Data-Mining experiment report

小组成员：
15331061 邓旺
15331059 邓松华

What features do you use in your classifier? Why are they important and what information do you expect them to capture?

根据在project中给出的发音表：

- Vowel phonemes: AA, AE, AH, AO, AW, AY, EH, ER, EY, IH, IY, OW, OY, UH, UW
- Consonant phonemes: P, B, CH, D, DH, F, G, HH, JH, K, L, M, N, NG, R, S, SH, T, TH, V, W, Y, Z, ZH

Vowel	IPA	consonant	IPA	consonant	IPA
AA	ɑ	P	p	S	s
AE	æ	B	b	SH	ʃ
AH	ə / ʌ	CH	ʧ	T	t
AO	ɔ	D	d	TH	θ
AW	aʊ	DH	ð	V	v
AY	aɪ	F	f	W	w
EH	ɛ	G	g	Y	j
ER	ɜːr	HH	h	Z	z
EY	eɪ	JH	ʤ	ZH	ʒ
IH	ɪ	K	k		
IY	i	L	l		
OW	oʊ	M	m		
OY	ɔɪ	N	n		
UH	ʊ	NG	ŋ		
UW	u	R	r		

发现元音字母都是以‘A’，‘E’，‘I’，‘O’，‘U’开始的双字母音节，而辅音字母有的是单字母音节，有的是双字母音节，但是都不包含元音字母，因此，可以通过验证一个字母给出的pronunciation的组成音节中是否满'XY' (X in [‘A’，‘E’，‘I’，‘O’，‘U’])的格式，如果满足这个特征，说明该音节为元音音节。当通过读入文件把发音传入后，获取每个单词的发音，逐个验证是否满足特征表达式，然后获取满足该features的音节。

我们实验的目的是预测单词的重读，根据重读的规则：

- A word only have one pronunciation (we do not consider words with multiple pronunciations)
- A word must have one and only one primary stress
- Only vowels are stressed

发现只有元音才被重读，因此，要预测一个单词重读的位置，捕捉单词的元音音节就显得非常重要，通过上述分类器的特征，就能够很好的捕捉元音音节，从而正确预测一个单词重读的位置，达到实验目的。

How do you experiment and improve your classifier?

通过以上分类器，我们基本能实现预测和捕捉单词的重读。但是，在实验过程中，我们发现给出的测试样例中存在如'AY'这样的音节，该音节并不是元音音节，但是任然满足'XY' (X in ['A', 'E', 'I', 'O', 'U'])的格式，假如说我们还是按照上面规定的特征来写分类器的话，就会产生实验误差，因此要进行优化。

接下来的实验中，我们发现要捕捉的元音表[AA, AE, AH, AO, AW, AY, EH, ER, EY, IH, IY, OW, OY, UH, UW]可以通过列举的方式来写出特征，即
如果该音节的首字母是'A',那么第二个字母只能是'A', 'E', 'H', 'O', 'W';
如果该音节的首字母是'E',那么第二个字母只能是'H', 'R', 'Y';
如果该音节的首字母是'I',那么第二个字母只能是 'H', 'Y';
如果该音节的首字母是'O',那么第二个字母只能是'Y', 'W';
如果该音节的首字母是'U',那么第二个字母只能是'H', 'W';

```
def extract_vowels(phonetic):
    vowel_alphabet = ['A', 'E', 'I', 'O', 'U']
 sec_p_a = ['A', 'E', 'H', 'O', 'W']
 sec_p_e = ['H', 'R', 'Y']
 sec_p_i = ['H', 'Y']
 sec_p_o = ['W', 'Y']
 sec_p_u = ['H', 'W']
 ies = phonetic.split()
 s = list()
    for i in ies:
  if i.startswith('A') and i[1] in sec_p_a:
   s.append(i)
   continue
  elif i.startswith('E') and i[1] in sec_p_e:
   s.append(i)
   continue
  elif i.startswith('I') and i[1] in sec_p_i:
   s.append(i)
   continue
  elif i.startswith('O') and i[1] in sec_p_o:
   s.append(i)
   continue
  elif i.startswith('U') and i[1] in sec_p_u:
   s.append(i)
 return s
```

通过这种更严格的特征表达式来进行验证，可以消除部分音节对实验结果的影响，从而减少实验的误差，达到更好的实验效果。