Solving *The Wiki Game*: Efficient Traversal of the Wikipedia Hyperlink Graph

Russell Schwartz, Nayana Suvarna

Carnegie Mellon University - Graduate AI

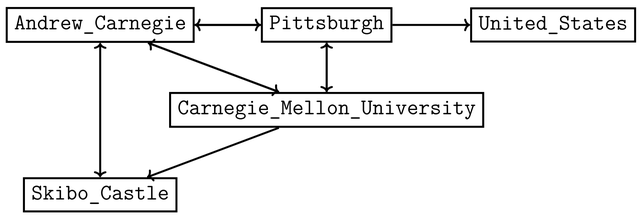
Introduction

Wikipedia is the largest and most-read reference work in history, containing over 16 million richly-typeset articles covering nearly every topic imaginable. Wikipedia articles are famous, in part, for their heavy use of hyperlinks, which connect articles to (presumably) related articles.

The Wiki Game is a popular game in which the player is given two randomly chosen wikipedia articles, the start and the goal, and is tasked with navigating from one to the other by clicking on available hyperlinks. The resulting path is often interesting or humorous. For example:

`Andrew_Carnegie` → `Carnegie_Mellon_University` →
↪ `List_of_Carnegie_Mellon_University_people` → `Shrek`

The Wiki Game can be played in real time against other players on websites like this one: `www.thewikigame.com`. Solutions are scored on their length (i.e. number of clicks) and the time it it took to find a path.



This is effectively a shortest-path problem on a directed unweighted graph. The obvious approach is to perform a breadth-first search from the start page. However, the size of the graph and the network latency involved in fetching a new page make this direct approach prohibitively slow.

We investigate more sophisticated techniques including A* search, iterative-deepening search, and local caching of the graph, with the goal of achieving super-human performance under modest memory constraints.

Related Work

Wikipedia is a commonly used dataset in natural language processing (e.g. [4]). It is generally valued as a corpus of text spanning a large number of disciplines. Only limited work has been done to solve the Wikipedia shortest path problem, most notably, the *6 degrees of Wikipedia* project [3]. However, this implementation can be quite slow for pages of distance ≥ 5 (taking on the order of minutes).

On the other hand, the more general problem of developing heuristics for shortest-path queries in large graphs is very well studied. One potentially useful family of heuristics involves selecting "landmark" nodes within the graph, pre-computing the distance from every node to these landmarks, and then using these values to bound cost-to-go [2]. This is commonly used in GPS route-planning [1].

Methods

Data Processing

In order to efficiently analyze the data, the entirety of English Wikipedia (text only) was downloaded from an internal server data dump (about 25GB compressed). Each page was assigned a unique 32-bit ID. All pages were then scanned, their links extracted, and the resulting adjacency lists written to a local database (about 2GB). The resulting graph has **16,714,619 vertices** and **207,586,495 edges**.

Static Analysis

Some statistics were computed on the graph to inform search algorithm development:

- distribution of out-degrees and in-degrees (via the transpose graph)
- number of strongly connected components (via Kosaraju's Algorithm)

Search Algorithms

BFS, Iterative Deepening DFS, and A* were all implemented in python. The landmarks-based heuristic described in [2] was used for A*. A set of 15 landmarks L was selected and all distances $D(v, \ell)$ were computed. The heuristic is evaluated as

$$h(v) = \max_{\ell \in L} \{D(v, \ell) - D(t, \ell)\}$$

where t is the goal node. This can be shown to be admissible by the triangle inequality. Landmarks were selected by iteratively choosing the vertex furthest from all previously selected landmarks.

Testing

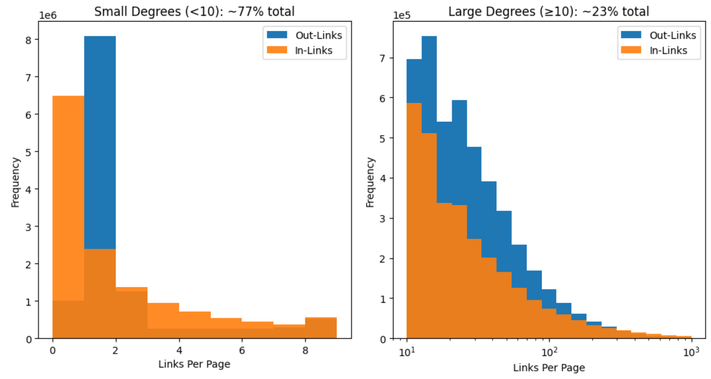
Each search algorithm was tested on a fixed set of 100 pairs of articles and evaluated based on the run time and number of nodes visited.

Statistical Findings

Degree Distribution

The majority of articles (about 77%) have fewer than 10 in-links or out-links, but there is a long tail. The average degree is **12.4**.

- Most linked to: `United_States` (270,266 in-links)
- Most linked from: `Index_of_Singapore-related_articles` (12,351 out-links)



Connected Components

An iterative form of Kosaraju's algorithm was run on the graph in order to extract all strongly connected components, with some unexpected results. The top 4 by size:

- **8,928,883 pages** - the main bulk of Wikipedia
- **35 pages** - yearly results for the polish football club "Wisła Kraków" (1922 - 1956)
- **34 pages** - yearly composition of the "All-Eastern football team" (1947 - 1979)
- **33 pages** - an assortment of seemingly-unrelated disambiguation and surname pages all beginning with the letter L

Results

The algorithms were evaluated on an Intel i7 4790K @ 4.0GHz with 16GB memory. The tests consisted of a fixed set of 100 pairs of pages drawn from the largest SCC.

Algorithm	Avg Runtime	Avg Nodes Visited	Max RAM Usage
Breadth-First Search	365 ms	3334	173 MB
Iterative Deepening	600 ms	655492	<1 MB
A* with Landmarks	4718 ms	1826	>1 GB*

There are tradeoffs for each search algorithm. A* visited the fewest number of nodes, but required the most memory and was the slowest due to the overhead of the priority queue. The longest path encountered by randomly generating pages was length 8 (from `Linear_inequality` to `Metrocorp_Bancshares`). Some other fun shortest paths:

- `Artificial_intelligence` → `Alan_Turing` → `Diethylstilbestrol` → `Hamster`
- `Kevin_Bacon` → `Erdős_number` → `Paul_Erdős`

Contrary to the popular "Six-Degrees-of-Separation" hypothesis [5], we found the true diameter of the graph to be at least **103**, evidenced by an isolated doubly-linked-list of pages `List_of_highways_numbered_XXXX` from `XXXX` = `1084` to `1187`.

Conclusion

The tradeoffs in the 3 search algorithms make them each suitable for different settings. In an online setting where network latency is the bottleneck, A* might be fastest overall since it requires fetching the fewest pages. With a local cache of the data, BFS can solve the Wiki Game much faster than any human player.

Work is ongoing to accelerate the A* implementation and to refine the search heuristic with better landmark selection.

References

[1] R. Chen and C. Gotsman. Efficient fastest-path computations for road maps. *Computational Visual Media*, 7(2):267–281, 2021.

[2] A. Goldberg and C. Harrelson. Computing the shortest path: A* search meets graph theory. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 04 2003.

[3] J. Wenger. Six Degrees of Wikipedia — sixdegreesofwikipedia.com. `https://www.sixdegreesofwikipedia.com`, 2018.

[4] T. Yano and M. Kang. Taking advantage of wikipedia in natural language processing. 2008.

[5] L. Zhang and W. Tu. Six degrees of separation in online society. 2009.

*including pre-computed landmark distance arrays (comparable to BFS without)