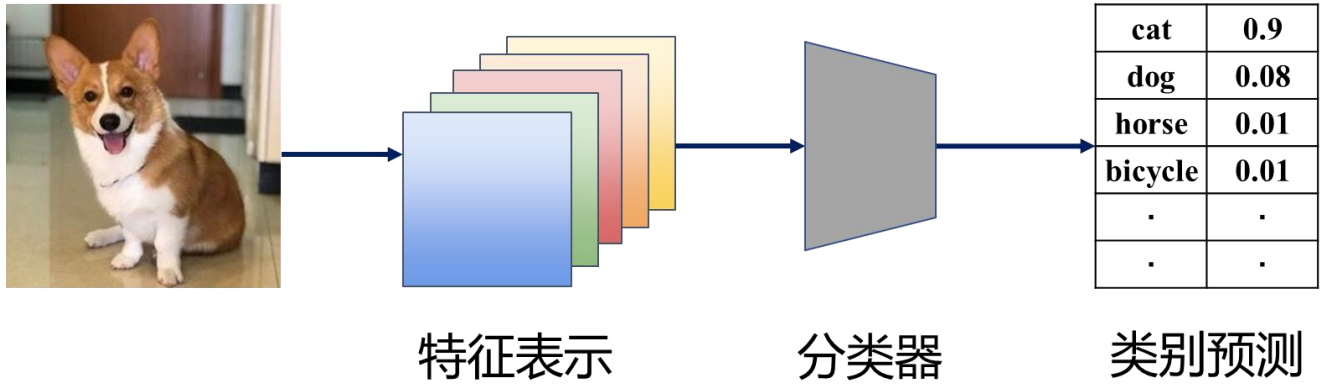
- **文法**（grammar）：定义合法句的句法（syntax）
- **语义规则**（semantic rule）：定义其含义

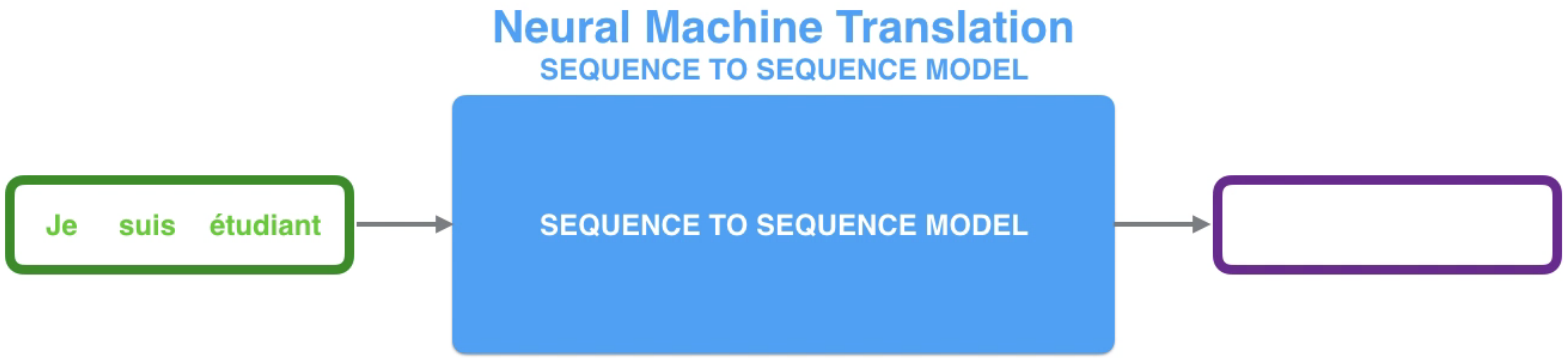
自然语言无法像形式语言一样清晰地表示：

- 不同人在不同时间对语言的判断有区别
  - “Not to be invited is sad”☑
  - “To be not invited is sad”☒?☑?
- 自然语言存在歧义，也是模糊不清的
  - “He saw her duck”-> 她的鸭子？她躲避？
  - “That’s great!”-> how? what?
- 自然语言没有正式定义从符号到对象的映射

自然语言的表示

- 如何将自然语言用数字表达？





自然语言处理

北京大学 PEKING UNIVERSITY

- 分词
- 独热表示与词袋模型
- 分布式词表示
- 递归神经网络

分词（Tokenization）

北京大学 PEKING UNIVERSITY

- **句子切分(断句)（Sentence Tokenization）**
  - 将文本序列分割为独立的句子
  - 例1：“I cannot waste time over this sort of fantastic talk, Sherlock. If you can catch the man, catch him, and let me know when you have done it. “

分词（Tokenization）

北京大学 PEKING UNIVERSITY

- 句子切分(断句)（Sentence Tokenization）
  - 将文本序列分割为独立的句子
  - 例1："I cannot waste time over this sort of fantastic talk, Sherlock. If you can catch the man, catch him, and let me know when you have done it. “

分词（Tokenization）

北京大学 PEKING UNIVERSITY

- **句子切分(断句)（Sentence Tokenization）**
  - 将文本序列分割为独立的句子
  - 例1："I cannot waste time over this sort of fantastic talk, Sherlock. If you can catch the man, catch him, and let me know when you have done it. "

分词（Tokenization）

北京大学 PEKING UNIVERSITY

- **句子切分(断句)（Sentence Tokenization）**
  - 将文本序列分割为独立的句子
  - 例2：“I cannot waste time over this sort of fantastic talk, Mr. Holmes. If you can catch the man, catch him, and let me know when you have done it.“

分词（Tokenization）

北京大学 PEKING UNIVERSITY

- **句子切分(断句)（Sentence Tokenization）**
  - 将文本序列分割为独立的句子
  - 例2："I cannot waste time over this sort of fantastic talk, Mr. Holmes. If you can catch the man, catch him, and let me know when you have done it.“

分词（Tokenization）

北京大学 PEKING UNIVERSITY

- **句子切分(断句)（Sentence Tokenization）**
  - 将文本序列分割为独立的句子
  - 例2：“I cannot waste time over this sort of fantastic talk, Mr. Holmes. If you can catch the man, catch him, and let me know when you have done it.“

分词（Tokenization）

北京大学 PEKING UNIVERSITY

- 句子切分(断句)（Sentence Tokenization）
  - 将文本序列分割为独立的句子
  - 例2："I cannot waste time over this sort of fantastic talk, Mr. Holmes. If you can catch the man, catch him, and let me know when you have done it.“

分词（Tokenization）

- 句子切分(断句)（Sentence Tokenization）
  - 将文本序列分割为独立的句子
  - 例3： "I cannot waste time over this sort of fantastic talk, Mr. Holmes," he said. "If you can catch the man, catch him, and let me know when you have done it."

分词（Tokenization）

北京大学 PEKING UNIVERSITY

- 句子切分(断句)（Sentence Tokenization）
  - 将文本序列分割为独立的句子
  - 例3："I cannot waste time over this sort of fantastic talk, Mr. Holmes," he said. "If you can catch the man, catch him, and let me know when you have done it."

分词（Tokenization）

北京大学 PEKING UNIVERSITY

- 将文本序列分割为独立的“单词”
- 例1： "Whatever remains, however improbable, must be the truth.“
  - ✓ ["Whatever", "remains,", "however", "improbable,", "must", "be", "the", "truth."]
  - ✓ ["Whatever", "remains", "however", "improbable", "must", "be", "the", "truth"]

分词（Tokenization）

- 例2：“Just before nine o'clock Sherlock Holmes stepped briskly into the room.“
- 例3：“He was dressed in a sombre yet rich style, in black frock-coat, shining hat, neat brown gaiters, and well-cut pearl-grey trousers. “

中文分词

- 中文在词与词之间没有任何空格之类的显示标志指示词的边界。因此，中文分词是很多自然语言处理系统中的基础模块和首要环节。

■ **基于字符串匹配的分词方法**

✓ 按照一定的策略将待匹配的字符串和一个已建立好的“充分大的”词典中的词进行匹配，若找到某个词条，则说明匹配成功，识别了该词。

■ **基于机器学习的分词方法**

✓ CRF(条件随机场)、HMM(隐马尔可夫)、SVM、深度学习(BiLSTM-CRF)等。

✓ 例：CRF：对汉字进行标注训练，不仅考虑了词语出现的频率，还考虑上下文。

■ **常用的中文分词工具包：**

✓ jieba分词、SnowNLP、THULAC、NLPIR 等。

中文分词

```
import jieba
u="我来到中国北京大学"
#全模式
test1 = jieba.cut(u, cut_all=True)
print("全模式: " + "| ".join(test1))
#精确模式
test2 = jieba.cut(u, cut_all=False)
print("精确模式: " + "| ".join(test2))
#搜索引擎模式
test3= jieba.cut_for_search(u)
print("搜索引擎模式:" + "| ".join(test3))
```

例1：“我来到中国北京大学"

```
全模式: 我| 来到| 中国| 北京| 北京大学| 大学
精确模式: 我| 来到| 中国| 北京大学
搜索引擎模式:我| 来到| 中国| 北京| 大学| 北京大学
```

例2：“北大是所好大学"

```
全模式: 北大| 是| 所| 好| 大学
精确模式: 北大| 是| 所| 好| 大学
搜索引擎模式:北大| 是| 所| 好| 大学
```

Byte Pair Encoding (BPE)

北京大学 PEKING UNIVERSITY

- 算法
  - 数据预处理：收集大量文本数据，通常来自多种来源
  - 构建初始词汇表：将文本拆分为单个字符，形成初始的词汇表
  - 统计字节对频率：计算所有相邻字符对的频率，找出最常见的字节对
  - 合并最频繁的字节对：将这些字节对合并成一个新的单元，更新词汇表
  - 重复：重复上述步骤，直到达到预定的词汇表大小或没有更多的字节对可以合并
- 优点：
  - 处理稀有词
  - 减少词汇表大小
  - 对错别字的处理
  - 多语言支持

自然语言处理

北京大学 PEKING UNIVERSITY

- 分词
- 独热表示与词袋模型
- 分布式词表示
- 递归神经网络

词表示（Word Representation）

北京大学 PEKING UNIVERSITY

- **独热表示**(one-hot representation)

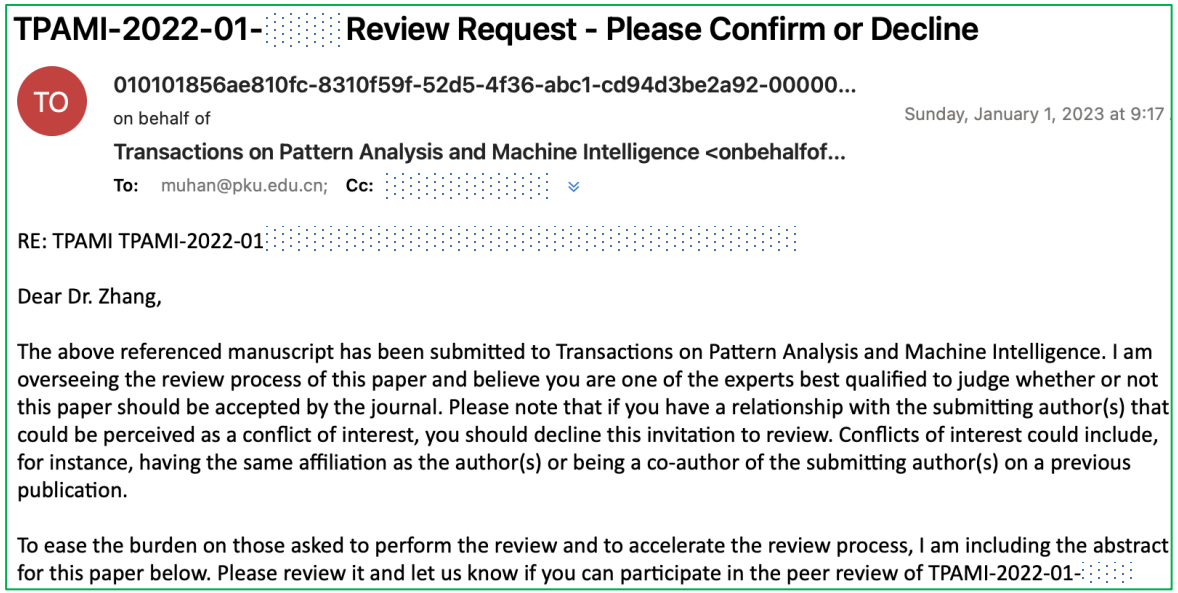
"He wrote a book."

he: [1, 0, 0, 0]
wrote: [0, 1, 0, 0]
a: [0, 0, 1, 0]
book: [0, 0, 0, 1]

回顾——垃圾邮件分类

北京大学 PEKING UNIVERSITY

输入: x = 邮件



TPAMI-2022-01-[redacted] Review Request - Please Confirm or Decline

TO

010101856ae810fc-8310f59f-52d5-4f36-abc1-cd94d3be2a92-00000...

on behalf of

Sunday, January 1, 2023 at 9:17

Transactions on Pattern Analysis and Machine Intelligence <onbehalfof...

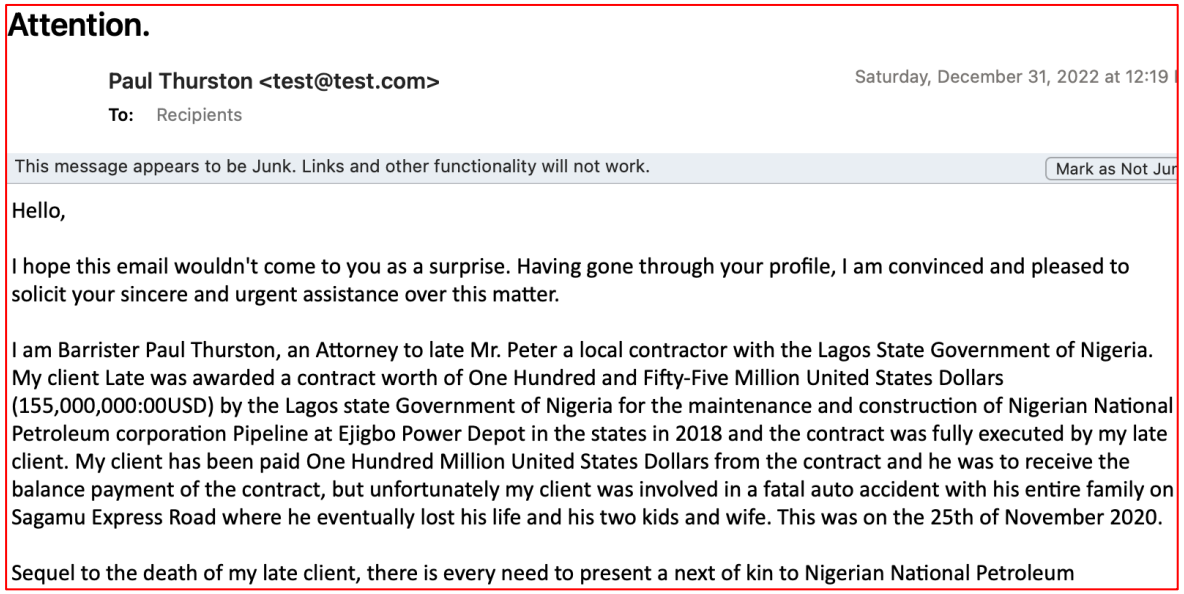
To: muhan@pku.edu.cn; Cc: [redacted]

RE: TPAMI TPAMI-2022-01[redacted]

Dear Dr. Zhang,

The above referenced manuscript has been submitted to Transactions on Pattern Analysis and Machine Intelligence. I am overseeing the review process of this paper and believe you are one of the experts best qualified to judge whether or not this paper should be accepted by the journal. Please note that if you have a relationship with the submitting author(s) that could be perceived as a conflict of interest, you should decline this invitation to review. Conflicts of interest could include, for instance, having the same affiliation as the author(s) or being a co-author of the submitting author(s) on a previous publication.

To ease the burden on those asked to perform the review and to accelerate the review process, I am including the abstract for this paper below. Please review it and let us know if you can participate in the peer review of TPAMI-2022-01-[redacted]



Attention.

Paul Thurston <test@test.com>

Saturday, December 31, 2022 at 12:19

To: Recipients

This message appears to be Junk. Links and other functionality will not work. Mark as Not Jur

Hello,

I hope this email wouldn't come to you as a surprise. Having gone through your profile, I am convinced and pleased to solicit your sincere and urgent assistance over this matter.

I am Barrister Paul Thurston, an Attorney to late Mr. Peter a local contractor with the Lagos State Government of Nigeria. My client Late was awarded a contract worth of One Hundred and Fifty-Five Million United States Dollars (155,000,000:00USD) by the Lagos state Government of Nigeria for the maintenance and construction of Nigerian National Petroleum corporation Pipeline at Ejigbo Power Depot in the states in 2018 and the contract was fully executed by my late client. My client has been paid One Hundred Million United States Dollars from the contract and he was to receive the balance payment of the contract, but unfortunately my client was involved in a fatal auto accident with his entire family on Sagamu Express Road where he eventually lost his life and his two kids and wife. This was on the 25th of November 2020.

Sequel to the death of my late client, there is every need to present a next of kin to Nigerian National Petroleum

输出: $y \in$ {正常邮件，垃圾邮件}

目标：训练模型 f，使得 $f(x) = y$

回顾——垃圾邮件分类

训练数据 (Training Data)

(email1, not_spam)
(email2, spam)
(email3, spam)
……
(email97, not_spam)
(email98, spam)
(email99, not_spam)
(email100, spam)

特征提取
(Feature extraction)

x_i: 特征向量 (feature vector)

y_i : 标签 (label)

(x_1, y_1)
(x_2, y_2)
$\cdots$
(x_{100}, y_{100})

x_i 词频向量

attention:	1
million:	2
dollars:	1
prize:	0
payment:	3
government:	1
…	

$y_i = \{-1, 1\}$

模型训练
(Model Training)

Learning a function f

$y_i \approx f(x_i) \quad \forall i$

词袋模型（Bag-of-Words Model）

- **词袋模型**（BoW）：不考虑文本中词与词之间的上下文关系，仅仅只考虑所有词的权重，而权重与词在文本中出现的频率有关。以词的统计直方图作为该文本（文档/句子）的特征

- 原始BoW模型的两要素：
  - ✓ 构建一个包含所有已知单词的字典
  - ✓ 统计文本中单词出现的频率
- 改变字典构建方法和单词出现情况统计策略可得到升级版本的BoW

词袋模型（Bag-of-Words Model）

北京大学 PEKING UNIVERSITY

- **词袋模型**（BoW）构建实例

步骤1：数据收集

It was the best of times,
it was the worst of times,
it was the age of wisdom,
it was the age of foolishness,

From “A Tale of Two Cities”

- 整个语料库包括4行文本
- 每行是一个独立的”文档”

数据参考：https://machinelearningmastery.com/gentle-introduction-bag-words-model/

词袋模型（Bag-of-Words Model）

北京大学 PEKING UNIVERSITY

- **词袋模型**（BoW）构建实例

步骤2：字典构建

It was the best of times,
it was the worst of times,
it was the age of wisdom,
it was the age of foolishness,

字典
“it”
“was”
“the”
“best”
“of”
“times”
“worst”
“age”
“wisdom”
“foolishness”

- 把所有单词列出
- 去重，不考虑大小写和标点符号
- 字典包含10个词
- 整个语料库包括24个词

词袋模型（Bag-of-Words Model）

北京大学 PEKING UNIVERSITY

- **词袋模型**（BoW）构建实例

步骤3：生成文本特征向量

① It was the best of times,
② it was the worst of times,
③ it was the age of wisdom,
④ it was the age of foolishness,

字典	①	②	③	④
“it”	1	1	1	1
“was”	1	1	1	1
“the”	1	1	1	1
“best”	1	0	0	0
“of”	1	1	1	1
“times”	1	1	0	0
“worst”	0	1	0	0
“age”	0	0	1	1
“wisdom”	0	0	0	0
“foolishness”	0	0	1	1

提问：输入以下新文本，其特征向量是?

This is the best of times but also the worst of times.

[0, 0, 2, 1, 2, 2, 1, 0, 0, 0]

BoW模型的缺点：字典可能极大；文本向量稀疏；关键词的重要性未体现.....

独热表示困难

北京大学 PEKING UNIVERSITY

✓ **问题1**：字典若很大，则词向量很长很稀疏，例如字典中有50000个单词

"He wrote a book."

he: [1, 0, 0, 0, 0, 0, 0, 0, 0.....]
wrote: [0, 1, 0, 0, 0, 0, 0, 0, 0.....]
a: [0, 0, 1, 0, 0, 0, 0, 0, 0.....]
book: [0, 0, 0, 1, 0, 0, 0, 0, 0.....]

50000

独热表示困难

北京大学 PEKING UNIVERSITY

✓ **问题2**：仅将词符号化，不包含任何语义信息，没有考虑词间的相关性

"He wrote a book."
"He authored a novel."

wrote [0, 1, 0, 0, 0, 0, 0, 0, 0]
authored [0, 0, 0, 0, 1, 0, 0, 0, 0] ≠

book [0, 0, 0, 0, 0, 0, 1, 0, 0]
novel [0, 0, 0, 0, 0, 0, 0, 0, 1] ≠

自然语言处理

北京大学 PEKING UNIVERSITY

- 分词
- 独热表示与词袋模型
- 分布式词表示
- 递归神经网络

词表示（Word Representation）

北京大学 PEKING UNIVERSITY

- **分布式表示** (distributed representation)
  - ✓ **理论基础**：词的语义由其上下文决定! 上下文相似的词，其语义也相似。

"He wrote a book."

he	[-0.34, -0.08, 0.02, -0.18, 0.22, ...]
wrote	[-0.27, 0.40, 0.00, -0.65, -0.15, ...]
a	[-0.12, -0.25, 0.29, -0.09, 0.40, ...]
book	[-0.23, -0.16, -0.05, -0.57, ...]

词表示（Word Representation）

北京大学 PEKING UNIVERSITY

- **分布式表示** (distributed representation)
  - ✓ **理论基础**：词的语义由其上下文决定! 上下文相似的词，其语义也相似。

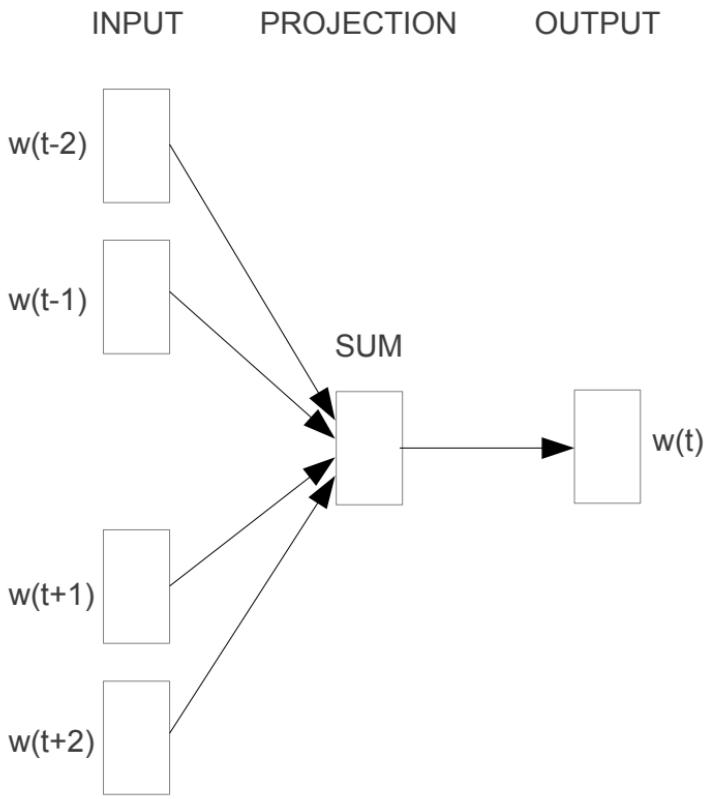
for	*	he	ate
for	**breakfast**	he	ate
for	**lunch**	he	ate
for	**dinner**	he	ate

词表示（Word Representation）

北京大学 PEKING UNIVERSITY

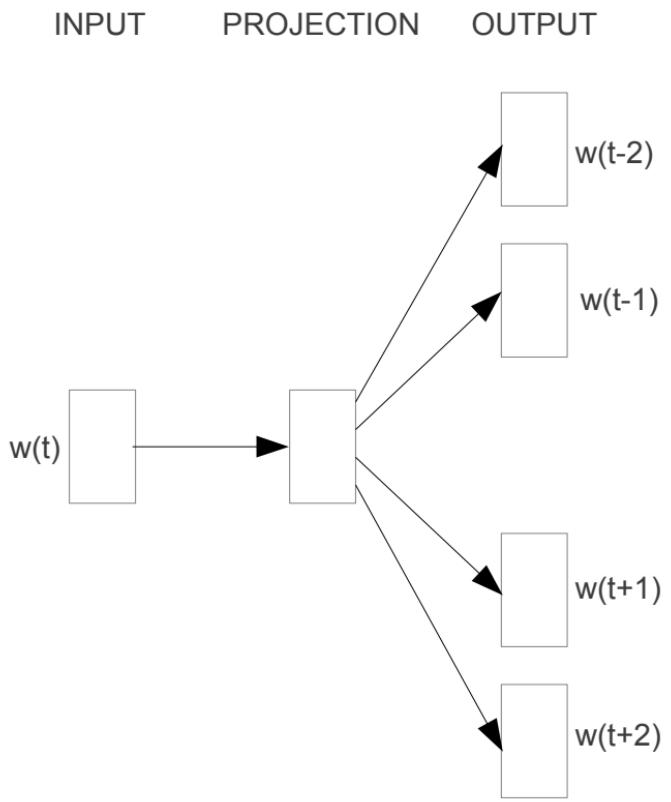
- **分布式表示** (distributed representation)
  - ✓ 基于矩阵的分布表示
  - ✓ 基于聚类的分布表示
  - ✓ 基于神经网络的分布表示
  - **核心思想：**
    - ➢ 选择一种方式描述上下文
    - ➢ 选择一种模型刻画目标词与其上下文之间的关系

word2vec



CBOW

（Continuous Bag Of Word）



Skip-gram（略）

Efficient Estimation of Word Representations in Vector Space. ICLR 2013

word2vec

- **CBOW模型**

① 对每个词随机初始化为一个$1 \times N$向量

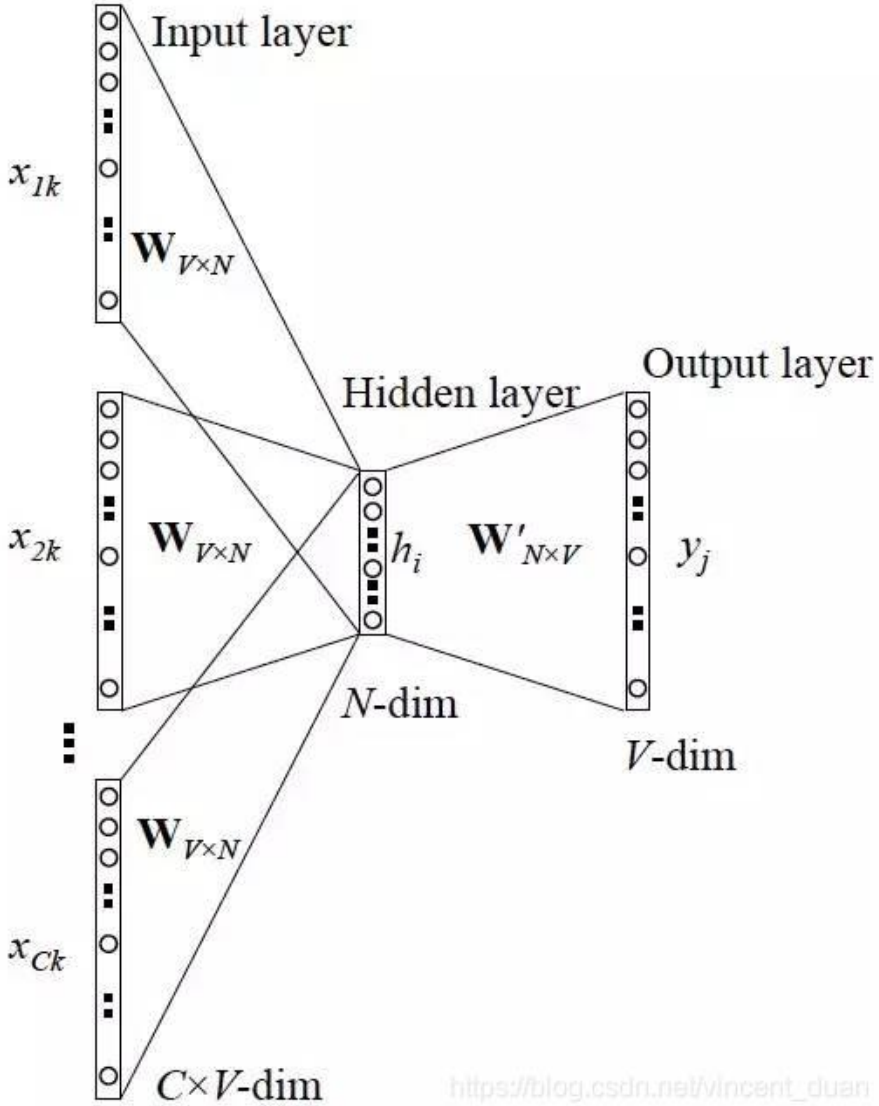
② 把当前词的上下文词语的向量各乘同一个矩阵$\boldsymbol{W}$（周围词向量矩阵）得到各自的$1 \times N$向量

③ 将这些$1 \times N$向量取平均为一个$1 \times N$向量

④ 将这个$1 \times N$向量乘矩阵$\boldsymbol{W}'$ (中心词向量矩阵)，变成一个$1 \times V$向量

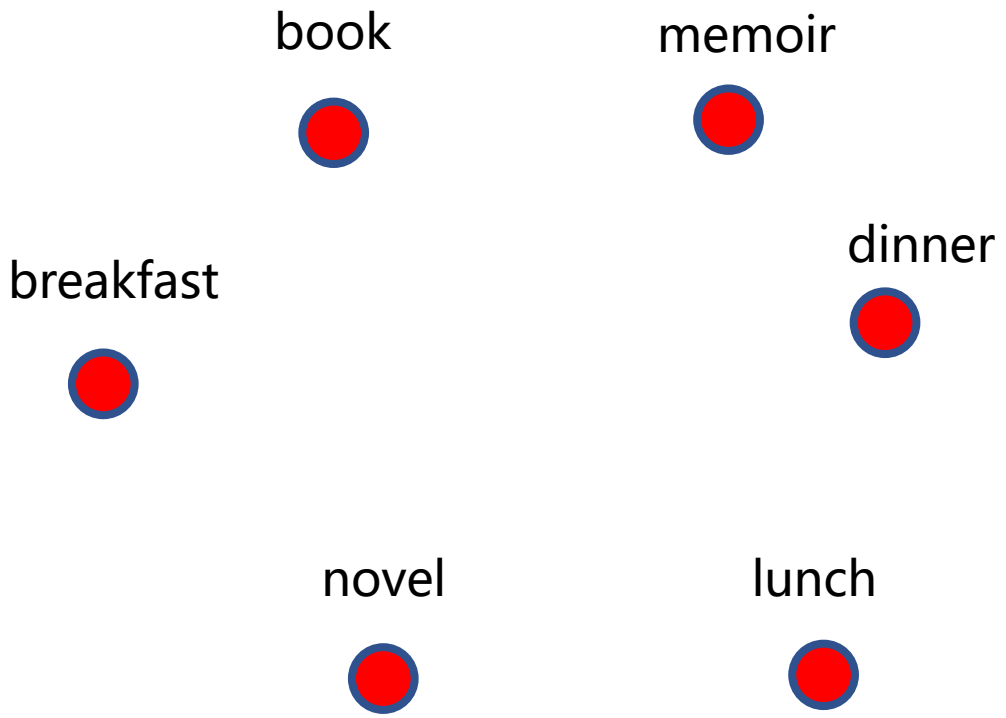
⑤ 做Softmax分类，与真实标签$1 \times V$向量计算交叉熵损失

⑥ 每次前向传播之后反向传播误差，调整矩阵$\boldsymbol{W}$和$\boldsymbol{W}'$的值



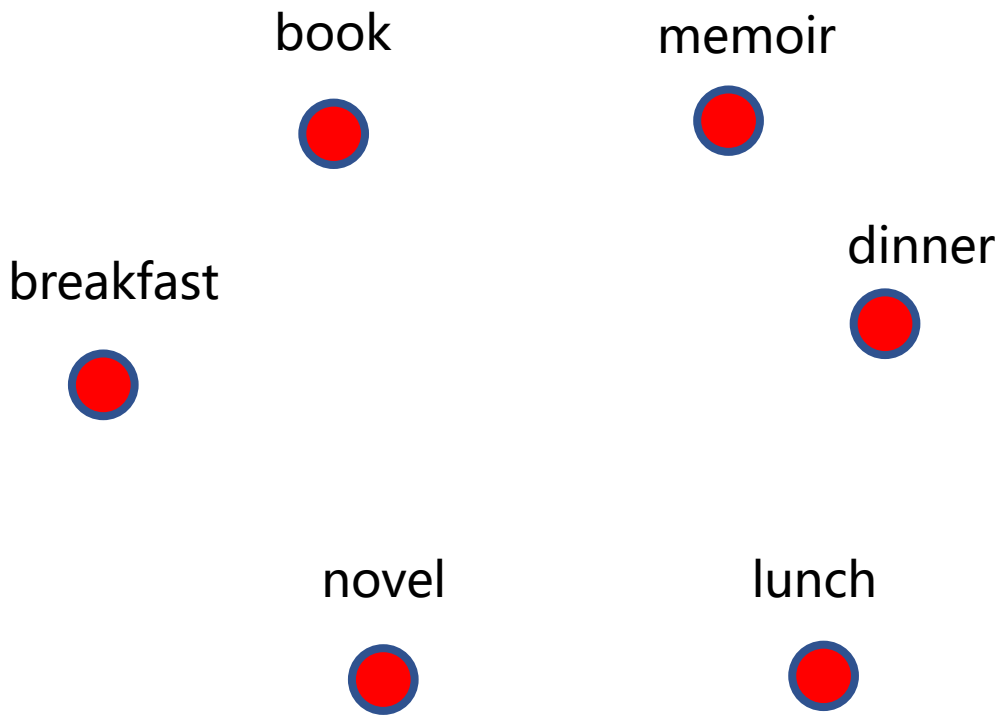
Efficient Estimation of Word Representations in Vector Space. ICLR 2013

word2vec



独特表示的词特征空间示例

word2vec



自监督学习后的word2vec词特征空间示例

word2vec

自监督学习后的word2vec词特征空间示例

word2vec——词向量示例

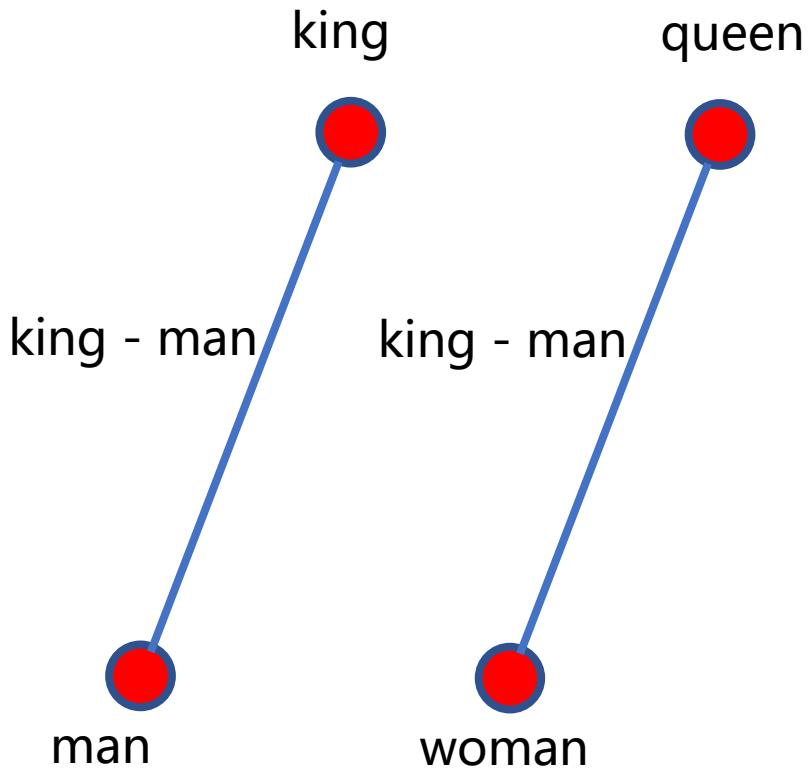
```
>>> words["city"]
array([ 0.231087, -0.238098,  0.584713, -0.524351,  0.40278 ,  0.148448,
        0.386096, -0.493994, -0.198922, -0.411161,  0.556962,  0.220978,
       -0.304637, -0.499713, -0.092555,  0.262613,  0.752704,  0.463667,
        0.054477,  0.155809, -0.195134, -0.009269,  0.378139, -0.651306,
       -0.029372, -0.563472,  0.024709,  0.366842, -0.476904, -0.42565 ,
       -0.094642, -0.052822,  0.124612,  0.296046, -0.244881,  0.195957,
        0.223666,  0.064116,  0.577874,  0.083096, -0.378262,  0.196044,
       -0.220993, -0.630213, -0.311214,  0.435611,  0.351486,  0.342794,
       -0.229961, -0.157521,  0.204315,  0.253944, -0.562277,  0.534482,
       -0.4158  ,  0.120161,  0.649395, -0.227012, -0.130488, -0.332326,
        0.691952, -0.400436,  0.410125,  0.026237, -0.408483,  0.188236,
        0.130957, -0.320686,  0.225932, -0.171665, -0.335107, -0.009982,
        0.680831, -0.023788, -0.165798,  0.345986, -0.232295,  0.021137,
        0.08515 , -0.24387 , -0.142469, -0.058325,  0.086046, -0.173068,
        0.198108,  0.009103,  0.381725,  0.095911,  0.317972, -0.10012 ,
        0.143178,  0.106724, -0.419844, -0.175785, -0.251805,  0.211927,
        0.411175,  0.317378,  0.450316, -0.252661])
```

```
>>> distance(words["book"],words["book"])
0
>>> distance(words["book"],words["breakfast"])
0.6351827719357863
>>> distance(words["book"],words["novel"])
0.3436623421719047
>>> distance(words["lunch"],words["breakfast"])
0.2006302059301045
```

```
>>> closest_words(words["book"])[:6]
['book', 'books', 'essay', 'memoir', 'essays', 'novella']
>>> closest_words(words["lunch"])[:6]
['lunch', 'dinner', 'lunches', 'snack', 'meal', 'brunch']
```

```
>>> closest_word(words["woman"] + words["king"] - words["man"])
'queen'
>>> closest_word(words["england"] + words["paris"] - words["france"])
'london'
```

```
>>> closest_word(words["japan"] + words["beijing"] - words["china"])
'tokyo'
>>> closest_word(words["hospital"] + words["teacher"] - words["school"])
'nurse'
>>> closest_word(words["hospital"] + words["student"] - words["school"])
'hospital'
```




```
>>> closest_words(words["hospital"] + words["student"] - words["school"])
['hospital', 'patient', 'icu', 'nurse', 'nurses', 'transplant', 'gyn', 'neurosurgery', 'inpatient', 'medical']
```

自然语言处理

北京大学 PEKING UNIVERSITY

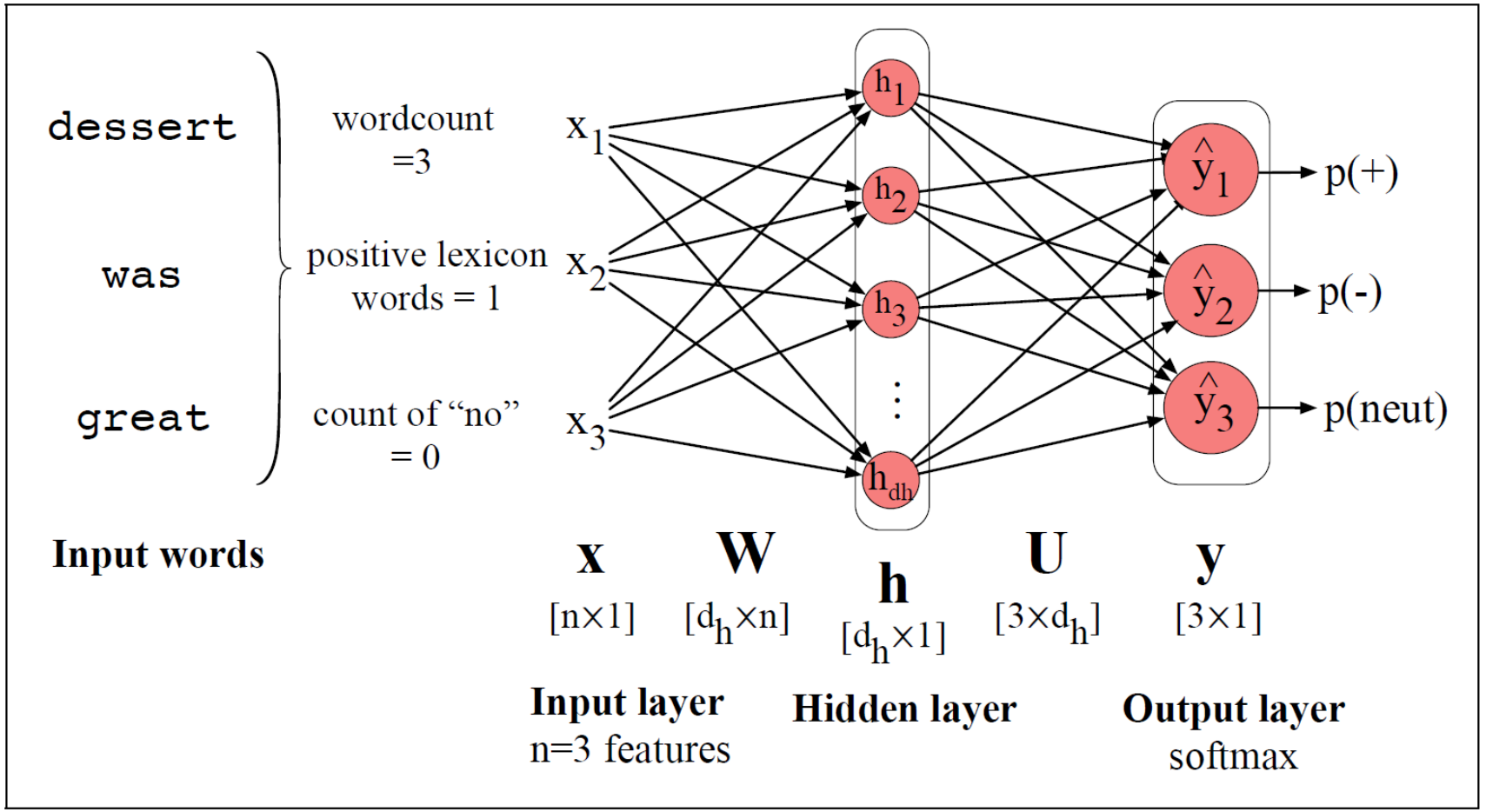
- 分词
- 独热表示与词袋模型
- 分布式词表示
- 递归神经网络

基于神经网络的自然语言处理方法

文本情感分类应用：基于人工设计文本特征

- $\mathbf{x}_i$是人工设计特征，可以是标量

$$\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N]$$
$$\mathbf{h} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$
$$\mathbf{z} = \mathbf{U}\mathbf{h}$$
$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{z})$$

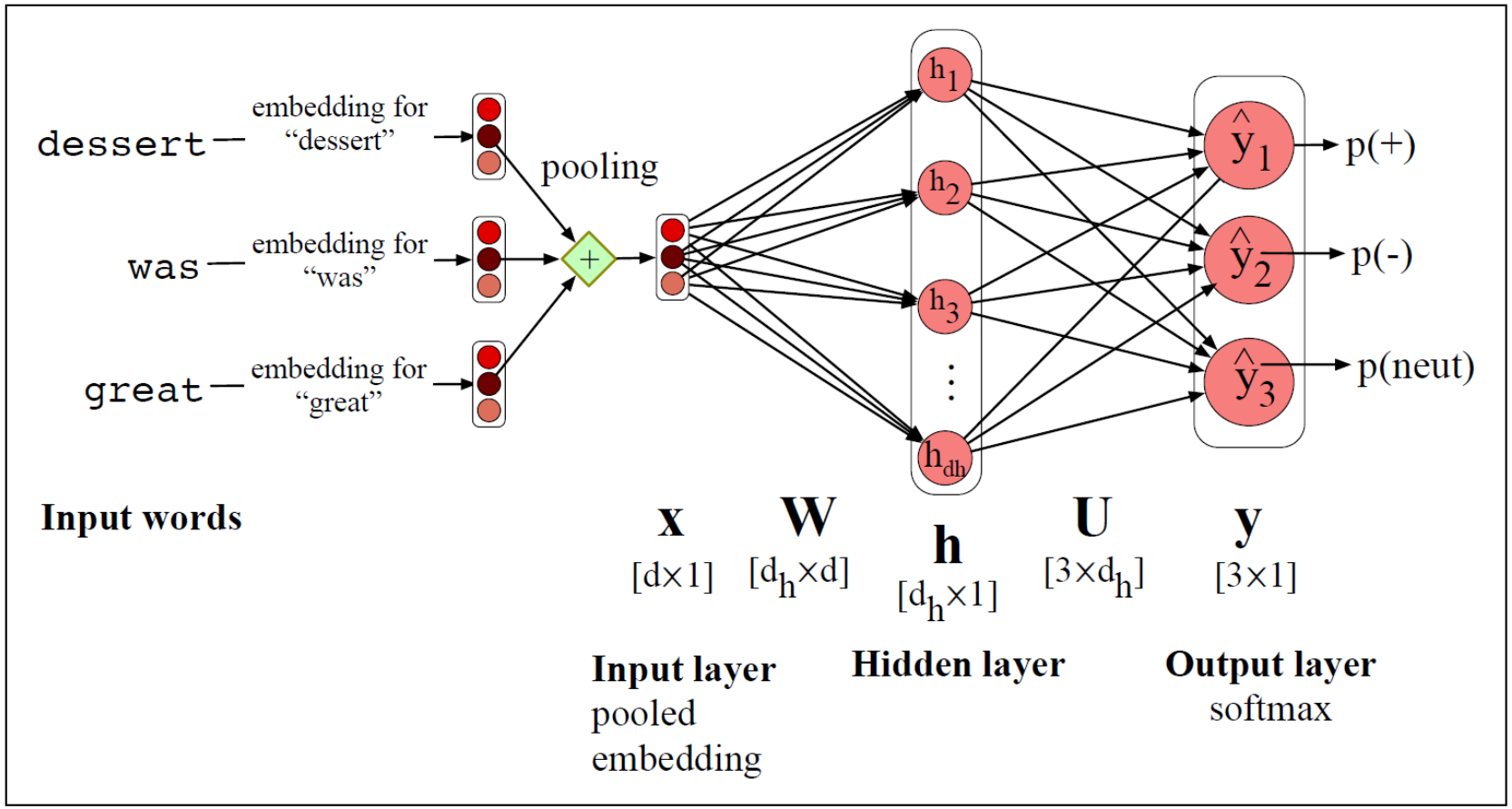


基于神经网络的自然语言处理方法

文本情感分类应用：基于word2vec/GloVe的文本特征

- 文本包含n个词w_i
- $\mathbf{e}(w_i)$: word2vec词嵌入

sum, avg, concat等均可



$$\mathbf{x} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{e}(w_i)$$

$$\mathbf{h} = \sigma(\mathbf{W}\mathbf{x}+\mathbf{b})$$

$$\mathbf{z} = \mathbf{U}\mathbf{h}$$

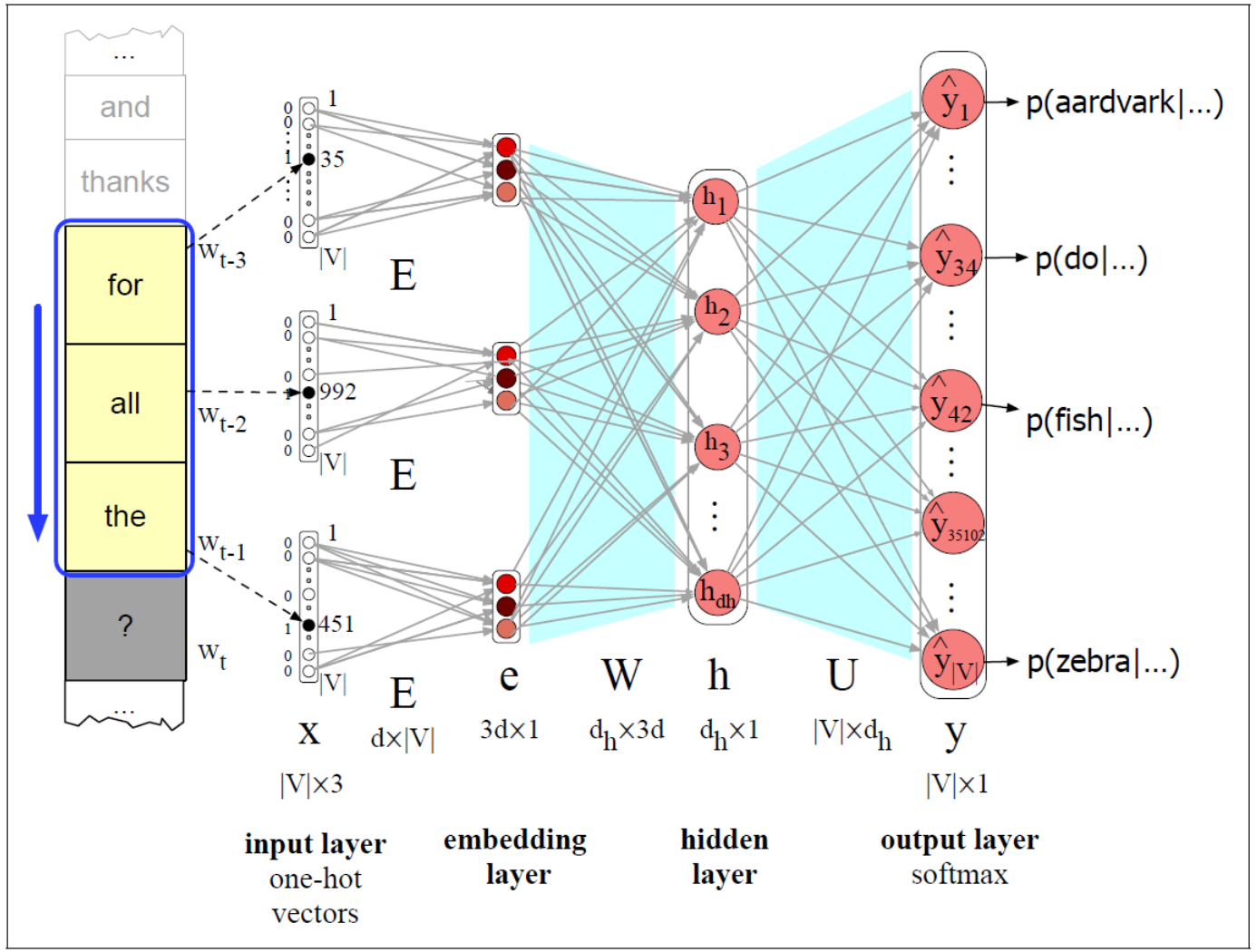
$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{z})$$

基于神经网络的自然语言处理方法

前馈神经语言模型：基于前词上下文预测接下来的词

- 输入独热向量/或嵌入向量
- 给定窗口大小(如3)

$$
\begin{aligned}
\mathbf{e} &= [\mathbf{E}\mathbf{x}_{t-3}; \mathbf{E}\mathbf{x}_{t-2}; \mathbf{E}\mathbf{x}_{t-1}] \\
\mathbf{h} &= \sigma(\mathbf{W}\mathbf{e} + \mathbf{b}) \\
\mathbf{z} &= \mathbf{U}\mathbf{h} \\
\hat{\mathbf{y}} &= \mathrm{softmax}(\mathbf{z})
\end{aligned}
$$

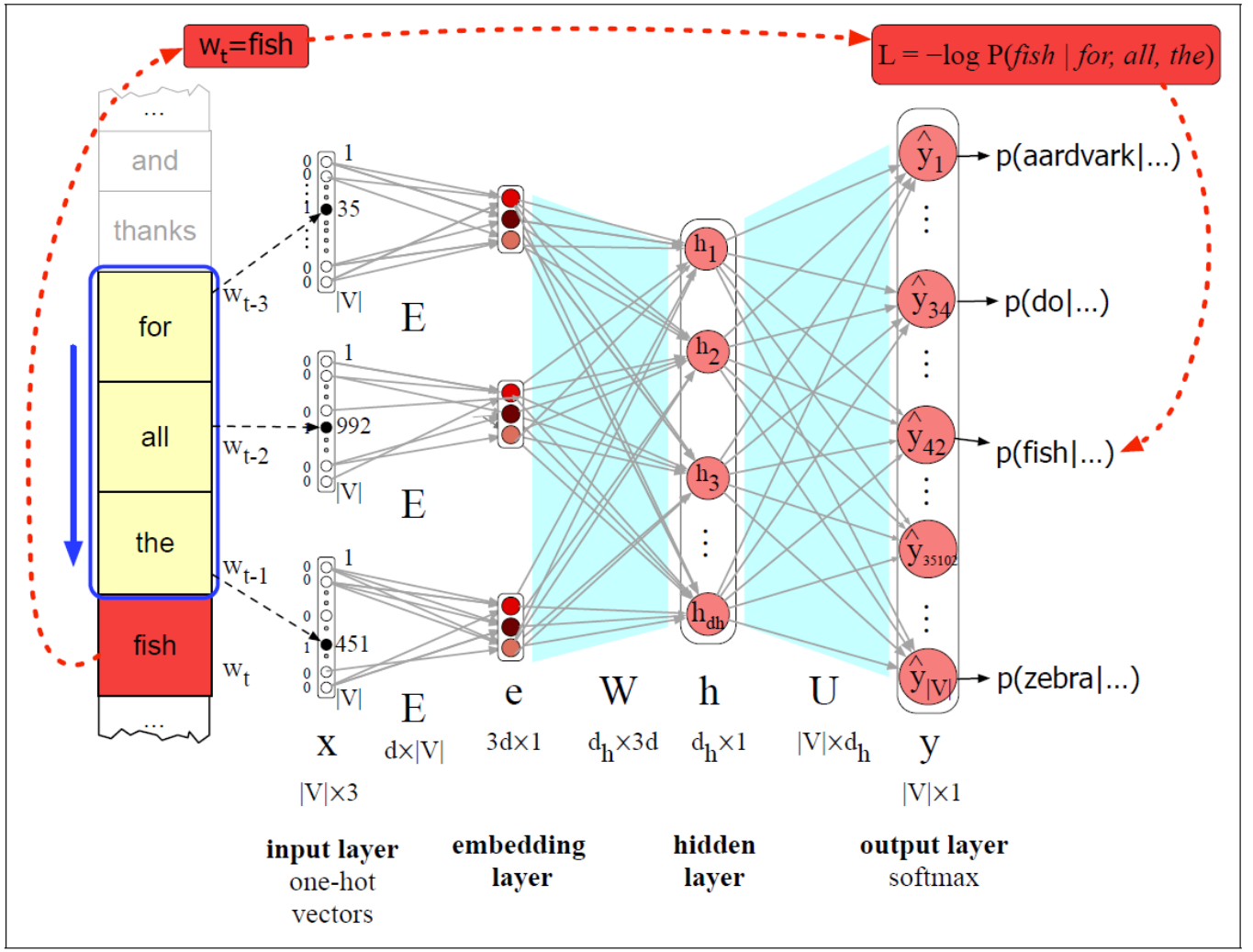


基于神经网络的自然语言处理方法

前馈神经语言模型：基于前词上下文预测接下来的词

- 输入独热向量/或嵌入向量
- 给定窗口大小(如3)

$$
\begin{aligned}
\mathbf{e} &= [\mathbf{E}\mathbf{x}_{t-3}; \mathbf{E}\mathbf{x}_{t-2}; \mathbf{E}\mathbf{x}_{t-1}] \\
\mathbf{h} &= \sigma(\mathbf{W}\mathbf{e} + \mathbf{b}) \\
\mathbf{z} &= \mathbf{U}\mathbf{h} \\
\hat{\mathbf{y}} &= \mathrm{softmax}(\mathbf{z})
\end{aligned}
$$

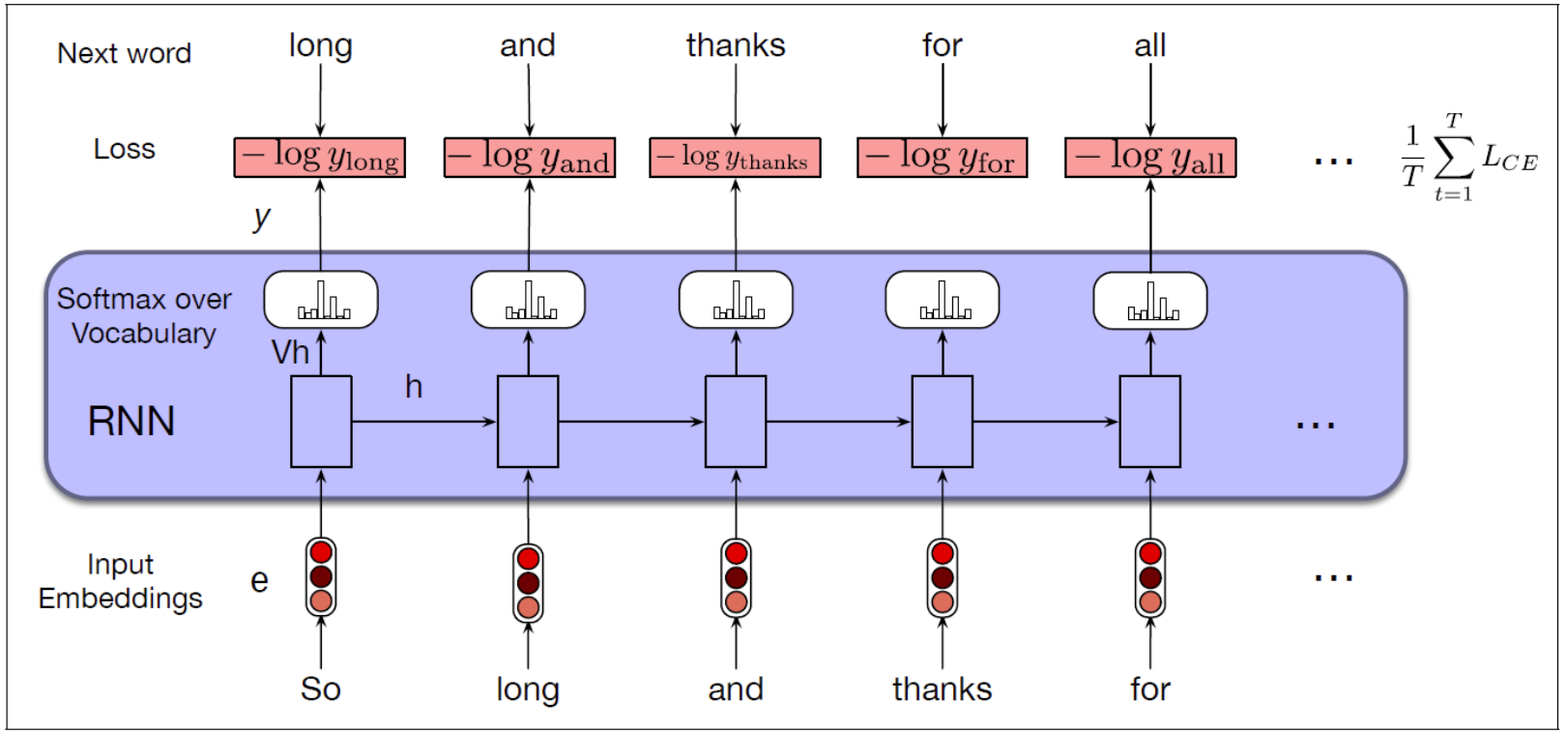


基于神经网络的自然语言处理方法

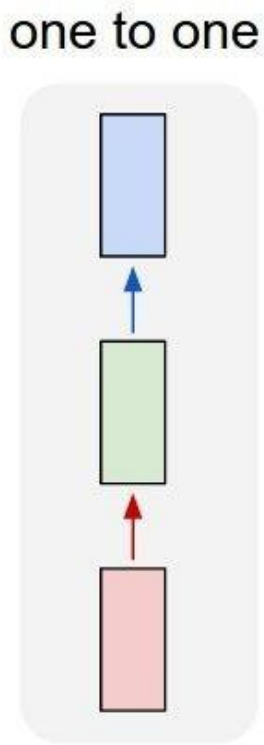
递归神经语言模型：基于前词上下文预测接下来的词

- 输入独热向量/或嵌入向量
- RNN递归生成
- 无窗口限制

$$\mathbf{e}_t = \mathbf{E}\mathbf{x}_t$$
$$\mathbf{h}_t = g(\mathbf{U}\mathbf{h}_{t-1} + \mathbf{W}\mathbf{e}_t)$$
$$\mathbf{y}_t = \text{softmax}(\mathbf{V}\mathbf{h}_t)$$

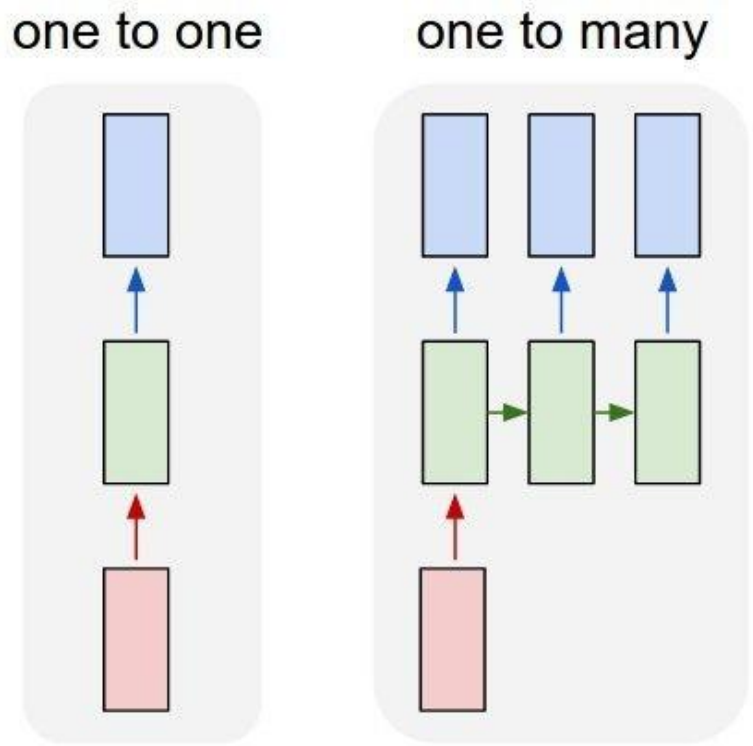


Recurrent Neural Networks



Vanilla Neural Networks

Recurrent Neural Networks



例如：图像描述
图像 -> 文字描述

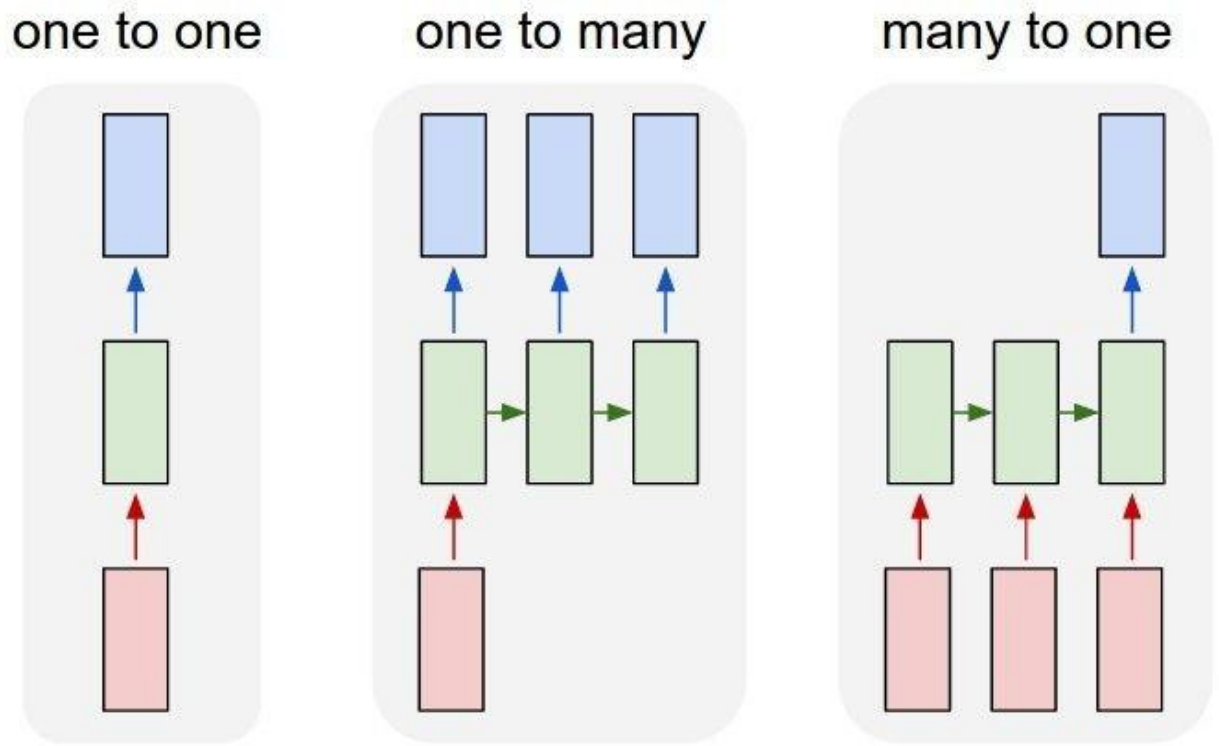
A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

Two people walking on the beach with surfboards

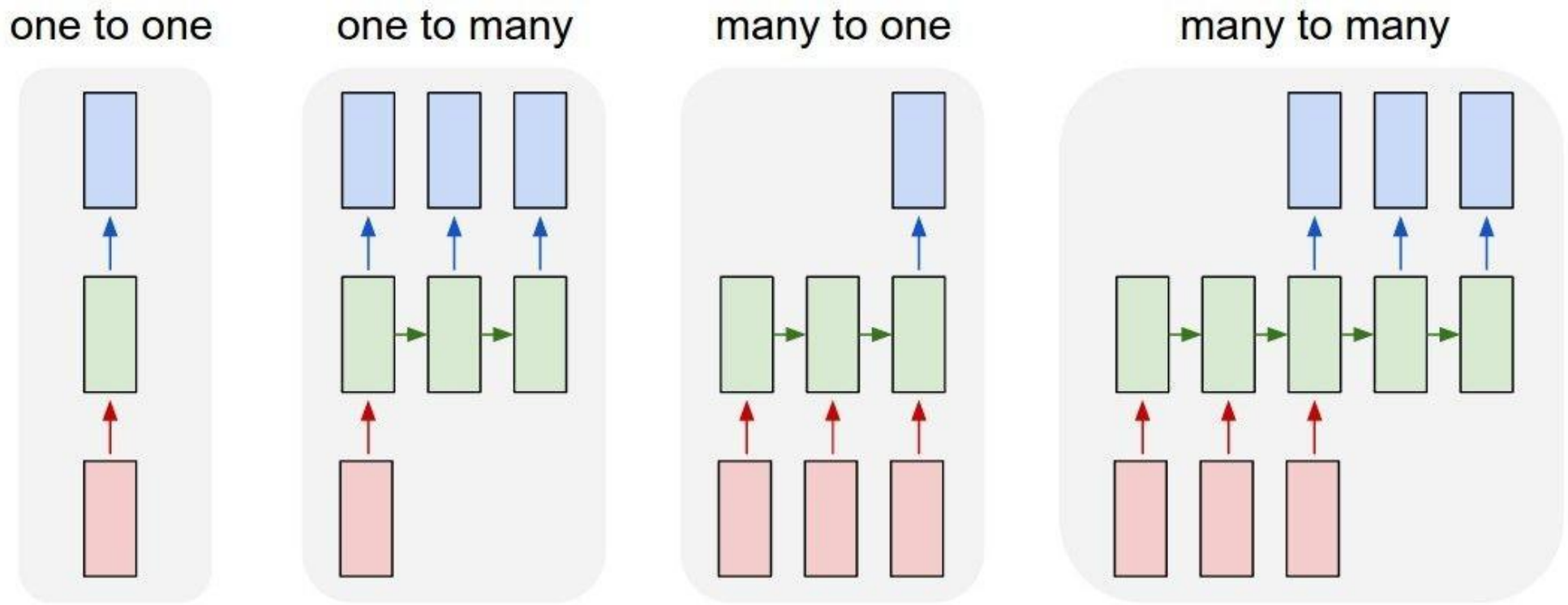
A tennis player in action on the court

Recurrent Neural Networks



动作预测
视频时间帧 -> 动作类别

Recurrent Neural Networks



例如：文本翻译
中文句子->英文句子

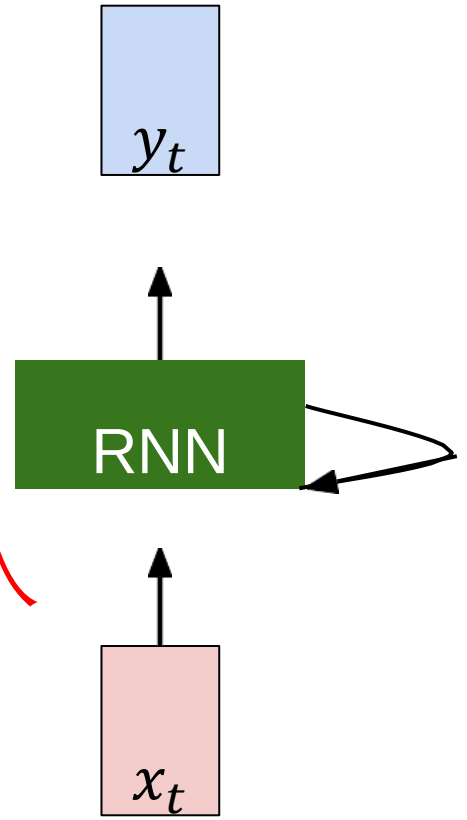
RNN隐状态更新

循环地使用RNN更新输入x每个时间步的隐状态 h_t ：

$$h_t = f_W(h_{t-1}, x_t)$$

新状态　　旧状态　　某一时间步的输入

更新状态的函数
（参数为W）



RNN输出预测

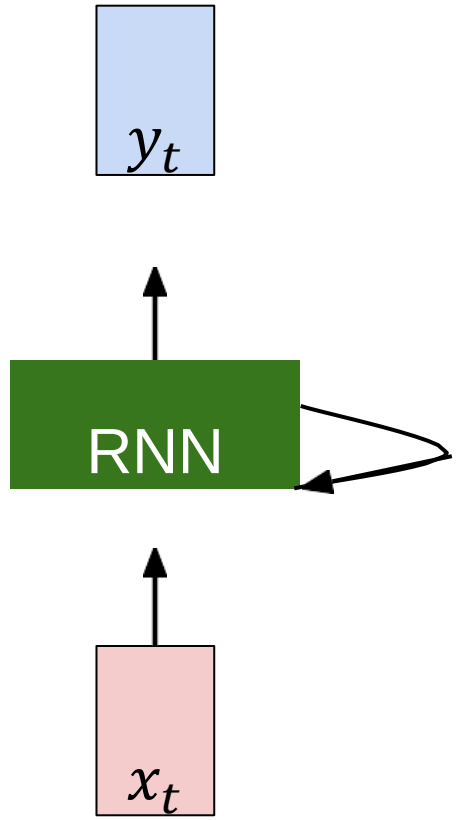
循环地使用RNN更新输入x每个时间步的预测值y_t：

$$y_t = f_{W_{hy}}(h_t)$$

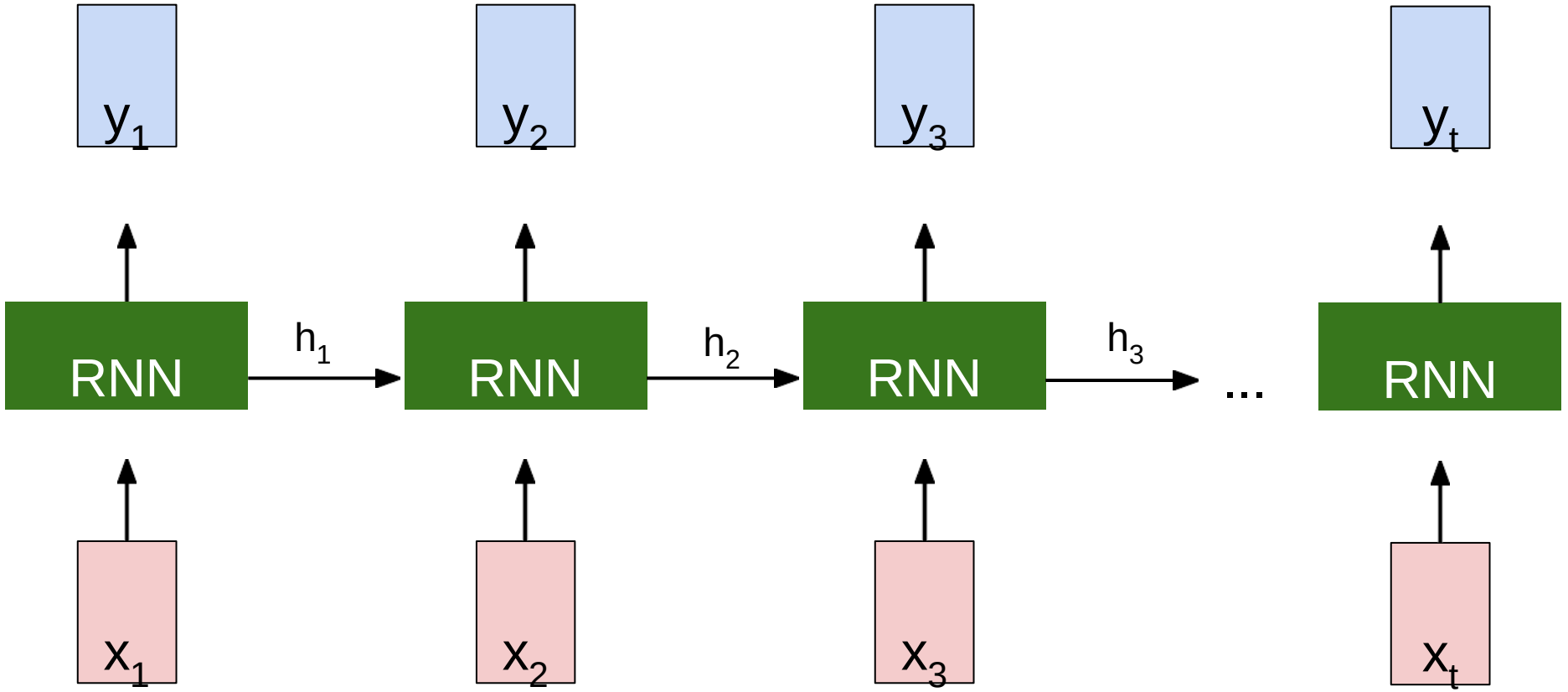
时刻t的预测

时刻t的隐状态

输出预测的函数
（参数为W_{hy}）

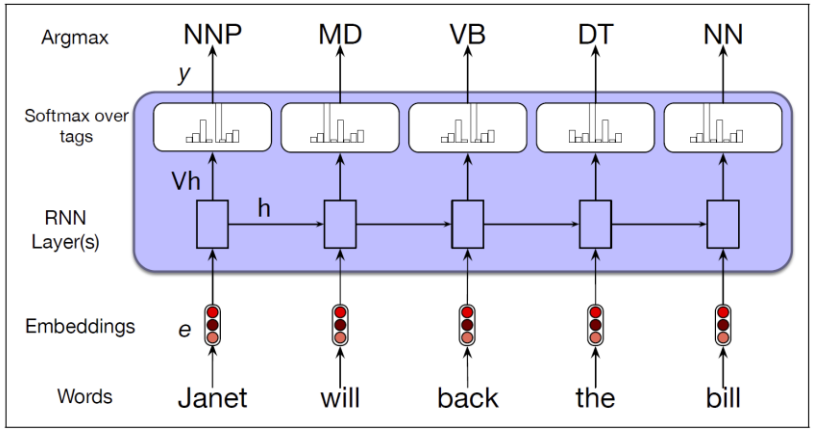


展开 RNN

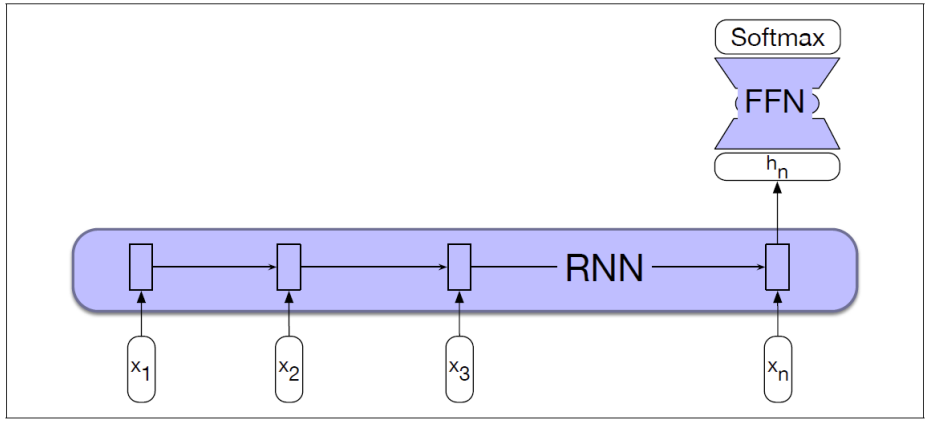


基于神经网络的自然语言处理方法

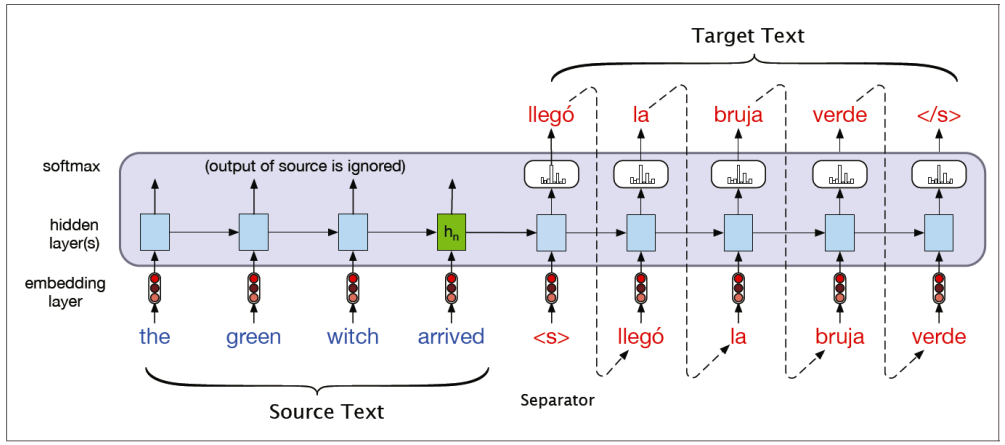
递归神经网络RNN在NLP其他任务中的应用：



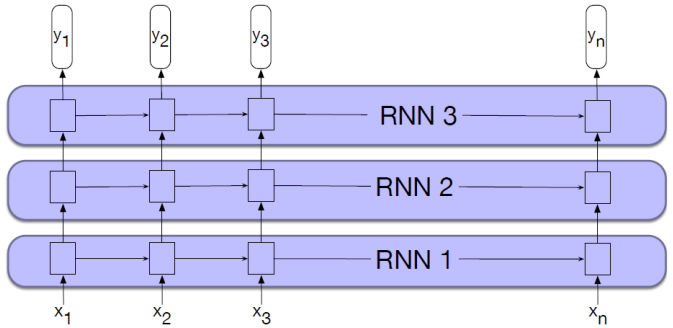
序列标注：例如词性标注



序列分类：例如情感分类、垃圾邮件分类等



文本生成：例如机器翻译、文本摘要、问答等



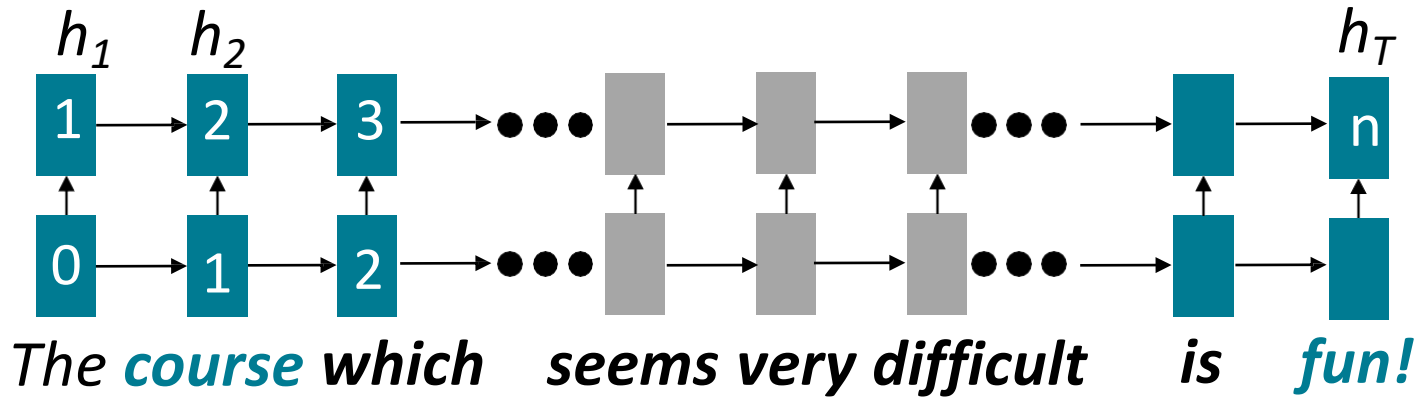
改进：多层堆叠RNN等...

RNN的优缺点

- RNN的优点:
  - 能够处理任意长度数据
  - 输入变长，模型大小不变
- RNN的缺点:
  - 未来状态的计算依赖于过去的状态（无法并行计算）
  - 很难处理长距离依赖（由于梯度衰减问题）

RNN的优缺点

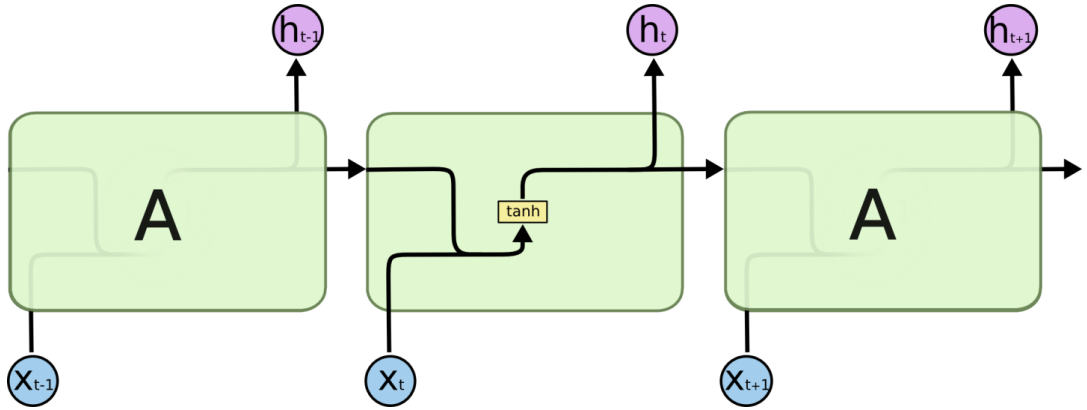
上面一列是隐状态，下面一列是输入序列。



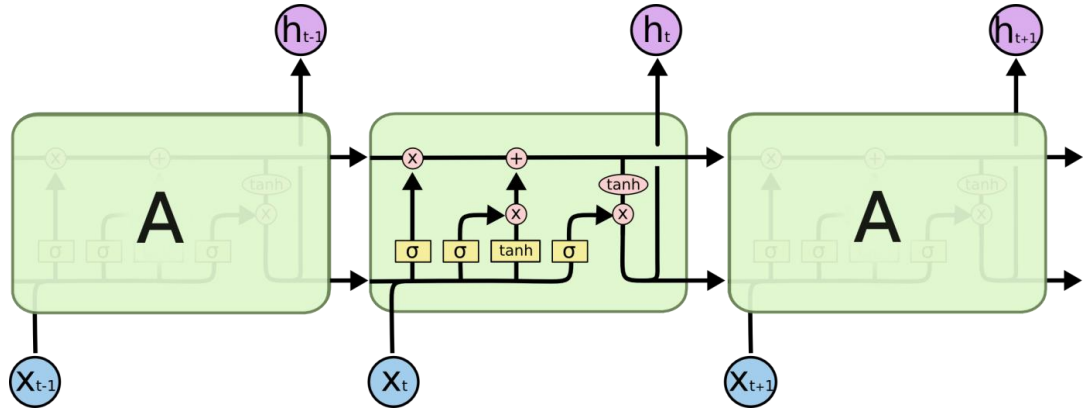
主语(course)和形容词(fun)相隔了许多词，隐状态更新太多次已经遗忘了主语。RNN很难捕获长距离的依赖！

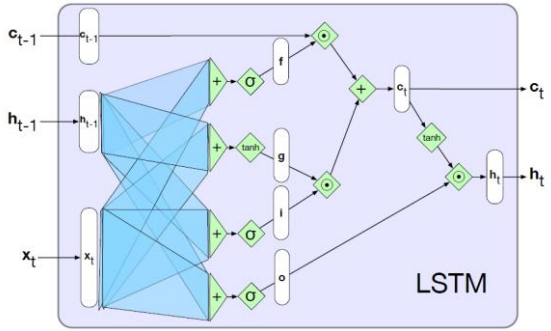
改进：LSTM

RNNs



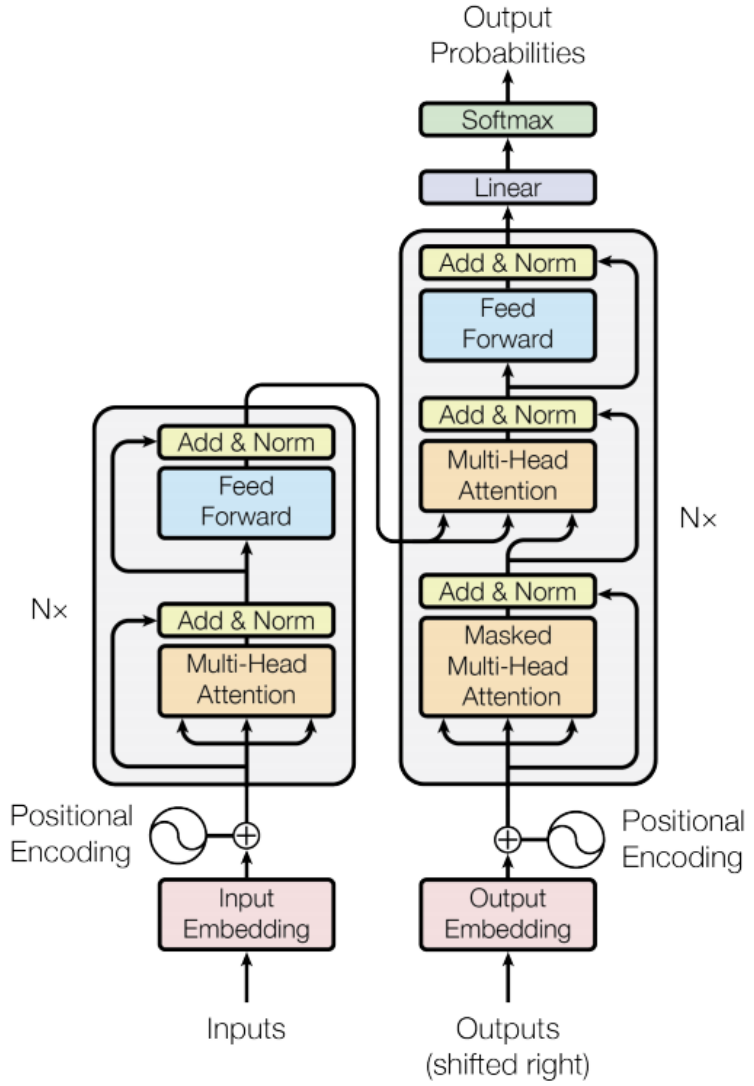
LSTM



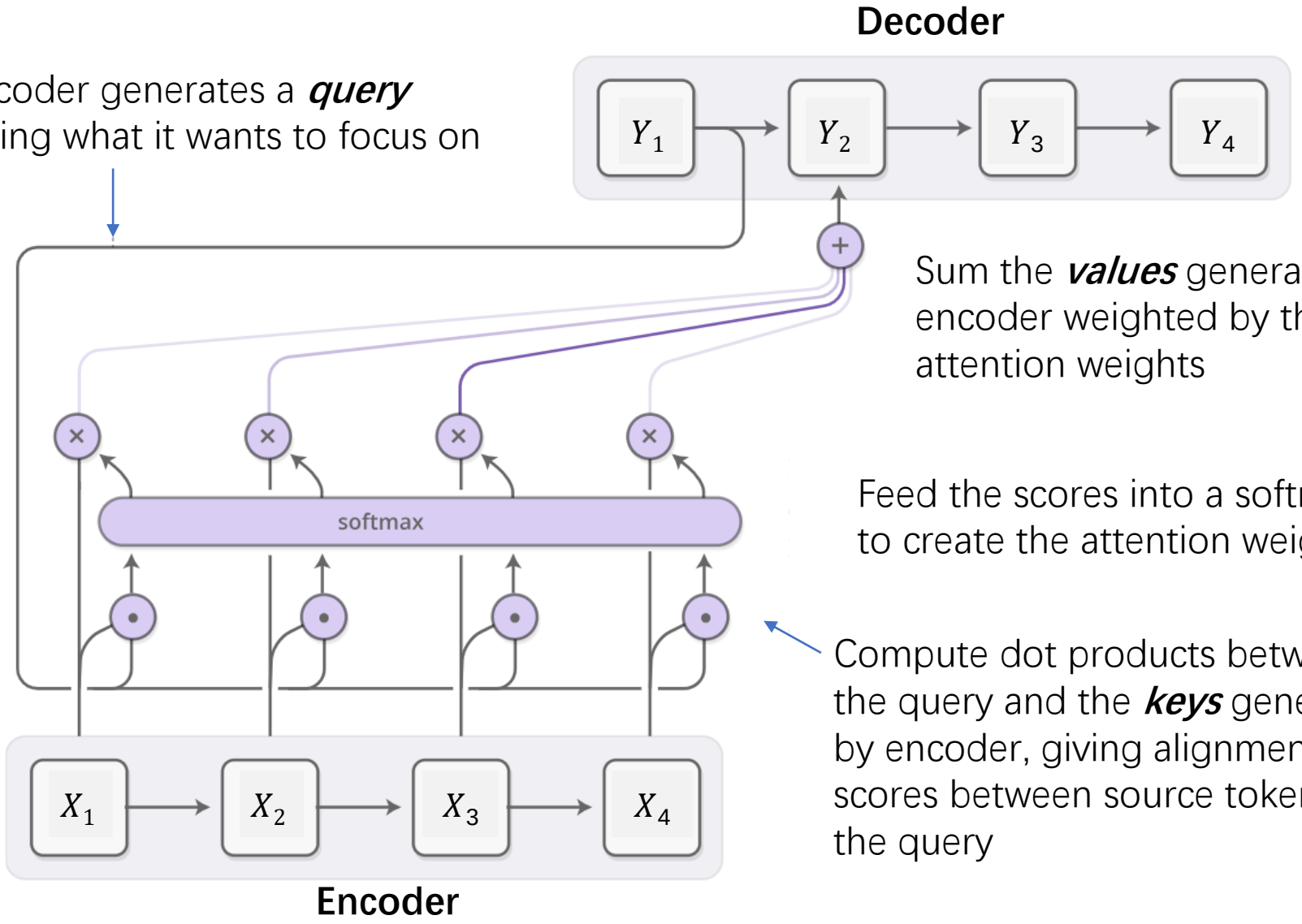


S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Computation 9 (8), pp. 1735–1780, 1997

改进：注意力机制与Transformer



Key-Value-Query attention model



The decoder generates a ***query*** describing what it wants to focus on

Sum the ***values*** generated by encoder weighted by the attention weights

Feed the scores into a softmax to create the attention weights

Compute dot products between the query and the ***keys*** generated by encoder, giving alignment scores between source tokens and the query

Image source

谢谢

北京大学
PEKING UNIVERSITY

Self-attention layer

- Query vectors: $Q = XW_Q$
- Key vectors: $K = XW_K$
- Value vectors: $V = XW_V$
- Similarities: *scaled dot-product attention*

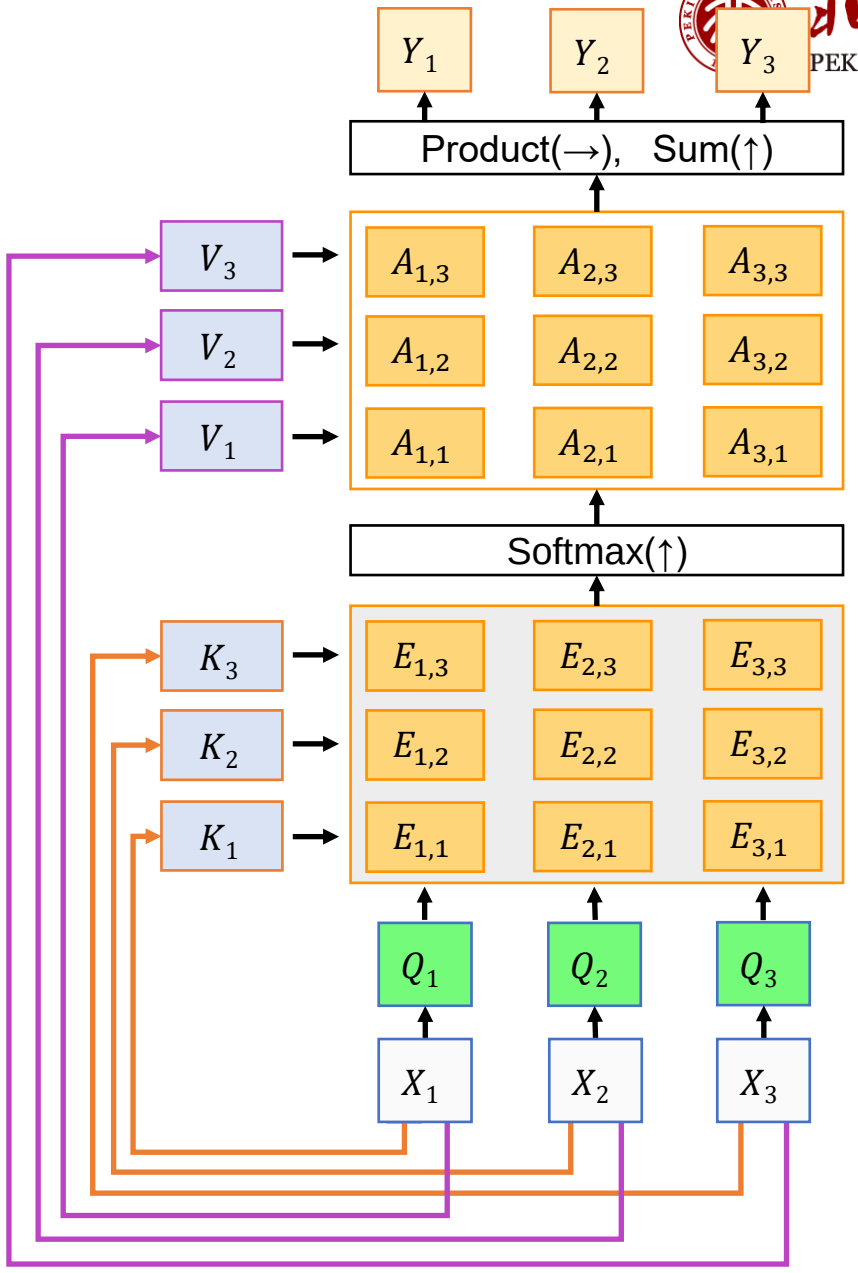
$$E_{i,j} = \frac{(Q_i \cdot K_j)}{\sqrt{D}} \text{ or } E = QK^T/\sqrt{D}$$

(D is the dimensionality of the keys)

- Attn. weights: $A = \text{softmax}(E, \text{dim} = 1)$
- Output vectors:

$$Y_i = \sum_j A_{i,j} V_j \text{ or } Y = AV$$

One query per input vector



Adapted from J. Johnson