The handle http://hdl.handle.net/1887/29764 holds various files of this Leiden University dissertation.

**Author:** Takes, Frank Willem  
**Title:** Algorithms for analyzing and mining real-world graphs  
**Issue Date:** 2014-11-19
Part II

Path Traversal Patterns
This chapter introduces a set of measures for determining the difficulty — for a human — of traversing paths in networks. The focus is on determining which node-based and path-based structural graph properties and measures say something about the difficulty of finding a certain path between two given nodes in a graph. Using a large corpus of over two million traversed paths on the online information network Wikipedia it is possible to demonstrate how the proposed techniques are able to accurately assess the human difficulty of finding a path between two given Wikipedia articles. The clickpaths analyzed in this chapter originate from the Wiki Game, an online game in which the main task is to connect two given random Wikipedia articles in as few clicks as possible. This chapter is based on:

7.1 Introduction

Searching and navigating through structured information such as Wikipedia, the web or a social network has become a common activity for many users. In this chapter we will analyze the way in which humans traverse structured data in search of a specific piece of information. The main goal of this study is to measure, understand and predict the difficulty of finding a path between two documents within a structured collection of information.

The motivation for this work comes from the idea that understanding the difficulty of path traversal may lead to a better understanding of human search behavior in general [62], which may in turn lead to improvements in the search strategy of an artificially intelligent search algorithm. Moreover, if we understand the aspects which complicate path traversal within structured data, then this information can possibly be used to improve the structure of the linked data itself [16].

Although search engines [40] can often help to find the content within a structured dataset that the user is looking for, sometimes search engine performance does not exactly meet the needs of the user [133]. This can happen for example because the user does not know the exact keyword that describes what he is looking for, because the search query was misinterpreted by the search engine, or because the required information is not indexed and is possibly located within the so-called Deep Web [58]. The Deep Web is the part of the internet which is not accessible to search engines, for example because the content resides within a database, because the pages are dynamic based on specific properties or settings of the user or because the content is only accessible from a limited range of machines.

When browsing for a piece of information within an information network, the user will have to reach the desired article by traversing the links that exist between the articles within the information network, forming a path towards the correct piece of information. We will study this type of path traversal by analyzing over two million paths traversed by (human) users of the well-known online encyclopedia Wikipedia (http://www.wikipedia.org). An advantage of studying paths on Wikipedia compared to for example clickstreams from the world wide web [11] is that Wikipedia contains much less “noise”, referring to to duplicate, false or untrusted information.

The Wikipedia paths analyzed in this chapter were gathered from the Wiki Game (http://www.thewikigame.com), a free online game in which the user is asked to connect two given random articles on Wikipedia. That is, starting from a certain source article, the main objective of the user is to reach the goal article by repeatedly following the clickable links within Wikipedia articles. This paper studies the difficulty of this particular task. If we are able to a priori determine the expected difficulty of a task, then this can be used to define multiple levels of difficulty for the Wiki Game.
Chapter 7. The Difficulty of Path Traversal in an Information Network

As an example of a path traversal task which is to be solved by a player of the Wiki Game, consider the path from the Wikipedia article on MP3 to the article on Northern Ireland. An actual (computed) shortest path of length 3 runs subsequently via the articles on the United States and Ice Hockey (see Figure 7.1). Human users attempting to find a path tend to know that Northern Ireland is somewhere in Europe, so from the article on MP3 they first find their way to an article related to Europe, for example via the page on the Internet which is a direct link from the article on MP3. Next, they will for example navigate to the article on the United Kingdom, from where they find the article on Northern Ireland. Some users take another detour on the way, for example via the page on the Republic of Ireland and the page on Ireland (island).

In turns out that humans, especially after some practice, are often able to link two given random articles on Wikipedia in less than ten clicks. This is actually a quite remarkable accomplishment, because even though a standard backtracking algorithm is certainly able to match or even beat humans in terms of path length, a human instead does not use millions of backtracking steps, but rather relies on background knowledge in terms of expected semantic relatedness [48] to find a path. Incorporating such extensive knowledge into an algorithm for classifying path difficulty, for example via ontologies [146], may in large information networks such as Wikipedia be very complex.

Throughout this chapter a range of node-based and path-based structural network properties and measures are proposed as indicators for the difficulty of connecting two articles. An advantage of considering structural features is that they may capture the direct relationship between the various concepts within the network, independent of which exact information network is studied. Also, structural properties are
relatively easy to derive and compute, and do not require prior knowledge about the dataset. Moreover, while both the content as well as the linking structure of Wikipedia are subject to change, the classifiers that are proposed will only be affected by the second type of change, as article semantics are not considered. We will measure the quality of the proposed difficulty indicators by comparing their performance with the average human performance in terms of success or failure at completing a path.

The rest of this chapter is organized as follows. First, Section 7.2 discusses some notation, the various datasets used in this chapter and the main problem statement. We discuss related work in Section 7.3. Next we describe, analyze, test and compare the aspects which influence the difficulty of path traversal, at a node-based and a path-based scale, in Section 7.4 and 7.5, respectively. Finally, Section 7.6 concludes.

7.2 Preliminaries

In this section we discuss various concepts, definitions, notation and the considered datasets. Finally, we formulate the main problem statement and verification approach.

7.2.1 Concepts & definitions

The information network is represented by a directed graph $G = (V, E)$ with $n = |V|$ nodes and $m = |E|$ links. When we talk about a path between two nodes $u, v \in V$, we mean a sequence consisting of at least two nodes, starting at $u$ and ending at $v$, where there is a link from each node to the next node in the sequence. A shortest path between two nodes $u, v \in V$ is a path of length $\ell \geq 1$ between $u$ and $v$ for which there is no other path from $u$ to $v$ of length smaller than $\ell$. The length of such a shortest path, or in short the distance, is denoted by $d(u, v)$. Obviously, cycles may occur in paths, but not in shortest paths. Of course, it can happen that there are no (shortest) paths ($d(u, v) = \infty$) or that there are multiple (shortest) paths connecting two nodes. Because the graph is directed, it can happen that $d(u, v) \neq d(v, u)$. We define the indegree of a node $v \in V$ as the number of links pointing to node $v$, and similarly, the outdegree as the number of links pointing from node $v$ to some other node.

7.2.2 Wikipedia

According to its own definition, “Wikipedia is a free, web-based, collaborative, multilingual encyclopedia project with over 3.9 million articles in English alone” (as observed in 2011). Considering solely the content of the articles and the links it contains, Wikipedia can be seen as a large directed graph, where each node represents an article, and each directed link a hyperlink within the source article pointing to the
target article. In this study we will use the August 2011 English dataset of Wikipedia pagelinks from DBpedia version 3.7 (see [10] or http://dbpedia.org), from which we consider only the links to other Wikipedia articles, so we exclude links to external websites. Links to “special” articles such as articles describing a file, the category to which an articles belongs, or translations of the article, are also ignored. After some pruning and cleaning, the final Wikipedia graph that will be used for this study has statistics as presented in Table 7.1.

We note that the edge-to-node ratio, the diameter, defined as the length of a longest shortest path, the effective diameter (the 90-th percentile of the cumulative distribution of shortest path lengths), the average distance (between two nodes, sampled over 10,000 node pairs) and the size of the largest weakly connected component (WCC) are consistent with that of other small-world networks [140]. The Wikipedia network furthermore has a power-law node degree distribution [149], suggesting that the Wikipedia graph indeed resembles other frequently studied real-world networks, such as the world wide web [12], internet topology networks [46] and social networks [76].

### Table 7.1: Wikipedia dataset.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Articles ((n))</td>
<td>3,464,902</td>
</tr>
<tr>
<td>Directed links ((m))</td>
<td>82,019,786</td>
</tr>
<tr>
<td>Largest WCC</td>
<td>99.9%</td>
</tr>
<tr>
<td>Average indegree</td>
<td>26</td>
</tr>
<tr>
<td>Average outdegree</td>
<td>22</td>
</tr>
<tr>
<td>Average distance ((\bar{d}))</td>
<td>4.8</td>
</tr>
<tr>
<td>Effective diameter</td>
<td>7</td>
</tr>
<tr>
<td>Diameter</td>
<td>11</td>
</tr>
</tbody>
</table>

#### 7.2.3 The Wiki Game

The Wiki Game is an online game launched in 2009, in which the user is assigned the task of connecting two given random articles on Wikipedia. Starting from a certain source article, the main objective is to reach the goal article by repeatedly clicking links on the page of the current article. While various types of games such as “Five clicks to Jesus”, and “Six degrees of Wikipedia” exist, we will solely focus on “Speed Race” games, in which the task is to connect two given random Wikipedia articles in as few steps as possible, as quickly as possible, ultimately with a time limit of 120 seconds.
The Wiki Game dataset $T$ consists of games (or tasks) and associated user-generated paths. A task $t \in T$ is essentially a (start, goal) pair $(u, v)$ indicating between which two articles $u$ and $v$ (with $u, v \in V$) a path has to be formed. For each of these tasks we have a list of paths generated by the (fully anonymized) users that made an attempt at solving this task. These paths describe either a successful or a failed attempt at finding the goal article, and each have an associated path length. The data was filtered to exclude non-serious attempts (more than 40 clicks per task, or no clicks at all). This resulted in a dataset as presented in Table 7.2. Apparently, on average a task was performed by 5 to 6 users, and little less than one third of the total set of tasks presented to the users was successfully completed.

Figure 7.2 further clarifies the shortest path lengths of the tasks in the dataset, as well as the path length of the user-generated paths. We observe that even though shortest paths of length greater than 6 exist within Wikipedia, none of these tasks were included in the database of attempted tasks. Most tasks had a shortest path length somewhere between 2 and 4. Apparently, the average distance between the two pages composing a task is lower than the average distance in the entire Wikipedia dataset, indicating a small bias towards less obscure start and goal pages in the task database.

Figure 7.2 also shows how the distribution of the successful user-generated paths follows the same distribution as that of the shortest paths, but with an average path length that is roughly 2 times larger than the shortest path length (between 5 and 7), and a relatively fat tail. The distribution of the path length over all user-generated paths is clearly dominated by the failed paths, but as opposed to the successful paths, these distributions roughly follow a fat-tailed power law, indicating that when people “drop out” the path traversal process, they frequently do this early in the traversal process.

### 7.2.4 Problem definition

The main goal is to assess the difficulty of finding a path between two nodes in a directed graph:
Given a directed graph $G = (V, E)$ and nodes $u, v \in V$, can we assign a function value $f(u, v) \in [0; 1]$ indicating the difficulty of finding a path from $u$ to $v$?

In this chapter we will consider various approaches (or difficulty classifiers) of assigning such a function value. We will evaluate the quality of an approach based on a comparison with the results obtained by the users on tasks from the Wiki Game. For each of the user-generated paths of a certain task $t \in T$ we know whether or not the path was successfully formed, allowing the definition of the average percentage of success $g(t) \in [0; 1]$ for task $t$. This information will serve as a ground truth for assessing the quality of the various difficulty classifiers.

Each difficulty classifier $f$ can assign a function value $f(t)$ to all tasks $t \in T$, which allows us to create a partition $\{T_1, T_2, \ldots, T_q\}$ of the set of tasks $T$. The partitioning is done in such a way that the tasks within each $T_i$ have the same function value (range), so that the (average) function value of the tasks in $T_i$ is always greater than the average function value of the tasks in $T_{i-1}$, and where every $T_i$ is maximal in size. The partitions can be used to define $q$ different difficulty levels for the Wiki Game.

The overall quality of a classification measure will be determined by computing the Pearson correlation coefficient $c(f, g)$ of the classifier $f$ and average percentage of success of the user-generated paths $g$, defined as:
\[ c(f, g) = \frac{q \sum_i f(i) g(i) - \sum_i f(i) \sum_i g(i)}{\sqrt{q \sum_i f(i)^2 - (\sum_i f(i))^2} \sqrt{q \sum_i g(i)^2 - (\sum_i g(i))^2}} \]

Here, \( \bar{f(i)} \) is equal to the average function value \( f(t) \) of paths \( t \in T_i \), and \( \bar{g(i)} \) is the average percentage of success of the paths in \( T_i \). As a second measure of comparison we will also consider the Spearman rank correlation coefficient \( rc(f, g) \) of \( f \) and \( g \), defined as:

\[ rc(f, g) = \frac{\sum_i (f(i) - \bar{f})(g(i) - \bar{g})}{\sqrt{\sum_i (f(i) - \bar{f})^2 \sum_i (g(i) - \bar{g})^2}} \]

Here, \( \bar{f} \) and \( \bar{g} \) are equal to the average value over all \( i \) of \( f(i) \) and \( g(i) \), respectively. This coefficient measures the extent to which the relation between the classifier output and path difficulty can be described using a monotonic function. If we want a task at a certain difficulty level to always be harder than a task at the previous level, then we primarily aim for a high rank correlation coefficient.

In general, we will call a measure \( f \) correlated with path difficulty if it has a correlation greater than 0.8 (or smaller than \(-0.8\)) with the percentage of success \( g \). For simplicity, we will denote the correlation coefficient and rank correlation coefficient by \( c \) and \( rc \), respectively.

### 7.3 Related work

The structure behind Wikipedia has been analyzed in great detail, addressing tasks such as improving the linking structure [102] and automatic disambiguation of articles [63]. Furthermore, Wikipedia is frequently used as a knowledge base for external knowledge discovery tasks [138], and can serve as an excellent platform for computing (semantic) relatedness of concepts [48]. Patterns within clickpaths have also been analyzed extensively [24], and have proven useful for tasks such as page prediction [1, 119]. These patterns are often found within clickstreams from the web, where there is a great deal of “noise”, i.e., duplicate, false or untrusted information. On the web, information is frequently authored by one person or a very small group of people, whereas the number of participating users of Wikipedia is sufficiently large to counter spammers that spread for example false or biased information.

West and Leskovec [141] have compared human navigation in information networks such as Wikipedia with that of agents, using a dataset similar to the dataset
studied in this chapter. They found that humans, when navigating within an information network, have expectations about what links should exist and base a high level reasoning plan upon this, and then use local information to navigate through the network. They furthermore mention that humans often miss “good” link opportunities on a page as their idea of semantic relatedness often overrules opportunistic clicking. In [142], the same authors show that progress in a goal-finding task is easiest far from and close to the target, with hubs being crucial in the beginning. To the best of our knowledge, the issue of path difficulty has so far not been addressed.

7.4 Node-based difficulty measures

In this section we consider node-based difficulty measures, by which we refer to properties that can be derived solely based on a node and its neighborhood (so, local information), in this case the Wikipedia article and its linked or linking articles. The advantage of such properties is that they are relatively easy to compute, and that they do not require knowledge about the entire dataset, which can be an advantage in extremely large datasets such as the world wide web or Wikipedia.

7.4.1 Degree measures

Having a large number of outgoing links for a certain node is likely to make it easier to directly reach a larger part of the graph from that particular node. Similarly, we expect that the number of incoming links of a node will probably make it relatively more easy to reach that node from any other node. We will verify the actual influence of these two measures of path difficulty by analyzing $q = 100$ ranges of the indegree of the goal article and the outdegree of the start article. The result is depicted in Figure 7.3, and a Bezier curve is drawn to better visualize the overall correlation. We observe no real significant correlation with the outdegree of the starting article ($c = 0.637$ and $rc = 0.789$). However, a strong correlation ($c = 0.850$ and $rc = 0.960$) is noticeable with respect to the indegree of the goal article and the actual percentage of success.

We can conclude from these results that the degree of the goal article is of significant influence to the difficulty of finding a certain path, whereas the degree of the starting node does not appear to play a notable role. An advantage of the node-based degree measure is that because the graph is stored as an adjacency list, the measure can be computed in $O(1)$. 

7.4.2 Neighborhood measures

As the indegree is apparently a relevant indicator for the difficulty of finding a certain goal, it makes sense to refine this measure. Therefore we define the $h$-neighborhood $N_h(v)$ of a node $v \in V$ as the set of nodes with distance at most $h$ from $v$, more specifically: $N_h(v) = \{w \in V \mid d(v, w) \leq h\}$. Similarly, we can define $N'_h(v) = \{u \in V \mid d(u, v) \leq h\}$, the reverse neighborhood, which is the set of all articles $u$ with distance at most $h$ to $v$. The $h$-neighborhood size is the number of nodes in the neighborhood of $v$, denoted by $|N_h(v)|$, and similarly we can define the reversed $h$-neighborhood size $|N'_h(v)|$. Obviously, 1-neighborhood size and reversed 1-neighborhood size are equal to the outdegree and indegree of a node plus 1 (the node itself), respectively.

We compared the neighborhood measures described above with path difficulty and found a significant correlation with the reversed 2-neighborhood size, which is essentially looking one step further than indegree. The functionality of this method can be explained by looking at the example graph in Figure 7.1. There, the article on Ice Hockey and the article on Ireland (island) both have an indegree of 1, while based on the degree of the neighbors, Ice Hockey seems much easier to reach than Ireland (island). This is nicely reflected by the reversed 2-neighborhood size, as $|N'_2(\text{Ireland (island)})| = 3$ and $|N'_2(\text{Ice Hockey})| = 6$, whereas the indegree would consider the

![Figure 7.3](image-url)  

Figure 7.3: Start outdegree and goal indegree (horizontal axes, logarithmic) vs. percentage successful (vertical axis).
two nodes equally difficult to reach.

Figure 7.4 shows a plot of \( q = 100 \) intervals of the reversed 2-neighborhood of the goal article, again compared to the success percentage, and strong correlation coefficients \( (c = 0.915 \) and \( rc = 0.978 \)) can be observed. Especially for the hardest tasks in the database \((g(t) < 0.35)\), looking beyond the indegree helps to increase the amount of monotonicity.

In line with results from the previous section, the 2-neighborhood of the starting node did not appear to be correlated with the path difficulty \( (c = 0.397 \) and \( rc = 0.492)\). Furthermore we mention that, even though in some graphs it might make sense to look at (reverse) neighborhoods larger than \( h = 2 \), in the dense Wikipedia graph, considering more than the 2-neighborhood will quickly yield almost the entire graph, and indeed, correlation coefficients lower than 0.5 are observed when considering larger neighborhoods.

The neighborhood measures discussed in this subsection can be computed in \( O((m/n)^{h-1}) \) time per task. The average node indegree (or outdegree), \( (m/n) \), is between 20 and 30, still allowing for quick computation of the measure, especially in case of \( h = 2 \). So, the reversed 2-neighborhood size is a good indicator for path difficulty, whereas measures related to the degree of the starting article or neighborhood do not appear to be good at classifying path difficulty. This can be explained
by considering the small-world property of the Wikipedia: with relatively few steps it is possible to reach a large portion of the graph. It seems plausible that the starting node on its own is of little influence in general, because the user will often find his way to a hub-like node very quickly, from where the actual search for the goal node starts.

7.5 Path-based difficulty measures

In contrast with the previous section, we will now look at path-based properties, meaning that we look at actual paths between start and goal nodes in order to determine the difficulty of finding a path, possibly using global knowledge about the entire graph. Although the outcome in terms of difficulty prediction strength is expected to be higher, the computation time of path-based measures is longer: $O(m)$ per task.

7.5.1 Path length

When selecting two random articles on Wikipedia, due to the small-world property of the Wikipedia graph, the probability of selecting a pair of articles that is at a small distance of each other, is quite large. This is reflected in Figure 7.2, where the shortest path distribution of all played games is depicted, as well as the distribution of human formed path lengths of all successful paths. As mentioned in Section 7.2, the distribution of human path lengths does appear to have the same distribution shape as that of the actual shortest paths, suggesting a correlation between shortest path length and path difficulty.

Whereas we were able to aggregate the node-based measures from the previous section into $q = 100$ intervals, in case of path length we only have 6 different values. In Figure 7.6, the solid line shows for each actual distance (shortest path length) the percentage of successful human paths. This shows a strong correlation coefficient of $c = -0.957$ between the computed shortest path length and the percentage of successful paths, and an obvious rank correlation of $r_c = -1.000$. However, a downside of considering distance as an indicator for difficulty is obviously the fact that it is only possible to define $q = 6$ different difficulty levels.

7.5.2 Number of shortest paths

We may also choose to look at the number of shortest paths $\sigma(u,v)$ between the start and goal article $u$ and $v$. Intuitively, if there is only one shortest path from the start node to the end node, the task will be much harder compared to when there
Chapter 7. The Difficulty of Path Traversal in an Information Network

would have been thousands of shortest paths. Luckily, computing actual shortest path lengths is as easy as counting the number of shortest paths, as \( \sigma(u, u) = 1 \) and \( \sigma(u, v) = \sum_{w \in B(u, v)} \sigma(u, w) \) with \( B(u, v) = \{ w \in N_1'(v) \mid d(u, v) = d(u, w) + 1 \} \) [26]. The question is then how the number of shortest paths should be incorporated in a function value for assessing path difficulty. The number of shortest paths alone showed no significant correlation with path difficulty, which is understandable: a path of length 2 with 20 possible shortest paths is expected to be much easier to find than a path of length 4 with 20 shortest paths. So we propose to combine the distance with the number of shortest paths:

\[
d_{sp}(u, v) = d(u, v) + \alpha \left( 1 - \frac{\log \sigma(u, v)}{\max_{w, z \in V} \log \sigma(w, z)} \right)
\]

The reason why we take the log of \( \sigma(u, v) \) is motivated by Figure 7.5, where the plots of “shortest paths” indicate how the distribution of the number of shortest paths for each shortest path length decreases logarithmically. The parameter \( \alpha \geq 0 \) defines the amount of focus on the number of shortest paths as compared to the distance. If this parameter is set to 1, then a path of length 4 with only 1 possible shortest path is assumed to be easier to find than a path of length 5 with 2000 different shortest paths. Using linear parameter tuning with steps of 0.25, we obtained the best result for \( \alpha = 1.5 \), where we observe a strong correlation of \( c = -0.895 \) and \( r_c = -0.876 \).

Figure 7.5: Frequency (vertical axis) of the number of shortest paths and number of unique nodes (horizontal axis) on these paths for each distance.
with path difficulty. The results are depicted in Figure 7.6.

### 7.5.3 Uniqueness of shortest paths

To further refine the measure from the previous section, we propose to look at the number of distinct nodes that occur within these shortest paths. This measure is based on the intuition that shortest paths quickly overlap, and that the extent to which paths overlap may influence the difficulty of a path finding task. For example, in Figure 7.1, the 3 shortest paths of length 3 from MP3 to United Kingdom run through a total of 4 different nodes: United States, Internet, Europe and Ice Hockey. The maximum number of unique nodes on 3 shortest paths of length 3 is 6 (3 times 2 unique intermediary nodes). Somewhat inspired by betweenness centrality, we propose to divide the number of nodes on the actual shortest paths by the maximum possible number of intermediary nodes, a measure which we will call *shortest paths uniqueness*. For the example, this results in a score of \( \frac{4}{6} \approx 0.67 \). We will incorporate this measure along with the distance in the difficulty classifier defined as:

\[
dusp(u, v) = d(u, v) + \beta \left(1 - \frac{\log(\psi(u, v))}{\log(d(u, v) \cdot \sigma(u, v))}\right)
\]

Here, \( \psi(u, v) \) is a function that returns the number of distinct nodes on the shortest

![Figure 7.6: Various path-based measures (horizontal axis) vs. percentage successful (vertical axis).](image-url)
Chapter 7. The Difficulty of Path Traversal in an Information Network

Table 7.3: Summary of correlation coefficients ($c$), rank correlation coefficients ($rc$) and complexity (per task) of the proposed difficulty classifiers for $q$ difficulty classes.

<table>
<thead>
<tr>
<th>Difficulty classifier</th>
<th>Complexity</th>
<th>$q$</th>
<th>$c$</th>
<th>$rc$</th>
</tr>
</thead>
<tbody>
<tr>
<td>goal indegree</td>
<td>$O(1)$</td>
<td>100</td>
<td>0.850</td>
<td>0.960</td>
</tr>
<tr>
<td>start outdegree</td>
<td>$O(1)$</td>
<td>100</td>
<td>0.637</td>
<td>0.789</td>
</tr>
<tr>
<td>goal reversed 2-neighborhood size</td>
<td>$O(m/n)$</td>
<td>100</td>
<td>0.915</td>
<td>0.978</td>
</tr>
<tr>
<td>start 2-neighborhood size</td>
<td>$O(m/n)$</td>
<td>100</td>
<td>0.397</td>
<td>0.492</td>
</tr>
<tr>
<td>start-goal distance</td>
<td>$O(m)$</td>
<td>6</td>
<td>−0.957</td>
<td>−1.000</td>
</tr>
<tr>
<td>distance + number of shortest paths</td>
<td>$O(m)$</td>
<td>100</td>
<td>−0.895</td>
<td>−0.876</td>
</tr>
<tr>
<td>distance + shortest paths uniqueness</td>
<td>$O(m)$</td>
<td>100</td>
<td>−0.924</td>
<td>−0.925</td>
</tr>
</tbody>
</table>

paths between $u$ and $v$. The used values are again logarithmic as a result of the distribution of the number of unique nodes on the shortest paths, as depicted by the set of plots of “unique nodes” at various distances in Figure 7.5. The parameter $\beta \geq 0$ indicates the amount of focus on the number of distinct nodes over all shortest paths, and linear tuning of this parameter in steps of 0.25 showed that the best results were obtained for $\beta = 1.75$. The performance of the measure is displayed by the dotted line in Figure 7.6. We note that there are some “hickups” present in Figure 7.6, which might suggest that there is a better way of combining the two measures of distance and the uniqueness and number of shortest paths. The method nevertheless shows a correlation of $c = −0.924$ and $rc = −0.925$, demonstrating how shortest paths uniqueness does refine the path-based difficulty indicator from Section 7.5.1 based on the node-to-node distance.

7.6 Conclusion

Throughout this chapter we have proposed and analyzed the effectiveness of a range of techniques for classifying path traversal difficulty in information networks. The results are summarized in Table 7.3, in which the best-performing node-based and path-based measures are shown in bold.

With respect to the effectiveness of the various measures, we can generally say that node-based measures related to the goal article, such as the reversed neighborhood size, appear to be most effective, whereas node-based properties of the source article appear to be of little influence to path difficulty. In line with related work, we found that a user generally tends to quickly find his way to a hub node, from where the actual search process starts.

As for the path-based measures considered in this work, the distance between
two articles is a good measure of difficulty, although as a result of the small-world property of Wikipedia, the range of different distances and thus the range of difficulty levels is very limited. Incorporating the number of shortest paths and the percentage of unique nodes over all shortest paths results in a path-based classifier with slightly better performance. However, a clear downside of the proposed path-based methods based on the number of shortest paths and their uniqueness, is that they require one parameter to be tuned. Furthermore, due to the higher complexity of path-based measures, one may favor the node-based classifiers in a practical application, such as in the Wiki Game. There, the difficulty classifiers outlined in this chapter could be used to improve the user experience by allowing users to select a desired difficulty level at which they want to play.

In future work we would like to improve our difficulty measures by including more article-specific information, such as the link density of the considered article. Furthermore, we want to analyze the frequent subpaths that exist both in the successful as well as in the failed paths created by humans. This information may help to obtain a better understanding of the search process of a certain user or group of similar users, possible allowing personalization of the difficulty indicators.

Acknowledgment

We thank Alex Clemesha, creator of the Wiki Game, for providing the data.