算法描述：

**MM（正向最大匹配算法）：**

设MaxLen表示最大词长， D为分词词典  
(1) 从待切分语料中按正向取长度为MaxLen的字串str,令Len=MaxLen;  
(2) 把str与D中的词相匹配;  
(3) 若匹配成功， 则认为该字串为词， 指向待切分语料的指针向前移Len个汉字(字节)， 返回到（ 1） ;  
(4) 若不成功：如果Len>1， 则将Len减2， 从待切分语料中取长度为Len的字串str,返回到（ 2） 。 否则， 得到长度为2的单字词， 指向待切分语料的指针向前移1个汉字，返回（ 1） 。

**RMM（逆向最大匹配算法）：**

从被处理文档的末端开始匹配扫描，每次取最末端的2i个字符（i字字串）作为匹配字段，若匹配失败，则去掉匹配字段最前面的一个字，继续匹配。

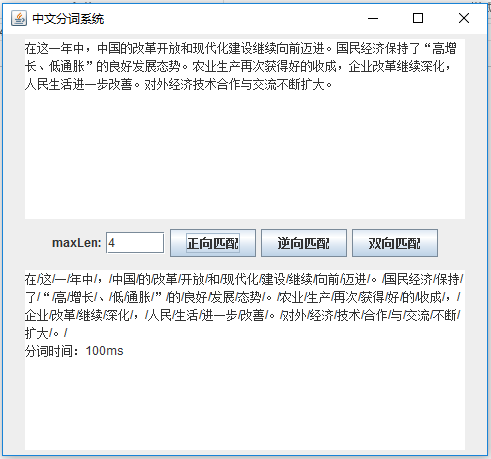
**软件演示截图**：

测试字符串：

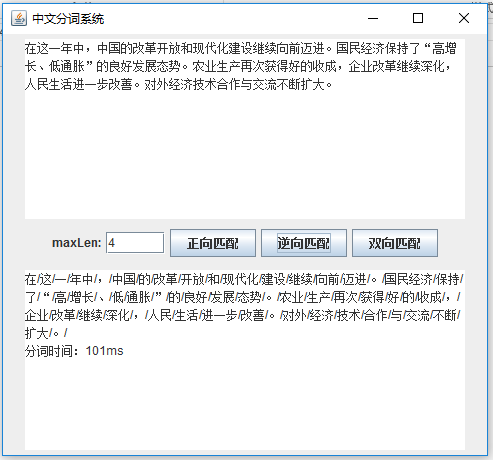
在这一年中，中国的改革开放和现代化建设继续向前迈进。国民经济保持了“高增长、低通胀”的良好发展态势。农业生产再次获得好的收成，企业改革继续深化，人民生活进一步改善。对外经济技术合作与交流不断扩大。

字典：chineseDic.txt

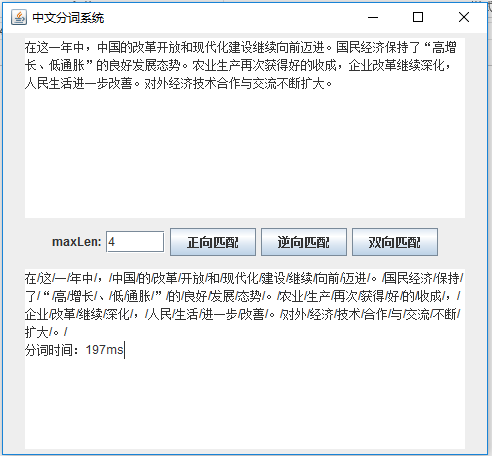
使用正向匹配：



使用逆向匹配：



使用双向匹配：



**总结**：

通过本次实验，我们对NLP规则分词的双向匹配都有了更深的理解。正向匹配算法和逆向匹配算法，其本质原理是一样的，实现起来有一点不同，主要体现在字符串的处理方向上。我们这次的程序实现的双向匹配算法使用了循环的逻辑，还可以像老师一样，使用递归实现，考虑到时间上的效益，我们选择了循环。本次实验采用了Java语言实现，Java有很多封装好的库，使得实验的难度降低了很多，站在巨人的肩膀上的确是赢在了起点。