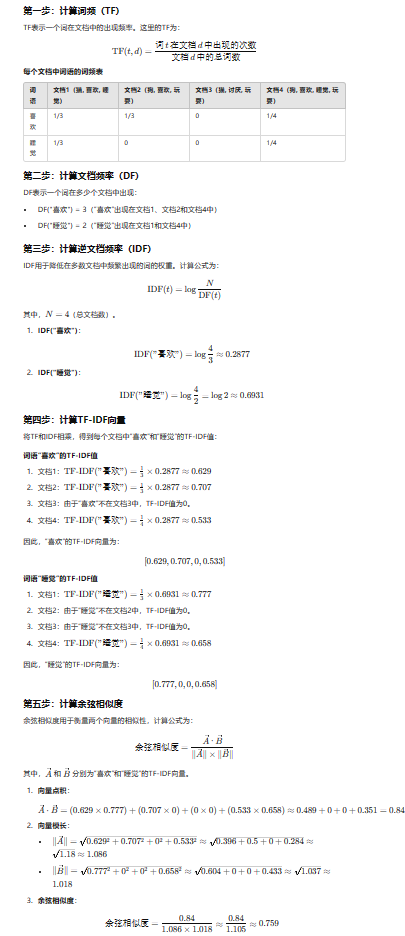
**Vector Semantic and Embedding**

**（一）假设有以下文档集合，每个文档包含的词语如下：**

* **文档1：猫，喜欢，睡觉**
* **文档2：狗，喜欢，玩耍**
* **文档3：猫，讨厌，玩耍**
* **文档4：狗，喜欢，睡觉，玩耍**

**请计算词语"喜欢"和“睡觉”的TF-IDF向量，并计算二者的余弦相似度。**

**答案：**



**（二）在Word2Vec模型中，下采样和负采样(negative sampling)的作用分别是什么？二者有什么区别。**

**答案：**

在Word2Vec模型中，下采样（subsampling）和负采样（negative sampling）是两种用于优化训练过程的技术，目的是提高模型的训练效率并提升词向量的质量。以下是两者的作用及其区别：

**1. 下采样（Subsampling）**

作用：

在训练词向量时，高频词（如“的”、“是”、“在”等）由于出现次数过多，会对训练产生过大的影响，导致模型过多关注这些常见词，而忽视了语义信息较丰富的低频词。下采样的目的是减少这些高频词对模型的影响。具体而言，模型会以一定概率随机丢弃高频词，从而让模型更关注信息量更大的低频词，提高模型的效果。

实现方式：

下采样是基于词频的。对于每个词，根据其出现的频率计算一个概率，当该概率高于某一阈值时，这个词会被随机跳过，从而减少其在训练集中的频率。

**2. 负采样（Negative Sampling）**

作用：

负采样用于简化Word2Vec模型的训练过程，特别是Skip-Gram模型的输出。Word2Vec的目标是让模型更好地预测目标词的上下文词，而负采样通过选择一些与目标词无关的负例词（即在上下文中不出现的词），让模型学习到哪些词不应该出现在目标词的上下文中。这种方法使得训练过程更高效，且减少了计算复杂度。

实现方式：

负采样在训练过程中，为每对正样本（目标词和其上下文词）随机选择一些负样本（一般为5-20个非上下文词），然后通过一个简化的损失函数来优化模型。这种方法替代了Softmax层，极大地降低了计算成本。

**区别**

- 目的不同：下采样用于减少高频词的干扰，使模型更关注低频词；负采样用于优化模型的训练效率，减少计算量。

- 应用位置不同：下采样作用于输入数据，通过跳过高频词减少它们的影响；负采样作用于模型的输出层，通过增加负例帮助模型更好地区分相关和不相关的词。

- 实现方式不同：下采样基于词频，直接对输入词语进行处理；负采样则是通过在训练时引入负例样本来优化目标函数。

**（三）在词嵌入模型中，为什么余弦相似度通常被用来衡量词向量之间的相似性？请解释余弦相似度的优势，并讨论其与欧氏距离的区别。**

**答案：**在词嵌入模型中，余弦相似度常被用来衡量词向量的相似性，因为它关注的是向量的方向而非长度。这种特性在词向量中尤为重要，因为相似的词通常在空间中指向相似的方向，即使它们的向量大小不同。余弦相似度对向量进行了“标准化”处理，不会因为词频等因素导致的向量大小差异而影响相似性判断，使其在比较词语语义相似性时更稳定。

相比之下，欧氏距离测量的是两个向量在空间中的绝对距离，它更适用于衡量位置差异。因为欧氏距离受向量长度的影响较大，若没有标准化处理，词频较高的词会导致较大的欧氏距离，影响相似性判断。因此，余弦相似度在词嵌入中更常用，因为它忽略了向量的大小差异，更关注语义上的接近性。

**Neural Language Model**

**（一）说明LSTM中有几个门结构，每个门的作用是什么。**

**答案：**

在LSTM（长短期记忆网络）中，有三个主要的门结构：遗忘门、输入门和输出门。每个门的作用如下：

1. 遗忘门（Forget Gate）

作用：决定哪些信息应该从细胞状态中丢弃。

原理：遗忘门根据当前输入和前一时刻的隐藏状态进行计算，输出一个0到1之间的值。这个值会与前一时刻的细胞状态相乘，从而决定哪些信息被保留（接近1）或遗忘（接近0）。

2. 输入门（Input Gate）

作用：决定哪些新的信息需要加入到细胞状态中。

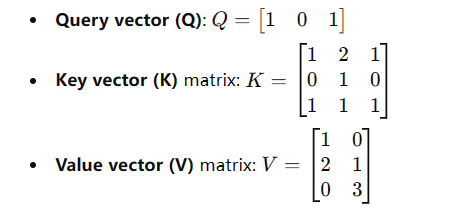
原理：输入门计算当前输入和前一时刻隐藏状态的加权值，以确定哪些信息是重要的，同时生成一个候选状态。输入门的输出会与候选状态相乘，再与细胞状态相加，以更新细胞状态。

3. 输出门（Output Gate）

作用：决定哪些信息从细胞状态中输出。

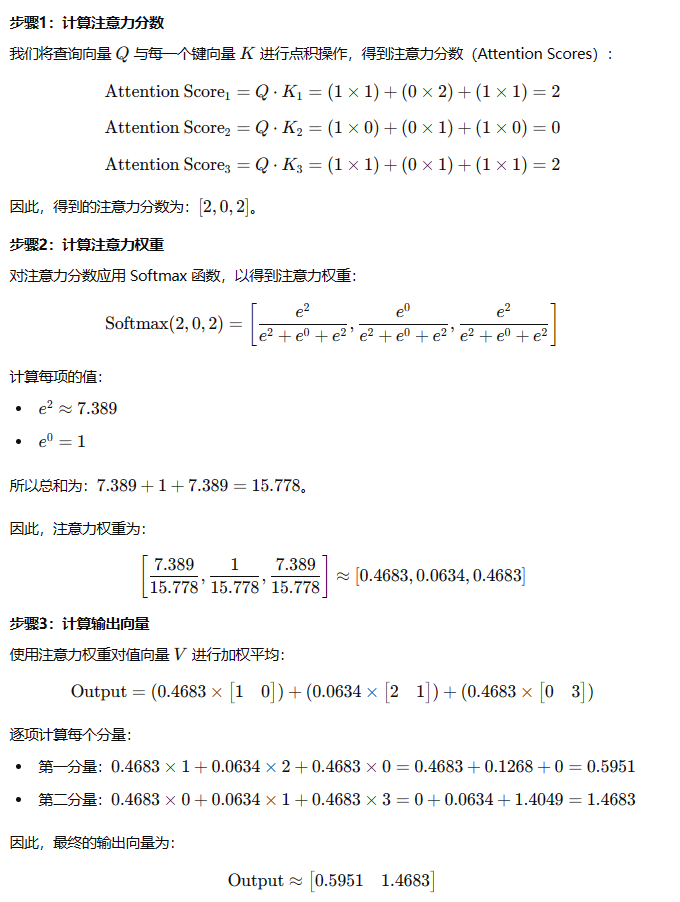
原理：输出门根据当前输入和前一时刻隐藏状态来生成一个值，用于调节细胞状态中哪些信息对输出重要，并生成当前的隐藏状态，以便传递给下一个时间步或输出层。

**（二）Here is a calculation question about the attention mechanism. In a simple attention mechanism, given the following matrices:**

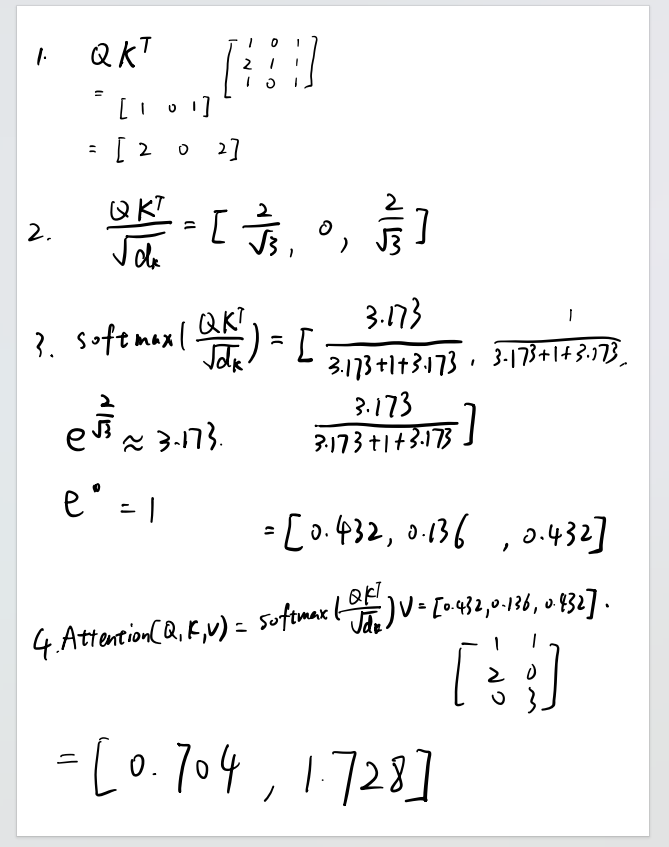


**Please calculate the attention weights of the query vector Q for each key vector and compute the final output vector based on these weights. The similarity can be calculated by dot product.**

**答案:**



改正之后的



**（三）说明BERT预训练中主要的机制和对应的损失函数是什么，并解释分别的原理和作用。**

**答案：**在BERT的预训练过程中，主要有两个核心机制以及相应的损失函数：遮掩语言模型（Masked Language Model, MLM）和下一句预测（Next Sentence Prediction, NSP）。这两个机制帮助BERT在大量无标注文本数据中学习丰富的语言表示，使得它在下游任务中表现优异。以下是对它们的原理和作用的解释：

**1. 遮掩语言模型（Masked Language Model, MLM）**

机制原理：在MLM中，BERT会随机遮掩输入文本中一部分单词（通常是15%），并尝试预测被遮掩的单词。具体来说，模型将这些单词替换为一个特殊的“[MASK]”标记，然后根据其上下文（即剩余未遮掩的单词）预测这些遮掩的单词是什么。

损失函数：MLM使用交叉熵损失函数来计算预测的词汇概率分布与真实单词之间的差异。

作用：MLM使模型能够理解单词与其上下文的关系，因为模型需要基于上下文推断被遮掩的单词。这种方法使得BERT学习到了更加细致的语义信息，尤其是在句子内部的词汇依赖关系上，从而获得了强大的语言理解能力。

**2. 下一句预测（Next Sentence Prediction, NSP）**

机制原理：在NSP中，BERT会输入两个句子，并让模型预测第二句是否是第一句的下文。训练数据包含50%的真实句对（第一句与第二句相连）和50%的伪句对（随机选择的无关句子作为第二句）。

损失函数：NSP使用二元交叉熵损失来衡量模型预测的句子关联概率与真实标签之间的差异。

作用：NSP帮助BERT理解句子间的语义关系，使其不仅能处理单句任务，还能处理句子对任务（如自然语言推断和问答）。NSP使BERT具备了识别句子间逻辑关系的能力，提高了其在更复杂语言任务中的表现。

**Part of Speech Labeling**

**（一）在词性标注中，常见的歧义问题会对标注准确率产生怎样的影响？请举例说明“back”在不同上下文中的词性变化，并讨论词性标注算法（如隐马尔可夫模型）如何处理这种歧义，可以从建模上下文的角度进行解释。**

**答案：**在词性标注中，词汇的多义性会显著影响标注的准确率，因为同一个词在不同的上下文中可能具有不同的词性。例如，“back”一词在以下句子中具有不同的词性：

1. **形容词 (Adjective)**：The back door is open. （back 在这里修饰名词 door，表示“后面的”）
2. **名词 (Noun)**：He hurt his back. （back 在这里表示“背部”）
3. **动词 (Verb)**：They will back the proposal. （back 在这里表示“支持”或“为…做后盾”）
4. **副词 (Adverb)**：Please come back soon. （back 在这里表示“返回”）

在处理这种多义性时，词性标注算法，例如隐马尔可夫模型（HMM），会利用词的上下文信息来确定其最可能的词性。HMM 的核心思想是，假设每个词的词性不仅依赖于当前词的特征，还依赖于前一个词的词性。通过计算各个词性序列的概率，HMM 能够选择一个最有可能的词性序列，从而解决多义性问题。

举例来说，对于“back”一词，HMM 会根据其前后的词（例如“the back door”中的“the”和“door”）来判断它更可能是形容词。而在“they will back the proposal”中，前一个词“will”通常是情态动词，因此HMM会判断“back”更有可能是动词。

通过这种方式，HMM 等序列模型在一定程度上能够处理词性歧义，提高词性标注的准确率。

**（二）在词性标注问题中，如何使用前向算法计算观测序列的概率？请结合隐马尔可夫模型的结构，说明前向算法的具体步骤和优势。**

**答案：**

**前向算法的作用:**

前向算法是一种动态规划方法，用于在隐马尔可夫模型（HMM）中计算一个给定观测序列（比如一串单词）在不同词性状态组合下的总概率。它主要解决的是：在所有可能的词性标注路径中，找到符合观测序列的总体概率，而不需要穷举所有可能路径。

**前向算法的步骤:**

1. 初始化：从观测序列的第一个单词开始，根据初始状态概率（各词性的起始概率）和每个词性生成该单词的概率，计算出这个单词在每个词性上的初始概率。

2. 递归计算：从第二个单词开始，对每个词性，根据之前的词性状态计算概率，并乘以该词性生成当前单词的概率。这个过程逐步累积概率值，一步步向前推进，直到序列的最后一个单词。

3. 终止：在最后一步，把所有可能的路径概率加总，得到观测序列的总概率。

**前向算法的优势:**

前向算法通过将每一步计算的结果存储下来并用于后续计算，避免了对所有路径的重复计算。这种方式使得我们可以在保证精确度的同时，极大地减少计算量，从而在处理长文本或复杂标注任务时更加高效。