BUG543-JiWenkai

问题说明

现有的词汇量评分机制不合理，导致放入大量单词时，词汇量评分为最近放入的10个单词

现有评分机制是：

$$\left(\prod_{i=1}^{n} difficulty_i\right)^{\frac{1}{n}} \qquad n \leq 10$$

那在生词本积累大量词汇的时候显然是不合理的，在原本有大量的困难词汇突然放入10个简单词汇会给一个很低的词汇量评分，这也是bug description里面只有3.0分的原因

问题改进

从直觉上来说，一个人如果没有掌握简单的单词，那么困难的单词可能也没有掌握，所以简单的单词的权重要相对更大。然而，在生词本中单词数量变多的情况下，如果高难度的单词占比提升，那么相对来说词汇量的评分也要相对变高，也就是说高难度词汇的权重要变大

所以可以给出一个比较简单的计算公式

设难度从3到8，每种难度的单词数量分别为$(N_3, N_4, N_5, N_6, N_7, N_8)$，总单词数量为 (N)。

1. 计算每种难度的占比： $p_i = \frac{N_i}{N}$ 其中 $(i = 3, 4, 5, 6, 7, 8)$。
2. 计算每种难度的权重： $weight_i = (10 - difficulty_i) \times p_i$ 其中$(i = 3, 4, 5, 6, 7, 8)$。
3. 归一化权重： $normalized\_weight_i = \frac{weight_i}{\sum_{j=3}^{8} weight_j}$
4. 计算加权平均难度： $average\_difficulty = \sum_{i=3}^{8} (difficulty_i \times normalized\_weight_i)$

综合以上步骤，公式如下：

$$average\_difficulty = \sum_{i=3}^{8} \left(difficulty_i \times \frac{(10 - difficulty_i) \times p_i}{\sum_{j=3}^{8} (10 - difficulty_j) \times p_j} \right)$$

在该公式下，**各个难度的单词占比相同的情况下，权重随难度提升而下降**，效果可以见**改进效果**中图片的最后一个测试用例

这里也给出一个更灵活的公式，$k0, k1$分别为难度系数和占比的影响因子（未使用）

$$average\_difficulty = \sum_{i=3}^{8} \left(difficulty_i \times \frac{exp(k_0(10 - difficulty_i) + k_1 p_i)}{\sum_{j=3}^{8} exp(k_0(10 - difficulty_j) + k_1 p_j)} \right)$$

改进效果

description中生词本词汇量评级为3.5

下面是对一些数据的测试情况，字典中键是难度。

```
(base) ➜ py python a.py
percentages: {1: 0.3, 3: 0.1, 4: 0.2, 5: 0.4}
weights: {1: 0.40909090909090906, 3: 0.10606060606060608, 4: 0.18181818181818185, 5: 0.30303030303030304}
average difficulty: 2.9696969696969697
(base) ➜ py python a.py
percentages: {1: 0.1, 3: 0.1, 4: 0.2, 5: 0.6}
weights: {1: 0.15517241379310343, 3: 0.12068965517241378, 4: 0.20689655172413793, 5: 0.5172413793103448}
average difficulty: 3.93103448275862
(base) ➜ py python a.py
percentages: {1: 0.25, 3: 0.25, 4: 0.25, 5: 0.25}
weights: {1: 0.3333333333333333, 3: 0.25925925925925924, 4: 0.2222222222222222, 5: 0.18518518518518517}
average difficulty: 2.9259259259259256
```

从测试效果来看，是符合直觉的权重比。

在原有公式上进行修改

修改后的公式，选择用全部的单词进行统计而不是最近的十个，可以使用对数计算的方式避免溢出

$$\left(\prod_{i=1}^{n} difficulty_i\right)^{\frac{1}{n}} = \exp\left(\frac{1}{n}\sum_{i=1}^{n}\ln(difficulty_i)\right)$$

进行化简后公式的计算更不容易溢出，此时description中生词本词汇量评级为3.8，与上文提出的方法大致相近