《自然语言处理》课程项目

《诗经》取名生成器

罗齐尧

15307130105

编程语言：Python2.7

所需 Python 库：

pypinyin：获取中文汉字的读音和音调

jieba： 用于中文分词和关键词提取

程序清单：

```
main.py      # 处理的主函数，包括读洗数据、产生名字等
test.py      # 测试函数，用来测试小组件和功能
tmp.py       # 处理古风人名文本的代码
```

关键词：人名生成 古风 《诗经》 寓意 读音 平仄

思路来源：我的 NLP 项目 Project 设计一个人名生成器，产生这个想法是因为关注的知乎问题：“有哪些令你感觉惊艳的人名？”，所以我想设计一个系统，通过给定一些限制条件来生成有意义有内涵、读来朗朗上口的名字。

一、问题概述

人名是一个人终生的代号，很多父母希望为自己的孩子取一个非常有意义的名字来寄托望子成龙、望女成凤的心情。但最近新闻中也有报道父母为了让孩子有个特立独行的姓名，取了很多雷人的称呼，例如：“王者荣耀”，“端木女王”等等。这不是否认他们的想法和初衷，但可能因为意识形态的不足导致了这种奇怪人名的出现。

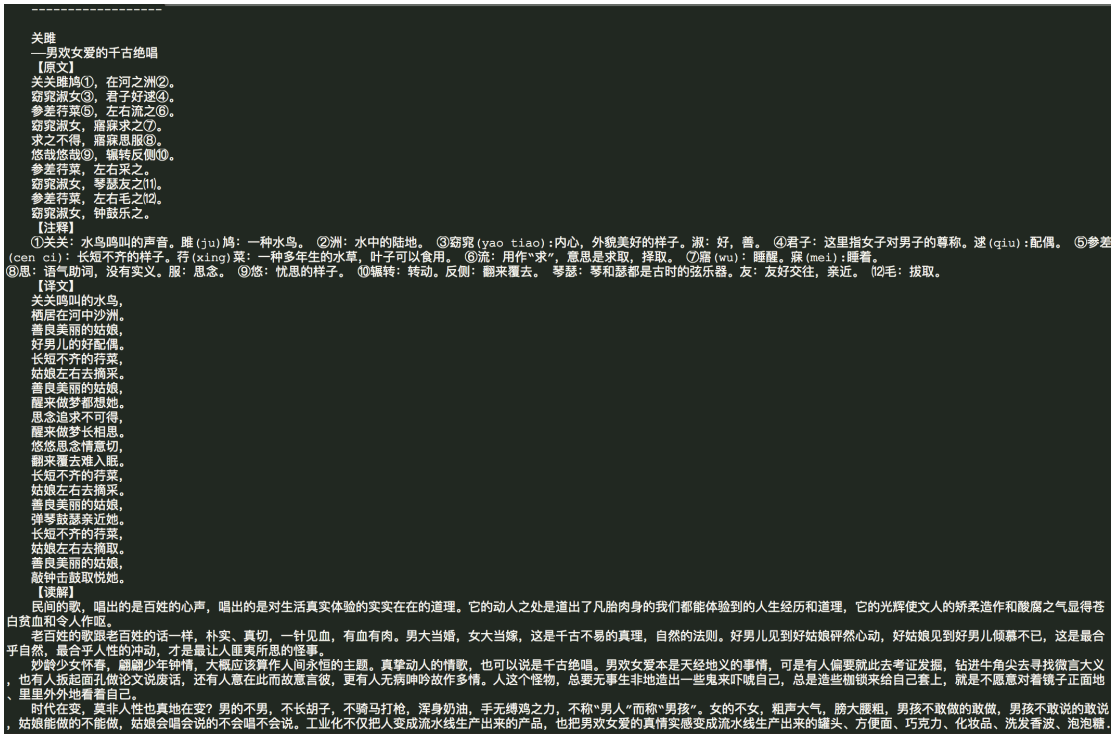
目前在互联网上也存在许多人名生成器，他们种类繁多，功能独特：包括人名、网名、昵称等等，但我从搜索的结果来看，很多都是随机抽取的几个汉字的拼接，没有实际的中文意义。如下方截取的某软件的取名结果，存在了很多生僻词，而且类似于“漾漾”这样的没有实际意义的形容词，又或是像“桃粉”这种可能会引起读音误会的词语。

而一些好的人们则会让人乍见惊艳，比如著名的屠呦呦教授。她的名字就取自诗经“呦呦鹿鸣，食野之苹。”又如医学家钟南山，“南山”是隐士的处所，出自于陶渊明《饮酒》：“采菊东篱下，悠然见南山。

人名本该是包含父母美好的寄托，一个好的人名往往会带有一定的典故或是寓意。所以我想做一个自动的古风姓名生成器，在给定一些限制条件的基础上，帮助打造一个个性人名。取出的名字能够有一定的含义，同时也要朗朗上口，没有歧义。

二、语料库获取和数据处理

首先帮人取名，需要有一定的文化水平。那么机器要完成这些是，就需要一定的语料库作为支持，这里我选取的是中国古代第一步诗歌总集《诗经》，也有一定收到“呦呦鹿鸣”的启发[1]，所以以《诗经》中的文本作为语料库进行处理。我从网上找到了相关文档[2]，这篇文档包含了几乎所有的诗经内容，同时也对文章进行了中文解析和情感分析，非常有利于给古文中的词语进行词性标注和关键词分析。其中的一篇文章内容如下：



关雎
—男欢女爱的千古绝唱
【原文】
关关雎鸠①，在河之洲②。
窈窕淑女③，君子好逑④。
参差荇菜⑤，左右流之⑥。
窈窕淑女，寤寐求之⑦。
求之不得，寤寐思服⑧。
悠哉悠哉⑨，辗转反侧⑩。
参差荇菜，左右采之。
窈窕淑女，琴瑟友之⑪。
参差荇菜，左右毛之⑫。
窈窕淑女，钟鼓乐之。
【注释】
①关关：水鸟鸣叫的声音。雎(ju)鸠：一种水鸟。 ②洲：水中的陆地。 ③窈窕(yao tiao)：内心、外貌美好的样子。淑：好，善。 ④君子：这里指女子对男子的尊称。逑(qiu)：配偶。 ⑤参差(cen ci)：长短不齐的样子。荇(xing)菜：一种多年生的水草，叶子可以食用。 ⑥流：用作“求”，意思是求取，择取。 ⑦寤(wu)：睡醒。寐(mei)：睡着。
⑧思：语气助词，没有实义。服：思念。 ⑨悠：忧思的样子。 ⑩辗转：转动。反侧：翻来覆去。 琴瑟：琴和瑟都是古时的弦乐器。友：友好交往，亲近。 ⑫毛：拔取。
【译文】
关关鸣叫的水鸟，
栖居在河中沙洲。
善良美丽的姑娘，
好男儿的好配偶。
长短不齐的荇菜，
姑娘左右去摘采。
善良美丽的姑娘，
醒来做梦都想她。
思念追求不可得，
醒来做梦长相思。
悠悠思念情意切，
翻来覆去难入眠。
长短不齐的荇菜，
姑娘左右去摘采。
善良美丽的姑娘，
弹琴鼓瑟亲近她。
长短不齐的荇菜，
姑娘左右去摘取。
善良美丽的姑娘，
敲钟击鼓取悦她。
【读解】
民间的歌，唱出的是百姓的心声，唱出的是对生活真实体验的实实在在的道理。它的动人之处是道出了凡胎肉身的我们都能体验到的人生经历和道理，它的光辉使文人的矫柔造作和酸腐之气显得苍白贫血和令人作呕。
老百姓的歌跟老百姓的话一样，朴实、真切，一针见血，有血有肉。男大当婚，女大当嫁，这是千古不易的真理，自然的法则。好男儿见到好姑娘砰然心动，好姑娘见到好男儿倾慕不已，这是最合乎自然，最合乎人性的冲动，才是最让人匪夷所思的怪事。
妙龄少女怀春，翩翩少年钟情，大概应该算作人间永恒的主题。真挚动人的情歌，也可以说是千古绝唱。男欢女爱本是天经地义的事情，可是有人偏要就此去考证发掘，钻进牛角尖去寻找微言大义，也有人扳起面孔做论文说废话，还有人意在此而故意言彼，更有人无病呻吟故作多情。人这个怪物，总要无事生非地造出一些鬼来吓唬自己，总是造些枷锁来给自己套上，就是不愿意对着镜子正面地、里里外外地看看自己。
时代在变，莫非人性也真地在变？男的不男，不长胡子，不骑马打枪，浑身奶油，手无缚鸡之力，不称“男人”而称“男孩”。女的不女，粗声大气，膀大腰粗，男孩不敢做的敢做，男孩不敢说的敢说，姑娘能做的不能做，姑娘会唱会说的不会唱不会说。工业化不仅把人变成流水线生产出来的产品，也把男欢女爱的真情实感变成流水线生产出来的罐头、方便面、巧克力、化妆品、洗发香波、泡泡糖。

[1] 《从屠呦呦的名字说起》http://news.163.com/15/1227/14/BBRO133900014AED.html
[2] 见附录《【电子书必备】《诗经》全文及注译(收藏版).txt》

虽然文档有大致的分类，但是仍然需要一定的字符串处理才能将其格式化成对应的模式。中文文本的处理流程如下：

1） 过滤所有的空行和回车；

2） 获取第一行的标题以及对这段诗篇的简单描述，及横线之后的部分；

3） 将原文、注释、译文、解读四段文字分别整合成一段字符串；

4） 对原文字符串处理：制作一个过滤器，过滤掉所有的标识符，保留下中文和标点符号；

5） 对注释字符串直接忽略；

6） 对译文和解读字符串，直接将若干跨行字符串合并成完整的即可；

7） 完成后将标题、梗概、原文、译文、解读按顺序存入一个 list 中，再将所有 list 合并成一个完整的语料库。

三、构造候选名字

首先从文档中构造一些候选名字，用户需要提供一些信息用以构造名字，如果没有提供信息，程序只能从文档中的关键词里找出若干构成姓名，则没有独特性可言，名字的构造方法大致分为三类：

1、 给定名字中的一个字

这里的应用场景通常有：名字中包含父母双方的姓，名字中有家族辈分的（字辈，也叫做字派，是指名字中用于表示家族辈份的字，多为名字中间的字）等等。

通常这就需要给定一个字作为条件，母亲的姓、孩子的字辈。然后就在语料库中找到一个与之搭配的词语构成名字的两个字组成，构造的方法采用了以下三种：

1） 找到这个字组成的 bigram，开头或者结尾都可以。在其中找到最常见的几个作为候选字对，但是很显然这里的 bigram 需要过滤掉原文中的标点符号，因为在古文中，每个字出现在各个位置的概率都是存在的，没有一定的语法逻辑性可言，所以需要过滤掉标点符号。以“齐”字为例，在所有 bigram 之中，出现最多的 bigram 为（“齐”，“。”），其次才是（“齐”，“之”）、（“齐”，“圣”）等文字搭配，所以在 bigram 处理的时候过滤掉所有标点。在找到的这些 bigram 里面，分别摘取的诗句是：“必齐之姜岂其食鱼”，“有怀二人人之齐圣”，其中“齐圣”二字就能体现出一种对孩子未来的殷切期待。

【方法】：这里就采用上课时描述的 bigram 方法，加上了 stopword 的过滤设定。

2） 找到这个字在句中的对应字。古诗中常常讲究平仄想合、字字对应。所以在一句诗中间，会有一一对应的字词存在，比如诗经《采薇》中有一段“昔我往矣，杨柳依依。今我来思，雨雪霏霏。”可以看到“依依”和“霏霏”相对，各取一字组成“依霏”，不失为一个好的名字，寓意女子美好淡然。

【方法】：对应词语的获取，由于古诗中非常讲究对仗工整，所以对应词语在句子中出现的位置应该也是相同的，所以找到这个词所在句子的位置，对应到前后两句中的同样下标的位置，找到对应的字。

3） 根据词性关系找到这个字之前和之后的字。因为古诗中会有很多感叹词或者连接词，尤其是在《诗经》这样的用以歌唱的诗词中，叹词会较为明显，那么这时候 bigram 找到的词语就没有一些实际意义的搭配。虽然在某些名字中加入叹词不会有什么大的影响，例如：“王羲之”中的“之”字。但例如在《鱼丽》中“维其嘉矣”，若以“维”字为参考，如果直接以 bigram 找到的人名是“维其”，但显然这里的“维其”中的“其”字只是一个指示代词，而如果找到真正的对应词“嘉”字便是“维嘉”，也确有人名此。而这里就需要对句法进行分析。

【方法】：句法分析利用了结巴分词工具。首先利用结巴分词对语句进行分词，确定出一些单独的字或者词，这样就可以忽略到一些固定搭配的二元或者三元词组，然后重新对整个句子进行磁性分析，忽略其中的连词（c）、叹词（r）、量词（o）、代词（r）、助词（u）等辅助修饰型词语，主要保留名词（n）和动词（v）。例如对“维其嘉矣”，分析出来的结果为：“维 v / 其 r / 嘉 nr / 矣 zg”，丢弃相关修饰词之后得到的重要名词和动词就是“维嘉”二字。下图为句法分析的运行结果：

```
sen = u"维其嘉矣";
seg_list = jieba.cut(sen, cut_all=True);
ls = ','.join(seg_list);
lst = ls.split(',');
print ls;
print lst;
for i in lst:
    words = jieba.posseg.cut(i)
    for w in words:
        print w.word, w.flag;
```

```
Building prefix dict from the default dictionary ...
Loading model from cache /var/folders/b4/dyn5s_9s14s25dmj0dstr_r80000gn/T/jieba.cache
Loading model cost 0.412 seconds.
Prefix dict has been built succesfully.
维,其,嘉,矣
[u'\u7ef4', u'\u5176', u'\u5609', u'\u77e3']
维 v
其 r
嘉 nr
矣 zg
```

2、 给定名字中的一个音

在有些取名中，可能会有一些特殊的字但放在名字中不好构名，需要一个谐音字，所以这时候就需要用一个音相同的字来构造名字。这时候取名的要求限制就是一个拼音。这里用到了 pypinyin 库，利用 pypinyin.pinyin(sentence, style=pypinyin.TONE2)语句可以为句子中所有的字标注拼音和音标:

```
>>> import pypinyin
>>> sent = u"鸿雁于飞，肃肃其羽。";
>>> pypinyin.pinyin(sent, style=pypinyin.TONE2);
[[u'ho2ng'], [u'ya4n'], [u'yu2'], [u'fe1i'], [u'\uff0c'], [u'su4'], [u'su4'], [u'qi2'], [u'yu3'], [u'\u3002']]
```

这里的数字表示的是音标，所以编写了程序 pinYin()获取一个字的拼音和音标，将他们分开存储成 list 的两个元素，音标之后会在平仄处理时用到。

```
def pinYin(c):
    res = pypinyin.pinyin(c, style=pypinyin.TONE2);
    lang = res[0][0];
    pyin = "";
    tone = -1;
    for i in lang:
        if i.isalpha():
            pyin += i;
        else:
            tone = i;
    return [pyin, tone];
```

```
>>> def pinYin(c):
...     res = pypinyin.pinyin(c, style=pypinyin.TONE2);
...     lang = res[0][0];
...     pyin = "";
...     tone = -1;
...     for i in lang:
...         if i.isalpha():
...             pyin += i;
...         else:
...             tone = i;
...     return [pyin, tone];
...
>>> list = pypinyin.pinyin(sent, style=pypinyin.TONE2);
>>> lst = [];
>>> for i in list:
...     lst.append(pinYin(i));
...
>>> print lst;
[[u'hong', u'2'], [u'yan', u'4'], [u'yu', u'2'], [u'fei', u'1'], ['', u'\uff0c'], [u'su', u'4'], [u'su', u'4'], [u'qi', u'2'], [u'yu', u'3'], ['', u'\u3002']]
```

这样就将拼音和音标分离出来，然后对文章中所有原文内容建立一个 pinyin 的字典，查找所有的字他们的拼音是否和给定需要的拼音字符串相同，如果相同就把这个字加入待选列表中，最终综合给出所有符合的字。

然后就获得了上述拼音对应的一些可选字的集合，在对找到的字的集合去除重复之后，对这些字集合进行处理。现将出现的所有字去重，以“qi”这个音为例，找到的大多数字都是指示代词“其”，所以采用上个方法的（3），对找到这个字给定一个词性，主要采用动词和名词词性对应的字词。

3、 给定名字的寓意

这里的应用场景通常有：父母寄托的情感和期待，或者一些寓意等等。例如希望孩子一飞冲天父母可能会取名“展鹏”，又例如孩子五行缺木，会用“森”、“林”等有木的字。所以目标是可以根据给定的一些寓意产生相应的名字。

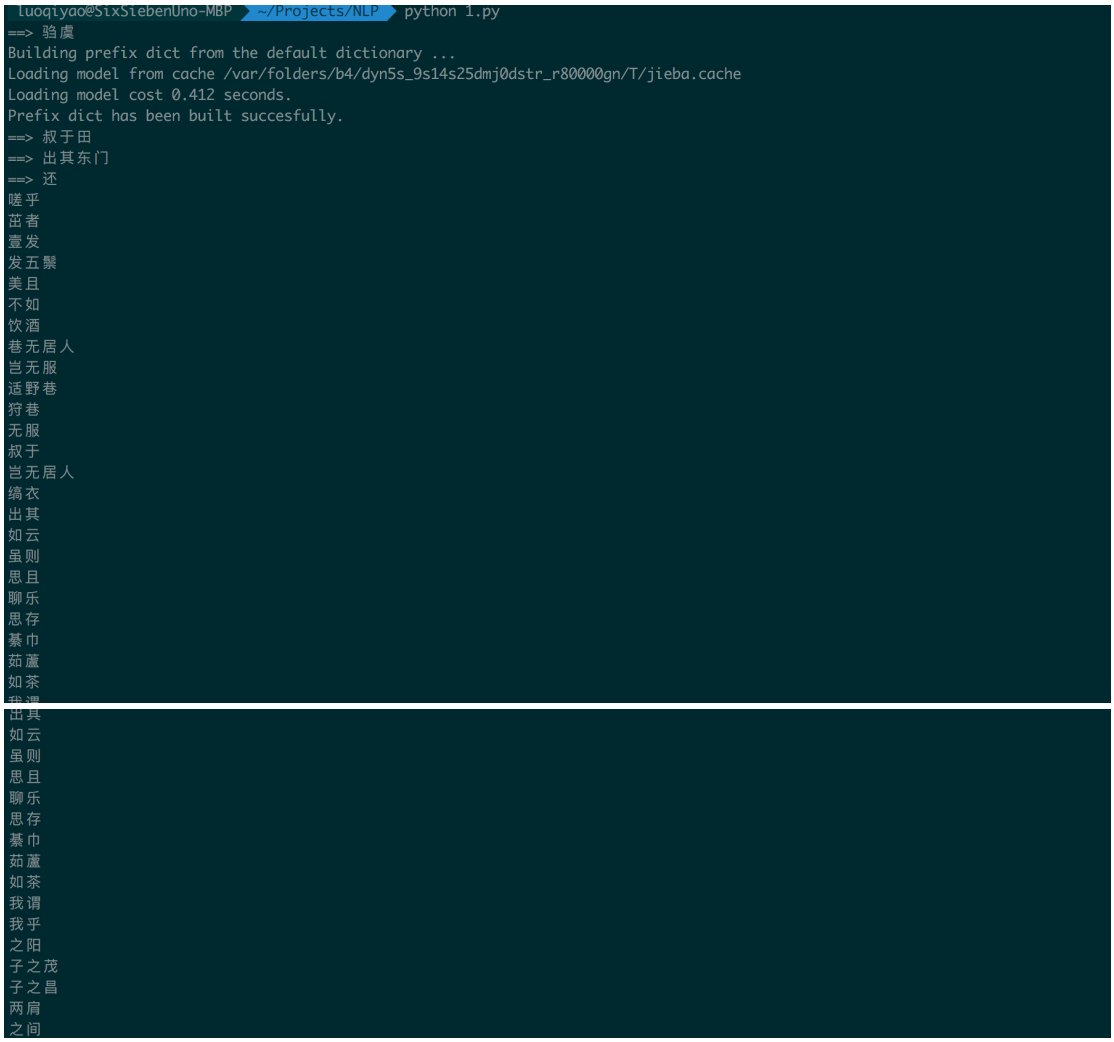
如何明确诗句的寓意，通过直白的文言文译文很难找到文章的感情，现有的 NLP 技术我也没有发现很好的感情分析的工具，但是我在处理文本的时候会发现文章中给了梗概和解读，所以就可以帮助获得这篇文章大致的情感取向。所以可以通过给定的定位获得相应的文章。

这里的训练集显然不能使整本《诗经》，因为与要求的立意可能有些不太契合，所以单独从某一些诗句中考虑。这些诗句就是上面通过立意筛选出来的有相关性的诗歌。

以“男子汉”这个寓意为例，找到的《诗经》中的内容有《驺虞》（——男子汉的赞歌）、《叔于田》（——骑马射箭的男子汉）、《出其东门》（——坐怀不乱的男子汉）、《还》（——赞美男子汉）。然后需要从每篇文章中提取一些关键字来，用以作为候选的词语。

对于提炼每个文章中的关键词，采用了 jieba 分词中的关键词提取功能，利用 jieba.analyse.extract_tags(sentence,topK)函数获得整个诗词中的关键词。找到

的关键词如下：



显然有一些词提取的并不是很关键，但有些词还是可以体现一定的代表性，例如驺虞、之阳等等。所以仍然对其加上词性分析，这里主要提取一些形容词、名词、动词。同时需要严格限制每个词的长度，人名一般在一到两个字之间。之后对于提取出来的这些关键词，做成一个列表作为待选词语列表。

四、筛选名字

通过上面三种构造名字的方法，在给定一些限定的时候就可以构造出一些名字了。但是注意到这里的名字空有一些文字上的意义，没有考虑到发音上的问题。名字是用来被称呼的，所以不光要有寓意，读音也非常的重要，所以在构造出来

的名字列表中，加入读音的选项：对读起来朗朗上口的名字提高优先级；对产生读音歧义的名字则从待选列表中删除。

考虑以下两类因素影响：如何判断一个名字是否读来上口好听?

如何判断一个名字会否产生歧义?

1、 名字可读性是否很高

1） 拼音构成

读来顺口，中国的汉语拼音只有几个固定的格式，比如“ing”、“ong”、“ang”等音节，两个音如果读起来较为相似那么显然他们之间拼音上的差距也就不会太大。比如上面的“ing”、“ang”差距只有一个字符，所以读来也很相似，而“ing”和“ou”两个音发起来就差别很大。所以我的想法是找到一些顺口的词语或者人名[3]，用他们作为训练集或者说标准集，用构造出来的新名字与这些名字比较，如果编辑距离越小那么这个名字可读性就越好。

【方法：编辑距离】

考虑两个音节如何变化，一个拼音字符串如果通过增加元素、删除元素、替换元素的方法变成另一个拼音字符串，次数越少必然约接近，这里就考虑变成相同的最少步骤，也就是编辑距离问题。

那么这就考虑两个字符串的编辑距离，可以采用的操作是添加、删除、替换。用动态规划的思想来解决这个问题：

f[i][j] 字符串 A 前 i 个字符匹配字符串 B 前 j 个字符的最少编辑步数

f[i][j] = f[i-1][j-1] (A[i] == B[j])

f[i][j] = min(f[i-1][j], f[i][j-1], f[i-1][j-1]) + 1 (A[i] != B[j])

[3] 数据取自《知乎-650 个古代经典人名》，https://zhuanlan.zhihu.com/p/21303404

最终答案为 f[len(A)][len(B)]

2） 声调平仄

古人作诗讲究平仄相合，起名的时候也有这样的讲究，例如，一个人的名字全是去声（第三声）那么读来一定非常不顺口，根据语言学家的经验总结，三连平或是三连仄的音是不可取的，通常平仄交替出现的名字读来才比较顺口。

【方法】：在处理这个问题的时候，就用到了之前 pypinyin 处理的音标，根据音标区分出平仄之声（“1”声为平，“2”、“3”、“4”为仄），然后排除掉所有不符合平仄要求的构造名字。也就是根据声调序列排除一些不合法结果，利用正则表达式表达就是：“(111)|([2-4]{3})”

2、 名字是否会产生读音歧义

读音歧义就是如果名字的读音和别的常用词语产生混淆就容易给孩子成长中带来一些不必要的麻烦。如果谐音的词语带有褒义还好，如果有贬义性的词语，同时被同学起了外号就会给生活带来一些不必要的麻烦。例如我曾经看到过一篇文章，作者自称姓名为“朱亦知”，但谐音却有些不雅，给他的生活带来了很多困扰。所以我们的取名就考虑到了这样的问题，如果我们的名字加上给定的姓之后的读音与一些日常生活用词有冲突，那么就考虑删掉这些带选项。

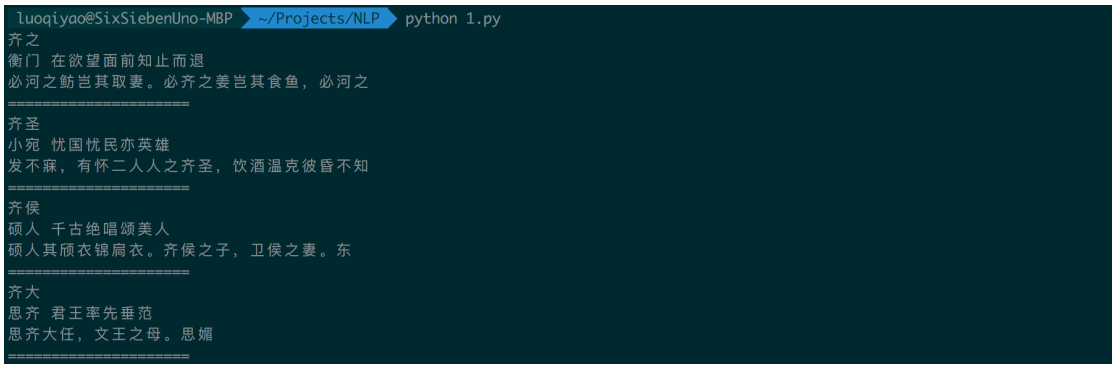
这里的冲突就定义成：两个词语的拼音字符串编辑距离小于等于 1，两个字符串几乎相同，则很容易产生误解。这也是非常直观而且不违反常理的想法，这个仍然使用之前编辑距离的处理方法即可。

通过从构造的候选列表中删掉这些元素来消除读音歧义。

五、名字与寓意结合

在经过三和四的构造和筛选之后得到的最终结果就是我们构造出来的名字，那么最终的输出方案就是给定一些生成出来的名字，同时给定他们的出处，来自于《诗经》中的哪一部中的哪一句，以及这句是的寓意和期待。

这就需要实现对给定的词语找到对应的诗句的出处，也就是类似于 nltk.concordance 功能，在原文中找到对应的文字出处，然后截取相应的部分输出，也就是类似于实现中文版的 concordence 功能。通过对字符串进行比较操作完成。以搜索“齐”字辈之后跟的名字结果，如下：



输出结果分别是：

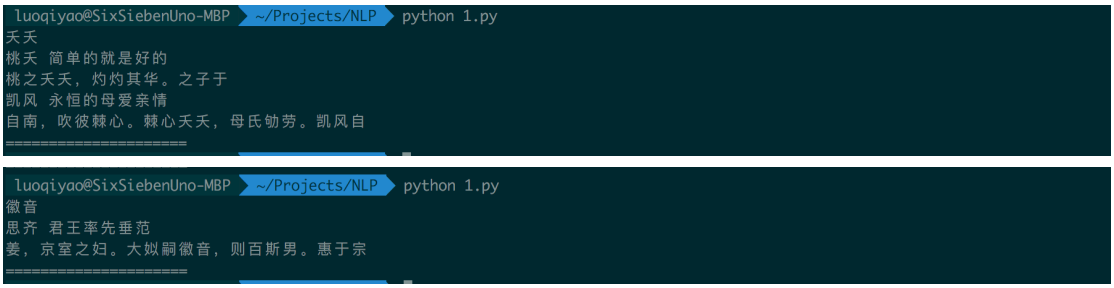
取名

文章名称 文章梗概

文章的具体诗句

最终通过取名生成器，调整其中一些参数设置，生成出来一些名字，从中找到一些具有代表性的好名字：“夭夭”（取自“桃之夭夭，灼灼其华”，看见春天柔嫩的柳枝和鲜艳的桃花,联想到新娘的年轻貌美）用以对女孩的美好期待；“圭璋”（取自《卷阿》“令闻令望，如圭如璋”）君子以玉比德，期望孩子品德高

尚；“邦彦”（取自《郑风 · 羔裘》“彼其之子,邦之彦矣”）是成为国家的栋梁之才，等等。



六、总结

在完成这个 PJ 的过程中，我更加熟悉了自然语言处理的一些技巧和方法，更加熟练的学会了 Python 的字符串处理方法。同时也思考并分析了如何判断中文相似度的问题，比如说在处理两个读音是否相似的问题上，就将拼音和字符串距离的问题联系在了一起，这是一种将理论模型和实际应用结合的样例。

在完成 PJ 的过程中，我认为整个 PJ 的难点在于中文信息的处理，从网上下载的数据大多会有很多杂项，而将他们清理干净变成程序可以识别的字符串是一件非常复杂的事情，获取诗经数据、常用名字等信息花费了整个 PJ 接近一般的时间；另一方面，中文分词尤其是对于古文的分词并没有完整的工具，不像 nltk 等工具对英文有全面的工具，很多类似的 nltk 方法实现在中文文本中就需要自己实现，比如 nltk.concordance 函数就是我手写 findSent(sentence)函数。最重要的一点，也可能是我自己选择语言的问题，选用了对编码支持不佳的 Python2.7 而不是 Python3，所以导致经常函数会出现编码错误的报告，程序内部 unicode 和 utf8 编码的冲突，经常函数会出现编码错误的警告。

但这个生成器在产生名字的时候仍然，会产生一些无效的名字，关系不是很大的虚词列表，虽然设计了一些消除的过滤器，但仍然有一定的缺陷。所以最终还是需要人工的筛选。

最后，考虑项目之后的想法和改进措施，一个是扩大语料库的范围，可以加入更多的古文语料，开发更多的古文专区，古言曾对取名有过这样的评价“男楚辞女诗经”，意思是男孩的名字从《楚辞》中取，女孩的名字从《诗经》中取，当然对于之后的唐诗宋词里的巧妙词句也是可以拿来加以运用的；另一个就是强化对文言文的句法分析，之前所做的句法分析只是单独的针对词性提取，重点考虑了名词和动词，猜想是否可以增加一维向量考虑每个词和主题之间的关系程度，这样可以更好的提取出文章的关键词句，更能提炼出好的名字。

PJ 完成略有仓促，不足和缺陷在所难免，希望老师和助教悉心指出，我会在之后对此加以改进。谢谢！