

这是一篇发表在《中国新闻周刊》上的文章的同行评议后的预编辑版本。基于语义的结构化数据源关键词搜索。IKC 2017。Lecture Notes in Computer Science, vol 10546, p. 70-79.最终的认证版本可在网上查阅：[https://doi.org/10.1007/978-3-319-74497-1\\_7](https://doi.org/10.1007/978-3-319-74497-1_7)

## 评估类似问答游戏的单词难度

Jakub Jagoda 和 Tomasz Boin´ski

计算机结构系

格但斯克科技大学电子、电信和信息学系

[tobo@eti.pg.gda.pl](mailto:tobo@eti.pg.gda.pl)

**摘要。**映射验证是一项艰巨的任务。我们的研究旨在提供一个框架，使用众包方法对映射进行人工验证。为此，我们计划实施一个类似测验的游戏。为此，必须对映射的难度进行评估，以便更好地根据游戏级别来展示文本。在本文中，我们提出了一种评估单词难度的算法。本文介绍了三种方法，并展示了实验结果。还提供了未来工作的计划。

### 1 简介

在我们的研究中，主要是在Colabmap<sup>1</sup>项目中，我们创建了一套英语维基百科文章和WordNet同义词之间的映射[1-4]。每个映射由一个WordNet同义词、同义词的定义和维基百科文章的标题组成，并使用RFC 3986进行编码。这些映射，一旦被证明是正确的，将允许维基百科结构的形式化。获得的映射集包含通过算法创建的54475个连接。我们的目标是创建100%正确的映射体，因此这组映射需要人工验证。

2006年，Luis von Ahn提出将计算机游戏作为比纯娱乐更重要的东西来使用，从而创造了所谓的GWAP（有目的的游戏）[5]。GWAPs是典型的游戏，它提供了用户期望的标准娱乐价值，但其设计方式允许通过解决一个需要智力活动的问题来产生附加价值。值得注意的是，GWAPs并不允许对工作进行经济上的满足。继续游戏的意愿应该被视为满足用户的唯一方式[6]。在三星调查期间获得的结果的诱惑下，我们决定按照基于人的计算模型[8]，实施一个GWAP来验证这些联系[7]。

最初获得的映射被扩展为三个额外的“次佳”映射，其目的是向用户提出一个问题（一个同义词的定义），并有四个可能的答案（维基百科的文章标题）。开始时，另外3个答案是从维基百科的页面中随机选择的，但是

<sup>1</sup><http://kask.eti.pg.gda.pl/colabmap>

这种方法很快被证明是不正确的，因为 "下一个最佳 "的映射与问题完全无关。相反，我们使用维基百科的搜索功能来选择备选答案（根据维基百科）。为了验证，我们实现了一个名为TGame<sup>2</sup> ("Tagger Game") 的二维平台游戏，作为一个二维平台游戏，遵循输出-协议模型[9]。

在下一步的映射验证中，我们决定实施一个类似于 "谁想成为百万富翁 "的测验游戏（有一个小小的区别，那就是我们主要问的不是真正的问题，而是由两个或多个单词组成的单词或短语）。这里的一个主要问题是如何安排所问问题的顺序？很明显，当玩家开始游戏时，他或她不应该立即得到一个困难的问题--否则玩家会很快离开游戏，并会感到气馁。难度应该从一个相对较低的水平开始，在玩家回答接下来的问题时增加。为此，我们需要以某种方式将单词从最难到最难排序。在本文中，我们提出了一种验证这种排序的单词难度的方法。

本文的结构如下。第2节定义了什么是文字困难并介绍了一些方法。第3节介绍了拟议的方法。第4节给出了对拟议算法的评估，最后第5节给出了最终的结论，并提出了一些未来可以做的工作，以进一步加强拟议的解决方案。

## 2 定义单词的难度和相关工作

为了评估单词的难度，我们需要一种算法来对只有那个单词进行分类，也就是一组标记（这里是字母）。这样的集合对计算机程序来说并没有真正的意义，因为它们对普通人来说是有意义的。

有许多方法可以解决这个问题。然而，它们中的大多数都集中在评估整个文本[10, 11]，并经常利用复杂的技术，如Coh- Metrix[12]。在我们的案例中，需要的是单字，而不是整个文本的特征。人们还可以找到大量与单词有关的游戏和谜题，它们使用各种评分系统和分类策略。例如，拼字游戏根据每个字母在特定语言中的频率使用不同的分数，促进那些 "罕见 "的字母获得更高的分数。在这种情况下，一个困难的单词可能是一个由罕见字母组成的单词。在这种情况下，单词的含义或复杂性并不重要，重要的是它所包含的字母的分数。另一方面，如果游戏中玩家的任务是在标准的QWERTY键盘上尽可能快地输入单词，那么这些规则可能就不重要了--相反，难写的单词是指难以书写的单词，例如，它的字母之间有一定的距离，或者需要用不同的手指来输入它。因此，为了正确地对单词进行分类，我们需要定义哪些单词是我们要考虑的困难。

<sup>2</sup><https://play.google.com/store/apps/details?id=pl.gda.eti.kask.tgame>,  
<http://kask.eti.pg.gda.pl/tgame/>



人们有很多方法来判断他们认为一个词是否困难。我们可以假设，一个普通人知道一组普通的单词，所以大多数人对单词的难度分类或多或少都是相似的。当然，让一群人按难度排列单词，最后得到完全相同的阵容是极不可能的。而是要求他们给这些词贴上一些标签，例如“在1-5的范围内，这个词对你来说有多难？”。我们可以认为，对于一个普通人来说，一个困难的词是一个不经常看到的词，当他们看到这个词时，它的结构只让他们对它的起源或可能的含义有一点了解，如果有的话。很明显，一组常见的日常用语是存在的，并且在普通人中具有很大的共同部分。同样清楚的是，一个与某些特定领域没有任何关系的普通人可能会认为来自该领域的词是困难的，因为这些词对他们来说不是日常用语。因此，在我们的案例中，我们将单词的难度定义为从1到5的分数，说明这个人知道这个单词的含义的可能性有多大。

研究表明，一个词的频率可能与它的难度相关[13]。观察到在文本中出现频率较低的词可以被认为是更难的，这是估计词的难度的最好和最广泛使用的方法之一。这样的观察结果导致了统计学词汇评估的产生，将难度定义为衡量该词在给定领域或日常生活中出现的频率，也就是说，它有多容易被看到。如果这个词在文本中经常出现，那么对于大多数人来说，它就被视为容易的，反之亦然[13, 14]。这种方法也导致了許多现代文本评估服务的实现，如Twinword API [15]。

### 3 我们的方法

在我们的解决方案中，我们评估了与维基百科文章相关的单词的难度。因此，我们决定使用维基百科作为文本发生率分析的语料库。在所有的情况下，我们对来自英语维基百科的5000多个页面的映射子集进行了评估，每个案例都是在之前的解决方案基础上进行的。

#### 3.1 天真的方法

第一种算法（算法1）是直接基于第2节中给出的观察结果。

该算法的结果是一组从最容易到最难的单词**W**的排序。这种方法有一些局限性。

- 一个即使在各种类型的文本中也非常罕见的短小而简单的单词可能被错误地归类为一个困难的单词，例如 "moo" 这个单词，即使是非常年轻的儿童也清楚地知道。
- 很多词都得到相同的分数。
- 使用的语料库必须有一个平均的单词分布，否则会产生不正确的结果，这一点很难用维基百科来验证。



---

### 算法1 天真的方法

---

- 1: 阅读文本语料库**T**
  - 2: 阅读一组词, 对**W**进行分类
  - 3: 对于集合**W**中的所有单词**w**, 做
  - 4: 计算**W**在文本中出现的次数 **T**
  - 5: 将结果**r**与**w** 分配。
  - 6: 结束
  - 7: 按出现次数**r**升序排列**W**中的词。
- 

### 3.2 增加字长

在这种方法中, 我们决定将被分析的单词的长度纳入算法, 以减少上一节中给出的前两个限制的影响。在这种方法中, 我们可以把困难的词看作是那些既罕见又长的词。短的词以及经常出现的词, 将更有可能被归类为容易的词。

在实施过程中我们还发现, 根据文本语料库和输入词的大小正确调整权重系数(长度和出现次数)是非常困难的, 因此我们决定使用相对值。该算法如算法2所示。

---

### 算法2 第二种方法

---

- 1: 阅读文本语料库**T**
  - 2: 阅读一组词, 对**W**进行分类
  - 3: 选择**W**中最长的字, 并将其长度存储为**lMax**
  - 4: 从集合**W**中找出在文本**T**中出现次数最多的词, 并将出现的次数存储为**oMax**。
  - 5: 对于集合**W**中的所有单词**w**, 做
  - 6: **c** = 文本中**w**出现的次数 **T**
  - 7: **lw**=**w**的长度
  - 8:  $S = \frac{lw}{lMAX} * \left(\frac{c}{oMAX}\right)$
  - 9: 将分数**s**与**w** 分配。
  - 10: 结束
  - 11: 将**W**中的单词按分数**s**降序排列。
- 

再一次将得到的单词集**W**从最容易的单词到最难的单词进行排序。分数的计算方法是将单词的长度和文本中出现的次数相乘。在这种方法中, 最难的词(最低分)是长和出现次数最少的组合。如果两个词有相似的频率得分, 那么较长的那个词就会变得更难。在所有情况下, 得分都是在现有文本中相对计算的。



表1.文本难度的组别

分数	笔记
90.0及以上	非常容易阅读, 对一个11岁的孩子来说很容易理解
80.0 - 90.0	易于阅读, 适合普通人的英语会话
70.0 - 80.0	相当容易阅读
60.0 - 70.0	通俗易懂的英语, 容易被13至15岁的学生理解
50.0 - 60.0	相当难读
30.0 - 50.0	难以阅读
30.0及以下	非常难以阅读

### 3.3 最后的方法

在这个方法中, 我们试图处理第三个问题。想象一下, 我们的文本集由十个长度相似的文本组成: 其中五个是关于分布式计算的学术论文, 五个是蛋糕食谱。如果只有一个蛋糕需要用到香蕉, 那么 "香蕉" 这个词可能会比 "矩阵" 这个词出现的次数少, 这意味着前者会被归类为更难, 而我们可能会认为对于一个普通人来说, 情况正好相反。为了消除这个问题, 我们可以考虑这个词是来自简单还是困难的文本, 并对所得到的分数应用适当的权重。为了确定一个文本是否困难 (在英语中), 我们可以使用现有的方法之一。在我们的解决方案中, 我们选择了Flesch-Kincaid易读性测试[16]。这个测试是由Rudolf Flesch和J. Peter Kincaid在1975年为美国海军发明的, 对一个文本应用以下公式1。

$$fkScore = 206.835 - 1.015 * \left( \frac{totalwords}{句子总数} \right) - 84.6 * \left( \frac{totalsyllables}{总字数} \right) \quad (1)$$

结果 (fkScore) 是一个分数, 大多数数值在0到100之间 (尽管低于和高于这个分数都有可能实现)。这个分数显示了文本的阅读难度--分数越低, 文本就越难。有趣的是, 得分116的最简单的句子是 "猫坐在垫子上", 而titin (一种蛋白质) 的化学名称有189819个字符, 由72443个音节组成, 得分是-6128472。评定的文本可以分为表1所示的几组。

纳入Flesch-Kincaid得分需要进一步修改算法。最终的算法如算法3所示。

和以前的方法一样, 得到的词集W从最容易的词到最难的词进行排序。与之前的方法类似, 最终的分数是长度和发生率分数的乘积。在这种情况下, 一个词的长度分数在 "1, 2" 范围内取值, 因此整个分数不会为零。在需要对多词短语进行分类的情况下, 应将短语分成独立的词, 并计算其得分。短语的得分应该被计算为它所包含的所有词的平均得分。



---

**算法3**：第三种方法与Flesch-Kincaid评分法

---

- 1: 阅读一组文本**T**
  - 2: 阅读一组词, 对**W**进行分类
  - 3: 对于**i = 0**到**i = size(T)**做
  - 4:  $fkScore[i] = 206.835 - 1.015 * (\frac{totalwords(T[i])}{句子总数(T[i])}) - 84.6 * (\frac{totalsyllables(T[i])}{总字数(T[i])})$
  - 5: 结束
  - 6: **for i = 0 to i = size(T) do**
  - 7:  $fkScore[i] = \frac{fkScore[i]}{\max(fkScore)}$
  - 8: 结束
  - 9: 选择**W**中最长的字, 并将其长度存储为**lMax**
  - 10: **for i = 0 to i = size(W) do**
  - 11:     **for j = 0 to j = size(T) do**
  - 12:         让**occurrences[i][j]**为文字**T[j]**中**W[i]**字的出现次数。
  - 13:         让部分频率得分**[i][j]**成为**发生率[i][j]\*fkScore[j]**。
  - 14:     **结束**
  - 15:  $frequencyScore[i] = \frac{\sum_{n=0}^{size(T)-1} 部分频率得分[i][n]}{大小(T)}$
  - 16:  $lengthScore[i] = (2 - \frac{length(W[i])}{AAA})$
  - 17:  $s[i] = frequencyScore[i] * lengthScore[i]$
  - 18: **结束**
  - 19: 将**W**中的单词按分数降序排列。
- 

## 4 评价

为了评估的目的, 我们用随机词生成器工具<sup>3</sup>, 创建了一组**20**个词。这些词被展示给**90**人的小组。每次单词都被随机洗牌, 以避免对难度的顺序提出任何建议。每个人都被要求从最容易的单词到最难的单词排序。然后使用以下步骤对结果进行累积和合并。

1. 对于每个参与的人, 获得他们的结果列表
2. 对于每一个列表, 每一个词都要附上分数, 因此, 第一个词(最简单的, 根据参与者的意见)得到**1**分, 最后一个(最难的)得到**20**分。
3. 对于每一个词, 然后从所有列表中计算出平均分数
4. 按计算出的平均分升序排列词语

在结果中, 我们得到了一个有序的单词列表(从最容易到最难), 如表**2**所示。

就像我们假设的那样, 常见的、知名的和短的单词占据了表格的开头。有些结果可能被认为是出乎意料的(如 "whistle"), 但在这些情况下, 除了公式中考虑的因素外, 可能还有其他影响其难度的因素(例如, 一个词的书写方式和发音之间的差异--这是英语语言中常见的东西)。

<sup>3</sup><https://randomwordgenerator.com/>



表2.按人类排序的评估词集

词语	秩序	词语	秩序	词语	秩序	词语	秩序
开始	1	污垢	6	小包	11	全景	16
杯子	2	单位	7	爵士乐	12	哨子	17
艰苦	3	兔子	8	不同意	13	巨大的	18
假的	4	剪发	9	可怕的	14	摇摆不定	19
可爱的	5	滴水不漏	10	观察	15	乏味的	20

表3.按人类排序的评估词集

调查	算法#1	算法#2	算法#3
开始	杯子	杯子	杯子
杯子	开始	开始	<i>单位</i>
艰苦	<i>单位</i>	<i>单位</i>	艰苦
假的	艰苦	艰苦	<i>滴水不漏</i>
可爱的	假的	假的	开始
污垢	<i>巨大的</i>	污垢	污垢
单位	<i>观察</i>	<i>巨大的</i>	<i>摇摆不定</i>
兔子	污垢	<i>观察</i>	<i>假的</i>
剪发	<i>哨子</i>	<i>哨子</i>	兔子
滴水不漏	小包	<i>可爱的</i>	<i>可爱的</i>
包裹	兔子	包裹	爵士乐
爵士乐	全景	兔子	<i>哨子</i>
不同意	可怕的	可怕的	小包
可怕的	<i>可爱的</i>	全景	<i>剪发</i>
观察	不同意	不同意	可怕的
全景	<i>爵士乐</i>	<i>滴水不漏</i>	<i>观察</i>
哨子	乏味的	<i>爵士乐</i>	乏味的
巨大的	<i>滴水不漏</i>	乏味的	<i>不同意</i>
摇摆不定	<i>剪发</i>	<i>剪发</i>	全景
乏味的	摇摆不定	摇摆不定	巨大的

调查结束后，使用所提出的三个版本的解决方案对同一组词进行了分类。作为文本语料库，使用了英语维基百科（由4 838 000页组成）。

我们将结果分为四组，单词数量相等。我们假设一组中的词在难度上是接近的。斜体字的词是被算法分配到其他难度级别的词，而不是根据调查的结果。与调查相比，结果见表3。

我们可以看到，天真的版本正确地分配了9个词，第一次修改了10个词，最后的版本是11个词。分数的平均差异（以单词被转移的层数计算）在第一种情况下是1.45，第二种情况下是1.4，最后的算法是0.9。最终的解决方案，在大多数情况下，在相邻的组之间切换单词。



使用第三种方法得到的结果非常符合我们的目的。我们需要获得一个有序的单词难度列表，以满足问答游戏的需要。这个列表可以在以后的实际游戏过程中由玩家自己进行修正。在这种情况下，将单词的难度上调或下调是完全可以的。为了游戏的目的，我们计划对56000个单词进行分类，并将它们分配到10个等级。因此，与所产生的位置的小差异不太可能导致水平不匹配。

评估期间进行的实验显示了一些未解决的问题。要在算法中建立一个绝对的评分系统是非常困难的。这是因为一个词在所有文本中的平均频率取决于我们使用的文本数量，而一个词的平均长度则与此无关，并稳定在某个水平。这一点试图通过引入频率和长度两部分的权重系数来解决，但要适当地调整它们是非常困难的，特别是当文本语料库包含数百万的文本时。

Flesch-Kincaid评分对普通英语文本效果很好，但有时在语料库中会有一些文本导致非常低的负分，破坏了结果。这种情况发生在一次测试中。一些维基百科页面主要由中国人组成，得分为-2971。这样的文本在我们的计算中是没有用的，并在结果中引入了噪音。经过一些进一步的测试，我们决定排除任何得分低于0的文本，因为这表明要么文本不是英文的（至少是部分的），要么我们处理的是一些不寻常的东西，比如蛋白质的名字。

最后的方法也非常耗费内存。计算4800多个文本中56000个独特的词的出现次数需要大量的时间，并使用大量的内存。最早的计划是计算所有文本中所有独特词汇的出现次数，但这导致了数据库的指数式增长。因此，我们决定只对我们真正感兴趣的词进行计算，这样就可以控制住内存的限制。

## 5 结论和未来工作

对单词难度的评分是一项困难的任務，主要是由于缺乏一个标准的定义，即一个单词的难度是什么意思。然而，观察表明，这里提出的方法与对人类行为的观察是一致的--如果我们经常偶然发现一个词，那么这个词对我们来说就更容易。

所提出的方法使我们能够以一种不会使玩家气馁的方式来排列需要验证的短语集。此外，值得注意的是，这个算法的主要目的是提供一个初步的分类，以后可能会被玩游戏的玩家纠正或由主持人手动调整。

在未来，进一步扩展该算法可能是可行的。我们还应该考虑到该词的发音。拼写和发音之间的差异越大，这个词就越难。



认为。考虑到类似拼字游戏的字母排名，如促进罕见的字母组（一对或三对）或计算独特的字母与单词长度的比例（越是独特的字母，单词就越难），可能会允许独立于单词领域的更好的分类。

## 参考文献

1. Korytkowski, R., Szymanski, J.: WordNet和Wikipedia整合的合作方法。在:第二届高级协作网络、系统和应用国际会议, COLLA, 第23-28页(2012)
2. Szyman´ski, J.: 维基百科类别之间的关系挖掘。In:网络化数字技术, 第248-255页。斯普林格(2010)
3. Szyman´ski, J.: 用于改进信息检索的词语语境分析。在。计算集体智能国际会议。318-325.斯普林格(2012)
4. Szyman´ski, J., Duch, W.: 用于类别可视化的自组织地图。在。第160-167页。斯普林格(2012)
5. Von Ahn, L.: 有目的的游戏。计算机39(6), 92-94 (2006)
6. Von Ahn, L., Dabbish, L.: 设计有目的的游戏。ACM通讯》51(8), 58-67 (2008)
7. Biuro Prasowe Samsung Electronics Polska Sp. z o.o. Prawie po lowa Polak´ow gra codziennie w gry wideo (in Polish)
8. Wightman, D.。众包基于人的计算。在:第六届北欧人机交互会议论文集。扩展边界。pp.551-560.ACM (2010)
9. Boin´ski, T.: 有目的的游戏, 用于映射验证。在。计算机科学与信息系统 (FedCSIS), 2016年联邦会议。405-409.IEEE (2016)
10. Binkley, M.R.: 评估文本难度的新方法。可读性。它的过去、现在和未来 第98-120页(1988)
11. Kauchak, D., Leroy, G., Hogue, A.: 使用解析树自由度测量文本难度。信息科学与技术协会期刊(2017)
12. Crossley, S.A., Greenfield, J., McNamara, D.S.:使用基于认知的指数评估文本的可读性。Tesol Quarterly 42(3), 475-493 (2008)
13. Breland, H.M.: 词汇频率和词汇难度。四个语料库中的计数比较。心理科学》7(2), 96-99 (1996)
14. Carroll, J.B.: 一个替代juilland的词汇频率使用系数的方法。ETS研究报告系列1970(2) (1970)
15. Inc., T.: Twinword API. <https://www.twinword.com/api/language-scoring.php> (2011), [Online: 10.05.2017]
16. Kincaid, J.P., Fishburne Jr, R.P., Rogers, R.L., Chissom, B.S.:为海军士兵推导新的可读性公式(自动可读性指数、雾数和肉眼可读性公式)。技术报告, DTIC文件(1975)

