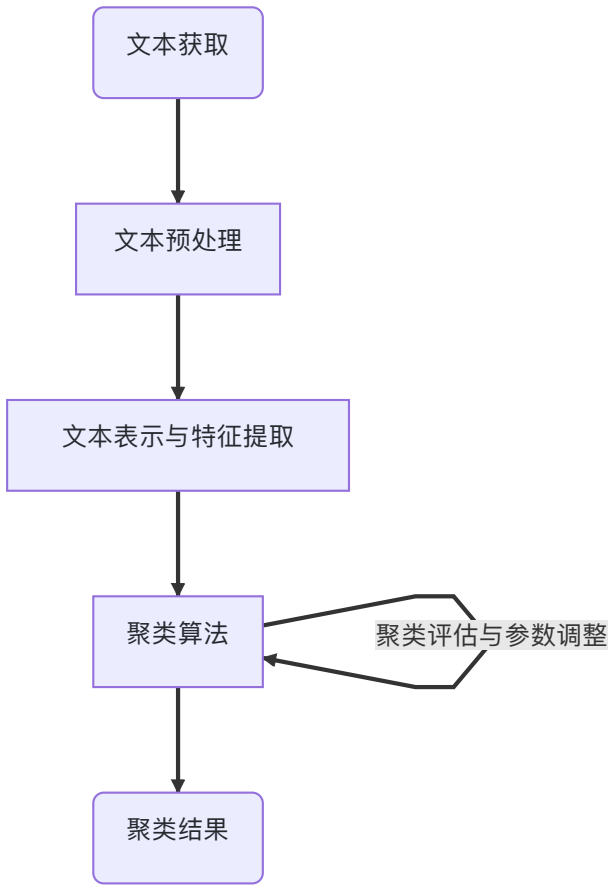
基于 20 NewsGroups 数据集的文本聚类

承子杰，202228000243001，中科院数学与系统科学研究院（AMSS,CAS）

一.Introduction

1.1 问题背景

聚类算法是利用无监督学习方法将数据划分为不同的簇，使得同一簇内的对象彼此相似，不同簇间的对象彼此相异。在存在大量无标注数据的文本领域，文本聚类可以自动将文本归类，大大提高文本信息挖掘获取的效率。文本聚类一般可以分为如下几个步骤：



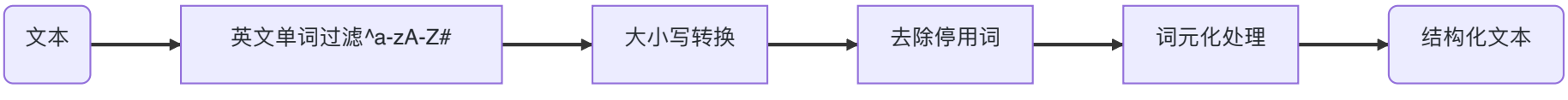
我们下面将按照这几个步骤逐一进行介绍。

1.2 文本获取

本文使用的数据集为 `20 NewsGroups`。该数据集包含 `18000` 篇新闻文章，涉及 `20` 个类别，分为训练集和测试集两个部分，是常用于文本分类、文本聚类、文本挖掘和信息检索等任务研究的国际标准数据集之一。本文只使用 `20 NewsGroups` 的训练集部分，数据标签在聚类算法期间进行剔除，只在最后评估期间使用。`20 NewsGroups` 数据集是 `sklearn` 的内置数据库，可以通过 `fetch_20newsgroups()` 函数很方便的获得。

1.3 文本预处理

文本预处理可以使得文本更加结构化、规范化，同时，去除停用词、去除低频词等处理还可以缓解文本信息稀疏的问题。在实验中我们主要进行了文本过滤、大小写转换、去除停用词、词元化等操作。具体地，在文本过滤操作中，我们使用正则表达式，仅保留英文大小写字母，将其余的字符均用空格进行替代。在大小写转换中，我们将所有过滤剩余的英文字母转化为小写字母。在去除停用词操作中，我们使用 `nltk.corpus` 提供的停用词表，将停用词用空格进行替代。最后，在词元化中，我们将一句英文句子按照单词为最小单元进行切割，并将过滤后的空句子文本进行剔除。



1.4 文本表示和特征提取

在实验中，由于新闻文本较短，我们对于文本表示主要有两种思路。一种是先对词向量进行建模，获得词的表征；再将整个文本中的每个词均用词向量进行表示，最后对文本中的所有词进行平均，用平均后的向量作为该文本的表征。另一种思路是直接使用预训练模型进行句子嵌入，获得文本表征。对于词向量的表示，我们分别尝试了基于TF-IDF特征的词袋模型(Bag of Word)、词向量模型(Word2Vec)、基于 Glove 预训练的词向量模型；对于第二种思路，我们使用的是 Roberta-Large 模型。

1.5 聚类算法

对于获得的文本特征表示，我们实验中主要实用了较为常见的 KMeans 算法， DBSCAN 算法（密度聚类）和自底向上的层次聚类算法。对于 KMeans 算法和层次聚类算法，我们都将簇的类别数设置为20；而对于 DBSCAN 算法，我们进行了简单的参数调整使其获得一个较为不错的结果。

> KMeans 聚类算法

KMeans 聚类算法由 MacQueen 于 1967 年提出，其基本思想是对于给定的数据集$\{x_1, x_2, \ldots, x_N\}$,把这 N 个样本划分到 $K(K \leq N)$ 个簇中，并使得簇内样本之间的距离平方和($\arg\min_S \sum_{k=1}^{K} \sum_{x \in S_k} \|x - m_k\|^2$)最小,因此这种方法也常被称为簇内平方和法(WCSS)。形式化地，给定初始聚类中心点 $m_1^{(0)}, m_2^{(0)}, \ldots, m_K^{(0)}$， KMeans 聚类算法主要分如下两步进行迭代：

1. (划分)：将每个样本划分到簇中，使得簇内平方和（$\arg\min_{S^{(t)}} \sum_{k=1}^{K} \sum_{x \in S_k^{(t)}} \|x - m_k^{(t)}\|^2$）最小。直观地，把样本划分到离它最近的均值点所在的聚类即可。
2. (更新)：根据上述划分计算新的簇内样本间距离的平均值，作为新的聚类中心点：

$$m_k^{(t+1)} = \frac{1}{|s_k^{(t)}|} \sum_{x_i \in S_k^{(t)}} x_i \tag{1}$$

> DBSCAN 聚类算法（密度聚类）

密度聚类方法的基本思路是，样本空间中分布密集的样本点被分布稀疏的样本点分割,连通的稠密度较高的样本点集合就是我们所要寻找的目标簇。 DBSCAN 聚类算法是该类方法中的经典算法，它具有两个超参数：邻域半径 r 和形成高密度区域所需要的最少样本数 n。根据这两个参数，我们可以定义如下几个基本概念：

- (r邻域)：以样本 P 为中心、r 为半径形成圆形领域。
- (核心样本):如果某点 P 的 r 邻域中的样本数不少于 n，则称 P 为核心样本。
- (密度直达):如果样本 Q 在核心样本 P 的 r 邻域内，则称 Q 从 P 密度直达。
- (密度可达):如果存在一个样本序列$P_1, P_2, \ldots, P_T$，且对任意 $t = 1, \ldots, T-1, P_{t-1}$ 可由 P_t 密度直达，则称 P_T 从 P_1 密度可达。根据密度直达的定义，序列中的传递样本 $P_1, P_2, \ldots, P_{T-1}$ 均为核心样本。
- (密度相连):如果存在核心样本 P，使得样本 Q_1和 Q_2 均从 P 密度可达，则称 Q_1 和 Q_2 密度相连。

DBSCAN 算法认为，对于任一核心样本 P，样本集中所有从 P 密度可达的样本构成的集合属于同一个聚类。因此该算法从某个核心样本出发，不断向密度可达的区域扩张，从而得到一个包含核心样本和边界样本的最大区域，该区域中任意两点密度相连，聚合为一个簇。接着寻找未被标记的核心样本，重复上述过程，直到样本集中没有新的核心样本为止。样本集中没有包含在任何簇中的样本点就构成噪声点簇。

> 层次聚类算法

层次聚类方法依据一种层次架构将数据逐层进行聚合或分裂，最终组织成一棵聚类树状的结构。自底向上的聚合式层次聚类方法初始时将每个数据都视 为单独的一类，然后每次合并所有类别中最相似的两个类别，直至所有的样本都合并为一个类别或者满足终止条件时结束。在聚类过程中，其主要关注两个问题：选择哪个类进行分裂和采用哪一种分裂策略。在实验中我们使用类内散度最大的类进行分裂，其类内散度衡量指标采用类内距离最远的两个样本之间的距离，距离度量指标采用欧式距离；而分裂策略采用扁平聚类算法进行中间类别。

1.6 聚类评估方法

对于聚类结果，我们主要使用了外部指标： Adjusted Rand Score， Jaccard Score和Fowlkes Mallows Score 。对于内部指标我们采用轮廓系数。

具体地， Adjusted Rand Score 是调整的兰德系数。若对于数据集$D = \{d_1, d_2, \ldots, d_n\}$，假设聚类标准为 $P = \{P_1, P_2, \ldots, P_m\}$，聚类算法实现的聚类结果为 $C = \{C_1, C_2, \ldots, C_k\}$。对于 D 中任意两个不同的样本 d_i 和 d_j，我定义如下几个参数关系：

参数	参数含义
a	d_i 和 d_j 在 C 中属于相同簇，在 P 中也属于相同簇
b	d_i 和 d_j 在 C 中属于相同簇，在 P 中属于不同簇
c	d_i 和 d_j 在 C 中属于不同簇，在 P 中属于相同簇
d	d_i 和 d_j 在 C 中属于不同簇，在 P 中也属于不同簇

则可以定义兰德指数：

$$RI = \frac{a + d}{a + b + c + d} \tag{2}$$

为了实现在聚类结果随机产生的情况下，指标应该接近于零的假设，我们定义调整的兰德系数如下：

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \tag{3}$$

Jaccard Score是杰卡德分数，它的具体定义如下：

$$JC = \frac{a}{a+b+c} \tag{4}$$

而Fowlkes Mallows Score常称为FM指数，它的定义如下：

$$FM = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}} \tag{5}$$

上述外部指标的取值范围均为$[0,1]$，且值越大表明 C 和 P 吻合的程度越高，即 C 的聚类效果越好。它们主要都考察聚类的宏观性能，在传统的聚类有效性分析中被较多地使用，但在文本聚类研究中并不多见。

对于内部指标，较为常用的是轮廓系数。它的主要思路是希望簇间越分离(相似度越低)越好，簇内越凝聚(相似度越高)越好。对于数据集中的样本 d，我们假设 d 所在的簇为 C_m，并计算 d 与 C_m 中其他样本的平均距离 $a(d)$ 和 d 与其它簇中样本的最小平均距离 $b(d)$，具体的计算公式如下：

$$\begin{aligned} a(d) &= \frac{\sum_{d \in C_m, d \neq d'} dist(d, d')}{|C_m - 1|} \\ b(d) &= \min_{c_j: 1 \leq j \leq k, j \neq m} \{ \frac{\sum_{d' \in C_j} dist(d, d')}{|C_j|} \} \end{aligned} \tag{6}$$

因此，样本 d 的轮廓系数 $SC(d)$ 和聚类总的轮廓系数 SC 分别定义如下

$$\begin{aligned} SC(d) &= \frac{b(d) - a(d)}{\max\{a(d), b(d)\}} \\ SC &= \frac{1}{N} \sum_{i=1}^{N} SC(d_i) \end{aligned} \tag{7}$$

二、Experiment

2.1 文本预处理

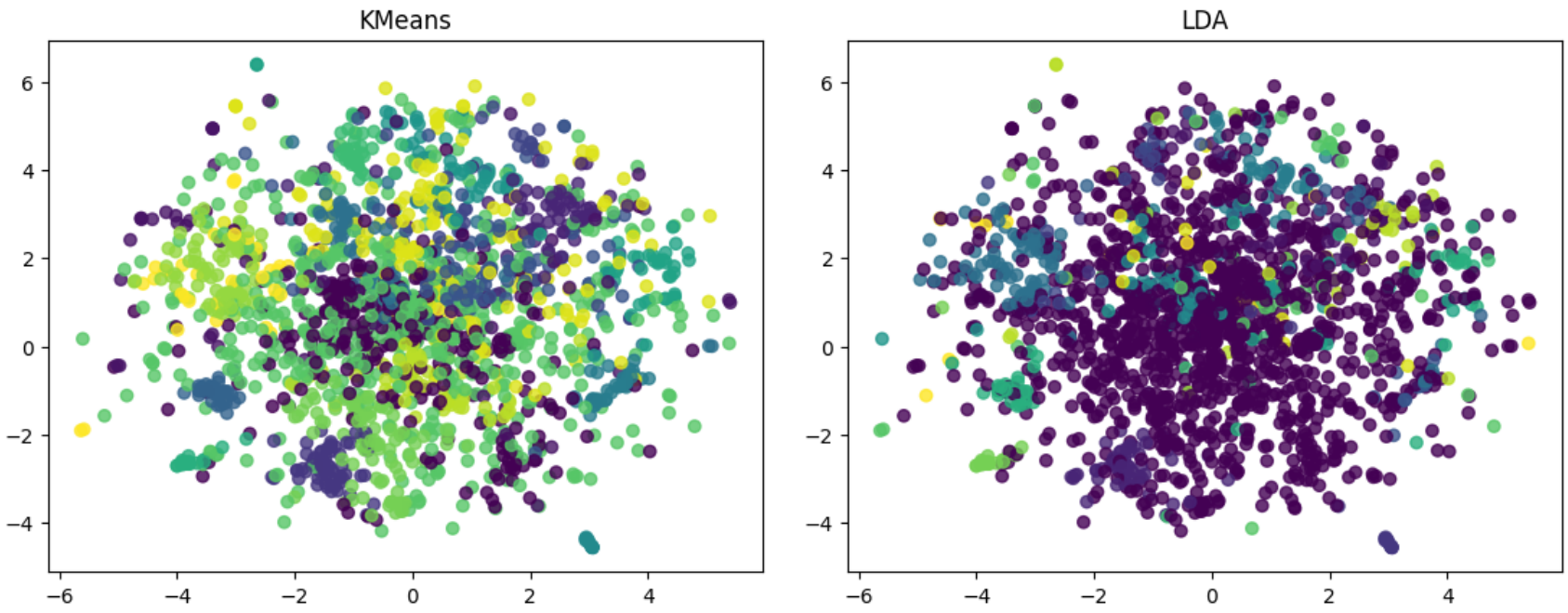
数据预处理前共有 11314 条文本，经过文本过滤、大小写转换、去除停用词、词元化等基本处理操作后，将空白文本进行移除整合，最后共获得 11000 条干净文本。

2.2 基于 TF-IDF 特征的词袋模型的文本聚类

我们使用离散特征TF-IDF进行句子的向量表示。实验中，每个句子均用一个72585的向量表示，对于得到的 11000×72855 特征矩阵，我们使用 KMean聚类算法进行实验。考虑到句子特征向量的维数过高，我们进一步使用 PCA 算法进行降维至100维，最后再进行聚类。最终的实验结果如下：

	KMeans(k=20)	
	不使用PCA	使用PCA
ARI	0.08	0.05
JC	0.05	0.03
FM	0.16	0.17
SC	0.005	0.005

可以发现，使用TF-IDF特征得到的聚类结果并不理想，进行 PCA 特征降维后并不能提升聚类效果。最后，使用TSNE将特征压缩至2维，并画出前2000样本的聚类结果如下图（左）：



2.3 基于 LDA 主题模型的文本聚类

我们使用 LDA 对单词-文本矩阵进行主题建模，取 20 个主题（本质上是进行了截断的 SVD 分解）。在得到的主题-文本矩阵中，将取值最大的主题作为该文本类别从而实现聚类，具体的实验结果如下所示

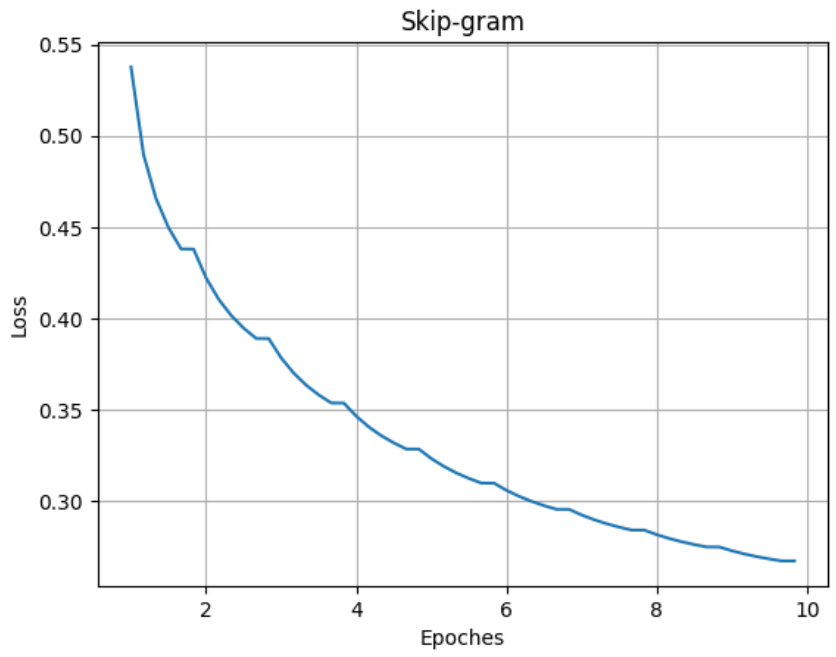
评价指标	结果
ARI	0.01
JC	0.02
FM	0.17
SC	0.07

基于主题模型的聚类结果相比于词袋模型+KMeans 聚类算法在内部指标上有显著提升，但是和外部指标相比，结果略差。同样的，我们画出其前2000个样本最后的聚类结果。

2.4 基于 Skip-gram 特征的文本聚类

使用 TF-IDF 特征的词袋模型无法刻画词语之间的相似性，而基于 skip-gram 和语言模型建模的词向量特征则可以很好的解决这一问题。我们将给定文本的所有词向量特征取均值来作为句子的特征，再进行下一步的聚类操作。

我们选取词向量的嵌入大小为 100 维， skip-gram 训练过程采用 5-gram，并使用了负采样技术与噪声对比估计算法。训练优化器采用 Adam 优化器，学习率为 0.005，10 个 epoches 下的训练损失图如下所示：



对得到的句子特征，使用 KMeans 聚类算法的实验结果如下表：

	KMeans(k=20)
ARI	0.28
JC	0.05
FM	0.32
SC	0.03

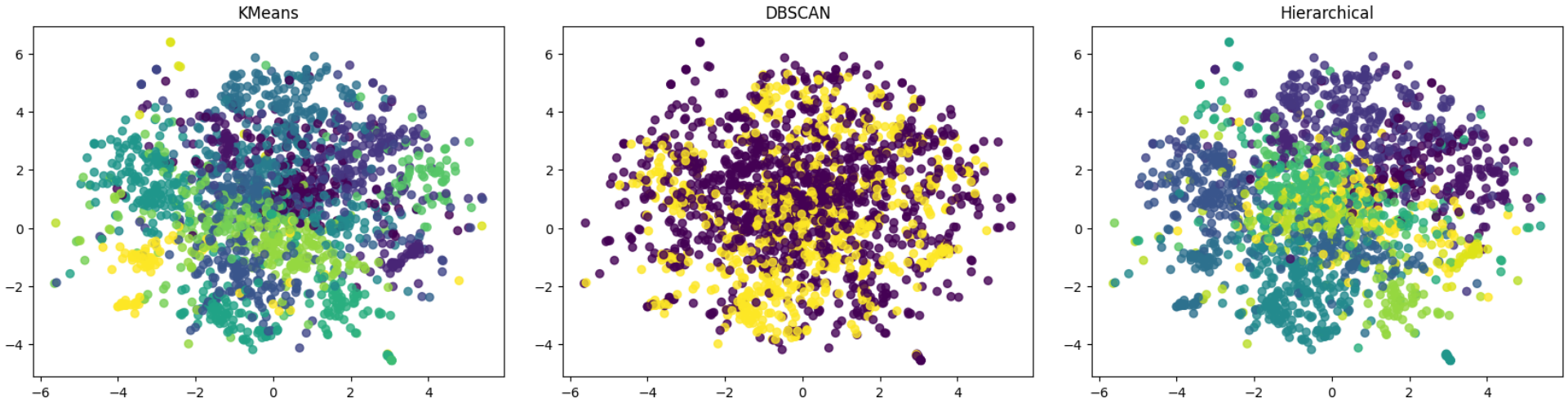
相比于 TF-IDF 特征，可以发现各项指标都有显著提升。 DBSCAN 聚类算法的实验结果如下表：

	DBSCAN(r=0.9,n=10)
ARI	0.007
JC	0.009
FM	0.17
SC	-0.009

层次聚类算法的实验结果如下表：

	层次聚类
ARI	0.27
JC	0.04
FM	0.32
SC	0.01

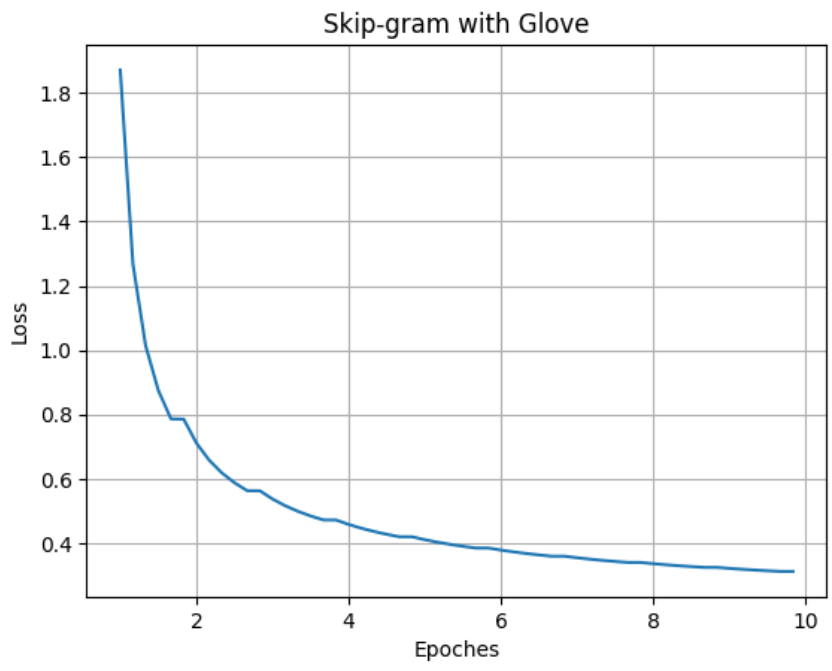
三者对于前2000个样本的聚类结果如下：



2.5 基于 Glove 微调的Skip-gram特征的文本聚类

我们使用 Glove-6B-100D 作为词向量初始化，然后再使用 skip-gram 在数据集上对词向量表示进行微调，同样的，我们将给定文本的所有词向量特征取均值来作为句子的特征，再进行下一步的聚类操作。

同样的，我们选取词向量的嵌入大小为100维， skip-gram 训练过程采用 5-gram，并使用了负采样技术与噪声对比估计算法。训练优化器采用Adam优化器，学习率为0.005，10 个 epoches 下的训练损失图如下所示：



对得到的句子特征，使用KMeans聚类算法的实验结果如下表：

	KMeans(k=20)
ARI	0.25
JC	0.01
FM	0.31
SC	0.03

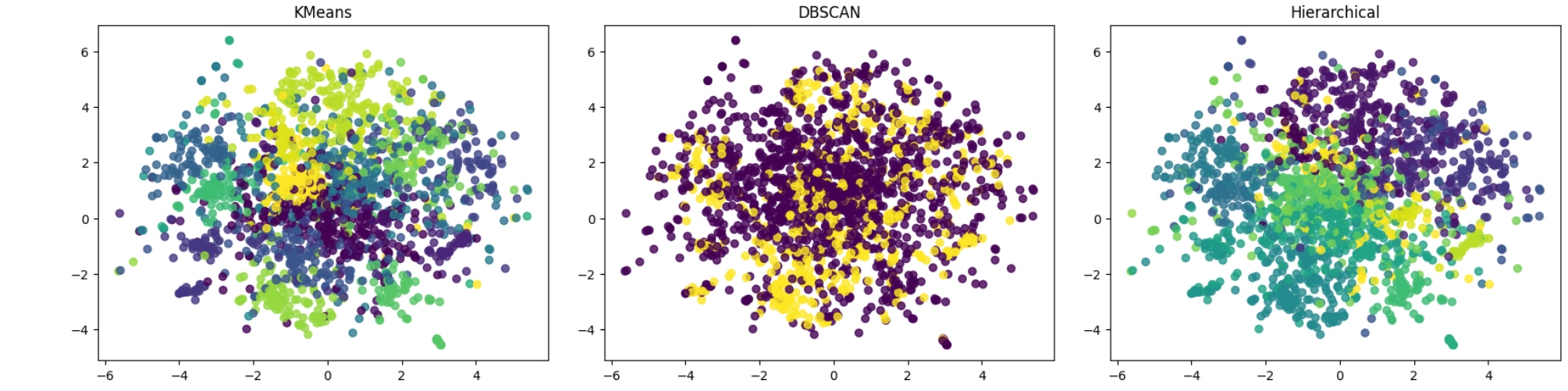
DBSCAN聚类算法的实验结果如下表：

	DBSCAN(r=0.9,n=10)
ARI	0.007
JC	0.008
FM	0.18
SC	-0.04

层次聚类算法 的实验结果如下表：

	层次聚类
ARI	0.24
JC	0.02
FM	0.30
SC	0.01

三者对于前2000个样本的聚类结果如下：



2.6 基于预训练模型Roberta-Large句子嵌入模型

由于在词向量模型表示文本时，我们都是只用了文本所有词向量的平均值来表示，因此忽略了文本中各个词之间的位置信息。我们使用基于 Transformer 架构的预训练模型 Roberta-Large，对句子直接进行特征表示，并对输出的特征进行下一步的聚类操作。

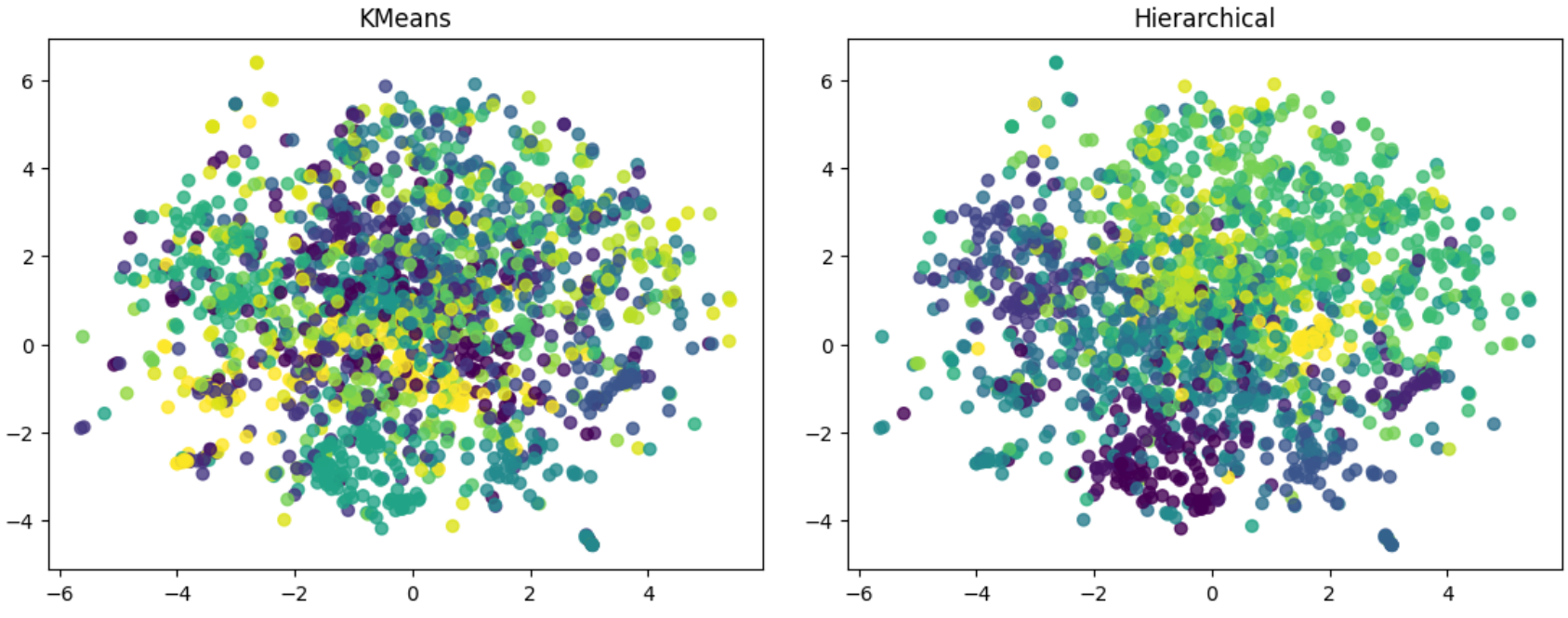
对得到的句子特征，使用 KMeans 聚类算法的实验结果如下表：

	KMeans(k=20)
ARI	0.17
JC	0.03
FM	0.21
SC	0.02

层次聚类算法 的实验结果如下表：

	层次聚类
ARI	0.14
JC	0.01
FM	0.19
SC	0.003

两者对于前2000个样本的聚类结果如下：



三.Conclusion

在上述实验中，整体而言直接使用 Skip-gram 对词向量进行特征建模，并使用 KMeans 算法进行聚类得到的结果最好， DBSCAN 算法普遍性能不太好。在 20 NewsGroups 数据集上，由于类别数量过多，且有很多类别的新闻主题十分相似，因此分类难度较大，得到效果普遍不好。可以考虑将聚类类别进行缩减，即将一些类似门类进行合并，从而获得一定性能的提升。