arXiv:2305.03144v1 [cs.LG] 4 May 2023

不同文本嵌入对自然语言处理聚类性能的影响

Rohan Saha

2023年4月

摘要

随着电子商务平台的兴起，评论对于客户评估产品的可信度至关重要。星级评分并不总是与客户撰写的评论文本相符。例如，三星评分（满分为五星）可能与评论文本不一致，而该评论文本可能更适合五星评分。我们可以采用聚类方法，将文本评论分组成不同的簇，从而重新标注正确的星级评分。在本研究中，我们探讨了如何选择不同的文本嵌入来表示这些评论，并进一步研究嵌入选择对不同类别聚类算法性能的影响。我们使用上下文相关（BERT）和非上下文相关（Word2Vec）文本嵌入来表示文本，并衡量它们对三类聚类算法的影响：基于划分的聚类（KMeans）、单链接凝聚层次聚类，以及基于密度的聚类（DBSCAN 和 HDBSCAN），每类算法均采用多种实验设置。我们使用轮廓系数、调整兰德指数分数和聚类纯度分数等指标评估算法性能，并讨论不同嵌入对聚类性能的影响。结果表明，所选嵌入类型会显著影响算法性能；不同类型的聚类算法之间性能差异很大；没有哪种嵌入类型优于其他类型；DBSCAN 的表现优于 KMeans 和单链接凝聚聚类，但也会将更多数据点标记为离群点。我们全面比较了不同算法的性能，并提出了诸多思路，以推动文本聚类领域的进一步研究。

关键词 ***rds*** 聚类 · 产品评论 · 机器学习 · 文本嵌入 s

1 引言

在数字时代，电子商务网站十分普遍，在线订购商品也十分便捷。电子商务网站设有反馈系统，客户可以在其中提交产品评论，通常包括星级评分和文本评论。这些评论旨在体现客户对特定产品的情感。产品评分较高，说明该产品达到了预期用途，进而增强客户对该产品的信任。此外，评分及相应的评论也有助于企业改进产品。然而，有时给出的评分并不能反映评论所表达的情感。例如，一条评分为两星的评论，其评论文本表达的意思可能更适合四星评分。换言之，评分与评论文本所表达的情感不匹配，可能会误导潜在客户，甚至损害企业利益。使用聚类算法可以帮助我们将情感相似的评论归为一组，并有望根据评论所表达的情感为其分配恰当的评分。但我们该如何为这类数据选择聚类算法？又该使用哪种数据表示方式来描述评论文本？本文将探讨这些问题。

在文本挖掘领域，聚类算法已被广泛用于发现数据中的潜在模式（[1, 16, 7, 6] 举几个例子）。这类研究聚焦于特定的文本聚类任务，并采用特定类型的数值向量来表示文本。目前，探索不同文本表示对不同类型聚类算法性能影响的研究仍然很少。换言之，针对特定类型的聚类算法，我们应如何选择文本表示？我们的工作朝这一方向迈出了关键一步。

在本研究中，我们比较了将不同类型的聚类算法应用于不同类型的文本表示（嵌入）时的性能。我们从 Amazon.com 的产品评论中获取文本数据，并使用不同类型的嵌入对其进行表示。针对每种嵌入类型，我们训练了四种聚类算法，即划分式聚类、层次聚类和两种基于密度的算法，并使用内部验证和外部验证技术比较各算法的性能。结果表明，文本表示的选择会对聚类算法的性能产生巨大影响，基于密度的算法可能优于其他类型的聚类算法。我们还发现，聚类算法确定的簇数并不总是等于预定义标签的数量（在我们的案例中，这些标签是为每条评论分配的评分）。所有数据和代码均可在 `https://github.com/simpleParadox/cmput_697_project` 获取

2 方法

在本节中，我们将介绍实验范式的不同组成部分。为简明起见，我们在图1中展示了实验框架的概览。

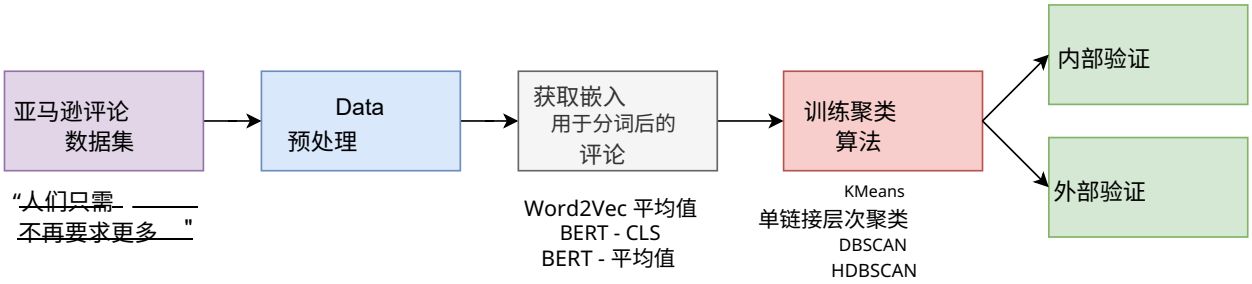


图1：实验框架图。首先，加载并预处理评论，其中包括对评论进行分词。随后，为每条评论获取不同的嵌入，并基于这些嵌入训练不同的聚类算法。为评估聚类算法的性能，我们使用内部验证（轮廓系数）和外部验证（调整兰德指数和簇纯度）。

2.1 数据集与预处理

在本研究中，我们使用亚马逊产品评论数据集[1]，该数据集包含1597个样本和27列数据，主要涉及亚马逊品牌的消费电子产品。我们删除了评论中缺少评分的样本。最终的数据集包含1177个样本。每个样本包含产品名称、产品品牌、评分星数、评论文本、评论标题等信息（有关各项特征的完整描述，请参阅数据集链接）。根据本研究的范围，我们仅使用评论文本、评论标题以及每条评论附带的星级评分。星级评分的范围为1至5。我们将评论标题与评论文本拼接起来，以表示数据集中的样本。通过拼接，评论标题的信息得以纳入评论的整体信息中。此外，我们还将每个拼接后的样本截断，使其最多包含512个词元，因为语言模型的输入最大长度有此要求（详见第2.2节）。我们在图2中展示了预处理步骤，并在图3中展示了评分的分布。我们观察到，数据集中的评分并非均匀分布。

[1]https://www.kaggle.com/datasets/yasserh/amazon-product-reviews-dataset

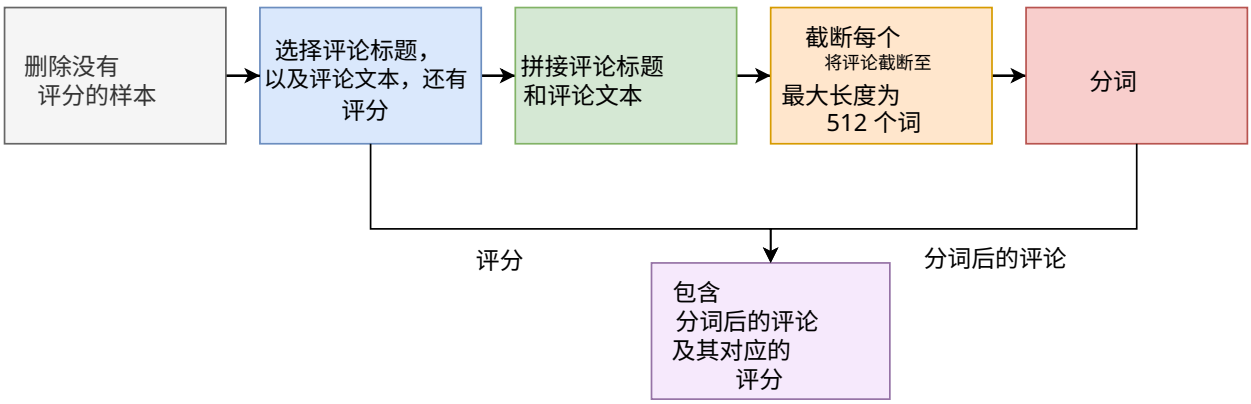


图2：预处理框架展示了数据清洗所涉及的步骤。首先，我们移除没有评分的评论。然后，选取评论标题和评论文本，并将二者拼接起来。我们将拼接后的样本截断，使其长度最多为512个词。最后，我们对输入进行分词，以获得文本表示。随后，形成一个包含这些数据的数据集。

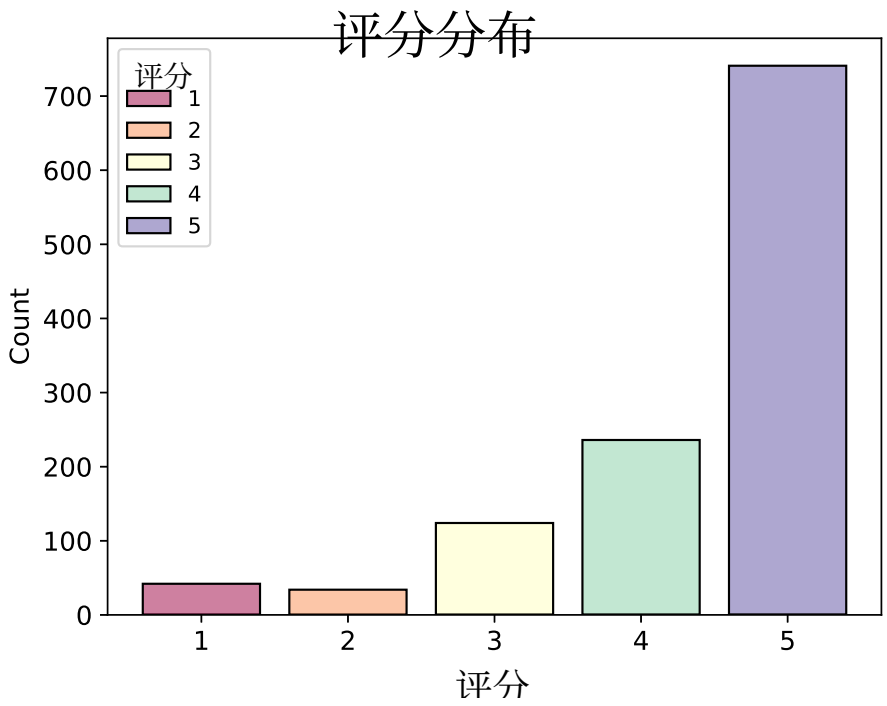


图3：数据集中的评分分布。数据集中的大多数评论都倾向于较高评分。

2.2 嵌入

我们的目标是研究不同类型的文本嵌入对聚类算法性能的影响。为此，我们使用两个语言模型：预训练的 BERT [4]

（在 BookCorpus 数据集和英文维基百科数据集上预训练）以及预训练的 Word2Vec [12]（在 Google 新闻数据集上预训练）来表示数据集中的每条评论。我们使用 Gensim [14] 提供的 Word2Vec 实现，以及 Huggingface[2]提供的 BERT 实现。对于每个样本，我们先将评论标题和评论文本拼接起来，再将其输入语言模型。对于 Word2Vec，我们先对样本进行分词，然后从语言模型中获取每个词元（单词）的隐藏向量（嵌入）。每个隐藏向量的维度为 300。最后，我们对每个词元的隐藏向量取平均值，为每个评论样本得到一个 300 维的向量。Word2Vec 嵌入无法捕捉评论中存在的上下文信息。因此，我们还使用 BERT 来考虑评论文本中的上下文变化，因为先前的研究表明，BERT 嵌入优于其他文本嵌入 [16]。要获取 BERT 嵌入，我们使用预训练的 BERT 分词器对样本进行分词，然后将分词结果输入 BERT 模型。我们使用两种类型的 BERT 嵌入来表示每条评论。首先，我们使用 `<CLS>` 标记来表示评论，该标记是一个 768 维的向量，能够综合考虑输入评论文本中的所有词元。其次，我们获取每个词元的最后隐藏状态（768 维）向量，并对这些向量取平均值，得到一个 768 维的向量。对于每种嵌入，我们使用 scikit-learn [2] 中的 StandardScaler 对数据进行 z 分数标准化。总而言之，我们使用三种类型的嵌入。

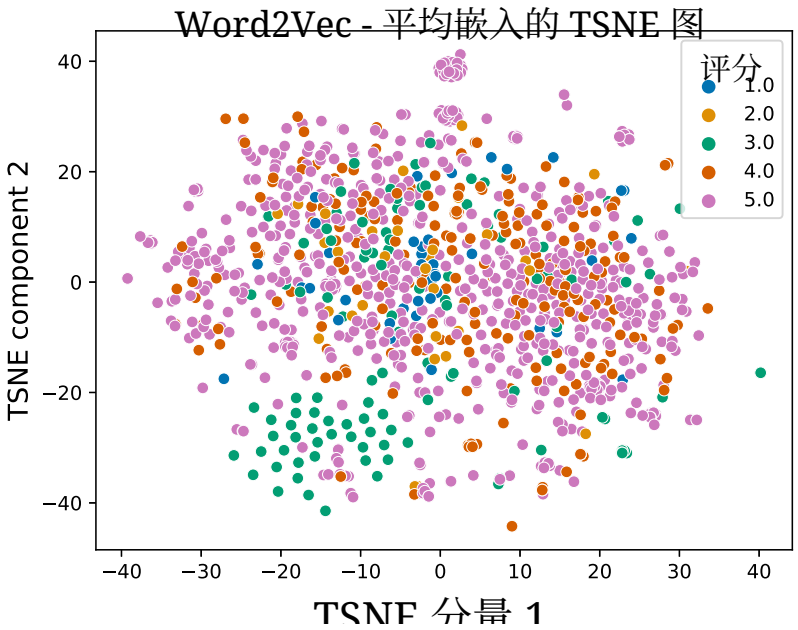
- Word2Vec - 平均值 - 300 维。
- BERT - CLS - 768 维。
- BERT - 平均值 - 768 维。

每种嵌入类型对文本的编码方式各不相同。此外，由于嵌入的维度数量不同，它们在空间中的分布也有所不同。为了直观展示嵌入在空间中的分布，我们使用一种降维技术，将嵌入绘制在二维空间中。我们对每种嵌入类型应用 t-SNE [9]，将其降至 2 个分量。图 4 展示了每种嵌入对应的图。我们观察到，每种嵌入类型在空间中的分布各不相同。数据的这一特性可能导致不同聚类算法的性能有所差异。

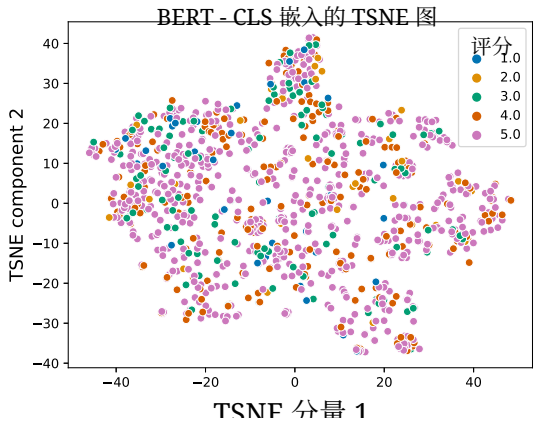
2.3 聚类算法

为了在评论的向量表示中发现模式，我们应用四种聚类算法，并比较各算法的性能。我们使用 KMeans [8, 10], 单链接

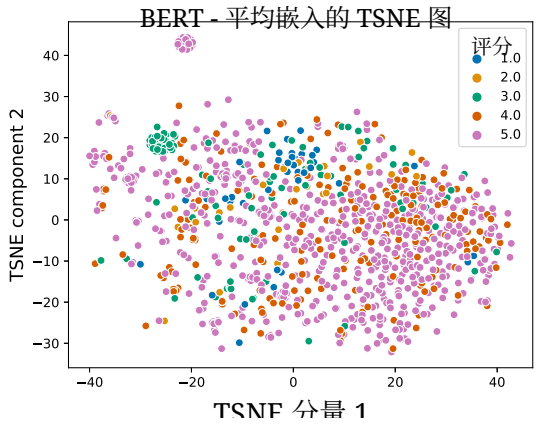
[2]https://huggingface.co/bert-base-uncased



(a)



(b)



(c)

图 4：本研究使用的不同嵌入类型的二维 TSNE 图。图 4a 展示了 Word2Vec 嵌入的分布，图 4b 展示了 BERT - CLS 嵌入的分布，图 4c 展示了 BERT - 平均嵌入的分布。这些嵌入在空间中的分布均不相似。

凝聚层次聚类 [13] 分别作为划分式聚类和层次聚类方法，DBSCAN [5], 和 HDBSCAN [3] 则作为基于密度的聚类方法。对于 KMeans、单链接凝聚层次聚类和 DBSCAN，我们使用 scikit-learn [2]中的实现。对于 HDBSCAN，我们使用 `hdbscan` 库 [11]中的实现。

2.4 评估

已有大量验证指标被提出，用于评估聚类算法的性能（综述见 [18]和 [19] ）。验证指标通常分为内部验证和外部验证。对于内部验证，我们使用轮廓系数[3]。对于外部验证，我们使用调整兰德指数[4] 和簇纯度，后者衡量一个簇中包含单一类别的程度。

2.5 实验设置

我们的数据集包含文本评论，其评分范围为 1 到 5。对于 KMeans 和单链接凝聚层次算法，我们设计了两种实验方案。首先，将簇数设为 5（每个评分值对应一个簇），并将各种聚类算法应用于该数据集。其次，假设数据集中有 3 个簇，并应用聚类算法。我们通过以下转换将评分转换为三个类别标签。

$$\text{rating} = \begin{cases} 1 & \text{if rating} < 3 \\ 2 & \text{if rating} = 3 \\ 3 & \text{if rating} > 3 \end{cases}$$

值得注意的是，对于 KMeans 和凝聚层次聚类，我们仅将簇数指定为超参数。

对于 KMeans，我们使用 k-means++ 方法初始化算法，并从各点对总体惯性[5]的贡献所形成的经验概率分布中，采用基于采样的方法选取初始簇质心。我们将最大迭代次数设为 300。对于 DBSCAN，我们调整超参数 `epsilon` ϵ，该参数控制两个样本被视为彼此处于邻域内时的最大距离； `min samples`则固定为 5，用于定义一个点被视为核心点时其邻域内的样本数量。对于 HDBSCAN 算法，我们只调整参数`min cluster size`，该参数控制形成一个簇所需的最少点数。我们在结果部分（第 3 节）讨论这些超参数的取值。对于所有算法，我们均使用欧氏距离度量。

[3]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score. html[4] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html[5] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

3 结果与讨论

不同类型的嵌入可能会影响不同聚类算法的性能，本节将对此进行探讨。

3.1 KMeans

我们首先探讨选择不同嵌入类型对 KMeans 算法的影响。KMeans 算法的一个关键超参数是簇数，因为该算法要求用户指定数据中的簇数。评论数据集中的评论被划分为五个类别（评分）。然而，这种分类未必会反映为嵌入空间中的簇数。因此，问题来了：簇数应设为多少？选择最佳簇数的一种常用方法是肘部法（或膝点法）[17]，即将惯性（簇内平方和）绘制在纵坐标（y 轴）上，并将簇数绘制在横坐标（x 轴）上；图中出现“拐点”或“膝点”的位置即被选为最佳簇数。然而，我们绘制惯性与簇数的关系图后，并未发现明显的拐点。因此，我们采用轮廓系数法来确定最佳簇数。

轮廓系数通常用于聚类算法的内部验证，但也可以用于选择 KMeans 算法的簇数（`n clusters`）[15]。我们测试了一系列取值（3 - 19），并绘制了每个取值对应的轮廓系数。由于 KMeans 算法的性能还取决于簇质心的初始化方式，因此，对于每个 `n clusters`，我们使用 1 - 50 之间的随机种子值来初始化簇质心。在图5中，我们展示了每种嵌入类型下 KMeans 的内部验证分数，_ 这些分数是对多个随机种子值取平均后得到的，曲线同时显示了均值标准误。

图表中的每条曲线代表一种嵌入类型。红色曲线表示使用 BERT - CLS 嵌入训练 KMeans 算法时的结果，蓝色曲线表示 BERT - 平均嵌入的结果，蓝色曲线表示 Word2Vec - 平均嵌入的结果。我们观察到，当簇数为 3 时，三种嵌入的轮廓系数均达到最大值，并且随着簇数增加，该系数趋于稳定。这一结果表明，最佳簇数为 3。然而，我们的数据集有五个评分（类别）。由此引出一个问题：数据是否属于三个簇，而不是五个？为此，我们使用外部验证来评估每种嵌入类型下 KMeans 算法的性能。

我们在图6中展示了三簇和五簇情况下的外部验证分数。

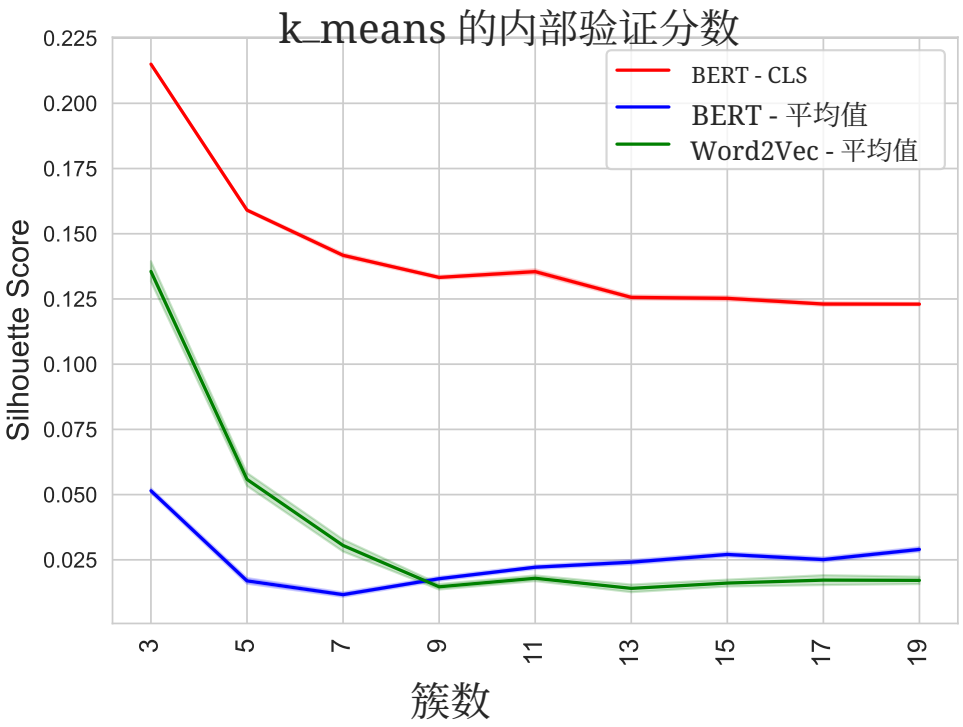


图5：不同簇数下 KMeans 的内部验证分数。由于 KMeans 的初始化质心会影响算法性能，我们使用范围为 1-50 的随机种子值运行该算法。此处展示的是平均结果，曲线上的阴影区域表示均值标准误。

BERT - CLS 嵌入的调整兰德指数优于 BERT - 平均值和 Word2Vec - 平均值。尽管这一结果与内部验证分数的结果一致，但分数之间的差异并不显著。从外部验证结果来看，无法明确判断选择 3 个簇还是 5 个簇会带来更好的模型性能。此外，三簇和五簇情况下的调整兰德指数都接近于零，表明聚类质量较差，因此有必要对聚类算法进行评估。

我们还使用簇纯度作为评估方法，以衡量一个簇在多大程度上只包含一个类别。图7展示了簇纯度的结果。需要注意纯度值的量级。簇数接近3或5时，纯度值最低。随着簇数增加，纯度值也随之提高。这是因为簇数越多，每个簇包含的点就越少。当簇数等于3时，轮廓系数最高（见图5）。纯度分数也表明，KMeans算法识别出的簇可能质量较差。

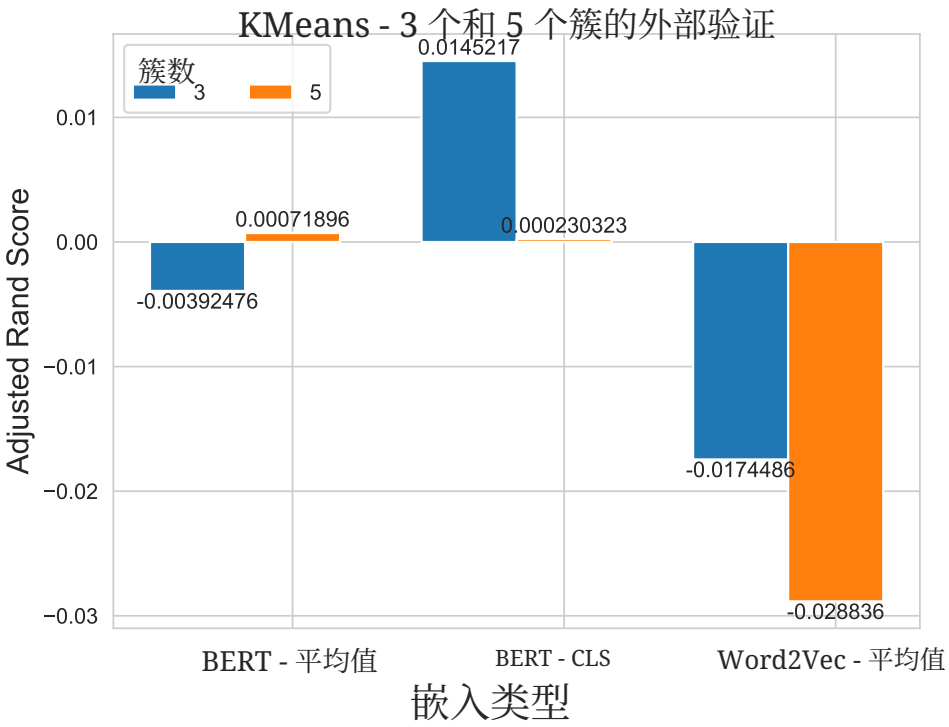


图6：KMeans 在五十个随机种子值（种子 1-50）上的外部验证结果取平均值。对于 n 个簇，BERT - CLS 嵌入 $= 3$ 似乎优于其他嵌入。不过，这些数值接近于零，表明聚类质量较差。

总的来说，如果使用KMeans对文本嵌入进行聚类，BERT - CLS可以作为嵌入选项，但必须谨慎，因为无法保证获得最优结果。下一节将探讨凝聚式单链接聚类算法对文本嵌入进行聚类的性能。

3.2 单链接凝聚层次聚类

尽管我们的数据集包含被划分为 5 个类别（评分）的评论，但 K均值分析（见第 3.1 节）表明，数据可能属于不同数量的簇。通常在簇数事先未知时使用层次聚类，并可在树状图上应用距离阈值（截断阈值）以获得固定数量的簇。为简单起见并保持一致，我们只调整一个超参数。我们为 `n clusters` 指定一个取值范围，算法会自动确定截断阈值，以获得相应的簇数。

-

与第3.1节的分析类似，我们将超参数`n clusters`的取值范围设为3 - 19。图8展示了内部验证结果。当簇数为3时，BERT - CLS的单链接凝聚聚类算法优于其他嵌入；

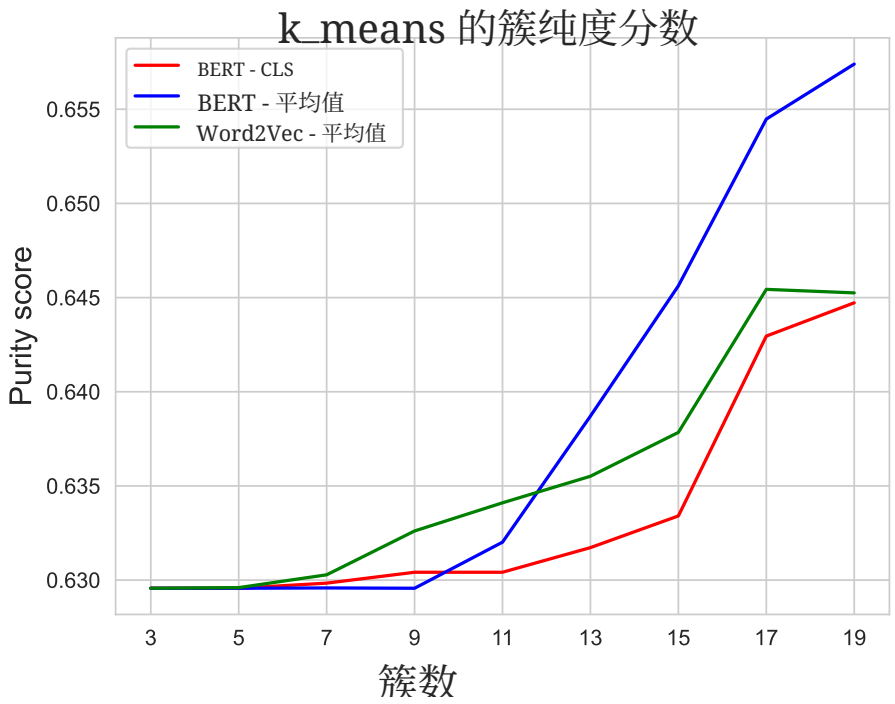


图7：KMeans 算法的簇纯度分数。随着簇数增加，纯度分数也随之提高。然而，约为 0.65 的纯度分数表明聚类质量不够理想。

然而，这种优势只是暂时的，当簇数为5时，轮廓系数会迅速下降。我们还观察到，使用BERT - 平均嵌入训练模型时也会出现类似现象。另一方面，使用Word2Vec - 平均嵌入时，当簇数大于3，模型的轮廓系数高于使用BERT - CLS和BERT - 平均嵌入训练所得的轮廓系数。这一结果表明，Word2Vec嵌入在空间中的分布方式可能使聚类算法能够找到凝聚度良好、且彼此相对分离的簇。然而，由于单链接层次聚类在合并两个簇之前只考虑它们之间距离最近的点，因此可能导致簇以链状方式连接。即使簇数增加，这也可能使轮廓系数保持相对稳定。

比较单链接凝聚层次聚类的轮廓系数与 KMeans 分析中观察到的轮廓系数时，单链接凝聚聚类的轮廓系数高于 KMeans 算法得出的结果。不过，这可能表明单链接凝聚聚类的表现更好。

与 KMeans 相比，图9中的外部验证结果表明聚类质量较差，因为调整兰德指数接近于零。

图10中观察到的簇纯度分数也表明单链接凝聚聚类算法的聚类质量较差。与 KMeans 的结果类似，纯度分数介于 0.60 至 0.67 之间，这进一步证明单链接凝聚层次聚类算法并不擅长在底层数据空间中发现簇。

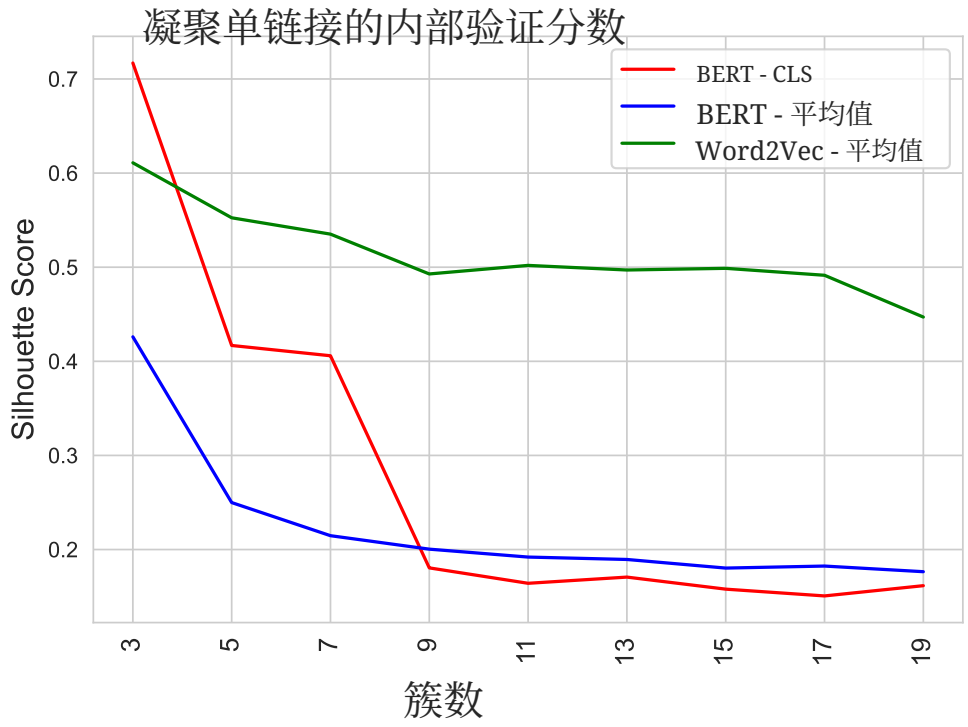


图8：单链接凝聚层次聚类的内部验证分数

3.3 DBSCAN

除了探索 KMeans 和单链接层次聚类之外，我们还希望应用基于密度的算法。为了评估基于密度的算法，我们只使用轮廓系数作为内部验证指标。如前文所述，我们只调整 epsilon 超参数的值，并将 `min samples` 超参数的值固定为 5。对于 DBSCAN算法，结果如图11所示。我们观察到，较低的 epsilon 值对应的轮廓系数接近 1.0，这表明 DBSCAN 能够找到定义良好的簇，其性能优于 KMeans 和单链接凝聚聚类。然而，DBSCAN 识别出的簇数并不等于

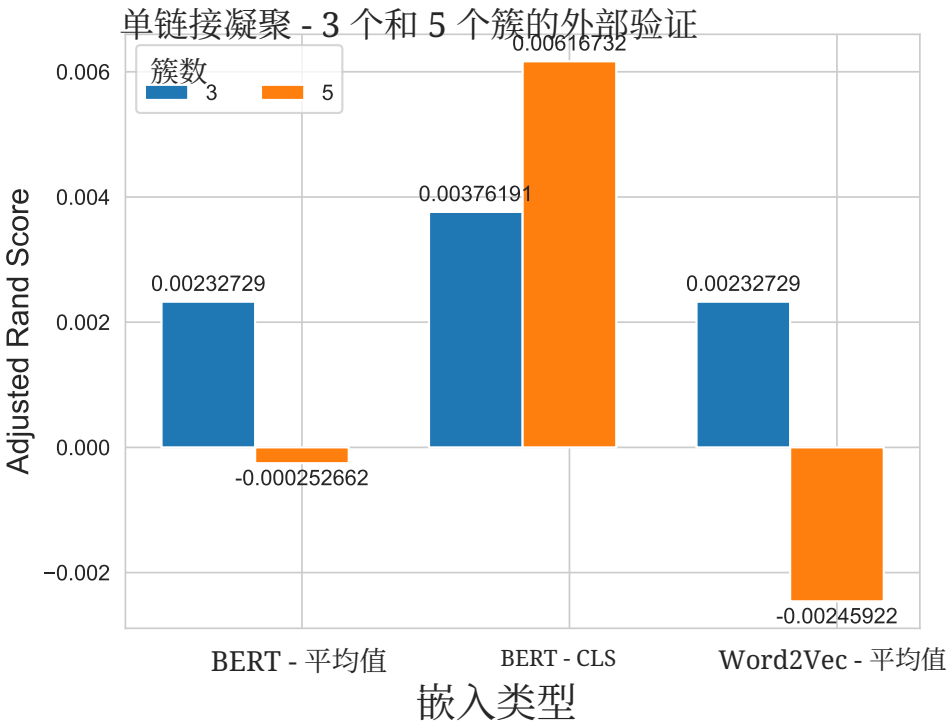


图9：外部　　验证分数：单链接凝聚层次聚类

KMeans 和单链接凝聚聚类识别出的簇数。我们在第 3.1 节和第 3.2 节中看到，簇数较少（约为 3 个）时，轮廓系数较高。另一方面，对于 DBSCAN，轮廓系数较高时，我们观察到的簇数也更多。当 epsilon=0.5 时，BERT - 平均嵌入有 13 个簇，BERT - CLS 嵌入有 16 个簇，Word2Vec - 平均嵌入有 22 个簇。这些结果表明，对于同一数据应用不同算法时，模型性能并不总是一致的。

当 epsilon 设为 15.0 时，轮廓系数的变化也呈现出一个有趣的现象。BERT - CLS 和 Word2Vec - 平均值的轮廓系数大幅下降。然而，BERT - 平均值的轮廓系数并未大幅下降。这可能是因为 BERT - 平均嵌入在数据空间中的底层分布，使算法能够识别出比 BERT - CLS 和 Word2Vec - 平均嵌入定义得更清晰的簇。因此，可以进一步研究使用 BERT - 平均嵌入通过 DBSCAN 对文本评论数据进行聚类的可能性。

这些结果或许表明，数据中确实存在更多簇。但在得出结论之前必须谨慎，因为每个已识别簇中的点数可能很少。值得注意的是，在训练DBSCAN算法时，我们将

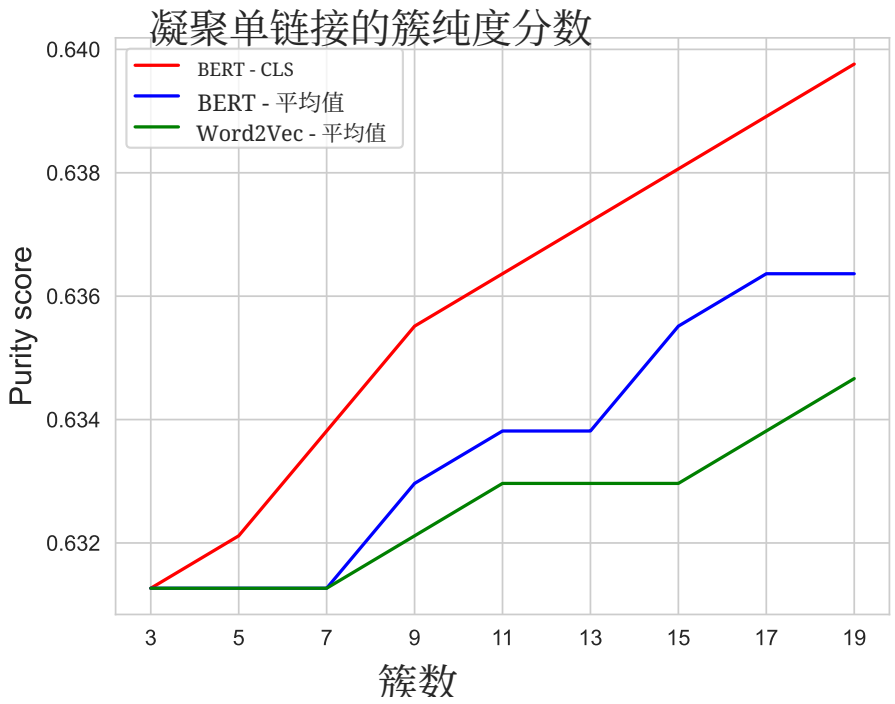


图10：DBSCAN 的簇纯度分数。对于每种嵌入类型，簇纯度都会随着簇数的增加而提高。

`min samples` 的值固定为 5，这可能影响了结果。此外，我们还观察到，当 ϵ 设为 0.5 时，许多点被识别为噪声/离群点。我们在表1中列出了 $\epsilon = 0.5$和 $\epsilon = 15.0$ 的噪声点数量。因此，每个簇中的点都很少。

嵌入类型	$\epsilon = 0.5$	$\epsilon = 15.0$
BERT - CLS	1056	574
BERT - 平均值	1099	931
Word2Vec - 平均值	973	376

表1：当 ϵ 值为 0.5 和 15.0 时，DBSCAN 的噪声点数量 .

每种嵌入类型的内部验证分数和离群点数量表明，尽管轮廓系数看起来可能很高，但离群点的存在并不一定意味着 DBSCAN 算法更好于 KMeans 或单链接凝聚层次算法。对此类结果的一种可能解释与嵌入空间中数据点的分布有关，我们将在局限性部分（第4节）中对此进行更详细的讨论。

图12所示的簇纯度结果进一步支持了这样一种观点：

当 epsilon 较低时（并将 `min samples` 固定为 5），生成的簇较小，且大多数数据点都被标记为噪声点；由于簇纯度仅根据已识别的簇计算，因此其值较高。此外，当 $\epsilon = 15.0$ 时，噪声点的数量会减少，表明更多数据点被分配到某个簇中，但图11中观察到的轮廓系数下降表明聚类性能较差。

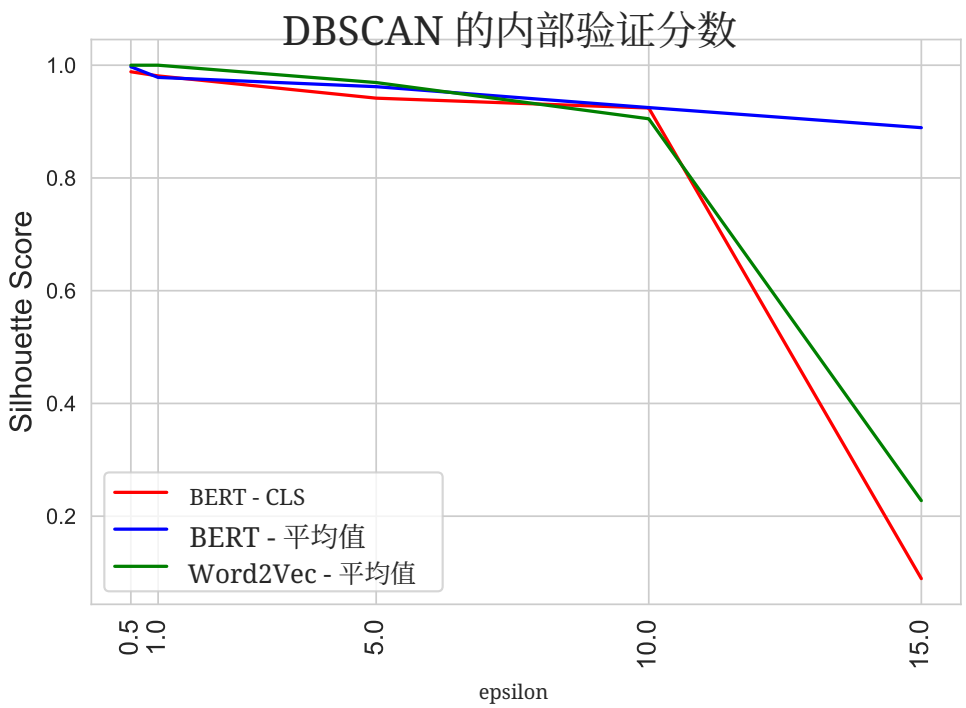


图11：DBSCAN算法在各种epsilon值下的内部验证分数。选择的epsilon值为 [0.5、1.0、5.0、10.0、15.0]。

总而言之，结果表明，DBSCAN算法的性能与嵌入的分布、 ϵ 的取值以及 `min samples` 的取值密切相关。尽管可以使用轮廓系数和簇纯度分数来评估 DBSCAN 的性能，但还必须考虑计算噪声点的数量，因为算法识别出的噪声点相对比例会影响对结果的最终解读。

3.4 HDBSCAN

DBSCAN 的一个缺点是无法识别密度各异的簇。正如第 2.2 节所述，嵌入在空间中的分布密度可能存在差异。

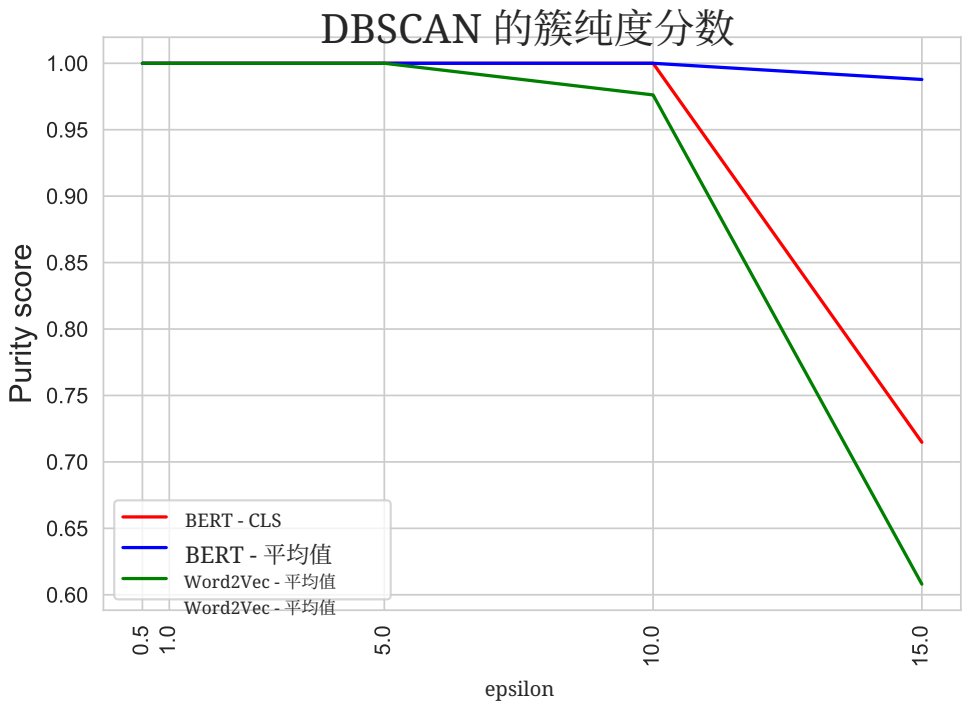


图12：DBSCAN算法的簇纯度分数。

因此，我们还分析了 HDBSCAN 对各种嵌入类型进行聚类的性能。图13展示了 HDBSCAN 的内部验证结果。

为简单起见，我们仅调整 HDBSCAN 模型的超参数 `min cluster size`，该参数控制簇的最小大小；低于此大小的簇中，所有点都会被视为噪声点。其他所有超参数均保持各自的默认值（参见第 2.3 节）。我们观察到，当模型使用 Word2Vec - 平均值和 BERT - 平均值训练时，`min cluster size` 为 10 时轮廓系数达到最大值。此外，对于使用 BERT - 平均嵌入训练的模型，当 `min cluster size` 大于 10 时，其轮廓系数急剧下降。然而，使用 BERT - CLS 嵌入训练模型时并未观察到这种影响。BERT - CLS 的轮廓系数在 `min cluster size` 的所有取值下都相对稳定。这些结果表明，使用 HDBSCAN 等算法时，与 BERT - CLS 嵌入相比，BERT - 平均值或 Word2Vec - 平均值等平均嵌入可能更适合对文本嵌入进行聚类。

轮廓系数能否反映聚类质量？我们在图14中观察到，使用 BERT - 平均值和 Word2Vec - 平均值嵌入时，HDBSCAN 簇的纯度高于使用 BERT - CLS 嵌入时的情况。总体而言，所有嵌入的聚类纯度分数都在 0.8 以上（但使用 BERT - 平均值时，当 `min cluster size` 值大于

10 后除外），这表明这些簇的整体质量较好。然而，比较 DBSCAN 和 HDBSCAN 识别出的簇数时，观察到了有趣的结果。HDBSCAN 识别出的簇数少于 DBSCAN。以下列出了每种嵌入对应的 HDBSCAN 簇数（当 `min cluster size` 为 10 时）。

- BERT - CLS $= 9$
- BERT - 平均值 $= 7$
- Word2Vec - 平均值 $= 7$

即使簇数较少，轮廓系数和簇纯度分数仍相对较高，这表明 HDBSCAN 在文本嵌入聚类方面可能更加稳健。不过，在得出结论前仍需谨慎，因为 HDBSCAN 识别出的噪声点数量可能会抬高结果。事实上，我们观察到，当 `min cluster size` $= 10$ 时，HDBSCAN 识别出的噪声点数量很高；对于 BERT - CLS 嵌入、BERT - 平均值嵌入和 Word2Vec - 平均值嵌入，HDBSCAN 识别出的离群点分别为 945、1046 和 1048 个。HDBSCAN 在表现最佳的超参数下识别出的离群点数量，与 HDBSCAN 识别出的离群点数量相当。换言之，这两种基于密度的算法似乎并没有哪一种明显优于另一种。

综合考虑上述所有算法，DBSCAN 和 HDB-SCAN 的表现似乎优于 KMeans 和单链接凝聚层次聚类算法。不过，必须计算基于密度的算法识别出的离群点数量，因为这可能会影响比较分析的最终结论。

4 局限性与未来工作

我们的研究存在一些潜在局限性。我们使用的数据集范围相当有限，主要包含某一个品牌的消费电子产品样本。因此，如果使用规模更大、更多样化的数据集，聚类算法的性能可能会出现细微差异。未来可扩展本研究，对不同类型的数据集进行全面比较（包括其他类型的物品、多种语言等）。

我们没有考虑评论的人口统计特征。同一产品的评论可能以多种语言存在，而现有语言模型可能会以不同方式表示不同语言中的同一条评论，这可能导致聚类算法的性能出现差异。换言之，文本最佳表示方面的进展是一个瓶颈，阻碍着.....

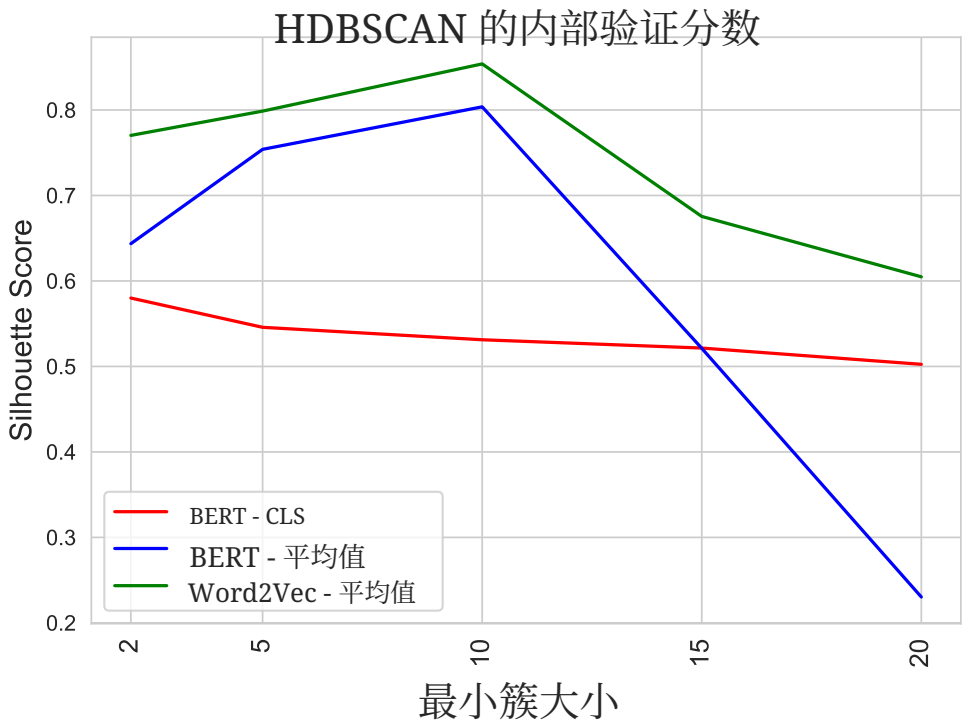


图13：HDBSCAN算法的内部验证分数。超参数 `min cluster size` 的选定值为 $[2, 5, 10, 15, 20]$。 – –

评估嵌入对聚类算法的影响。这可以视为推动更多关于多语言文本表示对齐研究的动机。

用户出于多种原因为产品指定星级评分，而产品也可能因多种原因（包装不佳、不喜欢某项特定功能等）而被评为一星。在本项目中，对于给定评分的所有评论，无论该评分对应的数据集中评论文本或原因是什么，我们都将其视为同等重要。然而，客户会出于众多原因赋予产品特定的评分值。例如，某位客户可能因为喜欢产品的电池容量而给出 5 分；另一位客户则可能因为产品尺寸小巧而给出 5 星评分。这可能是我们的外部验证得分接近于零的原因之一，因为采用 1—5 的评分范围进行评分可能不够充分，或者范围过小。此外，第 3.3 节和第 3.4 节中的基于密度的算法所检测出的簇数支持这样一种观点：底层评论数据集中存在超过 5 个簇。此外，由于 DBSCAN算法将许多点识别为噪声点，因此数据空间中可能还存在更多底层簇。未来仍需进一步研究，以评估评论数据上的聚类算法，同时考虑簇数可能多于预定义标签数量这一情况。

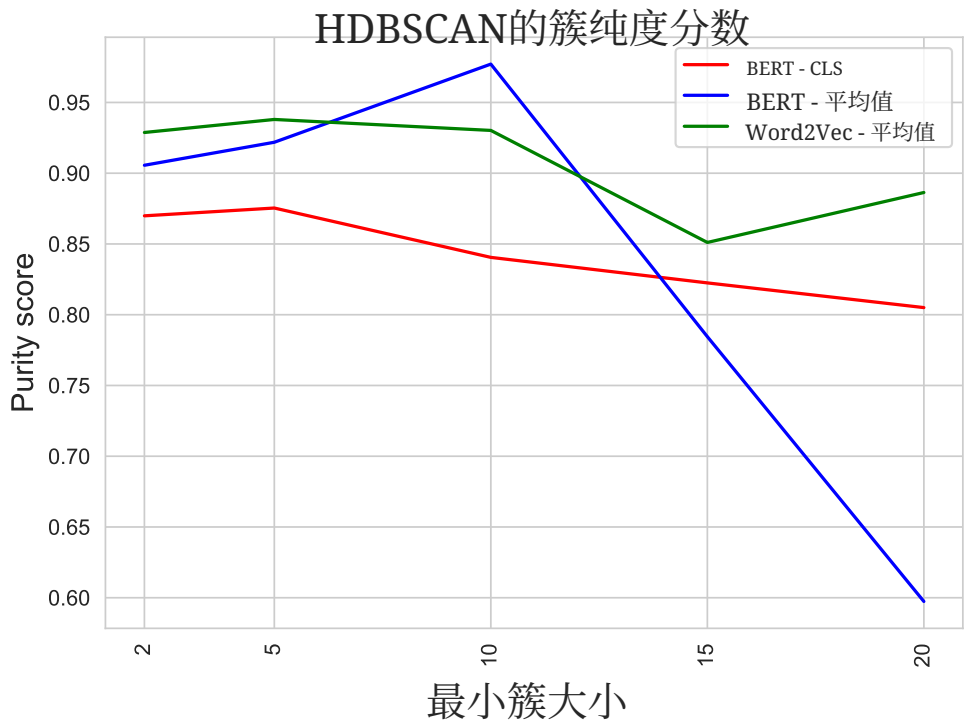


图14：HDBSCAN算法的簇纯度分数。超参数 `min cluster size` 的取值为 [2, 5, 10, 15, 20]。

本研究采用了三种不同的评估指标，因为仅基于单一指标来解释聚类质量可能并不准确，尤其是在处理复杂或高维数据时。然而，所有指标都各有优势和局限性。因此，未来也可以采用专门为文本嵌入设计的更复杂指标，或结合现有指标来评估聚类结果。

在训练聚类算法时，为简化流程，我们只对少量超参数进行了调优。更精细的超参数调优将有助于我们更深入地了解聚类算法在高维文本嵌入下的性能。未来的研究可以通过为各种算法调优多个超参数，进一步探索本文提出的思路。

为了表示文本评论，我们使用了从预训练模型中获得的嵌入（在 Google 新闻数据集上预训练的 Word2Vec，以及在 BookCorpus 和英文维基百科数据集上预训练的 BERT 模型），这些嵌入可能并不是评论数据集数据的最佳表示。换言之，对预训练语言模型在评论数据集上进行微调，或许能够得到更适合聚类的嵌入。未来，研究人员可以考虑在多个多样化的数据集上对语言模型进行微调，以获得更准确地表征文本评论的嵌入——

在多个不同的数据集上微调语言模型，以获得能够比预训练模型更准确地表示文本评论的嵌入。

最后，需要注意的是，算法的选择对于文本数据聚类至关重要，因此需要进行谨慎规划。例如，我们的结果显示，DBSCAN的性能优于KMeans和单链接层次聚类算法，但DBSCAN也识别出了大量噪声点。随着 ϵ 值的增加，噪声点的数量会减少，但轮廓系数和聚类纯度分数也会降低（除BERT - 平均值之外的所有嵌入均如此）。这些结果可能有助于推动相关研究，开发专门用于聚类嵌入的聚类算法。尽管存在一些局限性，我们的工作也提出了许多值得未来深入探讨的思路，以推动文本聚类领域的研究。

5 结论

在本研究中，我们考察了用于文本嵌入聚类的各种类型算法的性能，以及选择不同类型的嵌入如何影响聚类性能。我们使用了一个包含Amazon.com上列出的产品评论的数据集，其中每条评论均被赋予1 - 5的评分。我们使用Word2Vec和BERT嵌入表示这些评论。我们采用了KMeans、单链接凝聚层次聚类、DBSCAN和HDBSCAN。我们使用轮廓系数、调整兰德指数和簇纯度等各种指标对每种算法进行了评估。我们观察到，基于密度的算法通常似乎优于KMeans和单链接层次聚类算法。此外，基于密度的算法所得结果表明，数据集中的潜在簇数量可能多于数据集中的标签数量（对于该数据集而言为3或5个）。然而，我们的结果也凸显了对高维文本数据进行聚类所面临的挑战，例如相同评分值对应的嵌入在空间中的分布存在差异。在比较聚类算法时，还必须谨慎得出结论，因为不同的超参数设置可能会对结果产生影响。例如，调整DBSCAN算法的 ϵ 值以及HDBSCAN的 `min cluster size` （第3.3节），会导致算法识别出的噪声点数量不同。这会影响聚类算法的性能。

总而言之，本工作提供了一个概念验证，针对不同文本嵌入对聚类算法性能的影响开展了比较分析。我们希望，这项工作能够为文本聚类领域的研究以及开发能够充分表示文本数据的新嵌入方法开辟新的方向。

参考文献

[1] Charu C. Aggarwal 和 ChengXiang Zhai。《文本聚类算法综述》。载于：《文本数据挖掘》。Springer US，2012年，第 77–128 页。 DOI： 10.1007/978-1-4614-3223-4_4。 URL： https://doi.org/10.1007/978-1-4614-3223-4_4。[2] Lars Buitinck 等。《机器学习软件的 API 设计：scikit-learn 项目的经验》。载于：《ECML PKDD 研讨会：数据挖掘与机器学习语言》。2013年，第 108–122 页。[3] Ricardo J. G. B. Campello、Davoud Moulavi 和 Joerg Sander。《基于层次密度估计的基于密度的聚类》。载于：《知识发现与数据挖掘进展》。Springer Berlin Heidelberg，2013年，第 160–172 页。 DOI： 10.1007/978-3-642-37456-2_14。 URL： https://doi.org/10.1007/978-3-642-37456-2_14。[4] Jacob Devlin 等。BERT：语言理解的深度双向 Transformer 预训练。2019年。arXiv： 1810.04805 [cs.CL]。[5] Martin Ester 等。《一种利用噪声在大型空间数据库中发现簇的基于密度的算法》。载于：《第二届知识发现与数据挖掘国际会议论文集》。KDD'96。俄勒冈州波特兰：AAAI 出版社，1996年，第 226–231 页。[6] Rafael Gallardo Garcia 等。《文本聚类任务中聚类算法的比较》。载于：Computacion y Sistemas 24.2（2020年6月）。 DOI： 10.13053/cys-24-2-3369。URL： https://doi.org/10.13053/cys-24-2-3369。[7] Radulescu Iulia-Maria 等。《使用文档嵌入的基于密度的文本聚类》。2020年11月。[8] S. Lloyd。《PCM 中的最小二乘量化》。载于：IEEE 信息论汇刊 28.2（1982年），第 129–137 页。 DOI： 10.1109/TIT.1982.1056489。[9] Laurens van der Maaten 和 Geoffrey Hinton。《使用 t-SNE 可视化数据》。载于：机器学习研究杂志 9.86（2008年），第 2579–2605 页。 URL： http://jmlr.org/papers/v9/vandermaaten08a.html。[10] J MacQueen。《多变量观测的分类与分析》。载于：《第五届伯克利数学、统计学与概率研讨会》。加利福尼亚大学洛杉矶分校，美国洛杉矶。1967年，第 281–297 页。[11] Leland McInnes、John Healy 和 Steve Astels。《hdbscan：层次基于密度的聚类》。载于：开放源代码软件期刊 2.11（2017年），第 205 页。

[12] Tomas Mikolov 等人。向量空间中词表示的高效估计。2013年。arXiv: 1301.3781 [cs.CL].[13] Daniel M¨ullner。现代层次聚类与凝聚聚类算法。2011年。arXiv：1109.2378 [stat.ML].[14] Radim ˇReh ̊uˇrek 和 Petr Sojka。“使用大型语料库进行主题建模的软件框架”。英文。载于：*LREC 2010*自然语言处理框架新挑战研讨会论文集。 http://is.muni.cz/publication/884893/en。马耳他瓦莱塔：欧洲语言资源协会，2010年5月，第45–50页。[15] Erich Schubert。停止在 *k*均值中使用肘部准则，改用其他方法选择簇数。2022年。arXiv： 2212.12189 [stat.ML].[16] Alvin Subakti、Hendri Murfi 和 Nora Hariadi。“BERT 作为文本聚类数据表示的性能”。载于：《大数据期刊》9.1（2022年2月）。 DOI：10。1186/s40537-022-00564-9。 URL: https://doi.org/10.1186/s40537-022-00564-9。[17] Robert L. Thorndike。“谁属于这个家族？”载于：《心理测量学》18.4（1953年12月），第267–276页。 DOI: 10.1007/bf02289263。 URL： https://doi.org/10.1007/bf02289263。[18] Marc Wegmann 等人。聚类算法系统化选择及其评估综述。2021年。arXiv: 2106.12792 [cs.LG].[19] Dongkuan Xu 和 Yingjie Tian。“聚类算法综合调查”。载于：《数据科学年刊》2.2（2015年6月），第165–193页。 DOI： 10.1007/s40745-015-0040-1。 URL： https://doi.org/10.1007/s40745-015-0040-1。