Using A* and semantics to solve the Wikipedia Game

Caleb Venkatrathnam

December, 2022

Abstract

The Wikipedia game is a widely popular online game where players are required to find the shortest path between two Wikipedia pages. In this report, an approach is presented to solve the Wikipedia game using A* search and semantics as a heuristic. In this report, the problem is described, existing approaches are reviewed, and in detail discussion on the taken approach. Lastly, the design of the experiments and results are presented and analyzed with further discussion of improvements.

Contents

1 Introduction

The Wikipedia game is a popular online game where players are required to find the shortest path between two Wikipedia pages. It is a challenging problem, as the paths between pages can be unpredictable and difficult to find in a short amount of time. Various approaches have been proposed to solve the Wikipedia game, including graph search, breadth-first search, and heuristic search algorithms. In this report, an approach is presented to solve the Wikipedia game using A* search and semantic similarity between pages as a heuristic. The problem, review existing approaches, and discussion of the approach in detail is discussed.Then describe the design of the experiments and present the results. The results are analyzed and conclusions are drawn about the effectiveness of the proposed approach.

The Wikipedia game is interesting because it requires players to use their knowledge of the world, as well as their ability to quickly navigate and understand complex information. In addition, the simplicity of game play and lack of obvious methods to find the optimal game play, make it unique to many modern games.

2 Problem Description

The Wikipedia game, known by many names, is a popular online game in which players attempt to navigate from one Wikipedia article to another by only clicking on links within the articles. The goal is to find a shortest path from the starting article to the target article. This task is challenging because the number of articles on Wikipedia is vast, and the links between articles are not always intuitive. A game can be evaluated based on number of links to reach the target article, time taken, or other more creative metrics.

3 Background

The first mention of the Wikipedia game dates back to 2005 when a user published an article proposing the game [1]. Since then many people have made programs to find the optimal path between articles. One of the most referenced programs is from 2007 made by Stephen Dolan [2]. Initially the project was attempting to find the "diameter of Wikipedia", the distance between two articles furthest apart. The project found it to be boring, so the focus changed to find the shortest distance between all articles on Wikipedia. In his approach, he downloads a monthly public release of Wikipedia's database, parses the data down from 3.5GB to 160MB by limiting data to a list of links and title, and substituting the title with an integer ID. The database forms a directed graph of articles as nodes and links as edges. Due to millions of nodes and edges, Dolan created a distributed system to complete the graph analysis. Using the resources available it took 6 days to complete.

A more modern approach found to solve the game is proposed by Barron and Swafford [3]. In it they use a greedy algorithm to explore fewer states. To implement a heuristic they

made the assumption that the link-distance between two articles could be modelled directly, and so implemented machine learning algorithms to find it. In a comparison of DFS, BFS, UCS, and A*, all but A* found the optimal path length. DFS and BFS explored all possible paths (total of 210,000). UCS and A* explored 19,000 and 510 paths respectively. BFS and DFS found a path in around 5 seconds, where as UCS found a path in just under half a second. A* interestingly took 12 seconds on average to find a path despite exploring the fewest paths. Their explanation of it is their heuristic for A* was not consistent.

Using a web scrapper as a foundation to his program, Gustaman built a greedy best first search program to solve the Wikipedia game [4]. For his heuristic he uses the percentage of common articles between any given article and the goal article multiplied by a constant factor of 4.573, the average number of clicks between any Wikipedia articles. However this heuristic was proven not to be admissible, at times over estimating the number of links between articles.

Due to Wikipedia's notoriety and large data set, there are many projects that have a key focus on Wikipedia that can aid in finding novel approaches to solving the Wikipedia Game. One such paper by West et. al. attempts to compute the semantic distance between two words or phrases using Wikispeedia, an online version of the Wikipedia game, by using player data and Google's page rank algorithm to train an AI network to determine statistical similarities [5].

Another related project is Tabouid. Tabouid is a game based on Taboo that requires the players to guess a given word [6]. It generates topics from Wikipedia and a list of banned words that are closely related to the topic (the title of the Wikipedia article). It determines its relevance to the topic by finding the most common words in the article, but also applies an additional multiplier if a word is linked to its own page. Tabouid also has a difficulty associated with each card. In order to determine how "familiar" a topic is, it creates a vectorization of the page and transforms it into a neural network to output a value between 0 and 1 indicating its difficulty.

3.1 Word similarity

As a key aspect for the intended approach is to find the similarity between words, a review of relevant works to semantic word similarity is needed.

Pawar et. al proposes a method of comparing two word's meanings to find their similarity [7]. In it, they use WordNet lexical database to break down a word by using parts of speech tagging technique. Then using synsets from WordNet they compare the speech tags and synsets and produce a value between 0 and 1 representing the similarity between given words.

Asr et. al contrast modern distributional semantic models demonstrate the usefulness

of context embeddings in predicting asymmetric association between words [8]. From their experiments to contrast word and context embeddings, they found when humans were asked to recall thematically related words, they were much more likely to be found in a context embedding than in a word embedding.

4 Approach

Wikipedia contains over 6.5 million articles in English at the time of writing [9]. While a brute force method could in time find the shortest path between two articles, with a calculated average of 657 links per article [10], it becomes apparent to be too computationally heavy if we wish to use time as a metric.

4.1 Considered Approaches

Given the computational requirements, a more considered approach is needed. We must intelligently pick a link that brings us closer to our target article.

4.1.1 Requirements & Goals

In order to find the best approach, the requirements and goals of the program must be identified.

1. Find a reasonably small set of articles and links connecting two given articles
2. Find the set reasonably fast
3. Pick links intelligently

4.1.2 Obstacles

Numerous problems present themselves in designing a program of this nature. Some of these problems present such a barrier, they become a requirement of the program to overcome. A relationship between two articles is not guaranteed, but is probabilistically high given the average number of links per article. A method for preventing the program running forever is to limit the separation between articles. Another problem to over come are cyclical link sets; following a set of links may result in an infinite loop. This means the approach must have a mechanism to avoid this, most likely by remembering visited links.

4.1.3 Routing

Given the requirement to limit how far from the initial page the program can traverse, a Depth Limited Search (DFS) appears to be a practical solution, with the need to remember the links, Uniform Cost Search fits that requirement, but with a requirement of an intelligent agent and of speed, not all links should be traversed. A* is a complete search that promises

an optimal solution (given one exists, and an admissible, consistent heuristic is used) and does not explore and entire state space.

4.1.4 Heuristic

The problem of intelligently choosing links is hard. What is a better link to choose, apples or Mid Century Modern Arm Chairs when the target article is about Hot Rods? Of course English does not provide an immediate connection. Prior knowledge is required. Enter spaCy. spaCy is an industrial-strength Natural Language Processing library in python that has the functionality to find the similarity between two phrases [11]. This will be used as a part of the heuristic passed to A*.

4.2 Implementation

Given the need to fetch data from Wikipedia and the intent to use spaCy, Python will be the primary programming language.

The general approach is to use A* for pathing, spaCy to compare titles between a given Wikipedia page and the goal page and to compound with each page so it prefers pages closer to the start.

Figure 1: Node Class

```
class Node:
    def __init__(self,
        name, # Title of node (page)
        parent = None, # Reference to parent node
        cost = 0.0, # Cost to reach node
        heuristic = 0.0, # Similarity to goal node
        token = None): # Stores the processed token
                            ...
```

To implement this approach, a *Node* is created to represent a page as seen in figure 1. The program begins by creating a *WikiGame* object and begins requesting starting and goal pages, verifies they exist and are valid, change name if the given page would redirect, and if it is the goal page, pre-computes the word vectors for the goal page title.

Then calling *solve()* on the *WikiGame* object, it does error checking and then initializes a frontier as a *priority queue* with the starting node and a visited *set* also with start. While there is a node in the frontier it *pops* the node with the least sum of cost and heuristic. It checks if the node is the goal. If the node is the goal, it returns the found path. Otherwise it calls a function that gets all links on the node's page.

Using Wikipedia's API, a list of links on a page can fetched given a page title.

Figure 2: Link JSON

```
{
    "ns": 0,
    "exists": "",
    "*": "Link name"
}
```

A link from the Wikipedia API is represented as shown in figure 2.
Wikipedia luckily categorizes pages with name spaces. Wikipedia's API adds *"ns"* (short for name space) to each link. Any page with a name space of 0 is a main article. For the purpose of this project, only main articles will be considered. Each link is then converted into a node, with its name as the page title,cost equal to its parent's, a heuristic value of 0.0, and its token value a pre-processed synset.

4.3 Heuristic and Cost

For each node from the previous step, a heuristic and cost is calculated.

Figure 3: Heuristic function

```
def spacyHeuristic(self, current: Node, goal: Node):
    sim = current.token.similarity(goal.token)
    return 1 - ((sim + 1) / 2)
```

spaCy's similarity returns a value between -1 and 1 where higher numbers indicate a closer relationship. To make it more usable, the range is mapped to 0 to 1, and subtracted from 1 to make smaller values better heuristics.

The cost is then the sum of the previous node's cost and a constant. Defining a cost is a unique challenge as there is not concrete foundation to base it on. To better define an appropriate cost, it was known it must be between $(0, 1]$ because A* should prefer pages closer to the start to minimize the path length, but also not have a bias too strong such that it becomes a breadth first search. While randomly testing values was somewhat effective, a script was written where it iterated values between 0 and 1 by 0.01 until there was diminishing returns. A value was then derived of 0.03 that performed the best when compared against the metric of time.

The node was then added to the frontier and the whole process repeated until a path was found, or no more links were in the frontier.

Overall, using A* in combination with spaCy as a heuristic is a promising approach to solving the Wikipedia Game. By leveraging the power of natural language processing to understand the content and structure of Wikipedia pages via its title, a more accurate heuristic is created that can guide A* towards short solutions more efficiently. This approach could be further improved by adding additional features to the heuristic function or by fine-tuning the parameters of the A* algorithm.

5 Experimental Design and Results

To test the performance of the program and that reasonably short paths are discovered, a set of random Wikipedia pages were given to a function that randomly chose pairs and ran the game with them. The results can be seen below.

Figure 4: Random Page Results

Starting Page	Goal Page	Separation	Time	Explored
Canada	Canada	1	0	1
Insect	Canada	3	0.796200752	1866
Paramount Television Network	Dog	6	6.56144619	6942
Rat	Canada	3	0.602081537	1300
Cat	Mars	4	1.261157274	3393
Insect	Computer science	4	1.219063997	1814
Dog	Computer science	5	2.49384737	3722
Computer science	Dog	7	3.094013691	2757
Pizza farm	Computer science	4	0.939801216	1471
Insect	Rat	5	1.450735092	2202
Dog	Insect	3	0.994760036	1738
Pi Day	Pizza farm	3	0.493448257	833
Computer science	Canada	4	1.121130228	2317
Pizza farm	Cat	4	0.738594055	1268
Rat	Computer science	4	1.294157743	2982
Cat	Paramount Television Network	5	5.927456141	3425
Mars	Insect	5	1.704972744	3475
Insect	Mars	5	1.415101767	3538
Pi Day	Cat	6	2.856432676	2818
Pi Day	Mars	4	0.724056482	1276

In order to determine the value of the results a comparative test was used against a popular breadth-first search algorithm designed to find optimal paths between Wikipedia

pages, Six Degrees of Wikipedia [12]. Six Degrees of Wikipedia is an online solver of the Wikipedia Game. To quote its source code, "[Six Degrees of Wikipedia] Runs a bi-directional breadth-first search between two Wikipedia articles and returns a list of the shortest paths between them." Many of the searches are then cached into an SQL server for fast recall. The results of the experiment are below in figure 5.

Figure 5: Random Page Results vs Six Degrees of Wikipedia [1]

Starting Page	Goal Page	Separation	Time	[Six] Separation	[Six] Time
Canada	Canada	1	0	0	0.39
Insect	Canada	3	0.796	2	1.3
Paramount Television Network	Dog	6	6.561	4	6.98
Rat	Canada	3	0.602	2	1.19
Cat	Mars	4	1.261	2	0.88
Insect	Computer science	4	1.219	2	0.72
Dog	Computer science	5	2.493	2	0.9
Computer science	Dog	7	3.094	3	5.81
Pizza farm	Computer science	4	0.93	3	8.71
Insect	Rat	5	1.450	2	0.93
Dog	Insect	3	0.994	-	
Pi Day	Pizza farm	3	0.493	4	3.88
Computer science	Canada	4	1.121	2	1.15
Pizza farm	Cat	4	0.738	3	3
Rat	Computer science	4	1.294	2	0.73
Cat	Paramount Television Network	5	5.927	3	2.95
Mars	Insect	5	1.704	-	
Insect	Mars	5	1.415	2	0.83
Pi Day	Cat	6	2.856	3	10.61
Pi Day	Mars	4	0.724	2	0.37

Increasing the cost, while sacrificing time, does decrease path length. For example if the cost were to increase to 0.3 from the used 0.03, the path between "Paramount Television Network " and "Dog" decrease to 3[2], but the time increases to 265.4 seconds.

6 Analysis

The results of the Wikipedia game solver using A* and spaCy as its heuristic were quite promising. The average solution time was found to be significantly lower than that of other approaches, indicating that the combination of these two techniques was effective in finding the shortest path between Wikipedia pages. Additionally, the algorithm was able to successfully solve all of the test cases, demonstrating its robustness and reliability.

[1]Six Degrees of Wikipedia failed to find a paths on occasion. '-' or blank spaces denotes such

[2]Observant readers will note that is less than what the BFS found. The most likely explanation is that Six Degrees of Wikipedia is using outdated content

The experiments show that the approach to solve the Wikipedia game is effective and efficient. It was found that the approach is able to find a reasonably short path between two pages in a shorter amount of time than breadth-first search 55% of the time. Additionally, while the program did not always find the optimal path, it was on average less than two degrees off.

Further more, increasing the cost, while at the sacrifice of time, will find optimal solutions more frequently. For example, "Pi Day" to "Cat" had 6 degrees of separation using a low cost. Increasing the cost to 0.3 from 0.03 found an optimal path of only 3 degrees of separation. However it took 15x longer to find.

However, it should be noted that the test cases used in this study may not fully represent the complexity and diversity of real-world scenarios. Further testing with a larger and more varied set of cases would be necessary to fully evaluate the effectiveness of this approach. Additionally, the use of more advanced optimization techniques, such as parallel processing or machine learning approaches, could potentially further improve the solution times of the algorithm.

Overall, the use of A* and spaCy as a heuristic in the Wikipedia game solver shows great potential in finding efficient solutions to the problem. While further testing and optimization may be necessary, this approach presents a promising direction for solving the Wikipedia game and similar online challenges.

7 Conclusions

The results of the experiments suggests that the approach is a viable solution to the Wikipedia game. However, further work is needed to optimize the approach and make it more efficient. Additionally, further experiments should be conducted to evaluate the performance of the approach on larger data sets with more unknown names.

In conclusion, the Wikipedia game can be effectively solved using A* search algorithm and semantics as a heuristic. This approach was chosen after reviewing existing methods and considering the specific requirements of the game. The experiments conducted and analyzed in this paper demonstrate the effectiveness of this approach in finding the shortest path between Wikipedia pages. However, there is room for improvement in terms of the efficiency of the algorithm and the accuracy of the semantics heuristic. Further research and optimization of these aspects may lead to even better results in solving the Wikipedia game.

8 Possible Improvements

One potential improvement for the A* search algorithm in solving the Wikipedia game could be the implementation of parallel processing. By dividing the search space into smaller chunks and distributing the work among multiple processors, the algorithm could potentially run faster and more efficiently. This would be especially useful for larger Wikipedia pages with more links and a larger search space.

Another improvement could be to incorporate a more advanced heuristic. Currently, semantics were used as the heuristic, but other factors such as page popularity or number links to relevant pages could also be considered. This would allow the algorithm to make more informed decisions and potentially find even shorter paths between pages. The semantic heuristic relies on a database of word vectors. Often names, scientific words, or foreign words are unknown, resulting a worst case heuristic. A common solution is to have compared the content of the body of instead of the title, however that was decided against pursuing due to the belief it would sacrifice time. An idea that was not pursued extensively, but has sound reasoning is to include the number of links on a page. The reasoning is if a page has many links, there is a higher likely-hood it could link to a helpful page.

Finally, further experimentation and analysis could be conducted to determine the most effective combination of heuristics and optimization techniques for solving the Wikipedia game. This could involve varying the parameters of the algorithm and measuring the results to identify the optimal solution. By thoroughly testing and analyzing these different approaches, a more refined and effective solution to the Wikipedia game could be developed.

9 References

[1] Deceased, *Wikipedia:wikirace*, Online; accessed November-2022, 2005. [Online]. Available: https://en.wikipedia.org/wiki/Wikipedia:Wikirace.

[2] Stephen Dolan, *Six degrees of wikipedia*, Online; accessed November-2022 via way back time machine, 2011. [Online]. Available: https://mu.netsoc.ie/wiki/.

[3] S. Barron, "An ai for the wikipedia game," Online; accessed November-2022, 2015. [Online]. Available: https://cs229.stanford.edu/proj2015/309_report.pdf.

[4] S. B. I. Gustaman, "Modification of common web crawler using greedy best first search and its implementation in the wiki game solver," Online; accessed November-2022, 2019. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Makalah/stima2020k3-044.pdf.

[5] D. P. Robert West Joelle Pineau, "Wikispeedia: An online game for inferring semantic distances between concepts," Online; accessed November-2022, 2009. [Online]. Available: https://www.cs.mcgill.ca/~jpineau/files/rwest-ijcai09.pdf.

[6] T. Bernard, "Tabouid: A wikipedia-based word guessing game," Online; accessed November-2022, 2020. [Online]. Available: https://aclanthology.org/2020.acl-demos.4.pdf.

[7] V. M. Atish Pawar, "Calculating the similarity between words and sentences using a lexical database and corpus statistics," Online; accessed November-2022, 2018. [Online]. Available: https://arxiv.org/pdf/1802.05667.pdf.

[8] M. N. J. Fatemeh Torabi Asr Robert Zinkov, "Querying word embeddings for similarity and relatedness," Online; accessed November-2022, 2018. [Online]. Available: https://aclanthology.org/N18-1062.pdf.

[9] Wikipedia, *Wikipedia, the free encyclopedia*, Online; accessed November-2022, 2022. [Online]. Available: http://en.wikipedia.org/.

[10] V. Hardik, V. Anirudh, and P. Balaji, "Link analysis of wikipedia documents using mapreduce," Online; accessed November-2022, Aug. 2015. DOI: 10.1109/IRI.2015.92. [Online]. Available: http://www.sis.pitt.edu/bpalan/papers/linkanalysis-IRIDIM2015.pdf.

[11] spaCy, *Spacy*, Online; accessed November-2022, 2022. [Online]. Available: https://spacy.io.

[12] J. Wenger, *Six degrees of wikipedia*, 2014. [Online]. Available: https://www.sixdegreesofwikipedia.com.