作业程序需求描述：

TextRank算法抽取关键词

\*1) 将给定的文本分句切割。

\*2) 对于每个句子，分词和词性标注，只保留指定词性的单词，如名词、动词、形容词。

\*3) 构建图G = (V,E)，V为节点集，由（2）生成的候选关键词组成，然后采用共现关系构造任两点之间的边，两节点之间存在边仅当它们对应的词在长度为K的窗口中共现。

\*4) 根据PageRank算法思想，迭代传播各节点的权重，直至收敛。

\*5) 对节点权重进行倒序排序，得到最重要的T个单词，作为候选关键词。

\*6) 由5得到最重要的T个单词，在原始文本中进行标记，若形成相邻词组，则组合成关键短语。

\*将文本中的每个句子看做一个节点，如果两个句子相似度高于指定阈值，则认为这两个句子对应的节点之间存在一条无向有权边。

\*选出TextRank值最高的K个节点对应的句子作为文档摘要。

程序结构描述和解释：

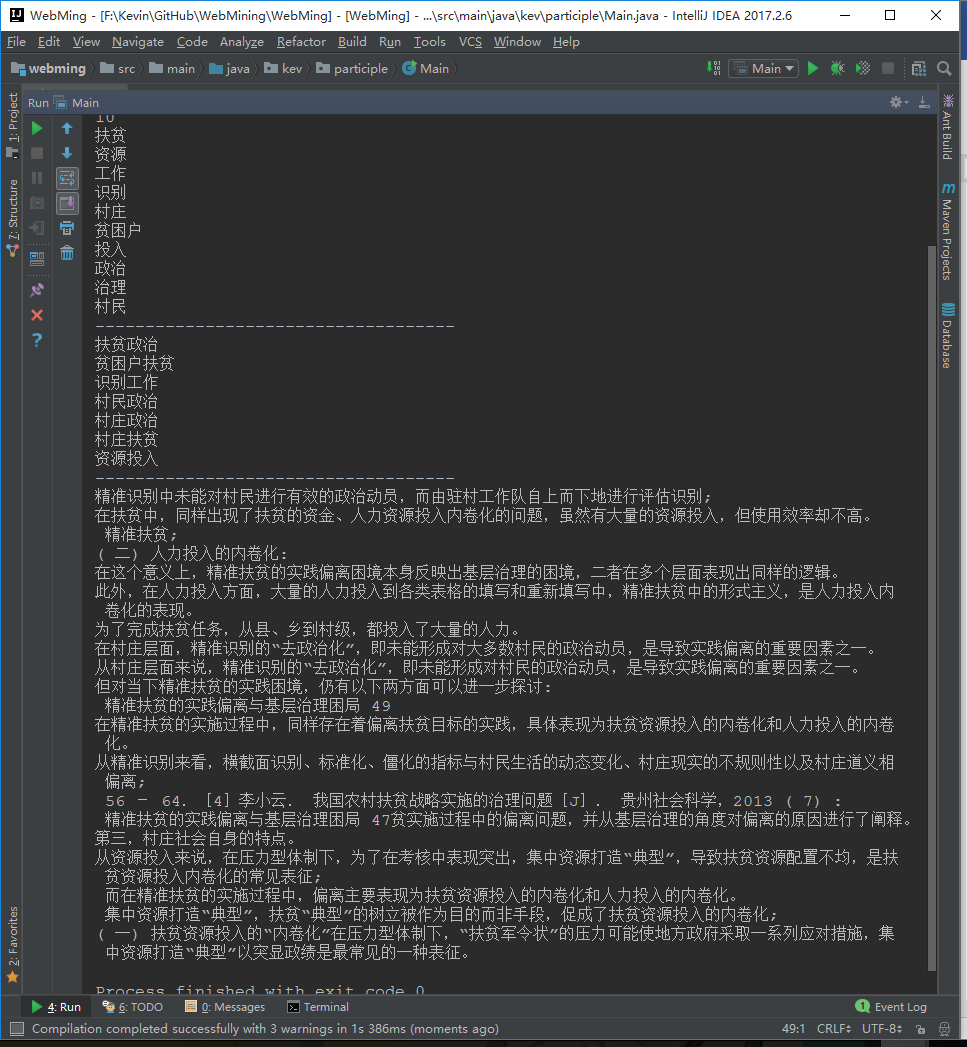
程序使用Java语言编写，使用的工具是IDEA，并使用配备的Maven资源管理工具对所需要的Java类库进行自动下载导入。

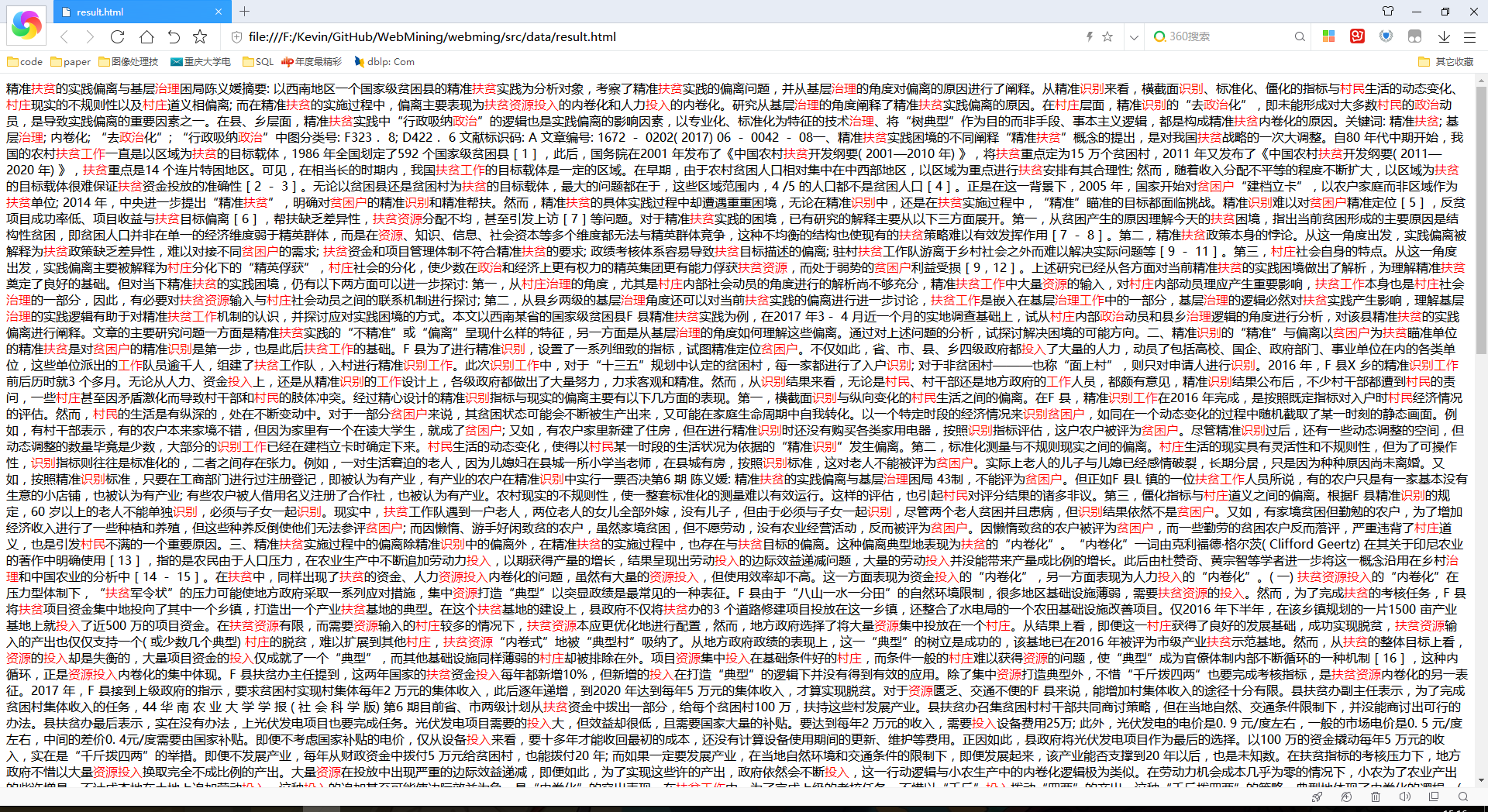
程序的主要运行流程为：

分词，从磁盘中读入一篇文章作为一个String text，调用封装好的方法把text分割成词语列表并且标注词性，然后删除词语列表中的停用词，根据设定的K值(程序设置K为10)，构造词语的图结构G=(V,E)，V为所有的次，E为两个词之间是否有边（两个词相隔小于K则有边），出事的PR值设置为该词语在文中出现的次数，图无环（即任意节点不与自己相连）。然后调用封装好PageRank算法的方法进行迭代传播权重，每一条边的权重为与相邻节点的PR值除以改节点的度，两个方向计算的时候虽然度一样但是PR值不同。迭代值变化误差小于指定阈值时结束（程序设置阈值为0.001）。排序后选择PR值最高的T个作为关键词（程序设置T为10）。然后把关键词红色高亮标记写入一个html文件中。把在text中位置相邻的两个词语连接作为一个短语打印到屏幕显示。

分句，把text根据中文断句的标点符号分割成一个句子列表，对于每一个词语,在词语列表中都有一个index位置标记，则形成一个向量，每一个向量的属性表示一个词语，向量的维数表示不同词语的数量，检测每一个词语在每一个句子中是否出现，出现则该句子的词语向量对应的分量设置为1，然后计算两个句子的词语向量的cosine值作为衡量两个句子的相似度的标准，句子的PR值以所有词语的PR值只和作为新的关键词，把相似度处于前百分之n的关系元组看作存在边联系，则构造好了图结构G=（V，E），V表示句子，E表示两个句子之间的相似度出去前百分之n，同理使用与上述分词相似的分句PageRank算法迭代直到收敛。程序中，相似度取前百分之20的句子设置边联系（计算相似度cosine值时已经过滤掉了cosine值小于0.2的关系元组），PageRank算法收敛阈值设置为0.01，最后选出前20个作为关键句。

程序运行结果如下图所示：





程序中使用了一些封装好的比较成熟的java库，分别是：

Ansj——用于已经训练好的用于智能中文分词并标注词性

Lucene——用于对文本进行高亮设置

Math3——用于处理句子向量的点积和cosine运算

Lombok——用于自动生成类属性的setter和getter方法，避免代码看起来冗余繁多

程序实验使用的是一篇关于精准扶贫的论述论文，分词效果相对还不错，提取出的基本上是文中的主要关键字，但是关键句子提取效果不是特别好，虽然也提取出了一些关键论述的句子，但是也有一些文献引用也被提取出来了。

分词的运行效率比较快，丝毫没有等待的感觉结果就已经出来了，但是句子的PR算法迭代的效果不是很好，句子的词向量采用的是所有的关键字作为向量空间，句子中存在大量的0元素，特征向量比句子本身都要长，运行效率非常差，而且PR算法的收敛速度比词语要慢许多，但是没有找到造成收敛慢的原因。

本来的想法是使用Word2Vec算法对词语进行向量话，但是通过查找网络上的文献和资料，发现使用这种方法需要先进行语料分析，进行相对而言比较大量的训练形成一个模型，才能用于把词语向量化，短时间内在网上没有找到已经训练好的模型，又没有精力自己去训练一个模型，就采用了这种最简单最方便的方法对句子进行向量化，最后虽然有了看起来还不错的结果，但是有非常大的可以改进的地方。