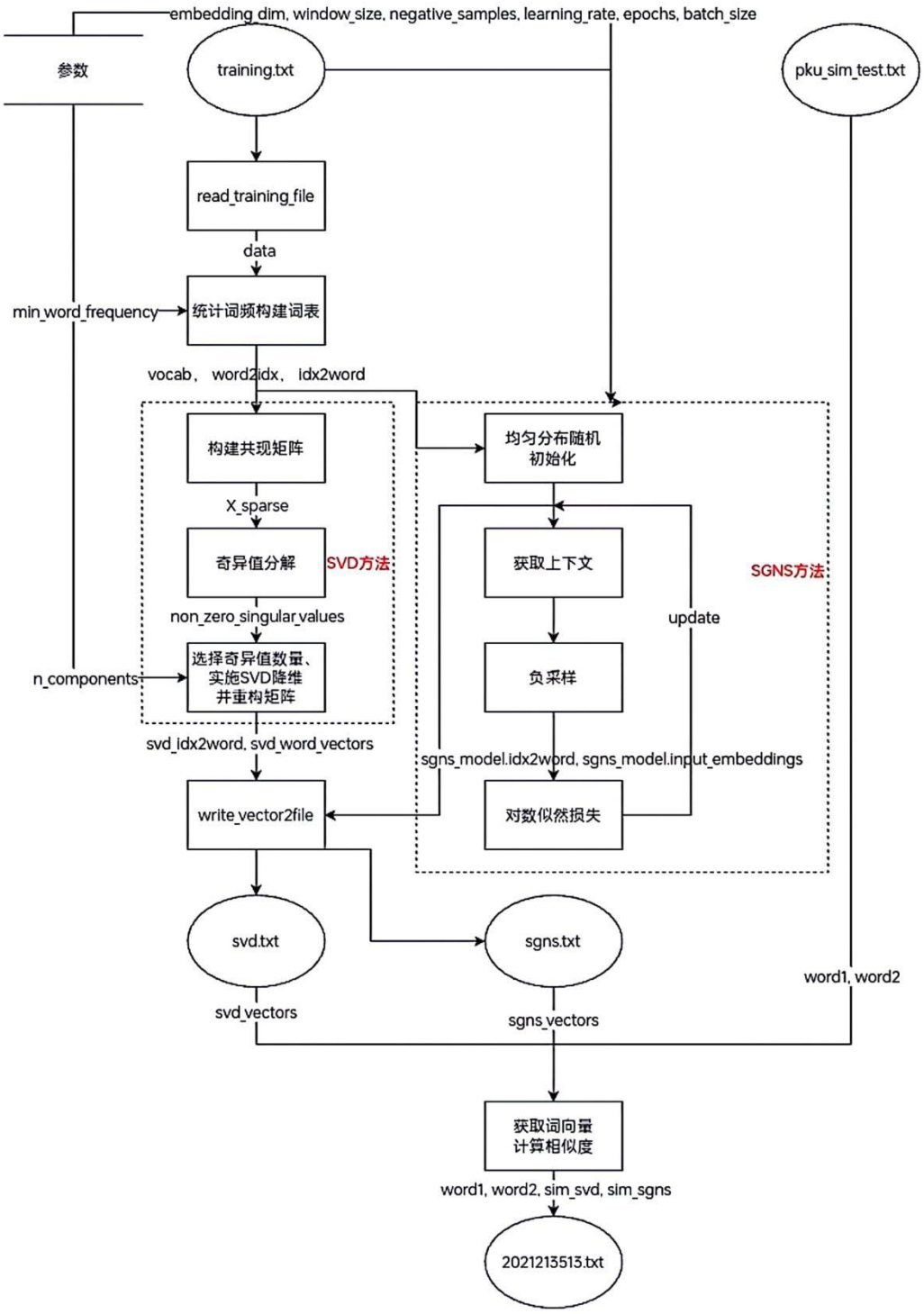
1、算法数据流图



2、SVD 方法

（1）模型参数

名称	值	说明

最小词频	默认值为 5。	
非零奇异值的数量	15595（这是最小词频为 10 时的非零奇异值）	取决于词汇表的大小和语料库的复杂性。在 SVD 分解后，得到的奇异值中，除了最大的几个之外，其余的可能会非常小，接近于零。这些接近于零的奇异值通常被视为非信息性的，因此可以忽略。 在实验中，我们对原始数据矩阵进行奇异值分解（k=min(X_sparse.shape)-1）来监控非零奇异值的数量，不过 k 值太大，会出内存问题。
选取的奇异值的数量	100	在 SVD 分解中，通常只会选择最大的几个奇异值来保留最重要的信息。选择多少个奇异值取决于所需的模型复杂度和性能。常见的选取策略是基于奇异值的大小，例如保留能量占比超过 95%的奇异值。 在实验中，我们设置的默认值为 5。这个数量可以指定，但不能超过非零奇异值的数量，如果指定的数量大于了非零奇异值的数量，则选取所有非零奇异值。
选取的奇异值之和	8140.157	
全部奇异值之和	120406.016	
选取的奇异值之和 / 全部奇异值之和	0.068	这个比例可以用来评估模型压缩信息的效率，这里并不是很高。

```
>>> ========================== RESTART: F:\课程\nlp\SGNS.py ==========================
data_ok
non_zero_singular_values 15595 all_singular_values_sum (120406.016+0.013518344j)
X_ok
原始数据中非零奇异值的数量: 15595
选取的奇异值数量: 100
选取的奇异值之和: 8140.157
全部奇异值之和: (120406.016+0.013518344j)
奇异值选取比例: (0.0676059-7.590317e-09j)
15595 15595 15595 15595
svd_ok
7487.873281955719
```

（2）执行细节

①数据预处理：

在数据预处理阶段，对原始文本数据进行清洗和分词处理，过滤符号集：chinese_punctuation = "、，．。！！？？；；：：“”'‘’（（））[【】]<《>》——"。

然后，构建词汇表，将单词映射到整数索引，为后续训练做准备。

接着，从预处理后的语料数据中提取所有单词：使用 collections 库的 Counter 函数统计它们在整个语料中出现的频率。

最后，根据设定的最小词频阈值，过滤出现次数低于阈值（min_word_frequency）的单词，得到词汇表（vocab）。

②构建共现矩阵：

构建一个稀疏矩阵（X_sparse），遍历每个文本句子，对于每个单词，如果其在词汇表中，则在对应位置加 1，形成共现矩阵（每个元素表示两个词在特定上下文窗口内共同出现

的次数）。

③奇异值分解：

对共现矩阵进行奇异值分解，将其分解为三个矩阵的乘积：U、s、Vt，其中 U 是左奇异向量矩阵，s 是奇异值矩阵（对角矩阵），Vt 是右奇异向量矩阵的转置。这一步将语料数据的高维表示转换为低维表示，并提取了数据中的关键模式。

④选择奇异值数量实施 SVD 降维并重构矩阵：

根据设定的维度（n_components），从奇异值矩阵 s 中选择最大的 n 个奇异值，其余的置为 0，从而实现对数据的降维。然后，使用被选取的奇异值和对应的左奇异向量矩阵 U，重构新的词向量矩阵。这一步是为了保留最重要的语义信息，同时减少数据的维度，以便后续的计算效率和存储开销。

这里具体实现直接调用了 sklearn.decomposition 库的 TruncatedSVD 函数。

将得到的矩阵转置则得到每一行对应一个词向量表示。

⑤输出信息：

为观察模型参数，我们打印出原始数据中非零奇异值的数量（non_zero_singular_values）、选取的奇异值数量（n_components）、选取的奇异值之和（selected_singular_values_sum）、全部奇异值之和（all_singular_values_sum）以及奇异值选取比例（singular_values_ratio），可以大致对降维效果进行评估。

⑥词向量保存：

build_svd_embeddings 方法返回词向量矩阵、词汇表、单词到索引的映射字典和索引到单词的映射字典，以便后续的应用场景中使用。

特别地，定义 write_vector2file 函数将词向量写入 txt 文档中，以便后续使用。

3、SGNS 方法

（1）模型参数

①初始词向量来源：

在实验中，我们使用均匀分布的随机数来生成初始的词向量。

②其它参数：

名称	值	说明
词向量维数	默认值为 5。	每个词向量的大小，通常在 50 到 300 维之间。较高的维数可以捕捉更复杂的语言特征，但也会增加计算成本。
窗口大小	默认值为 2。	上下文窗口越小，越倾向于学到句法关系；越大则越倾向于学到主题。在实验中我们固定窗口大小。
负采样数	默认值为 5。	负采样数决定了在每个训练样本中采样的负样本数量。增加负采样数可以提高模型训练的效率，但需要更多的计算资源。
学习率	默认值为 0.01。	学习率决定了在每次迭代中参数更新的幅度。学习率过高可能会导致训练不稳

		定，而学习率过低则可能导致训练过程缓慢。
训练批次大小	默认值为 32。	批次大小决定了每次迭代中用于更新权重的样本数量。较大的批次大小可以提高计算效率，但可能需要更多的内存。
训练轮数	默认值为 5。	训练轮数越多，模型可能会越精确，但也可能导致过拟合，或者梯度爆炸（比如原始 SGNS 实验中，训练 3 次之后词向量直接 nan 了，因此可以适当采用学习率衰减等策略来优化）。
裁剪梯度	5。	为优化训练设置的参数

（2）执行细节

①数据预处理：

同上 SVD 方法执行细节中的数据预处理。

②参数初始化：

Xavier 初始化中心词和上下文词的词向量。

③构建训练样本与负采样：

在每次迭代中，对于语料库中的每个中心词，随机选择一个上下文词作为正样本，同时随机选择 negative_samples 个未出现在上下文中的词作为负样本（即负采样）。这些正负样本对将用于训练过程中的损失函数计算。

④损失函数：

本实验目标是最大化正样本的概率同时最小化负样本的概率。

在实验中，我们先计算正样本和负样本的得分（对于正样本，计算目标词向量与上下文词向量的点积；对于负样本，计算目标词向量与负样本词向量的点积）。

接着通过 sigmoid 函数将得分转换为概率值。

最后根据这些预测概率来衡量损失。

⑤梯度下降优化：

在每个训练迭代中，使用梯度下降算法更新模型参数。对于每个训练样本，根据损失值计算其关于正样本和负样本的梯度并更新模型参数。

⑥模型评估：（未实现）

在训练过程中，定期在验证集上评估模型的性能，以监控训练进度并根据需要调整模型参数。

⑦词向量保存：

同样，write_vector2file 函数将词向量写入 txt 文档中，以便后续使用。