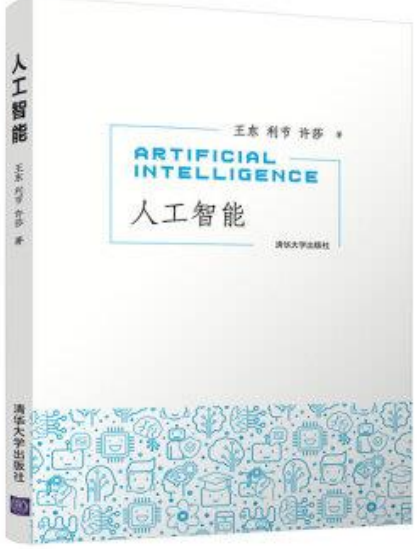
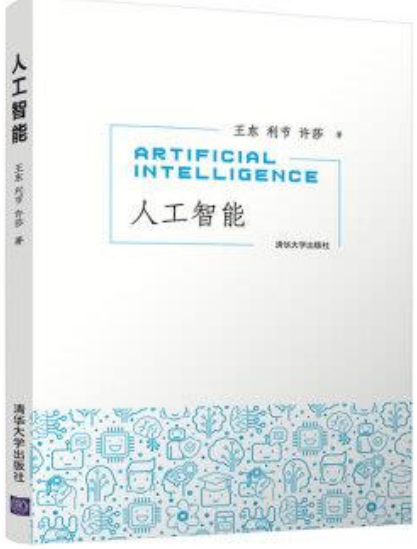

人工智能
王东 利节 许莎 著
ARTIFICIAL
INTELLIGENCE
人工智能
清华大学出版社
清华大学出版社

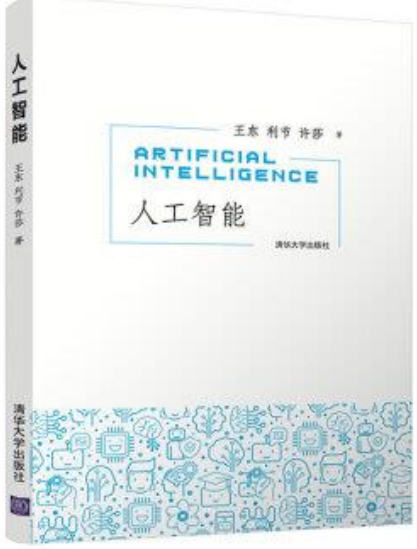
理解你的语言

利节


人工智能
王东 利节 许莎 著
ARTIFICIAL
INTELLIGENCE
人工智能
清华大学出版社
清华大学出版社

目录

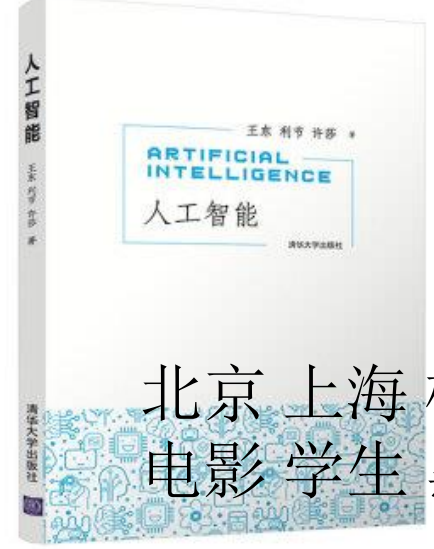
- 词向量
- 句向量


人工智能
王东 利节 许莎 著
ARTIFICIAL
INTELLIGENCE
人工智能
清华大学出版社
清华大学出版社

目录

- 词向量
- 句向量

传统自然语言处理方法中，“词”一直被认为是独立的个体，每个词都需要有明确定义的词性和词义，否则系统将无法对它进行处理。例如一个知识库中有“玫瑰有香味”这条知识，但这一知识并不会对“杜鹃有香味”是否正确有任何判断能力。这是因为“玫瑰”和“杜鹃”是两个独立的单词，无法通过“玫瑰”的知识实现对“杜鹃”的推理。



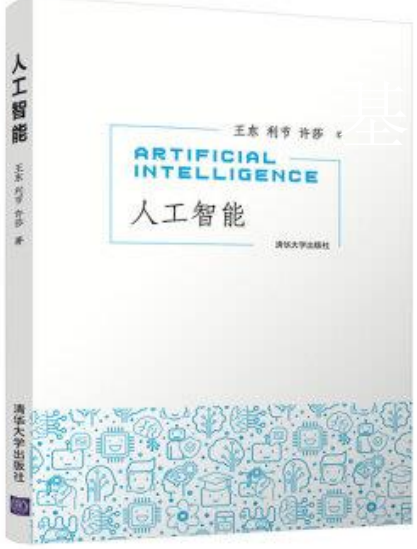
北京 上海 橘子 中国 美国 俄罗斯 台湾 华盛顿 汽车 日本 电影 学生 黑猫

和中国最相关的词是什么？

方法？

基于概念路径的计算方法　基于概念信息量的计算方法

对词性进行了分类和标注



城市
北京 华盛顿
上海 台湾

国家
中国 美国 俄
罗斯 日本

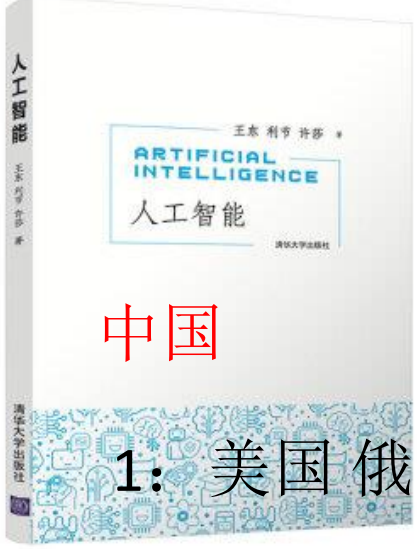
……

橘子

汽车

学生

黑猫



中国

1：美国 俄罗斯 日本

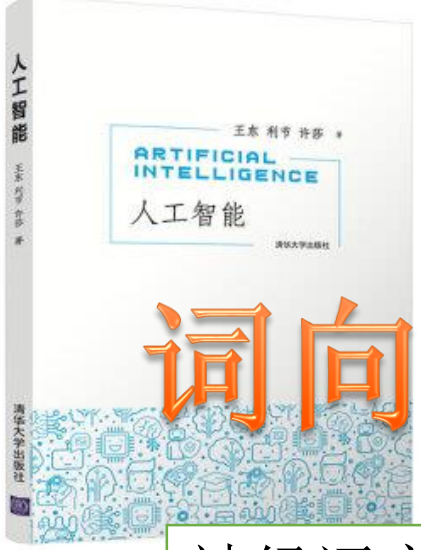
2：北京 华盛顿 上海 台湾

```
Enter word or sentence (EXIT to break): china

Word: china  Position in vocabulary: 486

                                              Word       Cosine distance
-------------------------------------------------------------------------
                                            taiwan              0.768188
                                             japan              0.652825
                                             macau              0.614888
                                             korea              0.614887
                                               prc              0.613579
                                           beijing              0.605946
                                            taipei              0.592367
                                          thailand              0.577905
                                          cambodia              0.575681
                                         singapore              0.569950
                                          republic              0.567597
                                          mongolia              0.554642
                                           chinese              0.551576
```

缺乏对词语的理解



词向量

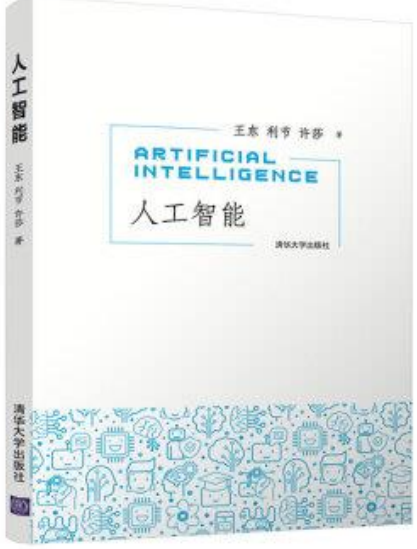
神经语言程序学NLP：研究我们的大脑如何工作

↓

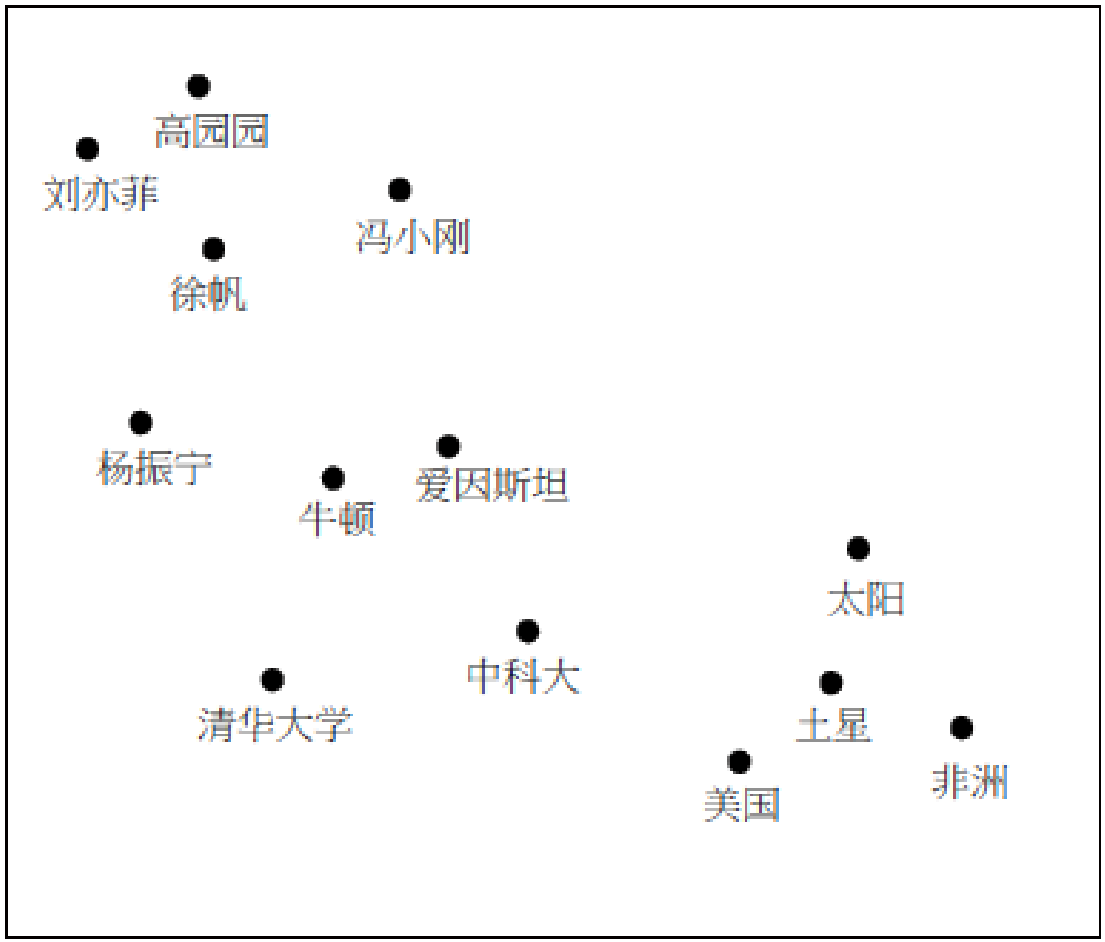
将自然语言理解的问题转化为机器学习的问题

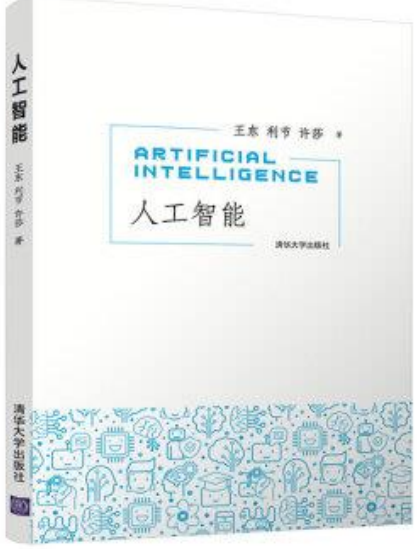
↓

第一步肯定是要找一种方法把这些符号数学化。

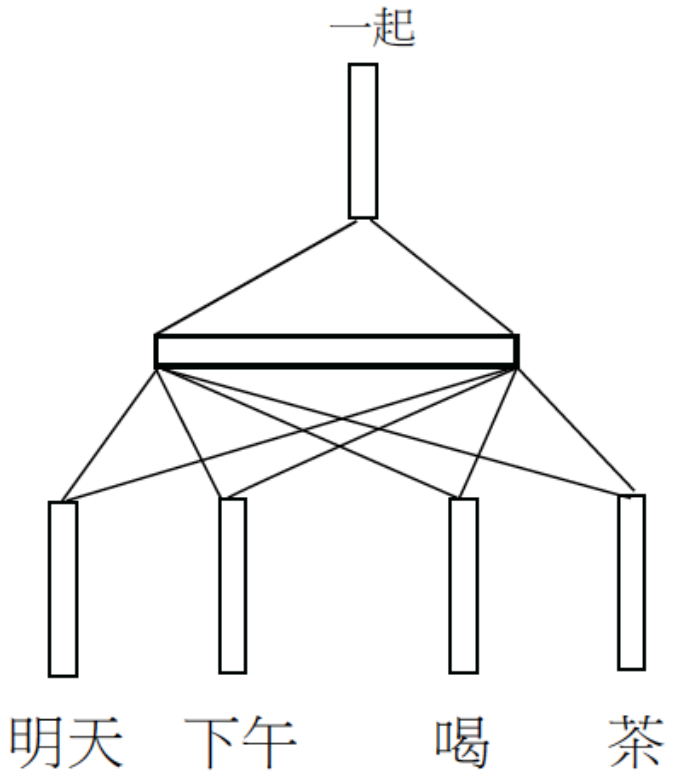


词向量实现了我们头脑里的单词地图。它将单词表示为一个高维空间中的连续向量（称为词向量），使得在该空间中语义越近的单词对应的向量距离更近。





如何实现这一映射呢？一个基本原则是使语义相近的单词在向量空间中尽可能接近，不相近的单词在向量空间中尽可能拉远。我们可以将这一原则写成一个目标函数，通过调整每个单词的词向量，使得目标函数最大化，即得到了一个词向量空间。



词的向量化

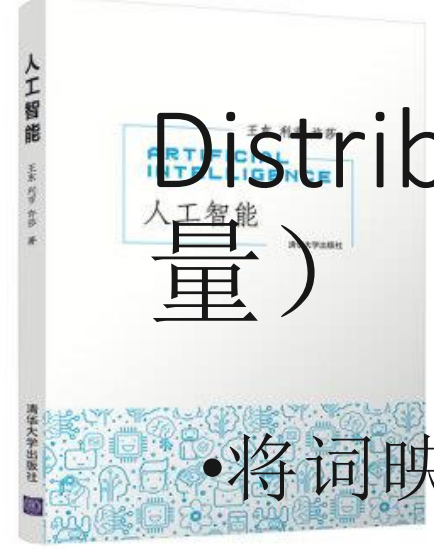
词的向量化，是将语言中的词用特定的向量来表示。词的向量化主要有两种表达方式：

- one—hot representation方式
- Distributed representation

one-hot representation

用一个很长的向量来表示一个词。词向量的长度为词典的大小，向量的分量只有一个1，其他的全为0，1的位置对应该词在词典中的位置。

举例来说：如果有一个词典大小为10^5，那么“土豆”可以表示为[1,0,0,0,...],而“马铃薯”表示为[0,1,0,0,...]。

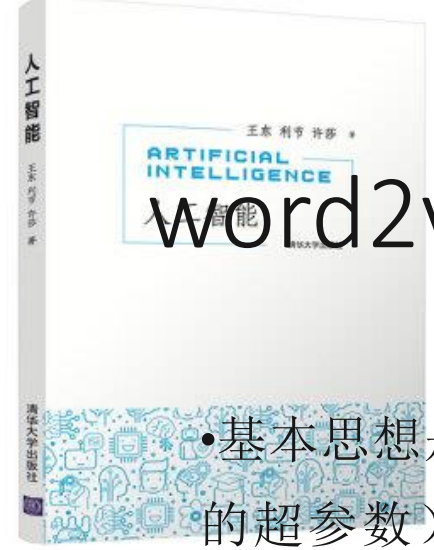


Distributed representation（词向量）

- 将词映射到一个低维、稠密的实数向量空间中，使得词义越相近的词在空间的距离越近。

“土豆”可以表示为：[0.843,-0.125,0.734,-0.345,...]

“马铃薯”表示为：[0.923,-0.231,0.698,-0.233,...]



word2vec模型

- 基本思想是：通过训练将每个词映射成K维实数向量（K一般为模型中的超参数），通过词向量之间的距离来判断他们之间的语义相似度（越相近的词在向量空间中越相近），其采用一个三层的神经网络，输入层-隐层-输出层。

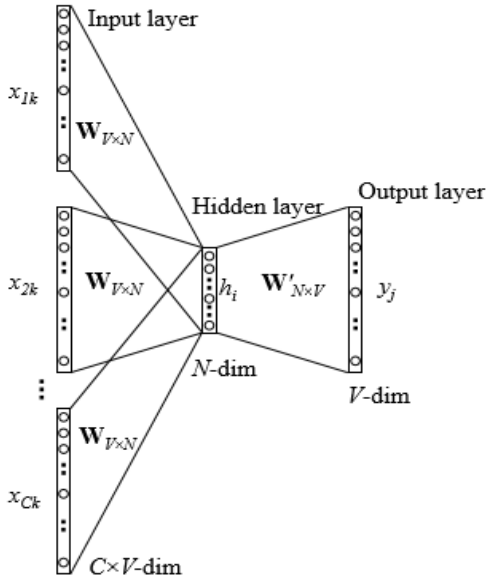
- word2vec模型有两种：

CBOW（Continuous Bag-of-Words）模型

Skip-gram模型

这两种算法都是利用人工神经网络作为它们的分类算法。起初，每个单词都是一个随机N维向量，经过训练之后，利用CBOW或者Skip-gram的方法获得了每个单词的最优向量。

CBOW模型



在计算隐层输出时，CBOW模型不是直接复制输入上下文单词的输入向量，而是取输入上下文单词向量的平均值，使用输入→隐层权重矩阵（v*N）与平均向量的乘积 作为输出。

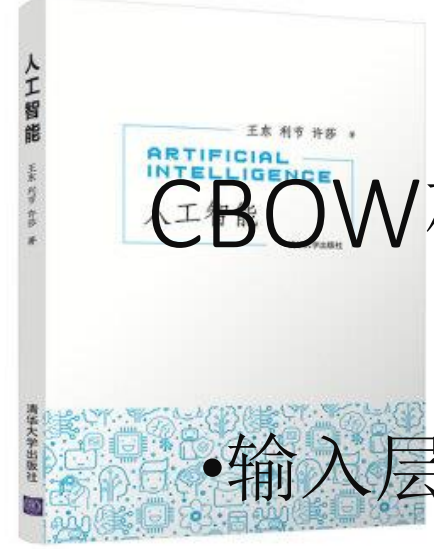
$$\begin{aligned}\mathbf{h} &= \frac{1}{C}\mathbf{W}^T(\mathbf{x}_1+\mathbf{x}_2+\cdots+\mathbf{x}_C)\\ &= \frac{1}{C}(\mathbf{v}_{w_1}+\mathbf{v}_{w_2}+\cdots+\mathbf{v}_{w_C})^T\end{aligned}$$

从隐藏层到输出层，是另一个不同的权重矩阵W/这是一个N×V矩阵。 使用这些权重，我们可以计算词汇表中每个单词的隐层→输出层权重

$$u_j = \mathbf{v}'^{T}_{w_j}\mathbf{h},$$

然后，我们可以使用对数线性分类模型softmax来获得目标词的后验分布，这是一个多项分布。

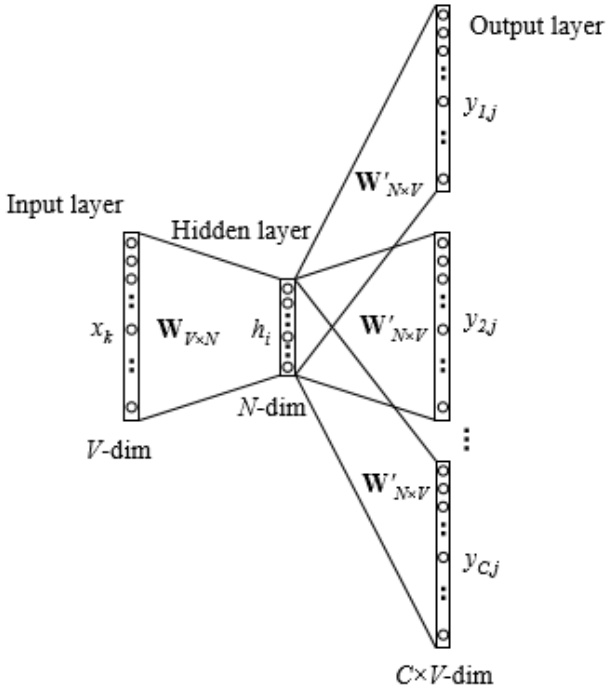
$$p(w_j|w_I) = y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V}\exp(u_{j'})},$$



CBOW模型

- 输入层：词w(t)的上下文中的2c个词向量
- 隐层：输入词向量的平均向量
- 输出层：是以训练语料库中出现过的词作叶子节点，以各词在语料库中出现的次数作为权值构造出一棵Huffman树。

Skip-gram模型



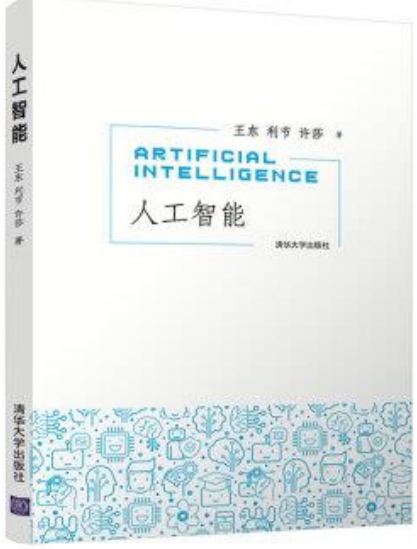
$$\mathbf{h} = \mathbf{W}^T_{(k,\cdot)} := \mathbf{v}^T_{w_I},$$

$$u_{c,j} = u_j = \mathbf{v}'^{\,T}_{w_j} \cdot \mathbf{h}, \text{ for } c = 1, 2, \cdots, C$$

各输出层上使用的是同一个权重矩阵

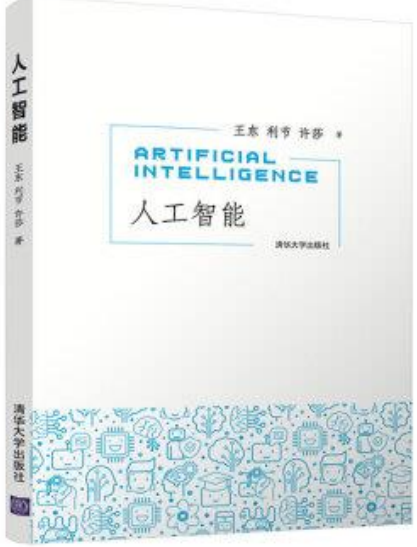
在输出层，不是输出一个多项分布，而是输出C个多项式分布。 每个输出使用相同的隐层→输出矩阵进行计算

$$p(w_{c,j} = w_{O,c} | w_I) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{j'=1}^{V} \exp(u_{j'})}$$


人工智能
王东 利节 许莎 著
ARTIFICIAL
INTELLIGENCE
人工智能
清华大学出版社
清华大学出版社

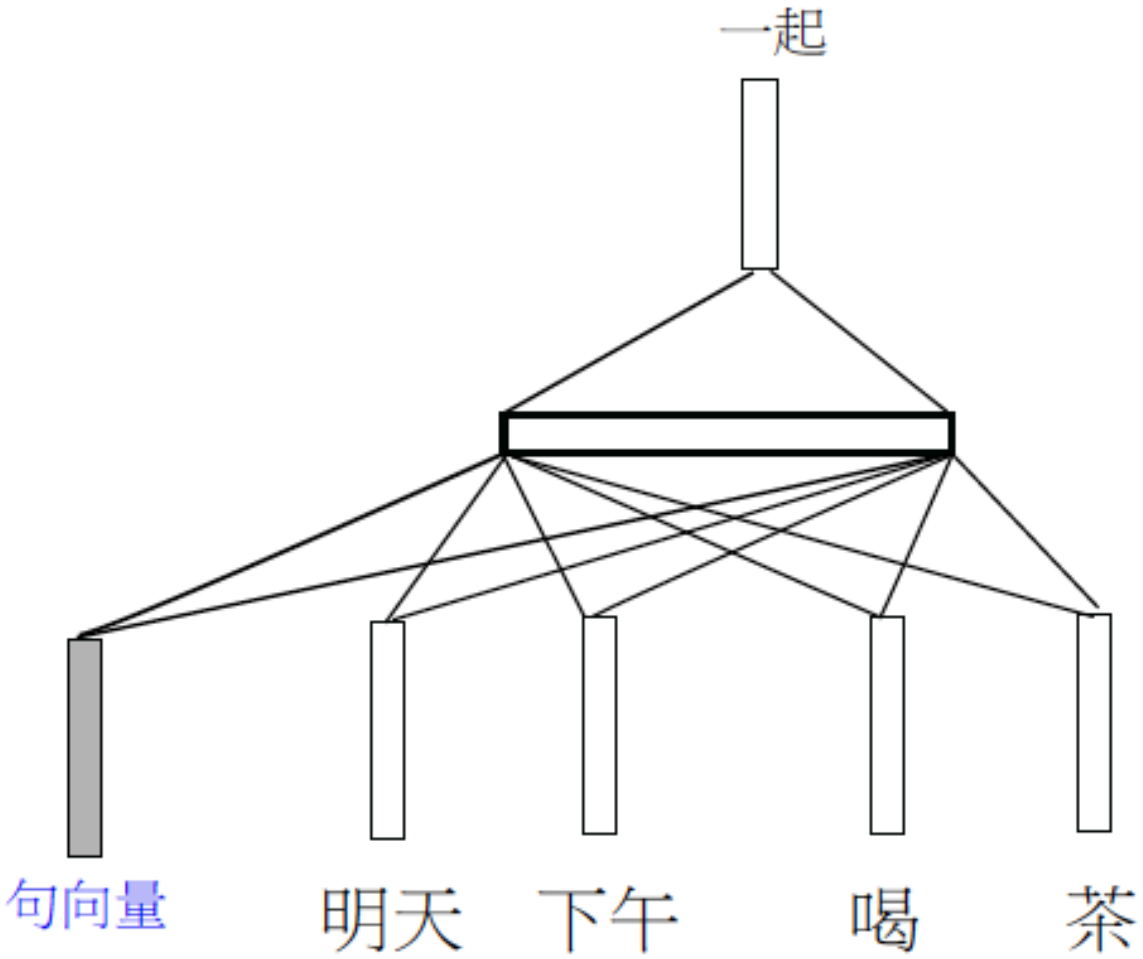
目录

- 词向量
- 句向量



词向量的提出使得语义在最小语言单元上具有了可计算性，由此可以扩展到对句子语义的计算，即句向量。

将句向量与词向量一起学习。经过学习以后的句向量会像词向量一样包含语义信息，只不过是整句话的语义信息。



基于句向量包含的语义信息可以实现很多语言理解任务。例如可以用来搜索FAQ（Frequently Asked Questions，常见问题解答），实现简单的问答。FAQ 是人为积累的问题-答案对，基于这一资源，对用户输入的新问题在FAQ 的问题集中进行搜索，找到最相似的问题，输出该问题对应的答案，即可实现一个问答系统。传统方法多用句子中所包含单词的统计信息来计算问题的相似性；基于句向量，可以在语义空间中通过句向量间的距离直接计算问题与问题之间的相似性，从而确定数据库中的相似问题并给出答案。

BOW（Bag-of-Words）模型

- 传统的BOW可以看作是词的one-hot representation向量的叠加。

- 例如：“土豆”表示为[1,0,0,0,...],而“马铃薯”表示为[0,1,0,0,...]

而一篇仅包含“土豆”和“马铃薯”这两个词的文本就可以表示为[1,1,0,0,...]

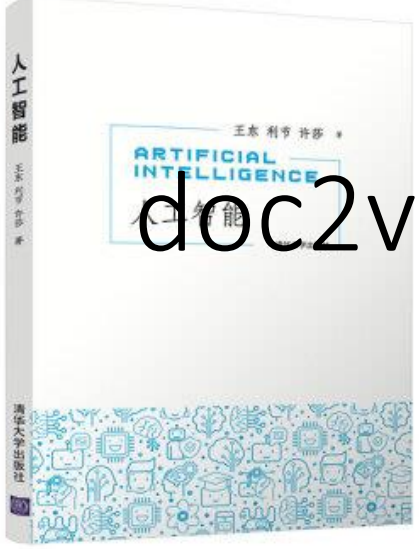
缺点：特征向量高维性和稀疏性，而且很难利用常规的向量距离公式有效的计算两篇文档之间的相似度。

doc2vec模型

- doc2vec模型的训练与word2vec模型类似，但在利用词的上下文对当前词进行预测的训练过程中添加了一个文档向量。
- 该模型也存在两种方法：

  Distributed Memory(DM) 和 Distributed Bag of Words(DBOW)

DM 试图在给定上下文和段落向量的情况下预测单词的概率。在一个句子或者文档的训练过程中，段落 ID 保持不变，共享着同一个段落向量。DBOW 则在仅给定段落向量的情况下预测段落中一组随机单词的概率。



doc2vec模型

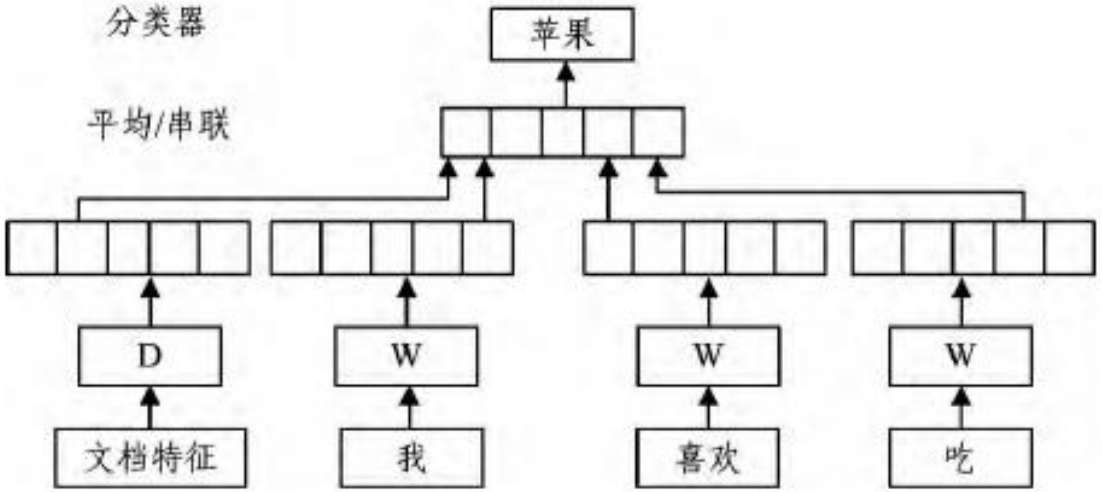


图 3　doc2vec 预测模型

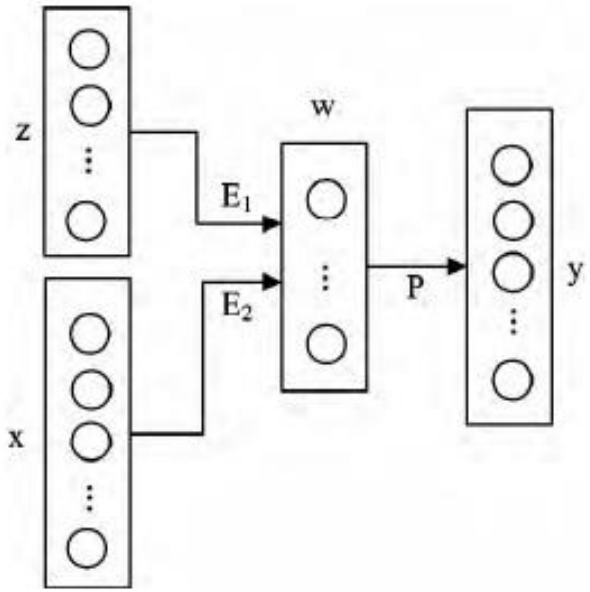


图 4　doc2vec 的训练过程

基于TF-IDF算法的word2vec改进方法

- 在word2vec词向量的基础上，结合TF-IDF算法提出了文档向量的表示方法。
- 对于包含M个文档的集合D，其中Di（i=1,2,...,M）已经采用分词工具对中文文档进行分词，将其通过word2vec模型训练，得到每个分词对应的N维词向量w，其中w=（v1，v2，...，vN）。
- 对于每类文档集中的每个文档里的每个分词，利用TF-IDF算法计算其在该文档中的权重值K（t,Di）（表示为词t在文档Di中的权重）。TF-IDF综合考虑了词在单个文档中出现的概率tf 以及该词在整个文档集中的权重idf。

基于TF-IDF算法的word2vec改进方法

TF-IDF的计算公式如下：

$$K(t,D_i)=\frac{tf(t,D_i)\times idf(t)}{\sqrt{\sum_{t\in D_i}[tf(t,D_i)\times idf(t)]^2}}$$

分子：tf（t,Di）为词 t在第i篇文档中的词频，idf是逆向文件频率
分母：归一化因子

$$TF_t=\frac{\text{在某一类中词条 t 出现的次数}}{\text{该类中所有的词条数目}}$$

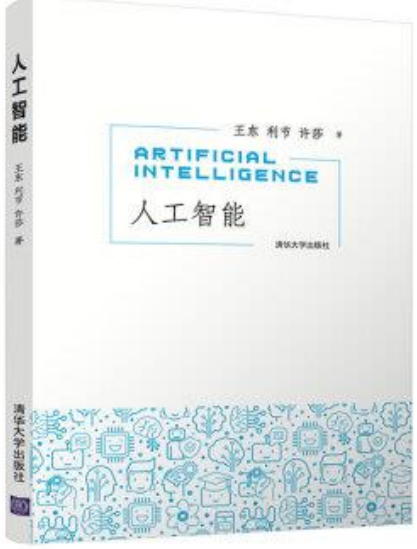
$$IDF=log(\frac{\text{语料库的文档总数}}{\text{包含词条 t 的文档数}+1})$$

基于TF-IDF算法的word2vec改进方法

- 对于每篇文档Di（i，2，...，M），其文档向量可以表示为如下形式：

$$d_i = \sum_{t \in D_i} w_t K(t, D_i)$$

- wt表示t的词向量，所以文档向量d也是一个N维的实数向量。


人工智能
王东 利节 许莎 著
ARTIFICIAL
INTELLIGENCE
人工智能
清华大学出版社
清华大学出版社

The end !