基于Word2vec的人物关系分析

汇报人：07172757-李治远

内容概要

1 项目目标

2 实现方法

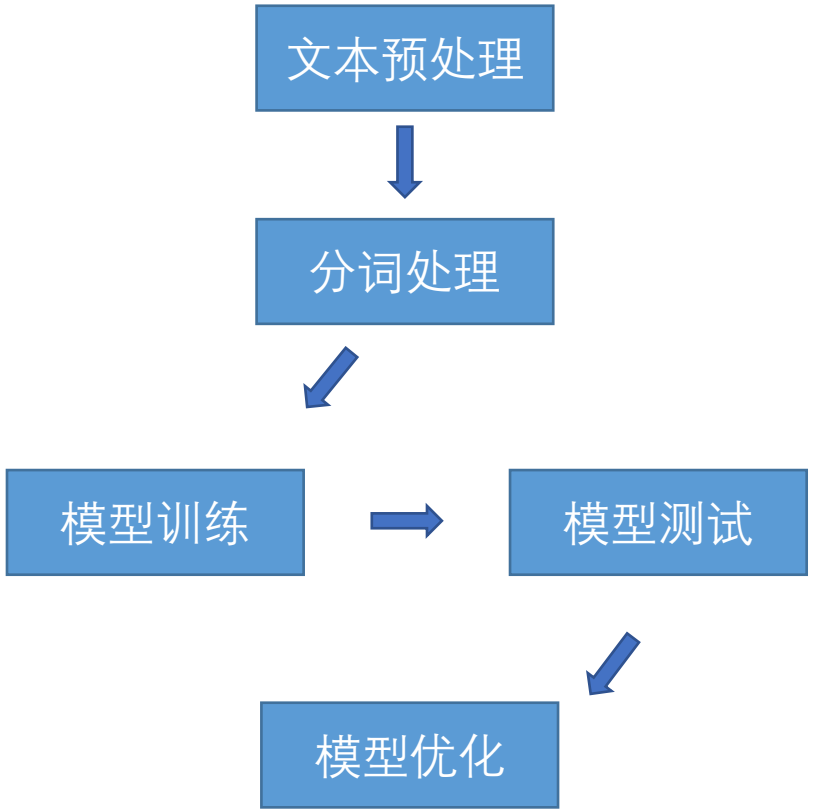
3 结果及分析

4 方法改进

1 项目目标

- 学习使用Word2vec模型分析文本
- 分析小说中人物之间的关系——以《都挺好》为例

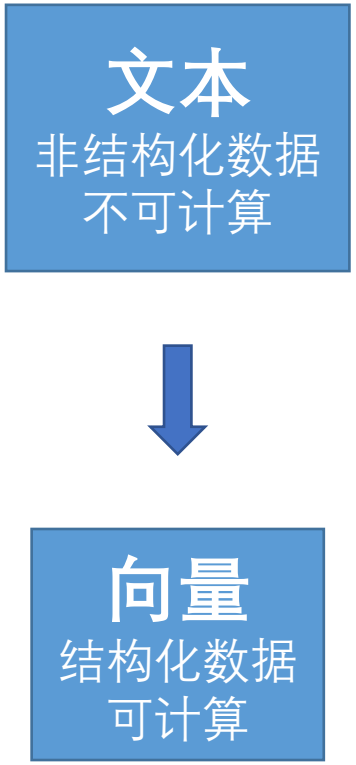
实现路线



2 实现方法

Word2vec

- 将不可计算、非结构的词转化为可计算、结构化的向量。
- 重点关注生成的词向量



2 实现方法

两种训练方式

CBOW

通过上下文来预测当前值。相当于一句话中扣掉一个词，然后猜这个词是什么

今天是一个____天气

Skip-gram

用当前词来预测上下文。相当于给一个词，然后猜前面和后面可能出现什么词

____ __ ____晴朗的____

2 实现方法

Word2vec优点

- 由于Word2vec会考虑上下文，较之前的Embedding方法相比，效果要更好
- 比之前的Embedding方法维度更少，所以速度更快
- 通用性很强，可以用在各种NLP任务中

Word2vec缺点

- 由于词和向量是一对一的关系，所以多义词的问题无法解决。
- 是一种静态的方式，虽然通用性强，但是无法针对特定任务做动态优化

2 实现方法

文本预处理

"我有没有良心,你没资格评论.至于寻你们开心,你配吗?"明玉冷着脸,满脸都是不屑一顾.当时她看着明成夫妻恸哭时候就想,这两人跑了一个米饭班主,如此伤心总算还是有点良心.

朱丽哽咽着道:"何必呢,对我们有怨气,何必拿到今天来现?很标新立异吗?"

明玉冷笑:"你不觉得今天是很好的机会吗?大哥,没事我先走,你什么时候需要用车,打我手机."

明成也是冷笑:"那么,谢谢您大驾到场."

明玉依然冷笑:"苏明成还轮不到你代表苏家说这句话."

"对,你最配.仗着有几个臭钱撑腰杆子."明成火了,还是朱丽伸手抱住他不让他冲动.

"很可惜,你有本事也拿出那几个臭钱来,你有种别问家里伸手要臭钱.我说你不配就是不配,论对苏家贡献,论为苏家牺牲,你排最末尾还是看你有苏家

文本中存在换行符等其他空白字符，使用python去除，方法如下。

```
@staticmethod
def get_content(filename):
    """
    加载文件内容
    :param filename:
    :return:
    """
    with open(filename, 'r', encoding='utf-8') as f:
        return f.read().replace('\n',
'').replace('\t', '').replace(' ', '').strip()
```

2 实现方法

分词处理

苏明玉、苏大强、苏明成、朱丽、苏明哲、吴非、蒙志远、柳青、赵美兰、小蒙、老蒙、明玉、明哲、明成、大强、舅舅、小咪、满总

人物字典

吧哒
把
罢了
被
本
本着
比
...

停顿词

```python
@staticmethod
def load_user_dict(user_dict_path):
    """
    加载用户自定义词典
    :param user_dict_path:
    :return:
    """
    jieba.load_userdict(user_dict_path)
    fp = open(user_dict_path, 'r', encoding='utf-8')
    for line in fp:
        line = line.strip()
        jieba.suggest_freq(line, tune=True)
```

```python
@staticmethod
def load_stop_words(stopwords_path):
    """
    加载停用词
    :param stopwords_path:
    :return:
    """
    return [line.strip() for line in open(stopwords_path,
'r', encoding='utf-8').readlines()]
```

2 实现方法

分词处理

为了提高效果，在分词的时候加入词性，下面是分词结果。

苏家 一门 退休 后 平静 生活 苏母 麻将 桌旁 猝死 打破 苏母 一向 争强好胜 人 退休 前 市里 医院 护士长 各色 奖章 取出 披挂 全身 俨然 领 金光闪闪 铠甲 苏母 工作 风风火火 带入 生活 苏父 苏大强 名不副实 长年累月 躲 苏母 高大 壮实 背影 后 做 小 男人 中学 图书馆 整理 图书 退休 退休 悄无声息 走 后 整个 学校 无 人 想起 苏大强 愈发 信心 走路 铁掌 水上漂 闻 一点 动静 苏母 铁腕 下养 三个 出色 儿女 个个 小学 初中 高中 尖子 年龄 顺理成章 进入 高等学府 左邻右舍 说 国家 重点 大学 苏家办 苏母 人 前 大声 欢笑 人 后 愁眉苦脸 自打 大儿子 苏明哲 考入 清华大学 始 苏母 逼 苏父 天天

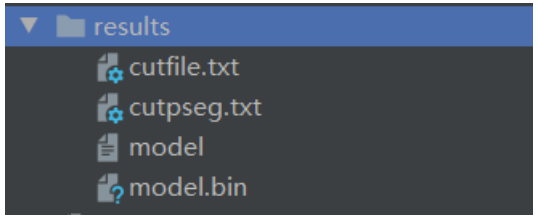
```python
def cut_text_pseg(self, text, cut_file, cut_file_pseg, path_stop_words):
    """
    分词和词性
    :param text:
    :param cut_file:
    :param cut_file_pseg:
    :param path_stop_words:
    :return:
    """
    self.load_user_dict("dict/dict.txt")
    cut_text = pseg.cut(text)
    stop_words = self.load_stop_words(path_stop_words)
    out_cut_text = []
    out_cut_pseg = []

    index = 0
    print(cut_text)
    for key, pg in cut_text:
        index += 1
        if key not in stop_words:
            out_cut_text.append(key)
            out_cut_pseg.append(pg)
    fo = codecs.open(cut_file, 'w', 'utf-8')
    fo.write(' '.join(out_cut_text))
    fo.close()
    fo = codecs.open(cut_file_pseg, 'w', 'utf-8')
    fo.write(' '.join(out_cut_pseg))
    fo.close()
```

2 实现方法

模型训练

```python
@staticmethod
def train(train_file_name, save_model_name):
    """
    训练形成模型
    :param train_file_name:
    :param save_model_name:
    :return:
    """
    setences = word2vec.LineSentence(train_file_name)
    model = gensim.models.Word2Vec(setences, min_count=1, size=200)
    model.save(save_model_name)
    model.wv.save_word2vec_format(save_model_name + '.bin', binary=True)
```



model与model.bin为训练结束保存的模型

3 结果及分析

模型预测

和苏明玉最相似的词有：
身份, 0.9999064803123474
钥匙, 0.9999038577079773
关心, 0.9999014139175415
车门, 0.9998956322669983
谢谢, 0.9998923540115356
电梯, 0.9998911023139954
孩子, 0.9998807311058044
饭店, 0.9998770952224731
脸色, 0.9998714923858643
明白, 0.99986732006073

苏明玉 与 苏明玉的相似度为： 1.000000
苏明玉 与 老蒙的相似度为： 0.999313
苏明玉 与 苏明哲的相似度为： 0.999297
苏明玉 与 舅舅的相似度为： 0.999122
苏明玉 与 苏大强的相似度为： 0.998884
苏明玉 与 小蒙的相似度为： 0.998583
苏明玉 与 吴非的相似度为： 0.997641
苏明玉 与 柳青的相似度为： 0.997400
苏明玉 与 苏明成的相似度为： 0.997377
苏明玉 与 朱丽的相似度为： 0.997341
苏明玉 与 明哲的相似度为： 0.996648
苏明玉 与 明成的相似度为： 0.993391
苏明玉 与 明玉的相似度为： 0.992809

结果分析

可以看出苏明玉与老蒙的关系很近，通过电视剧的观看，苏明玉是老蒙一手带大的，也是销售部的经理，说明模型预测的效果大致较好。

4 方法改进

通过预测结果的观察，发现苏明玉与舅舅的关系（0.999122）比与苏大强（0.998884）的关系更近，但是通过电视的观看，事实并不是这样的。

改进方法

- 将这些人名的词性加入到词性文件，这样在查找时，就可以找到。
- 直接寻找指定词与目标词的相似度
- 使用其他模型与方法进行训练测试

谢谢