815 final

可参考的素材：

<https://github.com/tsborade/Mutual-Fund-Style-Classification-from-Prospectus/blob/main/main.ipynb>

<https://github.com/weylandshane/Mutual-Fund-Strategy-NLP-Classification-Project>

<https://github.com/jifuli98/NLP-and-Classification>

看这个，这个思路最对：

<https://github.com/weylandshane/Mutual-Fund-Strategy-NLP-Classification-Project/blob/main/.ipynb_checkpoints/MF815%20Final%20Proj%20(ver%200502)%20(1)-Copy1-checkpoint.ipynb>

现在colab地址：

<https://colab.research.google.com/drive/1hxKinUgyLLOcRP08CN7V_OkntQQXIE0m>

目标：use the mutual fund text summaries to predict which investment strategy each fund uses

步骤：

1. Split the data into training, validation, and testing.
2. use the skip-gram model to build a word embedding dictionary from the mutual fund summaries in the training set.
3. Design a strategy to build knowledge bases associated to aforementioned three main mutual fund types.
4. Measure distance of each summary to each knowledge base. Design a classification algorithm to predict the investment strategy of each fund.
5. Use validation data to tune your parameters of your classification algorithms.
6. Apply your classification algorithm to predict the investment strategy of each fund in the test data.
7. Instead of building word embedding ourselves, we can also use pre-trained model (for example, sentence Bert) to extract key sentences of each summary. If you use one of pre-trained models, compare the performance of your classification model in the test set with the model using your own word embedding.

我们希望完成的目标是用mutual fund text summaries来预测每个fund用的investment strategy。

介绍下MutualFundLabels.csv的情况，它存储了mutual fund的一些基础信息：id、fund\_name、Performance fee?、Ivestment Strategy、Leverage?、Portfolio composition、Concentration。

Performance fee?这一列缺失的数据达到447个，所以我们直接删去这个自变量。

数据中需要预测的investment strategy总共有五类，

Ivestment Strategy

Equity Long Only (Low Risk) 248

Fixed Income Long Only (Low Risk) 130

Balanced Fund (Low Risk) 84

Long Short Funds (High Risk) 4

Commodities Fund (Low Risk) 1

但是第四类和第五类都很少，drop掉。

这样剩下462个观测。

fund\_name里面存储了mutual fund的name，然后它能够和summary一一匹配上。我们通过fund\_name把txt格式的summaries和表格中的lable匹配上。

原始数据共有545个summary。其中84个summary没有对应的lable，把这些没有lable的summary整理出来作为测试集。留下的summary和整理好的MutualFundLabels.csv中的数据拼起来，整理成我们的原始数据集。

现在划分训练集和验证集。按照8:2的比例划分训练集和验证集。得到的训练集有368

个观测，测试集有93个观测。

下面进行数据清洗，因为我们用于预测的输入是文本，所以要把完整的句子整理为单词，并且去除没有意义的停用词。具体来说，将文本中的换行符\n和制表符\t替换为空格,并将文本转换为小写；使用来自nltk库的word\_tokenize函数对文本进行分词,将文本拆分为单词列表；过滤掉停用词。停用词是指在文本处理中通常被忽略的常见词,如"the"、"a"、"an"等；通过正则表达式过滤掉非字母字符，返回过滤后的单词列表filtered\_sentence。

使用Skip-gram Model来做词汇的向量化。细说Skip-gram Model原理

使用词形还原(Lemmatization)将一个单词还原为其基本形式。通过词形还原,我们可以将文本中的单词统一为其基本形式,从而减少词汇的多样性,提高文本处理的效率和准确性。这里我们用的是NLTK自然语言处理库提供的词形还原器(WordNetLemmatizer)来进行词形还原。

有关创建knowledge\_base。使用TF-IDF提取关键词。对每个关键词,使用预训练的词向量模型找到与其最相似的一些单词。将关键词和相似单词组合起来,构建一个扩展的知识库。

TF-IDF的计算公式如下:

* TF(Term Frequency):一个词在一个文档中出现的频率。
* IDF(Inverse Document Frequency):一个词在整个语料库中的稀有程度。IDF = log(总文档数 / 包含该词的文档数)。
* TF-IDF = TF \* IDF

TF-IDF的值越高,说明一个词在一个文档中出现的频率越高,且在整个语料库中越稀有,因此它对于该文档的重要性越高。

接下来计算句子与知识库的相关性:

* 将句子中的词向量求平均,得到句子的重心向量。
* 计算句子重心与知识库中每个关键词的余弦距离。
* 取前n\_closer个最小距离的平均值作为句子的相关性得分。

最后,函数返回得分最小的num\_sent个句子,即与知识库最相关的句子。

通过这样的方法把训练集验证集测试集中的关键句子筛选出来，