

使用字符统计为Wordle选择最佳种子词

尼桑萨-德席尔瓦

莫拉图瓦大学计算机科学与工程系

Moratuwa, Sri Lanka

NisansaDdS@cse.mrt.ac.lk

摘要

Wordle，一个猜词游戏在2021年12月上升到全球流行。该游戏的目标是在六次尝试中猜出一个五个字母的英文单词。每一次尝试都会通过变色砖为玩家提供提示，告知玩家某个字符是否是解决方案的一部分，以及在它是解决方案的一部分的情况下，它是否在正确的位置。人们已经做了许多尝试，以找到解决每日*Wordle*的最佳起始词和最佳策略。本研究使用五个字母的单词的字符统计来确定最佳的三个起始词。我们表明，在三个不同的数据集上，我们提出的起始词比目前可用的建议词配置表现得更好：其中，两个来自官方的*Wordle*词库。

关键词：字词、字符统计、文字游戏、优化、解谜

I. 简介

基于网络的文字游戏*Wordle*[1]，自2021年11月推出以来，已经俘获了互联网的心。游戏的玩法相当简单。它给玩家六次机会来猜一个五个字母的单词。在每次尝试中，都会用彩色瓷砖给玩家以提示。该游戏的诱惑力一方面是由于它通过限制每天一道谜题而创造了虚假的稀缺性。在医学等领域，对通过参与这种日常智力活动所开发的技能的价值分析已经开始[2]。《纽约时报》在2022年1月收购了*Wordle*，以利用其数百万的每日用户[3]，

4]。鉴于该游戏的受欢迎程度[3-5]，已经有很多人尝试寻找最佳的第一个词，以快速准确地解决每日的难题。Kandabada[6]的工作建议手动选择四个词，[SPORT, CHEWY, ADMIX, FLUNK]，作为最佳起始词。Sidhu[7]的工作试图从语言学的角度找到最佳起始词，而Horstmeyer[8]的工作则建议通过运行超过100万次的模拟来完成同样的工作。Bram和Cardin[9]提出了两种赢得游戏的策略，这些策略是他们从填字游戏的经验中得出的。有趣的是，也有一些研究是关于以最差的词来开始填词的[10]。Anderson和Meyer[5]使用机器学习来寻找人类解决字谜的最佳策略。Short[4]的工作提出

最快速递减（MRD）算法和最大期望概率（GEP）算法来确定最佳起始词。这两种算法都预测它是TARES。

这项工作的目标是推导出3个最优的涵盖15个不同的字符并按重要性降序排列的*wordle*起始词。本文的其余部分组织如下，第二节简要介绍了*Wordle*，第三节描述了我们的方法，第四节报告了我们的实验和结果。最后，第六节是本文的结论。

II. 背景介绍

该游戏接受12972个词作为可能的猜测解法，而它有2315个秘密词作为实际解法[5]。给予玩家的基于瓦片的提示如下。

- 1) **绿色**。输入的字符在预期的解决方案中，并且在预期的位置。
- 2) **黄色**。输入的字符在预期的解决方案中，但它不在预期的位置。
- 3) **灰色**：输入的字符不在预期方案中。

图1显示了四个字谜解决方案的例子，瓷砖的颜色为玩家进展提供了提示。从图1d和图3b中可以看出，解决方案中有可能出现重复的字符。从图3b中还可以看出，这些重复的字符并没有用颜色代码来提示。第一行提示E是在解决方案中。但并没有说明这个字符会出现多少次。另一点需要注意的是，从图1c中可以看出，尽管*Wordle*是由英国人开发的，并以.co.uk的网地址托管，但却使用了美式拼写。*Wordle*的完全自动解决是一个令人满意的约束问题，与de Silva等人[11]的工作类似。

III. 方法论

我们提出了以下新颖的方法。我们获得一个人工准备的单词列表G，并得出所有的单词列表，A如公式1所示，其中，len(-)函数给出了词的长度，这将删除所有属于长度不超过5的（如：猫、龙）。

$$A = \bigcup_{w \in G} \begin{cases} \{w\} & \text{如果 } \text{len}(w) = 5 \\ \emptyset & \text{否则} \end{cases} \quad (1)$$

S	A	L	E	S	S	P	O	R	T
C	O	V	E	R	A	D	M	I	X
R	I	N	G	S	F	L	U	N	K
R	U	I	N	S	C	H	E	W	Y
D	R	I	N	K	J	A	N	T	Y
					T	A	N	G	Y

(a)猜词 饮酒 (b)猜词 唐吉珂德

C	O	V	E	R	C	O	V	E	R
G	I	V	E	R	P	I	N	G	S
F	A	V	O	R	H	A	T	E	D
					A	B	B	E	Y

(c) 猜测 青睞 一词 (d) 猜测 Abbey 一词

图1：Wordle解决方案的几个例子

接下来，我们初始化字符频率图， F 的如公式2所示，格式为<字符，值>，其中 C 为字母表。

$$\forall_{c \in C} F(c) = 0 \quad (2)$$

然后，我们填充字符频率图， F 如公式3所示，其中 A 是所有单词列表， C 是字母表， c_2 是单词 w 中的一个字符。

$$IF(c) = F(c) + \sum_{w \in A} \sum_{c_2 \in w} 1 \quad \text{if } c_1 = c_2 \quad \text{否则为0} \quad (3)$$

接下来，我们更新字符频率图。 F ，如图所示

在公式4中， A 是所有单词列表， C 是字母表， $len(w)$ 给出单词的字符数（长度）。

w 。这导致 F 登记了字母表中每个字符的全球频率。

$$\forall_{c_1 \in C} IF(c_1) = \frac{F(c_1)}{\sum_{w \in A} len(w)} \quad (4)$$

我们定义唯一的单词列表， W 如公式5所示，其中 A 是所有单词列表， c 是单词 w 中的一个字符。

和 $len(-)$ 函数给出了集合的大小。这个函数确保 W 只包含有5个字的单词。

独特的字符。这就删除了有重复字符的5个字母的单词（例如，感觉）。

$$W = \bigcup_{w \in A} \{w\} \quad \text{如果 } len(w) = 5 \quad \text{否则 } \emptyset \quad (5)$$

我们定义词值图， M 的格式为

$$\forall_{w \in W} M(w) = \sum_{c \in w} 1 \quad (6)$$

接下来，我们定义单词重叠，如公式7所示，其中 w_1 和 w_2 是候选单词， i 是一个字符。因此， I_{w_1, w_2} 将携带布尔值TRUE，即 w_1 和 w_2 之间至少有一个共同的字符，否则携带布尔值FALSE。

$$I_{w_1, w_2} = \exists i \text{ s.t. } i \in w_1 \wedge i \in w_2 \quad (7)$$

接下来我们定义一个贪婪的算法来选择当前的最佳词汇集，如公式8所示，其中， B 是最佳词汇集， B_0 是 B 的第一个元素， M 是格式为<词，值>的词值图（包括但不限于公式6中定义的）。这个过程作为 $B(M)$ 的结果，返回 M 中价值最高的词。

$$B(M) = \begin{cases} B_0 & \text{如果 } B = \emptyset \text{ 或者 } M(w) > M(B_0) \\ B \cup \{w\} & \text{如果 } M(w) = M(B_0) \\ B & \text{否则} \end{cases} \quad (8)$$

我们定义简化的最佳单词列表，如公式9所示，其中 L 是一个单词列表， L_0 是 L 的第一个单词。

$\{L_1, \dots, L_n\}$ 是 L 中除 L_0 以外的单词列表，而 n 是它的作用是，给定一个 L ，它删除 $\{L_1, \dots, L_n\}$ 中与 L_0 有字符重叠的字，以迭代的方式删除，其余的保留

$$S(L) = \begin{cases} L & \text{如果 } n \leq 1 \\ L_0 & \text{如果 } L_0, l \text{ 有字符重叠} \\ L \setminus \{L_1, \dots, L_n\} & \text{否则} \end{cases} \quad (9)$$

我们定义过滤后的字值图， M 的格式为<字，值>，如公式10所示，其中 M 是格式为<字，值>的字值图（包括但不限于公式6中定义的字值）， w_1 是一个给定的过滤词。这个算法确保了 M_{w_1} 包含 M 中<字，值>对的子集，这样，没有一个键与 w 重合的字符 1 。

$$\forall_{w_2 \in keys(M)} M_{w_1}(w_2) = M(w_2), \quad \text{如果 } I_{w_1, w_2} = \text{FALSE} \quad (10)$$

最后，我们在方程11中定义了候选处理，其中 M 是一个字值图，格式为

<字, 值> (包括但不限于公式6中定义的), 函数 $B(-)$ 如公式8中定义, 函数 $S(-)$ 如公式9中定义。
过滤后的字值图, M 如公式10所定义, w 是一个字。

$$p(m) = s(b(m)) \cup_{w \in S(B(M))} P(M', w) \tag{11}$$

IV. 实验

对于人工准备的单词表 G , 我们使用了来自github¹的公开可用的单词表, 其中有超过466k个英语单词。我们注意到, A 只包含2195个单词。另外, 为了利用方程4, 我们计算了 A 中单词的总字符数, 发现它有109760个字符。我们在表一中显示了我们为字符频率图 F 计算的字符频率, 如公式4所示。

表一:计算出的字符频率

角色	频率	角色	频率
a	0.1124	n	0.0546
b	0.0268	o	0.0653
c	0.0330	p	0.0263
d	0.0355	q	0.0015
e	0.0994	r	0.0648
f	0.0147	s	0.0754
g	0.0236	t	0.0491
h	0.0290	u	0.0399
i	0.0661	v	0.0114
j	0.0057	w	0.0135
k	0.0224	x	0.0042
l	0.0556	y	0.0314
m	0.0318	z	0.0069

从表一中我们可以看到一个有趣的现象, 即字符 a 比字符 e 更频繁。除此以外, 所有其他的字符频率都与常识一致。在使用公式5得到 W 之后, 我们观察到还有13672个词。有趣的是, 在5个字母的词域中, 大多数的词似乎都有5个独特的字符, 而不是重复的字符。图2显示了部分词值图, M 的格式为<词, 值>, 如方程6中生成的。请注意, 这甚至包含了专有名词(如abdul), 而当前版本的Wordle并没有这样的解决方案。为了概括起见, 我们保持原样。

当我们执行方程11, 并提取了我们的单词建议。这时, 我们面临一个问题。那就是一些被建议为候选人的词没有被Wordle接受为有效的词。我们观察到, 这是由于我们的词表比Wordle中使用的词表更丰富这一事实。正因为如此, 我们不得不手动放弃一些高排名的词。我们不得不放弃的词是

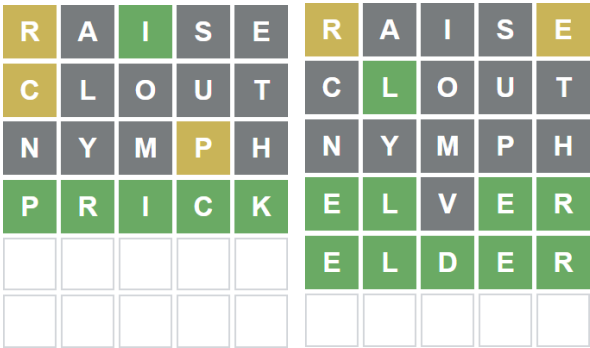
¹<https://github.com/dwyl/english-words>

```
{'abdel': 0.329701166180758,
'abdom': 0.27143768221574344,
'abdon': 0.29451530612244897,
'abdul': 0.2702259475218659,
'abend': 0.32860787172011663,
'abert': 0.3523870262390671,
'abets': 0.36307397959183674,
'abhor': 0.2982325072886297,
'abide': 0.34013301749271135,
'abied': 0.3401330174927114,
'abyed': 0.305493804664723,
'abies': 0.38010204081632654,
'abyes': 0.34546282798833816,
'abihu': 0.27418002915451894,
```

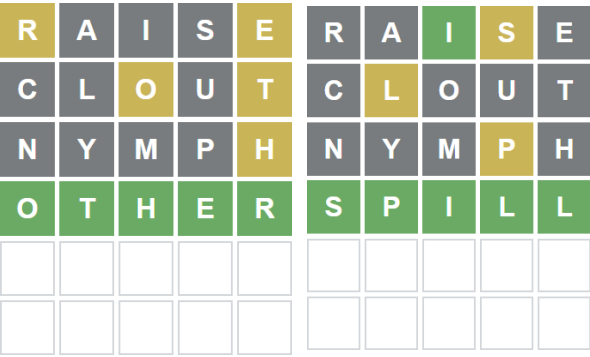
图2:

它们是: Aires, erisa, saire, luton, tould, unold, dunlo, xdmcp, 和白羊座。

被Wordle接受的最高等级的建议是['serai', 'nould']。然而, 由于只有两个词, 因此只涵盖了10个字符, 这不得不被放弃。被wordle接受的排名最高的三个词组是['Aesir', 'donut', 'lymph']。从逻辑上讲, 这三个词是最适合使用的。然而, Aesir并不是我们的阿尔法测试者所使用的一个非常普遍的词。因此, 我们选择了下一个最佳的单词列表['raise', 'clout', 'nymph']。图3显示了两个用所选单词解决的单词的例子。



(a) 猜测Prick这个词 (b) 猜测Elder这个词



(c) 猜测 "其他" 一词 (d) 猜测单词Spill

图3: 使用所选词语 ['提高'、'吹捧'、'仙女'] 的四个Wordle解决方案的例子

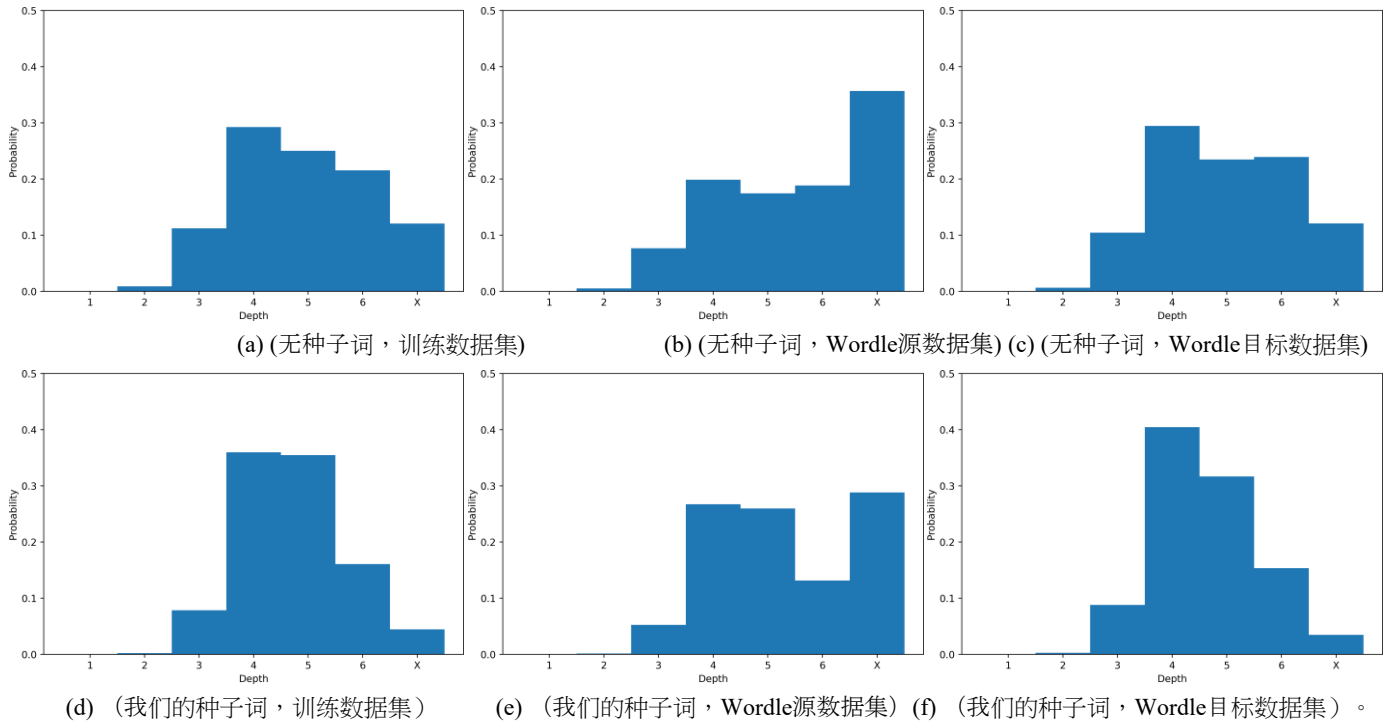


图4：使用不同的数据集以及使用和不使用我们提出的种子词['raise', 'clout', 'nymph']的Wordle仿真结果。深度表示谜题被解开的深度，特殊深度X表示在规定的6次尝试深度内没有找到解决方案的情况。每张图的标题都是按照惯例 (S, D)，其中， S 描述的是没有使用种子词还是使用了我们建议的种子词，而 D 表示的是使用什么数据集来生成该图。

V. 结果

为了进一步评估我们提出的种子词的有效性，我们决定收集Wordle运行数据。然而，鉴于官方Wordle每天只发布一道谜题，这不足以评估我们的种子词并将其与文献中的种子词进行比较。因此，我们决定采取Horstmeyer[8]所提出的模拟路线。为了建立一个模拟器，我们首先重新创建了由Petersen[14]提出的Wordle解题器。然而，后来我们注意到，虽然该实现将黄色标签的字符从相关位置移除，但它随后并没有强制要求上述字符应该在不同的位置被纳入解法。这导致接下来的迭代有更大的搜索空间，阻碍了在预期最大深度内的收敛。在我们的实施中，我们增加了这种强制力。图5显示了消融研究。

然后，我们用Wordle反馈系统的响应模型替换了人类输入的部分。最后，我们修改了求解器的算法以接受外部种子词。此外，从这一点开始，我们在第四节中用来生成种子词的词表将被称为训练数据集。除了这个数据集，我们还从官方的Wordle应用程序[1]中获得了所有可能的未来和过去的谜题目标词的列表，以及所有被接受的源词的列表。这些数据分别被命名为Wordle目标数据集和Wordle源数据集。如前所述

在第四节中，训练数据集包含21952个词。新的Wordle目标数据集和Wordle源数据集，分别包含2309个和10638个词。图4显示了所有三个数据集的模拟结果，其中有我们的种子词和没有我们的种子词。从这些图表中可以看出，Wordle源数据集是最难解决的，而且在所有情况下（包括Wordle源数据集），有我们的种子词的模型有更高的解题机会（即在深度1到6内得到一个解决方案）。图中还显示，相对而言，我们的种子词比没有种子的配置在较低的深度上得到了解决。

接下来，我们想把我们的种子词与先前工作[6-8, 10, 15-17]提出的种子词进行比较。为了达到这个目的，我们总共进行了180万次模拟（超过了Horstmeyer[8]所进行的100万次模拟）。我们在表二、三和四中报告了这些实验的结果。在每个表中，我们都报告了图4中讨论的和Petersen[14]提出的无种子词的基线，以及使用我们提出的一个、两个或全部三个种子词的结果。鉴于Neil[17]和Groux[16]提出了与我们相同的起始种子词"提高"，后一项分析很重要。

为了更好地模拟人类解决Wordle的方式，我们还引入了一个例外，即使用包含一个以上单词的种子词组的配置。在模拟中使用了一个或多个种子词的情况下

表二：训练数据集的模拟结果

种子 词条	计数							累计百分比					
	1	2	3	4	5	6	X	1	2	3	4	5	6
没有种子[14]	1	196	2464	6418	5492	4731	2650	0.00	0.89	12.11	41.35	66.37	87.92
运动型耐力赛的Admix flunk	1	51	1179	3880	9750	5957	1134	0.00	0.23	5.60	23.27	67.69	94.83
[6].													
飙升 [7, 15]	0	190	2433	6496	5452	4668	2713	0.00	0.87	11.95	41.54	66.38	87.64
精通夹子格子 [8]	1	43	498	1574	6560	7869	5407	0.00	0.20	2.47	9.64	39.52	75.37
切片[8]	1	181	2449	6853	5663	4557	2248	0.00	0.82	11.98	43.20	69.00	89.76
试过[8]	1	182	2455	6949	5871	4422	2072	0.00	0.83	12.01	43.67	70.41	90.55
起重机 [8]	1	182	2477	6696	5654	4448	2494	0.00	0.83	12.11	42.61	68.37	88.63
稗子[10]	1	209	2598	6640	5466	4494	2544	0.00	0.95	12.78	43.03	67.93	88.40
拉雷斯[10]	1	201	2539	6675	5429	4519	2588	0.00	0.92	12.49	42.90	67.63	88.22
声带[10]	1	198	2511	6732	5417	4492	2601	0.00	0.90	12.34	43.01	67.69	88.15
率[10]。	1	202	2549	6609	5516	4510	2565	0.00	0.92	12.53	42.64	67.77	88.31
cares [10]	1	208	2442	6631	5569	4547	2554	0.00	0.95	12.07	42.28	67.65	88.36
石板[16]	1	196	2538	6760	5578	4486	2393	0.00	0.89	12.45	43.24	68.65	89.09
雷亚尔[16]	0	195	2456	6367	5422	4688	2824	0.00	0.89	12.08	41.08	65.78	87.14
出现[16]	1	178	2418	6391	5600	4649	2715	0.00	0.81	11.82	40.93	66.44	87.62
提高 [16, 17]	1	194	2414	6361	5582	4727	2673	0.00	0.88	11.88	40.86	66.29	87.82
提高 影响力	1	48	2805	7533	5797	4085	1683	0.00	0.22	13.00	47.32	73.73	92.34
提高 克劳特蛱蝶	1	48	1721	7893	7786	3526	977	0.00	0.22	8.06	44.02	79.49	95.55

表三：Wordle源数据集的模拟结果

种子 词条	计数							累计百分比					
	1	2	3	4	5	6	X	1	2	3	4	5	6
没有种子[14]	0	55	816	2114	1856	2004	3793	0.00	0.52	8.19	28.06	45.51	64.35
运动型耐力赛的Admix flunk [6]。	0	29	428	1449	3557	2180	2995	0.00	0.27	4.29	17.91	51.35	71.84
飙升 [7, 15]	1	58	818	2122	1808	1987	3844	0.01	0.56	8.25	28.20	45.20	63.88
精通夹子格子 [8]	0	15	161	646	2225	2814	4777	0.00	0.14	1.65	7.72	28.64	55.09
切片[8]	0	48	757	2247	1939	1984	3663	0.00	0.45	7.57	28.69	46.92	65.57
试过[8]	0	82	861	2277	2000	1881	3537	0.00	0.77	8.86	30.26	49.06	66.74
起重机 [8]	0	59	874	2255	1892	1837	3721	0.00	0.55	8.77	29.97	47.76	65.03
稗子 [10]	1	100	842	2182	1860	1836	3817	0.01	0.95	8.87	29.38	46.86	64.12
拉雷斯[10]	1	85	832	2193	1818	1887	3822	0.01	0.81	8.63	29.24	46.33	64.07
声带[10]	1	81	809	2230	1798	1902	3817	0.01	0.77	8.37	29.33	46.23	64.11
率[10]。	1	88	867	2144	1855	1857	3826	0.01	0.84	8.99	29.14	46.58	64.04
cares [10]	1	78	818	2159	1866	1909	3807	0.01	0.74	8.43	28.73	46.27	64.22
石板[16]	0	60	829	2244	1939	1866	3700	0.00	0.56	8.35	29.44	47.67	65.21
雷亚尔[16]	1	68	823	2060	1808	1953	3925	0.01	0.65	8.39	27.75	44.75	63.11
出现[16]	0	53	779	2055	1893	1994	3864	0.00	0.50	7.82	27.14	44.93	63.67
提高 [16, 17]	0	59	791	2074	1852	2020	3842	0.00	0.55	7.99	27.49	44.90	63.89
提高 影响力	0	15	932	2525	1992	1781	3393	0.00	0.14	8.90	32.64	51.37	68.11
提高 克劳特蛱蝶	0	15	557	2841	2762	1397	3066	0.00	0.14	5.38	32.09	58.05	71.18

表四：Wordle目标数据集的模拟结果

种子 词条	计数							累计百分比					
	1	2	3	4	5	6	X	1	2	3	4	5	6
没有种子[14]	0	15	241	680	542	552	279	0.00	0.65	11.09	40.54	64.01	87.92
运动型耐力赛的Admix flunk [6]。	1	7	180	431	1053	523	114	0.04	0.34	8.14	26.81	72.41	95.06
飙升 [7, 15]	0	18	241	663	557	538	292	0.00	0.78	11.22	39.93	64.05	87.35
精通夹子格子 [8]	1	8	80	187	719	764	550	0.04	0.39	3.85	11.95	43.09	76.18
切片[8]	1	17	254	724	557	525	231	0.04	0.78	11.78	43.14	67.26	90.00
试过[8]	1	22	233	719	610	483	241	0.04	0.99	11.08	42.22	68.64	89.56
起重机 [8]	1	30	267	723	537	493	258	0.04	1.34	12.90	44.21	67.47	88.82
稗子 [10]	0	6	242	684	570	550	257	0.00	0.26	10.74	40.36	65.05	88.87
拉雷斯[10]	0	10	222	687	559	554	277	0.00	0.43	10.04	39.79	64.00	87.99
声带[10]	0	12	222	694	577	530	274	0.00	0.52	10.13	40.19	65.18	88.13
率[10]。	0	11	232	673	580	548	265	0.00	0.48	10.53	39.68	64.80	88.53
cares [10]	0	7	262	671	575	543	251	0.00	0.30	11.65	40.71	65.61	89.13
石板[16]	1	18	273	675	567	504	271	0.04	0.82	12.64	41.87	66.43	88.26
雷亚尔[16]	0	19	224	642	595	536	293	0.00	0.82	10.52	38.32	64.09	87.30
出现[16]	1	18	217	685	583	525	280	0.04	0.82	10.22	39.89	65.14	87.88
提高 [16, 17]	1	16	245	677	552	556	262	0.04	0.73	11.34	40.66	64.57	88.65
提高 影响力	1	6	342	767	575	455	163	0.04	0.30	15.11	48.33	73.23	92.94
提高 克劳特蛱蝶	1	6	203	934	731	354	80	0.04	0.30	9.09	49.54	81.20	96.53

词，并且可能的解决方案的数量已经小于剩余的尝试数量，算法将从这一点上恢复到最佳的建议词，而不是尝试剩余的种子词。鉴于每组种子词的最大数量是4个[6]和

至少要输入一次尝试才能开始这个过程，这种交替的特征可能只在第2-

4位发挥作用。此外，据观察，在其模拟的情况下，他们的仿真在早期阶段发现了四个正确的字符，它就会陷入一个高原期，在多个同等价值的答案中盲目猜测，这可能会导致失败。例如，当目标词是wreak时，模拟在深度4时以-

reak告终。但随后它必须在几乎同等价值的freak、break和wreak之间进行选择，但只剩下2个机会。为了减少这种类型的失败，我们引入了一种启发式方法，即模拟首先检查它是否已经获得了四个正确的字符。接下来，它检查剩余的可能的猜测的数量是否大于剩余的深度。如果是这样，模拟就会从剩余的可能的猜测中获得独特的字符，并尝试用另一个词来明确地消除疑问。在上面的例子中，它将提取字符f、b和w。然后它将尝试单词befit作为下一个猜测。这将确认f或b是正确的选择，或者消除它们。在这个例子中，唯一剩下的选择是w。但如果不是这样的话，这可以迭代以消除更多的字符。但是考虑到深度有限，我们发现单轮是我们能够为这个启发式方法牺牲的全部。在图5所示的消融研究中，可以看到这种补充的有效性。

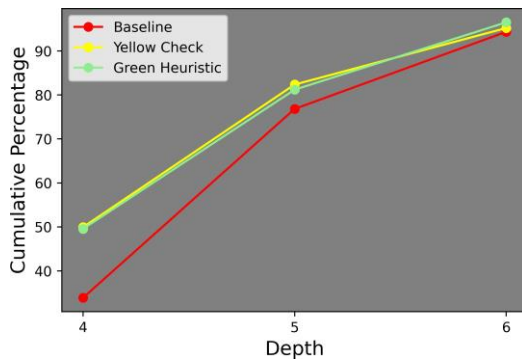


图5：使用最佳种子词['raise', 'clout', 'nymph']对Wordle目标数据集进行的消融研究。我们只显示了以下的累积百分比

深度4-6，以便能够看到微小的差异。红色基线显示的是未经修改的求解器的数值[14]。

在使用我们所有三个种子词的配置中，显示了两个数据集的最低失败次数（重新fer的X列）。这包括Wordle目标数据集，它代表了玩家在可预见的未来在游戏中可能遇到的词语。此外，累积百分比列显示了我们的种子词是如何以可靠的方式带来解决Worldles的能力的。这

最终，我们的三个单词配置在深度为4的Wordle目标数据集上取得了最高的累积百分比，并保持领先，直到指定的最大深度为6。

VI. 总结

这项工作的目标是得出3个最佳的词组，涵盖15个不同的角色，并按重要性降序排列，作为词组的起点。我们通过发现['raise', 'clout', 'nymph']这些词成功实现了这个目标。我们的第一个种子词raise与Groux[16]和Neil[17]所建议的一样。我们提出了一个三种子词的策略，并表明我们的种子词比现有的单词方法以及多词方法[6, 8]（其中一些方法有4个种子词）具有更优越的性能。我们提出的种子词集在所有测试的数据集中，在所有被检查的方法中获得了最高的分配深度内的累积百分比。特别要注意的是，在表四中，我们的词始终优于其他所有的方法，这些方法包含了已经和将要被Wordle作为日常目标词的词。因此，可以得出结论，本研究中提出的种子词，是Wordle尽早获得答案的最佳起始策略。

参考文献

- [1] J.Wardle, "Wordle - A daily word game," https://bit.ly/Wordle_, 2022, [Online; accessed 07-February-2022].
- [2] P.Lok, "在玩文字游戏时反思医学院," *BMJ*, 第376卷, 2022.
- [3] M.Tracy, "纽约时报购买Wordle," <https://nyti.ms/31xaCAB>, 2022, [在线; 2022年3月1日访问].
- [4] M.B. Short, "Winning wordle wisely," *arXiv preprint arXiv:2202.02148*, 2022.
- [5] B.J. Anderson and J. G. Meyer, "Finding the optimal human strategy for wordle using maximum correct letter probabilities and reinforcement learning," *arXiv preprint arXiv:2202.00557*, 2022.
- [6] T.Kandabada, "A Wordle Hack," <https://bit.ly/3HFcp5Y>, 2022, [Online; accessed 09-February-2022].
- [7] D.Sidhu, "Wordle--根据一位语言研究者的说法，开始游戏的最佳词汇", <https://bit.ly/3GrLxoQ>, 2022, [在线; 2022年2月7日访问].
- [8] D.Horstmeyer, "想掌握Wordle？这是你第一次猜测的最佳策略," <https://bit.ly/3Gynfta>, 2022年, [在线; 访问时间: 07-February-2022].
- [9] U.Bram and N. Cardin, "赢得Wordle的两种最佳方式", <https://www://bit.ly/3gLq2oJ>, 2022, [在线; 2022年2月7日访问].
- [10] M.Butterfield, "科学确定了最糟糕的Wordle起始词", <https://bit.ly/3J9FOWj>, 2022, [在线; 2022年2月7日访问].
- [11] N.D. de Silva, S. M. Weerawarana, and A. S. Perera, "Enabling effective synoptic assessment via algorithmic constitution of review panels," in *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALe)*. IEEE, 2013年, 第776-781页。
- [12] B.Keating, "英语中字母的频率", <https://bit.ly/3ow5Pr5>, [在线; 2022年2月7日访问].
- [13] T.Wall和L. Smithline, "英语字母频率（基于40,000个单词的样本）", <https://bit.ly/3sgRgIQ>, [在线; 访问时间: 07-February-2022].
- [14] M.Petersen, "用基本的Python解决Wordle难题", <https://bit.ly/3M7Om2w>, 2022, [在线; 2022年2月28日访问].
- [15] B.Fan, "WORDLE中的最佳起始词", <https://bit.ly/3K6Hlx7>, 2022, [在线; 2022年2月28日访问].
- [16] C.Groux, "根据科学，20个最好的Wordle起始词," <https://bit.ly/3owlqH8>, 2022, [在线; 2022年2月7日访问].
- [17] T.尼尔, "破坏乐趣：一个Wordle自动解算器", <https://bit.ly/34gRmZj>, 2022, [在线; 2022年2月7日访问].