* 用户输入
* 生成编辑距离为1，2的字符串
* 过滤
* 返回

过滤：

给定一个字符串s，我们要找出最有可能成为正确字符串c，也就是

例如：

输入：

正确：apple

用户1：app

用户2：appl

用户3：appl

用户4：app

用户5：appla

用户6：appl

：对于一个正确的字符串，有百分之多少人写成了s的形式

：Unigram probability

数据说明：

1. vocab.txt：词典文件，作为判断单词是否拼错的依据，任何未出现在词典中的词都认为是拼写错误。
2. spell-errors.txt：该文件记录了很多用户写错的单词以及对应的正确的单词，可以通过该文件确定每个正确的单词所对应的错误拼写方式，并计算出每个错误拼写方式出现的概率
3. testdata.txt：记录了一些包含拼写错误的单词的文档，用于最终的测试。

流程：

1. 找出拼写错误的单词，不存在与词典中的单词都认为拼写错误
2. 生成与错误单词编辑距离不大于2的候选单词，过滤掉不在字典中的单词
3. 根据贝叶斯公式选择最适合的单词

## 拼写纠错实验

### 问题描述

识别到拼写错误的单词，并给出正确的单词（返回最大的概率）

### 算法设计

1. 找出拼写错误的单词（不存在字典中的单词都属于拼写错误）
2. 生成与错误单词编辑距离为1、2的所有候选单词，并过滤掉不在字典中的单词
3. 根据贝叶斯公式选择最合适的单词

### 数据来源：

1. vocab.txt，词典文件，作为判断单词是否拼写错误的依据，任何未出现在词典中的单词都认为拼写错误
2. spell-errors.txt，该文件记录了用户拼写错的单词和对应的正确的单词，可以通过该文件确定每个正确的单词所对应的错误拼写方式，并计算出每个错误拼写方式出现的概率，例如：

raining: rainning, raning

writings: writtings

disparagingly: disparingly

1. test.txt，记录了一些包含拼写错误的单词的文档，用于最终的测试。例如：

In Taawin, businessmen and officials are also worried.

We are aware of the seriousnyss of the U.S.

1. news.txt，来自于wmt新闻领域的英文数据集，用于训练语言模型（默认Bigram）