

Technische Universität Graz
Dekanat für Informatik
Institut für Wissensmanagement



Analyse und Modellierung des menschlichen Navigationsverhaltens in einem Wikipedia-Netzwerk

**Masterarbeit
von
Florian GEIGL, B.Sc.**

Vorgelegt zur Erlangung des
akademischen Grades eines Master
der Studienrichtung Informatik

Graz, im März 2013

Betreuer der Masterarbeit:
Assoc.Prof. Dipl.-Ing. Dr.techn. Denis HELIC

.....

EIDESSTÄTTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am 14. März 2013

.....
(Unterschrift)

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

14th March 2013

.....
(signature)

Bei allen Bezeichnungen, die auf Personen bezogen sind, meint die Formulierung beide Geschlechter, unabhängig von der in der Formulierung verwendeten konkreten geschlechtsspezifischen Bezeichnung.

Danksagung

Ich möchte mich herzlichst bei meinem Betreuer Dipl.-Ing. Dr.techn. Denis Helic bedanken, der diese äußerst interessante Masterarbeit ermöglichte, sich um die dafür nötigen Ressourcen kümmerte und jederzeit ein offenes Ohr für etwaige Fragen hatte. Des Weiteren gilt mein Dank Dipl.-Ing. Dr.techn. Markus Strohmaier, welcher mich in das Donnerstagstreffen des Institutes aufgenommen hat, wodurch ich einen guten Einblick in professionelles, wissenschaftliches Arbeiten bekommen habe. Auch lernte ich dort andere, in diesem Themengebiet forschende Studenten und ihre Arbeiten kennen, wodurch mein Wissen über dieses Themengebiet deutlich erweitert wurde und ich qualitativ hochwertiges Feedback zur eigenen Arbeit bekam.

Ein großer Dank gilt meinen Kollegen Michael Eder und Markus Horwath, mit welchen ich gemeinsam dieses Studium von Anfang an erfolgreich bestritten habe und welche mir auch bei dieser Arbeit mit Rat und Tat zur Seite standen.

Außerordentlich möchte ich mich auch bei meinen Eltern Andrea Geigl und MR. Dr. Horst Geigl, M.Sc. bedanken, welche mir nicht nur dieses Studium ermöglicht haben, sondern mir vor allem jederzeit liebevoll zur Seite standen und sich immer ausreichend Zeit für meine Anliegen nahmen. Weiters gilt hier der Dank meiner Schwester Dr. Iris Geigl. Ganz besonders möchte ich mich auch bei meiner Freundin Carolina Probst bedanken, welche mir gerade in den stressigen Phasen der Masterarbeit eine große, liebevolle und wichtige Stütze war.

Kurzfassung

Diese Masterarbeit versucht das menschliche Navigationsverhalten in einem Wikipedianetzwerk möglichst gut zu modellieren. Sinn dieser Modellierung ist es die Änderungen von Netzwerkstrukturen bezüglich der Navigierbarkeit für Menschen zu testen. Für die Modellierung wurden bereits aufgezeichnete Navigationen von Menschen analysiert. Dabei wurden im Detail die Abbruchrate und der Knotengrad von besuchten Knoten untersucht.

Bezüglich der Knotengrade stellte sich heraus, dass Menschen anfangs versuchen sich einen Überblick über das Netzwerk zu verschaffen, indem sie Knoten mit hohem Grad besuchen. Diese Eigenschaft wurde in bereits zuvor durchgeführten Forschungsergebnissen in diesem Themengebiet hervorgehoben. Auch wurde schon gezeigt, dass mit einfachsten Mitteln ein Algorithmus geschaffen werden kann, welcher effizienter als Menschen navigiert. Ziel dieser Masterarbeit war es jedoch keinen besseren Algorithmus zu finden, sondern bereits vorhandene Algorithmen möglichst gut dem menschlichen Verhalten anzupassen.

Dazu wurden in jedem Navigationsschritt die möglichen, nächsten Knoten anhand ihrer Gradeigenschaften gefiltert. Zu diesem Zweck wurde ein bereits vorhandenes Navigationsframework um ein Modul zur Beschränkung von Links erweitert. Durch etliche Versuche mit unterschiedlichen Parametern stellte sich diese Restriktion jedoch als zu strenges Verfahren heraus. In Zukunft wäre es empfehlenswert in diesem Bereich mit Gewichtungen zu arbeiten.

Die Simulation der Abbruchrate brachte äußerst erfolgreiche Ergebnisse zum Vorschein. Es konnte nicht nur das Verhältnis von erfolgreichen zu nicht erfolgreichen Navigationen angepasst werden, sondern auch die Längen der einzelnen Navigationen an die aufgezeichneten Benutzerdaten angeglichen werden.

Als wichtiger Aspekt stellt sich heraus, dass zukünftig viel Arbeit in die Modellierung des menschlichen Hintergrundwissens gesteckt werden muss, um ein möglichst menschenähnliches Verhalten simulieren zu können. Außerdem sollte das Hintergrundwissen nicht immer eindeutige Ergebnisse liefern, da der Mensch auch sehr oft intuitiv handelt.

Abstract

This master thesis tries to create a model of the human behavior of navigation in a wikipedia-network as good as possible. The aim of this modeling is to test the change of network-structures regarding the navigability for humans. To this purpose recorded navigations of humans were analyzed. Especially the attrition rate and node degrees of all visited nodes were explored in detail.

Concerning the node degrees it turned out that at the beginning people always try to get an overview of the network by visiting nodes with a high degree. This characteristic was already mentioned in earlier research results on this topic. Furthermore it was shown that it is possible to create an algorithm by using very basic methods, which is able to navigate more efficient than people do. After all it was the aim of this master thesis to modify an already existing algorithm to match the human behavior instead of creating a better one.

To attain this goal possible next nodes were filtered in each step of the navigation according to their degree-properties. For this purpose an already existing navigation-framework was extended for one module which can restrict the possible next nodes. Giving a lot of different parameters a trial this pathway turned out to be too strict. For the future it would be recommendable to work with weights in this section.

The simulation of the attrition rate furnished extreme successful results. It's not only that the ratio of successful to unsuccessful navigations matches the human generated datasets, but also the distribution of the lengths of the generated navigations could be adapted.

One of the most important outcomes is that for future it is recommendable to work on the background-knowledge to adjust the model to the human navigation behavior. Furthermore the background-knowledge should not always produce clear results, because humans often use their intuitions in making decisions.

Inhaltsverzeichnis

1. Einführung.....	1
1.1 Motivation	1
1.2 Aufgabenstellung	2
1.3 Aufbau der Arbeit / Themengebiet	3
1.3.1 Flussdiagramm / Projekt-Überblick.....	3
2. Grundlagen	5
2.1 Graphentheorie.....	5
2.1.1 Knoteneigenschaften	9
2.1.2 Wege und Pfade	10
2.2 Hierarchien	12
2.3 Navigation in Graphen	12
2.4 Decentralized Search.....	13
2.4.1 Hintergrundwissen	13
2.4.2 Greedy-Algorithmen	14
2.5 Wikipedia.....	14
2.6 Wikigame.....	19
2.6.1 Community.....	22
2.6.2 Einschränkungen	24
3. Related Work	25
3.1 Benutzerverhalten im Internet	25
3.2 Benutzerverhalten in Wikipedia-Netzwerken.....	26
3.2.1 Wikipedia Netzwerke.....	27
3.2.2 Benutzerverhalten	28
3.2.3 Zoom-out und Zoom-in Phase.....	28
3.3 Kleine-Welt-Phänomen	30
3.3.1 Benutzermodellierung in Wikipedianetzwerken	32
4. Daten und deren Repräsentation	34
4.1 Graph.....	34
4.2 Hierarchie	35
4.3 Klickpfade	35
4.4 Paare.....	36

4.5	Kürzeste-Distanzen.....	36
5.	Arbeitsumgebung.....	38
5.1	SNAP.....	38
5.2	MUN-Framework	38
6.	Node-Selectoren.....	41
6.1	Random.....	41
6.2	Greedy.....	41
6.3	Teleport	42
7.	Link-Restrictor.....	43
7.1	Motivation	43
7.2	Implementierung	44
7.3	Einstellungsmöglichkeiten.....	44
8.	Tools zur Auswertung	47
8.1	Hierarchie Validierung.....	47
8.2	Generierung von Plots	48
8.3	Konvertierung von Graphen	48
8.4	Plots über diverse Pfad-Eigenschaften	49
9.	Ergebnisse.....	51
9.1	Graph-Analyse.....	51
9.2	Klickpfad-Analyse	54
9.2.1	Analyse mittels verschiedener Hierarchien.....	57
9.3	Baseline Random	60
9.4	Greedy.....	62
9.4.1	Unique-Greedy.....	64
9.5	Link-Restrictor.....	65
9.6	Kombination Link-Restrictor mit Abbruchsimulator	68
9.7	Kullback-Leibler-Divergenzen der Ergebnisse.....	72
9.7.1	Längenverteilungen.....	73
9.7.2	Verteilung der Knotengrade.....	73
10.	Lessons learned.....	75

11. Future-Work	76
11.1 Hierarchien	76
11.2 Modifikation des Greedy-Algorithmus.....	76
11.3 Filterung der Vergleichsdaten	76
11.4 Kommunikation Link-Restrictor und Node-Selector.....	77
11.5 Verteilung der Knotengrade	77
11.6 Metriken.....	77
Literaturverzeichnis	78
Abbildungsverzeichnis	81
Listingverzeichnis.....	83
Tabellenverzeichnis	84
Abkürzungsverzeichnis	85

1. Einführung

Dieses Kapitel bietet eine Einleitung in die vorliegende Masterarbeit. Hier werden die Motivation für diese Arbeit, die Aufgabenstellung und auch die Einteilung in die einzelnen Kapitel erklärt.

1.1 Motivation

Ursprünglich wurde das Internet ausschließlich von Programmierern erstellt. Sie erstellten die Hyperlinks zwischen den einzelnen Webseiten und damit auch die Struktur des größten Netzwerkes der Welt. Damals gab es noch keine ausgereiften Suchmaschinen und so navigierten sich die Internet-User mittels Anklicken von Links durch das enorm wachsende Informationsnetzwerk. [1]

Als bereits 1997 die ersten Wiki-Systeme im Internet auftauchten, das heißt das Web 2.0 erfunden wurde, änderte sich die Struktur des Netzwerkes rasant. Der Laie, welcher keine Programmierkenntnisse besaß, konnte nun Links zwischen verschiedenen Webseiten erstellen und somit nicht nur den Inhalt, sondern auch die Struktur des World-Wide Web verändern. [2]

Etwa zur selben Zeit wurde „Google“ gegründet und trug auch zur Veränderung des Benutzerverhaltens im Internet bei. Jedoch schafft „Google“ es bis heute nicht, die User mit ihren Suchergebnissen vollständig zufrieden zu stellen. Die Masse der Internet-User wählt einen Link aus den Suchergebnissen aus und navigiert sich anschließend von dort mittels Anklicken von weiterführenden Hyperlinks zur gewünschten Information. [3] [4].

Doch wie kann diese Navigation für die User vereinfacht werden? Anhand welcher Informationen entscheiden User, welchen weiterführenden Link sie wählen? Welche Kriterien sind notwendig um ein Netzwerk benutzerfreundlich zu gestalten? Diese Fragen sind besonders in Wikipedia-Netzwerken wichtig, da hier User oft nur das Themengebiet zur gesuchten Information wissen und trotzdem möglichst schnell ihren benötigten Artikel finden möchten. Da es jedoch nicht von Vorteil ist die Struktur solch großer, bedeutender Netzwerke zu verändern, um anschließend auf die Reaktion der Benutzer zu warten, wäre es

zielführender, die Navigation von Benutzern zu analysieren und anschließend zu modellieren. Mit einem derartigen Modell des menschlichen Navigationsverhaltens in solch großen Informationsnetzwerken wäre es durchaus möglich, umstrukturierte Netzwerke einfach und vor allem vergleichsmäßig schnell auf ihre Benutzerfreundlichkeit zu testen, ohne dabei um seine User fürchten zu müssen. Ein weiterer Schritt wäre folglich die automatische Modifikation von Netzwerken zugunsten der Navigierbarkeit für Internet-User.

1.2 Aufgabenstellung

Ziel dieser Masterarbeit ist es ein vorhandenes Navigationsframework zu erweitern um das menschliche Navigationsverhalten in Netzwerken simulieren zu können.

Das bereits vorhandene Framework stellt zwei Schnittstellen zur Verfügung um dieses zu erweitern. Eine davon ist das Node-Selector-Interface, an welches verschiedene Navigationsmechanismen hinzugefügt werden können. Anhand dieser Mechanismen wird der jeweils nächste Knoten in jedem Navigationsschritt gewählt. Das zweite Interface ist der Link- bzw. User-Interface-Restrictor. Diese Schnittstelle bietet die Möglichkeit, alle potentiellen nächsten Knoten, anhand eigens gewählter Kriterien zu filtern, bevor ein Node-Selector daraus den Nächsten auswählt.

Mittels Analyse von bereits gesammelten User-Navigationsdaten aus einem Online-Navigationsspiel sollen die Parameter des Simulators möglichst so eingestellt werden, um dem menschlichen Verhalten am ähnlichsten zu sein.

Zu diesem Zweck werden auch diverse Metriken benötigt, um einen Vergleich zwischen Simulation und Mensch anstellen zu können. Diese entstehen größtenteils aus bereits durchgeführten Versuchen und Erkenntnissen anderer Forschungsarbeiten in diesem Themenbereich.

Zur Benutzerfreundlichkeit des Frameworks sind auch diverse Plots gefordert, um die Ergebnisse der Simulationen graphisch aufbereitet vergleichen zu können.

1.3 Aufbau der Arbeit / Themengebiet

Anfangs wird in Kapitel **2** auf die Grundlagen des Themengebiets eingegangen. Hier werden grundlegende Begriffe der Masterarbeit erklärt, welche den Grundbaustein für das spätere Verständnis bilden.

Kapitel **3** widmet sich den bereits durchgeführten Forschungsergebnissen, auf welche sich diese Arbeit bezieht und auch teilweise anknüpft.

In Kapitel **4** wird kurz auf die bereits aufbereiteten Daten eingegangen und gezeigt, in welcher Form und Format diese für die durchgeführten Versuche bereits vorhanden waren.

Auf das Framework, mit welchem die Ergebnisse dieser Masterarbeit erzielt wurden, wird in Kapitel **5** eingegangen.

Die Erweiterungen und Tools des Frameworks, welche benötigt und eigens programmiert wurden, werden in Kapitel **6, 7** und **8** detailliert beschrieben.

Kapitel **9** beinhaltet alle produzierten Ergebnisse, inklusive einer ausführlichen Analyse dieser.

Gegen Ende wird in Kapitel **10** ein Resümee über die Arbeit gezogen und daraus gelernte Aspekte aufgezeigt.

Letztlich wird in Kapitel **11** erwähnt, welche weiteren Forschungen in diesem Themengebiet wünschenswert wären und eventuell auch zielführend sind.

Anschließend findet sich noch das Literaturverzeichnis, in welchem alle verwendeten Unterlagen angeführt sind, sowie ein Abbildungsverzeichnis und ein Abkürzungsverzeichnis.

1.3.1 Flussdiagramm / Projekt-Überblick

Da drei Masterarbeiten ([5], [6] und die vorliegende) aus dem zugrundeliegenden Forschungsprojekt entstanden sind, befindet sich hier eine graphische Übersicht über dieses Projekt inklusive detaillierter Einteilung in die besagten Masterarbeiten.

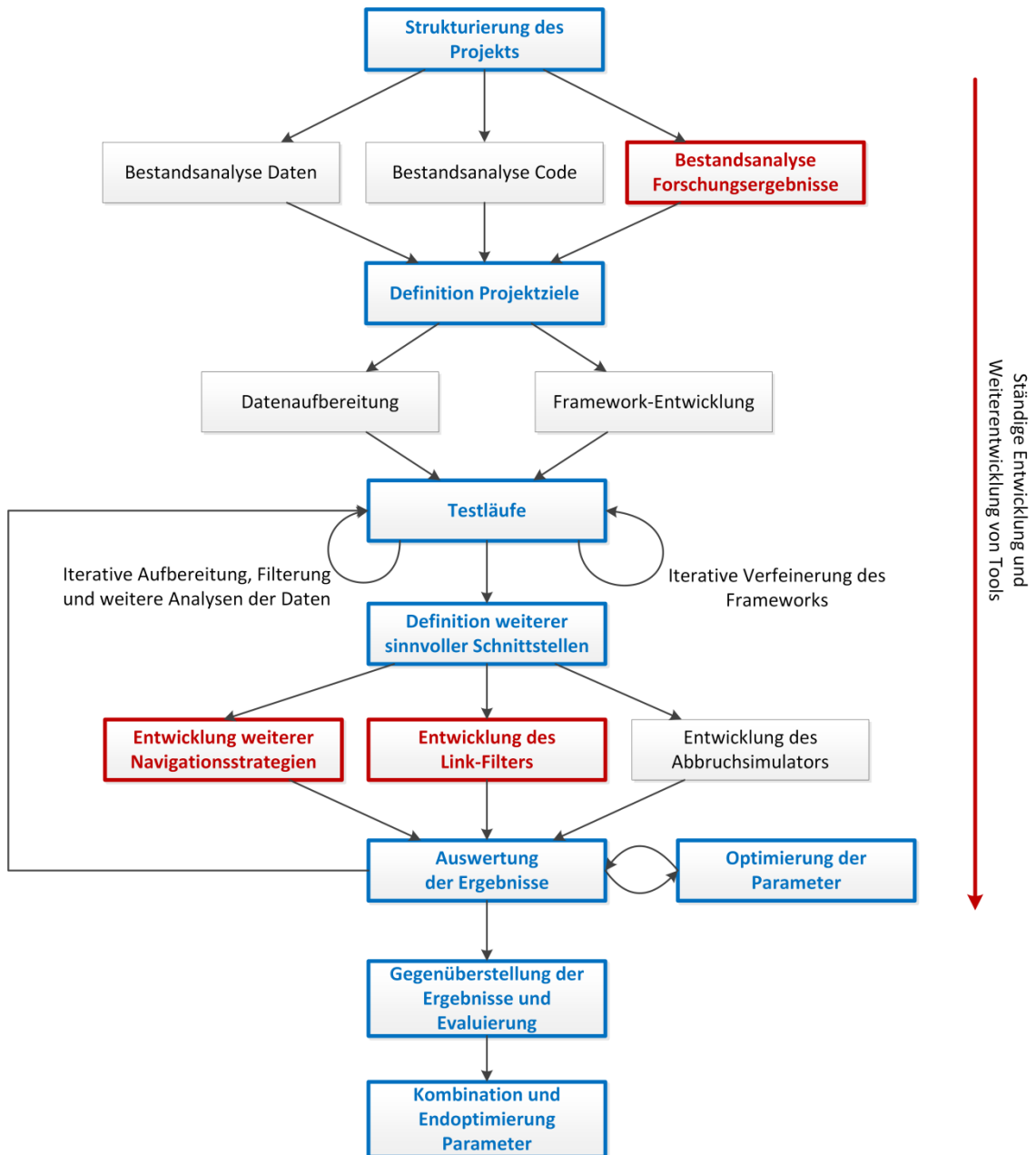


Abbildung 1: Projektübersicht

Rot markierte Felder sind Einzelarbeiten der vorliegenden Masterarbeit. **Blau** gekennzeichnete Objekte sind Tätigkeiten bei welchen zumindest eine Absprache mit dem Team notwendig war, Teamarbeit für optimale Ergebnisse benötigt wurde oder Kollegen dieselbe Arbeit durchführen mussten.

2. Grundlagen

Dieses Kapitel behandelt die thematischen Grundlagen dieser Masterarbeit. Es erklärt einige grundlegende Begriffe und Metriken der Graphentheorie als auch den Hintergrund und die Herkunft der verwendeten Daten.

2.1 Graphentheorie

Grundlage der Graphentheorie ist der Graph. Dieser besteht aus Knoten und Kanten. Knoten werden häufig verwendet, um reale Objekte oder Zustände zu modellieren. Die Kanten des Graphen stellen die Verbindung zwischen den einzelnen Knoten her und modellieren so die Relationen der Objekte oder einen Übergang von einem zum nächsten Zustand. In Abbildung 2 zeigt A einen Graph bestehend aus 3 Knoten und 3 Kanten. Knoten sind durch blau, numerisch beschriftete Kreise dargestellt, während Kanten die roten Verbindungen zwischen den Knoten sind.

Grundsätzlich wird bei Graphen zwischen endlichen und unendlichen Graphen unterschieden. Endliche besitzen im Gegensatz zu unendlichen eine endliche Menge an Knoten und Kanten. Unendliche werden jedoch zum Großteil nur in der Theorie verwendet, da reale Objekte und deren Verbindungen meist eine endliche Menge bilden. So wird auch in dieser Arbeit ausschließlich mit endlichen Graphen gearbeitet.

Des Weiteren wird zwischen gerichteten und ungerichteten Graphen differenziert. Diese beiden Begriffe beziehen sich auf die Relationen zwischen den einzelnen Knoten. In einem ungerichteten Graphen wird eine Verbindung von Knoten A zu Knoten B von beiden Seiten aus gleich angesehen, wohingegen im gerichteten Graphen jede Kante von einem Knoten zu einem anderen zeigt. Dies lässt sich am besten durch die Modellierung einer Straßenkarte als Graphen erklären. Dazu werden Kreuzungen als Knoten und Straßen zwischen den Kreuzungen als Kanten dargestellt. Straßen welche in beide Richtungen befahren werden können, sind ungerichtete Kanten, wohingegen Straßen welche nur in eine Richtung befahren werden dürfen, also Einbahnen, durch gerichtete Kanten modelliert werden. Ungerichtete Graphen können einfach in gerichtete

Graphen umgewandelt werden indem für jede ungerichtete Kante, eine Kante in jede Richtung eingefügt wird. Eine Umwandlung von einem gerichteten Graphen in einen ungerichteten ist auch möglich, jedoch gehen dabei Informationen über die Verbindung verloren. Einige Eigenschaften von gerichteten Graphen beziehen sich jedoch auf ihre Darstellung als ungerichtete Graphen. Dabei wird einfach die Richtung der Kante ignoriert. [7, p. 114] In Abbildung 2 zeigt A einen ungerichteten Graphen, wohingegen B ein gerichteter Graph ist.

Eine weitere Artenunterscheidung von Kanten sind gewichtete und ungewichtete Kanten. Während bei ungewichteten Kanten nur die Möglichkeit besteht, dass diese entweder vorhanden oder nicht vorhanden sind, so gibt es bei gewichteten Kanten die zusätzliche Option auf vorhandene Kanten Werte zu speichern, welche diese Verbindung genauer spezifiziert. Diese Werte werden Gewichte genannt. Ein Graph, welcher aus gewichteten Kanten besteht, wird gewichteter Graph genannt. In dieser Arbeit werden jedoch nur ungewichtete Graphen verwendet. Abbildung 2 Graph C zeigt einen gewichteten, gerichteten Graphen. Das Gewicht der Kante von Knoten 2 zu Knoten 3 beträgt in diesem Fall 7.

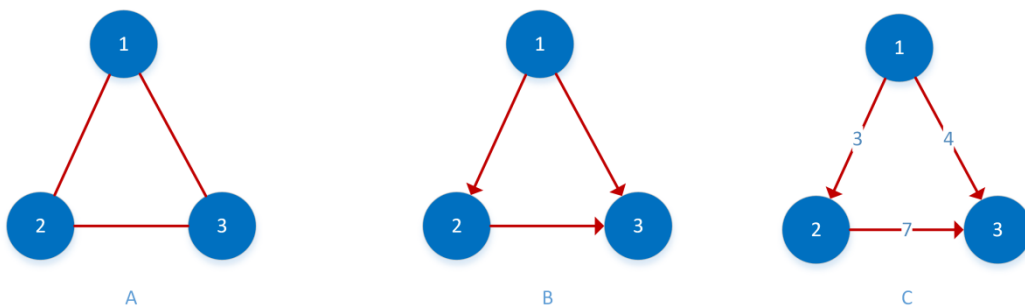


Abbildung 2: Graphentheorie Teil 1

Mehrfach-Kanten beschreiben einen Graph, in welchem eine Kante von Knoten A zu Knoten B mehrfach vorhanden sein kann. Bei gerichteten Graphen gelten Kanten nur als Mehrfach-Kanten, insofern diese in dieselbe Richtung zeigen. Bei den Experimenten dieser Masterarbeit wurden nur Einfach-Kanten verwendet. Dies beruht auch darauf, dass es laut Wikipedia-Regeln nicht erlaubt ist, in einem Artikel mehrfach auf einen bestimmten anderen Artikel zu verlinken. Graphen welche Mehrfach-Kanten besitzen werden Multigraphen genannt. Wird

im weiteren Verlauf dieser Masterarbeit von Graphen gesprochen, so sind immer Graphen mit Einfach-Kanten gemeint. [7, p. 112]

Graphen können zusätzlich noch in „nicht zusammenhängend“, „zusammenhängend“ und „stark zusammenhängend“ eingeteilt werden. Nicht zusammenhängend bedeutet, dass eine Teilmenge von Knoten keine Verbindung zu einer anderen Teilmenge von Knoten besitzt. Im Gegensatz dazu sind in zusammenhängenden Graphen, alle Knoten über Kanten mit allen anderen Knoten verbunden. In gerichteten Graphen kann hier jedoch der Sonderfall eintreten, dass ein Knoten nur über eine gerichtete Kante mit allen anderen verbunden ist. Um diesen Ausnahmefall beschreiben zu können, gibt es für gerichtete Graphen zwei weitere Spezifikationen: schwach- bzw. stark zusammenhängend. Wird ein gerichteter Graph als ungerichteter Graph angesehen und ist dieser zusammenhängend, so ist der zugrundeliegende Graph schwach zusammenhängend. Gibt es wiederum im gerichteten Graphen von jedem Knoten zu jedem anderen Knoten eine gerichtete Verbindung, so ist dieser stark zusammenhängend. Im weiteren Verlauf dieser Arbeit wird auch über den größten, schwach zusammenhängenden und den größten, stark zusammenhängenden Subgraphen gesprochen. Dies sind die jeweils größten Komponenten eines Graphen, bezogen auf die Anzahl der Knoten, für welche die jeweilige Eigenschaft zutrifft. In gerichteten Graphen gibt es zusätzlich noch die Begriffe ausgehende und eingehende Komponente von einem Knoten A. Zur ausgehenden Komponenten von A zählen alle Knoten, welche entlang den gerichteten Kanten erreichbar sind, wohingegen zur eingehenden Komponente all jene Knoten gehören, welche über die entgegengesetzte Richtung der Kanten zu Knoten A verbunden sind. [7, p. 143] In Abbildung 3 ist A ein ungerichteter, zusammenhängender Graph, wohingegen B ein ungerichteter, nicht zusammenhängender Graph ist, da zum Beispiel Knoten 5 ausgehend von Knoten 2 nicht erreicht werden kann. C in derselben Abbildung zeigt einen schwach zusammenhängenden Graph, da Knoten 1 ausgehend von Knoten 3 nicht erreichbar ist, jedoch der zugrundeliegende, ungerichtete Graph zusammenhängend ist. Hingegen ist D stark zusammenhängend, da von jedem Knoten aus jeder andere Knoten im Graph über gerichtete Kanten erreicht werden kann.

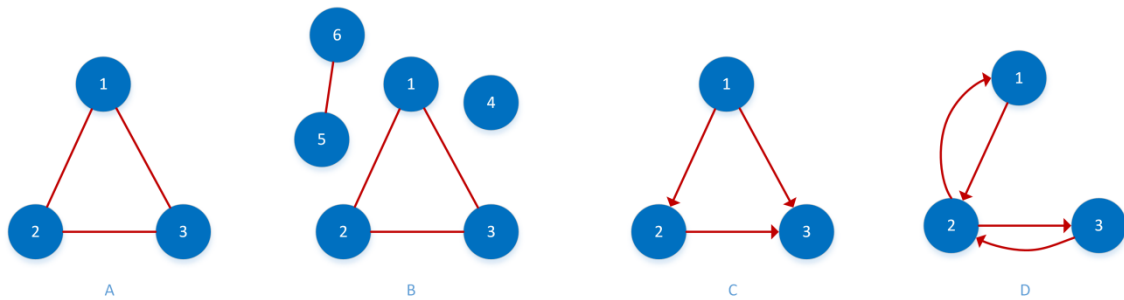


Abbildung 3: Graphentheorie Teil 2

Ein weiterer sehr wichtiger Begriff in der Graphentheorie ist der Kreis, oft auch Zyklus genannt. Dies beschreibt die Möglichkeit von einem ausgehenden Knoten entlang verschiedener Kanten zu diesem zurückzukehren. Bei gerichteten Graphen würde dies bedeuten, dass alle Kanten entlang eines Kreises in dieselbe Richtung zeigen. In Netzwerken sind Kreise oft gewünscht, da diese eine Form der Redundanz bilden. Fällt eine Kante aus, so kann das Ziel noch über andere Kanten erreicht werden. Graphen welche keinen Zyklus beinhalten werden auch als kreisfreie Graphen bezeichnet. [7, p. 118] [8, p. 25] Abbildung 4 Graph A zeigt einen Ausschnitt aus einem Graphen, welcher einen Kreis besitzt. Dieser ist hier farblich gekennzeichnet.

Bäume bilden eine Unterklasse von Graphen, welche spezielle Eigenschaften besitzen. In dieser Subklasse wird wiederum zwischen ungerichteten Bäumen und gerichteten Bäumen differenziert. Wie die Namen schon vermuten lassen, handelt es sich hierbei um Graphen mit entweder gerichteten oder ungerichteten Kanten. Graphen mit ungerichteten Kanten dürfen keine Kreise besitzen, um als Baum zu gelten. Gerichtete Graphen müssen kreisfrei sein und eine Wurzel besitzen, um dieser Unterklasse anzugehören. Eine Wurzel ist ein Knoten, der abhängig von der Richtung der gerichteten Kanten, keine eingehenden (d.h. alle Kanten zeigen von der Wurzel weg) bzw. keine ausgehenden Verbindungen (d.h. alle Kanten zeigen zur Wurzel) besitzen darf. Ein weiterer Sonderfall von Knoten sind sogenannte Blätter. Um als Blatt zu gelten, muss ein Knoten die exakt gegenteilige Anforderung, bezogen auf die eingehenden und ausgehenden Kanten, der Wurzel erfüllen. Besitzt die Wurzel keine eingehenden Kanten, so darf ein Blatt dieses Baumes keine ausgehenden Kanten besitzen und umgekehrt. Viele nicht zusammenhängende Bäume werden als

Wald bezeichnet. Bäume besitzen die Eigenschaft, dass diese immer planar gezeichnet werden können. Dieser Begriff beschreibt die Möglichkeit, den Graphen auf eine 2 dimensionale Fläche zu zeichnen, ohne dabei sich überkreuzende Kanten zu verwenden. Die bekanntesten Beispiele für nicht planare Graphen sind Kuratowski's Graphen K_5 und $K_{3,3}$. [7, p. 127] In Abbildung 4 ist Graph B ein Baum, insofern die grau gekennzeichnete Kante nicht existiert. Graph C zeigt einen gerichteten Baum mit Wurzel 1 und den Blättern 4,5 und 6.

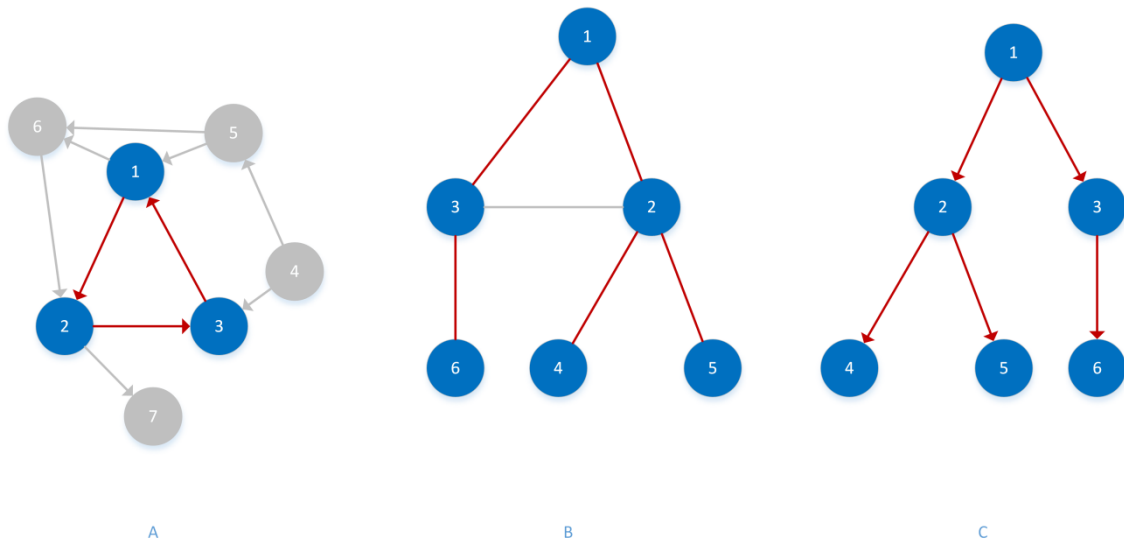


Abbildung 4: Graphentheorie Teil 3

2.1.1 Knoteneigenschaften

Wird in der Graphentheorie über Nachbarschaft gesprochen, so sind damit alle durch eine Kante verbundenen Knoten eines Knotens gemeint. Es wird zwischen offener und geschlossener Nachbarschaft unterschieden, wobei die offene Nachbarschaft im Gegensatz zur geschlossenen Nachbarschaft den ausgehenden Knoten nicht beinhaltet. Eine Erweiterung der Nachbarschaft sind N-Nachbarschaften. Ein Knoten ist in der N-Nachbarschaft eines anderen Knotens, wenn er zu diesem über maximal N Kanten verbunden ist. Graph A in Abbildung 5 zeigt die Nachbarschaften des rot markierten Knotens. Alle blau markierten Knoten sind in der Nachbarschaft, während blau und grün markierte Knoten zusammengefasst die 2er-Nachbarschaft bilden. Grau gekennzeichnete Knoten wären erst in der 3er-Nachbarschaft beinhaltet.

Der Grad eines Knoten gibt an wie viele Kanten dieser besitzt. Im Falle von gerichteten Graphen, wird hier noch zwischen Eingangsgrad und Ausgangsgrad unterschieden, wobei der Eingangsgrad die Anzahl aller eingehenden Kanten ist und der Ausgangsgrad die Anzahl aller ausgehenden Kanten. Knoten mit hohem Grad werden Hubs genannt, da diese zentrale Verteilerknoten darstellen. Oft wird auch der Eingangsgrad in Relation zum Ausgangsgrad gesetzt, um einen besseren Messwert für die Wichtigkeit eines Knotens zu erhalten. Dies wird in diesem Kapitel im Unterpunkt Hierarchie detaillierter beschrieben. In Abbildung 5 Graph B, besitzt Knoten 1 den Grad 4. Eingangsgrad des Knotens ist 1 und Ausgangsgrad 3.

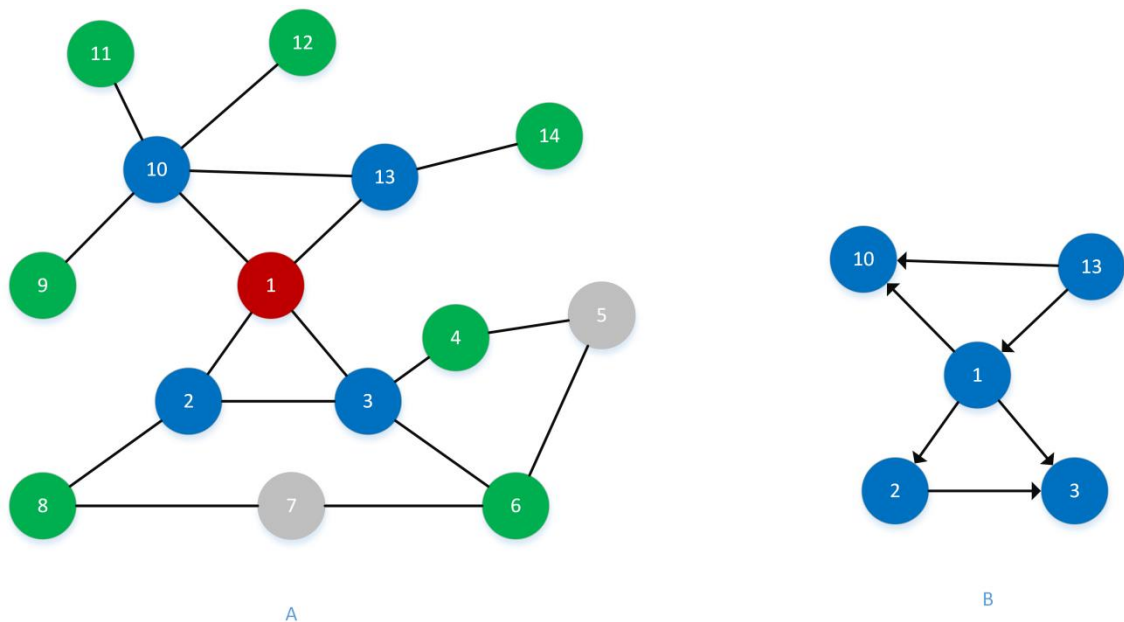


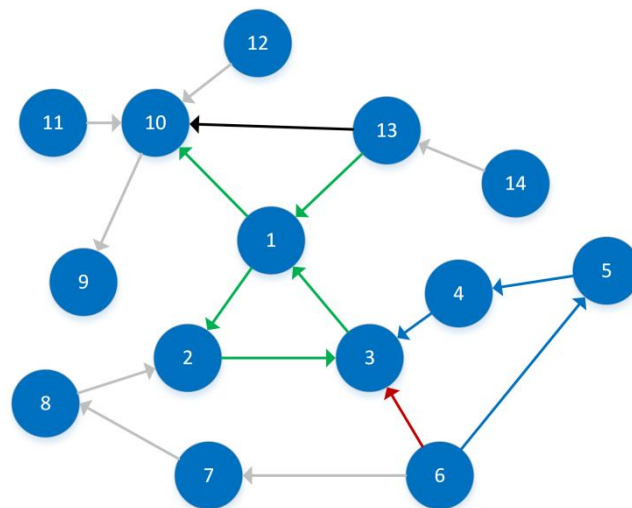
Abbildung 5: Graphentheorie Teil 4

2.1.2 Wege und Pfade

Ein Weg ist eine Folge von Knoten innerhalb eines Graphen. Die Distanz eines Weges ist die Anzahl der besuchten Kanten. Enthält der Weg keinen Kreis, so sollte die Distanz gleich der Anzahl von besuchten Knoten weniger 1 entsprechen. Diese besondere Art von Wegen werden auch Pfade oder selbstvermeidende Wege genannt. [7, p. 136] In Abbildung 6 ist durch grün gekennzeichnete Kanten ein Weg von Knoten 13 nach Knoten 10 dargestellt, da dieser den

Knoten 1 zweimal beinhaltet. Blau markierte Kanten bilden einen Pfad von Knoten 6 nach Knoten 3.

Der Pfad mit der kleinsten Distanz zwischen zwei verschiedenen Knoten, wird der kürzeste Pfad genannt. Die Distanz dieses Pfades wiederum ist die kürzeste Distanz zwischen dem Knotenpaar AB. Um dies berechnen zu können, wird meistens ein modifizierter Breiten-Suche-Algorithmus, z.B.: Dijkstra, ausgehend vom ersten Knoten, verwendet. Bei ungerichteten Graphen ist die Distanz von Knoten A nach B gleich der Distanz von B nach A. In gerichteten Graphen können sich diese jedoch unterscheiden, da Kanten von A nach B oft nicht in die entgegengesetzte Richtung existieren. [7, p. 139] In Abbildung 6 wurde der kürzeste Pfad zu Knoten 10 ausgehend von Knoten 13 durch eine schwarze Kante gekennzeichnet. Ebenso zeigt die rote gefärbte Kante den kürzesten Pfad von 6 nach 3.



A

Abbildung 6: Graphentheorie Teil 5

Als sogenannte „Random-Walks“ werden Wege bezeichnet, welche durch wiederholtes, zufälliges Wählen eines Nachbarknotens entstanden sind.

2.2 Hierarchien

Eine Hierarchie eines Graphen ist eine Teilmenge von Knoten des Graphen, welche einen Baum bilden. Häufig wird dafür ein Hub-Knoten als Wurzel verwendet. Die Grundidee hinter Hierarchien ist eine hierarchische Einteilung der Knoten des Graphen. Hierzu wird meist eine Breitensuche oder eine Sortierung nach Wichtigkeit der Knoten eingesetzt. Für letzteres wird ein hierarchischer Wert für alle Knoten berechnet.

Dieser ergibt sich aus $\frac{\text{Eingangsgrad}}{\text{Ausgangsgrad}} * \sqrt{\text{Eingangsgrad}}$, auch bekannt als Hierarchical-Score. Die Wurzel am Ende der Formel stellt sicher, dass Knoten mit Eingangsgrad 2 und Ausgangsgrad 3, einen niedrigeren Wert erhalten als Knoten mit Eingangsgrad 200 und Ausgangsgrad 300. Obwohl dies im Vergleich zu PageRank [9] oder HITS [10] eine relativ simple Formel darstellt, sind die Ergebnisse durchaus überzeugend und gut weiterverwendbar. [11] Details zum Aufbau der verwendeten Hierarchien in den Experimenten, finden sich in [6].

2.3 Navigation in Graphen

Durch die Tatsache, dass sich viele reale Gegebenheiten, wie zum Beispiel Landkarten, durch Graphen repräsentieren lassen, können auch viele damit verbundenen Aktionen in Graphen simuliert werden. Ein Beispiel dafür ist die Navigation. Täglich wird diese von Autofahrern benutzt um von A nach B zu kommen. Dabei wird die Landkarte durch einen Graphen repräsentiert und dem Benutzer durch einen modifizierten Breiten-Suche-Algorithmus, der kürzeste Weg von seinem Standort aus zum Ziel beschrieben. Dies ist jedoch bei weitem nicht die einzige Möglichkeit um Navigationen des alltäglichen Lebens in Graphenform darzustellen. Fast jeder von uns navigiert sich täglich durch das World-Wide-Web um an Informationen zu gelangen. Meist benutzen wir dabei anfangs eine Suchmaschine, um eine Webseite zu finden, welche möglicherweise unsere gesuchte Information beinhaltet. Sind wir dann auf dieser Webseite, folgen wir häufig Links zu Unterseiten, um das gesuchte Ziel zu erreichen. Exakt dieser Vorgang kann auf einfache Weise in einem Graph dargestellt

werden. Dabei werden die Webseiten als Knoten und die Links von einer Seite zur nächsten als gerichtete Kanten zwischen den beiden Knoten angesehen. Menschen haben bei dieser Navigation bestimmte Vorgehensweisen, welche im Kapitel „Related Work“ genauer, von bereits durchgeführten Studien und Versuchen, beschrieben werden.

2.4 Decentralized Search

Hat der Mensch einmal einen Link aus den Ergebnissen der Suchmaschine angeklickt, so betreibt er anschließend „Decentralized Search“. Dies bedeutet, dass er nur jene Links, ausgehend von der Seite auf welcher er sich befindet, sieht, jedoch nicht das gesamte Netzwerk, bzw. den gesamten Graphen in welchem er navigiert, kennt. Anhand des verlinkten Textes der Seite und seinem Hintergrundwissen, entscheidet er anschließend, welchem er weiter folgen wird, um an seine gesuchte Information zu gelangen. Oder aber, ob er im Browser einen Schritt zurückgeht, weil er nicht glaubt, dass einer der ausgehenden Links seinen Wissensbedarf stillen wird.

2.4.1 Hintergrundwissen

Zahlreiche Studien belegen, dass Menschen in stark verbundenen Netzwerken mittels „Decentralized Search“ sehr gut kurze Verbindungen zwischen zwei Knoten finden. [12] [11] Dazu verwenden sie meist ihr Hintergrundwissen. Navigiert der Mensch zum Beispiel auf Wikipedia von einem Artikel über Kalifornien zu einem über Mozart, so wird er anfangs überlegen, zu welchen Kategorien der Zielartikel gehört. Da in solchen Netzwerken oft anhand von geographischen Details navigiert wird, ist der erste Schritt die Zuordnung des Artikels zu einem geographischen Standort, in diesem Fall Österreich. Nun kann versucht werden über diverse Länderartikel, ausgehend von Kalifornien, den Artikel über Österreich zu finden. Meist wird dieser Weg über den Artikel „United States of America“ führen. Das Hintergrundwissen der Menschen ist dabei die Zuordnung von Mozart zu Österreich, als auch die Kategorisierung von Kalifornien zu Amerika.

Digital kann das Hintergrundwissen auch als Hierarchie dargestellt werden. Dabei wäre der Elternknoten von Kalifornien der Artikel „United States of America“. Dieser wiederum wäre ein Kind des Artikels „Welt“, welcher in diesem Themengebiet auch die Wurzel bildet, insofern das restliche Universum nicht betrachtet wird.

Durch dieses Hintergrundwissen besitzen die Menschen eine grobe Struktur des Netzwerkes und können anhand dieser offensichtlich enorm effizient navigieren. Des Weiteren lernen sie bei jeder Navigation die Struktur des zugrundeliegenden Netzwerkes besser kennen und verbessern sich dadurch ständig.

2.4.2 Greedy-Algorithmen

Um diesen Vorgang simulieren zu können, wurden Greedy-Algorithmen entwickelt. Aus dem Englischen übersetzt bedeutet greedy gierig, was genau die Vorgehensweise dieser Art von Algorithmen beschreibt, da dieser immer ausgehend von seiner aktuellen Position bzw. Situation, das für seine Zwecke beste, nächste Ziel auswählt, ohne einen Gesamtüberblick über das System zu haben. Durch diese Verfahrensweise können „Greedy-Algorithmen“ meist nur lokale Minima finden. Jedoch eignen sie sich gut, um den Grundgedanken des Menschen zu simulieren, da dieser sicherlich den Link auswählt, von welchem er am ehesten glaubt seinem Ziel näher zu kommen.

2.5 Wikipedia

Das Wort Wikipedia ist eine Kombination aus dem hawaiianischen Wort „wiki“, was übersetzt „schnell“ bedeutet, und dem Wort „Enzyklopädie“. Die Wikipedia ist eine kollaborative, online Enzyklopädie, welche derzeit in über 280 Sprachen verfügbar ist. Sie wurde am 15. Jänner 2001 von Jimmy Wales und Larry Sanger gegründet und erfreut sich seither immer größer werdender Beliebtheit. [2] Die größte Sprachversion ist bis dato die englische Version. Sie besitzt über 4.110.000 Artikel und hat somit einen großen Abstand zur zweitplazierten deutschen Version mit 1.510.000 Artikel. Heutzutage ist sie das schnelle Nachschlagewerk schlechthin. Ob vom Notebook oder Handy aus, Leute benutzen

die online Enzyklopädie Wikipedia täglich, um schnell Informationen zu einem gewünschten Wissensgebiet zu beziehen. [13] [14]

The screenshot shows the Wikipedia article for "Technische Universität Graz". The page layout includes a top navigation bar with "Artikel" and "Diskussion" tabs, and a search bar. The main content area features the article title, a brief history, and a table of contents. A sidebar on the left contains navigation options like "Hauptseite", "Themenportale", and "Mithraschen". A summary box on the right provides key statistics:

Technische Universität Graz Erzherzog Johann Universität	
Gründung	1811 (als Technische Lehranstalt)
Trägerschaft	staatlich
Ort	Graz, Österreich
Rektor	Harald Kainz
Studenten	12630 (Sommersemester 2012) - davon Frauen: 21,84 % - Ausländeranteil: 17,64 %
Mitarbeiter	2256 (31. Dezember 2011) - davon wissenschaftliches Personal: 1402 - davon nichtwissenschaftliches Personal: 854
Jahresetat	170,2 Mio. Euro Öffentliche Mittel: 113 Mio. Euro (Bundesbudget 2011) Drittmittel: 57,2 Mio. Euro (2010)
Website	www.tugraz.at

Abbildung 7: Wikipedia-Artikel der Technischen Universität Graz

Die Grundidee hinter Wikipedia ist ein freies Online-Lexikon, in welchem jeder Benutzer sein Wissen über bestimmte Themengebiete teilen und mit Wissen von anderen Benutzern, zu einem Artikel zusammenfügen kann. Dies klingt, wenn man es das erste Mal hört, gut, bringt jedoch auch einige Nachteile mit sich, wie später aufgezeigt wird.

Die Wikipedia beruht auf einem speziellen Hypertext-System, welches den Benutzern nicht nur erlaubt, online Inhalte zu lesen, sondern ihnen auch ermöglicht, diese direkt in ihren Web-Browsern zu bearbeiten. Diese Art von Web2.0 Systemen werden Wiki-Systeme genannt. Um diese möglichst einfach und benutzerfreundlich zu gestalten, wird beim Editieren der Seiten kein HTML-Quellcode angezeigt, sondern eine Auszeichnungssprache verwendet. Links werden hier zum Beispiel wie folgt erstellt: [[Wikimedia|Wikimedia Foundation, Inc.]]. Der erste Teil nach den doppelten, offenen, eckigen Klammern ist der Name des Artikel, auf welchen der Link zeigen soll. Der zweite Teil nach dem „|“ Symbol ist der Text, welcher im Artikel dann verlinkt wird. Mit dieser einfachen Syntax ist sichergestellt, dass auch Leute ohne HTML-Kenntnisse Artikel erstellen und bearbeiten können.

Wikipedia benutzt das MediaWiki-System. Dies ist ein Wiki-System, welches auf einer Kombination aus PHP¹ und MySQL² beruht. MediaWiki ist eine freie Software, welche unter der „GNU General Public License³“ lizenziert ist. Ein Vorteil dieses Systems ist die Redundanz. Verändert ein Benutzer eine Seite, so wird diese direkt in der Datenbank gespeichert ohne jedoch die alte Version zu überschreiben. Somit können falsch editierte Seiten, sei dies nun absichtlich oder unabsichtlich geschehen, einfach wieder zurückgesetzt werden. [15]

Ein Artikel ist laut Wikipedia eine Seite welche enzyklopädische Information beinhaltet. Somit zählen Redirects, d.h. Seiten welche nur die Aufgabe haben, auf andere Artikel weiterzuleiten, nicht als Artikel. Der typische Aufbau eines Wikipedia Artikels ist in Abbildung 7 zu sehen. Dieser besteht meist aus einer Überschrift, einem kurzen Absatz, welcher allgemein den Artikel beschreibt, gefolgt von einer Übersicht bzw. einem Inhaltsverzeichnis des Artikels. Anschließend stehen die einzelnen Kapitel des Inhaltsverzeichnisses. Auf der rechten Seite befindet sich bei manchen Artikeln eine Info-Box, in welcher sich grundlegende Informationen befinden. Bei Artikeln über Städte besteht diese zum Beispiel aus Lage, Land, Einwohnerzahl, Bürgermeister, Zeitzone, Postleitzahl und weiteren signifikanten Angaben für Städte.

Artikel werden auf Wikipedia in den Namespace „main“ bzw. „article“ einsortiert. Insgesamt gibt es 12 verschiedene Basis-Namespaces:

- Main: Wie schon erwähnt befinden sich hier alle Artikel und Weiterleitungen, sogenannte Redirects, auf Artikel.
- User: Hier befinden sich Seiten, welche von Benutzern für den persönlichen Gebrauch erstellt worden sind.

¹ PHP ist eine Skriptsprache, welche sich einfach in HTML einbetten lässt. Rekursives Akronym „Hypertext Preprocessor“

² MySQL ist ein open-source, relationales Datenbanksystem.

³ GPL ist eine weit verbreitete Lizenzform für freie Software. Sie erlaubt es jedem diese Software zu nutzen, zu studieren, zu verbreiten und zu ändern. Abgeänderte Software darf jedoch nur unter Einhaltung der ursprünglichen Lizenz weiterverbreitet werden.

- **Project:** In diesem Namespace befinden sich auf das Projekt Wikipedia bezogene Seiten, wie zum Beispiel Diskussionsseiten und Regelseiten.
- **File:** Dieses Präfix beschreibt Seiten, welche für Bildbeschreibungen gedacht sind. Jedoch sind in diesem Bereich nur die Bildbeschreibungen und nicht die Bilddateien selbst gespeichert.
- **MediaWiki:** Dieser Namensraum beinhaltet sogenannte Interface-Texte. Diese erscheinen auf automatisch generierten Wikipedia-Seiten.
- **Template:** Seiten mit diesem Präfix beinhalten Vorlagen wie zum Beispiel die Vorlage für eine Infobox.
- **Help:** Wie der Name schon sagt, befindet sich hier die Hilfestellung zur Benutzung von Wikipedia. Es wird hier Hilfe für normale Benutzer als auch für Editoren angeboten.
- **Category:** Hier werden die einzelnen Kategorien und oft eine kurze Erklärung dazu gespeichert.
- **Portal:** Portale bilden eine Sammlung von Artikeln, welche sich auf ein bestimmtes Themengebiet beziehen. Dies soll den Lesern ermöglichen, neue Artikel zu ihrem Themengebiet auf einfache Weise zu finden.
- **Book:** Dieser Bereich bildet eine Sammlung über viele Artikel, welche zusammen ein Buch ergeben. Diese Bücher können auch beispielsweise im PDF-Format heruntergeladen werden.
- **Education Program:** In diesem Namespace können spezielle Kurs-Seiten von Professoren oder Lehrern angelegt werden. Auf diesen Seiten können dann Studenten bzw. Schüler arbeiten, während ihre Professoren bzw. Lehrer den Überblick über die erbrachten Arbeit behalten.
- **TimedText:** Seiten mit diesem Präfix beinhalten synchronisierte Untertitel für diverse Mediendateien.

Im Hauptnamespace bzw. Artikelnamespace werden die einzelnen Artikel zusätzlich in 12 Haupt-Kategorien eingeteilt:

- Allgemeine Referenzen
- Kultur und Kunst
- Geographie und Orte
- Gesundheit und Fitness
- Geschichte und Events
- Mathematik und Logik
- Naturwissenschaften und Physik
- Leute
- Philosophie und Gedankenwissenschaften
- Religionen und Glaubenssysteme
- Gesellschaft und Gesellschaftsforschung
- Technologie und angewandte Wissenschaften

Diese Kategorien sind jedoch nicht gegenseitig ausschließend. Das heißt, ein Artikel kann sowohl zur Kategorie Geschichte als auch zur Kategorie Leute gehören, wie es zum Beispiel bei Biographien historischer Persönlichkeiten der Fall ist.

Leider bringt die kollaborative, online Enzyklopädie auch einige Nachteile mit sich. So ist es zum Beispiel jedem möglich, Artikel zu erstellen, oder vorhandene zu editieren. Die Idee von Wikipedia dahinter ist jedoch, dass es ausreichend Nutzer gibt, welche sich in den jeweiligen Themengebieten auskennen und falsch geschriebene Artikel korrigieren oder melden. Doch wie ein vor kurzem in den Medien berichteter Fall zeigt, funktioniert dies nicht zu 100%. Laut diverser Nachrichtenseiten [16] [17], stand ein komplett erfundener Krieg,

der „Bicholim-Konflikt“, ein Krieg zwischen Portugal und dem indischen Reich, über fünf Jahre lang auf Wikipedia. Seit 8. März 2006 führt Wikipedia sogar eine eigene Liste mit gefälschten bzw. frei erfundenen Artikeln, wobei diese wiederum auch von jedermann editiert werden kann. [18] Dies ist der Grund, wieso Wikipedia nicht als wissenschaftliche Quelle benutzt werden soll.

2.6 Wikigame

Der Name Wikigame spiegelt mittlerweile eine Kategorie von Spielen, welche sich als Grundlage den Datensatz eines Wiki-Systems nimmt und darauf basierend Spieler von definierten Start-Artikeln zu definierten Ziel-Artikeln navigieren lässt. Es gehört in die Kategorie der Online-Multiplayer-Games. Es gibt zurzeit 3 bekannte Versionen eines solchen Spieles:

- Wikispeedia [19]
- WikipediaMaze [20]
- The Wikigame [21]

Da bei den Experimenten dieser Arbeit ausschließlich Daten aus „The Wikigame“ verwendet wurden, wird hier auch nur dieses genauer beschrieben.

The WikiGame wurde von Alex Clemesha, einem Ocean- und Software-Enthusiasten aus San Diego entwickelt. [22] Er verwendete dazu diverse „open-source“⁴ Technologien:

- Python⁵
- Django⁶
- Redis⁷

⁴ Open-Source heißt, dass der Quellcode für jedermann offen zugänglich ist und sich jeder an dem Projekt beteiligen kann.

⁵ Höhere Programmiersprache mit Betonung auf lesbaren Programmcode. <http://python.org/>

⁶ Ein Python-Web-Framework für effiziente, leicht lesbare Webapplikationen. <https://www.djangoproject.com/>

- XMPP⁸ (mit Strophe)⁹

Das Spielprinzip gleicht dem aller Wikigames: Viele Spieler befinden sich zur selben Zeit auf der Webseite und treten gegeneinander an. Alle Spieler bekommen zur selben Zeit denselben Artikel, welcher den Start-Artikel bildet. Unter bestimmten Bedingungen müssen sie einen anderen, vordefinierten Artikel erreichen. Als Grundlage dient hierfür die englische Version von Wikipedia. Dabei gibt es bei „The Wikigame“ 5 verschiedene Spielarten.

Die beliebteste davon ist das Speed-Race, bei welchem es vor allem um die Zeit geht. Dabei bekommen alle aktuellen Mitspieler zur selben Zeit einen Start- und einen Ziel-Artikel. Ein solches Start-Ziel-Artikel-Paar wird oft auch Mission genannt. Alle Spieler haben von da an knapp zwei Minuten Zeit, um sich via Anklicken von Links, zu dem Ziel-Artikel zu navigieren. Gewinner der Runde ist derjenige, der es in der kürzesten Zeit geschafft hat, das Ziel zu erreichen. Dabei steht den Spielern jedoch nicht die gesamte Wikipedia-Seite, wie man sie von Wikipedia selbst kennt, zur Verfügung, da ansonsten wahrscheinlich alle das Suchfeld verwenden würden. Eine detaillierte Beschreibung der Einschränkungen befindet sich im Unterkapitel „Einschränkungen“.

The screenshot shows the 'The Wikigame' interface. At the top left is the logo 'THE WIKI GAME'. Below it, a timer indicates '32s remaining to find goal!' and the player 'Albatross' is listed. A table shows other players and their click counts. The main content is a Wikipedia article for 'Jimmy Carter', which is truncated. A sidebar on the right features a photo of Jimmy Carter and his biographical details, including his role as the 39th President of the United States and 76th Governor of Georgia.

Abbildung 8: Wikigame

⁷ Ein open-source Daten-Struktur-Server. <http://redis.io/>

⁸ Ein offenes Protokoll für Echtzeit-Kommunikation. <http://xmpp.org>

⁹ Ein JavaScript Aufsatz für XAMPP um dieses an das HTTP-Protokoll zu binden.

Eine andere Spielvariante ist die „Least-Clicks“-Version. Auch hier bekommen alle Mitspieler dieselbe Mission. Jeder hat wieder zwei Minuten Zeit um die gestellte Spielaufgabe zu erfüllen, jedoch gewinnt in dieser Variante jener Spieler, welcher die wenigstens Klicks dafür benötigt. Leider wird hier auch das Aktualisieren einer Seite, falls diese sich nicht vollständig lädt, als Klick gewertet. Auch Schummeln ist durch fehlende Sicherheitsüberprüfungen möglich, womit mittels eines Klicks direkt zum Ziel-Artikel gesprungen werden kann. Genaueres dazu findet sich in [6].

Die dritte Spielart nennt sich „6 Degrees of Wikipedia“, was sicherlich eine Anspielung auf das bekannte „Small-World-Phänomen“ bzw. „kleine-Welt-Phänomen“ von Milgram ist. Dieses Experiment wird im Kapitel „Related Work“ detailliert beschrieben. Die Kernaussage davon ist jedoch, dass sich in stark-verbundenen Netzwerken wie Wikipedia, die meisten Knoten-Paare mit 6 oder weniger Kanten verbinden lassen. Genau um diese Aussage geht es in dieser Spielversion. Wie immer bekommen alle Spieler dieselbe Mission und zwei Minuten Zeit um diese zu erfüllen. Als Gewinner geht hier jedoch derjenige hervor, dessen Anzahl der benötigten Klicks zum Erfüllen der Mission den geringsten Abstand zu 6 hat.

Bei der vorletzten Spielart handelt es sich um „5 clicks to Jesus“. Dabei bekommen alle Spieler denselben Start-Artikel zugeteilt und wie in allen Spielen auch wieder zwei Minuten Zeit. Diese Version ähnelt der „6 Degrees of Wikipedia“-Spielart sehr, da auch hier derjenige gewinnt, dessen Anzahl der benötigten Links am nächsten zu einer vorgegebenen Zahl ist. Im Gegensatz zu der vorherigen Art ist es hier jedoch 5 anstatt 6. Dies kommt daher, dass der Artikel „Jesus“ ein sehr häufig verlinkter Artikel ist und somit mit weniger als 6 Klicks erreichbar sein sollte.

Last but not least gibt es den Spieletyp „No united States“. Wie später in Kapitel „Related Work“ beschrieben wird, navigieren viele Leute anfangs anhand von geographischen Details. So werden die meisten auf der Suche nach Mozart anfangs wahrscheinlich versuchen, auf die Artikel-Seite von Österreich zu gelangen. Da die Seite „United States“ auf fast alle Länder dieser Welt verlinkt, ist diese ein beliebtes Zwischenziel in vielen Missionen. Genau dies wird jedoch

bei dieser Spielvariante verboten, um den Spielern die einzelnen Aufgaben zu erschweren. Gewonnen hat in diesem Spiel derjenige, der am wenigsten Klicks benötigte.

Das Projekt „The Wikigame“ besitzt seit 21.11.2011 auch eine eigene iPhone-App, welche derzeit um 0.89€ erworben werden kann. Diese stellt zusätzlich zu den bekannten Online-Spielvarianten noch Levels zur Verfügung, in welchen gegen sich selbst gespielt wird bzw. vorgegebene Zeiten oder Klicks unterboten werden müssen. [23]

2.6.1 *Community*

TheWikigame.com bietet Benutzern auch an, eigene Benutzerprofile zu erstellen. Dadurch kann statt dem Standard-Username „Guest-<zufällige-Buchstaben-Zahlen-Kombination>“ ein eigener Benutzername ausgewählt werden, insofern dieser noch nicht vergeben ist. Ein eigener Benutzername bringt viele Vorteile mit sich, wie zum Beispiel die Einsicht in die eigene Statistik über bereits gespielte Missionen. Dort werden alle bisher erfolgreich gespielten Spiele inklusive Platzierungen und die jeweiligen Start- und Ziel-Artikel mit ihren dazugehörigen gewählten Wegen angezeigt. Einen kleinen Zusatzanreiz bietet über dies hinaus die Tages- und Wochenstatistik. Dort werden automatisch alle Top-Spieler angezeigt. Hierfür gibt es ein Ranking nach Punkten, welche mittels Spielaufgaben erspielt werden können. Pro Spielrunde sieht das Punkteschema wie folgt aus:

- 1.: 1000 Punkte
- 2.: 500 Punkte
- 3.: 300 Punkte
- 4.: 200 Punkte
- 5.: 150 Punkte
- 6.-10.: 100 Punkte

- 11. - Letzter: 50 Punkte

Spieler erhalten nur Punkte, wenn diese eine Spielrunde auch erfolgreich abgeschlossen haben. Trifft dies zu, entscheidet die Platzierung über die Anzahl der Punkte, wobei auch der Letztplatzierte immerhin 50 Punkte erhält. Des Weiteren werden die Punkte pro Spielkategorie getrennt gesammelt, woraus sich auch pro Spielvariante zwei separate Highscore-Listen, die Tages-Highscore und die Wochen-Highscore, ergeben. Abbildung 9 zeigt die Ergebnisse des letzten Spiels eines Wikigames. Die Benutzernamen wurden aus Datenschutzgründen unkenntlich gemacht.









LAST GAME RESULTS	TODAY'S LEADERS	WEEK'S LEADERS
#1	  with 4 clicks in 37s. Pyrolysis → Greek language → Classical Athens → Architecture	
#2	  with 4 clicks in 53s. Pyrolysis → Fire protection engineering → Architects → Architecture	
#3	  with 8 clicks in 84s. Pyrolysis → Morpheme → Pyrolysis → Greek language → Ancient Greek → Ancient Greek architecture → History of architecture → Architecture	
#4	 with 7 clicks in 84s. Pyrolysis → Greek language → France → Atlantic Ocean → United States → Architecture of the United States → Architecture	
#5	 with 9 clicks in 104s. Pyrolysis → Volcanic eruption → Volcanologists → Roman mythology → Myth → Roman mythology → Roman Empire → Ancient Roman architecture → Architecture	

Abbildung 9: Wikigame Statistik

Um auch die Unterhaltung unter den Spielern zu fördern, existiert bei jeder Spielart am linken unteren Browserrand ein kleiner Chat. Ob dieser jedoch die Benutzer zusammenbringt ist fraglich, da viele nicht registrierte Benutzer dort mit Schimpfwörtern um sich werfen. Auch gibt es die Möglichkeit, Bekannte und Freunde über einen „invite friends button“ einzuladen. Dieser bietet die Optionen „Einladung via Mail“, „auf Facebook veröffentlichen“ und „auf Twitter veröffentlichen“ an.

2.6.2 *Einschränkungen*

Um den Spielern des Wikigames die Navigation zwischen verschiedenen Artikel-Paaren zu erschweren, wurde die zugrundeliegende Wikipedia beschränkt.

Die größte und vor allem wichtigste Einschränkung dabei ist sicherlich das Entfernen der Suchoption bzw. des Suchfeldes. Würde dies den Nutzern zur Verfügung stehen, so wäre das Spiel nicht spielenswert, da jeder per Eingabe des Ziel-Artikels in das Suchfeld, sofort die Mission mit nur einem Klick beenden könnte. Summa summarum wäre es weder ein sinnvolles noch belustigendes Spiel, wenn es darum ginge, wer von all den Spielern am schnellsten die Suchoption bedienen könnte.

Auch eine Art von Einschränkung ist der verwendete Datensatz. Dieser basiert auf MSQl-Dumps der offiziellen Wikipedia, jedoch wurden einige Links und Felder entfernt. Des Öfteren fehlt die Infobox und die Links zu Diskussionsseiten sind entweder gar nicht sichtbar oder zumindest nicht verlinkt. Dies wäre jedoch wünschenswert, da sich die Spieler ausschließlich über Artikel zum Ziel navigieren sollten. Leider fiel während der Experimente jedoch auf, dass nicht alle Links zu Diskussionsseiten ausrangiert wurden. Eine genaue Analyse zu diesem Fall und wie die Daten für die durchgeführten Versuche bereinigt wurden findet sich in [6].

Eine weitere Abänderung zur originalen Wikipedia-Version ist die Formatierung. Die Texte sind in einer eindeutig größeren Schriftart dargestellt, was die Übersichtlichkeit merklich negativ beeinträchtigt. Außerdem wird dadurch auch spürbar die Scrollarbeit erhöht, da sich des Öfteren die gesuchten Links am Ende der Seite befinden.

3. Related Work

Dieses Kapitel befasst sich mit bereits durchgeführten Experimenten und Forschungsergebnissen anderer Wissenschaftler bezogen auf dieses Themengebiet.

3.1 Benutzerverhalten im Internet

Internetuser sind großteils auf der Suche nach Informationen von verschiedenen Webseiten. Um ihren Wissensdurst zu stillen, gibt es verschiedenste Varianten, um sich durch das World-Wide-Web zu navigieren. Die größte Unterteilung liegt hier zwischen „Suchanfragen“ und „Links-folgen“. Bei Suchanfragen gibt der User meist Stichwörter, passend zu seiner gesuchten Information, in eine der vielen Suchmaschinen im Web ein und erhält anschließend eine Liste mit Links. Diese verweisen auf diverse Homepages, welche mehr oder weniger Wissen zum gesuchten Themengebiet beinhalten. Hat der Benutzer erst einmal einen Link ausgewählt, wird er, insofern die verlinkte Seite grob sein Themengebiet abdeckt, weiteren Links auf dieser Webseite folgen, bis er all seine gesuchten Information gefunden hat.

Die erste Studie zu diesem Themenbereich ist von Huberman im Jahre 1998 erstellt worden. [24] Seine Versuche zeigten, dass sich die Wahrscheinlichkeit, einem Link zu folgen, mit jeder weiteren besuchten Seite entsprechend eines Potenzgesetzes, verringert.

Eine Studie aus 2004 von Teevan [3] zeigt, dass sehr wenige Internetuser sich mit den ersten Suchergebnissen einer Suchmaschine zufrieden geben. Der Großteil in der Studie tätigt eine Suchanfrage, durchsucht einige der Ergebnislinks und kehrt anschließend zur Suchmaschine zurück um seine Suchanfrage genauer zu definieren.

Genau dieses Vorgehen bestätigt auch ein groß angelegtes Experiment aus dem Jahr 2010. [4] Dazu wurden eine Woche lang Log-Daten einer Microsoft-Browser-Toolbar gesammelt und anschließend ausgewertet. Dabei wurde jeder unverwechselbare Computer als 1 Benutzer angesehen, da die Forscher der

Meinung waren, dass Computer welche von mehr als nur einem User benutzt werden, relativ gering in der Anzahl sind. Außerdem wurden nur jene Aufzeichnungen betrachtet, bei welchen User auch wirklich einen Klick getätigt hatten. Hier wurde gezeigt, dass User mit einer Wahrscheinlichkeit von 60%-72,5% pro Klick einem Link auf der jeweiligen Seite folgen. Dies zeigt einen deutlichen Unterschied zu den ursprünglich angenommenen 85% von Page und Brin im Jahre 1998. [9]

Ein weiteres, sehr interessantes Versuchsmodell zu diesem Themengebiet war der sogenannte SNIF-ACT. [25] Dabei wurde ein kognitives Modell erstellt. Es wurde belegt, dass Internetuser sequentiell Links betrachten und sich pro Link zwischen „Link anklicken“ oder „Link nichtanklicken“ entscheiden. Ist der Benutzer, ohne einen passenden Link gefunden zu haben am Ende der Internetseite angelangt, so geht er zurück. Der Versuch basierte auf dem „Base- Satisficing“ Modell, welches Entscheidungen anhand bereits getroffener Entschlüsse fällt. Dabei wird aus allen bisherigen Benutzerentscheidungen ein „Information-Scent“ (-engl.: Informations-Duft bzw. Duftnote) berechnet. Anschließend wurde versucht, aus diesen berechneten Informationen Benutzerentscheidungen vorherzusagen. Später wurde das Modell durch Berücksichtigung der Link-Positionen auf den einzelnen Websites erweitert. Außerdem wurde ein Bayesian-Satisficing Mechanismus eingefügt, welcher berücksichtigt, dass User nicht unbedingt den lukrativsten Link einer Homepage wählen, sondern oft dem ersten, annäherungsweise passenden Link folgen. Im Vergleich des Modells mit gesammelten Daten einer Studie, in welcher 74 Benutzer Aufgaben im Internet erfüllen mussten, schnitt der erweiterte SNIF-ACT beachtlich gut ab. Die Wahrscheinlichkeit für eine richtige Vorhersage lag zwischen 64% und 91%.

3.2 Benutzerverhalten in Wikipedia-Netzwerken

Dieses Unterkapitel befasst sich mit dem Benutzerverhalten in Wikipedia-Netzwerken, wobei anfangs bisherige wissenschaftliche Analysen über derartige Netzwerke vorgestellt werden.

3.2.1 *Wikipedia Netzwerke*

2005 versuchte Jakob Voss die Wikipedia zu „vermessen“. [26] Neben dem enormen Wachstum an Artikeln und Autoren, beschäftigte er sich auch mit den Verlinkungen von Artikeln und mit den Links, welche ein Artikel beinhaltet. Die Diagramme dieser Arbeiten zeigten genau die Überlegenheit an Verlinkungen eines Artikels, im Vergleich zu dessen ausgehenden Links. Dies entspricht den allgemeinen Erwartungen, da die meisten Artikel von vielen verschiedenen Artikeln verlinkt werden, die einzelnen Artikel aber meist nur auf wenige weiterverlinken. Ein Beispiel dafür sind berühmte Personen. Viele Artikel von Filmen, Auftritten und ähnlichen Gegebenheiten werden auf diese Person verlinkt, diese wiederum wird nur auf wenige Artikel weiterverlinken.

Sieben Jahre später brachte Helic in seinem Paper „Analyzing User Click Paths in a Wikipedia Navigation Game“ [11] von einem Wikipedia-Dump des 4.1.2012 dieselben Ergebnisse zum Vorschein. Auch zu erkennen ist dabei das Vorhandensein von Hubs und die große Anzahl an abgelegenen Artikeln, welche von nur sehr wenigen verlinkt werden und wiederum nur auf sehr wenige weiterverlinken. Zusätzlich produzierte Helic noch weitere Diagramme betreffend der schwach- bzw. stark-verbundenen Komponenten von Wikipedia und fand heraus, dass sich in der größten, schwach verbundenen Komponente 99% aller Knoten befinden, wohingegen sich in der größten, stark verbundenen Komponente nur 55% aller Knoten befinden.

Bei diesen Ergebnissen wurden keine Redirects aufgelöst, was einen Großteil des Unterschieds dieser beiden Komponenten ausmacht. Dies begründet auch die Knotenzahl von über 10 Millionen, obwohl derzeit nur 4.110.000 Artikel existieren. Sehr interessant ist auch der effektive Durchmesser¹⁰ des Wikipedia-Graphen, welcher kleiner als 6 ist. Dies weist auf ein stark verbundenes Netz-

¹⁰ Die Länge des kürzesten Pfades zwischen 90% aller Knoten. Durchmesser: Länge des längsten kürzesten Pfades im Netzwerk.

werk hin und lässt es per Definition¹¹ zur Gruppe der „Kleinen-Welt Netzwerke“ gehören.

3.2.2 *Benutzerverhalten*

Zusätzlich zu den Klickanalysen im World-Wide-Web, hat Gleich in seiner Studie sich auch auf das Klickverhalten im Wikipedia-Netzwerk konzentriert. [4] Dabei verwendete er wiederum die schon zuvor erwähnten Log-Daten der Microsoft-Toolbar und filterte daraus alle Wikipedia-Klicks. Diese unterteilte er weiter in „Klicks zwischen Wikipedia Artikeln“ und „Links von außerhalb auf Wikipedia-Artikel“. Das heißt, ersteres beinhaltet nur Daten, wo Internetuser von einer Wikipedia-Seite zur nächsten navigierten. Als Teleportation, eine Navigation entlang einer nicht existierenden Verlinkung, wurden alle Klicks klassifiziert, welche als Resultat das Verlassen von Wikipedia hatten. Interessanterweise war die Wahrscheinlichkeit, dass ein Benutzer einem Link auf Wikipedia folgt, geringer als der im World-Wide-Web. Mit nur 32,5%-42,5% Wahrscheinlichkeit, folgten User weiteren Links auf Wikipedia. Jedoch ist dies noch immer ein beachtlicher Wert, welcher die Existenz der Navigation durch das Folgen von Links bestätigt.

3.2.3 *Zoom-out und Zoom-in Phase*

In der schon zuvor erwähnten Arbeit von Helic [11] wird von einer Zoom-in und Zoom-out Phase berichtet. Dieses Phänomen wurde schon zuvor von [27] entdeckt und beschrieben. Es handelt sich dabei um eine bestimmte Navigationsstrategie von Menschen in Wikipedia-Netzwerken. Um genauer zu sein, navigieren User, wenn sie von Artikel A nach Artikel B wollen zuerst zu einem Knoten mit hohem Knotengrad, einem sogenannten Hub. Der Weg von A zu diesem Hub wird als Zoom-out Phase bezeichnet, da A in fast allen Fällen kein Hub ist und somit einen viel geringeren Grad besitzt. Der Benutzer versucht einen Knoten zu erreichen, von welchem aus er möglichst viele andere Artikel auf direktem Weg erreichen kann. Ist er dort angekommen, beginnt die Zoom-In

¹¹ Die durchschnittliche Länge aller kürzesten Pfade muss kleiner $\log(\#\text{Knoten im Graph})$ sein.
 $\log(10\text{Mio.}) = \sim 7$

Phase. Der Suchende wählt einen Artikel der zu seinem gesuchten Artikel am ehesten passt, sei dies nun geographisch oder auf das Themengebiet bezogen. Von dort aus wird er sich zu immer weniger wichtigen Knoten im Graphen navigieren, bis er sein Ziel erreicht hat. Daraus resultiert die typische Eigenschaft für den Verlauf der Suche. Zuerst werden Knoten mit hohem Grad bevorzugt, anschließend verringert sich der Knoten-Grad der besuchten Knoten kontinuierlich.

Dieses Verhalten stellte auch [28] in einem groß angelegten Versuch fest. Dabei wurden 30.000 Benutzerpfade von 9.400 unterschiedlichen IP-Adressen aufgezeichnet und analysiert. Ziel der User war es in einem Informationsnetzwerk bestehend aus 4.000 Artikeln, welche durch 120.000 Links verlinkt waren, den möglichst kürzesten Pfad von einem zufällig ausgewählten Startartikel zu einem ebenso zufälligen Zielartikel zu finden. Der Durchschnitt aller Distanzen von kürzesten Pfaden zwischen fast allen Knoten in diesem Netzwerk betrug 3, was wiederum auf ein extrem verbundenes Netzwerk hinweist. Bei den Analysen wurde außerdem zwischen effektiven und normalen Pfaden unterschieden. Letztere beruhen auf den original aufgezeichneten Daten, während bei den effektiven alle Backtracks entfernt wurden, sodass daraus ausschließlich die Pfade direkt zum Ziel entstanden.

Im effektiven Datensatz wurde festgestellt, dass Menschen äußerst effizient im Finden von kürzesten Pfaden sind. Knotenpaare welche eine kürzeste Distanz von 3 im Graphen besaßen, wurden von den Spielern mit durchschnittlich nur einem Klick mehr (Distanz 4) verbunden. Sogar im unmodifizierten Datensatz, welcher alle Backtracks als Klicks wertete, konnten die Menschen mit durchschnittlich nur 5 Klicks dieselben Knotenpaare verbinden. Des Weiteren wurden die durchschnittlich besuchten Knotengrade des effektiven Datensatz genauer analysiert. Dabei stellte sich heraus, dass der Startknoten einen durchschnittlichen Knotengrad von 30 hatte. Der erste danach besuchte Knoten hatte hingegen einen durchschnittlichen Knotengrad von 80-100. Anschließend fiel der Knotengrad konstant bis zum Ende des Spiels ab. Dies spiegelt exakt die schon zuvor beschriebene Zoom-out und Zoom-in Phase von menschlichen Navigationspfaden wieder.

3.3 Kleine-Welt-Phänomen

Man fährt auf Urlaub oder ins Ausland, kommt dort zufällig mit einer bis dato fremden Person ins Gespräch und es stellt sich heraus, dass man gemeinsame Freunde oder Bekannte hat. Die typische Aussage ist dann „Ist die Welt nicht klein?“. Genau damit hat sich schon 1967 Stanley Milgram, ein sehr bekannter Experimental-Psychologe aus New York, beschäftigt. Um nun gleich zu Anfang eine Verbindung zu dieser Masterarbeit herzustellen, nehmen wir an, dass Menschen immer Knoten in einem Graphen und die Kanten die Freundschaften zwischen zwei Menschen repräsentieren. Milgram wollte damals wissen, ob eine typische kürzeste Distanz zwischen zwei Menschen in einem solchen Graphen existiert. Da es damals, als auch heute, unmöglich war alle Menschen dieser Welt mit all ihren Freundschaften zu erfassen, überlegt sich Milgram ein Experiment.

Er schickte 96 seriös wirkende Briefe, welche mit dem Wappen seiner Universität bedruckt waren, an verschiedene Leute in Omaha, Nebraska. Diese wählte er zufällig aus einem Telefonbuch aus. In den Umschlägen befand sich eine Anleitung, aus welcher hervorging, was die Empfänger mit diesem Brief machen sollten. Milgram bat die Adressaten, den Brief an seinen Kollegen in Boston, Massachusetts unter bestimmten Voraussetzungen weiterzuleiten. Anbei war Name, Adresse und sein Job. Das Besondere daran war allerdings die Regel, dass es nicht erlaubt war den Brief direkt an die angegebene Adresse zu schicken, solange man ihn nicht persönlich kannte. Da natürlich niemand Milgram's Freund persönlich kannte, sollten die Empfänger den Brief an einen ihrer Freunde schicken, von welchem sie dachten, dass dieser mit ihm befreundet sein könnte oder zumindest jemanden kennt, der mit ihm befreundet ist. Das heißt, sie sollten den Brief an denjenigen Freund weiterleiten, wo sie sich die größte Chance erhofften, dass der Brief baldigst sein Ziel erreicht. Aus diesen Bedingungen heraus, schickten die meisten Leute den Brief an Bekannte aus Massachusetts oder Bekannte, welche in derselben Branche wie die Zielperson arbeiteten. Die Wahl blieb aber jedem selbst überlassen. Für die Empfänger des Briefes galten wieder dieselben Voraussetzungen wie für die ersten Empfänger. Dieser Ablauf sollte solange durchgeführt werden, bis der

Brief sein Ziel erreichte. Außerdem sollte jeder Weiterleitende seinen Namen dem Brief hinzufügen. Aus der Anzahl der Namen ergab sich am Ende eine obere Grenze für die kürzeste Distanz zwischen den einzelnen Personen und der Zielperson im sozialen Netzwerk. [7]

18 der ursprünglich 94 Briefe erreichten am Ende ihr Ziel. Obwohl dies gering erscheint, ist es für dieses Experiment eine sehr hohe Erfolgsquote, da spätere Versuche dieses Experiment zu wiederholen viel schlechter abschnitten. [12] Die durchschnittliche Pfad-Länge aller erfolgreich zugestellten Briefe lag bei nur 5.9 Schritten. Daraus entstand auch der weitverbreitete Glaube, dass jeder Mensch mit jedem anderen Menschen auf dieser Welt über nur 6 Bekanntschaften verbunden ist.

Leider gibt es an diesem Experiment einiges zu kritisieren. Alle Start-Personen befanden sich geographisch gesehen am selben Standpunkt und keiner der erfolgreich zugestellten Briefe hat jemals das Land auf seinem Weg zum Ziel verlassen. Des Weiteren ist die Zielperson immer dieselbe gewesen. Hier müsste auch zwischen wenig bekannten Leuten und bekannten Leuten unterschieden werden, da letztere sicherlich leichtere Ziele darstellen. Außerdem existiert keine Referenz, mit welcher überprüfbar wäre, ob wirklich der kürzeste Pfad gewählt worden ist.

Später hat Kleinberg einen weiteren sehr interessanten Aspekt in Milgram's Experiment aufgezeigt. Das Experiment bestätigte nicht nur die Existenz von kurzen Pfaden, sondern brachte auch als zusätzliches Ergebnis hervor, dass die Menschheit relativ gut darin sein muss, diese zu finden.

2003 wurde ein ähnliches Experiment durchgeführt, jedoch mit der modernen Version von Briefen: Der E-Mail. Dodds nahm sich die Kritik an Milgram's Experiment zu Herzen und startete so 24.000 E-Mails von fast 24.000 ungleichen Startpersonen. Außerdem erhöhte er die Anzahl der Ziele von 1 auf 18, welche sich auch in 13 verschiedenen Ländern befanden. Wie schon zuvor erwähnt fiel die Erfolgsquote relativ gering aus. Nur 384 Mails, dies entspricht nur 1,5%, erreichten ihr vorgesehene Ziel. Zur Erinnerung: In Milgram's Experiment lag die Erfolgsquote bei 19%. Durch genaue statistische Analysen gli-

chen Dodds Ergebnisse denen von Milgram sehr. Er fand heraus, dass sich die durchschnittliche Distanz zwischen 5-7 Schritten befand, was Milgrams durchschnittlicher Distanz von 5,9 sehr ähnlich ist.

Killworth und Bernard wollten sich das „Kleine-Welt-Phänomen“ zunutze machen, um genaueres über die verwendete Navigationsstrategie der Menschen zu erfahren. Dazu sagten sie verschiedenen Versuchspersonen, dass diese an einem „Kleine-Welt Experiment“ teilnehmen würden. Anschließend wurden sie befragt, welche Eckdaten sie über ihre Zielperson gerne wissen würden. Das „Kleine-Welt Experiment“ an sich fand nie statt, jedoch konnten Killworth und Bernard aus den gewünschten Daten über die Zielpersonen, die Navigations-taktik der Menschen erfahren. Es stellte sich heraus, dass die wichtigste Information der Name, gefolgt vom geographischen Wohnort und dessen Job war. Interessanterweise kamen auch andere wichtige Merkmale zum Vorschein, als sie dasselbe Experiment in anderen Kulturen durchführten. Dort war oft die Religion oder die Abstammung, sprich Elternschaft, gefragt.

3.3.1 *Benutzermodellierung in Wikipedianetzwerken*

In [29] wurde sich die Frage gestellt, ob das enorme Hintergrundwissen der Menschen nötig ist, um gleichgut oder besser in großen Netzwerken zu navigieren. Die Versuche wurden mittels 30.000 gesammelter Userpfaden von Wikispeedia durchgeführt. Das zugrundliegende Netzwerk hatte 4.604 Knoten mit einer durchschnittlichen kürzesten Distanz von 3. Es wurden 5 verschiedene Suchagenten implementiert, welche alle nur einfachste, numerische Features für die Navigation verwendeten.

Die ersten drei dieser Agenten fallen in die Überkategorie „heuristische Agenten“. Hier wurde einmal mittels des Grades der Nachbarknoten und einmal mittels TF-IDF von Nachbar und Zielknoten, der nächste Knoten in jedem Navigationsschritt ausgewählt. Als dritter Versuch in dieser Kategorie wurden die beiden miteinander kombiniert. Dabei stellt sich heraus, dass der Agent, welcher nur anhand der Grade navigierte, am schlechtesten abschnitt. Dies lag auch an der Tatsache, dass Zielknoten nicht unbedingt Knoten mit hohem Grad waren. Die beiden anderen erzielten jedoch beachtliche Ergebnisse und konn-

ten den Menschen deutlich schlagen. Sie erzielten einen Stretch von 1,5, während der Mensch nur auf einen Durchschnitt von 2 kam. Es wurde auch festgestellt, dass sich Menschen weniger oft komplett im Netzwerk verirren, wohingegen die genannten Agenten in seltenen Fällen das gesamte Netzwerk durchsuchten, bis sie ihr Ziel gefunden hatten.

Die letzten beiden Suchagenten wurden mittels maschinellen Lernens gesteuert. Es stellt sich heraus, dass diese die besten Ergebnisse erzielten, sprich den geringsten Stretch hatten. Interessant für diese Arbeit ist hier die Tatsache, dass beim maschinellen Lernen in den ersten beiden Schritten die Grade der Nachbarsknoten sehr hoch gewichtet wurden, dann jedoch die Wichtigkeit dieser stetig fiel. Dies ist ein eindeutiges Anzeichen für die schon zuvor beschriebene Zoom-Out Phase. Da diese Arbeit jedoch keinen besseren Agenten als den Menschen modellieren will, sondern ohne maschinelles Lernen einen möglichst menschenähnlichen Agenten erzeugen versucht, wird auf diese beiden nicht näher eingegangen.

4. Daten und deren Repräsentation

In diesem Kapitel werden sämtliche Daten und ihr Format detailliert erklärt.

4.1 Graph

Der verwendete Graph wurde von Horwath [6] aus einem Wikipedia Dump vom 15.11.2011 generiert. Dabei wurden alle Redirects aufgelöst und etliche andere Modifikationen vorgenommen. Eine genaue Beschreibung der Aufbereitung findet sich in [6]. Der Graph besteht aus 3.8 Millionen Knoten und 232 Millionen Kanten.

Das Ergebnis der Datengewinnung ist eine Graph-Datei im Binärformat der SNAP-Bibliothek. Diese Bibliothek wird in Kapitel 5.2 genauer beschrieben. Mittels des GraphConverter-Tools des MUN-Frameworks kann diese in Klartext umgewandelt werden. In der daraus resultierenden Datei befindet sich eine Edgelist, das heißt eine Liste aller Kanten des Graphen. Jede Zeile repräsentiert eine Kante im Graphen, wobei diese jeweils aus der ID¹² des Startknotens, einem Leerzeichen oder Tabulator, und der Zielknoten-ID besteht. Die IDs der Knoten sind die IDs der zugrundeliegenden Wikipedia-Artikel und können in die Artikelnamen aufgelöst werden. Genauer zu dieser Auflösung findet sich in [6]. Da enorm große Graphen, wie der verwendete Wikipedia-Graph, nicht effizient mit der SNAP-Bibliothek eingelesen werden können, wurde in den Experimenten ausschließlich das Binärformat verwendet. Als kurzer Vergleich: Das Einlesen des im Klartext vorliegenden Wikipedia-Graphen kann bis zu drei Stunden dauern, wohingegen derselbe Graph im binären Format nur eine Minute benötigt. Außerdem stellte sich heraus, dass dadurch auch weniger Arbeitsspeicher nach dem Einlesen benötigt wird. Ein Beispiel für die Kantenliste eines kleinen Graphen findet sich in Listing 1.

¹² ID: Identifikator

2	23
23	1
1	11
1	12
1	13
1	14
23	15
23	22
2	24
24	25
2	21
21	3
3	31
3	32
3	33

Listing 1: Edgelist eines simplen Graphen

4.2 Hierarchie

Da Hierarchien Graphen mit besonderen Eigenschaften sind, liegen diese im selben Format vor. Meist sind diese jedoch nicht im Binärformat von SNAP, da sie bedingt durch ihre geringere Größe auch als Klartext effizient eingelesen werden können. Der Größenunterschied zwischen Graph und Hierarchie resultiert aus der Tatsache, dass eine Hierarchie $N-1$ Kanten besitzt, wobei N die Anzahl der Knoten ist. Graphen können hingegen viel stärker vernetzt sein, woraus sich auch eine größere Edgelist ergibt.

4.3 Klickpfade

Diese Dateien repräsentieren die einzelnen Navigationen der User oder des Simulators in einem Graph. Pro Zeile befindet sich jeweils ein Pfad bestehend aus der ID des Startknotens, ID des Zielknotens, gefolgt von den IDs aller besuchter Knoten in der Reihenfolge, in welcher sie besucht wurden. Getrennt sind die einzelnen IDs wieder durch Leerzeichen oder Tabulatoren. Die Folge aller verwendeten Knoten beinhaltet anfangs noch einmal die Startknoten-ID sowie am Ende die Zielknoten-ID, insofern dieser erreicht wurde. Dadurch lassen sich erfolgreiche Pfade von nicht erfolgreichen Pfaden unterscheiden. Eine Besonderheit der Klickpfade ist, dass bei Backtracks¹³ nur der nächste angeklickte Knoten eingetragen ist. Dadurch ergibt sich die Möglichkeit von

¹³ Ein Backtrack kann von Usern durch den Zurück-Knopf im Browser erzielt werden.

zwei nicht direkt verbundenen Knoten im Graph, welche aber in einem Klickpfad nacheinander vorkommen. Listing 2 zeigt vier Klickpfade, wobei die ersten beiden im Gegensatz zu den letzten beiden erfolgreich waren. Zur Verdeutlichung der Formatierung: Der zweite Klickpfad sollte ausgehend von Knoten mit ID 1 den Knoten mit ID 25 erreichen. Der User klickte sich dabei über 1->21->2->25 zum Zielknoten. Identische Klickpfade können in dieser Datei auch öfter vorkommen, wie im Beispiel ersichtlich ist.

1	3	1	21	3	
1	25	1	21	2	25
1	15	1	22	23	2
1	15	1	22	23	2

Listing 2: Klickpfade in einem simplen Graphen

4.4 Paare

Die Paare stellen die einzelnen Missionen der Spiele dar. Pro Zeile befindet sich in der Datei eine Spielaufgabe bestehend aus Startknoten-ID, Zielknoten-ID, Grad des Startknotens und Grad des Zielknotens. Die IDs und Grade der Knoten sind jeweils durch ein Leerzeichen oder Tabulator getrennt. Identische Paare können in der Datei öfter vorkommen, da eine Spielaufgabe öfter von verschiedenen Usern gespielt werden kann.

1	24	7	2
1	13	7	1
1	14	7	2
1	12	7	2
1	33	7	1
1	33	7	1

Listing 3: Darstellung von Spielaufgaben mittels ID-Paaren

4.5 Kürzeste-Distanzen

Für jede gegebene Spielaufgabe steht in dieser Datei die Länge des kürzesten Pfades zwischen den beiden Knoten. Die Formatierung dieser Datei ist im Vergleich zu den vorherigen Daten etwas komplizierter. Am besten lässt sich diese anhand des Beispiels aus Listing 4 erklären. -1 gefolgt von einer Knoten-ID kennzeichnet immer einen neuen Startknoten. Dieser gilt für alle folgenden

Zeilen, bis ein neuer definiert wird. Nach der Definition findet sich pro Zeile die ID des Zielknotens, gefolgt von der kürzesten Distanz zu diesem. Die erste Zeile in Listing 4 legt 1 als Startknoten fest. In der nächsten Zeile steht 11 für die ID des Zielknotens und 1 ist die kürzeste Distanz zu diesem. Man beachte, dass die kürzeste Distanz in gerichteten Graphen von Knoten A zu Knoten B, nicht unbedingt der von B nach A entsprechen muss.

-1	1
11	1
12	5
13	4
14	2
-1	2
22	1
23	3

Listing 4: Beispiel einer Kürzesten-Distanzen Datei

5. Arbeitsumgebung

Dieses Kapitel befasst sich mit dem verwendeten Framework und der davon verwendeten Bibliothek.

5.1 SNAP

SNAP steht für Stanford Network Analysis Platform. Es ist eine in C++ geschriebene Bibliothek, welche an der Stanford University entwickelt wird und derzeit in Version 1.11¹⁴ vorliegt. Das Projekt, welches unter der BSD-Lizenz steht, wird seit 2004 ständig weiterentwickelt und soll die Verarbeitung von Graphen und Netzwerken in C++ vereinfachen. Das Hauptaugenmerk liegt dabei auf Performance und Skalierbarkeit. Beispielsweise wurde bereits ein Datensatz des Microsoft Instant Messenger Netzwerk aus dem Jahr 2006 damit analysiert, welcher aus 240 Millionen Knoten und 1.3 Milliarden Kanten bestand. Grundfunktionen sind das Einlesen, Speichern, Modifizieren und Analysieren von gerichteten als auch ungerichteten Graphen. Da mit Hilfe dieser Bibliothek bereits in Stanford Wikipedia-Netzwerke erfolgreich analysiert wurden, ist sie ein hervorragender Grundbaustein für die Navigationssimulation in diesen Netzwerken.

5.2 MUN-Framework

Das MUN-Framework wurde von Eder [5] eigens für die Ergebnisse dieser Arbeit entwickelt. Die Buchstaben MUN stehen dabei für „Modeling User Navigation“. Es verwendete die SNAP-Bibliothek und ist ebenfalls in C++ geschrieben. Das Framework ist ein Navigationssimulator für Netzwerke und kann über etliche Parameter modifiziert werden. Die grundlegenden Parameter werden in [5] detailliert erklärt.

Wird das Framework gestartet, so werden zuerst der Graph und eine dazu passende Hierarchie eingelesen. Des Weiteren besteht die Möglichkeit vordefinierte Start- und Zielknoten einlesen zu lassen. Geschieht dies nicht, so werden

¹⁴ Stand 21.1.2013 <http://snap.stanford.edu/snap/download.html>

automatisch zufällige generiert. Anschließend werden die kürzesten Distanzen für alle Knotenpaare benötigt. Diese können schon zuvor generiert worden sein und an dieser Stelle eingelesen werden. Ist dies nicht der Fall, so werden diese automatisch berechnet. Da dies in großen Netzwerken jedoch sehr rechenintensiv ist, empfiehlt es sich, die erste Variante zu wählen. In dieser Masterarbeit wurden alle benötigten Daten bereits von Horwath [6] aufbereitet bzw. generiert.

Sind alle Daten eingelesen, so beginnt die Navigationssimulation. Dabei wird mittels verschiedener Navigationsstrategien versucht, für jedes gegebene Knotenpaar einen Weg im Netzwerk zu finden. Das Framework stellt für diesen Zweck eine Schnittstelle zur Verfügung, welche es erlaubt, dieses um sogenannte Node-Selectoren zu erweitern.

Die Aufgabe eines Node-Selectors ist es, für einen gegebenen Knoten einen nächsten Knoten für die aktuelle Simulation auszuwählen. Das heißt, dieser bestimmt den Großteil der Simulation. Im Zuge dieser Masterarbeit wurden zum bereits von Eder [5] implementierten Random Node-Selector zwei zusätzliche Node-Selectoren implementiert. Genaueres dazu findet sich in Kapitel 6.

Außerdem stellt Eder [5] die Möglichkeit zur Verfügung, beliebig viele Node-Selectoren mittels Wahrscheinlichkeiten und Verwendungshäufigkeit kombiniert zu verwenden.

Ist die Navigationssimulation vollständig durchgeführt worden, so werden automatisch Plots über verschiedene Pfadeigenschaften erzeugt. Zusätzlich können zuvor genierte Klickpfade angegeben werden, mit welchen dann die generierten Pfade verglichen werden.

Für den Speicherort der Ergebnisse gibt es einen Parameter, über welchen der Ausgabeordner gesetzt werden kann. Dies ist besonders hilfreich, wenn mehrere MUN-Framework Prozesse gleichzeitig am selben System laufen. Außerdem erleichtert dies, den Überblick über die Ergebnisse zu behalten.

Ein weiterer sehr wichtiger Parameter ist *global.amount*. Dieser erlaubt zu Testzwecken und Parameteroptimierung den Datensatz zu verringern und verkürzt somit die Wartezeit enorm.

Eine sehr positive Eigenschaft des MUN-Frameworks ist dessen Dokumentation. Es lässt sich eine vollständige Doxygen-Dokumentation¹⁵ in Englisch erzeugen, welche sehr ausführlich ausfällt. Dies ist auch der Grund, wieso in dieser Masterarbeit nicht detailliert auf die Implementierung eingegangen wird.

¹⁵ Ein freies Tool zur Generierung einer Dokumentation aus dem Source-Code.
<http://www.stack.nl/~dimitri/doxygen/>

6. Node-Selectoren

Wie schon im letzten Kapitel kurz angesprochen, sind die Node-Selectoren die Hauptentscheidungsträger für jeden Schritt während der Navigation. Per Interface bekommen diese den aktuellen Standpunkt, die möglichen nächsten Knoten, den Zielknoten und alle bereits besuchten Knoten mitgeteilt. Des Weiteren besitzen sie alle Informationen über den Graphen, in welchem navigiert wird und können über die Distanz-Klasse Distanzen in der geladenen Hierarchie berechnen. Anhand dieser Informationen entscheidet jeder Node-Selector auf seine eigene Weise, welchen nächsten Knoten er als Ergebnis zurückliefert.

6.1 Random

Dieser Node-Selector wurde von Eder [5] implementiert und wählt einfach einen zufälligen Knoten aus allen möglichen nächsten Knoten aus.

6.2 Greedy

Greedy – gierig - ist die Eigenschaft, anhand welcher dieser Node-Selector den nächsten Knoten auswählt. Er möchte, ausgehend von seiner aktuellen Position, den in seinen Augen lukrativsten nächsten Knoten auswählen. Um differenzieren zu können, welcher lukrativ ist und welcher nicht, wird für jeden möglichen nächsten Knoten die Distanz in der Hierarchie zum Zielknoten berechnet. Der Knoten mit der geringsten Distanz ist in diesem Fall der lukrativste. Es kann jedoch vorkommen, dass mehrere mögliche, nächste Knoten mit derselben Distanz existieren. Das heißt, es gibt mehr als nur einen Knoten mit der geringsten Distanz. Tritt dieser Fall ein, so wird aus diesen zufällig einer ausgewählt. Dabei wird jedoch darauf geachtet, dass kein bereits besuchter Knoten gewählt wird.

Grundsätzlich wählt dieser Node-Selector nie in diesem Navigationsweg bereits zuvor besuchte Knoten aus, insofern noch nicht alle schon besucht worden sind. Wurde der beste Knoten bzw. die besten Knoten schon verwendet, so werden diese ignoriert und der nächstbeste Knoten ausgewählt. Wurden alle

möglichen, nächsten Knoten bereits besucht, so werden alle als noch nicht verwendet angesehen.

6.3 Teleport

Geben User auf Wikipedia etwas in das Suchfeld ein, so können sie sich anhand der Ergebnisse auf eine Seite navigieren, welche nicht unbedingt eine direkte Verbindung von der aktuellen Seite im Netzwerk hat. Teleportation beschreibt in diesem Fall eine Navigation von Knoten A zu Knoten B, wobei diese nicht direkt miteinander verbunden sind.

Der Teleport-Node-Selector kann über zwei zusätzliche Parameter eingestellt werden:

- `teleport.targetneighbourhood:X`
Dieser Parameter gibt an, in welche X-Nachbarschaft des Zielknotens teleportiert werden soll. Intern wird dabei vom Zielknoten aus X-Schritte in eine zufällige Richtung gegangen und der daraus resultierende Knoten retourniert. Der reale Hintergrund zu diesem Parameter ist, dass im World-Wide-Web mittels Suchanfragen meistens in die unmittelbare Nähe der gesuchten Seite teleportiert wird.
- `teleport.tobetter:T`
„To better“ bedeutet übersetzt „sich verbessern“. Dies beschreibt die Aufgabe des Parameters sehr gut. Er kann entweder T für true oder F für false sein. Ist er auf T gesetzt, so wird solange nach einem zufälligen Knoten im Graphen gesucht, bis dieser eine geringere Distanz in der Hierarchie hat, als der aktuelle. Das bedeutet, der nächste Knoten ist in der Hierarchie näher beim Zielknoten, als die aktuelle Position.

7. Link-Restrictor

Dieses Kapitel befasst sich mit dem für diese Arbeit benötigten Link-Restrictor, welcher anhand von diverser Parameter die nächsten, möglichen Knoten in jedem Navigationsschritt vorfiltert.

7.1 Motivation

Die ursprüngliche Motivation für dieses Modul im MUN-Framework war die Annahme, dass Benutzer von Wikipedia nicht alle Links auf einer Seite verarbeiten können bzw. wollen und daher nur die ersten X-Links betrachten. Da jedoch der von Usern angeklickte Link eine große Streuung in den Klickdaten aufwies, wurde die Funktionalität geändert.

Wie in Kapitel 3.2.3 bereits erwähnt, existiert bei vielen Klickpfaden eine sogenannte Zoom-In und Zoom-Out Phase. Auch bei den von Horwath [6] aufbereiteten Klickpfaden des Wikigames wurden diese Phasen eindeutig festgestellt. Dies ist in Abbildung 10 zu sehen. Wobei hier auch auffällig ist, dass die Zoom-In Phase viel geringer ausfällt. Eine detaillierte Beschreibung diese Typs von Plots befindet sich in Kapitel 9.2.

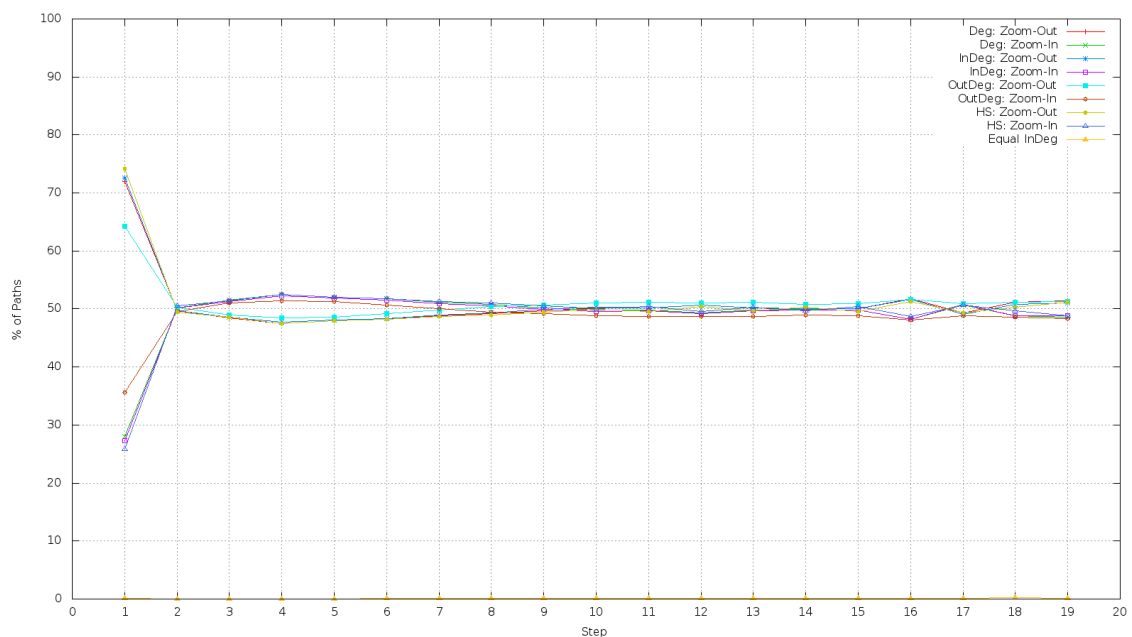


Abbildung 10: Zoom-In und Zoom-Out Phase der Wikigame-Klickdaten

7.2 Implementierung

Der Link-Restrictor wird in jedem Navigationsschritt vor dem Node-Selector aufgerufen. Dabei bekommt er eine Liste aller Nachbarknoten, das heißt, alle möglichen nächsten Knoten, übergeben. Anhand verschiedener Parameter entscheidet er nun, wie diese sortiert und vorgefiltert werden. Würde er beispielsweise nur die 10 möglichen nächsten Knoten mit dem höchsten Eingangsgrad herausfiltern, so erstellt er sich eine Liste mit 10 freien Speicherplätzen für Knoten. Anschließend geht er die Liste aller möglichen nächsten Knoten durch und fügt so lange Knoten in seine Liste ein, bis diese voll ist. Ist dieser Zeitpunkt erreicht, werden die Knoten in der Liste anhand ihrer Eingangsgrade sortiert. Für jeden weiteren Knoten der möglichen, nächsten Knoten wird nun nur mehr mit dem letzten in der Liste verglichen, da dieser den kleinsten Eingangsgrad besitzt. Ist der Eingangsgrad einer dieser Knoten größer als der des letzten in der Liste, so wird der letzte durch diesen ersetzt, die Liste wieder sortiert und anschließend die Prozedur fortgesetzt.

Aus Performance Gründen wird zusätzlich die Anzahl der Ergebnisknoten mit der Anzahl aller möglichen, nächsten Knoten verglichen. Ist diese größer als die Hälfte, so wird die Komplementärmenge berechnet und diese Knoten anschließend aus den möglichen, nächsten Knoten entfernt. Dies bringt einen extremen Performancegewinn, wenn der Restrictor beispielsweise auf 90% filtern soll, da hier nur 10% der nächsten Knoten berechnet und sortiert werden müssen.

7.3 Einstellungsmöglichkeiten

Der Link-Restrictor besitzt insgesamt 12 Parameter, wovon 6 für spezielle Einstellungen von Zoom-In und Zoom-Out Phase gedacht sind.

Der Parameter *linkrestrictor.amount* gibt an, auf wie viele Knoten die nächsten möglichen Knoten beschränkt werden sollen. Ist dieser beispielsweise auf 10 gesetzt und der Link-Restrictor bekommt 50 mögliche nächste Knoten übergeben, so filtert er nur 10 heraus. Welche Kriterien diese 10 erfüllen müssen wird von den beiden Parameter *linkrestrictor.degreetype* und *linkrestrictor.asc* bestimmt. *Degreetype* bestimmt auf welchen Wert der möglichen nächsten Knoten

geachtet wird. Dabei kann zwischen Eingangsgrad, Ausgangsgrad, Grad und Hierarchical-Score gewählt werden. Der andere Parameter gibt an, ob die nächsten Knoten mit dem niedrigsten oder höchsten Wert ausgewählt werden sollen. Angenommen, *linkrestrictor.degree* ist auf den Eingangsgrad gesetzt, *linkrestrictor.asc* auf T (true) und wie schon zuvor die Anzahl auf 10, so würden in diesem Fall die 10 nächsten möglichen Knoten mit dem niedrigsten Eingangsgrad retourniert werden.

Da in großen Netzwerken die Knotengrade sehr variieren, wurde die Option *linkrestrictor.usepercent* eingeführt. Ist diese aktiviert, so werden sämtliche *.amount* Angaben als Prozent verarbeitet. Die Prozente beziehen sich dabei auf die Anzahl der möglichen, nächsten Knoten.

Um hier auch zu beachten, dass es durch die Einschränkung mittels Prozent passieren kann, dass manche Ziele unerreichbar werden, wurde der Parameter *.stopat* entwickelt. Dieser gibt an, ab welchem Schritt der Link-Restrictor aufhört die möglichen, nächsten Knoten zu filtern. Im Standard-Konfigurationsfile ist dieser auf -1 gesetzt, was bedeutet, dass der Link-Restrictor nie deaktiviert wird.

Zusätzlich zu dieser Grundfunktionalität, kann der Link-Restrictor in zwei Phasen aufgeteilt werden. Den Übergang von Phase 1 auf Phase 2 bestimmt dabei der Parameter *linkrestrictor.swapat*. Ist dieser beispielsweise auf 2 gesetzt, so befindet sich der Link-Restrictor in jeder Navigation bis zum zweiten Schritt in Phase 1 und anschließend bis zum Ende der aktuellen Navigation in Phase 2.

Um nun diese zwei Phasen getrennt voneinander modifizieren zu können, gibt es für jede Phase die drei zuvor beschriebenen Grundeinstellungsmöglichkeiten. Die Parameter dazu lauten wie folgt: *linkrestrictor.first.degree*, *linkrestrictor.first.asc*, *linkrestrictor.first.amount*, *linkrestrictor.second.degree*, *linkrestrictor.second.asc* und *linkrestrictor.second.amount*.

Soll eine der beiden Phasen deaktiviert werden, so muss lediglich der betreffende *.amount*-Parameter auf -1 gesetzt werden. Um nun schon beim Start des Navigationssimulators sicher zu sein, dass alle Parameter richtig gesetzt und erkannt wurden, gibt dieser auf der Konsole die Einstellungen noch einmal aus.

In Listing 5 ist beispielsweise diese Ausgabe zu sehen. In diesem Fall ist die erste Phase deaktiviert, nach dem zweiten Schritt wird die Phase gewechselt und in der zweiten Phase werden 10 mögliche nächste Knoten mit dem höchsten Eingangsgrad vorgefiltert.

```
:: Link restriction activated  
First Phase NOT activated  
Change Phase after Step: 2  
Second Phase Degree-Type: i  
Second Phase Ascending: 0  
Second Phase Amount: 10
```

Listing 5: Ausgabe der Einstellungen des Link-Restrictors

8. Tools zur Auswertung

Da im Laufe des Projektes immer wieder kleinere Tools benötigt wurden, findet sich hier eine kurze Beschreibung dieser. Alle Tools verwenden die SNAP-Bibliothek oder einzelne Klassen aus dem MUN-Framework. Viele wurden nicht in den Ablauf des Navigationssimulators eingebaut, da diese die Performance merklich beeinträchtigen würden. Ein weiterer Grund war, dass der Simulator seine Ergebnisse zum Beispiel in Form einer Klickpfad-Datei abspeichert, diese aber erst danach analysiert wurden. Dies ersparte viel Zeit, da der Simulator nicht nach jeder Änderung im Analysetool neu gestartet werden musste.

8.1 Hierarchie Validierung

Dieses Tool war anfangs enorm wichtig, da so manche generierte Hierarchie von anderen Arbeitsgruppen nicht alle Eigenschaften einer solchen erfüllte und der Navigator damit nicht umgehen konnte. Das MUN-Framework unterstützt ausschließlich Hierarchien, welche eine Wurzel mit Eingangsgrad null besitzen. Außerdem darf kein Knoten mehr als einen Elternknoten besitzen, was ohnehin eine grundlegende Eigenschaft dieser speziellen Art von Graphen ist.

Wird dieses Werkzeug gestartet, so wird nur der Parameter *-h*: erwartet. Dieser gibt den Pfad zur Hierarchie an. Anschließend werden alle Knoten mit Eingangsgrad null auf der Konsole ausgegeben. Da eine Hierarchie nur eine Wurzel besitzen darf, sollte dies nur ein Knoten sein. Des Weiteren werden alle Knoten angeführt, welche einen Eingangsgrad größer als eins besitzen. Trifft dies auf einen Knoten zu, so ist der Graph keine Hierarchie, da er Knoten mit mehr als einem Elternknoten besitzt.

Erfüllt die Hierarchie alle gewünschten Eigenschaften, wird zusätzlich noch die ihre Tiefe berechnet und ausgegeben.

8.2 Generierung von Plots

Dieses Tool kann von allen verwendeten Daten Plots generieren. Welche Plots produziert werden, wird anhand der Parameter entschieden. Hier eine Auflistung aller Parameter.

- -p: Klickpfad-Datei
- -g: Graph
- -d: Ist der Graph gerichtet? (T/F)
- -h: Hierarchie
- -s: kürzeste Distanzen
- -o: Ausgabeordner

Um das Tool ausführen zu können, müssen nur die Daten angegeben werden, von welchen Plots erzeugt werden sollen. Jedoch können durch Kombinationen von Daten oft mehrere Plots erzeugt werden. Wird zum Beispiel nur die Klickpfad-Datei angegeben, so kann lediglich ein Plot betreffend deren Pfadlängen erstellt werden. Wird zusätzlich der Graph, in welchem diese Pfade generiert wurden angegeben, so wird auch ein Plot über den durchschnittlichen Grad der besuchten Knoten pro Schritt generiert. Die einzelnen Plots sind in Kapitel 9 zu sehen. Dort sind diese auch detailliert beschrieben und analysiert.

8.3 Konvertierung von Graphen

Da das Einlesen von großen Graphen in Form von Edgelisten enorm viel Zeit im Vergleich zum Einlesen desselben Graphen im Binärformat kostet, ist ein Tool, welches Konvertierungen zwischen den beiden Formaten beherrscht, von großem Vorteil.

Die Bedienung dieses Tools ist denkbar einfach. Es benötigt lediglich drei Parameter. Die ersten beiden davon beziehen sich auf den zu konvertierenden Graphen. `-g`: gibt den Dateipfad des Graphen an und `-d`: kann auf T für gerich-

tete und F für ungerichtete Graphen gesetzt werden. Der letzte Parameter ist $-o$. Dieser gibt den gewünschten Speicherort des konvertierten Graphen an. Dabei ist es wichtig, dass dieser die Dateierweiterung *.bin* besitzt. Wird dieser Parameter nicht gesetzt, so wird der konvertierte Graph automatisch unter dem Namen des eingelesenen Graphen plus der Erweiterung *.bin* gespeichert.

Das Tool beherrscht außerdem auch die Umwandlung von Graphen im Binärformat in Edgelisten. Dazu muss einfach der angegebene Graph auf *.bin* enden.

8.4 Plots über diverse Pfad-Eigenschaften

Dieses Tool wurde speziell für die Erstellung von Plots über Klickpfad-Daten implementiert. Es bietet zum normalen Plots-Generierungs-Tool noch die Option an, dass die verwendeten Strategien der einzelnen Schritte in den Klickpfaden wiederhergestellt werden.

Dabei wird für jeden Knoten in einem Klickpfad überlegt, mit welchem Node-Selector ausgehend vom vorherigen Knoten, dieser erreicht werden konnte. Ist der vorherige Knoten in der direkten Nachbarschaft des aktuellen Knotens, so muss nur noch die Entscheidung zwischen Greedy und Random gefällt werden. Dies geschieht mit Hilfe der Hierarchie.

Ist der aktuelle Knoten der lukrativste Nachbarknoten des vorherigen Knotens, das heißt er besitzt die kürzeste Distanz zum Zielknoten in der Hierarchie, so wurde hier mittels der Greedy-Navigationsstrategie navigiert. In diesem Fall kann noch zusätzlich zwischen Unique-Greedy und Random-Greedy unterschieden werden. Unique-Greedy bedeutet, dass der aktuelle Knoten als einziger Nachbar des vorherigen Knotens die geringste Distanz in der Hierarchie zum Zielknoten hat und kein weiterer mit derselben Distanz existiert. Hingegen werden bei Random-Greedy Navigationen erkannt, die nicht eindeutig waren.

Es kann auch vorkommen, dass der aktuelle Knoten kein direkter Nachbar des vorherigen Knotens ist. Hier wird dann zwischen Backtrack oder Teleportation unterschieden. Die Strategie wird als Backtrack erkannt, wenn der aktuelle Knoten ein Nachbarknoten der vorherigen Knoten in diesem Pfad ist. Ist dies nicht der Fall, so wurde Teleportation angewendet.

Die wiederhergestellten Strategien müssen natürlich nicht den wirklich verwendeten Strategien entsprechen, jedoch zeigen diese, wie die Navigation mit Hilfe des Simulators nachmodelliert werden kann. Ein weiterer sehr wichtiger Aspekt ist hierbei die Beurteilung der verwendeten Hierarchie. Werden zum Beispiel menschliche Klickdaten eingelesen und versucht, mit Hilfe dieses Tools die Strategien wiederherzustellen, so kann anhand der Anzahl der verwendeten Greedy-Navigationen erkannt werden, wie gut die Hierarchie das Menschenwissen modelliert. Natürlich wird hierbei angenommen, dass der Mensch auch immer den für ihn lukrativsten nächsten Knoten auswählt. Außerdem deutet ein großer Unterschied zwischen Unique-greedy und Random-greedy auf eine flache Hierarchie hin, da viele Knoten scheinbar auf demselben Level liegen.

9. Ergebnisse

Dieses Kapitel befasst sich mit den Ergebnissen, welche durch Anpassungen verschiedenster Parameter erzielt wurden. Des Weiteren wird zu Vergleichszwecken eine sogenannte Base-Line präsentiert. Diese bildet die Ausgangslage und soll vor allem verdeutlichen, welche Verbesserungen erzielt worden sind. Auch werden hier die für die Simulation verwendeten Daten graphisch als auch schriftlich analysiert.

9.1 Graph-Analyse

Der von Horwath [6] erzeugte Graph, stellt grundlegend das Wikipedia-Netzwerk zum Zeitpunkt des 15.11.2011 dar. Er besteht aus 3.826.689 Knoten welche durch 232.513.920 Kanten verbunden sind. Jeder Knoten repräsentiert dabei einen Artikel und jede Kante einen Link von einem Artikel auf einen anderen Artikel. Der gesamte Graph ist schwach zusammenhängend und besteht aus 219.765 stark zusammenhängenden Komponenten. Die größte, stark zusammenhängende Komponente beinhaltet 3.600.965 Artikel. Dies entspricht 94,1% aller Artikel. Alle weiteren stark zusammenhängenden Komponenten beinhalten jeweils unter 61 Artikel, das heißt unter 0.002%. In Abbildung 11 ist die Verteilung der Eingangsgrade aller Knoten in diesem Graph zu sehen. Die X-Achse gibt dabei den Eingangsgrad des Knotens an, wohingegen die Y-Achse angibt wie viele Knoten mit diesem Eingangsgrad im Graph existieren. Hier ist deutlich die Struktur des Graphen zu erkennen. Er besitzt eine Unmenge an Knoten, welche von sehr wenigen Artikeln verlinkt sind und nur sehr wenige Hubs, welche von enorm vielen anderen Artikeln verlinkt sind. Dasselbe gilt auch für die Verteilung der Ausgangsgrade der Knoten. Dies ist in Abbildung 12 gut zu erkennen.

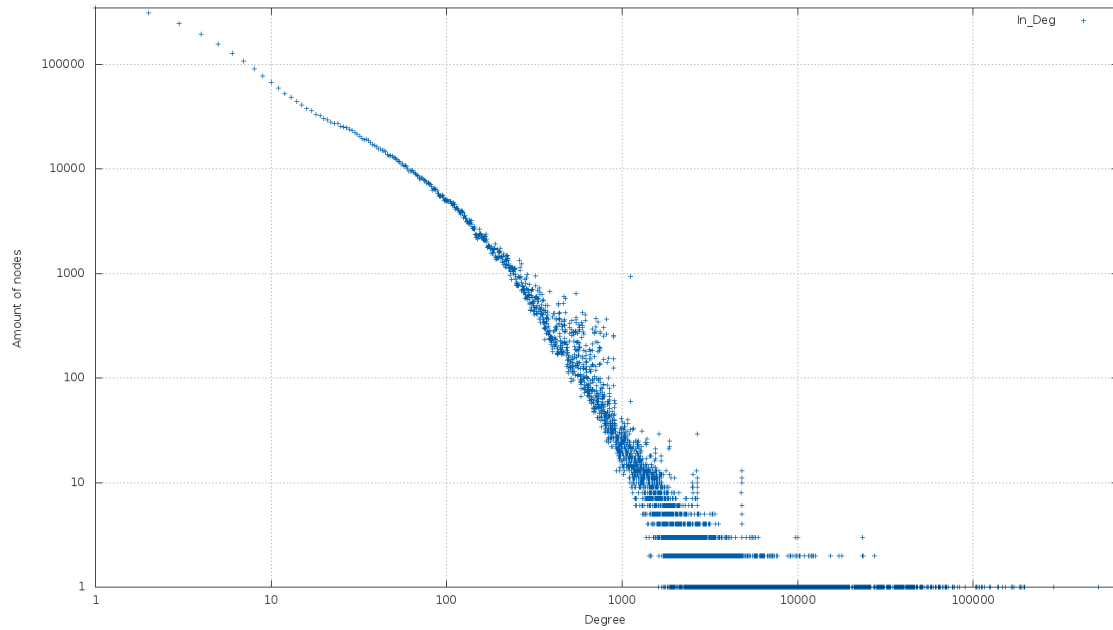


Abbildung 11: Verteilung der Knoteneingangsgrade

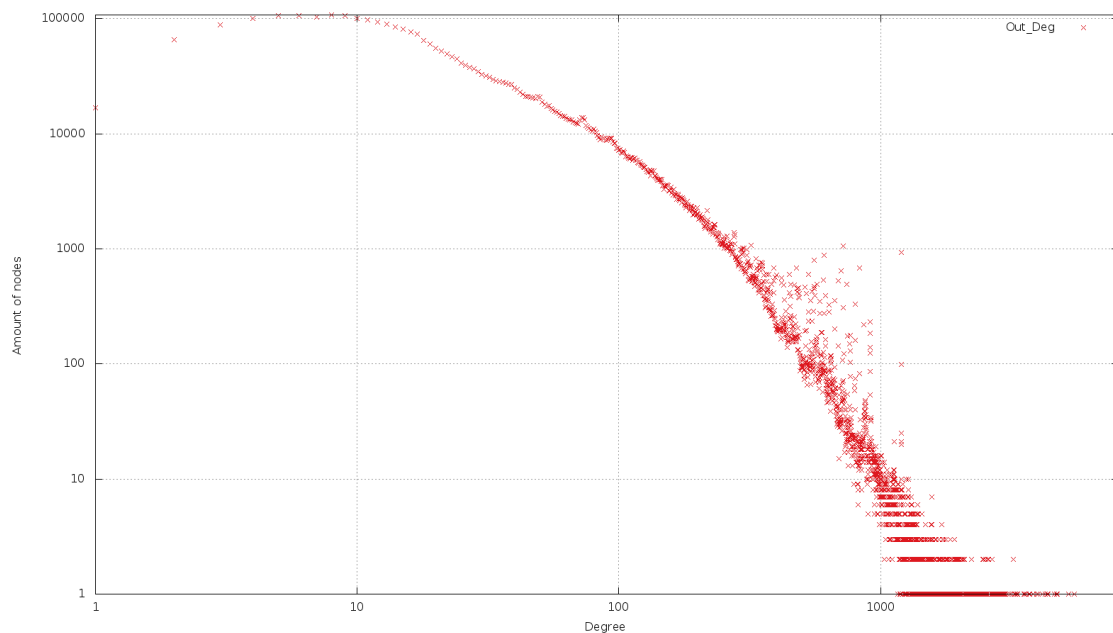


Abbildung 12: Verteilung der Knotenausgangsgrade

Zusätzlich wurden noch die kürzesten Distanzen aller 404.376 einzigartigen Missionen genauer analysiert. Dabei stellte sich heraus, dass zirka 55% davon eine kürzeste Distanz von 3 und zirka 41% von 2 hatten. Des Weiteren besitzen nur ca. 3% aller Spiele eine kürzeste Distanz von 4 und kein einziges Spiel eine von 5. Die Definition von „Kleinen-Welt-Netzwerken“ besagt, dass die durchschnittliche kürzeste Distanz kleiner als der Logarithmus zur Basis 10 aller

Knoten sein muss. Wird nun die Anzahl der Knoten in diese Formeln eingesetzt, so ergibt sich: $\text{Log}(3.826.689) \sim 6,58$. Da keine einzige kürzeste Distanz mit Länge 5 existiert, liegt der Durchschnitt sicherlich unter diesem Wert. Somit zählt dieses Netzwerk, wenn nur die Missionen betrachtet werden, zu den „Kleine-Welt-Netzwerken“. Die genaue Verteilung der kürzesten Distanzen aller Missionen ist in Abbildung 13 geplottet. Die X-Achse gibt dabei die kürzeste Distanz an und die Y-Achse die Prozent aller Missionen, welche diese besitzen.

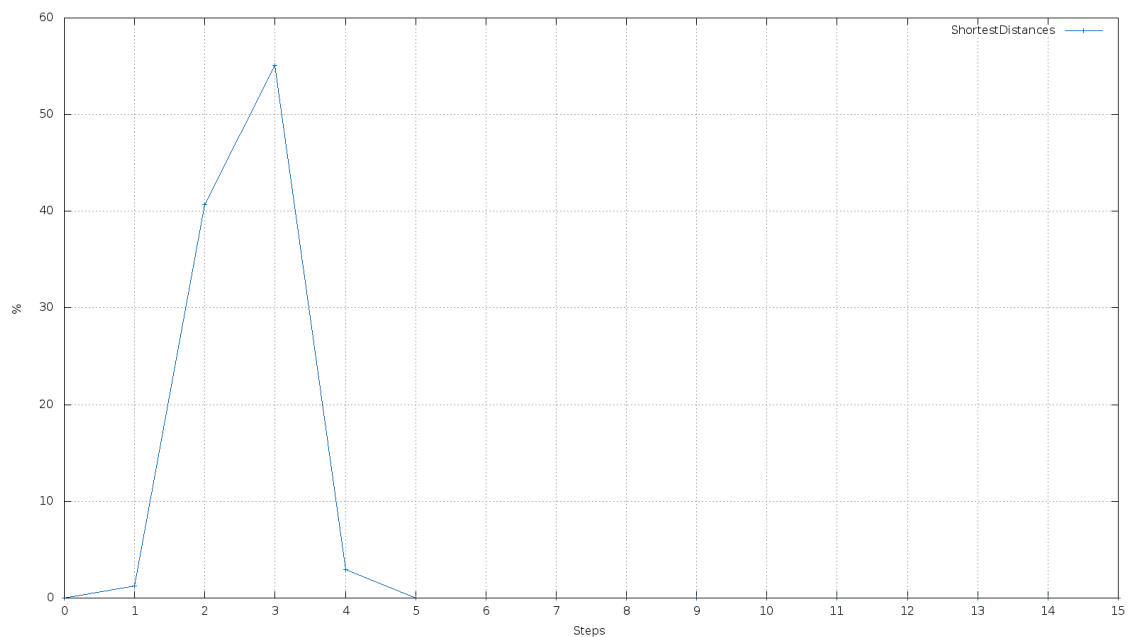


Abbildung 13: Verteilung der kürzesten Distanzen aller Missionen

Es wurden auch die Knoten mit dem höchsten Grad, Eingangsgrad, Ausgangsgrad und hierarchischen Wert aus dem Graphen von Horwath [6] herausgesucht. Eine Liste dieser findet sich in Tabelle 1.

Tabelle 1: Wichtige Knoten des Wikipediagraphen

ID	Artikelname	Eingangsgrad
48361	Geographic_coordinate_system	671160
3434750	United_States	518668
14919	International_Standard_Book_Number	288384
ID	Artikelname	Ausgangsgrad
24669262	List_of_Italian_communes_(2009)	8103
21824714	List_of_municipalities_of_Brazil	5515
274621	Index_of_India-related-articles	5251
ID	Artikelname	Grad
48361	Geographic_coordinate_system	671260
3434750	United_States	520327
14919	International_Standard_Book_Number	288764
ID	Artikelname	Hierarchischer Wert
48361	Geographic_coordinate_system	5498435
16130497	Population_without_double_counting	2350949
351656	Geographic_Names_Information_System	1900303

9.2 Klickpfad-Analyse

Der nächste Schritt ist die Analyse der gesammelten Benutzerdaten. Insgesamt waren dies 1.898.401 unterschiedliche Klickpfade. Davon wurden 786.845 erfolgreich beendet. Dies entspricht 41,45%. Interessant dabei ist, dass sowohl bei den erfolgreich gespielten Spielen, als auch bei den nicht erfolgreichen die meisten die Länge 4 besitzen. Bei den gewonnenen Missionen besitzen über 20% diese Länge und jeweils ca. 16% die Länge 3 und 5. Der Stretch der Klickpfade, das heißt das Verhältnis der Länge der Klickpfade zu ihren kürzesten Distanzen im Graph, wurde in Abbildung 14 graphisch aufbereitet. Der durchschnittliche Stretch liegt hier immer über 1,5.

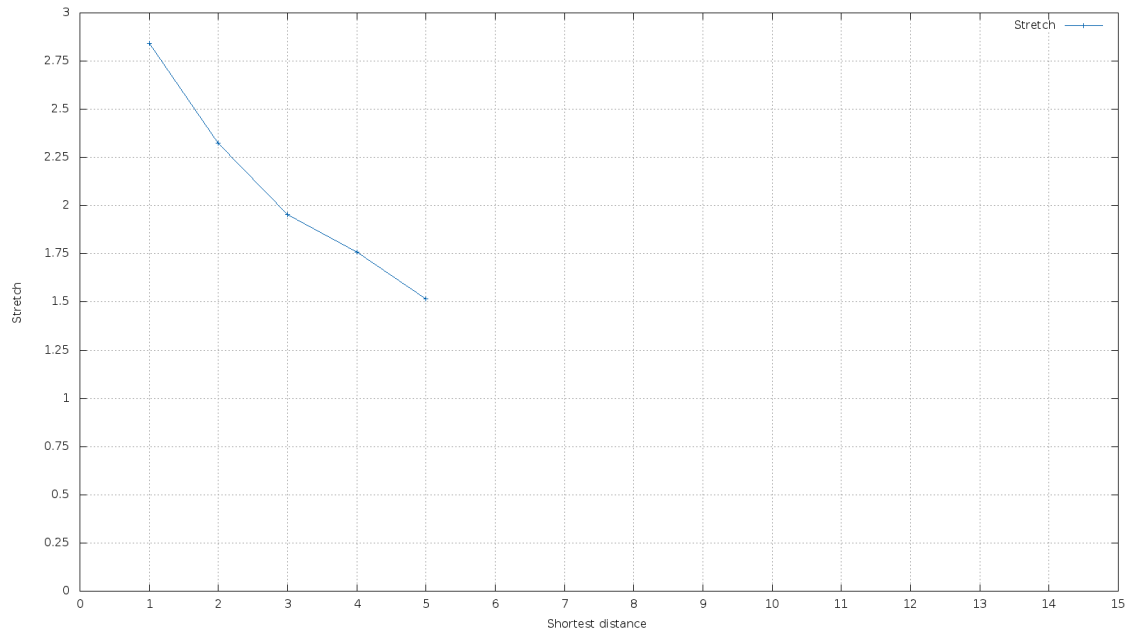


Abbildung 14: Stretch der Klickdaten

Bei den nicht erfolgreich gespielten Missionen ist deutlich zu erkennen, dass ab einer Länge von 4 viele Benutzer das Spiel aufgeben. Die Verteilung der Pfadlängen ist in Abbildung 15 zu sehen. Im Plot ist zu erkennen, dass Pfade mit der Länge 0 bereits herausgefiltert wurden, da diese keine Information über die menschliche Navigation beinhalten können.

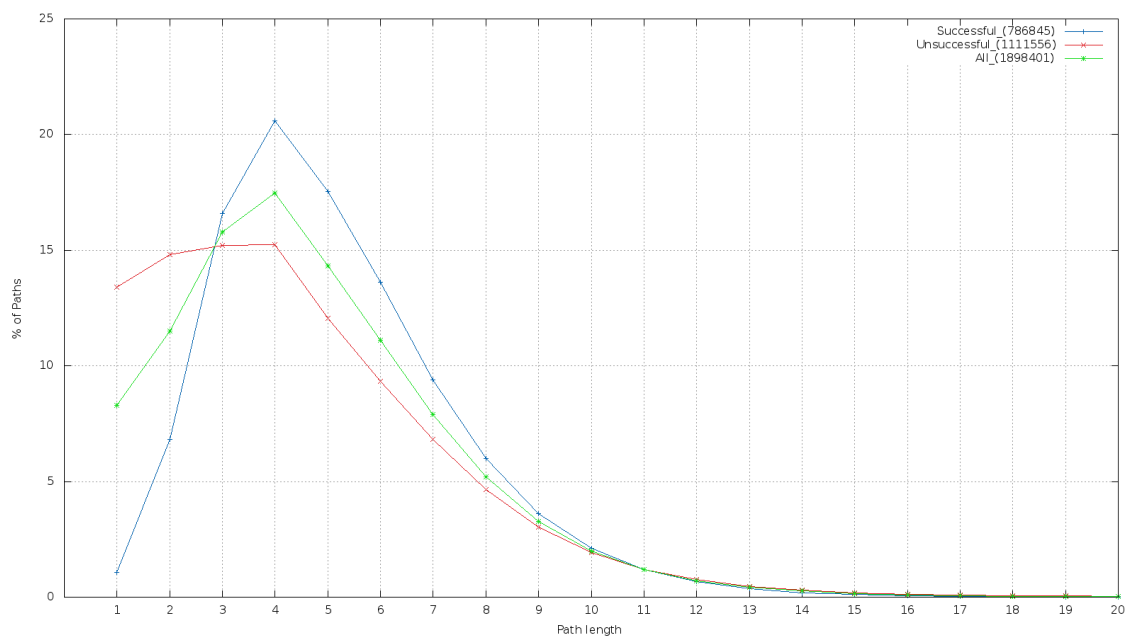


Abbildung 15: Verteilung von Pfadlängen in den Klickdaten

Ebenfalls wurde die Zoom-Out und Zoom-In Phase der Klickdaten genauer betrachtet. Daraus ergaben sich zwei verschiedene Plots.

Der Erste dieser beiden befasst sich mit dem durchschnittlichen Knotengrad in jedem Schritt der Spieldaten. Wie in Abbildung 16 zu sehen ist, starten die Benutzer in Schritt 0 mit einem durchschnittlich sehr geringen Knotengrad. Bezüglich des Eingangsgrades ändert dies sich aber in Schritt 1 und 2 enorm. Hier werden Knoten mit hohem Eingangsgrad bevorzugt, was exakt die Zoom-Out Phase widerspiegelt. Ab Schritt 2 verringert sich der durchschnittliche Eingangsgrad wieder rasch und flacht dann ab Schritt 6 etwas ab. Dies ist die bereits bekannte Zoom-In Phase der Benutzer. In der Grafik werden außerdem noch der durchschnittliche Grad, Ausgangsgrad und hierarchische Wert abgebildet. Hier fällt auf, dass der Eingangsgrad und der Grad die größten Unterschiede verzeichnen, wodurch diese sich für spätere Modellierungen am besten eignen.

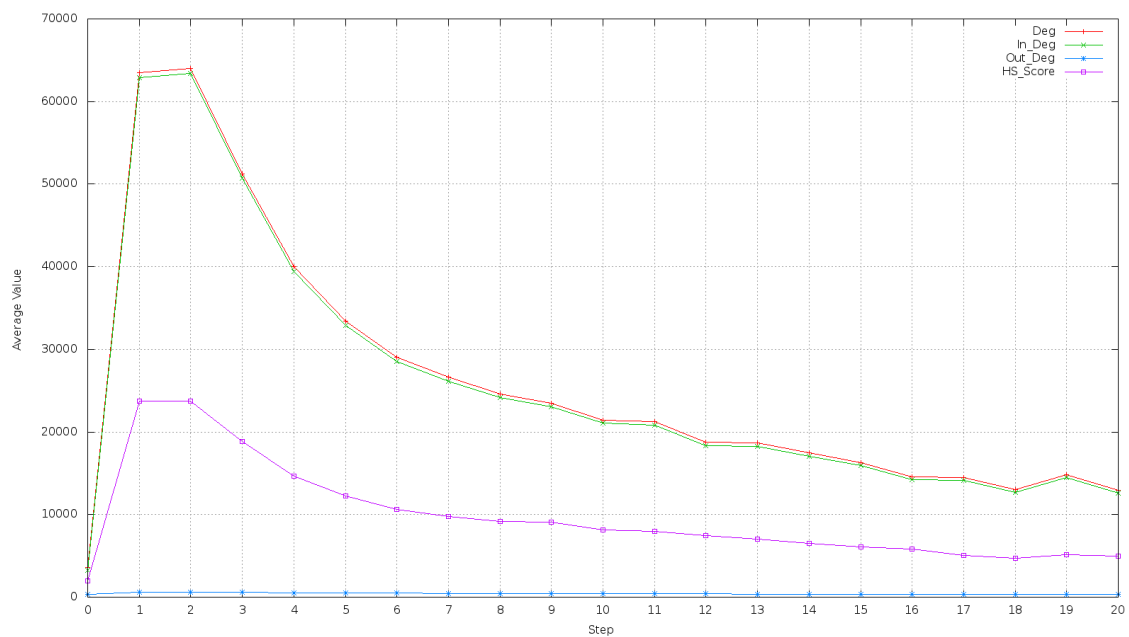


Abbildung 16: Durchschnittlicher Knotengrad pro Schritt der Klickdaten

Eine weitere interessante Visualisierung der Zoom-Out und Zoom-In Phase ist Abbildung 17. Die X-Achse kennzeichnet dabei den aktuellen Schritt in der Navigation. Die Y-Achse hingegen zeigt, ob dieser durch Hinaus- oder Hineinzoomen erreicht wurde. Hinauszoomen heißt, dass der vorherige Knoten einen geringeren Grad als der aktuelle besitzt. Auffallend ist hierbei der stark ausge-

prägte erste Schritt. Dieser entsteht durch einen durchschnittlich sehr geringen Grad des Startknotens. Der zweite Schritt verhält sich hier eher neutral, was auch aus Abbildung 16 hervorgeht, da hier der durchschnittlich gewählte Knotengrad ziemlich genau dem vorherigen entspricht. Ab Schritt drei beginnt die Zoom-In Phase. Diese erreicht bei Schritt vier den Höhepunkt und flacht anschließend wieder ab.

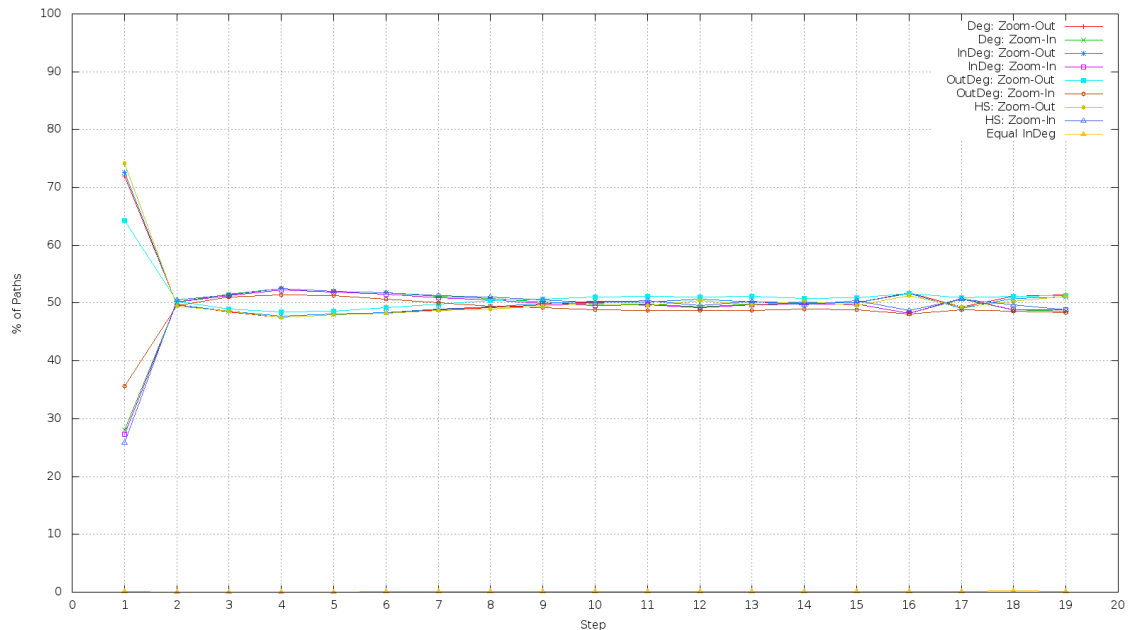


Abbildung 17: Zoom-Out und Zoom-In Phase der Klickdaten

9.2.1 Analyse mittels verschiedener Hierarchien

Um die Parameter für die Verwendung der Node-Selectoren möglichst gut abschätzen zu können, wurde, wie schon im letzten Kapitel genau beschrieben versucht, die verwendeten Strategien wiederherzustellen. Ob ein Klick als Greedy oder Random dabei erkannt wird, hängt ausschließlich von der Hierarchie ab. Horwath [6] hat die verwendeten Hierarchien mittels eines modifizierten Breiten-Such-Algorithmus erstellt und dabei verschiedene Knoten als Startpunkt verwendet. Für diese Ergebnisse wurden Hierarchien, ausgehend von den jeweils 3 Knoten mit dem höchsten Grad, Eingangsgrad, Ausgangsgrad und hierarchischen Wert verwendet. Da jedoch die meisten gleich schlecht abschnitten, werden hier nur ein schlechter und ein überdurchschnittlich guter präsentiert.

Abbildung 18 zeigt, dass diese Hierarchie sich nicht besonders gut für eine Greedy-Navigation eignet. Alle weiteren Hierarchien schnitten noch deutlich schlechter ab, außer der Hierarchie ausgehend vom Knoten mit dem zweit-höchsten Eingangsgrad.

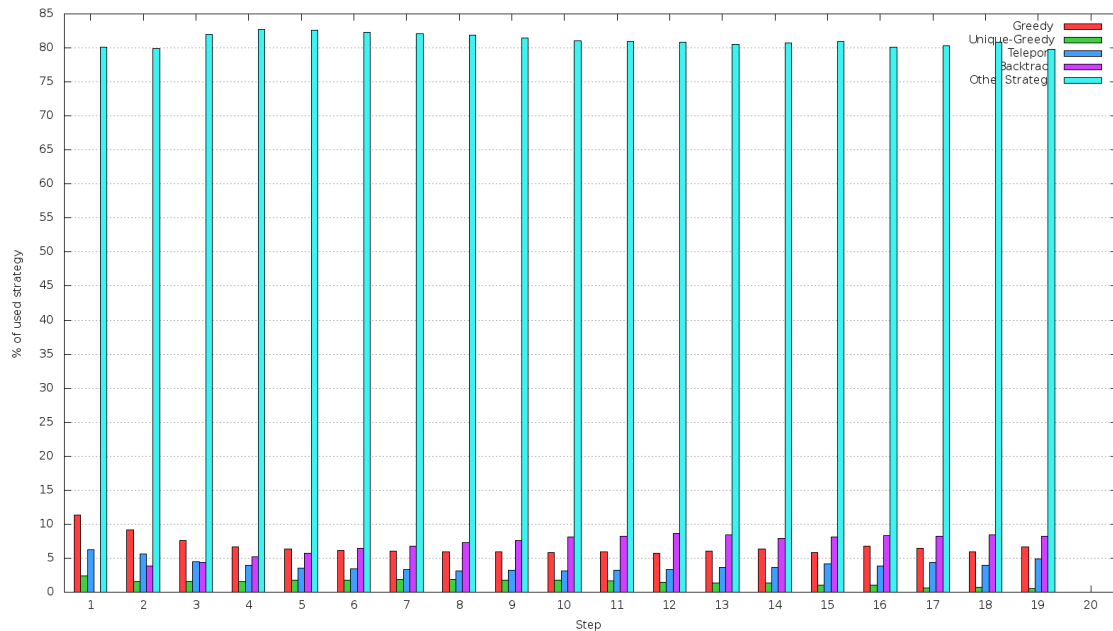


Abbildung 18: Analyse der verwendeten Strategien mit Top-In-Hierarchie

Diese zeigt eine deutlich bessere Abbildung des menschlichen Navigationsverhaltens. Im ersten Schritt bildet sie fast 35% richtig ab. Wird hierbei der Knoten, von welchem aus diese Hierarchie erstellt worden ist, genauer betrachtet, so gibt es dafür auch eine logische Erklärung. Wird die ID in den Artikelnamen aufgelöst, so stellt sich heraus, dass dieser Knoten den Artikel „United States of America“ repräsentiert. Da gerade in den ersten Schritten viele User diesen Knoten aufsuchen, ist er als Startknoten für eine mittels Breitensuche erstellte Hierarchie gut geeignet.

Wird die Tiefe der Hierarchie und die Verteilung der Knoten pro Level in dieser Hierarchie betrachtet, so fällt auf, dass diese extrem flach ist. Dies resultiert aus der Kombination einer Breitensuche mit einem enorm stark vernetzten Graphen. In Abbildung 20 ist festzustellen, dass sich fast alle Knoten auf Level 3 bis 5 befinden. Allein Level 4 beinhaltet über 2.000.000 Knoten. Das entspricht über 50%. Zu Vergleichszwecken wurden hier auch Hierarchien geplottet, welche pro Knoten 2,3 oder 4 Kinder besitzen.

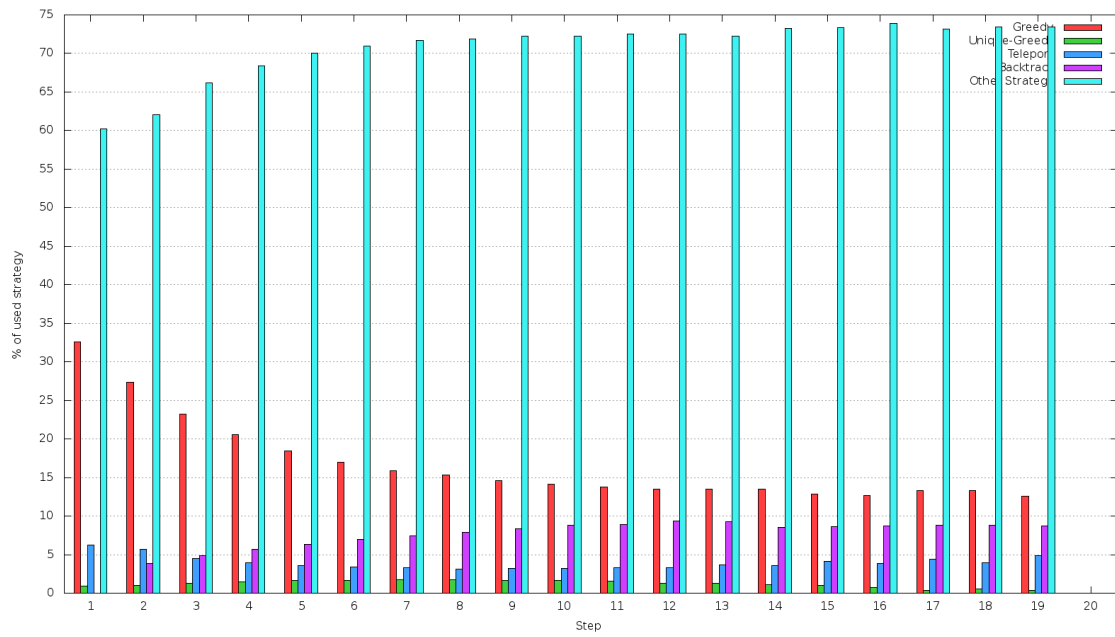


Abbildung 19: Analyse der verwendeten Strategien mit Top2-In-Hierarchie

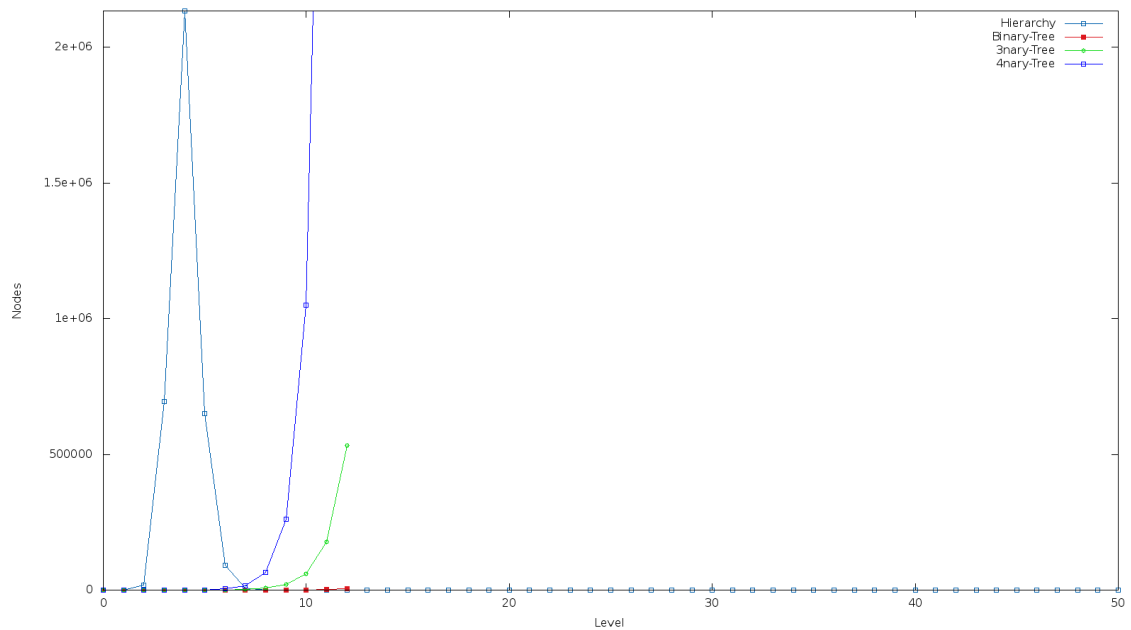


Abbildung 20: Knoten pro Level der Top2-In-Hierarchie

Durch diese Gegebenheit liegen enorm viele Knoten im selben Level, was wiederum zur Folge hat, dass ein Greedy-Algorithmus keine eindeutige Entscheidung treffen kann, sondern nur einen zufälligen Knoten aus den gleichbesten Nachbarn auswählt. Somit ist diese Hierarchie ausschließlich dadurch gut, dass viele Benutzer anfangs genau die Wurzel erreichen wollen. Betrachtet man die Unique-Greedy Balken in Abbildung 19, so stellt sich heraus, dass

nicht sehr viele Entscheidungen des Greedy-Node-Selectors einzigartig getroffen werden können und somit auch die Hierarchie sich nicht besonders gut eignet.

9.3 Baseline Random

Diese Einstellung des MUN-Frameworks legt die Grundlinie für alle weiteren Ergebnisse fest. Hier wird für jede Mission eine Simulation gestartet, welche einfach immer einen zufälligen Nachbarn auswählt. Folglich werden jeweils Random-Walks generiert. Wie in Abbildung 21 zu sehen ist, wurden pro Schritt ca. 5% aller erfolgreichen Pfade beendet. Hier muss beachtet werden, dass dies jedoch insgesamt nur 1325 Pfade sind. Dies entspricht nur 0,07% aller begonnenen Simulationen. Da bei allen Simulationen, welche mehr als 20 Schritte benötigten, abgebrochen wurde, ergibt sich in der Grafik der rasante Anstieg der nicht erfolgreichen Spiele im letzten Schritt. Der Abbruch bei Schritt 20 ist damit begründet, dass die Klickdaten nur wenige Pfade mit der Länge größer dieser Zahl beinhalten. Somit befinden sich alle relevanten Informationen in den ersten 20 Schritten.

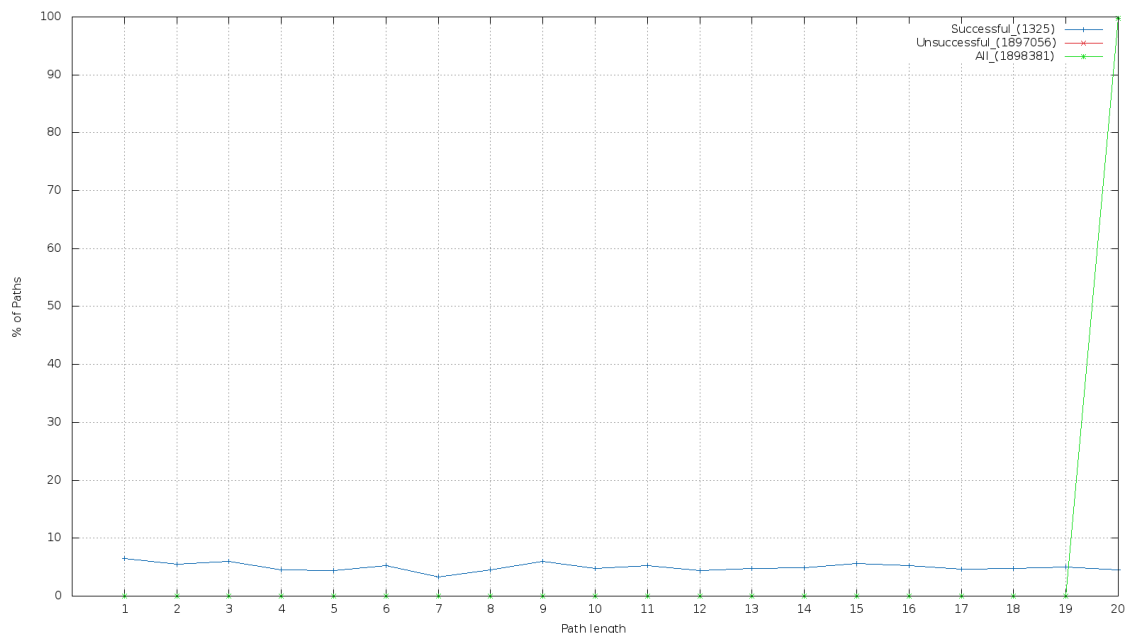


Abbildung 21: Pfadlängen der Random-Walks

Bezüglich der Zoom-Out und Zoom-In Phase von Random-Walks ist deutlich zu sehen, dass der Knotengrad der Startknoten sehr gering ist, da sogar bei zufälligem Auswählen von Nachbarn eine Zoom-out Phase im ersten Schritt entsteht. Anschließend bleiben die durchschnittlich besuchten Knotengrade jedoch sehr konstant. Der durchschnittliche Grad der besuchten Knoten in jedem Schritt ist in Abbildung 22 dargestellt.

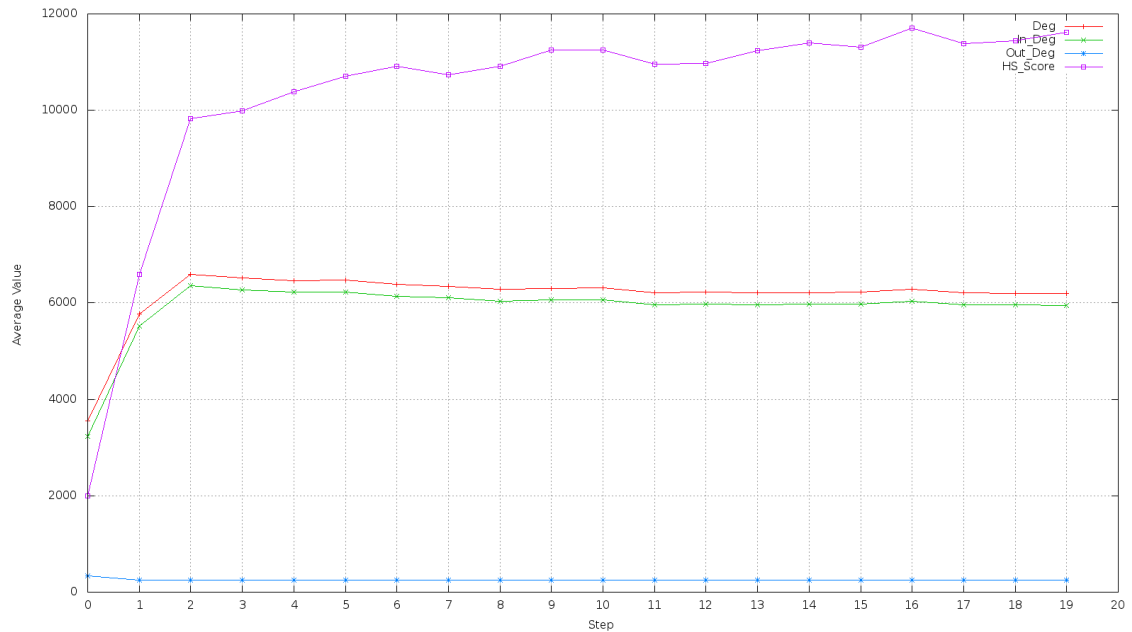


Abbildung 22: Durchschnittlicher Grad der Random-Walks

Abbildung 23 zeigt dieselben Ergebnisse, wobei auch hier die Zoom-Out Phase im ersten Schritt deutlich auffällt.

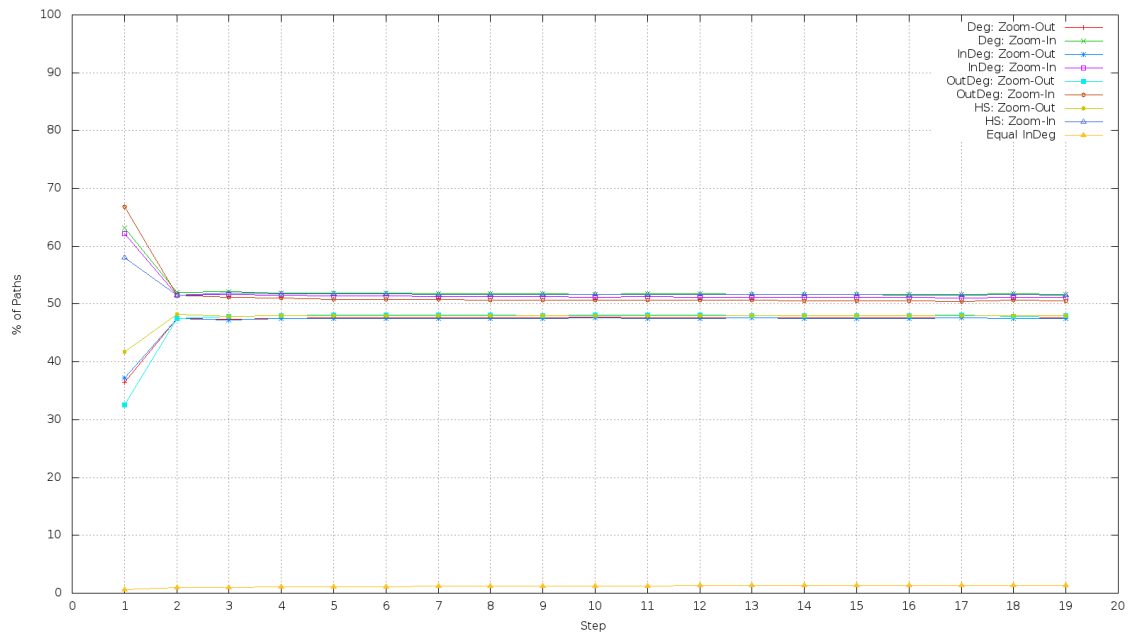


Abbildung 23: Zoom-Out und Zoom-In Phase der Random-Walks

Der Plot über die verwendete Strategie wird hier nicht dargestellt, da dieser in diesem Zusammenhang nicht von Bedeutung ist und außerdem ausschließlich aus „other strategy“-Balken besteht.

Im Vergleich mit den Klickdaten aus dem Wikigame resultiert hier, dass durchschnittlich nur 0,152% Prozent der Kanten eines generierten Pfades auch in einem von Benutzer produzierten Pfad vorkommen.

9.4 Greedy

Bei diesem Setup wurde in jedem Navigationsschritt ein Greedy-Algorithmus verwendet. Die Entscheidung, welcher der lukrativste nächste Knoten ist, wird dabei über die Distanzberechnung in der Hierarchie durchgeführt. Zusätzlich bevorzugt der Algorithmus noch nie zuvor besuchte Knoten, solange noch nicht alle Nachbarknoten besucht worden sind. Existieren zum Beispiel 10 Nachbarknoten, wovon 3 die kürzeste Distanz in der Hierarchie besitzen und diese alle schon zuvor besucht wurden, so ignoriert er diese und wählt den nächstbesten Knoten bezüglich der Distanz in der Hierarchie aus. Das heißt, er wählt den besten, noch nicht besuchten Knoten aus.

Zusätzlich wurde Backtracking verwendet, um den Algorithmus vor Sackgassen zu bewahren. Bevor ein Node-Selector aufgerufen wird, überprüft das Backtracking-Modul, ob schon alle Nachbarknoten besucht wurden. Ist dies der Fall, so geht es im bereits generierten Pfad so lange zurück, bis es einen noch nicht besuchten Nachbarn entdeckt. In Kombination mit dem Greedy-Algorithmus entspricht dieser Vorgang einer Tiefensuche, welche durch die Distanzen in der Hierarchie geleitet wird.

Da der Greedy-Algorithmus die Hierarchie als Hintergrundwissen verwendet, ist dieser auch komplett von ihr abhängig. Hier tritt nun wieder das schon in 9.2 erwähnte Problem auf. Der Greedy-Algorithmus will den lukrativsten Nachbarknoten auswählen, jedoch befinden sich in der Hierarchie über 50% aller Knoten auf demselben Level. Dies erhöht die Wahrscheinlichkeit, dass zwei Nachbarn dieselbe Distanz zum Ziel haben, enorm. Folglich kann der Algorithmus keine eindeutige Entscheidung mehr fällen und muss einen Knoten zufällig aus allen Nachbarknoten mit derselben kürzesten Distanz wählen.

Der Greedy-Algorithmus erzielt mit einer Hierarchie, welche mit einer Breiten-suche ausgehend von jenem Knoten im Graph, welcher den dritt größten Eingangsgrad besitzt, aufgebaut wurde, die besten Ergebnisse. Hierbei wurde eine Ähnlichkeit von 3,73% erzielt. Im Vergleich zum Random-Walk bedeutet dies eine 24-fache Verbesserung. Dieses Ergebnis klingt anfangs etwas ernüchternd, jedoch muss hier beachtet werden, welche Metrik für die Auswertung verwendet wurde und was es heißen würde, wenn man hier 100% erzielte. Dies würde bedeuten, dass in jedem Schritt der nächste Schritt des Benutzers vorhergesagt werden kann.

Des Weiteren muss beachtet werden, dass viele der generierten Pfade viel länger als die Benutzerpfade sind. Dies verringert wiederum die Prozentzahl der identischen Kanten. Würde der Navigator zum Beispiel 9 von 10 Schritten identisch zum Klickpfad eines Benutzers generieren und anschließend 11 falsche Schritte, so würde dies ein Ergebnis von unter 50% liefern.

Zudem wurden immer nur Pfade der gleichen Mission miteinander verglichen. Da in den Daten viele Missionen nur einmal von einem Benutzer gespielt wor-

den sind, bedeutet dies auch, dass der Navigator in diesem Fall genau diesen Benutzer modellieren müsste.

Interessant ist hier ebenfalls der Vergleich mit Ergebnissen der Hierarchien ausgehend von den Knoten mit dem höchsten und zweithöchsten Eingangsgrad. Erstere erzielt hier ein Ergebnis von 3,44%, wohingegen letztere nur 2,5% erreichte.

Für den durchschnittlichen Knotengrad in den Schritten erwies sich die Hierarchie ausgehend von „United States“ am ähnlichsten zu den Userdaten. Der durchschnittliche Grad der besuchten Knoten ist in Abbildung 24 geplottet.

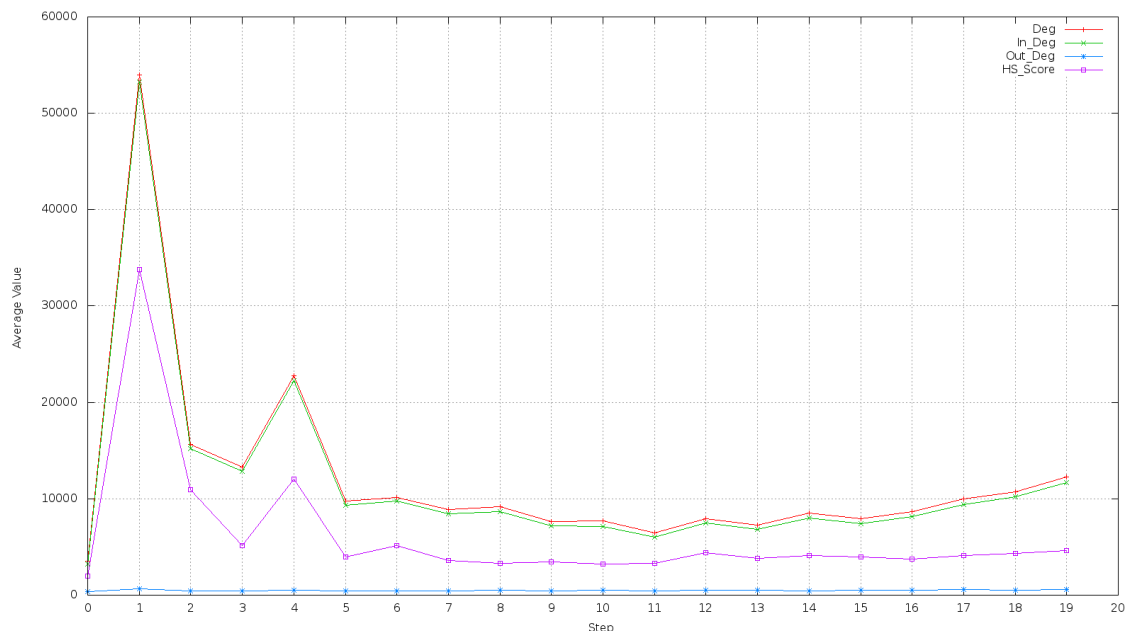


Abbildung 24: Durchschnittlicher Knotengrad der generierten Pfade

9.4.1 Unique-Greedy

Um zu erkennen, wie viele Navigationsschritte der Greedy-Algorithmus eindeutig auswählen konnte, muss der Plot über die verwendeten Strategien der generierten Pfade betrachtet werden. Abbildung 25 zeigt diesen Plot. Auch hier stellt sich wieder heraus, dass durch viele Nodes auf demselben Level in der Hierarchie, keine eindeutigen Entscheidungen getroffen werden können und somit die Navigation zum Großteil aus einem eingeschränkten Zufall besteht.

Eingeschränkter Zufall deswegen, weil er nur aus allen gleichbesten, nächsten Knoten zufällig auswählt.

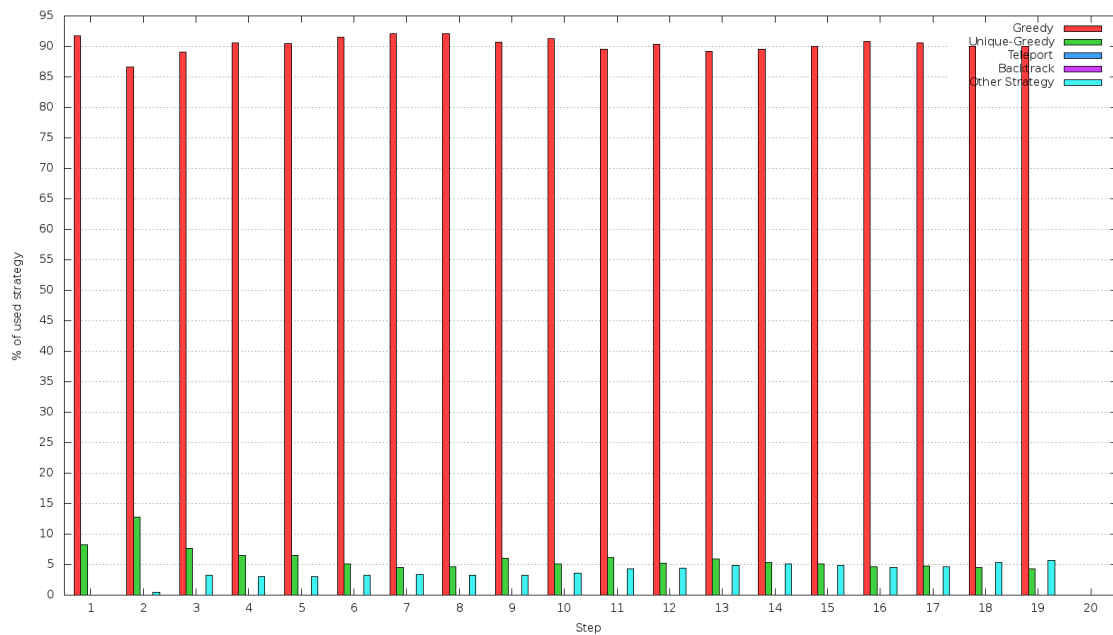


Abbildung 25: Strategien der generierten Pfade

9.5 Link-Restrictor

Die Konfiguration des Link-Restrictors erwies sich als äußerst schwierig. Da die Klickdaten eine Zoom-Out Phase bis zum zweiten Schritt, gefolgt von einer Zoom-In Phase, beinhalten, wurde auch der Link-Restrictor dahingehend eingestellt. Dies erwies sich jedoch als Fehler, wie in Abbildung 26 zu sehen ist. Der Greedy-Algorithmus konnte mit dieser Konfiguration ausschließlich aus Knoten mit extrem hohem Eingangsgrad auswählen, was die Zoom-Out Phase der Nutzer um ein Vielfaches überstieg. Auch durch die Kombination mit Backtracking entstand hier ein Zick-Zack-Muster im Zoom-In und Zoom-Out Plot der generierten Pfade. Der Link-Restrictor ließ für den Greedy-Algorithmus ausschließlich Knoten mit sehr geringem Eingangsgrad zur Auswahl übrig. Dadurch verirrte sich dieser oft und wurde durch Backtracking wieder zu vorherigen Knoten zurückgebracht. Diese wiederum besitzen einen viel höheren Eingangsgrad. Dies ist in Abbildung 27 zu sehen.

Werden die durchschnittlichen Eingangsgrade des Greedy-Algorithmus aus Abbildung 24 mit jenen der Benutzerdaten aus Abbildung 16 verglichen, so sind

im ersten Schritt deutliche Ähnlichkeiten zu erkennen. Durch diese Tatsache wurde der Link-Restrictor so konfiguriert, dass er erst ab Schritt 2 aktiv wird. Doch auch hier muss zuerst der Greedy-Algorithmus von zuvor mit den Nutzerdaten verglichen werden, um eine gute Annahme machen zu können. Hier fällt auf, dass der bereits vorhandene Algorithmus eine zu stark ausgeprägte Zoom-In Phase besitzt. Folglich ist es hier nicht zielführend, diese mit Hilfe des Link-Restrictors noch weiter zu verstärken. Das Gegenteil sollte hier versucht werden. Um dies zu erreichen, wurde die zweite Phase des Link-Restrictors so konfiguriert, dass dieser aus den möglichen, nächsten Knoten jene herausfiltert, welche den höchsten Eingangsgrad besitzen. Durch diese Einstellung des Link-Restrictors kann der Greedy-Algorithmus ab dem zweiten Schritt keine Knoten mit niedrigem Eingangsgrad mehr auswählen. Es wurde über lange Zeit versucht, die Parameter hier zu optimieren, jedoch konnte die erste Phase nicht gut angenähert werden.

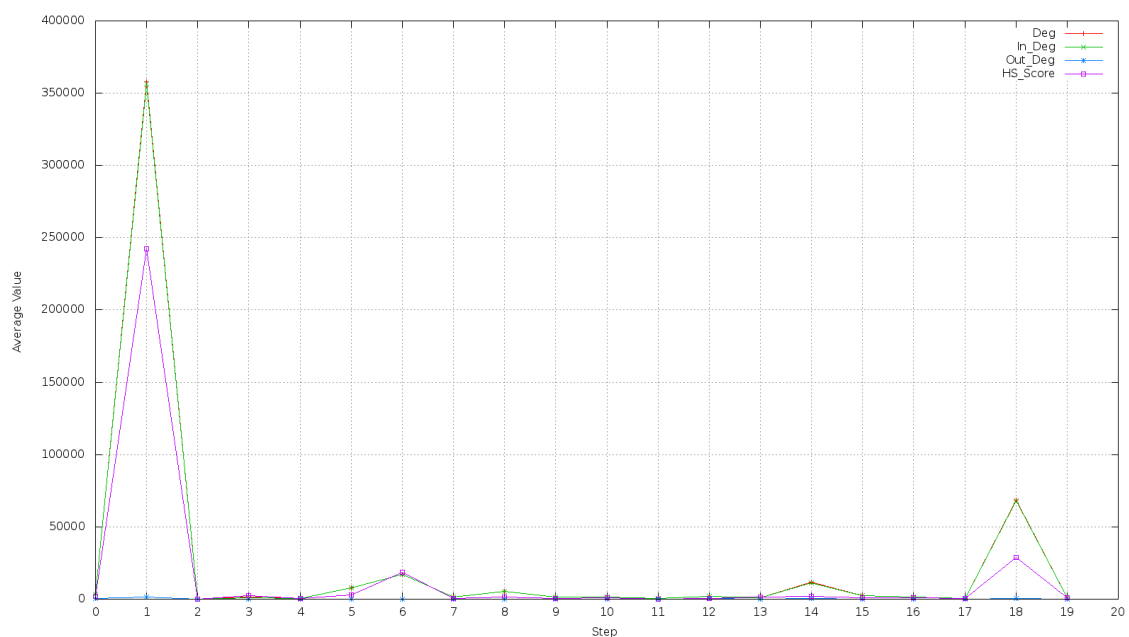


Abbildung 26: Verstärkte Zoom-Out und Zoom-In Phase

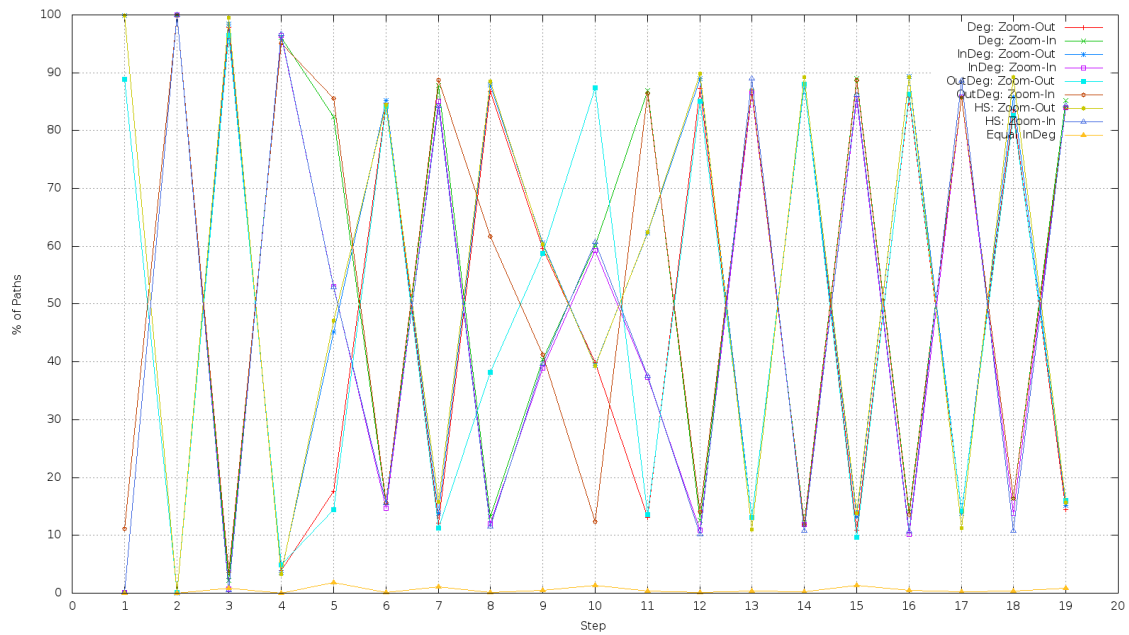


Abbildung 27: Zick-Zack-Muster im Zoom-Out und Zoom-In Plot

Durch etliche Versuche stellte sich heraus, dass die verwendete Hierarchie das Problem darstellt. In Zusammenarbeit mit Horwath [6] wurde an einer neuen Herstellungsmethode für Hierarchien gearbeitet. Daraus entstand eine Hierarchie, welche ebenfalls durch Breitensuche aufgebaut wurde, jedoch pro Knoten nur die 5 Nachbarknoten mit höchstem Eingangsgrad hinzufügte. Folglich wurden die Knoten des Graphen auf mehrere Level verteilt, wodurch auch eine bessere Adaption der Link-Restrictor Parameter möglich wurde. Dadurch konnte bis Schritt 5 eine gute Annäherung des durchschnittlichen Eingangsgrades zu den Klickdaten aus Abbildung 16 erreicht werden. Ab Schritt 5 können durch die Parameterangabe in Prozent jedoch die Zielknoten mit oft kleinem Eingangsgrad nicht mehr erfolgreich erreicht werden. Das Ergebnis ist in Abbildung 28 zu sehen.

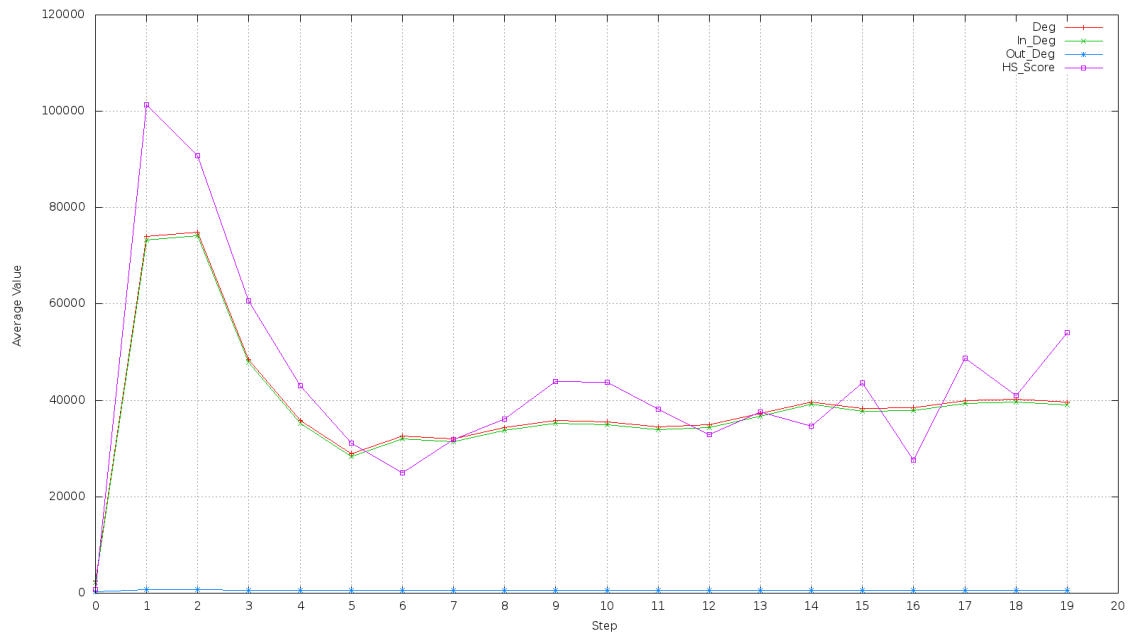


Abbildung 28: Durchschnittlicher Knoten Grad mit Link-Restrictor

9.6 Kombination Link-Restrictor mit Abbruchsimulator

Horwath [6] hat in seiner Arbeit einen Abbruchsimulator entwickelt. Durch Anpassung der dazugehörigen Parameter, wurde damit die Abbruchrate der Menschen sehr gut nachmodelliert. Interessant ist nun die Kombination dieses Features mit dem Link-Restrictor, da dies den Simulator deutlich an den Menschen anpassen sollte.

Der Abbruchsimulator lieferte mit einer quadratischen Annäherung, einer linearen Begrenzung und einer Abbruchresistenz von 0,96 die besten Ergebnisse. Details zu den Einstellungen des Abbruchsimulators finden sich in [6]. In Kombination mit der besten Link-Restrictor Konfiguration aus Kapitel 9.5 wurden Ergebnisse erzielt, welche den Eigenschaften des menschlichen Navigationsverhaltens schon sehr ähnlich scheinen. Es findet sich hier nun ein Vergleich aller Plots der Klickdaten mit denselben Plots der generierten Pfade. Links befinden sich immer die Klickdaten und rechts die generierten Pfade, wobei zu beachten ist, dass die Y-Achse nicht immer denselben Wertebereich besitzt. Auffallend ist hier, dass der durchschnittliche Knotengrad annähernd gleich ist, jedoch der hierarchische Wert sich deutlich unterscheidet. Dies deutet bereits auf Unstimmigkeiten hin, welche in Kapitel 9.7.2 aufgeklärt werden.

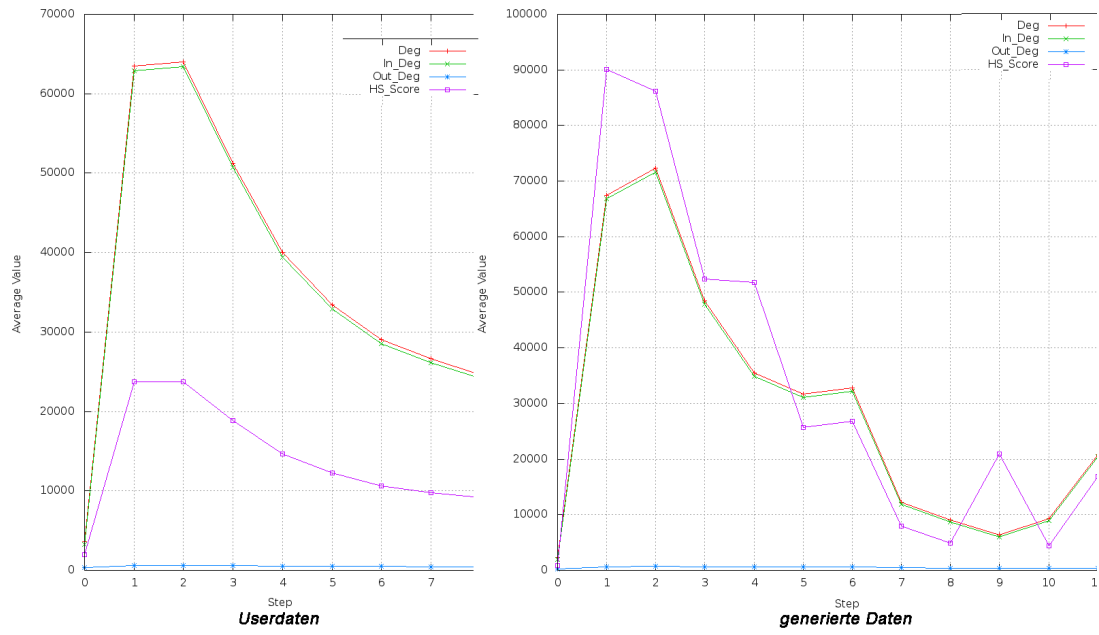


Abbildung 29: Vergleich der durchschnittlichen Grade

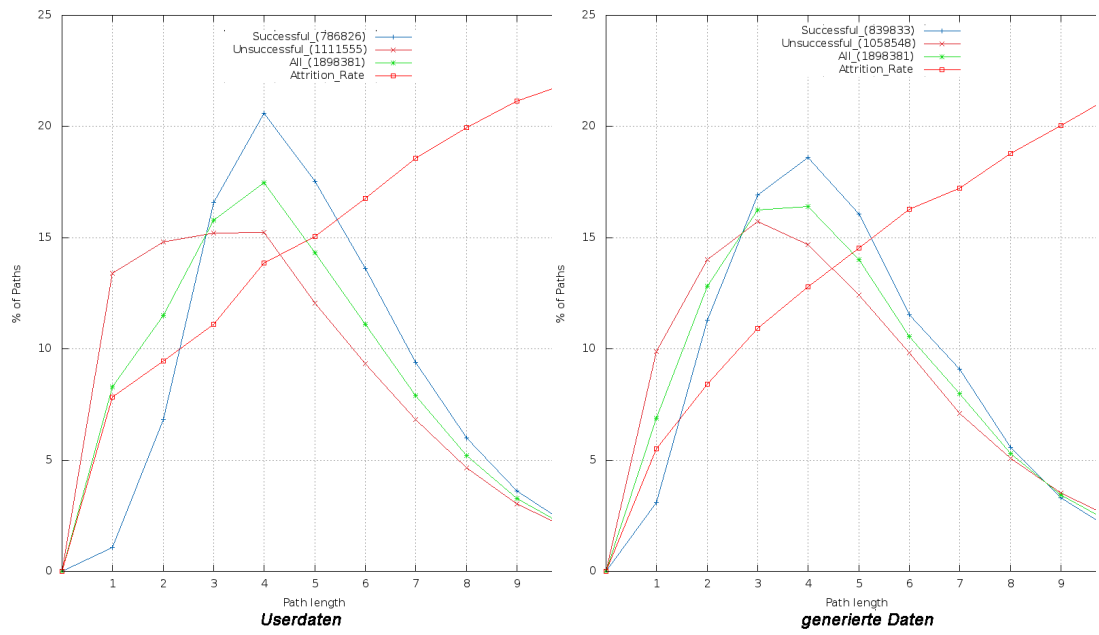


Abbildung 30: Vergleich der Abbruchrate und Längen

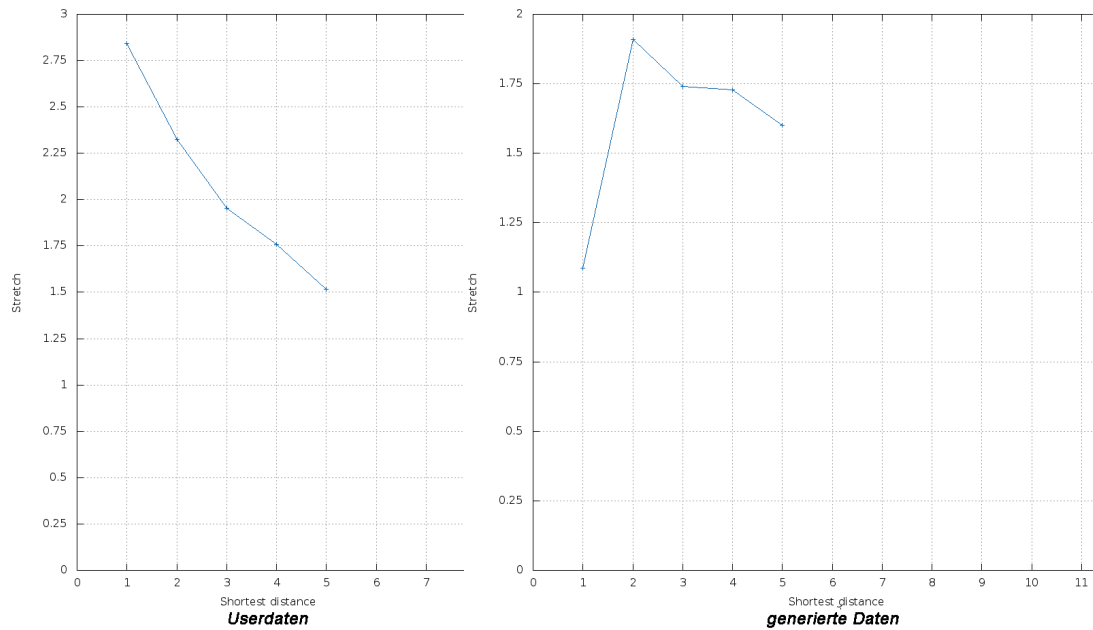


Abbildung 31: Vergleich des Stretch

Sehr gut ist hier auch das Verhältnis von nicht erfolgreichen Spielen zu erfolgreichen Spielen angepasst worden. In den Klickdaten beträgt dies 1,41. Die generierten Pfade besitzen hier ein Verhältnis von 1,26. Für das Testen von veränderten Netzwerkstrukturen spielt dies eine zentrale Rolle. Würde man das Netzwerk verändern und der Simulator dadurch ein geringeres Verhältnis erzielen, wäre diese Netzwerkveränderung wahrscheinlich auch für die User eine positive Veränderung. Des Weiteren wurde im direkten Pfadvergleich 4,73% erzielt, was eine deutliche Verbesserung ist. In Abbildung 30 ist ebenfalls zu erkennen, dass die Abbruchrate gut an die aus den Klickpfaden angepasst wurde.

Auffallend ist in Abbildung 31 jedoch, dass im ersten Schritt der Stretch der generierten Pfade viel geringer ist. Durch Verwendung eines Random-Node-Selectors im ersten Schritt in Kombination mit einem Link-Restriction, welcher 5,5% der möglichen, nächsten Knoten mit dem höchsten Eingangsgrad filtert, kann dies jedoch gut angenähert werden. Abbildung 32 zeigt hier den Vergleich. Links ist der Stretch zu sehen welcher durch eine reine Greedy-Navigation entstanden ist. Rechts wurde im ersten Schritt der Navigation mittels des Random Node-Selectors aus den möglichen, nächsten Knoten mit höch-

tem Eingangsgrad gewählt. Für die restlichen Schritte wurde der Greedy Node-Selector zur Navigation verwendet.

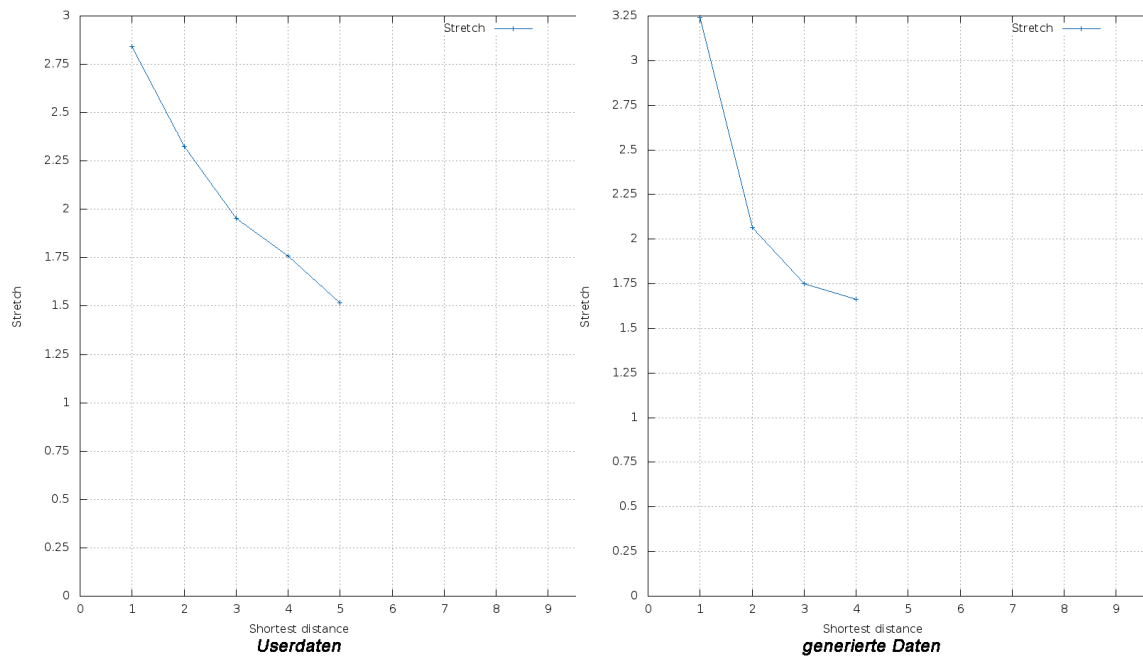


Abbildung 32: Vergleich des Stretch

Leider konnte der Zoom-Out und Zoom-In Plot nicht gut an die Klickdaten angepasst werden, was wiederum auf einen Unterschied in den Verteilungen der Knotengrade pro Schritt hindeutet. Es fällt hier auf, dass beide Phasen viel stärker ausgeprägt sind. Dadurch ist anzunehmen, dass die Einschränkungen des Link-Restrictors zu stark sind. Diese können jedoch nicht gelockert werden, da sonst, bedingt durch die Hierarchie, keine guten Ergebnisse erzielt werden. Abbildung 33 zeigt die Zoom-Out und Zoom-In Phase der generierten Pfade. Der Gipfel bei Schritt 7 ist durch die Deaktivierung des Link-Restrictors in diesem Schritt zu erklären.

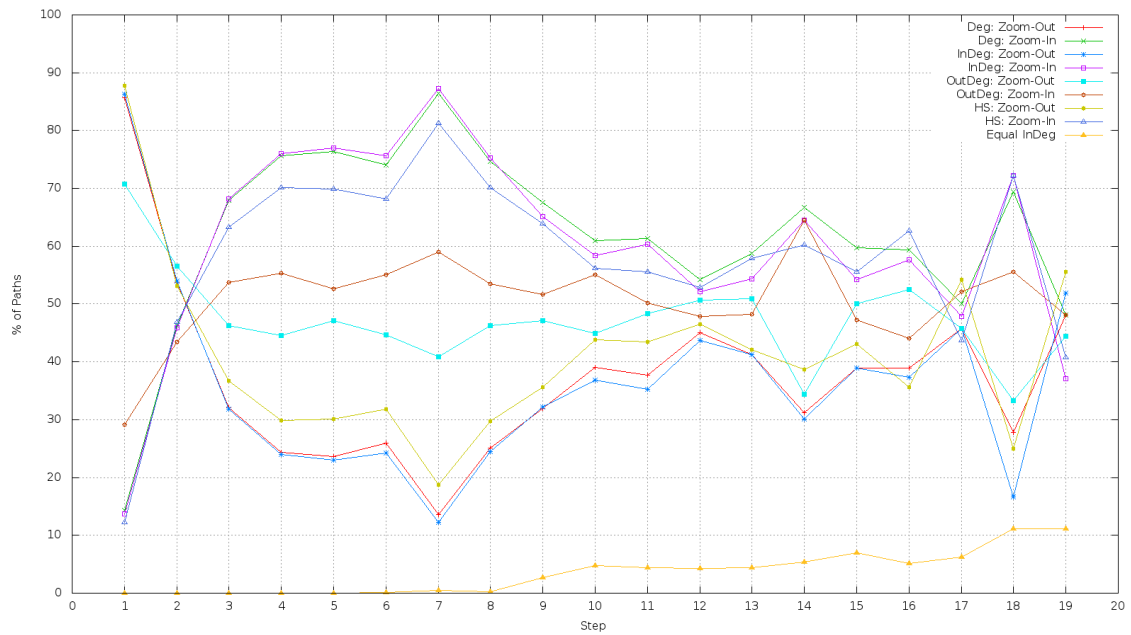


Abbildung 33: Zoom-Out und Zoom-In Phase der generierten Pfade

9.7 Kullback-Leibler-Divergenzen der Ergebnisse

Um die Ergebnisse besser vergleichen zu können und die Unstimmigkeit in Abbildung 29 bzw. die große Differenz von Abbildung 33 zu Abbildung 17 aufklären zu können, wurden von der Verteilung der Längen und den Verteilungen der Knotengrade in jedem Schritt die Kullback-Leibler-Divergenz zu den Klickpfaden berechnet. Die Formel hierfür lautet: $\sum_{x \in X} P(x) * \log \frac{P(x)}{Q(x)}$. Das Ergebnis zeigt die Unterschiedlichkeit der beiden Verteilungen, wobei hier gilt: Je niedriger desto besser. Die generierten Daten waren dabei immer das Modell, wohingegen die Benutzerdaten die Präzise Verteilung repräsentierten. Bei der Kullback-Leibler-Divergenz musste in dieser Arbeit sogenanntes smoothing verwendet werden, um auch Werte berücksichtigen zu können, welche in der anderen Verteilung 0 entsprechen.

	bl_greedy_top_in1 bl_greedy_top_deg1 bl_greedy_top_hs1	bl_greedy_top_in2 bl_greedy_top_deg2	bl_greedy_top_in3 bl_greedy_top_deg3	bl_greedy_top_out1	bl_greedy_top_out2	bl_greedy_top_out3
Lengths all	0,5963	0,7481	0,4298	1,337	0,891	1,1195
Lengths successf.	0,3155	0,3441	0,1932	0,2801	0,3216	0,4492
Lengths unsuccessf.	7,6429	8,8998	7,8284	6,3648	6,9086	7,8698
Deg. All	0,4308	0,7272	0,5247	1,0123	0,7083	1,0618
Deg. 1	0,5381	1,3099	0,7908	0,8448	0,7457	0,7911
Deg. 2	0,5975	0,9907	0,8364	0,9053	0,9331	1,1756
Deg. 3	0,6404	0,8323	0,7594	1,0148	0,9558	1,325
Deg. 4	0,6135	0,8799	0,7074	1,0773	0,9058	1,3292
Deg. 5	0,6091	0,9212	0,7389	1,1355	0,8853	1,3205
Deg. 6	0,6045	0,9658	0,7396	1,1985	0,8769	1,3328
Deg. 7	0,6138	0,9992	0,7462	1,2944	0,8872	1,3674
Deg. 8	0,6194	1,0324	0,7564	1,3732	0,8906	1,3937

	bl_greedy_top_hs2	bl_greedy_top_hs3	bl_greedy_nobacktop_in1	bl_random	bl_greedy_2hs	bl_greedy_3hs ~_4/5/6/10hs
Lengths all	0,7668	0,7285	0,5961	6,6053	1,8784	1,8623
Lengths successf.	0,3514	0,3553	0,3143	0,8401	0,5596	0,52
Lengths unsuccessf.	7,8957	8,5073	7,6339	6,9029	9,5047	10,9531
Deg. All	0,8311	0,649	0,4309	0,625	0,5775	0,5693
Deg. 1	1,0467	0,9845	0,5387	0,8714	0,5247	0,5412
Deg. 2	1,1384	0,9347	0,5934	0,9344	0,7514	0,7603
Deg. 3	1,1337	0,8079	0,6374	0,9238	0,8798	0,8331
Deg. 4	1,0819	0,8317	0,6152	0,8404	0,969	0,9164
Deg. 5	1,078	0,8444	0,6069	0,7374	0,9983	0,9455
Deg. 6	1,0937	0,8655	0,6053	0,6862	1,0271	0,9713
Deg. 7	1,1102	0,8964	0,6158	0,6515	1,0465	0,9896
Deg. 8	1,1325	0,9234	0,6202	0,6213	1,0453	0,9753

	bl_greedy_top_bfs5	bl_greedy_top_bfs10	attrition_bl_random	attrition_q096x12_top_bfs5	combi	combi_random
Lengths all	0,323	0,2471	0,1509	0,0108	0,0438	0,0437
Lengths successf.	0,1701	0,1206	0,456	0,0349	0,0561	0,056
Lengths unsuccessf.	6,4484	5,4261	0,19	0,0161	0,0419	0,0418
Deg. All	0,3075	0,3051	0,5549	0,3104	0,3853	0,3855
Deg. 1	0,5755	0,6045	0,6542	0,5729	0,5144	0,515
Deg. 2	0,7461	0,7563	0,6756	0,7444	1,2866	1,2879
Deg. 3	0,7068	0,6528	0,6462	0,7095	1,3399	1,3371
Deg. 4	0,564	0,4961	0,5486	0,5622	1,2564	1,2509
Deg. 5	0,4452	0,4301	0,4355	0,446	1,1381	1,1329
Deg. 6	0,3937	0,3855	0,361	0,3954	1,0671	1,0648
Deg. 7	0,3751	0,3683	0,3095	0,3734	0,5692	0,5742
Deg. 8	0,3773	0,3595	0,2687	0,3779	0,4583	0,455

Abbildung 34: Kullback-Leibler-Divergenzen

9.7.1 Längenverteilungen

Wie man in Abbildung 34 gut erkennen kann, wurden die Verteilungen der Längen sehr gut an die der Klickpfade angepasst. Eine genauere Analyse dazu findet sich in [6].

9.7.2 Verteilung der Knotengrade

Vergleicht man in Abbildung 34 die Kullback-Leibler-Divergenz der Knotengrade in den einzelnen Schritten der Kombinationsergebnisse (prefix: combi) mit denen der Abbruchsimulationsergebnisse (prefix: attrition), so erkennt man eindeutig, dass letztere ein besseres Ergebnis erzielen. Dies hängt mit dem

starken Eingriff des Link-Restrictor zusammen. In Abbildung 35 sind in der ersten Zeile die Verteilungen von Knotengraden der ersten 5 Schritte zu sehen. In der zweiten Zeile befinden sich dieselben Plots des Ergebnisses, welches durch Kombination von Abbruchsimulator und Link-Restrictor erzielt wurde. Rot sind in beiden Zeilen die Verteilungen der Benutzerdaten, wohingegen grün die Verteilungen der generierten Pfade darstellen. Beim Kombinationsergebnis ist deutlich zu erkennen, dass ab Schritt 2 der Link-Restrictor aktiviert wird und von dort an, der Navigator keine Knoten mit geringem Knotengrad mehr auswählen kann. Abbildung 36 zeigt hier vergrößert den fünften Schritt aus Zeile zwei. Man sieht hier eindeutig, dass nur mehr sehr wenig Knoten mit geringem Knotengrad ausgewählt werden, dafür jedoch zu viele mit hohem Knotengrad. Interessant ist auch, dass sich die Verteilung der Knotengrade in den Benutzerdaten über die Schritte hinweg kaum verändert.

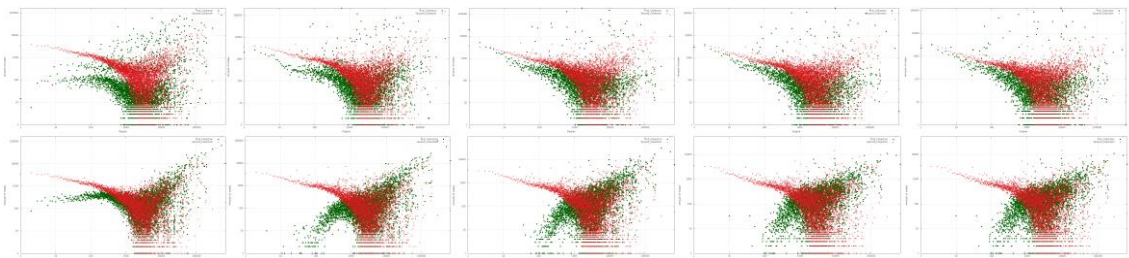


Abbildung 35: Vergleich der Verteilungen der Knotengrade

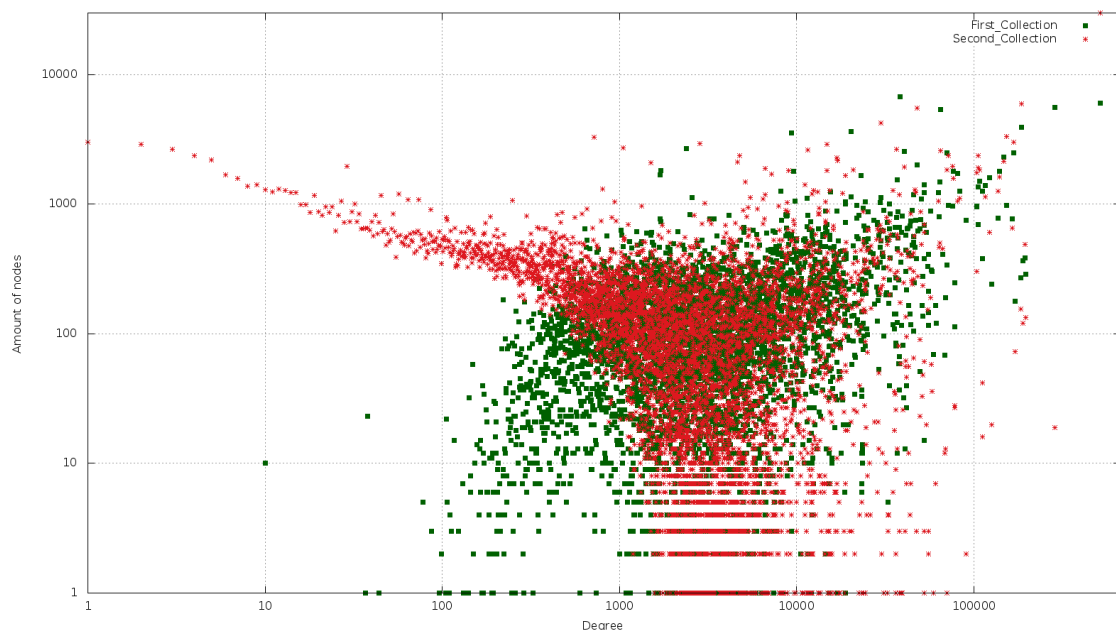


Abbildung 36: Verteilung der Knotengrade in Schritt 5

10. Lessons learned

Die Modellierung des menschlichen Navigationsverhaltens in Wikipedia-Netzwerken ist ein äußerst komplexes Themengebiet, welches noch in den Kinderschuhen steckt. Die Ergebnisse dieser Arbeit liefern einen deutlichen Hinweis darauf, dass ein Hintergrundwissen über die Netzwerkstruktur ausreicht, um relativ kurze Pfade generieren zu können, jedoch für die Modellierung menschlichen Verhaltens nicht gut geeignet ist.

Es wurden in dieser Arbeit einige bisherige Forschungsergebnisse mit einem vergleichsweise sehr großen Datensatz bestätigt und teilweise genauer spezifiziert. Des Weiteren entstand ein wertvolles Wissen über die Struktur von Hierarchien, welche sich für diese Zwecke gut eignen. Diese sollten ein gutes Verhältnis von Breite und Tiefe aufweisen und ihre Knoten möglichst gut auf die einzelnen Levels verteilt haben.

Das MUN-Framework bietet durch den Vergleich der Strategie-Plots eine gute Möglichkeit, um Hierarchien auf ihre Qualität für die Modellierung von gegebenen Klickdaten zu überprüfen. Mit Hilfe dieser Plots kann auch eine Aussage über die Eindeutigkeit der generierten Pfade getroffen werden. Folglich bietet es eine Bewertungsmethode für die Qualität der Hierarchie, falls diese einem Greedy-Algorithmus als Hintergrundwissen dienen soll. Außerdem ist es eine Maßanfertigung an die Bedürfnisse der Navigationssimulation in großen Netzwerken und bietet viele Konfigurationsmöglichkeiten, als auch Schnittstellen zur Erweiterung an. Die Konfiguration des Link-Restrictor und des Abbruchsimulator erwies sich als äußerst komplex, jedoch konnte zumindest durch den Abbruchsimulator eine deutliche Anpassung an die Nutzerdaten erzielt werden. Der Link-Restrictor hingegen zeigte, dass man den Navigator nicht strikt einschränken darf.

Schlussendlich ist zu sagen, dass viele wertvolle Resultate durch diese Arbeit entstanden sind, welche für weitere Forschungsarbeiten in diesem Themengebiet einen guten Grundstein bilden.

11. Future-Work

Dieses Kapitel beinhaltet Ideen, welche anhand der erzielten Ergebnisse entstanden sind und möglicherweise zielführend für zukünftige Arbeiten in diesem Themengebiet sind.

11.1 Hierarchien

Wie schon erwähnt, waren die Knoten in den verwendeten Hierarchien auf nur sehr wenige Levels verteilt. Dies hatte zur Folge, dass der Greedy-Algorithmus keine eindeutigen Entscheidungen treffen konnte und der Zufall mitspielte. Daher ist es in Zukunft sinnvoll, sich mit der Generierung von Hierarchien, welche das menschliche Navigationsverhalten besser abbilden, zu beschäftigen.

11.2 Modifikation des Greedy-Algorithmus

Eine weitere Idee ist die Modifikation des Greedy-Algorithmus dahingehend, dass dieser die nächsten, besten Knoten zum Teil anhand ihres Grades gewichtet. Dadurch würde er nicht ausschließlich immer den besten Knoten auswählen, was dem menschlichen Verhalten sicherlich eher entspricht, da der Mensch nicht alle Links einer Seite betrachtet, bevor er einen auswählt. Somit sollte nicht zwingend immer der lukrativeste, nächste Knoten ausgewählt werden.

11.3 Filterung der Vergleichsdaten

Ein zusätzlicher Vorteil wäre sicherlich auch eine weitere Filterung der Klickdaten. Hier wäre es sinnvoll, aus diesen nur Missionen auszuwählen, welche öfters als einmal gespielt wurden. Dadurch würde ein allgemeineres Benutzerverhalten entstehen und der Navigator hätte eine größere Chance, zumindest einen der Klickpfade zu modellieren.

11.4 Kommunikation Link-Restrictor und Node-Selector

Der Link-Restrictor könnte dahingehend modifiziert werden, dass er die nächsten Links gewichtet und diese Gewichte anschließend in die Entscheidung des Node-Selector einfließen.

11.5 Verteilung der Knotengrade

Wie man aus dem Plot der Verteilungen von Knotengraden in jedem Schritt gut erkennen kann, verändern sich diese bei den Userdaten kaum. Man könnte daher versuchen einen Node-Selector mit diesen Informationen zu gewichten und dadurch vielleicht eine Anpassung an das menschliche Verhalten erzielen.

11.6 Metriken

Im MUN-Framework steht auch ein Distanzenvergleich zur Verfügung. Dieser misst in jedem Schritt die durchschnittliche kürzeste Distanz zu einem Knoten aus einem Pfad der Klickdaten. Da hier jedoch für alle generierten Pfade pro Schritt eine Breitensuche ausgeführt werden muss, ist dies enorm rechenintensiv. Jedoch verkürzt sich die Rechenzeit je näher man den Simulator an die Userdaten annähert. Interessant wäre hier eine Auswertung der daraus resultierenden Daten, wodurch man erkennen könnte, in welchem Schritt der Navigator derzeit am schlechtesten den Menschen modelliert. Jedoch ist bis dato die Rechenzeit viel zu hoch.

Literaturverzeichnis

- [1] J. Waterworth und M. Chignell, „A Model of Information Exploration“, *Hypermedia* 3, pp. 35-58, 1991.
- [2] T. Alby, Web 2.0, München: Hanser Verlag, 2008.
- [3] J. Teevan, C. Alvarado, M. Ackerman und D. Karger, „The Perfect Search Engine Is Not Enough: A Study of Orienteering Behavior in Directed Search“, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Wien, 2004.
- [4] D. Gleich, P. Constantine, A. Flaxman und A. Gunawardana, „Tracking the Random Surfer: Empirically Measured Teleportation Parameters in PageRank“, in *Proceedings of the 19th international conference on World wide web*, Raleigh, 2010.
- [5] M. Eder, *Entwicklung eines Frameworks zur Navigationssimulation*, Graz: Technische Universität Graz, noch nicht veröffentlicht.
- [6] M. Horwath, *Datenaufbereitung und Entwicklung einer Abbruchsimulation zur Modellierung von Benutzernavigation*, Graz: Technische Universität Graz, noch nicht veröffentlicht.
- [7] M. E. J. Newman, *Networks: An Introduction*, Oxford: Oxford University Press, 2009.
- [8] D. Easley und J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge University Press, 2010.
- [9] S. Brin und L. Page, „The Anatomy of a Large-Scale Hypertextual Web Search Engine“, Stanford University, Stanford, 1998.
- [10] J. Kleinberg, „Authoritative sources in a hyperlinked environment“, *ACM*, pp. 604-632, 1999.
- [11] D. Helic, „Analyzing User Click Paths in a Wikipedia Navigation Game“, in *Proceedings of the 35th International Convention of Information Communication Technology, Electronics and Microelectronics*, Opatija, 2012.
- [12] P. Dodds, R. Muhamad und D. Watts, „An Experimental Study of Search in

- Global Social Networks“, *Science (Vol. 301)*, pp. 827-829, 2003.
- [13] „Wikipedia“, [Online]. Available: <http://www.wikipedia.org/>. [Zugriff am 10 1 2013].
- [14] P. Ayers, C. Matthews und B. Yates, *How Wikipedia works And How You Can Be a Part of It*, San Francisco: No Starch Press, 2008.
- [15] „MediaWiki“, [Online]. Available: http://www.mediawiki.org/wiki/Manual:What_is_MediaWiki%3F/de. [Zugriff am 12 1 2013].
- [16] „DiePresse“, [Online]. Available: <http://diepresse.com/home/techscience/internet/1330423/Erfundener-Krieg-stand-fuenf-Jahre-lang-auf-Wikipedia>. [Zugriff am 11 1 2013].
- [17] „DieWelt“, [Online]. Available: <http://www.welt.de/vermischtes/article112655248/Ausgedachter-Krieg-steht-jahrelang-auf-Wikipedia.html>. [Zugriff am 11 1 2013].
- [18] „Wikipedia's Liste der gefälschten Artikel“, [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia:List_of_hoaxes_on_Wikipedia. [Zugriff am 11 1 2013].
- [19] „Wikispeedia“, [Online]. Available: <http://www.cs.mcgill.ca/~rwest/wikispeedia/>. [Zugriff am 11 1 2013].
- [20] „WikipediaMaze“, [Online]. Available: <http://wikipediamaze.com/>. [Zugriff am 11 1 2013].
- [21] „TheWikigame“, [Online]. Available: <http://thewikigame.com/>. [Zugriff am 11 1 2013].
- [22] „Alex Clemesha's Twitter-Account“, [Online]. Available: <https://twitter.com/clemesha>. [Zugriff am 11 1 2013].
- [23] „TheWikiGame iPhone App“, [Online]. Available: <https://itunes.apple.com/at/app/wiki-game-wikipedia-game-racing/id459318432?mt=8>. [Zugriff am 11 1 2013].
- [24] B. Huberman, P. Pirolli, J. Pitkow und R. Lukose, „Strong Regularities in World Wide Web Surfing“, in *Proceedings of the seventh international conference on World Wide Web*, Amsterdam, 1998.

- [25] W. Fu und P. Pirolli, „SNIF-ACT: A Cognitive Model of User Navigation on the World Wide Web“, *Human-Computer Interaction (Vol. 22)*, pp. 355-412, 2007.
- [26] J. Voss, „Measuring Wikipedia“, in *Proceedings of the International Conference of the 10th International Society for Scientometrics and Informetrics*, Stockholm, 2005.
- [27] M. Boguñá, D. Krioukov und K. Claffy, „Navigability of complex networks“, *Nature Physics*, pp. 74-80, 2009.
- [28] R. West und J. Leskovec, „Human Wayfinding in Information Networks“, in *Proceedings of the 21st international conference on World Wide Web*, Lyon, 2012.
- [29] R. West und J. Leskovec, „Automatic versus Human Navigation in Information Networks“, in *Proceedings of the 23rd ACM conference on Hypertext and social media*, Milwaukee, 2012.

Abbildungsverzeichnis

Abbildung 1: Projektübersicht.....	4
Abbildung 2: Graphentheroie Teil 1.....	6
Abbildung 3: Graphentheorie Teil 2.....	8
Abbildung 4: Graphentheorie Teil 3.....	9
Abbildung 5: Graphentheorie Teil 4.....	10
Abbildung 6: Graphentheorie Teil 5.....	11
Abbildung 7: Wikipedia-Artikel der Technischen Universität Graz.....	15
Abbildung 8: Wikigame	20
Abbildung 9: Wikigame Statistik.....	23
Abbildung 10: Zoom-In und Zoom-Out Phase der Wikigame-Klickdaten	43
Abbildung 11: Verteilung der Knoteneingangsgrade	52
Abbildung 12: Verteilung der Knotenausgangsgrade	52
Abbildung 13: Verteilung der kürzesten Distanzen aller Missionen.....	53
Abbildung 14: Stretch der Klickdaten	55
Abbildung 15: Verteilung von Pfadlängen in den Klickdaten	55
Abbildung 16: Durchschnittlicher Knotengrad pro Schritt der Klickdaten	56
Abbildung 17: Zoom-Out und Zoom-In Phase der Klickdaten	57
Abbildung 18: Analyse der verwendeten Strategien mit Top-In-Hierarchie	58
Abbildung 19: Analyse der verwendeten Strategien mit Top2-In-Hierarchie	59
Abbildung 20: Knoten pro Level der Top2-In-Hierarchie	59
Abbildung 21: Pfadlängen der Random-Walks.....	60
Abbildung 22: Durchschnittlicher Grad der Random-Walks	61
Abbildung 23: Zoom-Out und Zoom-In Phase der Random-Walks	62
Abbildung 24: Durchschnittlicher Knotengrad der generierten Pfade	64
Abbildung 25: Strategien der generierten Pfade	65

Abbildung 26: Verstärkte Zoom-Out und Zoom-In Phase	66
Abbildung 27: Zick-Zack-Muster im Zoom-Out und Zoom-In Plot	67
Abbildung 28: Durchschnittlicher Knoten Grad mit Link-Restrictor.....	68
Abbildung 29: Vergleich der durchschnittlichen Grade.....	69
Abbildung 30: Vergleich der Abbruchrate und Längen	69
Abbildung 31: Vergleich des Stretch	70
Abbildung 32: Vergleich des Stretch	71
Abbildung 33: Zoom-Out und Zoom-In Phase der generierten Pfade	72
Abbildung 34: Kullback-Leibler-Divergenzen	73
Abbildung 35: Vergleich der Verteilungen der Knotengrade	74
Abbildung 36: Verteilung der Knotengrade in Schritt 5.....	74

Listingverzeichnis

Listing 1: Edgelist eines simplen Graphen	35
Listing 2: Klickpfade in einem simplen Graphen.....	36
Listing 3: Darstellung von Spielaufgaben mittels ID-Paaren	36
Listing 4: Beispiel einer Kürzesten-Distanzen Datei.....	37
Listing 5: Ausgabe der Einstellungen des Link-Restrictors	46

Tabellenverzeichnis

Tabelle 1: Wichtige Knoten des Wikipediagraphen	54
---	----

Abkürzungsverzeichnis

bzw.	beziehungsweise
ca.	zirka
d.h.	das heißt
et al.	et alii / und andere
F	False / falsch
GB	Gigabyte
ID	Identifikationsnummer
MUN	Modeling User Navigation
pp.	Pages / Seiten
SNAP	Stanford Network Analysis Platform
T	True / wahr
u.a.	unter anderem
z.B.	zum Beispiel