

在应用程序中打开 ↗

报名 登入

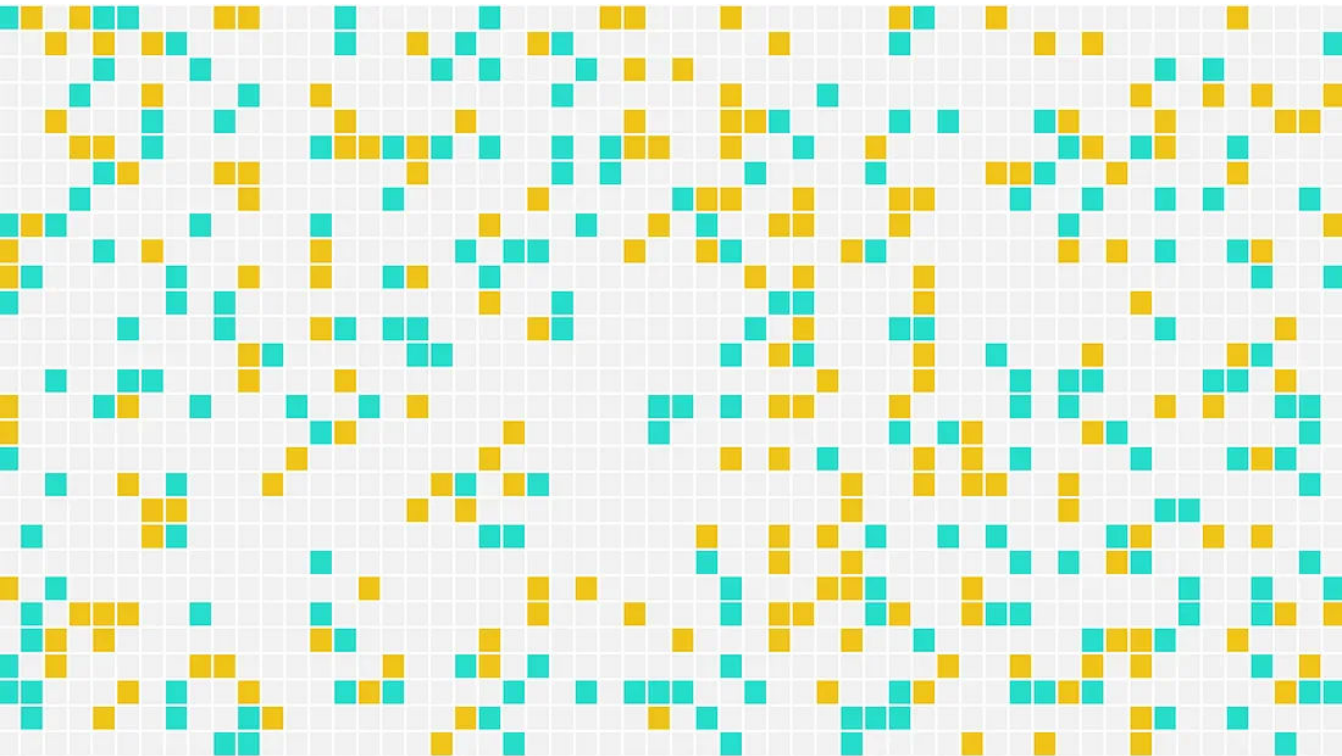


出版于 迈向数据科学



克里斯·周 跟随

2022 年 2 月 10 日 · 12 分钟阅读 · 听



图片由作者提供。

单词

Wordle 中的加载词

为什么“最好的”Wordle 种子词并不是真正最好的

在过去的几周里，我注意到越来越多的绿色、黄色和黑色/白色网格发布在 Facebook 上，这就是我发现 Wordle 的时候。我着迷了——与其说是以人性化的方式玩游戏，不如说是开发一个系统来尝试以最佳方式玩游戏。当我阅读其他作者的现有作品时，我发现压倒性地强调了起始词/种子词。然而，关于最佳使用的建议相互矛盾。优化游戏显然比种子词更重要！

75

2个

这篇文章的主要贡献是通过展示 Wordle 策略的其他组成部分影响这些是什么来解释为什么我们应该对“最佳”Wordle 种子词持保留态度。

Wordle 上的单词

关于这个主题已经写了很多文章。大多数作者使用模拟来找到最佳的起始词/种子词，同时固定其他参数（稍后会详细介绍）。一些推荐的种子词基于最终游戏结果，其他一些基于计算预期信息增益的算法。下表总结了每位作者的一般方法和建议。

S/N	Source	Ranking Algorithm	Recommended Seed Word	Average No. of Steps	Success Rate
1	Barry Smyth [2]	Minimum set covers, coverage, entropy, and letter frequencies	tales	3.66	>95%
2			Two words: cones-trial	3.68	96%
3			Three words: hates-round-climb	3.68	96%
4	Tyler Glaiel [3]	Expected remaining candidates	roate	3.494	100%
5			raise	3.495	Not Provided
6		Expected green / yellow / grey tile scores	soare	3.69	Not Provided
7	Tom Neill [4]	Expected remaining candidates	roate	Not Provided	Not Provided
8	Sejal Dua [5]	Average green / yellow / grey tile scores	soare , stare , roate , raile , arose	N.A.	N.A.
9	Behrouz Bakhtiari [6]	Letter frequencies	aries	N.A.	N.A.
10	John Stechschulte [7]	Information entropy for expected green / yellow scores	tares , lares , rales , rates , nares , tales , tores , reais , dares , arles , lores	N.A.	N.A.
11	Mark M Liu [8]	Information entropy for expected green / yellow scores	tares	N.A.	N.A.

关于 Wordle 的文章。图片由作者提供。

Wordle 是一个相对较新的游戏，覆盖整个解决方案空间需要大量的努力和计算能力。因此，还有许多想法有待探索，有许多问题需要解决。第一个问题是，正如更全面的研究（主要是Smyth）所暗示的那样，最佳游戏比种子词要复杂一些。第二个问题是“最佳”的定义各不相同，并非所有研究都包含足够的指标来衡量策略的绩效。

执行摘要

这篇文章试图解决上述问题。此处列出了这篇文章的目标和主要见解。

首先，我们展示了“最佳”种子词取决于 Wordle 策略的其他组成部分：

- Wordle 策略的其他组成部分是排名算法和在求解和信息收集之间确定优先级的决策规则。
- 排名算法严重影响了 Wordle 的性能。
- 所选择的指标还决定了什么是最好的意思，因此，什么词是最好的。

其次，我们引入了几个指标来衡量性能并定义“最佳”是什么。这些曾经是：

- 达成解决方案所采取的平均步骤数
- 求解成功率
- 在 3 步或更短时间内解决的挑战比例

游戏

对于外行来说，Wordle 是 5 个字母单词的策划者，加上社交媒体上的一些谦虚吹嘘。游戏的目的是在六次尝试中猜出一个未公开的单词。在每次猜测时，Wordle 都会告诉您每个字母是否：

- 在正确的位置（绿色）
- 是在字里，但是点错了（黄色）
- 根本不在单词中（灰色）

这里的所有都是它的！这听起来很简单，但游戏并不容易，因为存在大量的可能性。在 Wordle 中，有 2,315 个可能的解决方案词，以及另外 10,657 个被接受为猜测的词（“支持词”）。因此，我们的机器人的目标是利用完整的 12,972 个候选词集，将 2,315 个解决方案词的集合减少到六次尝试中的一个。

注：完整词组可从网站主脚本中检索。使用浏览器的开发人员控制台访问它。

字策略

就像命运之轮一样，在 Wordle 中，我们在解决问题（猜测一个我们认为是解决方案的单词）和收集信息（使用单词来梳理出解决方案中可能包含的字母）之间取得平

衡，以绿色、黄色和灰色表示瓷砖。人类可能会使用以下策略：

1. 第一轮：收集尽可能多的信息。我们此时没有信息，所以我们选择一些统计上最优的种子词。第一次猜测越好，我们可能收集到的信息就越多。
2. 第 2 轮：使用第 1 轮的反馈收集尽可能多的信息。虽然我们可以通过分两步解决游戏来获得巨大的街头信誉，但这非常困难。因此，我们在第 2 轮中能做的最好的事情就是使用与种子词具有完全不同字母的词来收集更多信息。
3. 第 3 轮：取决于！如果我们获得了足够的信息，我们就可以解决问题。否则，最好谨慎行事，选择另一个词来获取更多线索。
4. 第四轮：同样，这取决于。我们的做法与第 3 轮相同。我们可以通过优先解决信息收集来更加积极。
5. 第 5 轮：再次，这取决于。我们做与第 3 轮和第 4 轮相同的操作。到现在为止，我们应该已经缩小了足以解决问题的选项。
6. 第 6 轮：100% 解决。到第 6 轮我们仍然有几个可行的解决方案是完全有可能的。如果仍然不清楚解决方案是什么，请大胆猜测！我们有什么损失？

我们可以看到策略不仅仅是种子词。它还包括 (1) 优先解决与收集信息的决策规则，以及 (2) 选择单词的方法。

注意：如果我们在游戏开始前应用排名算法对整个候选集进行排名，我们实际上可以完全删除作为策略组成部分的种子词。事实上，这就是上面来源 8-11 中的作者所做的。

模拟Wordle

概述

My Wordle bot 遵循广泛的策略并实施该策略的其他两个组成部分：决策规则和用于选择单词的排名算法。

机器人从 (1) 包含所有 12,972 个接受词的候选集，以及 (2) 包含所有 2,315 个解决方案词的解决方案集开始。它会反复测量 (1) 和 (2)，并在每场比赛中更新它们。机器人一次移动一步，在每一步/每一轮都做同样的事情：

1. 将剩余的候选人和解决方案放入排名算法中，以计算每个候选人的分数。
2. 按分数对剩余候选人进行排序。

3. 提交得分最高的候选人作为该步骤的猜测。
4. 使用反馈来 (a) 筛选候选人和 (b) 筛选潜在的解决方案。我们还消除了已经猜到的候选者，以及包含不再出现在剩余解决方案集中的字母的候选者，即它们没有进一步过滤候选者的价值。
5. 从步骤 1 开始重复，直到步骤 4 的反馈为 GGGGG 。

我开发了一个 `Wordle` 类来促进模拟或其他游戏。由于这不是本文的重点，因此我将跳过其实施细节。[您可以在我的GitHub 存储库中访问完整代码](#)。我提到这个课程只是为了简要说明我是如何运行模拟的。

```
def play_game(input_word, solution):

    game = Wordle(wordle, wordle_answers, solution=solution,
        verbose=False)

    而不是 game.solved:
        if game.step == 0:
            game.guess(input_word)
        else:
            game.guess(
game.optimisations[method.lower()].word.iloc[0])
            game.optimise(method='expected_gyx', n_jobs=-2)

    返回 game.records()
```

在决定所有步骤之前，机器人不会使用蛮力枚举所有游戏结果。这在计算上太密集了。

排名算法

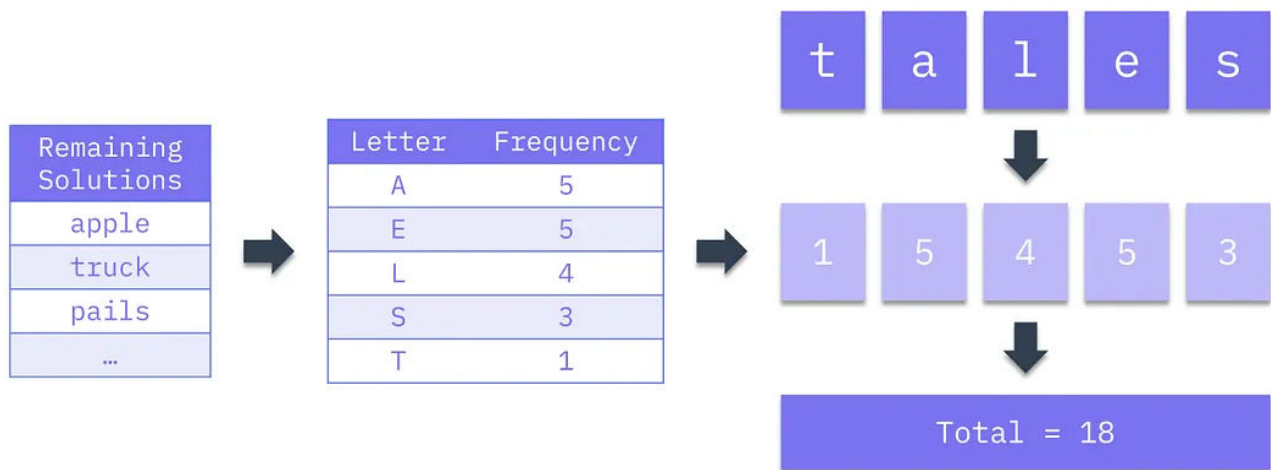
排名算法可以说是策略中最关键的组成部分，因为它决定了可用的猜测。决策规则只在算法提供的可用猜测中做出决定，种子词可以被认为是排名算法的产物，在游戏开始前计算。

My Wordle bot 有几个内置选项：(1) 字母频率，(2) 预期的绿色、黄色和灰色图块，以及 (3) 预期的最大剩余候选数。每个算法计算所有剩余候选者相对于剩余解决方案的分数。

字母频率

该算法根据组成字母的流行程度对单词进行排名：

1. 计算所有剩余解决方案的字母频率
2. 创建一个字母查找表来计数
3. 通过计算该候选词中字母的频率得分总和来为每个剩余的候选者打分
4. 选择得分最高的候选人



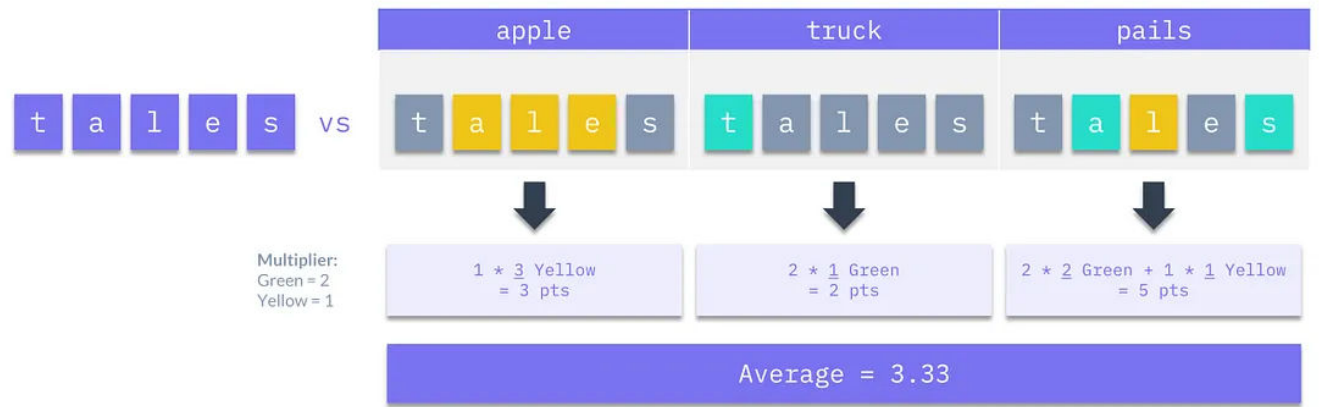
计算字母频率分数。图片由作者提供

预期的绿色/黄色/灰色瓷砖分数

该算法根据返回的绿色瓷砖和黄色瓷砖的数量，根据所有剩余解决方案的平均值，根据获得的预期信息对单词进行排名。x 为简单起见，我将此称为 **GYX** 分数，并且因为我在课堂反馈中编码灰色 () 的方式 Wordle。对于每个剩余的候选人：

1. 针对每个剩余解决方案执行以下操作：
 - 计算针对该解决方案的反馈
 - 计算 $\text{GYX Score} = 2 * \text{No. of Greens} + \text{No. of Yellows}$
2. 合并 **GYX** 分数列表
3. 取所有分数的平均值，为该候选人生成一个分数

最后，选择得分最高的候选人。



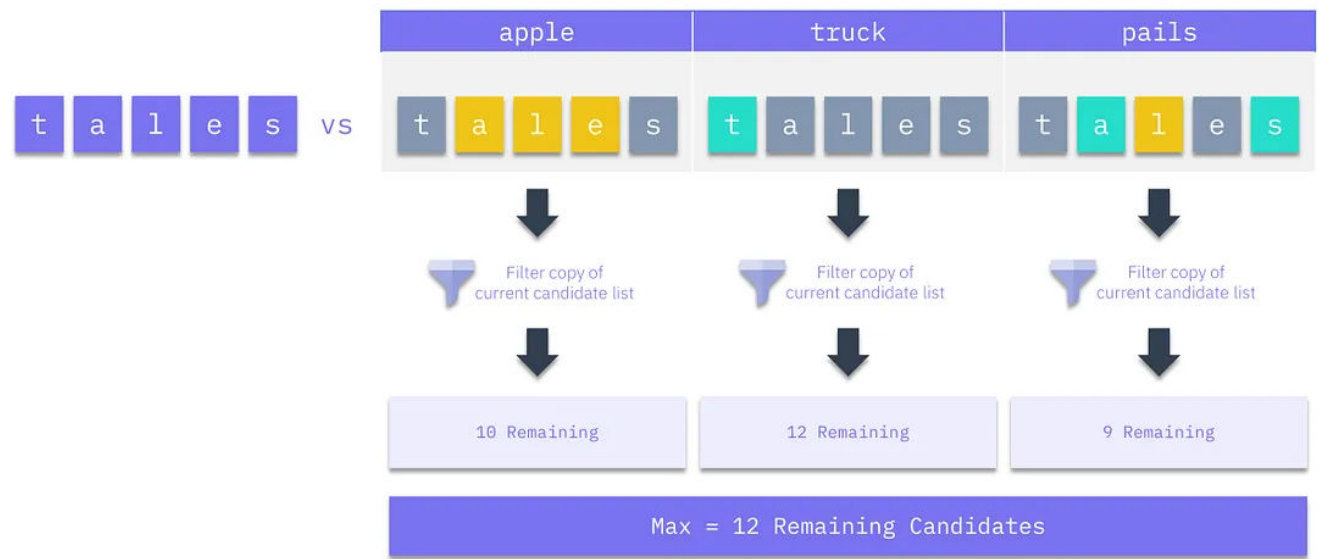
计算预期的 GYX 分数。图片由作者提供。

剩余候选者的最大数量

该算法根据在所有剩余解决方案中最坏的情况下，如果猜测他们将消除/留下多少可能性，对候选者进行排名。这个想法是选择最能减少候选集的词。对于每个剩余的候选人：

- 1. 针对每个剩余的解决方案执行以下操作：
 - 计算针对该解决方案的反馈
 - 使用反馈来过滤剩余候选集（的副本）
 - 计算剩余候选数的结果
- 2. 合并计数列表
- 3. 取所有计数的最大值

最后，选择分数最低的候选人，因为它在最坏的情况下留下的可能性最少。



计算剩余候选人的最大数量。图片由作者提供。

决策规则

基准决策规则是，如果剩余的解决方案不超过 20 个，我们只会猜测解决方案单词。这符合我们的策略，在我们集中解决方案时，我们优先考虑解决问题而不是收集信息。

测试的唯一其他可选决策规则是纯粹根据维基百科文章中单词频率衡量的流行度来选择单词（来源：[Lexipedia](#) [9]）。该规则在只剩下 10 个解决方案时生效，并被置于基线规则之上。我任意选择了 10 个剩余解决方案的阈值，但我确信有更好的方法来选择这个数字。在此规则下，机器人会猜测剩余解决方案中最流行的词。

仿真配置

我将上一节（见下文）的各种来源推荐的每个词与全套 2,315 个解决方案词进行了 5 次对比——每个排名算法/决策规则配置一个。

种子词：

1. arles 10. rates
2. arose 11. reais
3. dares 12. roate
4. lares 13. soare
5. lores 14. stare 6.
- nares 15. tales
7. raile 16. tares
8. raise 17. tores
9. 罗音

排名算法和决策规则：

Configuration	Ranking Algorithm	Decision Rule
1	Letter Frequencies (lf)	Baseline only
2	Letter Frequencies (lf)	Baseline + Popularity
3	GYX Scores (gyx)	Baseline only
4	GYX Scores (gyx)	Baseline + Popularity
5	Max Remaining Candidates (ncands)	Baseline only

测试了排名算法和决策规则配置。图片由作者提供。

总的来说，总共有**196,775** 个Wordle 游戏，总共有 17 个“最佳”单词和 5 个策略以及 2,315 个解决方案单词。

从这里开始，为简单起见，我将排名算法和决策规则的组合称为排名算法/算法，因为排名算法主要负责游戏的进展。

指标

最后，在我们讨论模拟结果之前，这些是建议的指标，以允许其他作者与测试的策略进行比较：

1. 达成解决方案所采取的平均步骤数
2. 求解成功率
3. 在 3 步或更短时间内解决的挑战比例

这些指标共同告诉我们（1）该策略在步骤方面的整体效果如何，（2）它可以解决 2,315 个解决方案单词的比例，以及（3）机器人在几个步骤中解决挑战的效果如何尽可能（即街头信誉）。

结果

总体结果

总体而言，结果表明“最佳”种子词取决于 (1) 其他策略组件——尤其是排名算法——以及 (2) 使用的性能指标。以下是相应策略设置和指标的排名第一的种子词：

Ranking Algorithm	Mean No. of Steps	Success Rate	% Solved Within 3 Steps
Max Remaining Candidates (ncands)	<div>tales</div> - 3.6017	<div>tares</div> - 99.78%	<div>raile</div> - 48.12%
Letter Frequencies (lf)	<div>stare</div> - 3.7287	<div>tales</div> - 99.44%	<div>arose</div> - 42.98%
Letter Frequencies + Popularity (lf-pop)	<div>tores</div> - 3.7702	<div>tores</div> - 99.22%	<div>arose</div> - 42.33%
GYX Scores (gyx)	<div>stare</div> - 3.8320	<div>tales</div> - 99.09%	<div>roate</div> - 38.57%
GYX Scores + Popularity (gyx-pop)	<div>tales</div> - 3.8898	<div>tales</div> - 98.79%	<div>roate</div> - 37.93%

通过排名算法和指标得出的总体结果。图片由作者提供。

下面，我展示了各种排名算法和指标中 17 个“最佳”种子词中每一个的得分箱线图。我还强调了几个种子词的分数和进展，这些种子词在排名算法中一直名列前茅。

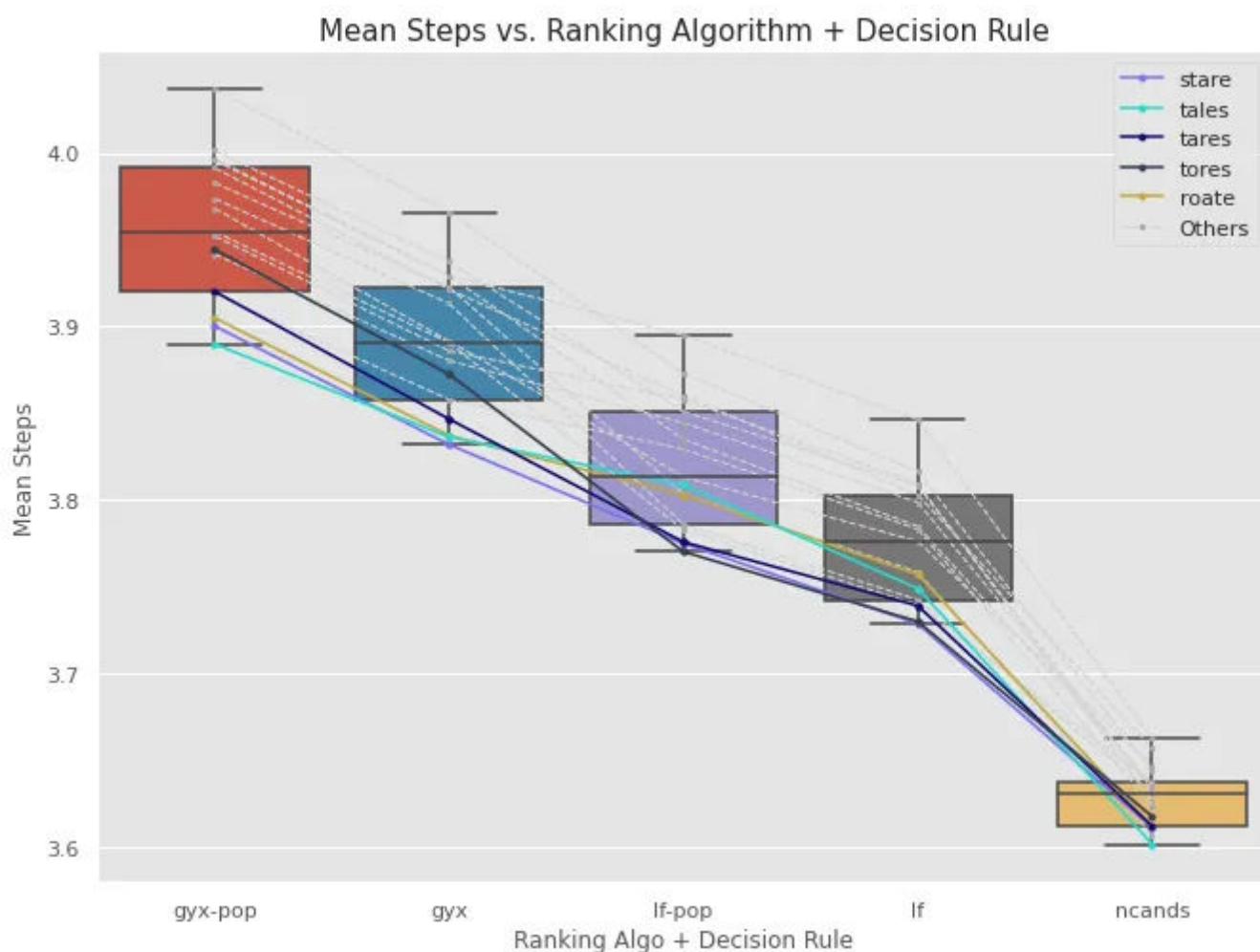
首先，我们注意到对性能影响最大的是排名算法。我们看到三个指标中每个指标的种子词性能分布存在明显差异。特别是，Max Remaining Candidates 排名算法比其他

算法表现得更好，以至于表现最差的种子词比所有其他策略中最好的种子词表现更好。

其次，我们看到当使用不同的排名算法时，“最佳”种子词的排名发生了变化。这对一些人来说比其他人更明显。例如，`tores` 在 GYX 分数算法的平均步数中排名 6-7，在字母频率算法中排名上升到 1-2，然后在最大候选数算法中下降到第 6。

. 这让我们有理由相信，使用不同的算法可能会产生不同的“最佳”种子词。

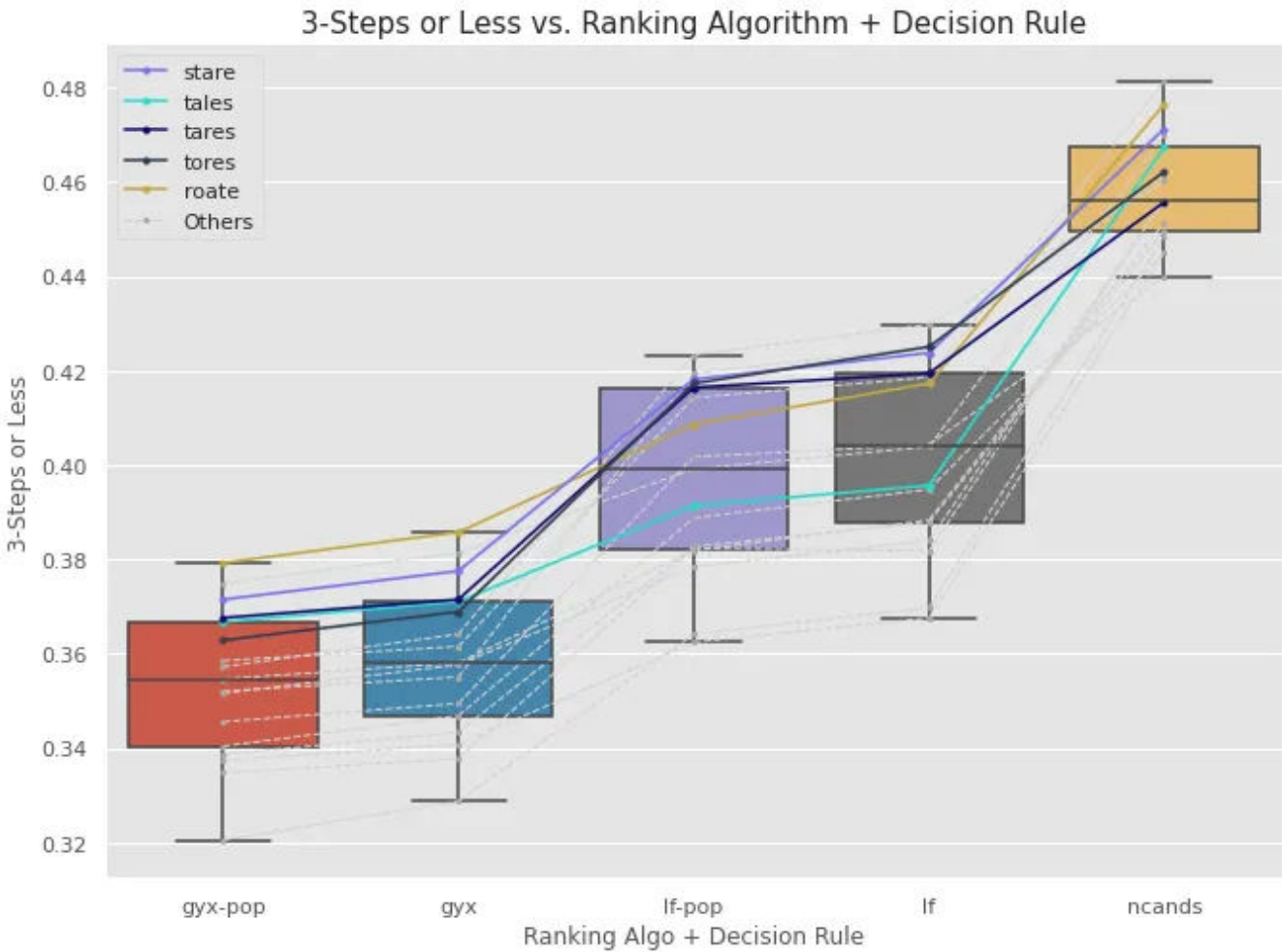
第三，“最佳”种子词并非对所有指标都是最佳的。事实上，每个指标都有不同的最优策略。在各种排名算法中表现相对较好的种子词是：`stare`、、、和。这些在下面的图表中以各自的颜色绘制。`tales` `tares` `tores` `roate`



图片由作者提供。



图片由作者提供。



图片由作者提供。

“最佳”策略

接下来，我们放大每个指标的前 5 个策略。所有这些都涉及最大剩余候选人排名算法和基线决策规则。结果显示了不同的种子词如何更好地实现不同的目标。

基于达到解决方案的平均步骤数的最佳策略是：

Rank	Seed Word	Ranking Algo	Decision Rule	Mean No. of Steps
1	tales	Max Remaining Candidates	Baseline	3.6017
2	raile	Max Remaining Candidates	Baseline	3.6069
3	stare	Max Remaining Candidates	Baseline	3.6112
4	roate	Max Remaining Candidates	Baseline	3.6117
5	tares	Max Remaining Candidates	Baseline	3.6121

图片由作者提供。

解决方案成功率最高的策略非常接近。排名 1 到 2 和 2 到 3 之间的差异为 0.0432%，这意味着 2,315 场比赛中只有 1 场。

Rank	Seed Word	Ranking Algo	Decision Rule	Success Rate	Lead Over Next Rank
1	tares	Max Remaining Candidates	Baseline	99.78%	1 game
2	tales	Max Remaining Candidates	Baseline	99.74%	1 game
3	tores	Max Remaining Candidates	Baseline	99.70%	-
3	arles	Max Remaining Candidates	Baseline	99.70%	-
3	rales	Max Remaining Candidates	Baseline	99.70%	1 game

图片由作者提供。

在 3 步或更少的时间内解决挑战的比例最高的策略是：

Rank	Seed Word	Ranking Algo	Decision Rule	% Solved in 3 Steps	Lead Over Next Rank
1	raile	Max Remaining Candidates	Baseline	48.12%	11 games
2	roate	Max Remaining Candidates	Baseline	47.65%	12 games
3	stare	Max Remaining Candidates	Baseline	47.13%	3 games
4	soare	Max Remaining Candidates	Baseline	47.00%	6 games
5	tales	Max Remaining Candidates	Baseline	46.74%	12 games

图片由作者提供。

限制：对于机器人，由机器人

这篇文章和其他类似文章的发现的主要局限性在于，推荐的助记词不一定适用于随意的 Wordlers。我们已经表明，最佳种子词取决于您玩游戏的方式以及您尝试优化的指标。除非你可以像机器人一样玩，它具有 (1) 关于解决方案空间的完美信息和 (2) 使用排名算法评估大部分的计算能力，否则适合你的最佳种子词可能会有所不同。

也就是说，人类玩家只关注种子词是可行的，因为这是可行的。如果我们绝对需要关于使用什么种子词的建议，我们应该看看那些在三个指标和各种排名算法中通常表现良好的词（在上图中以各自的颜色绘制）。他们可以在不同风格的比赛中表现出色，但需要做更多的工作来用更像人类的机器人来验证这一点，并理解为什么这些词能很好地发挥作用。

结论

在这篇文章中，我们展示了改变策略的其他部分，即 (1) 排名算法和 (2) 在解决问题与信息收集之间确定优先级的决策规则会影响“最佳”种子词。所讨论的指标，无论是 (i)

达到解决方案的平均步骤数，(ii) 成功率，还是 (iii) 在 3 个步骤内解决的挑战的比例，对于确定“最佳”的含义很重要，并且通过扩展，“最佳”种子词是什么。

基于这些结论，我们不应该简单地接受根据非人机器人玩的模拟游戏的结果生成的种子词推荐。需要进行更深入的研究，以确定无论是谁（或什么）在使用它们以及游戏是如何玩的，都能很好地发挥作用。

参考

1. J. Wardle, [Wordle](https://powerlanguage.co.uk) (2020), powerlanguage.co.uk。
2. B. Smyth, [我从玩超过一百万个单词游戏中学到的东西](#)(2022), 迈向数据科学。
3. T. Glaiel, [Wordle 中数学上最优的第一次猜测](#)(2021), Medium。
4. T. Neill, [破坏乐趣：Wordle 自动求解器](#)(2022), 在派对上不好玩。
5. S. Dua, [深入了解 Wordle, 新的流行拼图热潮](#)(2022), 迈向数据科学。
6. B. Bakhtiari, [关于 Wordle 的一句话](#)(2022), 走向数据科学。
7. J. Stechschulte, [■■■■■ Optimal Wordle](#) (2022), 迈向数据科学。
8. MM Liu, [字里行间是什么?](#) (2022), 中等。

数据科学 单词 优化 模拟

注册变量

通过迈向数据科学

每个星期四，Variable 都会提供 Towards Data Science 的精华：从实践教程和前沿研究到您不想错过的原创功能。[看一看。](#)

通过注册，您将创建一个Medium帐户（如果您还没有）。查看我们的[隐私政策](#)了解有关我们隐私惯例的更多信息。

 获取此时事通讯

[关于](#) [帮助](#) [条款](#) [隐私](#)

获取 Medium 应用程序

