基于机器学习的文本聚类实验

一、课题综述

1.1 课题说明

小组成员任务划分：刘依扬负责对数据进行处理以及特征提取的实现；王政尧负责对特征提取方法进行研究；牛昱琛负责聚类算法的实现以及训练模型并对结果进行分析，报告为三人共同完成。三人对该实验项目有同等贡献度。

1.2 课题目标

本课题旨在探索和评估不同的文本聚类[1]技术在真实世界数据集上的应用效果。我们选择的数据集源自于流行的新闻组数据集 20 Newsgroups，这个数据集包含了来自 20 个不同新闻组的文档，涵盖了从政治、宗教到科技等多个主题。课题的主要目标是对这些文本数据进行有效的聚类处理，从而探索和分析不同主题或类别间的关系。

首先，我们将对数据集进行预处理，包括文本的清洗、去噪和分词，以确保数据质量符合后续分析的要求。接着，我们将应用多种特征提取技术，如 TF-IDF、GloVe、Doc2Vec 和 Transformer 模型，以转换文本数据为适合机器学习的数值型特征。在特征提取完成后，我们将使用 K-Means 和层次聚类（Agglomerative Clustering）这两种聚类算法对文本进行聚类。我们的目标是比较这两种算法在不同特征表示下的聚类效果，并基于外部指标（如 NMI、FMI）和内部指标（如轮廓系数、Davies-Bouldin 指数）对聚类结果进行全面评估。最后，我们还探讨了不同特征提取方法和聚类算法的选择对最终聚类结果的影响，以及这些方法在实际应用中的优缺点。

1.3 课题数据集

本课题所选用的数据集为 20 Newsgroups (License: Public Domain / Source: http://qwone.com/~jason/20Newsgroups/)，是一个广泛用于文本分类和聚类任务的公开数据集。该数据集收集自 Usenet Newsgroups 论坛，包含约 18000 篇来自 20 个不同新闻组的帖子，涵盖了从科技、政治到宗教等多样化主题。在本实验中，我们选择了其中的三个主题，分别为：

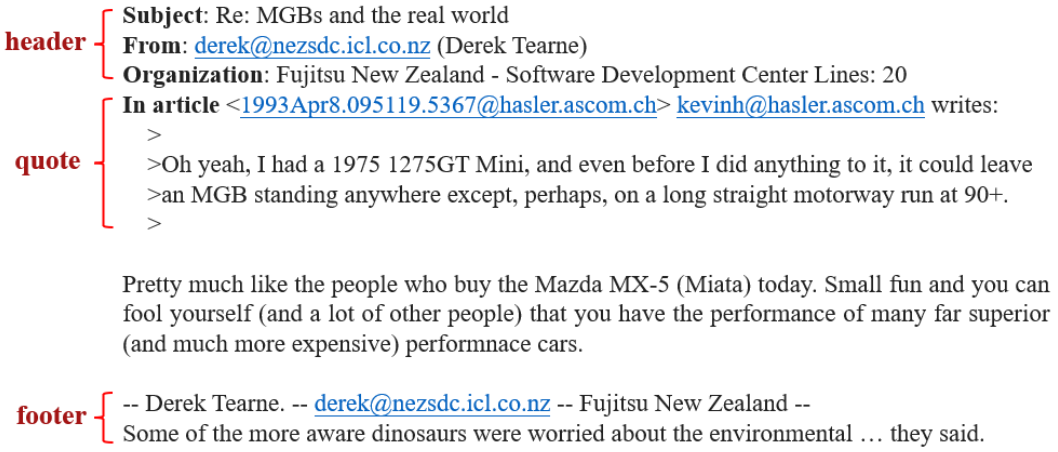
- **comp.os.ms-windows.misc**：涉及微软 Windows 操作系统的各种讨论。
- **rec.sport.hockey**：专注于 hockey 运动的讨论。
- **soc.religion.christian**：主要涉及基督教相关的讨论和话题。

原训练集中每一个样本都有一个类别标签，为了完成无监督的聚类任务，我们在训练阶段将标签去除，在测试阶段，我们使用了

二、实验内容

2.1 数据预处理

观察数据集中的样本结构，可以发现文本的基本组成如下：



可以发现，除了正文之外，原始文本包含许多与话题分析的聚类任务无关的信息，如作者、电子邮件地址、机构名称等。这些信息对于理解文本的主题没有帮助，反而会引入噪声，导致模型过拟合到这些无关的特征上。例如，在不进行预处理的情况下，模型可能会根据作者的电子邮件后缀或者某些特定的新闻组格式来区分文本类别，这些并不是我们希望模型学习到的特征。此外，如果模型学习了这些特定于数据集的特征，其很可能在新的、未见过的数据上表现不佳。移除这些特征可以帮助模型更专注于学习根据文本的内容来进行聚类，从而提高其在其他文本上的泛化能力。

据此，我们对数据集进行了以下预处理操作：

1. Headers：移除了头部信息，如发帖人、新闻组名称、发帖日期等。
2. Footers：清除了脚注信息，包括签名档或邮件结尾的致谢词。
3. Quotes：删除了引用文本，这通常是对其他帖子的回复或引用，以帮助模型专注于当前帖子的原始内容。

2.2 特征提取（文本向量化）

在对数据进行预处理之后，下一步是将非结构化的文本转换成数值向量形式，以便于机器学习算法能够处理。这个过程通常被称为文本嵌入，文本嵌入是将文本数据转换为数值向量的过程，这些向量能够表征文本中的词汇和语义信息。

目前有多种文本嵌入技术，包括但不限于以下几种：

- 词频-逆文档频率（TF-IDF）：这是一种基础的文本表示方法，通过考虑单词在文档中出现的频率以及在整个文档集中的罕见程度来量化单词的重要性。
- 词嵌入（Word Embeddings）：如 Word2Vec 和 GloVe[2]，这些技术生成更密集的向量表示，能够捕捉单词之间的语义关系。
- 文档嵌入（Document Embeddings）：如 Doc2Vec[3]，这类技术不仅考虑单个单词，还考虑整个文档的语境。

- Transformer-based 预训练语言模型：如 BERT 和 GPT，这些模型利用深度学习技术提供上下文相关的嵌入，能够更好地理解复杂的语言模式。

选择哪种嵌入技术取决于具体任务的需求、数据集的特性以及计算资源等因素。每种技术都有其优点和适用场景，例如，简单的 TF-IDF 适合基础的文本分类任务，而复杂的预训练模型则更适合需要深层语义理解的任务。通过合理选择和应用这些技术，可以大大提高文本相关机器学习任务的性能。

2.2.1 TF-IDF

首先简要介绍一下 TF-IDF 的原理和计算方法。TF-IDF，即词频-逆文档频率，是一种衡量词语对集合中某个文档的重要性的度量方法，并对某些词在一般情况下出现频率较高的事实进行了调整。

词频，表示为 $tf(t,d)$，是单词 t 在文档 d 中出现的频率，通常直接使用词在文档中的出现次数。为了减少高频词对整体文本表示的影响，我们使用子线性频率进行缩放。

逆文档频率是衡量一个词提供多少信息的指标，即这个词在所有文档中的常见程度或稀有程度。计算公式为：

$$\text{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

TF-IDF 由 tf 和 idf 相乘得到，一个词项获得高权重是由于它在给定文档中有高词频（TF），以及在整个文档集中有低文档频率（DF）。通过结合词频和逆文档频率，可以有效平衡词在文档中的局部重要性和在整个文档集中的普遍重要性。

通过使用 Scikit-learn 中的 TfidfVectorizer 可以方便地计算文档集的 TF-IDF 矩阵：

```
vectorizer = TfidfVectorizer(sublinear_tf=True, min_df=5, max_df=0.95, stop_words='english')
X = vectorizer.fit_transform(df['text']).toarray()
```

参数解释：通常，词频是根据单词在文档中出现的次数直接计算的，这里设置了 sublinear_tf = True ，TF-IDF 算法会采用次数的对数来替换原始的词频，这样做有助于减轻高频词的影响。此外，min_df = 5 表示如果某个单词在少于 5 个文档中出现，则会被忽略。这有助于过滤掉那些可能是噪声或对整个文档集贡献较小的稀有词汇。max_df = 0.95 指的是，如果某个词出现在超过 95%的文档中，那么它就不会被包括在词汇表里。这是为了过滤那些出现频率非常高但信息含量低的词汇，例如一些常见的停用词。stop_words='english' 的作用是自动去除英语中的常用停用词，比如“the”、“is”、“in”和“and”等。尽管这些停用词在文本中极为常见，但它们通常对于理解文档的主题和内容的贡献很小，即信息含量很低。

根据筛选条件，最终词表(vocabulary)中共有 5423 个单词。因此每一个样本的特征向量都是 5423 维向量，每个元素的值为对应单词的 TF-IDF。

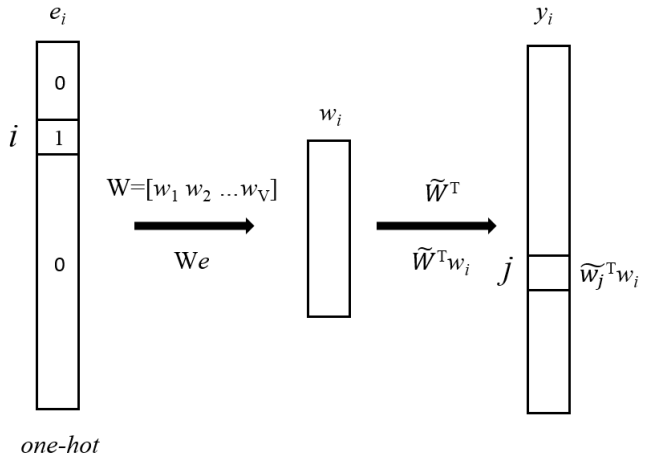
2.2.2 Word Embeddings (Glove)

1）词嵌入和 GloVe 简介

词嵌入是将文本信息转换为数值形式的一种方法，通过将词语映射到多维空间中的向量，捕捉词义和上下文关系，为机器学习提供丰富、可计算的文本表示。GloVe（全局词向量）是一种将词汇表中的单词转化为词向量的无监督学习模型。它的目标是将词语映射到一个多维空间中，使得这些词向量能够捕捉到词语之间的语义和句法关系。

2）GloVe 的工作机制

GloVe 通过构建一个共现矩阵来工作，该矩阵表示了词汇表中的每个单词与上下文中的其他单词共同出现的频率。之后，GloVe 使用这个共现矩阵来生成词向量。它对共现矩阵中的每一个元素（即每一对单词）进行操作，目的是使生成的词向量之间的点积等同于它们的共现概率的对数值。



与依赖于局部上下文信息的 Word2Vec 不同，GloVe 捕获了全局和局部上下文，为理解词义提供了更全面的信息。

3）实验中 GloVe 的实现

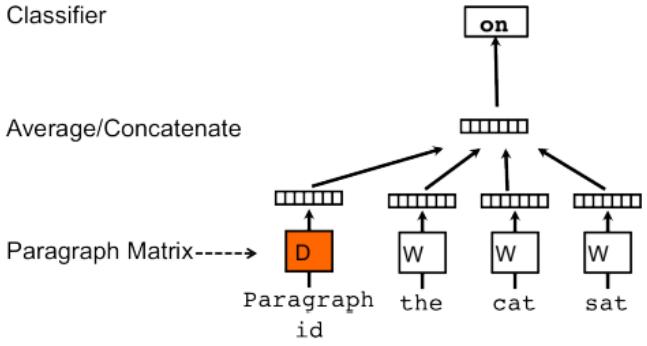
在本实验中，我们采用了一个 100 维的预训练 GloVe 模型（dim=100），用以提取和表示文本数据。我们定义了一个名为 sentence_embedding 的函数，负责处理每个句子。该函数首先将句子拆分为单词，然后将每个单词映射到其在 GloVe 模型中对应的词向量。如果 GloVe 模型中包含该词，我们就将其向量添加到 embedding_sentence 矩阵中。

为了维持数据的一致性和便于处理，我们将所有句子的长度统一固定为 256 个词。对于那些长度不足 256 的句子，我们通过在其末尾添加零向量来填充剩余部分；对于超出这一长度的句子，则进行适当的截断。这样处理后，每个句子都被转换成了一个固定大小的向量，其维度为 100×256，使得它们可以作为机器学习模型的输入，从而便于进行后续的聚类和分析工作。

2.2.3 Document Embeddings (Doc2Vec)

1）Doc2Vec 简介

Doc2Vec，也被称为文档嵌入，是一种有效的文本表示方法，用于将整个文档转换成固定大小的向量。与仅表示单个词的词嵌入技术（如 Word2Vec）不同，Doc2Vec 能够捕捉到整个文档的上下文信息，使其在处理文本相关任务时更为有效。



2）实验中 Doc2Vec 的实现

1. 数据准备

与 GloVe 不同，Doc2Vec 需要首先在本地数据集上进行训练。首先在数据准备阶段，我们采用了 nltk 库中的 word_tokenize 函数，对数据集内的文本进行的词语分割。每个文本被分解成单词序列，并与其相应的索引标签一起，构造成 TaggedDocument 对象，生成符合 Doc2Vec 模型训练要求的结构化数据。

2. Doc2Vec 模型训练

随后，我们利用 Gensim 库中的 Doc2Vec 类进行文档嵌入模型的训练。在这一过程中，设定的关键参数包括：向量大小定为 384；模型训练时的上下文窗口大小设置为 2，以考虑相邻词汇的影响；最小词频阈值为 1，确保文本中的稀有词汇也被考虑。

3. 文档嵌入的获取

我们定义了一个名为 document_embedding 的函数，实现从给定的文本中提取 Doc2Vec 嵌入。该函数通过调用 model.infer_vector 方法，并输入预处理后的文本数据，从而得到相应的文档向量。接下来，此函数被应用于数据集中的每个样本，以获取其 Doc2Vec 嵌入。

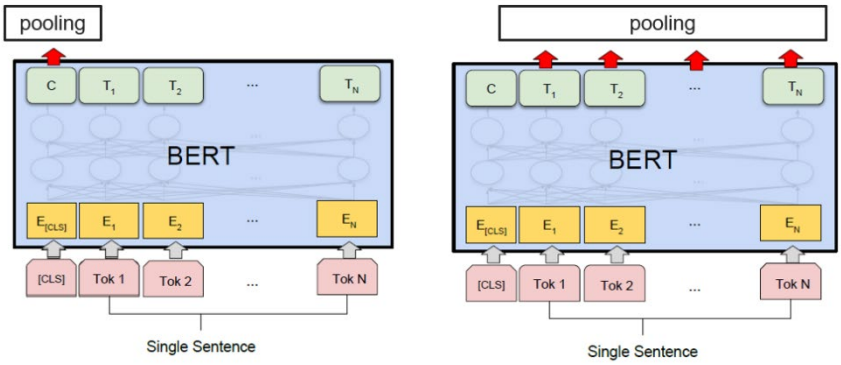
2.2.4 Transformer-based 预训练语言模型

1）方法简介

Transformer-based 预训练语言模型，如 BERT、GPT 等，通过在大规模语料库上进行预训练，掌握了语言的深层结构和丰富的语义信息。这些模型利用先进的自注意力机制（self-attention mechanisms），在理解文本的上下文关系方面表现出色，为各种自然语言处理任务提供了强大的基础。在本部分中，我们将使用预训练语言模型进行句子的特征提取。

2）实验中的实现

在本实验中，我们选择了 MiniLM[4]模型（由 Sentence Transformer 库[5]提供），这是一个基于 Transformer 架构的预训练模型，并使用 Knowledge Distillation 技术。它结合了高效的处理速度和出色的性能，适合于在资源有限的情况下生成句子级别的向量嵌入。



2.3 聚类实验

2.3.1 聚类算法介绍

在本实验中，我们采用了两种聚类方法：K-Means 聚类和层次聚类（Hierarchical Clustering）。以下是对这两种算法的简要介绍：

K-Means：K-Means 是一种广泛使用的聚类算法，它旨在将数据划分为预先指定数量的聚类。算法过程如下：首先随机选取 k 个数据点作为初始的聚类中心，然后按照数据点到这些聚类中心的距离，将数据点分配到最近的聚类中心所在的聚类。之后，重新计算每个聚类的中心，并重复这个过程，直至聚类中心不再显著变化。

层次聚类：层次聚类是一种构建聚类层次的方法。在这种方法中，每个数据点最初被视为一个单独的聚类，然后算法逐渐合并这些聚类，直到达到指定的聚类数量。我们在实验中使用的是 Agglomerative（自底向上）的方法，其中每次合并都是选择距离最近的两个聚类。我们选择了“Ward”链接方法，它最小化了合并聚类时的总方差。这种方法对于识别球状聚类特别有效。

2.3.2 实验实现

实验中，我们**手写实现**了这两种算法，此外我们也使用 sklearn 库中的聚类算法进行了实验。由于篇幅有限，这里不对算法实现的具体细节进行叙述，详细信息可以参考附件中的代码。我们分别使用两种聚类算法并使用上述 4 种特征提取方法进行聚类，其中的距离度量均使用欧几里得距离。

2.4 评价指标

在本实验中，我们采用了两类评价指标来评估聚类算法的效果：外部指标和内部指标。外部指标用于衡量聚类结果与已知的真实标签之间的相似度，这里使用标准化互信息（以下简称 NMI）和 Fowlkes-Mallows 指数（以下简称 FMI）；

内部指标是基于聚类结构本身的性质，不依赖于外部的真实标签，这里使用轮廓系数（以下简称 SS）和 Davies-Bouldin 指数（以下简称 DB）。指标详细介绍如下：

2.4.1 外部指标

1. 标准化互信息（Normalized Mutual Information, **NMI**）：
   NMI 是一种基于信息论的指标，用于衡量聚类结果与真实标签之间的相似性。它计算的是聚类结果与真实标签之间的互信息，并将其标准化，使得结果不依赖于聚类的数量。NMI 的值范围是[0, 1]，其中 1 表示完美的一致性。
2. Fowlkes-Mallows 指数（Fowlkes-Mallows Index, **FMI**）：
   FMI 是一种基于成对比较的度量，它计算的是同一聚类中成对点的几何平均数。这个指数考虑了成对样本在两个聚类结果中是否同时位于同一聚类。FMI 的值也在[0, 1]范围内，值越高表示聚类效果越好。

2.4.2 内部指标

3. 轮廓系数（Silhouette Score, **SS**）：
   轮廓系数是衡量聚类紧密度和分离度的指标。它是每个样本的轮廓系数的平均值。每个样本的轮廓系数是该样本与同一聚类中其他样本的平均距离（凝聚度）与该样本与最近聚类中所有样本的平均距离（分离度）之差，除以这两者之间的最大值。轮廓系数的取值范围是[-1, 1]，值越高表示聚类结构越清晰。
4. Davies-Bouldin 指数（Davies-Bouldin Index, **DB**）：
   这个指数是基于每个聚类的平均距离来定义的，考虑了聚类内部的紧密度和聚类之间的分离度。计算每个聚类与其最相似的其他聚类之间的相似度，然后取平均值。Davies-Bouldin 指数的值越小表示聚类效果越好。

在实验中，我们使用这些指标对 K-Means 和层次聚类算法的聚类结果进行了评估。通过这些评价指标，我们能够从不同角度了解聚类算法在特定数据集上的性能。

2.5 实验结果及分析

实验结果如下表所示：

Algorithm	Feature	NMI↑	FMI↑	SC↑	DB↓
K-means	TF-IDF	0.739	0.853	0.010	9.333
	GloVe	0.061	0.389	0.113	3.429
	Doc2Vec	0.018	0.572	**0.877**	**0.555**
	Transformer	**0.854**	**0.929**	0.090	3.178
HC	TF-IDF	0.667	0.820	0.010	9.408

GloVe	0.067	0.411	<u>0.150</u>	3.744
Doc2Vec	0.018	0.572	**0.879**	**0.562**
Transformer	**0.826**	**0.909**	0.087	<u>3.265</u>

我们可以从聚类算法和特征提取两个维度对结果进行分析：

2.5.1 聚类算法比较

对比相同特征提取方法下的两种聚类算法结果，可以发现 K-Means 聚类在外部指标（NMI、FMI）和内部指标（SC、DB）上均普遍优于层次聚类。表明 K-Means 在处理本实验中特定特征和数据集时更为有效，可能与其算法特性（如聚类形状的假设、对初始中心点的选择等）有关。层次聚类表现较差可能是由于其合并策略和聚类形状的假设不适用于这些特定的数据集。

2.5.2 特征提取比较

由于我们的任务是根据文本的主题和内容对文本进行聚类（即预先定义好了聚类的目标，而不是任意的探索性聚类），所以在对比不同的特征提取算法时，我们主要关注 NMI 和 FMI 两种外部指标，而内部指标仅能体现出聚类算法的好坏，无法表征出特征提取的优劣。

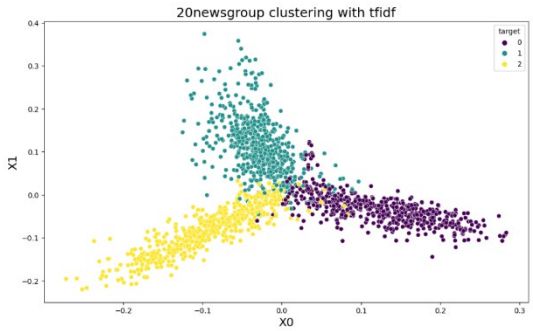
可以看出，使用基于 Transformer 的预训练语言模型提取特征在两种指标上的表现最优，这表明其在聚类任务中与真实标签的一致性最高。Transformer 模型的强大之处在于其能够捕捉文本的深层语义和复杂的上下文关系，这对于理解和分类文本特别有效。利用 TF-IDF 提取的特征表现仅次于 Transformer，显示出较高的与真实标签的一致性。其通过强调文档中独特和重要的词汇，能够有效地区分不同的文本类别。

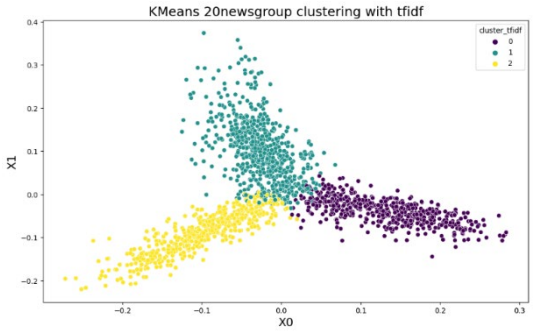
GloVe 和 Doc2Vec 表现较差，原因分析如下。GloVe 提供的是每个单词的向量表示，对于一个句子的表示是通过单词向量的简单组合（拼接）得到的。这种方法可能无法有效捕捉句子层面的语义结构和上下文关系。Doc2Vec 提取的是文档的特征，理论上应该比 GloVe 的表现更好，但在本实验中与理论预期不符。可能的原因是：本实验中使用的是在大规模语料库上预训练 GloVe，提取到的单词特征具有很好的鲁棒性；而 Doc2Vec 通常需要在特定数据集上训练以获得有效的文档表示。如果训练数据集不够大或多样化，模型可能无法充分学习文档之间的差异。在本实验中，数据集较小，不足以捕捉足够的语义差异，这可能是 Doc2Vec 表现不佳的原因之一。

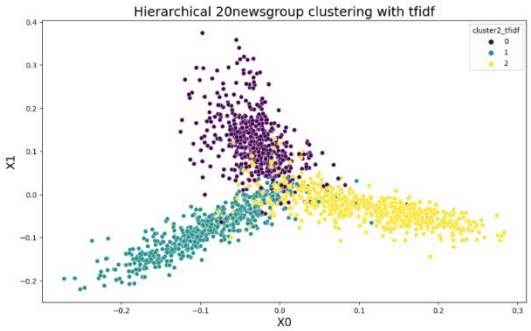
2.5.3 聚类结果可视化

下面对各种聚类的结果进行可视化展示，为了便于观察，我们使用了降维的方法，将特征降为两维。每一类特征提取方法中有三张可视化图，第一张为真实标签，后两张分别为 K-means 和层次聚类的结果：

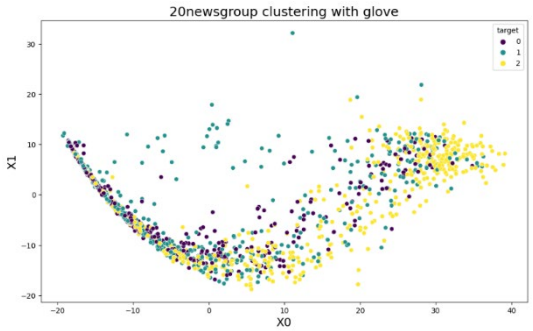
1）TF-IDF

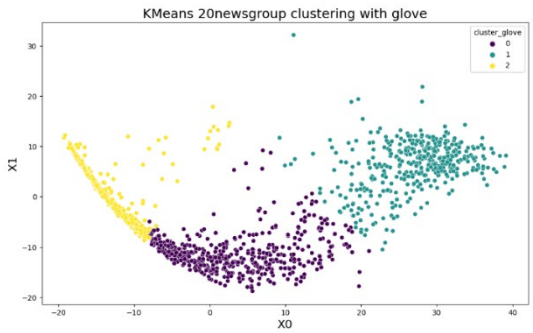


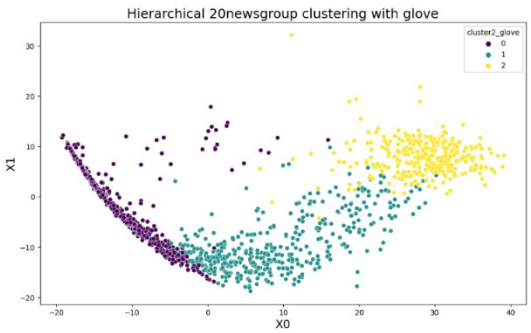




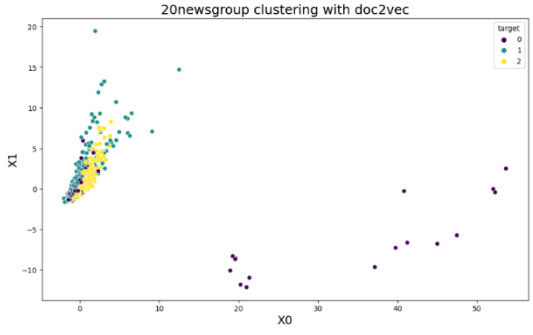
2）GloVe

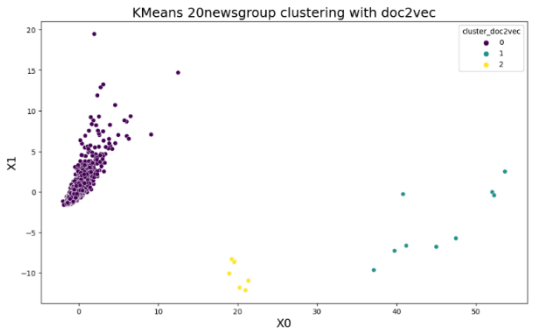


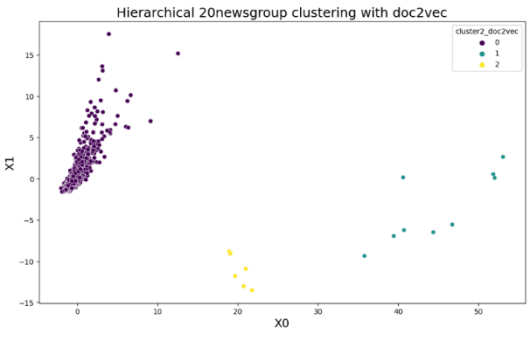




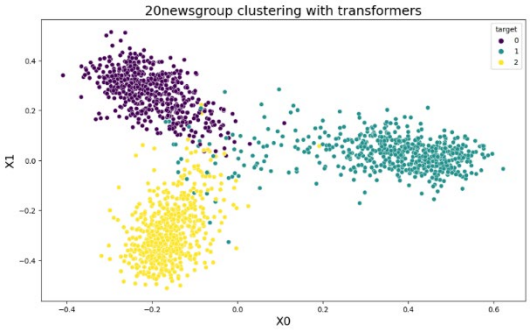
3）Doc2Vec

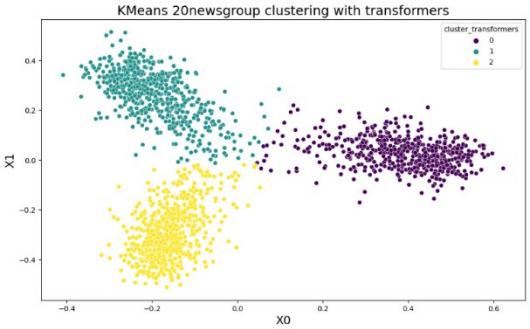


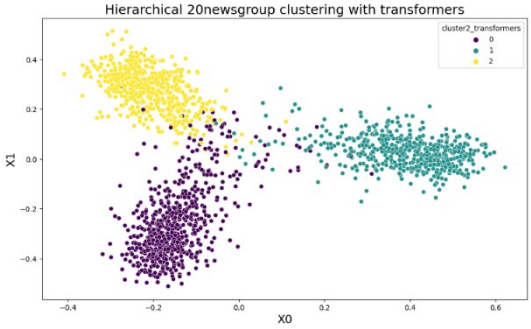




4）Sentence Transformer







可以发现使用 TF-IDF 和 Sentence Transformer 提取到的特征可以将各个类别很好的区分开，之后使用聚类算法便可以取得较好的效果。而 GloVe 和 Doc2Vec 提取出的特征并未将不同的类别区分开，所以即使使用再好的聚类算法，也不可能取得较好的效果。因此，特征提取的好坏对于聚类任务以至所有的传统机器学习来说都至关重要。

Reference

[1] Aggarwal, C.C., Zhai, C. (2012). A Survey of Text Clustering Algorithms. In: Aggarwal, C., Zhai, C. (eds) Mining Text Data. Springer, Boston, MA. https://doi.org/10.1007/978-1-4614-3223-4_4

[2] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

[3] Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." International conference on machine learning. PMLR, 2014.

[4] Wang W, Wei F, Dong L, et al. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers[J]. Advances in Neural Information Processing Systems, 2020, 33: 5776-5788.
[5] Reimers N, Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks[J]. arXiv preprint arXiv:1908.10084, 2019.