DRAFT 2.0

SpellsNFT

Probability of a boggle string in a grid of characters

L.C.L, J.P.S

January 4, 2022

1 Introduction

We wish to estimate the probability of finding a string of a certain length within a grid of arbitrary size. We assume that valid strings formed from the grid must correspond to those considered valid by the rules of Boggle.

Let the grid of characters be of size $N \times N$ cells. Let L be the length of the target string. From these assumptions, we note that $\binom{N^2}{L}$ unique combinations of L cells exist within the grid.

Furthermore, we note that $\frac{N^2!}{(N^2-L)!}$ permutations of those unique combinations of cells can be generated from the grid.

We will call each of these permutations a "path" through the grid. Some of the paths may be invalid paths due to cells not being boggle-adjacent to one another and some of the paths will be valid. We will denote a possible path from the set of $\frac{N^2!}{(N^2-L)!}$ paths as $\{e_i\}_{i=1}^{L-1}$.

Our initial goal will be to estimate the probability of finding an ordered string, $\{s_i\}_{i=1}^{L}$, and path of the grid, $\{e_i\}_{i=1}^{L-1}$. An edge, e_i, may be understood as $e_i = (e_i^{source}, e_i^{target})$, where e_i^{source}, e_i^{target} are coordinates, $(x_i^{source}, y_i^{source})$, $(x_i^{target}, y_i^{target})$, respectively. Specifically, we can write this probability down as,

$$\Pr(\{s_i\}_{i=1}^{L},\ \{e_i\}_{i=1}^{L-1})$$

It should be noted that for this derivation, edges, e_i, could be comprised of illegal pairings of grid points. For example, $e_i = ((0,1),\ (8,2))$ and $e_i = ((8,2),\ (0,1))$ can be part of a path. Of course, this example represents one of the invalid paths.

After estimating the above joint probability, we will marginalize out all possible paths (both legal and non-legal),

$$\Pr(\{s_i\}_{i=1}^{L}) = \sum_{\text{paths of length L}} \Pr(\{s_i\}_{i=1}^{L},\ \{e_i\}_{i=1}^{L-1})$$

Let's start by rewriting the joint probability,

$$\begin{aligned}
\Pr(\{s_i\}_{i=1}^{L},\ \{e_i\}_{i=1}^{L-1}) &= \prod_{j=2}^{L} \Pr(s_j|\{s_i\}_{i=1}^{j-1},\ \{e_i\}_{i=1}^{j-1}) \\
&\times \prod_{k=2}^{L-1} \Pr(e_k|\{s_i\}_{i=1}^{k},\ \{e_i\}_{i=1}^{k-1}) \\
&\times \Pr(e_1|s_1) \times \Pr(s_1)
\end{aligned}$$

1.1 Assumptions

Let us assume that characters in the target string are independent of the path locations and vice-versa,

$$\begin{aligned}
s_j|\{s_i\}_{i=1}^{j-1},\ \{e_i\}_{i=1}^{j-1} &\perp\!\!\!\perp \{e_i\}_{i=1}^{j-1} \\
e_k|\{s_i\}_{i=1}^{k},\ \{e_i\}_{i=1}^{k-1} &\perp\!\!\!\perp \{s_i\}_{i=1}^{k}
\end{aligned}$$

Our joint probability is then,

$$\begin{aligned}
\Pr(\{s_i\}_{i=1}^{L},\ \{e_i\}_{i=1}^{L-1}) &= \prod_{j=2}^{L} \Pr(s_j|\{s_i\}_{i=1}^{j-1}) \\
&\times \prod_{k=2}^{L-1} \Pr(e_k|\{e_i\}_{i=1}^{k-1}) \\
&\times \Pr(e_1) \times \Pr(s_1)
\end{aligned}$$

Assuming grid cells are populated with characters independently of one another, then

$$s_j|\{s_i\}_{i=1}^{j-1} \perp\!\!\!\perp \{s_i\}_{i=1}^{j-1}$$

Resulting in a joint probability of

$$\begin{aligned}
\Pr(\{s_i\}_{i=1}^{L},\ \{e_i\}_{i=1}^{L-1}) &= \prod_{j=1}^{L} \Pr(s_j) \\
&\times \prod_{k=2}^{L-1} \Pr(e_k|\{e_i\}_{i=1}^{k-1}) \\
&\times \Pr(e_1)
\end{aligned}$$

1.2 A computable expression

According to our current knowledge of SpellsNFT's design specifications,

$$\Pr(s_j) = \frac{1}{26} \ \forall j$$
$$\Pr(e_1) = \frac{1}{|\mathrm{Nei}(e_1^{source})|}$$

Furthermore,

$$\Pr(e_k|\{e_i\}_{i=1}^{k-1}) = \begin{cases} \frac{\mathbb{1}[e_k^{source} \in \text{visited}]}{|\mathrm{Nei}(e_k^{source}) - (\mathrm{Nei}(e_k^{source}) \ \cap \ \text{visited})|} & , e_k^{target} \in \mathrm{Nei}(e_k^{source}) - (\mathrm{Nei}(e_k^{source}) \ \cap \ \text{visited}) \\ 0 & , \text{otherwise} \end{cases}$$

where visited $= \bigcup_{i=1}^{k-1}\{e_i^{source}, e_i^{target}\}$ (i.e. the visited cells of the current path) and $\mathrm{Nei}(\cdot)$ is a function giving us the nearest boggle-adjacent cells (i.e. neighboring cells in the boggle-sense). One caveat being that you cannot be a neighbor of yourself.

1.3 Putting it all together

$$\begin{aligned} \Pr(\{s_i\}_{i=1}^{L}) &= \sum_{\text{paths of length L}} \Pr(\{s_i\}_{i=1}^{L}, \ \{e_i\}_{i=1}^{L-1}) \\ &= \sum_{\text{paths of length L}} \left[\prod_{j=1}^{L} \Pr(s_j) \times \prod_{k=2}^{L-1} \Pr(e_k|\{e_i\}_{i=1}^{k-1}) \times \Pr(e_1) \right] \\ &= \frac{1}{26^L} \sum_{\text{paths of length L}} \left[\frac{1}{|\mathrm{Nei}(e_1^{source})|} \prod_{k=2}^{L-1} \Pr(e_k|\{e_i\}_{i=1}^{k-1}) \right] \end{aligned}$$

Noting that $\Pr(e_k|\{e_i\}_{i=1}^{k-1}) = 0$ whenever we have an invalid path according to the rules of Boggle, the above sum can be reduced to

$$\Pr(\{s_i\}_{i=1}^{L}) = \frac{1}{26^L} \sum_{\text{valid paths of length L}} \left[\frac{1}{|\mathrm{Nei}(e_1^{source})|} \prod_{k=2}^{L-1} \frac{1}{|\mathrm{Nei}(e_k^{source}) - (\mathrm{Nei}(e_k^{source}) \ \cap \ \text{visited})|} \right]$$

where visited $= \bigcup_{i=1}^{k-1}\{e_i^{source}, e_i^{target}\}$.