Vector Semantics and Embeddings

向量语义学和词嵌入

Abstract

词汇中在相似语境中出现的词汇往往具有相似的含义。

本章内容向量语义学(vector semantics)，通过学习词汇在文本中的分布来直接表示词义，这些表示称为词嵌入(embeddings)。

6.1 Lexical Semantics词汇语义学

词义表示旧有的问题及需求

问题：传统NLPhe1n-gram模型中词汇通常只被表示为简单的字母，所表示含义不够。

需求：好的词义模型应该能够帮助我们进行推理，处理与意义相关的任务，如问答或对话。

接下来介绍了一些有关词汇语义的具体知识：

词元(lemma):指字典中用来代表一个词的基本形式，例如mouse是小鼠和计算机鼠标这两个概念的词元。同样，sing是sing、sang、sung的词元。

词义(sense):每个词元可以有多个意义，即词义。例如，mouse既可以指小鼠也可以指计算机鼠标，这两种意义称为mouse的两个词义。

多义性(polysemy):指的是一个词元具有多个词义的情况, 后续有词义消歧(word sense disambiguation)的任务，即确定在特定上下文中使用词的哪个意义。

同义词(Synonymy)：指的是具有完全相同或几乎相同含义的两个词的词义

词相似性(Word Similarity)：虽然词汇可能没有许多同义词，但大多数词汇都有许多相似的词。例如，cat和dog不是同义词，但它们是相似的词。

词关联性(Word Relatedness)：两个词的意义可以以除相似性以外的方式相关联。词关联性是指词之间的一种联系，这些联系可能是因为它们共同参与日常事件而成为关联词，如coffee和cup，尽管它们在特征上几乎没有共同点，但因为共同参与喝咖啡的事件而被认为是相关的。

语义场(Semantic Field)：语义场是一组覆盖特定语义领域并彼此承担结构关系的词汇集合。例如，医院、餐厅或房屋等领域的词汇构成不同的语义场。

语义框架和语义角色(Semantic Frames and Roles)：语义框架是一组表示特定类型事件中的视角或参与者的词汇集合。例如，在商业交易事件中，一个实体为了获取某种商品或服务而向另一个实体交换金钱，之后商品转手或服务被执行。这个事件可以通过使用诸如buy、sell、pay这样的动词或者名词如buyer来在词汇上进行编码。框架有语义角色（如买家、卖家、商品、金钱），句子中的词可以承担这些角色。

内涵(Connotations)：内涵是指一个词的含义与作者或读者的情感、情绪、观点或评价相关的方面。

情感意义的三个维度：早期关于情感意义的研究发现，词汇在情感意义上沿着三个重要的维度变化：情感价值（valence，即刺激的愉悦度）、唤醒度（arousal，即刺激引发的情绪强度）和支配度（dominance，即刺激施加的控制程度）。

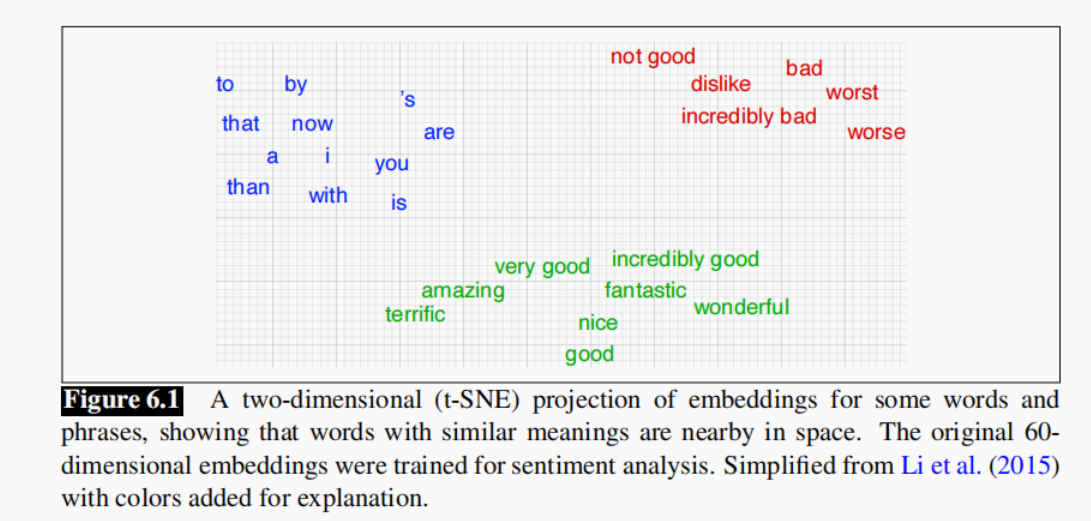
6.2Vector Semantics向量语义学

核心思想：向量表示

向量语义学的核心思想是将词表示为由词的分布导出的多维语义空间中的一个点。这些用于表示词的向量称为嵌入(embeddings)，这些嵌入可以是稠密的，如word2vec模型生成的，也可以是稀疏的，如基于tf-idf或PPMI的向量。

词嵌入的应用：

向量语义学为NLP应用提供了精细的词相似性模型。例如，情感分类器依赖于训练集和测试集中出现的相同词汇。通过将词表示为嵌入，分类器可以在见到具有相似含义的某些词时分配情感。向量语义模型可以从文本中自动学习，无需监督。



计算语义相似性：

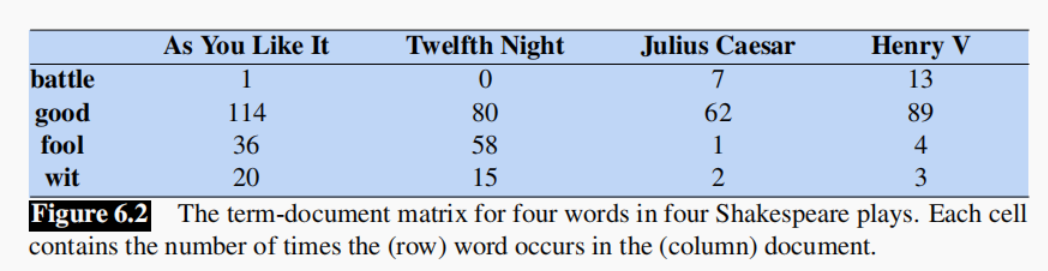
余弦相似度(cosine similarity)是使用嵌入来计算两个词、两个句子或两个文档之间语义相似性的标准方法，这在问答、摘要、自动作文评分等实际应用中是一个重要工具。

6.3Wrods and Vectors词与向量

探讨了向量语义学(Vector Semantics)中词与向量的关联，着重于词共现矩阵(co-occurrence matrix)：词-文档矩阵(term-document matrix)、和词-词矩阵(term-term matrix)的概念及其在自然语言处理(NLP)中的应用。

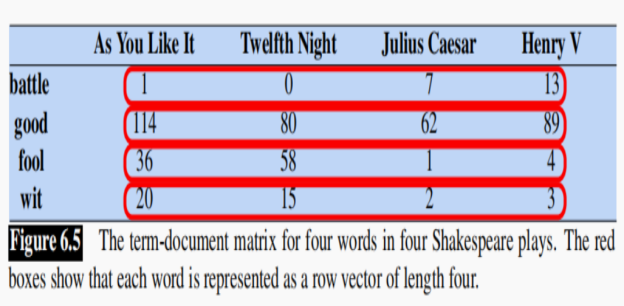
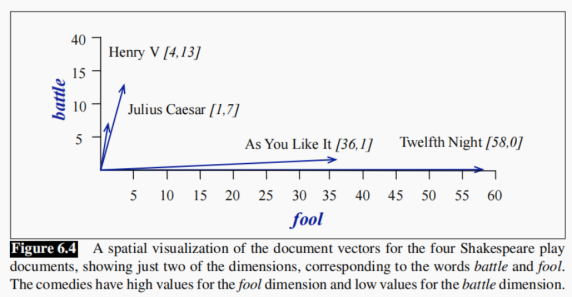
词-文档矩阵 (Term-Document Matrix)：

在词-文档矩阵中，每一行代表词汇表(vocabulary)中的一个词，每一列代表某个文档集合中的一个文档。这种矩阵用于表示特定文档中某个词出现的次数。



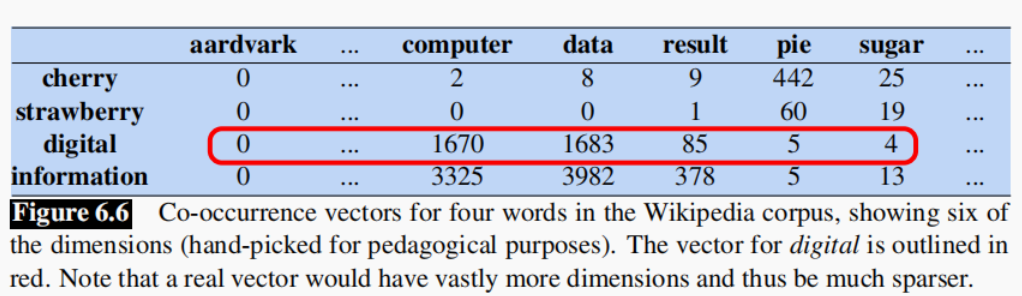
向量空间 (Vector Space)：

文档被表示为计数向量(count vector)，这是向量空间模型(vector space model)的一部分。一个向量(vector)本质上就是一系列数字的列表或数组，代表文档在不同维度上的特征。两个文档如果包含相似的词，则它们的列向量(column vectors)也会相似，这有助于根据查询向量(query vector)找到最匹配的文档。



词-词矩阵 (Term-Term Matrix)：

词-词矩阵，也称为词-项矩阵(word-word matrix)或词-上下文矩阵(term-context matrix)，其中列由词汇而不是文档标记。这种矩阵记录了行（目标）词和列（上下文）词在某个上下文中共同出现的次数，通常基于目标词周围的固定窗口(window)。



词向量的构建和意义：

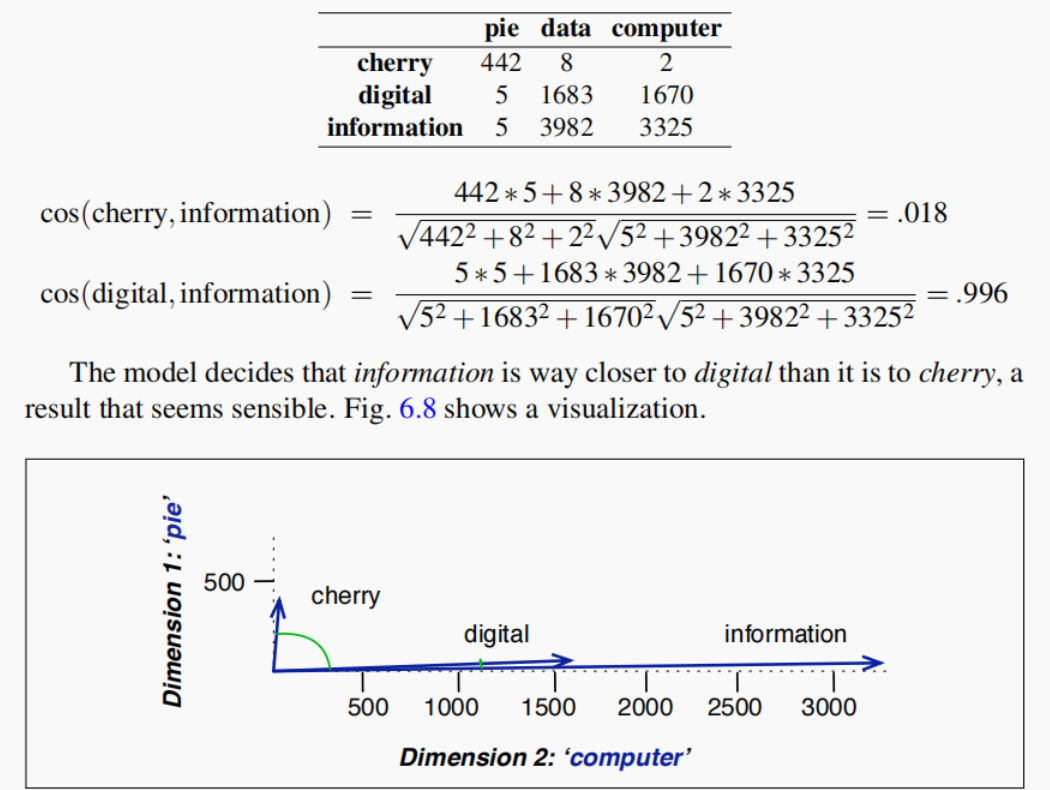
通过词-文档矩阵和词-词矩阵，我们可以构建代表词义的向量。如果两个词在相似的文档中出现，则它们的向量会相似。这种方法允许我们通过词在哪些文档中出现来捕捉一个词的含义，或者通过它们与其他词在特定上下文中共现的频率来表示词义。

6.4Cosine for measuring similarity余弦相似度

通过余弦相似度，我们能够在高维向量空间中有效地衡量词义的相似度

通过下面的例子可以很好的理解

通过余弦相似度，我们能够在高维向量空间中有效地衡量词义的相似度



6.5TF-IDF: Weighing terms in the vector

TF-IDF权重：

TF-IDF是两个术语的乘积：词频（Term Frequency，TF）和逆文档频率（Inverse Document Frequency，IDF）。

词频（TF）：

词频是指词t在文档d中的频率。它可以简单地使用原始计数tft,d=count(t,d)，或者更常见地，使用频率的对数来稍微压缩原始频率，这是基于出现100次的词并不意味着它对文档含义的相关性是出现1次的100倍。

逆文档频率（IDF）：

IDF用于给只在少数文档中出现的词更高的权重。这样的词对于区分这些文档与集合中其他文档非常有用。IDF是通过总文档数N除以词t出现的文档数dft来定义的idft=log10(N/dft)，一个词在越少的文档中出现，其IDF值越高。

TF-IDF计算：

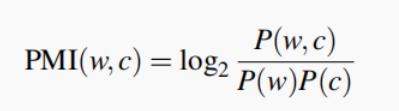
一个词在文档中的TF-IDF权重是它的TF值和IDF值的乘积wt,d=tft,d×idft。这种方法可以平衡词的出现频率和它的区分度，使得既不是过于常见也不是罕见的词被认为是重要的。

TF-IDF的应用：

TF-IDF权重用于信息检索中的权重共现矩阵，也在自然语言处理的许多其他方面发挥作用。它是一个很好的基线方法，通常是首先尝试的简单方法。

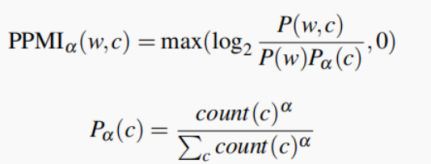
6.6Pointwise Mutual Information(PMI)点互信息

点互信息是一种衡量两个事件x和y共同出现的频率与它们独立出现的频率之比的度量。其公式为



其中P(x,y)是x和y共同出现的概率，P(x)和P(y)分别是x和y各自出现的概率。PMI给出了两个词共现超过随机概率的程度，PMI用于找出强关联的词。其值范围从负无穷到正无穷。正值表示两个词共现的频率超过了随机期望，负值表示共现频率低于随机期望。

正面点互信息（PPMI）：

由于负PMI值往往不够可靠，尤其是在语料库不是非常大的情况下，通常使用PPMI来代替，将所有负PMI值替换为零

PPMI特别适用于词-词矩阵，当向量维度对应于词而非文档时。

PPMI的计算：

通过给定的共现矩阵F，可以将其转化为PPMI矩阵，其中每个元素PPMIij给出了词wi与上下文cj的PPMI值。这个过程涉及到计算词和上下文的共现概率、边缘概率，然后利用这些概率计算PPMI。

PMI的偏见问题：

PMI倾向于对低频事件赋予很高的值。为了减少这种偏见，可以通过修改上下文概率P(c)的计算方法来调整，例如使用Pα(c)来增加稀有上下文的概率，从而降低它们的PMI值。另一种方法是使用拉普拉斯平滑（Laplace smoothing），在计算PMI之前给每个计数加上一个小常数k。

6.7Applications of the tf-idf or PPMI vector models

总结了之前提到的TF-IDF或PPMI的应用

构建向量语义模型：

描述的向量语义模型将目标词表示为一个向量，其维度对应于大型文档集合中的文档（词-文档矩阵）或某个邻近窗口中的词计数（词-词矩阵）。每个维度的值是计数，通过TF-IDF（对于词-文档矩阵）或PPMI（对于词-词矩阵）进行加权，向量是稀疏的（因为大多数值为零）。

计算词汇相似度：

通过计算两个词x和y的TF-IDF或PPMI向量的余弦值来计算相似度；余弦值越高，相似度越高。整个模型有时被称为TF-IDF模型或PPMI模型，以加权函数命名。

文档相似性应用：

TF-IDF模型常用于判断两个文档是否相似的文档功能。文档相似性对于信息检索、抄袭检测、新闻推荐系统，甚至数字人文学任务（如比较文本的不同版本以查看哪些彼此相似）都非常有用。

计算词相似性：

PPMI模型或TF-IDF模型可用于计算词相似性，用于任务如找到词的释义、追踪词义的变化，或在不同语料库中自动发现词的含义。

6.8Word2vec

这部分内容介绍了word2vec模型，一种生成词嵌入（embeddings）的方法，即用短密集向量表示词汇，与之前介绍的长稀疏向量（如基于词-文档矩阵或词-词矩阵的TF-IDF或PPMI加权向量）形成对比。密集向量在每个NLP任务中比稀疏向量表现更好。密集向量可能更好地捕捉同义词之间的关系，而且在表示词汇时需要学习的参数更少，有助于泛化和避免过拟合。

word2vec提供了一种强大的词表示方法：词嵌入。这些嵌入是短而密集的向量，其维数（d）通常在50到1000之间，与之前见过的词汇大小|V|或文档数量D相比，这是较小的维度范围。这些d维通常没有明确的解释，且向量是密集的，即向量条目是实数值，可能为负。

Skip-gram模型：

word2vec包含两种算法，这里介绍的是Skip-gram模型，它通过负采样（SGNS）进行训练。Skip-gram模型的直觉是，不是计算每个词w在特定词（如apricot）附近出现的频率，而是训练一个分类器来预测词w是否可能出现在apricot附近。Skip-gram模型利用运行文本作为隐式监督训练数据，通过将目标词w附近出现的词c作为“正确答案”来训练分类器，避免了对手动标注的监督信号的需求。

Skip-gram训练过程：

训练过程包括将目标词和相邻上下文词视为正例，随机采样其他词汇作为负例，使用逻辑回归训练分类器以区分这两种情况，并将学习到的权重作为嵌入。

其他静态嵌入模型：

除word2vec外，还有其他静态嵌入模型，如fastText和GloVe。fastText通过使用子词模型来处理未知词和词稀疏性问题，而GloVe（Global Vectors）基于全局语料统计的概率比率，结合了基于计数的模型和像word2vec这样的线性结构。

TF-IDF和word2vec两个技术的关联性：

结合使用：尽管TF-IDF提供了一种基于文档频率的权重计算方法，Word2Vec则通过学习词汇的向量表示来捕捉深层的语义关系，但这两种方法可以结合使用来提高某些NLP任务的性能。例如，在文档分类或信息检索任务中，可以使用TF-IDF来过滤或加权词汇，然后使用Word2Vec提供的密集向量作为机器学习模型的输入特征。

互补优势：TF-IDF在处理稀疏数据和识别关键词方面表现出色，而Word2Vec则擅长捕捉复杂的词义关系和同义词。通过结合这两种技术，可以同时利用它们的优势，提高文本分析的准确性和深度。

6.9Visualizing Embeddings

讨论了如何将词嵌入（word embeddings）进行可视化。词嵌入是一种高维度的词汇表示，理解和应用这些模型是NLP中的一个重要目标。内容提到了几种可视化词嵌入的方法：

查找最相似的词：

最简单的可视化方法是列出与目标词w最相似的词。这通常是通过计算目标词向量与词汇表中所有其他词向量之间的余弦相似度来完成，并排序找到最接近的几个词。例如，使用GloVe算法计算出的词嵌入，列出了与“frog”最相似的七个词。

层次聚类法：

另一种方法是使用聚类算法来显示词嵌入空间中哪些词与其他词相似。文中未标注的图表采用了层次聚类方法，展示了名词的词嵌入向量作为可视化的一种形式。

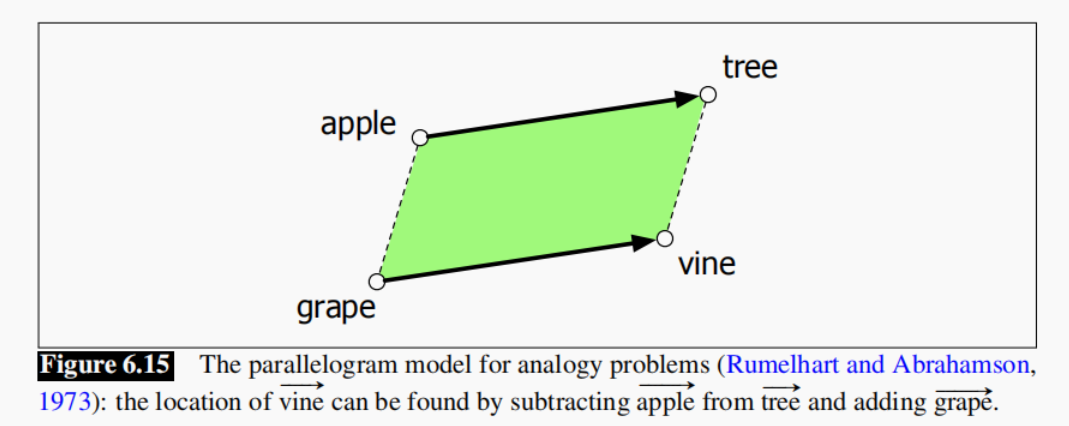
t-SNE降维：

最常见的可视化方法是将词的100维嵌入投影到2维空间中。文中提到了一种叫做t-SNE（t-distributed Stochastic Neighbor Embedding）的投影方法，它能够将高维数据有效地映射到低维空间。

6.10Semantic properties of embeddings词嵌入的语义属性

类比问题的解决方法：

对于类比问题，使用“平行四边形方法”，通过将一个词到另一个词的向量（如 tree−apple）添加到第三个词的向量（如 grape）上，找到与该点最近的词。



类比问题的挑战：

虽然词嵌入在涉及频繁词、小距离和特定关系（如国家及其首都或动词/名词及其变形）的任务中表现良好，但对于其他关系，平行四边形方法并不总是有效。并且，该方法可能过于简单，无法模拟人类形成类比的认知过程。

词嵌入与历史语义学：

词嵌入也可用于研究意义随时间的变化。通过从特定时间段编写的文本中计算多个嵌入空间，可以观察到过去两个世纪英语词汇意义的变化。

6.11Bias and Embeddings

词嵌入的社会影响：

词嵌入能从文本中学习词义，但不幸的是，它们也会再现文本中潜在的隐性偏见和刻板印象。例如，word2vec嵌入在新闻文本上的训练可能会显示出性别刻板印象，如将“计算机程序员”与“家庭主妇”关联起来，或者将“父亲”与“医生”、“母亲”与“护士”联系起来。

分配性伤害（Allocational Harm）：

这些偏见可能导致分配性伤害，即系统不公平地向不同群体分配资源（如工作或信贷）。使用词嵌入作为搜索潜在程序员或医生的工具可能会不公正地降低带有女性名字的文档的权重。

偏见放大（Bias Amplification）：

词嵌入不仅反映输入文本的统计特性，还会放大偏见。嵌入空间中的性别词汇比输入文本统计中更具性别化，偏见比实际劳动就业统计数据更夸张。

隐性关联测试（Implicit Association Test）：

隐性关联测试测量人们将概念（如“花”或“昆虫”）与属性（如“愉快”和“不愉快”）联系起来的倾向。研究表明，人们倾向于将非洲裔美国人的名字与不愉快的词汇联系起来，而将男性名字与数学、女性名字与艺术联系起来。

词嵌入中的隐性关联：

Caliskan等人（2017）使用GloVe向量和余弦相似度复现了这些隐性关联的发现，发现非洲裔美国人的名字与不愉快的词汇有更高的余弦相似度。

代表性伤害（Representational Harm）：

当系统贬低或忽视某些社会群体时，就会产生代表性伤害。任何使用词情绪的嵌入意识算法都可能加剧对非洲裔美国人的偏见。

减少偏见的研究：

最近的研究集中于尝试去除这些偏见，例如通过开发一个转换嵌入空间的方法，消除性别刻板印象，但保留定义性别，或者改变训练过程。然而，尽管这些去偏见方法可能减少了嵌入中的偏见，但并不能完全消除它，这仍然是一个开放的问题。

历史嵌入和偏见：

历史嵌入也被用来衡量过去的偏见。例如，研究者使用历史文本中的嵌入来测量职业与不同族群或性别名称的关联

6.12Evaluating Vector Models评估向量模型

外部评估 (Extrinsic Evaluation)：使用向量在NLP任务中的表现来评估，比较是否比其他模型有更好的性能。

内部评估 (Intrinsic Evaluations)：测试它们在相似度任务上的性能，计算算法的词相似度得分与人类评定的词相似度之间的相关性。

提出的相关问题：

词嵌入的评估：

如何有效地对比不同词嵌入模型（如word2vec, GloVe, fastText）在特定NLP任务上的性能？

词嵌入的语义属性：

如何使用词嵌入来探索和量化词汇的语义属性，例如同义词、反义词、上下位关系等？

长期语义变化的研究：

我们如何使用词嵌入来研究和追踪长期的语义变化，特别是在跨越几个世纪的文本语料中？

实际项目问题：

我在之前的实践项目中完成过一个使用bert对不同消息进行自动分类，模型能够学习并区分不同类别的句子，以实现自动分类。调整模型对句子进行分类的效果主要是通过对训练集调整来做到的，现在我想能否通过其它方式主动改变模型效果。