**1.pipeline**

分词：中文/英文

清洗：

停用词，根据场景不同选择也不同；有很多停用词的词库

标准化：对英文来说很重要

把多个单词合并成一个单词，apple&apples—》apple

需要一定的工程能力

图片包含 屏幕截图

描述已自动生成

**2.word segmentation分词**

**分词工具：**

Jieba分词 <https://github.com/fxsjy/jieba>

SnowNLP <https://github.com/isnowfy/snownlp>

LTP <http://www.ltp-cloud.com/>

HanNLP <https://github.com/hankcs/HanLP/>

图片包含 屏幕截图

描述已自动生成

见jupter3

**2.1Max Matching最大匹配（贪心算法）**

必定有一个词典库，进行参考。

<https://blog.csdn.net/selinda001/article/details/79345072>

**1）forward-max matching前向最大匹配**

max\_len参数（例如=5）

“我们经常有意见分歧”

Step1:我们经常有—》词典库没有

Step2:我们经常--〉词典库没有

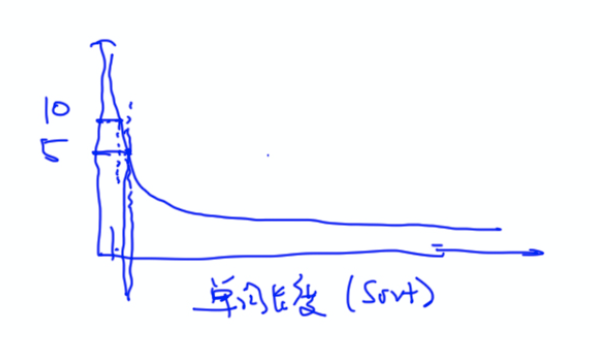
Step3:

Step4:我们--》词典库有

图片包含 文字

描述已自动生成

max\_len如何选取，越高时间复杂度越高。所以，首先参考词典库，大概百分比。



**2）backward-max matching后向最大匹配**

图片包含 文字

描述已自动生成

前向与后向，结果基本一样。

**3）缺点**

-细分，有可能是更好

-局部最优

-效率低

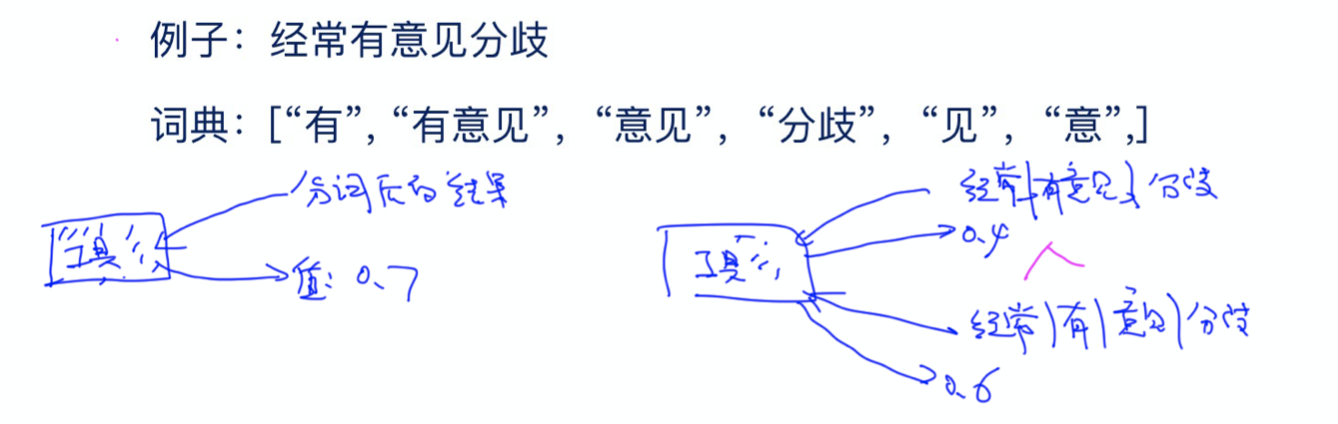
-歧义，不考虑语义

图片包含 文字, 白板

描述已自动生成

**2.2Incorporate Semantic考虑语义**

存在一个“**工具**”， 会返回一个数值，帮助判断语义正确性。

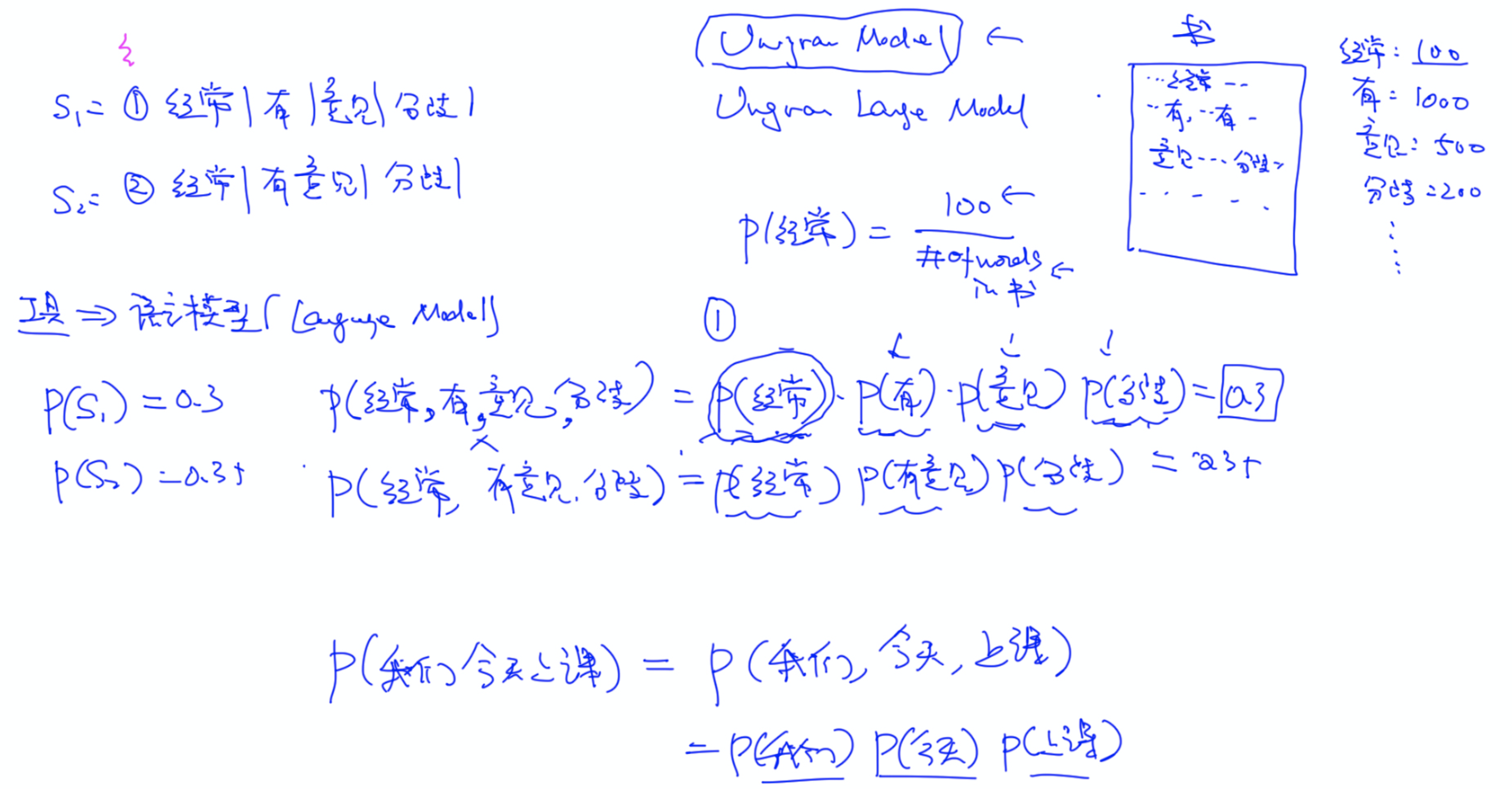


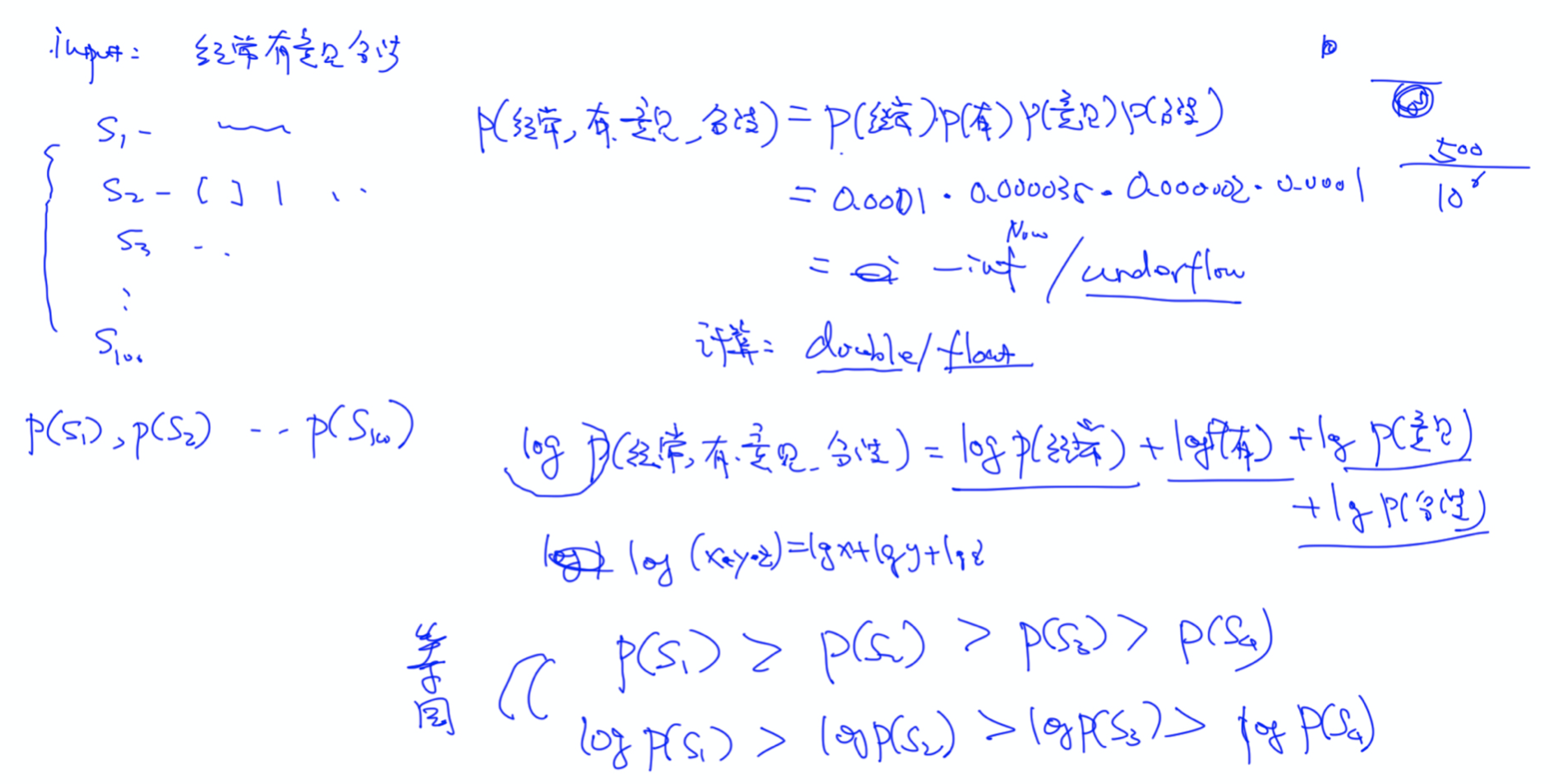
**方法**：输入---生成所有可能的分割---选择其中最好的（工具language model）

图片包含 文字, 白板

描述已自动生成

**Language model：**





**2.3维特比算法**

从一个、到两个字组合、到三个字组合，如此循环

目标：找路径之和最小的路径。

实质：最短路径算法

图片包含 文字

描述已自动生成

图片包含 文字

描述已自动生成

**2.4summary**

图片包含 屏幕截图

描述已自动生成

**3. Spell Correction拼写错误纠正**

**两类：**错别字 or语法错误（上下文）

错别字：字典里不存在

**3.1 edit distance**

图片包含 文字

描述已自动生成

**Edit：insert、delete、replace**

图片包含 文字

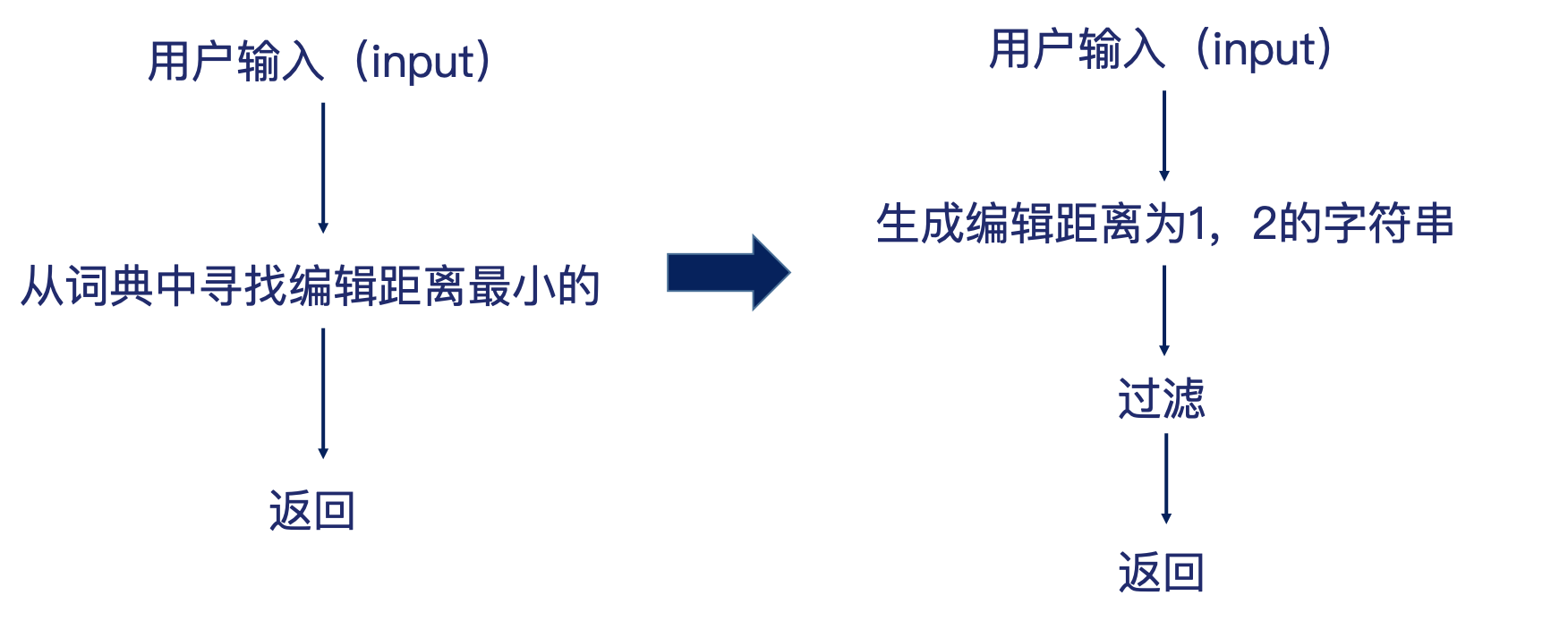
描述已自动生成

**3.2 dp算法**

图片包含 文字

描述已自动生成

**3.3 better way**



为什么只生成1，2距离：绝大部分情况正确单词都落在距离为2的字符串内。

**Select：**

图片包含 屏幕截图

描述已自动生成

S：输错字符。C：原本想表达的意思。

**4. Words Filtering**

对于NLP的应用，我们通常先把**停用词**、**出现频率很低**的词汇过滤掉

在英文里，比如 “the”, “an”, “their”这些都可以作为停用词来处理。但是，也需要考虑自己的应用场景

**5. Words Normalization**

**5.1 Stemming**



Stemming未必把一个单词还原到词典库的词根，它有自己的规则。

Porter Stemmer的一些规则：

https://tartarus.org/martin/PorterStemmer/java.txt

