6 综述—— 词（词义）的向量表示

第六章概述

我们在对一段原始文本进行分词后，一种方式是直接基于词语序列中的词的共现频率来构建N元语言模型，但也可以考虑直接对词的词义进行表征。所谓“观其伴，知其义”、“词本无意，意由境生”，说明词义需要借助其上下文来表达。早期的**符号表示**将每个词视为一个独立、离散的符号，方法简单却需要大量的专家工作，并且计算机无法直接在其上进行量化计算，也难以完全隐藏在专家定义规则中的语义关系。与之相对的，基于统计学或机器学习方法的**词向量表示**能够很好的用于计算机计算与度量，并能非常自然的用于下游的自然语言处理任务。最初的词向量表示通过统计**词频**和**词共现频率**来构建，即**基于统计学的稀疏高维词向量表示**，如词袋模型。**词袋模型**简单将文本表示为一个词汇表中每个词出现频率的向量的操作存在明显的局限性，因此发展出利用**TF-IDF**（词频-逆文档频率）和**PPMI**（正互信息）来缓解词袋模型在词频特征重要性和词语关联性方面的不足。

由于共现关系的双向性与稀缺性，这类词向量的长度往往等于词表大小（$|\mathrm{V}|$ 一般很大）且非常稀疏，使得计算成本与内存消耗大的同时信息利用率低，模型对词义的表征能力下降。一种解决思路是采用各种**降维**方法实现高维向量的低维化，但更佳的方法是直接构建词义的低维向量表示。因此**基于机器学习方法的稠密低微词向量表示**方法诞生，最典型的代表是**Word2Vec**中的**CBOW模型**和**Skip-Gram模型**，结合**层次化Softmax**和**负采样**两种训练方法来预测词向量。由于窗口大小的限制，Word2Vec只能基于局部上下文来学习词向量，因此发展出了基于全局词共现矩阵的**GloVe模型**。上述提到词向量都是**静态词向量**，即给每个词分配一个唯一的向量表示，无法根据上下文调整词义，因此无法处理多义词或词义随上下文变化的情况。因此发展出一种能够根据不同上下文动态调整词向量的方法，这推动了**动态词向量**（或上下文敏感词向量）的出现。**ELMo**是最早实现动态词向量的模型之一，之后便基于强大的Transformer架构发展出了各种预训练语言模型如Bert、GPT等。

评估词向量优劣的常用方法可以分为**内在评估**和**外在评估**两类。前者通过专门设计的基准数据集，直接衡量词向量在语义相似度、类比推理等任务上的表现；后者将词向量模型应用于下游NLP任务，间接评估其质量。

词义及其符号表示

词义可分为概念义与色彩义。概念义反映了客观事物自身，色彩义则强调人们对事物的主观认识。根据词义的结构和层次可以划分出义素、义位和义项：

- 义素：词义的最小组成单位，也称为语义特征或语义成分。它是词义内部的基本成分，用于表示词的核心特征。其特点是自身不能独立成词，通过彼此组合才可以构成词的完整意义。
- 义位：一个词的最小独立语义单位，也是词义在语境中具体的一种表现形式。每个义位代表一个词或词组的某个独立的意义。词的不同义位通过语义、句法或语境的不同使用，展现出词义的不同面向。

- 义项：词在词典中列出的具体的解释或意义。一个词可以有多个义项，每个义项对应于该词的一个具体用法或语义层次。

这三个概念不同层次理解和分析词汇的结构和意义。义素揭示了词的语义组成，义位分析了词的多义性和语义变化，而义项则提供了词义的具体化解释。

符号表示是指通过符号或符号系统来表达事物、概念、关系或信息的一种表示方式。我们可以基于基本词对词义进行表示。使用一组数量有限的词义已知的基本词，再加上与其他词之间的关系来表示其他所有词的语义 。这里的基本词也就是义素，通常由专家定义。但是，基本词的选择是存在一定争议的，不同的专家可能有不同的选择可能。因此，还可以考虑基于词间之间的关系来进行符号表示。词间关系由词义关系反映，常见的有上下义位关系、全体-成员关系、整体-部分关系和同义关系等。基于词间关系的典型代表有同义词词林，其基于对中文词汇的系统化语义分类，通过层次结构展示词汇的同义性和近义性，揭示出词汇之间的语义微差。具体来说，同义词词林首先将最相似的词义组成原子词群，进而组成词群、小类、中类，最后到大类。

基于统计学的高维向量表示

词袋模型

词袋模型是自然语言处理和信息检索领域中一种最基本且常用的文本表示方法。尽管简单，词袋模型在文本分类、情感分析、信息检索等任务中也表现出一定的效果。顾名思义，“词袋”模型意味着将文本看作一个“词袋”，词汇的顺序无关紧要，只关注词汇的出现与否，以及出现的频率。词袋模型将文本中的词汇看作是独立的个体，不考虑它们在文本中的语法结构或词序信息。在这种模型中，文本被表示为词汇集合的向量，每个元素代表某个词在该文本中的出现次数或是否出现。

词袋模型的核心思想：

- 文本中的每个词是模型的一个特征，所有词组成一个词汇表。
- 每个文本都用一个固定长度的向量表示，向量的维度等于词汇表中词的数量。
- 向量中的每一维表示某个词在该文本中的出现频次（即词频，Term Frequency，简称TF），或者表示该词是否在文本中出现（使用二值表示）。

词袋模型的优缺点都显而易见。优点在于其简单易实现，适用于想在大规模文本数据迅速构建词向量表示的场景。缺点在于：

a. 丧失词序信息：词袋模型不考虑词的顺序或上下文关系，因此它不能区分语义上有重要顺序的文本。
b. 高维稀疏性：词袋模型的词汇表规模可能非常大，尤其在处理大语料时，这会导致生成的词向量是高维稀疏的。稀疏向量会占用大量内存，并可能增加计算开销。
c. 词义孤立性：词袋模型不能捕捉到词与词之间的语义关系。

TF-IDF

词频-逆文档频率TF-IDF是Term Frequency-Inverse Document Frequency的缩写，对词袋模型的一种重要改进。它不仅仅考虑词在单个文本中的频次，还考虑了该词在整个语料库中的重要性。作为一种**加权技术**，它通过计算每个词在文本集合中的重要性来衡量词的相关性，能够突出文本中那些具有区分度的词汇，并减少常见词对结果的影响。TF-IDF的基本思想是，一个词在某个文档中出现得越多，且在其他文档中出现得越少，那么这个词在该文档中就越重要。TF-IDF 由两部分组成：

- TF（Term Frequency, 词频）：表示某个词在文档中出现的频率，用于衡量词在该文档中的重要性。TF的值越大，表示该词在文档中越常见。
- IDF（Inverse Document Frequency, 逆文档频率）：用于衡量词在整个语料库中的普遍性。IDF的值越高，表示该词在语料库中比较稀有，越具有区分性。IDF可以降低那些在很多文档中都普遍出现的词的权重（例如“的”、“是”等常见词）。

TF-IDF即TF与IDF的乘积，计算公式如下：

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

其中 $\text{TF}(t, d) = \dfrac{\text{词 t 在文档 d 中出现的次数}}{\text{文档 d 中的总词数}}$，$\text{IDF}(t) = \log\left(\dfrac{N}{\text{df}(t)}\right)$，N 表示语料库中文档的总数，$\text{df}(t)$ 表示包含词 ttt 的文档数。通过这两个因素的结合，TF-IDF可以有效突出那些在特定文档中频繁出现、但在其他文档中不常见的词，从而帮助区分不同文档的内容。

PPMI

正点互信息PPMI是Positive Pointwise Mutual Information的缩写，它基于点互信息（PMI, Pointwise Mutual Information）的概念，但通过正化（将负值设为0）来保留词与词之间的正相关性，同时过滤掉负相关性或噪音，从而增强模型的稳定性和解释性。点互信息PMI 是一种用于衡量两个事件（如词语）之间关联程度的度量方法，表示它们在一起出现的概率与它们独立出现时概率的对比。其公式为：

$$\text{PMI}(x, y) = \log\left(\frac{P(x, y)}{P(x) \cdot P(y)}\right)$$

其中 $P(x, y)$ 是词 x 和词 y 在同一上下文中共现的概率，$P(x)$ 和 $P(y)$ 则分别是词 x 和词 y 单独出现的频率。若 x 和 y 比独立出现时更频繁地在一起出现，则它们的 PMI 值会较高，表示它们之间有较强的关联性。相反，如果 x 和 y 的共现频率低于随机独立出现的概率，则 PMI 值为负，表示它们的关联性较弱或不相关。

PMI 的一个缺点是可能产生负值，因为词 x 和 y 的共现频率低于预期时，PMI 值为负数。为了避免负值影响解释和后续计算，PPMI 对 PMI 进行正化处理，将负值设为0，仅保留正值。PPMI 的公式为：

$$\text{PPMI}(x, y) = \max(\text{PMI}(x, y), 0)$$

这意味着当 PMI 为负时，PPMI 值为 0；当 PMI 为正时，PPMI 值等于 PMI。这种正化操作的目的在于只保留词对之间的正相关性，忽略负相关性和不相关的词对，从而在某些任务中提高模型的效果。

从模型上来看，PPMI 构建的是词与词之间的共现矩阵，因此它不仅能用于文本表示，还可以用于直接计算词汇之间的相似度。PPMI通过共现信息来捕捉上下文中词汇之间的关联性，相反的，词袋模型和TF-IDF都不考虑上下文信息，也就是说，它们不处理词与词之间在同一文档中的相对位置或相互关系。词袋模型和TF-IDF只处理词是否在文档中出现以及出现的频率。

基于机器学习方法的低微向量表示

CBOW与Skip-gram

CBOW模型

CBOW（Continuous Bag of Words）模型是Word2Vec算法中的一种，核心思想是通过给定上下文词汇来预测中心词，它利用词与上下文的共现信息来捕捉词汇之间的语义关系。CBOW是词嵌入学习的一种无监督学习方法，它通过训练生成的词向量可以用于自然语言处理任务中的词语相似度、文本分类、机器翻译等任务。

在 CBOW 模型中，给定一个上下文窗口（即目标词周围的若干个词），模型通过上下文词汇来预测目标词。CBOW 将上下文中的每个词映射为一个向量，并通过这些向量的平均值来预测目标词的向量。最终，模型学习到每个词的低维稠密向量表示，这些向量可以捕捉词与词之间的语义关联。

CBOW 的核心是一个浅层神经网络，网络的结构简单，通常包含输入层、隐藏层和输出层。具体工作流程如下：

a. 输入层

输入层接收的是多个上下文词汇，这些词通常通过独热向量表示。假设词汇表的大小为 V，每个词都会被表示为一个V 维的独热向量，其中只有一个位置为 1，其余位置为 0，表示该词在词汇表中的索引。假设上下文窗口大小为 c，则输入层会接收 2c 个上下文词，每个上下文词用一个长度为 V 的独热向量表示。

b. 隐藏层

所有上下文词的独热向量首先会映射为词向量，即通过一个权重矩阵（即嵌入矩阵）将高维独热向量转化为低维稠密向量。隐藏层的输出是上下文词向量的平均值。假设上下文词向量为 $w_1, w_2, ..., w_n$ ，则隐藏层的输出可以表示为：

$$\vec{h} = \frac{1}{2c} \sum_{i=1}^{2c} W \cdot \vec{x}_i$$

其中 W 是嵌入矩阵，大小为 $V \times d$， $\vec{x}_i$ 是第 i 个上下文词的独热向量， $\vec{h}$ 是上下文词向量的平均值。

c. 输出层

隐藏层的输出是一个低维向量表示，通过一个softmax分类器计算出目标词的概率分布。输出层的目标是预测词汇表中目标词的概率，公式如下：

$$P(w_{\text{target}}|w_1, w_2, ..., w_n) = \frac{\exp(\vec{v}_{w_{\text{target}}} \cdot \vec{h})}{\sum_{w \in V} \exp(\vec{v}_w \cdot \vec{h})}$$

d. 目标函数

CBOW模型的目标是最大化上下文词预测目标词的概率。其损失函数通常是通过最大化整个训练集上的对数似然来最小化，即：

$$L = -\sum_{(w_{\text{context}}, w_{\text{target}})} \log P(w_{\text{target}}|w_{\text{context}})$$

通过优化这个目标函数，CBOW能够学习到每个词的词向量，使其在上下文中的预测表现最优。

Skip-gram模型

与CBOW模型相对，Skip-gram模型采取的是通过中心词预测上下文，而不是通过上下文词预测中心词。给定一个中心词（也称目标词），模型尝试预测该中心词的上下文中的其他词汇。Skip-gram与 CBOW 的模型结构的主要差异如下表所示：

特征	Skip-gram	CBOW
输入	中心词（一个词的独热向量）	上下文词（多个词的独热向量）
隐藏层	中心词通过嵌入矩阵映射为词向量（低维稠密向量）	上下文词通过嵌入矩阵映射为词向量，取平均值作为隐藏层输出
输出	预测上下文中的词	预测中心词
目标	最大化中心词预测上下文词的概率	最大化上下文词预测中心词的概率
训练复杂度	计算时间更长，训练需要遍历多个上下文词对	计算时间更短，因为计算的是上下文词的平均值
适合的任务	适合小语料库和需要捕捉长距离词汇关系的任务	适合大规模语料库和上下文较短的任务

分层softmax与负采样

分层softmax

在标准的CBOW模型中，输出层使用Softmax函数来计算目标词在整个词汇表中的概率分布，需要计算每个词的得分 $(\exp(\vec{v}_w \cdot \vec{h}))$ ，并对整个词汇表中的所有词求和，计算量为 $O(V)$ 。这对于大规模词汇表来说，计算效率非常低。因此，Word2Vec算法设计了分层Softmax来避免上述一次进行 V 分类问题。

分层Softmax 通过构建霍夫曼树来优化Softmax的计算，将计算复杂度从 $O(V)$ 降低到 $O(\log V)$ 。其基本思想是将每个词的概率计算转化为一系列二分类决策，而不是直接计算每个词的概率。通过霍夫曼树的层次结构，分层Softmax可以更快速地确定目标词。

霍夫曼树是一种用于编码的二叉树结构，通常用于数据压缩。在分层Softmax中，每个词在霍夫曼树中对应于一个叶子节点，路径上的每个节点表示一个二分类决策。霍夫曼树的构建方式是基于词的频率越高，树中的路径越短，从而减少了高频词的计算成本。分层Softmax中的目标词通过树中的路径表示。假设有一个词 w_{target} 它在霍夫曼树中的路径是从根节点到叶子节点的路径。每条路径对应一个二分类问题，表示是选择左子树还是右子树。每次从根节点开始，通过一系列二元决策来选择目标词所在的分支，直到找到该词为止。每次决策都通过一个sigmoid函数来计算，这样可以将大规模的Softmax问题转化为多个简单的二分类问题，每个二分类进行如下决策：

$$P(w_{\text{target}}|w_{\text{context}}) = \prod_{i=1}^{L} P(n_i|\vec{h})$$

其中 L 是从根节点到目标词 w_{target} 所需的路径长度， n_i 是路径上的第 i 个节点，表示是否选择左子树或右子树, $\vec{h}$ 是上下文词向量的平均值， $P(n_i|\vec{h})$ 是通过sigmoid函数计算的二分类概率。

负采样

负采样是用于加速Skip-gram 和 CBOW 模型（特别是 Skip-gram）的重要优化技术，与分层Softmax一样是为了用于在大规模词汇表下简化Softmax 计算，提高训练效率。其核心思想是将原本需要计算的整个词汇表的概率分布问题简化为二分类问题。具体来说，对于每个正样本（即真实的上下文词对），模型不仅计算它们的概率，还随机采样一些与中心词无关的负样本（即不存在上下文关系的词对），并让模型区分这些正样本和负样本。

负采样的工作流程如下：

a. 正样本选择：正样本是 Skip-gram 模型中训练时所用的真实词对。对于给定的中心词 w_{center} ，模型会采集其上下文中的所有正样本词汇（即与中心词共同出现的词）

b. 负样本采样：负样本是与中心词没有直接关系的词，也就是说，负样本的词是那些不与中心词共同出现在上下文中的词。这些负样本是从整个词汇表中随机采样得到的，但为了避免采样过于随机导致模型不稳定，负样本的采样是基于词频分布的非均匀采样。常见的采样策略是根据词汇的频率进行调整，词频较高的词被抽取为负样本的概率也会较大。一种常用的负采样概率分布为：

$$P(w) \propto f(w)^{3/4}$$

其中， $f(w)$ 是词 w 在语料库中的频率。通过这种方式，常见词仍有较高的被采样概率，但过于频繁的词（如“的”、“是”等）会受到抑制。

c. 计算损失函数:负采样通过将正样本的概率最大化，并将负样本的概率最小化来优化模型。具体来说，Skip-gram 模型的损失函数可以简化为：

$$L = \log \sigma(\vec{v}_{w_{\text{context}}} \cdot \vec{h}) + \sum_{k=1}^{K} \log \sigma(-\vec{v}_{w_k} \cdot \vec{h})$$

第一部分 $\log \sigma(\vec{v}_{w_{\text{context}}} \cdot \vec{h})$ 用于最大化正样本的概率，即希望中心词与其实际上下文词的向量相似度较大，预测的概率较高；第二部分 $\sum_{k=1}^{K} \log \sigma(-\vec{v}_{w_k} \cdot \vec{h})$ 用于最小化负样本的概率，即希望

中心词与负样本（即随机采样的与中心词无关的词）的相似度较小，预测的概率较低。

本章问题

1. 将词以词向量的形式进行表示固然很好，现如今对其的使用也证明了这种表示方法确实有其强大之处。但是有没有可能还有另一种更好的词表示方法，它不以向量的形式但是也能很好的对词进行表示，只是暂时没被探索到呢？
2. 在当前这个大模型以及追求AGI的时代，多模态肯定是一个必然的趋势，那么单纯的词向量肯定是无法满足多模态的需求。我们该如何开发一种跨模态的词向量表示，使得词汇不仅能够嵌入到语言空间，还能够嵌入到与视觉、听觉、动作等其他模态共享的通用向量空间中且能很好的表示呢？
3. 当前的词向量模型（如BERT、GPT）通过上下文生成动态词向量，但这些词向量似乎往往都是固定维度的且线性。在人类智能中，语义理解是动态且多维的，根据具体的语境或认知需求，某个词的表示可能涉及多个维度的复杂互动。那么能否设计一种动态多维度的词向量模型，该模型可以根据具体任务或上下文生成不同维度的向量表示呢？