Fachhochschule
Bonn-Rhein-Sieg

University of Applied Sciences

Fachbereich Informatik
Department of Computer Science

Fraunhofer Institut
Intelligente Analyse- und
Informationssysteme

Abschlussarbeit
im Studiengang
Master of Science in Computer Science

Dynamic Real-time Interactive Clustering

Verbesserung von Clustering-Verfahren durch Dynamik und Interaktion

von
Stefan Winarzki

Erstbetreuer: Prof. Dr. Peter Becker
Zweitbetreuer: Prof. Dr. Rudolf Berrendorf

Externer Betreuer: Dipl.-Inform. Matthias Krauß

Eingereicht am:

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbst angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Unterschrift Sankt Augustin, den

Danksagung

Ich möchte mich bei Prof. Dr. Peter Becker und Prof. Dr. Rudolf Berrendorf für die Betreuung dieser Abschlussarbeit und viele aufschlussreiche Gespräche bedanken. Auch gilt mein Dank allen Kollegen der Abteilung Virtual Environments des Fraunhofer Instituts IAIS, insbesondere Dr. Manfred Bogen und Matthias Krauß. Des Weiteren möchte ich meiner Freundin Alex sowie meiner gesamten Familie für ihren stetigen Rückhalt und ihre Unterstützung danken.

Inhaltsverzeichnis

1 Einführung **1**
1.1 Einleitung . . . 1
1.2 Argunaut . . . 2
1.3 Ziel der Master Thesis . . . 3
1.4 Aufbau der Master Thesis . . . 4

2 Thematische Einordnung **6**
2.1 Konkretisierung des Problems . . . 6
2.1.1 Mustererkennung kombiniert mit dynamischer Interaktion . . . 6
2.1.2 Visualisierung von Argumentationsstrukturen . . . 8
2.1.3 Textanalyse . . . 9
2.2 Ähnliche Arbeiten . . . 9
2.2.1 DOCE-Algorithmus . . . 9
2.2.2 ScholOnto . . . 12
2.3 Verallgemeinerung der Problemstellung . . . 15

3 Verwendete Techniken **16**
3.1 Text Mining . . . 16
3.1.1 Statistische Verfahren . . . 17
3.1.2 Dokumentenähnlichkeit . . . 17
3.1.3 Anwendung . . . 19
3.2 Soziale Netzwerkanalyse . . . 20
3.2.1 Strukturanalyse-Verfahren . . . 20
3.2.2 Anwendung . . . 22
3.3 Cluster-Analyse . . . 24
3.3.1 Hierarchische Verfahren . . . 24
3.3.2 Nicht-Hierarchische Verfahren . . . 25
3.3.3 Selbstorganisierende Karten . . . 28
3.3.4 Anwendung . . . 31

4 Interaktionstechniken **33**
4.1 Interaktion und Clustering . . . 33
4.2 Visualisierung der Clustering-Ergebnisse . . . 34
4.3 Bewertungskriterien . . . 35
4.3.1 Aufgabenangemessenheit . . . 36
4.3.2 Selbstbeschreibungsfähigkeit . . . 36
4.3.3 Erwartungskonformität . . . 36
4.3.4 Lernförderlichkeit . . . 36
4.3.5 Steuerbarkeit . . . 37
4.3.6 Fehlertoleranz . . . 37
4.3.7 Individualisierbarkeit . . . 37
4.4 Evaluation der Kriterien . . . 38
4.4.1 Verschieben eines einzelnen Diskussionsbeitrages . . . 38
4.4.2 Verschieben mehrerer Diskussionsbeiträge . . . 40
4.4.3 Markierung ähnlicher Diskussionsbeiträge . . . 42
4.4.4 Manuelle Modifikation des Gewichtungsvektors . . . 44
4.4.5 Angabe eines Schlüsselparameters . . . 46
4.4.6 Angabe der wichtigsten Parameter . . . 48

4.4.7 History mit UNDO-Funktion 49
4.4.8 History mit wahlfreier UNDO-Funktion 51
4.5 Zusammenfassung 53

5 Konzept und Implementierung **54**
5.1 Grundkonzept 54
5.1.1 Mathematische Betrachtung 54
5.1.2 Ablaufplan 54
5.1.3 Einbettung in das Moderator's Interface 55
5.2 Clustering-Parameter 56
5.3 Interaktionstechniken 60
5.3.1 Angabe eines Schlüsselparameters 60
5.3.2 Angabe der wichtigsten Parameter 61
5.3.3 Manuelle Modifikation des Gewichtungsvektors 62
5.3.4 Markierung ähnlicher Diskussionsbeiträge 62
5.4 C-Means-Clustering 66
5.4.1 Bestimmung von C 66
5.4.2 Platzierung der Prototypen 67
5.4.3 Gruppierung der Diskussionsbeiträge 67
5.5 Visualisierung 71
5.6 Verwendete Software 72
5.6.1 Java Frameworks 72

6 Evaluation **74**
6.1 Validierung der Clustering Ergebnisse 74
6.2 Empirische Benutzerstudie des interaktiven Verfahrens 77
6.2.1 Forschungsdesign 78
6.2.2 Auswertung 79

7 Zusammenfassung und Ausblick **83**

Literaturverzeichnis **86**

A Empirische Benutzerstudie **87**
A.1 Freitext-Antworten 87
A.2 Fragebogen 87

Abbildungsverzeichnis

1 Internet-Forendiskussion mit typischer Baumstruktur. 1
2 Graphbasierte Diskussion im Argunaut Moderator's Interface. 2
3 Beispiel für einen interaktiven Clustering-Algorithmus. 7
4 Beispiel für Diskussionsstrukturen. 9
5 DOCE-Algorithmus. Quelle: [Mik07, S.21] 10
6 Ablauf des DOCE-Algorithmus. Quelle: [Mik07, S.30] 11
7 Beispielhafte ClaiMaker Diskussion. Quelle: [KSC03, S.194] 12
8 Beispiel für vereinfachte Graph-Visualisierung. Quelle: [KSC03, S.197] . . . 13
9 Visualisierung der Ergebnisse einer Abstammungsanalyse. Quelle: [KSC03, S.201] 14
10 Vier Phasen des interaktiven Clustering-Verfahrens. 16
11 Beispiel für den Ablauf einer Positions-Analyse. Quelle: [WF94, S.364] . . . 23
12 Verschiedene Abstandsmaße und Dendogramm. Quelle: [Run00, S.75] . . . 25
13 Netzstruktur der lernenden Vektorquantisierung. Quelle: [Zel94, S.172] . . 29
14 Netzstruktur der selbstorganisierenden Karten. Quelle: [Zel94, S.180] . . . 29
15 Modifikation der zweidimensionalen Gewichtsvektoren der Neuronen einer selbstorgansierenden Karte während des Lernvorgangs. Quelle: [Zel94, S.182] 31
16 Vier Phasen des interaktiven Clustering-Verfahrens. 33
17 Beispiel für die Anpassung des Gewichtungsvektors durch eine bestimmte Benutzerinteraktion. 34
18 Verschieben eines einzelnen Diskussionsbeitrages. 39
19 Verschieben mehrerer Diskussionsbeiträge. 41
20 Markierung ähnlicher Diskussionsbeiträge. 43
21 Manuelle Modifikation des Gewichtungsvektors. 45
22 Angabe eines Schlüsselparameters. 47
23 Angabe der wichtigsten Parameter. 48
24 History mit UNDO-Funktion. 50
25 History mit wahlfreier UNDO-Funktion. 52
26 Vier Phasen des interaktiven Clustering-Verfahrens. 55
27 Ausschnitt des Moderator's Interface (Screenshot). 55
28 Komponentendiagramm zur Plugin Struktur des Moderator's Interface. . . . 56
29 PreProcessing: Übersicht der Klassen der Vorverarbeitung. 59
30 Klassendiagramm: *InteractiveClusteringPluginPanel*. 60
31 Angabe eines Schlüsselparameters (Screenshot). 61
32 Angabe der wichtigsten Parameter (Screenshot). 62
33 Manuelle Modifikation des Gewichtungsvektors (Screenshot). 63
34 Markierung ähnlicher Diskussionsbeiträge (Screenshot). 66
35 Gewichtungsvektor: $[\,0.00,\ 0.39,\ 1.00,\ 0.26,\ 0.12,\ 0.60,\ 0.11\,]$ 69
36 Gewichtungsvektor: $[\,0.21,\ 1.00,\ 0.13,\ 0.12,\ 0.17,\ 0.00,\ 0.17\,]$ 69
37 Klassenübersicht: *CMeansHighDimensionalClustering* ist die Hauptklasse der Clustering-Anwendung und verbindet alle wichtigen Komponenten. 70
38 Anwendung der *Springy Physics* Visualisierung. 72

Zusammenfassung

Graphbasierte Diskussionen sind eine Form von Online-Diskussionen, bei denen eine Diskussion als Graph visualisiert wird. Beispielhafte Diskussionsanwendungen sind unter anderem Belvedere [SWCP95], FreeStyler [Gas03] oder Digalo [LK06]. Graphen dieser Art sind, was bestimmte Eigenschaften betrifft, vergleichbar mit Petri-Netzen [Pet62]. So gibt es bei Beiden gewichtete, gerichtete Kanten sowie Knoten verschiedenen Typs, die jeweils bestimmte Eigenschaften besitzen. Im Gegensatz zu einem Petri-Netz, das immer ein bipartiter Graph ist, können bei einem Diskussionsgraphen jedoch prinzipiell alle Knoten miteinander verbunden werden. Moderatoren solcher Diskussionen sind oftmals mit dem Problem konfrontiert, dass sie mehrere Diskussionen gleichzeitig beobachten wollen, was jedoch aufgrund der Komplexität der Struktur von Diskussionsgraphen kaum effizient möglich ist.

In dieser Abschlussarbeit wurde mit Techniken aus den Bereichen Cluster-Analyse, Text Mining und soziale Netzwerkanalyse ein interaktives Gruppierungsverfahren implementiert, das einen Moderator bei der Lösung dieses Problems unterstützen soll. Dabei kann der Moderator dem Verfahren auf intuitive Art und Weise mitteilen, nach welchen Kriterien der Diskussionsgraph gruppiert werden soll. Die Ergebnisse sollen dem Moderator eine bessere Übersicht der Diskussion bieten und eine einfachere Extraktion von Informationen ermöglichen. Sie werden in Echtzeit visualisiert und lassen eine weitere Modifikation zu, so dass sie in mehreren Schritten verfeinert werden können.

Die Komplexität dieser Arbeit besteht darin, die Interaktionsmöglichkeiten so zu entwerfen, dass der Moderator damit auf einfache Art und Weise für ihn sinnvolle Ergebnisse der Cluster-Analyse generieren kann. Dazu muss einerseits beachtet werden, dass die Interaktionstechniken interaktiv und einfach zu bedienen sind und andererseits muss der Clustering-Algorithmus so gut auf die Benutzereingabe reagieren, dass die Ergebnisse möglichst nahe an die Intention des Benutzers heranreichen.

Es wurden mehrere Interaktionstechniken implementiert, die für den Moderator intuitiv zu benutzen sind und dennoch präzise Rückschlüsse für den Algorithmus zulassen, nach welchen Kriterien eine Gruppierung vorgenommen werden soll. Wie intuitiv und effektiv die Interaktionstechniken zur Steuerung der Cluster-Analyse von Benutzern eingeschätzt werden, wird mit einer stichprobenartigen Benutzerstudie ausgewertet. Dabei wird ebenfalls bewertet, ob und wie sehr die Ergebnisse des interaktiven Clustering-Verfahrens dem Moderator bei der Beobachtung einer Diskussion helfen.

1 Einführung

In diesem Einführungskapitel werden zunächst die Rahmenbedingungen der Master Thesis vorgestellt, sowie deren Zielsetzung. Am Ende wird ein kurzer Überblick über den Aufbau der Abschlussarbeit gegeben.

1.1 Einleitung

Der Schwerpunkt dieser Abschlussarbeit ist die interaktive Steuerung und Anwendung von Cluster-Analyse-Techniken. Allgemein geht es dabei um die Erkennung von Mustern in nicht klassifizierten Daten, genauer gesagt das Auffinden von Gruppen.

Als Anwendungsfall werden dazu bestimmte sich derzeit entwickelnde Formen von elektronischen Diskussionen, sogenannte *eDiscussions*, betrachtet. Bei eDiscussions findet der Gedankenaustausch, im Gegensatz zu Diskussionen in der realen Welt, immer über einen Computer beziehungsweise ein Computer-Programm statt. In der Regel stellt dazu jemand eine Diskussionsplattform (Server) bereit, auf dem sich die Teilnehmer der Diskussion über ein Netzwerk mit Hilfe eines (Client-) Programms anmelden und diskutieren können. Es gibt verschiedene Formen von eDiscussions, die sich beispielsweise in der Visualisierung der Diskussion oder den Interaktionsmöglichkeiten der Benutzer unterscheiden.

Das bekannteste Beispiel stellen diesbezüglich Internet-Diskussionsforen dar. Diese geben grundsätzlich immer die gleiche (Baum-)Struktur vor, was die Möglichkeiten der Verbindung der Diskussionsbeiträge untereinander einschränkt. Diskussionsbeiträge können jeweils nur mit maximal zwei Arten von Beiträgen in Relation stehen: Mit der Ursprungsnachricht und einer oder mehrerer möglicher Antworten auf den Beitrag.

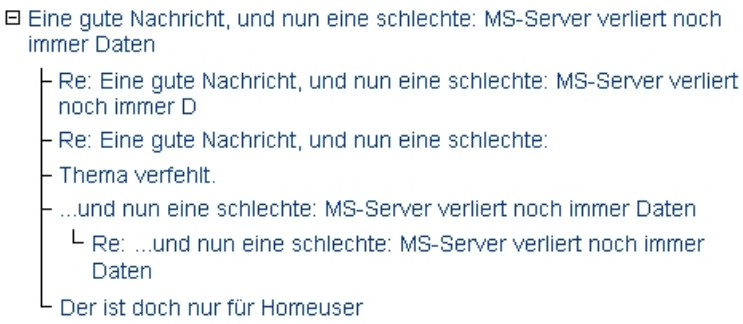


Abbildung 1: Internet-Forendiskussion mit typischer Baumstruktur.

Ein im Moment populäres Konzept, das auch komplexere Diskussionsstrukturen zulässt, sind graphbasierte Diskussionen. Eine der ältesten Diskussionsanwendungen, die diese Art der Visualisierung verwendet, ist *Belvedere* [SWCP95], dessen Modelle auf den Konzepten von Stephen Toulmin [Tou69] basieren.

Bei Diskussionen dieser Art findet die Präsentation in Form eines gerichteten Graphen statt. Knoten repräsentieren Diskussionsbeiträge und Kanten die Relationen zwischen den Beiträgen. Solche Graphen sind, was bestimmte Eigenschaften betrifft, vergleichbar mit Petri-Netzen [Pet62]. So gibt es bei Beiden gewichtete, gerichtete Kanten sowie Knoten

verschiedenen Typs, die jeweils bestimmte Eigenschaften besitzen. Im Gegensatz zu einem Petri-Netz, das immer ein bipartiter Graph ist, können bei einem Diskussionsgraphen jedoch prinzipiell alle Knoten miteinander verbunden werden. Speziell letztere Eigenschaft trägt zur Flexibilität graphbasierter Diskussionen bei, so dass beliebige Strukturen möglich sind. Diese können dabei so unübersichtlich werden, dass sie von einem Moderator nicht mehr ohne weiteres überschaubar sind.

Um Moderatoren bei ihrer Tätigkeit zu unterstützen, werden in dieser Arbeit Cluster-Analyse-Techniken dazu verwendet, um Diskussionsbeiträge von graphbasierten Diskussionen zu Gruppen zusammenzufassen. Diese Gruppierungen sollen dem Moderator einen besseren Überblick über die Diskussion liefern und ein schnelleres Ablesen von bestimmten Informationen ermöglichen.

1.2 Argunaut

Die Master Thesis wird hochschulextern bei dem *Fraunhofer Institut für Intelligente Analyse- und Informationssysteme (IAIS)* in der Abteilung *Virtual Environments (VE)* in Birlinghoven (Sankt Augustin) durchgeführt. Als Grundlage dient das von der *Europäischen Union (EU)* geförderte Projekt *Argunaut* (Förderkennzeichen: 027728), bei dem mit Techniken der künstlichen Intelligenz Werkzeuge zur Moderation von Diskussionen geschaffen werden. Beispielhafte Diskussionsanwendungen, für die Moderationswerkzeuge geschaffen werden, sind *FreeStyler* [Gas03] und *Digalo* [LK06].

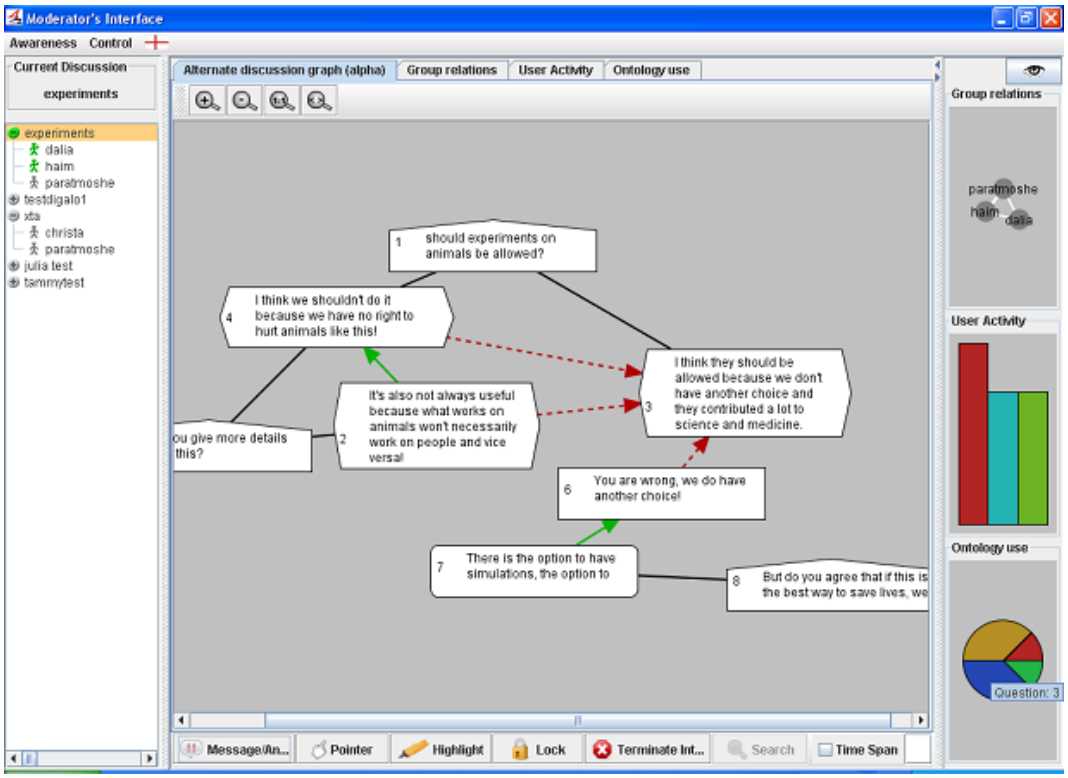


Abbildung 2: Graphbasierte Diskussion im Argunaut Moderator's Interface.

Ein im Rahmen des Argunaut Projektes entwickeltes Programm ist das *Moderator's Inter-*

face. Dieses bietet einem Moderator verschiedene Möglichkeiten zur Überwachung von eDiscussions. Es unterstützt zum Beispiel Diskussionen, die mit Digalo geführt werden. Abbildung 2 zeigt eine graphbasierte Diskussion im Moderator's Interface. In der Mitte ist der Diskussionsgraph selbst zu sehen, während sich auf der rechten Seite die Aufgaben einiger Plugins befinden, die den Moderator mit Zusatzinformationen unterstützen sollen. Aus diesen kann der Moderator beispielsweise ablesen, welche Benutzer untereinander wie viel miteinander kommuniziert haben oder welcher Benutzer wie aktiv war. Das unterste Plugin zeigt, welche Arten von Diskussionsbeiträgen und Verbindungskanten wie oft verwendet wurden. Je nach verwendeter Diskussionsanwendung gibt es unterschiedliche Typen von Diskussionsbeiträgen, beispielsweise Frage, Antwort, Kommentar oder Argument, und verschiedene Arten von Verbindungskanten, wie Widerspruch, Zustimmung oder eine neutrale Verbindung.

Im Laufe dieser Abschlussarbeit soll ein weiteres Plugin entwickelt werden, das dem Moderator Zusatzinformationen bereitstellt, die ihm eine bessere Übersicht über die Diskussion und eine schnellere Extraktion von spezifischen Informationen ermöglichen.

1.3 Ziel der Master Thesis

Die hohe strukturelle Komplexität graphbasierter Diskussionen macht es für Moderatoren schwierig, diese effizient zu beobachten, ohne den Gesamtüberblick zu verlieren. Zudem wird es problematisch, wenn es darum geht, spezifische Informationen aus einem Diskussionsgraph abzulesen. Selbst bei geringen Benutzerzahlen ändern sich, falls alle Benutzer aktiv an der Diskussion mitwirken, in kurzer Zeit so viele Beiträge, dass der Moderator nicht alle zur gleichen Zeit sichten und gegebenenfalls bearbeiten oder kommentieren kann. Es gibt viele Möglichkeiten, um aus einem Diskussionsgraphen Zusatzinformationen zu extrahieren, die dem Moderator seine Tätigkeit erleichtern. Eine Möglichkeit ist beispielsweise das Auffinden wichtiger Diskussionsbeiträge oder kritischer Stellen, bei denen Konfliktpotenzial besteht. In dieser Abschlussarbeit geht es darum, möglichst *sinnvolle Gruppierungen* von Diskussionsbeiträgen zu finden, wobei eine solche Gruppierung an dieser Stelle so definiert wird, dass sie dem Moderator dabei hilft, eine bessere Übersicht über die Diskussion zu gewinnen und schneller Informationen aus dieser ablesen zu können.

Ein einfacher Ansatz zur Lösung dieses Problems könnte es sein, die Diskussion statisch mittels eines Clustering-Algorithmus zu untersuchen. Der Algorithmus würde so ablaufen, dass die Objekte (Diskussionsbeiträge) des Diskussionsgraphen, falls deren Daten in einem homogenen Format vorliegen, mit einem Verfahren der Cluster-Analyse (siehe auch Abschnitt 3.3) analysiert und Gruppen zugeordnet werden. Dabei würden alle Eigenschaften der Diskussionsbeiträge betrachtet, unabhängig davon, ob der Moderator sie als wichtig oder unwichtig ansieht, da der Algorithmus ohne weitere Informationen nicht wissen kann, nach welchen Kriterien er die Beiträge gruppieren soll. Ob diese Gruppeneinteilung sinnvoll ist beziehungsweise den Ansprüchen des Moderators entspricht, ist jedoch fragwürdig.

Dieses Problem ließe sich durch Integration von Expertenwissen in die Logik des Algorithmus abschwächen, so dass eine Bewertung der Eigenschaften der Diskussionsbeiträge vorgeben wäre. So wäre ein Fall abgedeckt, der eine möglicherweise sinnvolle Gruppierung erreichen könnte. Jedoch wäre der Algorithmus auch weiterhin für alle anderen Fälle, nach denen der Moderator eventuell gruppieren möchte, sehr unflexibel. Ein weiterer

Nachteil dieses Ansatzes wäre die hohe Dimensionalität der Diskussionsbeiträge, da jeder Beitrag viele verschiedene Eigenschaften besitzt, die prinzipiell in einen Gruppierungsvorgang einbezogen werden können. Würde nun ein statischer Clustering-Algorithmus nach allen Parametern gruppieren, so würde er mit hoher Wahrscheinlichkeit kein gutes Ergebnis beziehungsweise keine gut separierten Gruppen berechnen.

Eine Cluster-Analyse funktioniert grundsätzlich am besten mit möglichst niedrigdimensionalen (zum Beispiel eindimensionalen) Daten, da in diesen am leichtesten klare Strukturen zu erkennen sind. Je mehr Dimensionen hinzu kommen, desto schwieriger beziehungsweise unwahrscheinlicher wird es für ein solches Verfahren, Gruppen von Objekten zu finden, die für alle Dimensionen ähnliche Werte besitzen.

Der Ansatz, den ich in dieser Master Thesis verfolge, ist die Konzeption und Entwicklung eines interaktiven, dynamischen Clustering-Algorithmus. Die Grundidee dabei ist, dass der Benutzer zu einem großen Teil selbst möglichst interaktiv definieren soll, welche Diskussionsbeiträge ähnlich zueinander sind. Die Aufgabe des Algorithmus besteht darin, die Benutzerinteraktion und die Daten des Diskussionsgraphen so zu analysieren, dass die Diskussionsbeiträge *sinnvoll*, also nach den Vorgaben des Moderators, gruppiert werden können. Der Algorithmus soll zudem dynamisch sein: Er soll den Moderator in Echtzeit mit dem Moderator's Interface interagieren lassen. Daher wird dem Moderator die Möglichkeit gegeben, nachdem eine Gruppierung berechnet wurde, das vorhandene Ergebnis weiter zu verfeinern bis es seinen Anforderungen gerecht wird.

Die Ergebnisse des interaktiven Clustering-Verfahrens sollen den Moderator dabei unterstützen, eine möglichst gute Übersicht über die Diskussion zu gewinnen. Zudem soll er durch die Zusatzinformationen der Gruppierung in der Lage sein, leichter spezifische Informationen aus dem Diskussionsgraphen extrahieren zu können. Ob und wie sehr die Gruppierungen des Verfahrens einem Benutzer bei der Beobachtung einer Diskussion helfen, untersuche ich in einer stichprobenartigen empirischen Benutzerstudie.

1.4 Aufbau der Master Thesis

Die Abschlussarbeit ist in sieben Kapitel unterteilt.

In Kapitel 2 wird die thematische Einordnung zu Mustererkennungstechniken, Visualisierungsformen von eDiscussions und Interaktionsmöglichkeiten hergestellt, sowie die Abstraktion zu anderen Themengebieten erläutert. Zudem werden hier ähnliche, statische Strukturierungsverfahren von Diskussionsstrukturen untersucht.

Kapitel 3 dient der Erklärung von Technologien, die zur Realisierung dieser Arbeit verwendet wurden. Dazu zählen Cluster-Analyse-Methoden, Techniken der sozialen Netzwerkanalyse zur Analyse struktureller Äquivalenzen, sowie Techniken aus dem Bereich Text Mining zur inhaltlichen Analyse der Diskussionsbeiträge.

In Kapitel 4 werden Kriterien zur Bewertung von Interaktionstechniken erläutert. Im zweiten Teil dieses Kapitels werden die Kriterien auf einige Interaktionstechniken angewendet, um eine Auswahl zur Implementierung zu treffen.

Kapitel 5 beschreibt die Konzepte zur Implementierung der Interaktionsmöglichkeiten und des Clustering-Algorithmus. Dabei wird im Detail auf Anpassungen eingegangen, die bezüglich des Anwendungsfalls dieser Abschlussarbeit (Diskussionsgraphen) vorgenommen werden mussten.

In Kapitel 6 werden die Ergebnisse des neuen Verfahrens vorgestellt. Dazu zählt eine statistische Auswertung sowie eine stichprobenartige Benutzerstudie.

In Kapitel 7 wird ein Fazit gezogen und ein Blick nach vorne geworfen, welche Ansätze in Zukunft verfolgt werden könnten, um den Algorithmus weiter zu verbessern.

2 Thematische Einordnung

Dieses Kapitel gibt einen Überblick, wie das von dieser Master Thesis behandelte Problem thematisch in verschiedene Themengebiete wie Mustererkennung, Interaktivität und die Visualisierung von Argumentationsstrukturen eingebunden ist. Zudem werden themenverwandte Arbeiten betrachtet und analysiert. Im letzten Abschnitt des Kapitels wird eine Verallgemeinerung des Problems vorgenommen.

2.1 Konkretisierung des Problems

In den folgenden Abschnitten wird das Problem weiter konkretisiert, indem die Hauptaspekte des interaktiven Clusterings erläutert werden.

2.1.1 Mustererkennung kombiniert mit dynamischer Interaktion

Grundsätzlich ist das hier betrachtete Problem, die Gruppierung von bestimmten Objekten, ein Problem aus dem Bereich Mustererkennung. Dieser kann wiederum in die Bereiche überwachtes und nicht überwachtes Lernen aufgeteilt werden. Mustererkennung beschäftigt sich allgemein mit der Fähigkeit, aus einer Menge von Daten Regelmäßigkeiten oder Ähnlichkeiten zu erkennen. Bei überwachtem Lernen wird diese Definition so spezifiziert, dass bereits eine Klassifikation der Daten vorliegt und eine Abgrenzung zwischen den verschiedenen Klassen gesucht wird, während bei nicht überwachtem Lernen nur die reinen Datensätze ohne Klassenzugehörigkeiten gegeben sind und letztere gesucht werden müssen. [Web02, S.5-6]

Im Folgenden wird diese Definition auf die aktuelle Situation übertragen. Gegeben ist ein Diskussionsgraph, dessen Knoten[1] zwar bestimmte Eigenschaften besitzen, die jedoch nicht klassifiziert sind. Die Beiträge sind von den Diskussionsteilnehmern auf der Ebene des Diskussionswerkzeugs[2] so verteilt worden, wie diese es während der Diskussion für richtig hielten. Graphbasierte Diskussionen werden teilweise nachbearbeitet, jedoch größtenteils nicht so, dass ein Moderator aus der neuen Anordnung eine sinnvolle Gruppierung der Beiträge schließen kann. Daher können allgemein keine sinnvollen Gruppierungen aus der visuellen Verteilung der Diskussionsbeiträge ablesen werden. Somit erhält ein Moderator keine Zusatzinformationen, welcher Diskussionsbeitrag zu welcher Gruppe von Beiträgen gehört, woraus gefolgert werden kann, dass Beiträge von Diskussionsgraphen grundsätzlich nicht klassifiziert sind.

Des Weiteren ist eine Definition notwendig, was überhaupt als eine sinnvolle Gruppierung anzusehen ist. Allgemein soll eine sinnvolle Gruppierung den Abstand von ähnlichen Objekten in der Darstellungsebene minimieren (Gruppen bilden) und den Abstand zwischen unähnlichen Gruppen maximieren. Diese allgemeine Betrachtung ist in der Realität aber oft unzureichend, da bei einer Gruppierung viele Parameter betrachtet werden können, die dem Moderator möglicherweise nicht wichtig sind. Möchte er beispielsweise Beiträge mit ähnlichem Inhalt gruppieren, so müsste er dies dem Clustering-Algorithmus mitteilen können, da ansonsten die vollständigen Datenvektoren mit (in diesem Beispiel)

[1] Ein Diskussionsgraph besteht nicht nur aus einzelnen, voneinander unabhängigen Elementen wie Knoten beziehungsweise Diskussionsbeiträgen, sondern auch aus Kanten. Inwiefern diese in den Strukturierungsprozess mit eingehen wird in Abschnitt 2.1.2 und besonders Kapitel 3 erläutert.

[2] Beispielsweise Digalo oder FreeStyler.

unwichtigen Parametern wie Zeit, Autor oder die Anzahl der ein- und ausgehenden Kanten ausgewertet werden.
Um dieses Problem zu lösen, verfolge ich in dieser Arbeit den Ansatz, dass der Benutzer interaktiv eine Vorgabe machen kann, nach welchen Kriterien der Algorithmus die Diskussionsbeiträge gruppieren soll. Er kann also seinen Ähnlichkeitsbegriff spezifizieren.

Durch den Aspekt der Benutzerinteraktion kann der Moderator daher einige wenige Elemente klassifizieren oder eine Vorgabe bezüglich der Aspekte des Clusterings machen. Abbildung 3 zeigt den beispielhaften Ablauf eines interaktiven Algorithmus. Gegeben ist eine ungeordnete Menge (Schritt 1). Zunächst interagiert der Benutzer mit dem Programm, zieht zum Beispiel zwei gleichfarbige Elemente mit einer Mausbewegung zusammen (Schritt 2-3). Der Algorithmus analysiert die aktivierten Elemente sowie die Elemente in nächster Umgebung, passt seine Gewichtungsdaten darauf an und gruppiert die restlichen Elemente schrittweise entsprechend (Schritt 4-6).

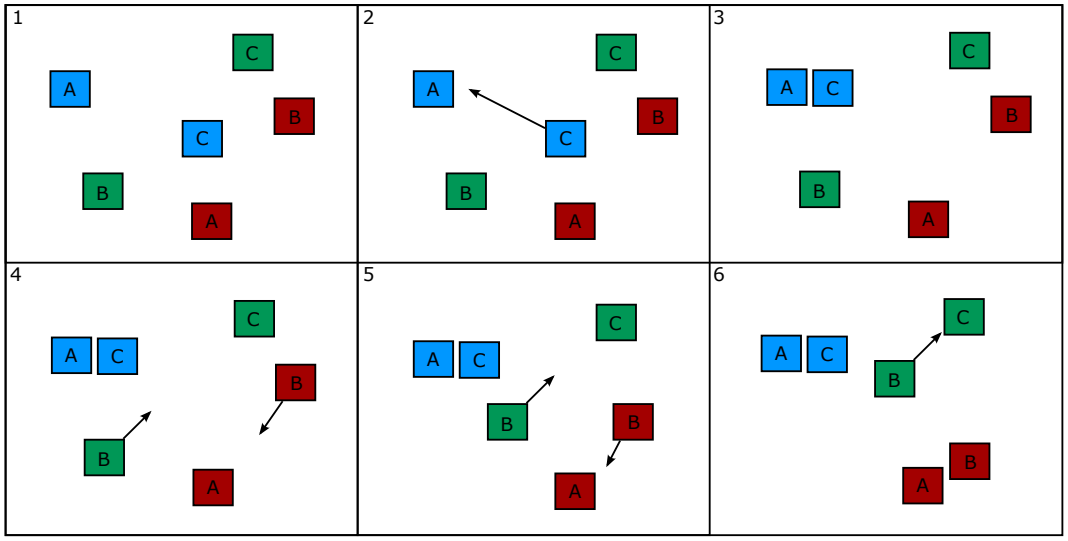


Abbildung 3: Beispiel für einen interaktiven Clustering-Algorithmus.

Nach dem Beispiel in Abbildung 3 würde der Benutzer die Beiträge, die er örtlich nah zueinander platziert hat, gruppieren. In einem anderen Fall könnte er aber auch mehrere Beiträge weit voneinander weg platzieren und dem Algorithmus damit mitteilen, dass diese Elemente nicht zu der selben Klasse gehören.

Der Moderator überwacht den Lernprozess demnach teilweise, unabhängig davon wie er mit den Diskussionsbeiträgen interagiert. Aus diesen Feststellungen kann gefolgert werden, dass es sich hier nicht um ein reines überwachtes oder nicht überwachtes Lernverfahren handelt, sondern um ein Hybridverfahren. Der Benutzer kann den Lernprozess zu so großen Teilen beeinflussen wie er möchte. Nimmt er die Gruppierung vollständig per Hand vor, wäre der Lernprozess praktisch überflüssig, jedoch per Definition überwacht, da jedes Element bereits vor Ausführung des Algorithmus klassifiziert ist. Greift er überhaupt nicht ein, handelt es sich um einen nicht überwachten Lernvorgang, da der Algorithmus die Gruppierung nur anhand der Daten des Diskussionsgraphen und der einzelnen Elemente vornehmen muss.

Der bisherige Abschnitt hat den Bereich des interaktiven Clusterings abgedeckt; zusätzlich soll der geplante Algorithmus dynamisch sein, also dem Benutzer während der Gruppierung selbst erlauben, in den Vorgang einzugreifen. Demnach soll das Ergebnis, nachdem der Algorithmus eine erste Gruppierung vorgenommen hat, je nach Interaktionstechnik vom Benutzer weiter verbessert werden können.

In einem statischen Verfahren hingegen kann es passieren, dass eine Gruppierung vorgenommen wird, die nicht den Anforderungen des Benutzers entspricht und er daran nichts ändern kann, außer ein neues Clustering mit anderen Werten durchzuführen. Je nachdem wie die Kriterien für das Clustering mit Werten besetzt werden, kann dieser Vorgang für den Benutzer umständlich sein.

2.1.2 Visualisierung von Argumentationsstrukturen

Die hier betrachtete graphbasierte Visualisierungsform von Argumentationsstrukturen hat starken Einfluss auf den geplanten Strukturierungsalgorithmus.

An dieser Stelle wird zunächst eine andere Visualisierungsform betrachtet: Internet-Foren. Hier gibt es in der Regel keine verschiedenen Beitragstypen. Verbindungen zwischen den einzelnen Beiträgen sind so einfach gehalten, dass eine strukturelle Analyse kaum Einfluss auf einen Klassifizierungsalgorithmus hat. Das einzige Merkmal, das zur Bestimmung der Relevanz eines Beitrags verwendet werden kann, ist die Anzahl der Verbindungen zu anderen Beiträgen.

In graphbasierten Argumentationsstrukturen gibt es deutlich komplexere Möglichkeiten. Im Gegensatz zu normalen Internet-Foren können Beiträge einen bestimmten Typ haben und jeder Beitrag kann prinzipiell mit jedem anderen verbunden werden. Zusätzlich haben die Verbindungen zwischen den Beiträgen einen Typ wie Widerspruch oder Zustimmung. Bei Digalo und FreeStyler gibt es für Knoten und Kanten festgelegte Typen, die in einer Diskussion viele mögliche Verknüpfungen bieten.

Abbildung 4 zeigt verschiedene Strukturen, die bei Forendiskussionen (links) und graphbasierten Diskussionen (rechts) entstehen können. Bei Forendiskussionen ist die Struktur deutlich einfacher als bei graphbasierten Diskussionen; die Beiträge haben alle den gleichen Typ, die Kanten sind ungerichtet und bilden immer einen Baum. Bei graphbasierten Diskussionen können Knoten einen bestimmt Typ haben und Kanten gewichtet und gerichtet sein. Jeder Beitrag kann zu jedem beliebigen anderen Beitrag verbunden werden, weshalb sowohl Zyklen als auch unabhängige Komponenten auftreten können.

Die strukturelle Beschaffenheit einer Argumentation bei graphbasierten Visualisierungen besitzt großen Einfluss auf die Ausdruckskraft der Diskussion. Um diese als konkrete Parameter in den Gruppierungsprozess mit einfließen zu lassen, werden Methoden aus dem Bereich der sozialen Netzwerkanalyse, beispielsweise eine Analyse struktureller Äquivalenz beziehungsweise Positions-Analyse [WF94, S.345-423], angewendet. Diese Techniken werden in Kapitel 3 erläutert.

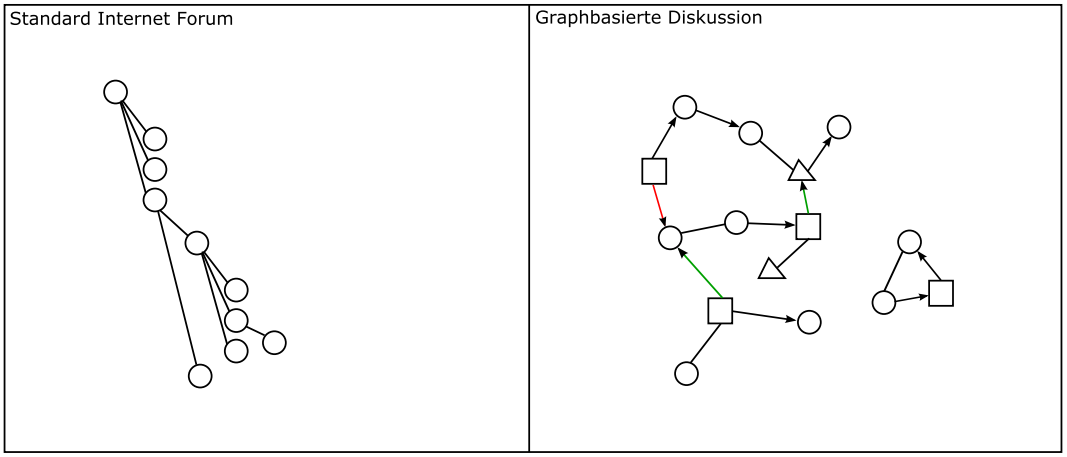


Abbildung 4: Beispiel für Diskussionsstrukturen.

2.1.3 Textanalyse

Ein weiterer Aspekt, der bei der Gruppierung von Diskussionsbeiträgen betrachtet werden kann, ist der Inhalt. Ein Problem diesbezüglich besteht darin, dass syntaktische oder semantische Fehler der Texte nicht ausgeschlossen werden können. Des Weiteren besitzt beispielsweise die deutsche Sprache[3] viele Fälle, die semantisch nicht eindeutig zuzuordnen sind, so dass eine inhaltliche Analyse schwer zu realisieren ist. Da dieser Aspekt kein Schwerpunkt dieser Abschlussarbeit ist, habe ich mich dazu entschlossen, eine statistische Analyse durchzuführen und beispielsweise für jeden Beitrag zu untersuchen, welche Buchstaben und Wörter am häufigsten vorkommen. [HQW06, S.195]

2.2 Ähnliche Arbeiten

Im folgenden Abschnitt werden einige themenverwandte Arbeiten vorgestellt und je nach Arbeit darauf eingegangen, wie die bestehende Problematik bearbeitet wurde. In den betrachteten Arbeiten geht es um die Analyse von einzelnen Dokumenten[4], so dass Zusatzinformationen diesbezüglich für einen Benutzer des jeweiligen Systems leichter erkennbar werden.

2.2.1 DOCE-Algorithmus

Die Diplomarbeit von Mikšátko [Mik07] verfolgte ein ähnliches Ziel wie diese Master Thesis: die Unterstützung von Moderatoren von eDiscussions, indem sinnvolle Gruppen beziehungsweise interessante Interaktionen erkannt werden (Cluster-Erkennung). Im Rahmen der Arbeit wurde der DOCE-Algorithmus (Detection of Clusters by Example) entwickelt. Der DOCE-Algorithmus basiert auf der früher nur in Datenbanken eingesetzten Abfragesprache *Query By Example (QBE)* [Mik07, S.20]. Die Idee hinter QBE besteht darin, dass als Abfrage ein Beispiel[5] angegeben wird.

[3] Der hier vorgestellte Algorithmus soll jedoch nicht nur deutsche Diskussionen bearbeiten können, sondern sprachunabhängig sein.

[4] Ein Diskussionsbeitrag einer graphbasierten Diskussion kann auch als Dokument betrachtet werden.

[5] Grundsätzlich eine Tabelle.

Der Algorithmus nimmt als Eingabe einen Diskussionsgraphen sowie ein sogenanntes Modell. Das Modell ist ein in der Regel kleiner Graph, der zuvor von pädagogischen Experten definiert wurde und eine interessante Interaktion repräsentieren soll. Der Ablauf von DOCE ist wie folgt:

1. Vorverarbeitung (Preprocessing): Zunächst wird aus den Daten des Modell- und Eingabegraphen jeweils eine Adjazenzmatrix generiert. Danach wird zu jedem Knoten mit Hilfe von Textanalyse und Knoteneigenschaften ein Datenvektor erstellt. Zur Textanalyse wird das Java-Tool TagHelper verwendet, das in einer Arbeit von Rosé et al. [RWA+08] entstanden ist und den Schwerpunkt auf Texte legt, die bei kollaborativen Lernprozessen entstanden sind.

2. Nun wird jeder Modell-Datenvektor mit jedem Eingabe-Datenvektor verglichen, woraus eine Ähnlichkeitsmatrix mit Wertepaaren resultiert.

3. Ein A*-Algorithmus wird verwendet, um alle möglichen Knoten-zu-Knoten Mappings zwischen Modell- und Eingabe-Graph zu finden (mittels Ähnlichkeits- und Adjazenzmatrix). Ziel ist es, das Mapping mit der minimalen Distanz[6] zu finden, indem Operationen wie “Kante hinzufügen”, “Kante löschen” oder “Knoten löschen” verwendet werden. Wurde schließlich ein komplettes Mapping gefunden, so wird dieses in die Ergebnisliste eingetragen.

4. Die Ergebnisliste wird entsprechend der Ranking-Werte aufgestellt, wobei die höchsten Werte an die ersten Stellen kommen.

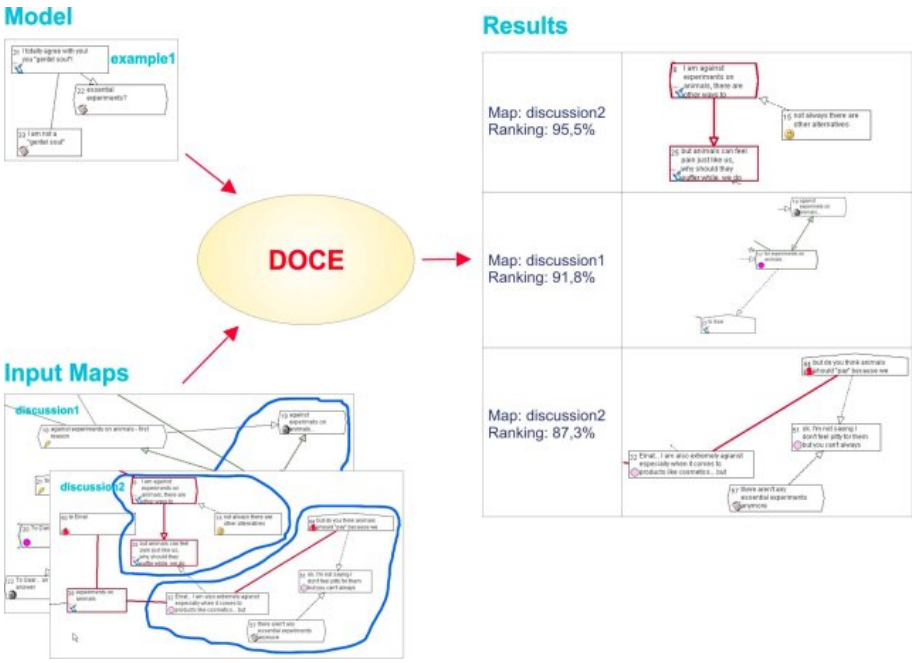


Abbildung 5: DOCE-Algorithmus. Quelle: [Mik07, S.21]

Die Ausgabe des Algorithmus ist eine Liste von Diskussionsgraphen, die zu einem bestimmten Ranking (in %) mit einem der Modell-Graphen übereinstimmen.

[6] Maximale inhaltliche Ähnlichkeit und minimale strukturelle Differenz.

DOCE hat in der statistischen Untersuchung der zugehörigen Abschlussarbeit gute Ergebnisse erzielt. Bis zu 80% der gesuchten Modelle wurden in Beispiel-Graphen gefunden. Als Nachteil könnte jedoch die komplizierte Benutzbarkeit des Verfahrens angesehen werden: Als Eingabe für den Algorithmus benötigt der Benutzer unter anderem Modell-Graphen, die zunächst (von pädagogischen Experten) definiert werden müssen. Der Benutzer muss sich an die vorgegebenen Modelle halten und, falls gewünschte andere Modelle nicht vorhanden sind, müssen diese erst erstellt und in einem bestimmten XML-Format gespeichert werden.

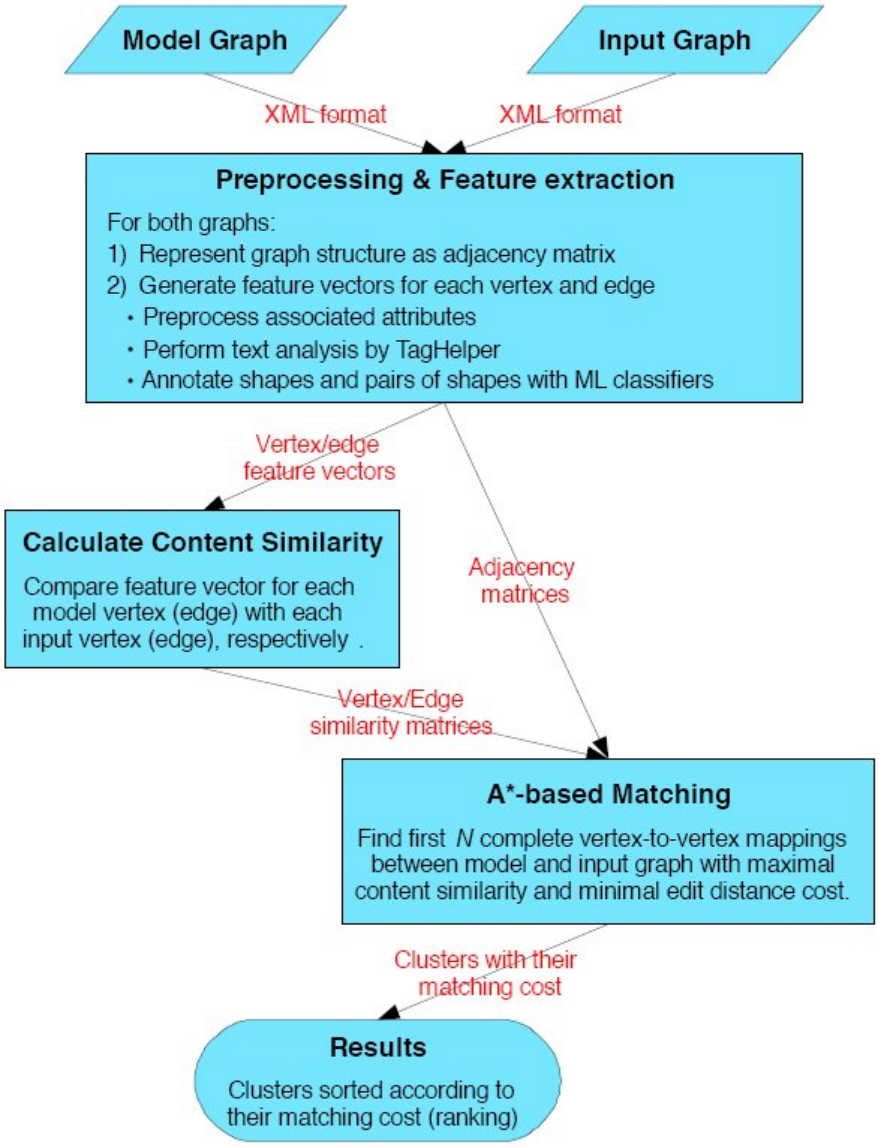


Abbildung 6: Ablauf des DOCE-Algorithmus. Quelle: [Mik07, S.30]

Fazit: DOCE erzielt gute Ergebnisse bei der Cluster-Erkennung, ist jedoch unflexibel in der Anwendung und benötigt, aufgrund der benötigten Vorgaben der pädagogischen Ex-

perten, lange Vorlaufszeit. Letztere Kritikpunkte sind dadurch zu begründen, dass das Verfahren nicht interaktiv ist.

2.2.2 ScholOnto

Im Projekt Scholary Ontologies (ScholOnto) von Kirschner et al. [KSC03, S.185-204] wurde eine Anwendung für graphbasierte Diskussionen über wissenschaftliche Arbeiten entwickelt. Das Programm ClaiMaker erlaubt es Benutzern, Objekte mit textuellem Inhalt sowie weiteren Attributen wie "Objekt-Typ" zu erstellen, die auf Literaturquellen verweisen, und diese beispielsweise mit eigenen Objekten oder Objekten von anderen Benutzern zu verknüpfen. Die Verknüpfung zwischen zwei Objekten ist wiederum bewertet, zum Beispiel mit Attributen wie "ist Beweis für", "löst dieses Problem" oder "ist identisch mit". Zwei Objekte, die mit einer gewichteten Verbindung miteinander verknüpft sind, heißen Claim (Behauptung).

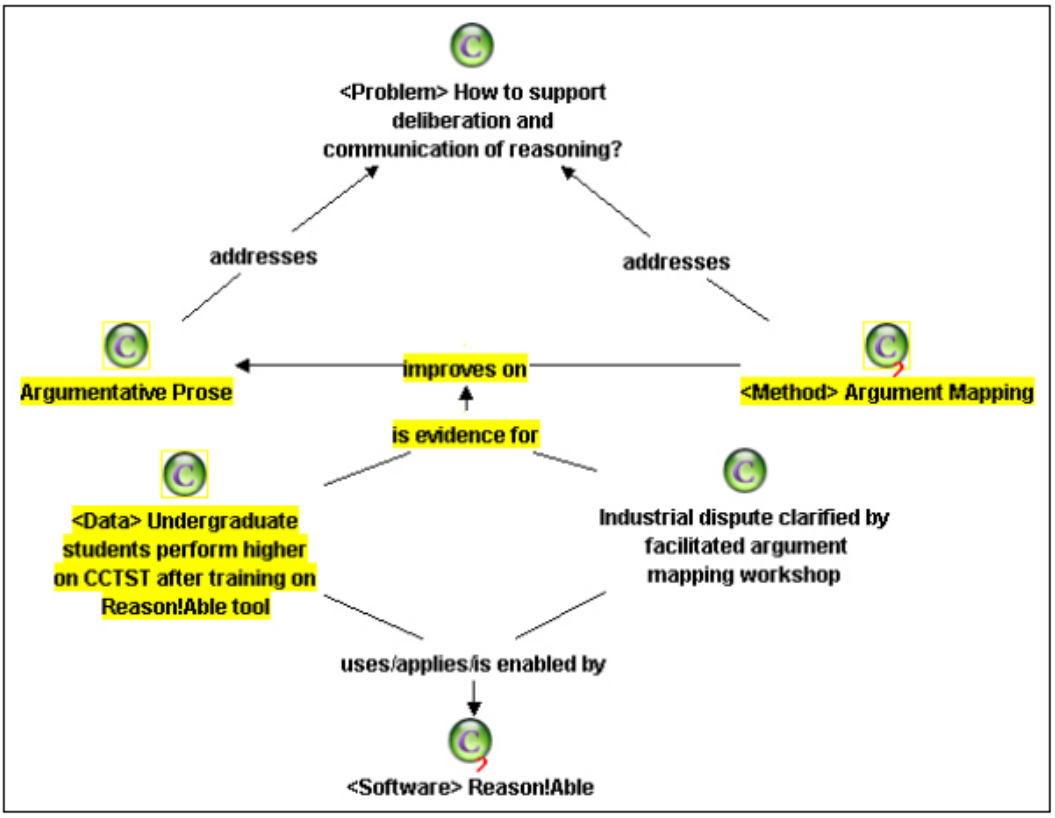


Abbildung 7: Beispielhafte ClaiMaker Diskussion. Quelle: [KSC03, S.194]

Auch bei diesem Projekt besteht das Problem, dass es viele Literaturquellen gibt, deren Menge ständig wächst und die zu umfangreichen Diskussionen anregen. Aus diesem Grund wurde nach *graphentheoretischen* und *semantischen* Analysemöglichkeiten für derartige Netzwerke gesucht.

Zur *graphentheoretischen* Analyse wurde zunächst Erdos' Annahme [ER60] hinzugezogen: Falls die Anzahl der Kanten eines Zufallsgraphen mindestens einhalbmal so groß

ist wie die Anzahl der Knoten, so bildet sich daraus eine sogenannte Riesenkomponente. Unter dem Begriff Riesenkomponente versteht man einen dicht verbundenen Subgraphen, der die meisten Knoten des Gesamtgraphen enthält. Weiterhin wurde angenommen, dass es sich bei jeder Riesenkomponente um ein Beispiel für ein Small-World-Netzwerk [Wat99] handeln kann. Small-World-Netzwerke sind Graphen, die sich dadurch auszeichnen, dass sie einen kleinen Durchmesser und einen hohen durchschnittlichen Clustering-Koeffizienten haben. Dadurch ist jeder Knoten von jedem anderen Knoten aus leicht erreichbar und die Knoten des Graphen sind generell stark untereinander verbunden.

Bezüglich ClaiMaker wurde die Hypothese aufgestellt, dass es in den Diskussionen Konzepte, also bewertete Relationen zwischen Objekten, gibt, die wichtig genug sind, dass sie in mehreren Forschungsdisziplinen vorkommen. Beispielsweise könnte das Konzept "Kleine Welt" mit Telekommunikationsnetzwerken, Graphentheorie und Studien von Nahrungsnetzen in Verbindung stehen. Eine Suche nach diesem Konzept ist also hilfreich für den Benutzer, da er über die Suchergebnisse schnell zwischen verschiedenen Regionen des Graphen wechseln kann.

Der erste Schritt, um solche "kurzen Pfade" im Graphen zu finden, ist die Identifikation von dicht verbundenen Dokumenten, die ein Small-World-Netzwerk bilden. Kirschner et al. erreichten dies mit bestimmten Graph-Visualisierungsalgorithmen zur vereinfachten Darstellung des Netzwerks. Durch diese Darstellung war es nun relativ einfach, wichtige Knotenpunkte zu identifizieren. Davon ausgehend konnten Subgraphen des Gesamtgraphen betrachtet und in diesen die einzelnen Cluster identifizieret werden. Im Subgraph wurden jedoch nur *wichtige* Knoten angezeigt; per Definition waren dies hier nur Knoten, die drei oder mehr Verbindungen zu anderen Knoten besaßen. Diese Visualisierung ist in Abbildung 8 zu sehen.

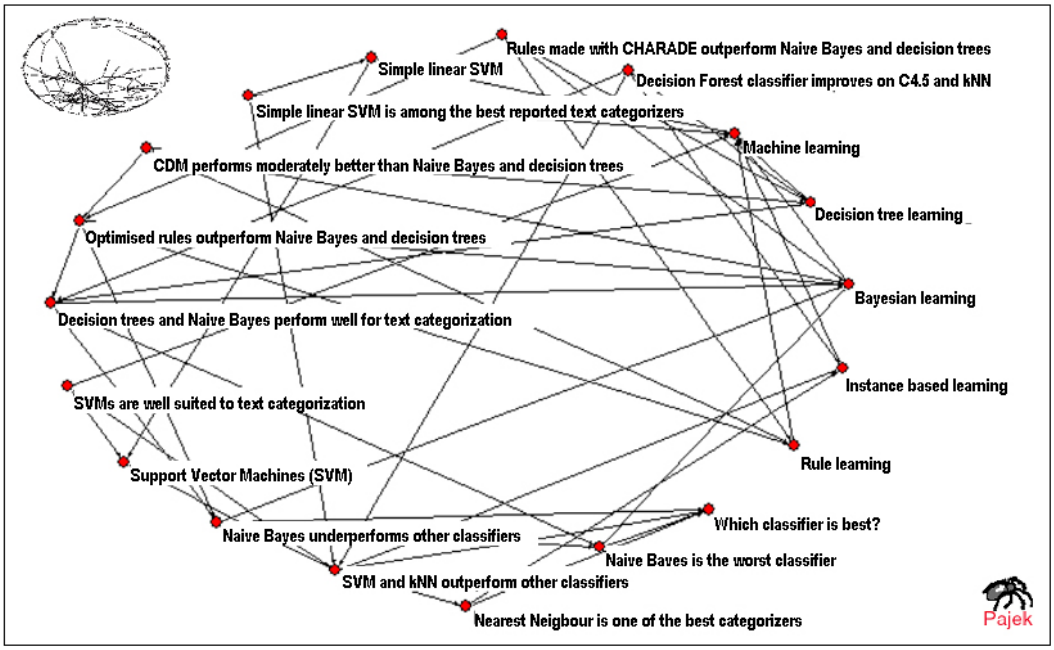


Abbildung 8: Beispiel für vereinfachte Graph-Visualisierung. Quelle: [KSC03, S.197]

Ein weiterer Ansatz, mit dem experimentiert wurde, war die Messung, wie oft ein Artikel zitiert wurde, um daraus weitere *Leistungsindikatoren* von Publikationen, Autoren oder Organisationen bestimmen zu können.

Wie bereits erwähnt wurden auch *semantische* Analysemöglichkeiten untersucht. In dieser Analyse wurde zunächst die Frage "Welche Argumente gibt es gegen diese Publikation?" untersucht. Dabei werden zunächst mit einer perspektivischen Analyse [KSC03, S.198] Konzepte untersucht, die in direktem Widerspruch zum angefragten Konzept stehen. Falls keine gefunden werden, werden in einem weiteren Schritt Konzepte gesucht, die mit dem angefragten Konzept positiv in Verbindung stehen und deren Widersprüche ermittelt. Dieses Vorgehen kann, falls auch bei den Nachbarn der benachbarten Konzepte nur ähnliche oder positive Verbindungen gefunden wurden, so lange fortgesetzt werden bis irgendwann eine negative Verbindung gefunden wird.

Zudem wurde die Frage "Woher stammt die Idee zu dieser Arbeit?" behandelt. Realisiert wurde sie durch eine sogenannte Abstammungsanalyse. Dabei werden grundsätzlich die Vorfahren eines bestimmten Beitrags weiter untersucht, bis der Ursprung gefunden wird. Während der Analyse werden Verbindungen wie "benutzt", "verbessert", "ist Teil von" oder "ist ähnlich zu" weiter verfolgt.

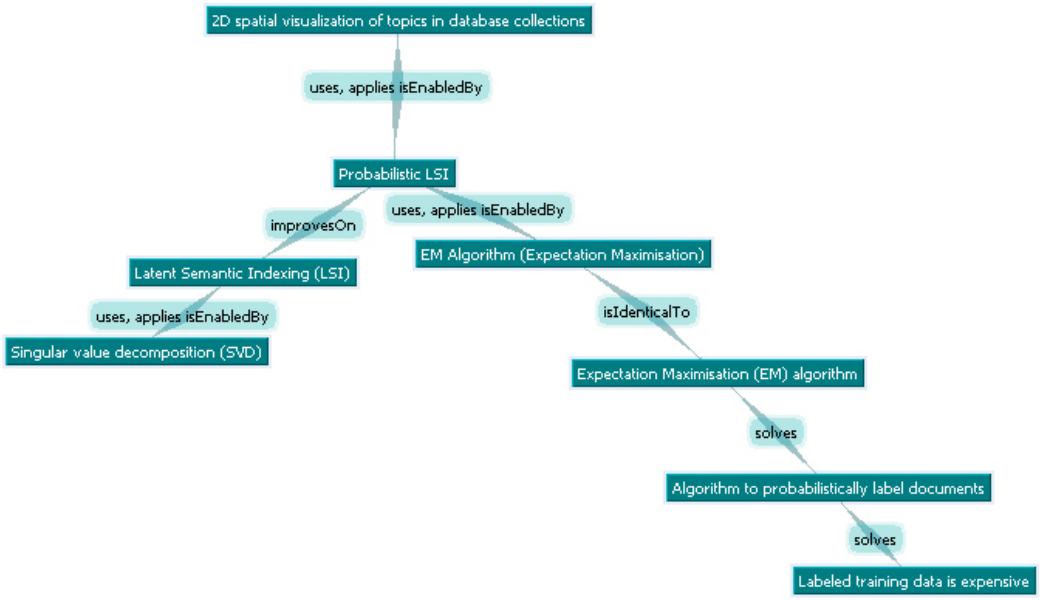


Abbildung 9: Visualisierung der Ergebnisse einer Abstammungsanalyse. Quelle: [KSC03, S.201]

Abbildung 9 zeigt eine beispielhafte Visualisierung einer Abstammungsanalyse. Sie zeigt den Kontext des Ausgangskonzeptes (ganz oben) zu themenverwandten Konzepten auf.

Fazit: Innerhalb von ScholOnto wurde ein Werkzeug geschaffen, um die Bedeutung von wissenschaftlichen Arbeiten in ihrem fachspezifischen Kontext zu untersuchen. Dabei wurden einige spezielle Probleme bearbeitet, jedoch kein allgemeines Verfahren zur Strukturierung von Dokumenten entwickelt.

2.3 Verallgemeinerung der Problemstellung

In diesem Abschnitt wird eine Untersuchung vorgenommen, ob und wie das Problem der Strukturierung von Diskussionsgraphen verallgemeinert werden kann. Weiterhin wird untersucht, ob die spezielle Problemlösung, die in dieser Arbeit vorgestellt wird, auch auf allgemeine Problem angewendet werden kann.

Die Grundaufgabe ist, wie bereits in Abschnitt 2.1 erläutert, einen ungeordneten Diskussionsgraphen *sinnvoll* zu strukturieren. Was eine *sinnvolle* Strukturierung ist, muss dem Lösungsverfahren von Außen mitgeteilt werden, zum Beispiel durch eine formale Spezifikation oder durch Benutzerinteraktion. Die Eingabe für ein Lösungsverfahren ist der Diskussionsgraph, also eine Menge von Knoten und Kanten, sowie die Definition von *sinnvoll*. Die Ausgabe des Verfahrens ist eine Strukturierung des Graphen.

Ein Diskussionsgraph besitzt allgemein folgende Eigenschaften:

- Diskussionsbeiträge mit bestimmten Eigenschaften (Autor, Zeit, Typ, Inhalt, Kantenstruktur zu anderen Beiträgen, ...)
- Gewichtete und / oder gerichtete Verbindungen zwischen Diskussionsbeiträgen (Widerspruch, Verknüpfung, Unterstützung)

Der Eingabewert *sinnvoll* muss derart definiert sein, dass der Algorithmus eine Vorgabe hat, nach welchen Gesichtspunkten er die Strukturierung des Graphen vornehmen soll. Er könnte beispielsweise eine Menge von Werten sein, die angibt, wie stark welche Eigenschaft des Diskussionsgraphen in den Strukturierungsprozess einfließt.

Die Ausgabe eines solchen Verfahrens ist eine Zuordnung, welcher Diskussionsbeitrag zu welcher Gruppe von Beiträgen gehört, oder alternativ, mit welcher Wahrscheinlichkeit ein Beitrag zu welchen Gruppen gehört.

Verallgemeinert man diese Problemstellung, so hat handelt es sich nicht mehr um einen Diskussionsgraphen, sondern einen Graphen mit folgenden Eigenschaften:

- Knoten mit bestimmten Eigenschaften (Datenvektor)
- Gerichtete, bewertete Kanten

Die Definition von *sinnvoll* sowie die Ausgabe eines Strukturierungsverfahrens erfolgen analog zum Spezialfall. Der einzige Faktor, der sich ändert, ist der Eingabegraph, der nicht mehr als Diskussionsgraph betrachtet wird.

Da es die Hauptaufgabe eines Lösungsverfahrens ist, zu bestimmen, wie der Graph *sinnvoll* strukturiert werden kann und diese Definition beim speziellen und allgemeinen Fall gleich ist, schließe ich draus, dass ein Algorithmus, der gute Ergebnisse für das spezielle Problem erzielt, auch gute Ergebnisse für das allgemeine Problem erzielen kann.

Sollte also der geplante dynamische, interaktive Strukturierungsalgorithmus bei Diskussionsgraphen gut funktionieren, ist auch eine Anwendung bei allgemeineren Graphen denkbar.

3 Verwendete Techniken

In diesem Kapitel werden die Grundlagen zu Techniken erläutert, die in dieser Abschlussarbeit verwendet werden, um das Problem des interaktiven Clusterings zu lösen. Dazu zählen Text Mining zur statistischen Analyse der Diskussionsinhalte, soziale Netzwerkanalyse zur strukturellen Analyse, sowie Cluster-Analyse-Techniken, die letztlich die Gruppenzuordnung der Diskussionsbeiträge bestimmen.

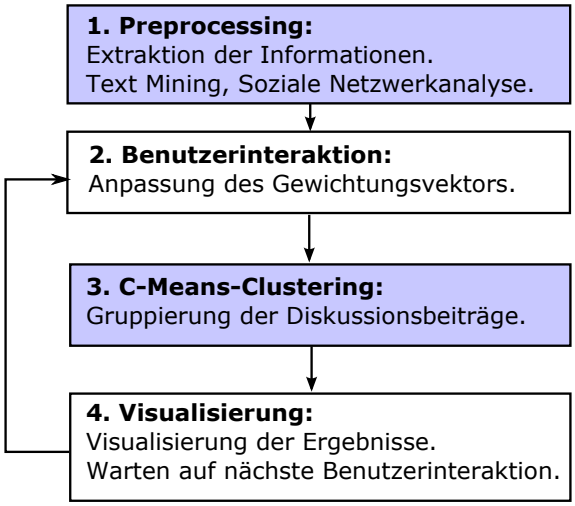


Abbildung 10: Vier Phasen des interaktiven Clustering-Verfahrens.

Abbildung 10 zeigt den konzeptuellen Ablauf des Clustering-Verfahrens. Im ersten Schritt werden die Daten aus dem Diskussionsgraphen extrahiert und im zweiten Schritt interagiert der Benutzer mit dem Moderationswerkzeug. Im dritten Schritt findet die Gruppierung der Diskussionsbeiträge statt, die im vierten Schritt visualisiert wird. In diesem Kapitel werden die Techniken vorgestellt, die in den Schritten 1 und 3 eingesetzt werden.

3.1 Text Mining

Für den Begriff *Text Mining* gibt es viele Definitionen. In dieser Arbeit richte ich mich nach der Definition von [HQW06]. Demnach ist Text Mining eine Gruppe methodischer Ansätze, die dazu verwendet werden, um Texte zu strukturieren und damit neue, relevante Informationen zu extrahieren.

Um dies zu erreichen, werden speziell musterbasierte und vor allem statistische Verfahren eingesetzt. Musterbasierte Verfahren haben das Ziel, relevante allgemeingültige Muster in einem Text zu finden und danach wiederum deren Auftreten in anderen Texten zu identifizieren. Technisch gesehen werden dabei reguläre Ausdrücke gebildet und diese mit einem gegebenen Text abgeglichen. Im geplanten Clustering-Algorithmus sollen jedoch Texte von Diskussionsbeiträgen miteinander verglichen werden, was mit den Ergebnissen von musterbasierten Verfahren nicht ohne weiteres möglich ist. Daher wird im Folgenden näher auf die statistischen Verfahren eingegangen, die bei diesem Szenario deutlich sinnvoller einsetzbar sind. Anschließend wird ein Verfahren zur Berechnung der Ähnlichkeit zwischen Dokumenten vorgestellt.

3.1.1 Statistische Verfahren

Zunächst wird die *Differenzanalyse* [HQW06, S.95-100] vorgestellt, die später zum Auffinden von ähnlichen Dokumenten verwendet wird.

Bei einer Differenzanalyse steht die Identifikation relevanter fachspezifischer Begriffe im Vordergrund. Gegeben ist ein Analysekorpus (Text einer Fachrichtung) und ein Referenzkorpus (Allgemeiner Text, zum Beispiel Zeitungsartikel). Zunächst werden die Auftretenswahrscheinlichkeiten von Wortformen beziehungsweise Wortformenkombinationen beider Textkorpora berechnet und miteinander verglichen. Daraus lassen sich vier Klassen ableiten:

- **Klasse 1:** Wortformen, die im Referenzkorpus nicht vorkommen. Diese sind mit sehr hoher Wahrscheinlichkeit Fachwörter aus dem Fachgebiet, über das im Analysekorpus berichtet wird.
- **Klasse 2:** Wortformen, die im Analysekorpus um einen bestimmten Wert häufiger vorkommen als im Referenzkorpus. Hier handelt es sich mit gewisser Wahrscheinlichkeit um Fachwörter des Fachgebietes des Analysekorpus. Zur Identifizierung eines Fachwortes muss ein Schwellwert angegeben werden, der angibt, wie groß die Differenz der Auftretenswahrscheinlichkeit zwischen Referenz- und Analysekorpus mindestens sein muss.
- **Klasse 3:** Wortformen, die im Referenz- und Analysekorpus etwa gleich häufig vorkommen. Diese sind mit hoher Wahrscheinlichkeit sogenannte Stoppwörter[7] oder allgemeine Begriffe. Sie enthalten demnach keine themenspezifischen Inhalte.
- **Klasse 4:** Wortformen, die im Referenzkorpus häufiger vorkommen als im Analysekorpus. Diese sind nach [HQW06, S.96] mit hoher Wahrscheinlichkeit ebenfalls keine Fachbegriffe aus dem Fachbereich des Analysekorpus.

Allgemein sind bei den Ergebnissen des Analyseverfahrens die Wörter aus Klasse 1 und 2 relevant, da diese die Fachwörter des Analysekorpus darstellen. Die Ergebnisse des Verfahrens können bei der Feststellung der Ähnlichkeit von Dokumenten verwendet werden, da es in diesem Fall selten sinnvoll ist, alle Wortformen aller Dokumente miteinander zu vergleichen, sondern stattdessen nur wichtige Schlagwörter.

3.1.2 Dokumentenähnlichkeit

Nach [HQW06, S.195] werden zur Gruppierung von ähnlichen Dokumenten oder Mengen von Wörtern Methoden der Cluster-Analyse eingesetzt. Insgesamt werden folgende Schritte durchgeführt [HQW06, S.202]:

1. Identifikation charakteristischer Merkmale
2. Erzeugen der Dokumentvektoren
3. Auswahl eines Ähnlichkeitsmaßes
4. Erzeugen der Ähnlichkeitsmatrix

[7] Beispielsweise Konjunktionen oder Präpositionen.

Anschließend wird eine Cluster-Analyse durchgeführt, um ähnliche Dokumente zu Gruppen zusammenzufassen. Da ein Clustering aber ohnehin im in dieser Arbeit entwickelten Verfahren geplant ist, wird dieser Schritt in Kapitel 3.3 weiter erläutert. Es folgt eine genauere Beschreibung der oben genannten Schritte.

1. Identifikation charakteristischer Merkmale: Um Dokumente miteinander zu vergleichen, werden die darin enthaltenen Wortformen verglichen. Es macht aber nur Sinn, das Auftreten bedeutsamer Wortformen[8] miteinander zu vergleichen. Zu diesem Zweck kann vorher eine Differenzanalyse durchgeführt werden, um *unnütze* Wörter vorab heraus zu filtern. [HQW06, S.202]

2. Erzeugen der Dokumentvektoren: Um die Dokumentvektoren zu bestimmen, werden einige Maßzahlen benötigt. Es wird davon ausgegangen, dass Wörter, die oft in einem Dokument vorkommen, das Dokument thematisch besser beschreiben als seltener vorkommende. [HQW06, S.203-206]

Die *Termfrequenz* $f_{i,m}$ gibt an, wie oft ein Term t_i in einem Dokument d_m vorkommt. Unter der Annahme, dass $f_{i,m}$ den Wert 5 hätte, würde es ein kurzes Dokument gut beschreiben, jedoch ein langes Dokument schlecht. Aus diesem Grund muss die *Termfrequenz normiert* werden, so dass die relative Häufigkeit berechnet wird:

$$nf_{i,m} = \frac{f_{i,m}}{\sum_{t_j \in d_m} f_{j,m}} \tag{1}$$

$nf_{i,m}$ beschreibt die Termfrequenz $f_{i,m}$ im Verhältnis zu den Termfrequenzen aller anderen vorkommenden Begriffe. Bei einem anderen Ansatz kann die Termfrequenz auch normier werden, indem $f_{i,m}$ durch die Termfrequenz des häufigsten Terms in d_m geteilt wird.

Eine weitere Forderung besagt, dass ein Wort besonders oft im Dokument, jedoch eher selten in allgemeinen Texten vorkommt. Diese Forderung wird durch die *inverse Dokumentfrequenz* idf_i erfüllt:

$$idf_i = log \frac{|d|}{|d : t_i \in d|} \tag{2}$$

Die *inverse Dokumentfrequenz* ergibt sich aus der Gesamtanzahl der Dokumente im Verhältnis zur Anzahl der Dokumente, die einen Term t_i enthalten. Die Wichtigkeit eines Terms t_i für ein Dokument d_m lässt sich mit der *normierten Termfrequenz* und der *inversen Dokumentfrequenz* berechnen:

$$w_{i,m} = nf_{i,m} \cdot idf_i \tag{3}$$

Bildet man alle Kombinationen von Termen t_i und Dokumenten d_m und berechnet daraus jeweils $w_{i,m}$, ergibt sich folgende Term-Dokument-Matrix:
Ein *Dokumentvektor* $\vec{d_m}$ besteht jeweils aus einer Zeile der Matrix:

$$\vec{d_m} = (w_{1,m}, w_{2,m}, w_{3,m}, ..., w_{n,m}) \tag{4}$$

3. Auswahl eines Ähnlichkeitsmaßes: Zur Berechnung der Ähnlichkeit von Dokumenten gibt es mehrere Maße. An dieser Stelle werden einige der wichtigsten Ähnlichkeitsmaße vorgestellt. [HQW06, S.206]

[8] Keine Stoppwörter oder allgemeine Begriffe.

	t_1	t_2	t_3	t_4	t_5	...	t_n
d_1	$w_{1,1}$	$w_{2,1}$	$w_{3,1}$	$w_{4,1}$	$w_{5,1}$	...	$w_{n,1}$
d_2	$w_{1,2}$	$w_{2,2}$	$w_{3,2}$	$w_{4,2}$	$w_{5,2}$	...	$w_{n,2}$
d_3	$w_{1,3}$	$w_{2,3}$	$w_{3,3}$	$w_{4,3}$	$w_{5,3}$	...	$w_{n,3}$
d_4	$w_{1,4}$	$w_{2,4}$	$w_{3,4}$	$w_{4,4}$	$w_{5,4}$	...	$w_{n,4}$
d_5	$w_{1,5}$	$w_{2,5}$	$w_{3,5}$	$w_{4,5}$	$w_{5,5}$	...	$w_{n,5}$

Tabelle 1: Term-Dokument-Matrix über die Dokumente d_1 bis d_5 und die Terme t_1 bis t_n. Quelle: [HQW06, S.205]

Skalarprodukt:

$$sim_{Skal}(\vec{d_i}, \vec{d_j}) = \sum_{k=1}^{n} w_{k,i} \cdot w_{k,j} \tag{5}$$

Cosinus-Maß:

$$sim_{Cos}(\vec{d_i}, \vec{d_j}) = \frac{\sum_{k=1}^{n} w_{k,i} \cdot w_{k,j}}{\sqrt{\sum_{k=1}^{n} w_{k,i}^2} \cdot \sqrt{\sum_{k=1}^{n} w_{k,j}^2}} \tag{6}$$

Mittels Umformung (8) lässt sich ein Distanzmaß in ein Ähnlichkeitsmaß überführen. So lassen sich auch Distanzmaße wie zum Beispiel die euklidische Distanz verwenden.

Euklidische Distanz:

$$dist_{Eukl}(\vec{d_i}, \vec{d_j}) = \sqrt{\sum_{k=1}^{n} (w_{k,i} - w_{k,j})^2} \tag{7}$$

$$sim(d_i, d_j) = \frac{1}{dist(d_i, d_j) + 1} \tag{8}$$

4. Erzeugen der Ähnlichkeitsmatrix: Aus den Dokumentvektoren und dem Ähnlichkeitsmaß kann die Ähnlichkeitsmatrix berechnet werden, indem von allen möglichen Kombinationen von Dokumenten die Ähnlichkeit berechnet wird. [HQW06, S.207]

Es entsteht folgende Matrix:

3.1.3 Anwendung

Um die inhaltlichen Eigenschaften von Diskussionsbeiträgen zu extrahieren, wird in dieser Arbeit das zuvor beschriebene Verfahren zur Bestimmung der Dokumentenähnlichkeit verwendet. Das Ergebnis ist eine Ähnlichkeitsmatrix, die jeden Diskussionsbeitrag mit jedem anderen verknüpft und deren inhaltliche Ähnlichkeit zueinander angibt.

Im geplanten interaktiven Clustering-Verfahren kann Text Mining schon in einer sehr frühen Phase zum Einsatz kommen, denn sobald die Ähnlichkeitsmatrix einmal berechnet wurde, ändert sie sich nur noch, falls ein Diskussionsbeitrag modifiziert, gelöscht oder

	d_1	d_2	d_3	...	d_k
d_1	$sim(d_1, d_1)$	$sim(d_1, d_2)$	$sim(d_1, d_3)$	...	$sim(d_1, d_k)$
d_2	$sim(d_2, d_1)$	$sim(d_2, d_2)$	$sim(d_2, d_3)$	...	$sim(d_2, d_k)$
d_3	$sim(d_3, d_1)$	$sim(d_3, d_2)$	$sim(d_3, d_3)$	...	$sim(d_3, d_k)$
d_4	$sim(d_4, d_1)$	$sim(d_4, d_2)$	$sim(d_4, d_3)$	...	$sim(d_4, d_k)$
...	...	...	...	...	...
d_k	$sim(d_k, d_1)$	$sim(d_k, d_2)$	$sim(d_k, d_3)$	...	$sim(d_k, d_k)$

Tabelle 2: Ähnlichkeits- / Dokument-Dokument-Matrix über die Dokumente d_1 bis d_k. Quelle: [HQW06, S.207]

neu hinzugefügt wird. Dies ist auch im Hinblick auf die Leistung des Verfahrens von Vorteil, da die Matrix nicht erst nach einer Benutzerinteraktion berechnet werden muss.

3.2 Soziale Netzwerkanalyse

Grundsätzlich untersucht die soziale Netzwerkanalyse die Beziehungen sozialer Entitäten untereinander, sowie deren Muster und Auswirkungen. Das Ziel ist es, aus Informationen von Individuen und deren Beziehungen, Aussagen über größere Strukturen des Netzwerkes treffen zu können und umgekehrt, aus strukturellen Daten bestimmte Eigenschaften von Individuen zu erklären. [Jan03]

Soziale Netzwerke werden oft als gerichtete Graphen dargestellt, wobei die Knoten einzelne Individuen und die Kanten bestimmte Beziehungen repräsentieren. In dieser Arbeit geht es jedoch nicht um solche Netzwerke, sondern um Diskussionsgraphen. Dennoch lassen sich die Methoden der sozialen Netzwerkanalyse prinzipiell auf Graphen jeder Art übertragen, um eventuell vorhandene Strukturen zu identifizieren. Die Feststellung, dass es sich bei dem Knoten eines Graphen um einen *Akteur* oder um einen *Diskussionsbeitrag* handelt, ist daher nebensächlich. Welche wichtigen Methoden es in diesem Bereich gibt und wie diese im interaktiven Clustering-Verfahren zum Einsatz kommen sollen, wird in den folgenden Abschnitten weiter erläutert.

3.2.1 Strukturanalyse-Verfahren

Ein Teilbereich der sozialen Netzwerkanalyse beschäftigt sich damit, Teilgruppen eines Netzwerkes zu betrachten und deren Eigenschaften zu analysieren. Dazu gibt es die Suche nach kohäsiven Untergruppen [WF94, S.249], die sogenannte *Cliquenanalyse*, sowie die Analyse struktureller Äquivalenz [WF94, S.361], auch *Positions-Analyse* genannt. Im Folgenden wird untersucht, wie diese Methoden im aktuellen Anwendungsfall verwendet werden können.

Die Cliquenanalyse ist im Hinblick auf diese Abschlussarbeit wenig hilfreich, da bei dem zu lösenden Problem nicht nach Diskussionsbeiträgen gesucht werden soll, die stark miteinander verbunden sind (was bei der Cliquenanalyse der Fall ist), sondern Diskussionsbeiträge, die ähnliche ein- und ausgehende Kanten haben, gefunden werden sollen. Die strukturellen Eigenschaften der Diskussionsbeiträge müssen in numerische Werte übertragen werden, damit sie miteinander verglichen werden können. Zu diesem Zweck eignet sich eine Analyse struktureller Äquivalenz.

Das Ziel der Analyse struktureller Äquivalenz ist in der sozialen Netzwerkanalyse die Suche nach sogenannten *Positionen*. Positionen bestehen aus Individuen, die auf ähnliche Art und Weise in ein Netzwerk eingebettet sind. Der Unterschied zur Cliquenanalyse besteht darin, dass Individuen, die die gleiche Position haben, weder direkt noch indirekt miteinander verbunden sein müssen; lediglich die Verbindungen zu anderen Akteuren müssen ähnlich sein.

Wie bereits erwähnt wurde, können die Ergebnisse einer Positions-Analyse dazu verwendet werden, um die Struktur eines Diskussionsbeitrages mit anderen Beiträgen vergleichen zu können. In diesem Abschnitt werden daher Techniken betrachtet, wie eine solche Analyse durchgeführt werden kann.

Nach [WF94, S.363] wird der Vorgang einer Suche nach Positionen grundsätzlich in vier Schritte unterteilt:

1. Formale Definition der Äquivalenz.
2. Maß, ob Akteure eines Netzwerks, abhängig von der Definition der Äquivalenz, äquivalent zueinander sind.
3. Repräsentation der Äquivalenzen.
4. Abschätzung der Angemessenheit der Repräsentation.

1. Formale Definition der Äquivalenz: Die Äquivalenz Definition spezifiziert, unter welchen Umständen zwei Akteure eines Netzwerkes als mathematisch äquivalent angesehen werden. Strukturelle Äquivalenz ist ein Beispiel für eine solche Definition. Es gibt noch viele weitere Definitionen, die an dieser Stelle aber nicht weiter betrachtet werden. Da es in realen Netzwerken unwahrscheinlich ist, dass mehrere Akteure perfekt äquivalent zueinander sind, wird durch das Maß der Äquivalenz bestimmt, ab welchem Grad der Ähnlichkeit zwei Akteure als äquivalent angesehen werden.

2. Maß der Äquivalenz: Das Maß der Äquivalenz ist ein mathematischer Wert, der zu der Entscheidung führen soll, ob mehrere Akteure äquivalent zueinander sind oder nicht (oder zu welchem Grad).

Die Äquivalenz kann beispielsweise durch die euklidische Distanz gemessen werden, die durch folgende Formel gegeben ist:

$$d_{ij} = \sqrt{\sum_{k=1}^{g} [(x_{ik} - x_{jk})^2 + (x_{ki} - x_{kj})^2]} \tag{9}$$

x_{ik} ist hier der Wert der Verbindung zwischen einem Akteur i und einem Akteur k. Falls zwei Akteure i und j die gleichen Verbindungen zu allen anderen Akteuren haben, ergibt sich für die euklidische Distanz der Minimalwert 0.

Eine andere Möglichkeit ist die Bestimmung der Äquivalenz durch den Korrelationskoeffizienten:

$$r_{ij} = \frac{\sum(x_{ki} - \bar{x}_{\bullet i})(x_{kj} - \bar{x}_{\bullet j}) + \sum(x_{ik} - \bar{x}_{i\bullet})(x_{jk} - \bar{x}_{j\bullet})}{\sqrt{\sum(x_{ki} - \bar{x}_{\bullet i})^2 + \sum(x_{ik} - \bar{x}_{i\bullet})^2}\sqrt{\sum(x_{kj} - \bar{x}_{\bullet j})^2 + \sum(x_{jk} - \bar{x}_{j\bullet})^2}} \tag{10}$$

x_{ik} ist auch hier der Wert der Verbindung zwischen zwei Akteuren i und k. $\bar{x}_{i\bullet}$ repräsentiert den Mittelwert der i-ten Zeile der Adjazenzmatrix, während $\bar{x}_{\bullet i}$ der Mittelwert der i-ten Spalte ist.

Dies sind die beiden am meist benutzten und bekanntesten Abstandsmaße bezüglich der Positions-Analyse; es gibt noch viele weitere, die hier jedoch nicht weiter erläutert werden.

3. Repräsentation der Äquivalenzen: Das Ziel des dritten Schrittes ist die Repräsentation der Netzwerkdaten in vereinfachter Form. Um dies zu erreichen werden zunächst die Akteure in Gruppen unterteilt und danach überprüft, auf welche Art und Weise diese Gruppen miteinander in Verbindung stehen. Letzteres wird mit sogenannten Blockmodellen realisiert.

Zur Gruppierung der Akteure in einem Netzwerk kann nach [WF94] unter anderem hierarchisches Clustering verwendet werden. Dazu muss zunächst ein Distanzmaß d_{ij} oder ein Ähnlichkeitsmaß r_{ij} bestimmt werden, sowie ein Grenzwert α. d_{ij} gibt an, wie weit zwei Akteure i und j voneinander entfernt sind, daher muss $d_{ij} \leq \alpha$ gelten, damit i und j der selben Gruppe zugeordnet werden können. Bei r_{ij} ist es dementsprechend der umgekehrte Fall $r_{ij} \geq \alpha$, da r_{ij} die Ähnlichkeit zwischen den Akteuren i und j repräsentiert. Beim hierarchischen Clustering gibt es zahlreiche Kriterien wie Single Linkage, Complete Linkage oder Average Linkage. Diese werden in Abschnitt 3.3 ausführlich beschrieben.

Die Beschreibung für das hierarchische Clustering soll an dieser Stelle nur ein Beispiel sein, denn grundsätzlich kann auch jedes andere Cluster-Analyse-Verfahren verwendet werden. Auch diese werden in Abschnitt 3.3 beschrieben.

4. Abschätzung der Angemessenheit der Repräsentation: Die Abschätzung der Angemessenheit wird in der sozialen Netzwerkanalyse zur Analyse der *Güte* eines Blockmodells, also der Beziehungen zwischen Positionen, durchgeführt. In dieser Arbeit sind diese Beziehungen jedoch nicht wichtig; die ersten Schritte der Positions-Analyse reichen zur Bewertung der strukturellen Ähnlichkeit von Diskussionsbeiträgen aus.

Abbildung 11 zeigt ein Beispiel für den Ablauf einer Positions-Analyse. Gegeben ist ein Graph (b.) und eine zugehörige Adjazenz- oder Soziomatrix. Nach Permutation und Partitionierung entsteht eine neue Matrix (c.), in der die Positionen besser erkennbar sind. Die Blöcke werden danach zu einzelnen Werten zusammengefasst, was eine sogenannte Image-Matrix (d.) ergibt. Der Graph (e.) repräsentiert die gefundenen Positionen.

Für den geplanten Clustering Algorithmus ist die Matrix (c.) bereits ein gutes Mittel, um die strukturelle Äquivalenz von bestimmten Diskussionsbeiträgen (zum Beispiel 6, 3 und 8) festzustellen.

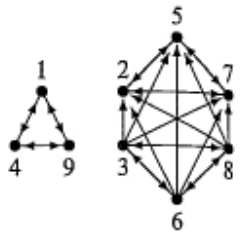
3.2.2 Anwendung

Um die strukturellen Eigenschaften von Diskussionsbeiträgen zu extrahieren, wird innerhalb des geplanten Clustering-Verfahrens eine Positions-Analyse durchgeführt. Das Ergebnis dieser Analyse ist eine permutierte und partitionierte Adjazenzmatrix, die jeden Diskussionsbeitrag mit jedem anderen verknüpft und deren strukturelle Ähnlichkeit zueinander angibt.

a. Sociomatrix

	1	2	3	4	5	6	7	8	9
1	-	0	0	1	0	0	0	0	1
2	0	-	0	0	1	0	1	0	0
3	0	1	-	0	1	1	1	1	0
4	1	0	0	-	0	0	0	0	1
5	0	1	0	0	-	0	1	0	0
6	0	1	1	0	1	-	1	1	0
7	0	1	0	0	1	0	-	0	0
8	0	1	1	0	1	1	1	-	0
9	1	0	0	1	0	0	0	0	-

b. Graph



c. Permuted and partitioned sociomatrix

	6	3	8	4	1	9	2	5	7
6	-	1	1	0	0	0	1	1	1
3	1	-	1	0	0	0	1	1	1
8	1	1	-	0	0	0	1	1	1
4	0	0	0	-	1	1	0	0	0
1	0	0	0	1	-	1	0	0	0
9	0	0	0	1	1	-	0	0	0
2	0	0	0	0	0	0	-	1	1
5	0	0	0	0	0	0	1	-	1
7	0	0	0	0	0	0	1	1	-

d. Image matrix

	$\mathscr{B}_1$	$\mathscr{B}_2$	$\mathscr{B}_3$
$\mathscr{B}_1$	1	0	1
$\mathscr{B}_2$	0	1	0
$\mathscr{B}_3$	0	0	1

e. Reduced graph

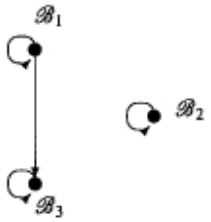


Abbildung 11: Beispiel für den Ablauf einer Positions-Analyse. Quelle: [WF94, S.364]

Eine Positions-Analyse kann bereits in der Initialisierungsphase des geplanten Clustering-Algorithmus zum Einsatz kommen. Wird der Diskussionsgraph, und damit die strukturellen Eigenschaften der einzelnen Diskussionsbeiträge, modifiziert so muss auch die zugehörige Adjazenzmatrix angepasst und neu analysiert werden. Dies ist notwendig, wenn ein Benutzer Knoten oder Kanten löscht oder hinzufügt. Die Veränderung von inhaltlichen Eigenschaften der Diskussionsbeiträge ist an dieser Stelle unwichtig.

3.3 Cluster-Analyse

Bei Cluster-Analyse Verfahren geht es darum, aus unstrukturierten Daten deren Klassenzugehörigkeiten zu schätzen beispielsweise eine Menge von unbekannten Objekten anhand ihrer Eigenschaften (Merkmalsvektoren) bestimmten Gruppen (Clustern) zuzuordnen. [Run00, S.72]

In diesem Abschnitt werden einige bekannte Clustering-Verfahren wie hierarchisches Clustering oder C-Means-Clustering vorgestellt. Danach werden diese ausgewertet und der am besten geeignete Algorithmus für die Implementierung des interaktiven Clustering-Verfahrens ausgewählt.

3.3.1 Hierarchische Verfahren

Eine gute Aufteilung ist allgemein dann gegeben, wenn alle Objekte innerhalb eines Clusters einen geringen Abstand zueinander haben, während alle Objekte aus verschiedenen Clustern einen großen Abstand zueinander haben. Aus dieser Definition lassen sich sequenzielle hierarchische nichtüberlappende Clustermodelle schließen. [Run00, S.73]

Die sequenzielle agglomerative hierarchische nichtüberlappende (SAHN) Cluster-Analyse [Run00, S.73] geht davon aus, dass jeder Datensatz x_k (mit $k = 1, ..., n$) zunächst ein eigenes Cluster ist. Demnach gibt es $c_0 = n$ verschiedene Cluster $C_{0k} = \{x_k\}$ (mit $k = 1, ..., n$). In jedem Iterationsschritt i des SAHN-Algorithmus werden zwei Cluster C_{ia} und C_{ib} gesucht, die den geringsten Abstand zueinander haben und danach zu einem einzigen Cluster verschmolzen.

Der Abstand zwischen zwei Clustern kann auf verschiedenste Weise gemessen werden. Es folgen einige bekannte Varianten:

Minimalabstand (Single Linkage):

$$d(C_a, C_b) = \min_{x \in C_a, y \in C_b} \|x - y\| \tag{11}$$

Maximalabstand (Complete Linkage):

$$d(C_a, C_b) = \max_{x \in C_a, y \in C_b} \|x - y\| \tag{12}$$

Mittlerer Abstand (Average Linkage):

$$d(C_a, C_b) = \frac{1}{\|C_a\| \cdot \|C_b\|} \sum_{x \in C_a, y \in C_b} \|x - y\| \tag{13}$$

Abbildung 12 (oben) zeigt die Anwendung verschiedener Abstandsmaße zwischen zwei Clustern. Links Maximal- und Minimalabstand und rechts der mittlere Abstand.

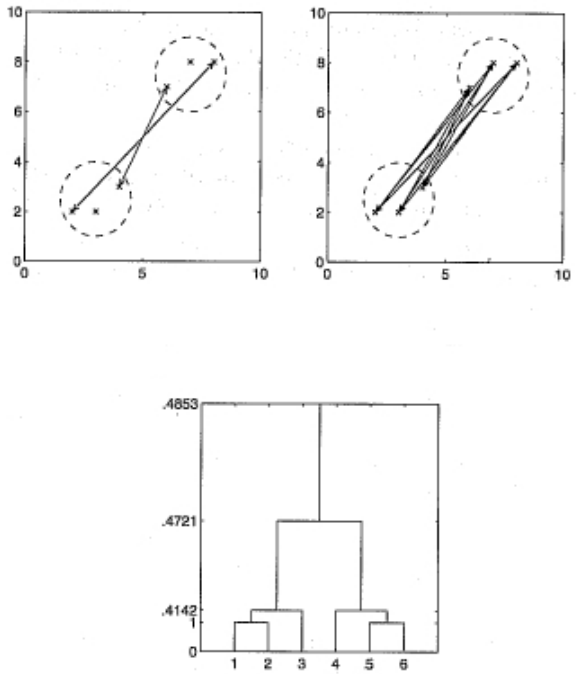


Abbildung 12: Verschiedene Abstandsmaße und Dendogramm. Quelle: [Run00, S.75]

Das oben beschriebene Verfahren der Verschmelzung von Clustern wird so lange durchgeführt bis es nur noch ein einziges, großes Cluster gibt, das alle Elemente enthält. Die Ergebnisse werden in der Regel in einem Dendogramm dargestellt, wie in Abbildung 12 (unten) zu sehen ist. Äquivalent können die Ergebnisse auch in einer *Partitionsmatrix* U dargestellt werden, deren Elemente wie folgt definiert sind:

$$u_{ik} = \begin{cases} 1, & \text{falls } x_k \in C_i \\ 0, & \text{falls } x_k \notin C_i \end{cases} \tag{14}$$

Eine Variante des SAHN-Algorithmus ist DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Der einzige Unterschied zum Grundverfahren ist hier die Wahl des Kriteriums zur Verschmelzung von Clustern. Anstelle von Abstandsmaßen werden Dichteschätzer verwendet. [Run00, S.76]

Analog zum SAHN-Algorithmus gibt es auch die sequenzielle divisive hierarchische nichtüberlappende (SDHN) Cluster-Analyse. Diese beginnt mit einem einzigen Cluster, das alle Datenelemente enthält und teilt dieses solange auf, bis es so viele Cluster wie Elemente gibt. [Run00, S.76]

3.3.2 Nicht-Hierarchische Verfahren

Anstatt wie beim hierarchischen Clustering Partitionsmengen und -matrizen zur Repräsentation von Clustern zu verwenden, gibt es auch die Möglichkeit, Prototypen wie beispielsweise Cluster-Zentren $V = \{v_1, ..., v_c\}$ zu verwenden. Jedes Cluster wird durch ein

Cluster-Zentrum v_i (mit $i = 1, ..., c$) repräsentiert. Gesucht ist die Zuordnung der Datenelemente x_k (mit $k = 1, ..., n$) zu den Cluster-Zentren v_i.

Die Partition von Datenelementen kann mit Hilfe einer Partitionsmatrix U vorgenommen werden. Sie enthält Elemente, die wie folgt definiert sind:

$$u_{ik} = \begin{cases} 1, \text{ falls } x_k \in C_i \\ 0, \text{ falls } x_k \notin C_i \end{cases} \tag{15}$$

Da jedes Cluster mindestens ein Element besitzt, gilt für jede Partitionsmatrix, dass die Summe jeder Spalte größer als 0 ist. Andererseits muss die Summe jeder Zeile genau 1 betragen, da die Cluster disjunkt sind und demnach jedes Datenelement genau zu einem Cluster gehört.

Eine Möglichkeit, um die Zuordnung von Datenelementen zu Cluster-Zentren zu bestimmen, ist die Nächster-Nachbar-Regel. Sie besagt, dass x_k genau dann zu einem Cluster i gehört, wenn

$$\|x_k - v_i\| = \underset{j\,=\,1,\,...,\,c}{min} \|x_k - v_j\| \tag{16}$$

Daher wird jedes Datenelement dem Cluster-Zentrum zugeordnet, dem es am nächsten ist.

Die Bestimmung der Cluster-Zentren erfolgt über das C-Means-Modell. [Run00, S.77] Dieses Modell wird über die Minimierung der Summe der quadratischen Abstände zwischen den Zentren und den zugehörigen Datenelementen definiert. Gesucht ist eine Minimierung der Zielfunktion:

$$J_{CM}(U, V; X) = \sum_{i=1}^{c} \sum_{x_k \in C_i} \|x_k - v_i\|^2 = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik} \|x_k - v_i\|^2 \tag{17}$$

Weiterhin folgt aus den Bedingungen für (lokale) Extrema für die Cluster-Zentren:

$$v_i = \frac{1}{\|C_i\|} \sum_{x_k \in C_i} x_k = \frac{\sum_{k=1}^{n} u_{ik} x_k}{\sum_{k=1}^{n} u_{ik}} \tag{18}$$

Die sogenannte alternierende Optimierung verwendet das C-Means-Modell, um ein vollständiges Clustering durchzuführen. Zu Beginn werden die Cluster-Zentren zufällig initialisiert. Danach erfolgt eine abwechselnde Berechnung der Partitionsmatrix und der Cluster-Zentren. Dieser Vorgang wird solange durchgeführt, bis sich die Cluster-Zentren V weniger als der Wert einer vorher festgelegten Abbruchgrenze ϵ ändern.

Zusätzlich zu diesem Verfahren gibt es das Konzept der unscharfen Partitionen, das besagt, dass ein Datenelement auch mehreren Cluster-Zentren zugehörig sein darf. Es kommt beispielsweise beim Fuzzy-C-Means-Clustering zum Einsatz. [Run00, S.79] Ziel des Verfahrens ist die Minimierung folgender Kostenfunktion:

$$J_{FCM}(U, V; X) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m \|x_k - v_i\|^2 \tag{19}$$

Zusätzlich gelten folgende Nebenbedingungen:

$$\sum_{k=1}^{n} u_{ik} > 0, i = 1, ..., c \tag{20}$$

Ein Cluster enthält somit mindestens ein Element, der Wert ist also echt größer als 0.

$$\sum_{i=1}^{c} u_{ik} = 1, k = 1, ..., c \tag{21}$$

Die Aussage dieser Formel ist, dass die Summe der Cluster-Zugehörigkeiten für ein Datenelement 1 ist. Beim normalen C-Means-Clustering oder beim hierarchischen Clustering ist dies trivialerweise immer gegeben, da ein Datenelement nur genau einem Cluster angehören kann. Beim Fuzzy-C-Means-Clustering ist diese Bedingung jedoch zwingend notwendig, damit ein Datenelement beispielsweise zwei Clustern zu Werten von jeweils $\frac{1}{2}$ angehört; die Summe muss immer genau 1 ergeben.

Der Parameter m aus der Fuzzy-C-Means-Kostenfunktion dient zur Festlegung der Unschärfe der Cluster und liegt im Wertebereich $m \in (1, \infty)$. Beim Spezialfall $m \to 1$ ist das FCM-Modell gleich dem CM-Modell; es gibt also keine Unschärfe. Ist $m \to \infty$ so erhalten alle Datenpunkte die gleiche Zugehörigkeit zu allen Clustern. Ein häufig verwendeter Wert ist $m \to 2$, da er einen guten Kompromiss zwischen Restriktion und Unschärfe bietet. [Run00, S.80]

Quick-Clustering-Verfahren: Die Güte des C-Means-Clustering hängt stark von der Initialisierung der Clustering-Prototypen ab. Ein Ansatz, der von [Bac96, S.340] vorgestellt wird, besteht darin, die Prototypen so zu platzieren, dass sie möglichst weit voneinander entfernt sind. Der Ablauf dieses Verfahrens ist wie folgt:

Schritt 1: Zunächst bilden die ersten C Datenelemente die Cluster-Prototypen c_C. Nun wird eine Zählervariable g auf den Wert $C + 1$ gesetzt, so dass das nächste Datenelement betrachtet wird. [Bac96, S.340]

Schritt 2: Für das Objekt g wird folgender Wert berechnet. $dmin_{g,q}$ ist die minimale Distanz des Objektes g zu einem der C Prototypen (der Index dieses Prototypen sei hier q). $dmin_{k^*,k^{**}}$ ist die minimale Distanz zwischen den C Prototypen, demnach sind die Prototypen mit den Indices k^* und k^{**} am nächsten zueinander. $dmin_{q,k}$ ist die minimale Entfernung des Cluster-Prototypen c_q zu den anderen C Prototypen. [Bac96, S.340]

Wenn eine der folgenden Bedingungen erfüllt ist, ist es sinnvoll, das Objekt g anstelle des bisherigen Cluster-Zentrums c_k als neuen Prototypen zu verwenden:

1. $dmin_{g,q} > dmin_{k^*,k^{**}}$: g sollte als neues Cluster-Zentrum verwendet werden, da es zu einer besseren Trennung der Cluster führt. Wenn die Distanz von g zu k^* kleiner ist als die Entfernung zu k^{**}, wird g anstelle von k^* verwendet, ansonsten anstelle von k^{**}. Diese Änderung vergrößert die kleinste Entfernung zwischen den Cluster-Prototypen. [Bac96, S.340]

2. Die erste Bedingung ist nicht erfüllt und es gilt $dmin_{g,q} > dmin_{q,k}$: In diesem Fall wird g anstelle von c_q als Prototyp verwendet. Diese Änderung vergrößert die kleinste Entfernung zwischen q (durch g ersetzt) und den restlichen Prototypen. [Bac96, S.340]

Schritt 3: Die Zählervariable g wird um 1 erhöht. Falls g größer als die Anzahl der Datenelemente wird, wird der Algorithmus beendet. Ansonsten wird mit Schritt 2 fortgefahren. [Bac96, S.340]

Silhouetten-Koeffizient: Ein weiteres Problem des C-Means-Modells ist die Berechnung der optimalen Cluster-Anzahl C, was mit Hilfe des Silhouetten-Koeffizienten gelöst werden kann [TKS05, S.541-542]. Der Silhouetten-Koeffizient ist eigentlich eine Maßzahl, die eine Aussage dazu macht, wie *gut* ein Clustering ist. Er kann jedoch auch zur Bestimmung von C verwendet werden, indem vorab Clusterings mit 2 bis $N-1$ Clustering-Prototypen berechnet und zu jedem der Silhouetten-Koeffizient berechnet. Das Clustering mit dem höchsten Koeffizienten ist das Beste und das entsprechende C wird im tatsächlichen Clustering verwendet.

Der Silhouetten-Koeffizient für ein Clustering mit x Prototypen berechnet sich wie folgt. Zunächst wird für jedes Objekt i die durchschnittliche Distanz zu allen anderen Objekten innerhalb seines eigenen Clusters bestimmt (a_i). Danach wird für jedes Objekt i die durchschnittliche Distanz zu allen anderen Objekten des nächstgelegenen Clusters bestimmt (b_i). Der Silhouetten-Koeffizient s_i ist damit für ein Objekt i:

$$s_i = \frac{b_i - a_i}{max(a_i, b_i)} \tag{22}$$

Falls $a_i = 0$, so gilt auch $s_i = 0$, da sich das betrachtete Objekt in diesem Fall alleine in einem Cluster befindet.

s_i muss für alle gegebenen Objekte berechnet und danach der Durchschnitt gebildet werden. Der daraus resultierende Wert liegt zwischen -1 und $+1$. -1 bedeutet für ein Objekt, dass es eher näher an Objekten des anderen Clusters und $+1$, dass es nahe bei den Objekten des eigenen Clusters liegt. Je näher der Durchschnittswert also an $+1$ ist, desto besser ist das Clustering gelungen, da dies bedeutet, dass die meisten Objekte nah an den anderen Objekten des eigenen Clusters sind. Silhouetten-Koeffizienten mit einem Wert über 0.51 deuten auf eine gute, Koeffizienten mit Werten über 0.76 auf eine sehr gute Struktur hin.

3.3.3 Selbstorganisierende Karten

Selbstorganisierende Karten (self organizing maps, SOM) [Zel94, S.179] sind unüberwachte Lernverfahren, die auf der lernenden Vektorquantisierung (learning vector quantization, LVQ) [Zel94, S.171] basieren, und letztlich auf Teuvo Kohonen [Koh90] zurückgehen.

Sowohl bei LVQ als auch bei SOM handelt es sich um einschichtige neuronale Netze, das heißt es gibt Ein- und Ausgabeneuronen, sowie eine Zwischenschicht mit Neuronen, bei denen eine Veränderung der Eingabedaten aufgrund bestimmter Gewichte stattfindet. Die Gewichte werden nur durch die Eingabe und bei LVQ durch die Klasseninformationen bestimmt. LVQ ist demnach, im Gegensatz zu SOM, ein überwachtes Lernverfahren.

Bei LVQ gibt es eine Menge von Eingabevektoren $X_1, X_2, ..., X_p$, sowie eine Menge von Codebook- beziehungsweise Referenzvektoren, die möglichst so im Eingaberaum verteilt werden sollen, dass sie ihn gut abdecken. Für jede Klasse der Eingabevektoren gibt es in

der Regel einen oder mehrere Codebook-Vektoren. Die Klassifizierung eines neuen Eingabevektors findet so statt, dass der Vektor mit allen Codebook-Vektoren verglichen wird und daraus der Ähnlichste gesucht wird. Der Codebook-Vektor ist ein Gewichtungsvektor $W_j = (w_{1j}, ..., w_{nj})$ eines Kohonen-Neurons j, das von n Eingabeneuronen seine Eingabe erhält. Er beinhaltet die Klasseninformationen für den neuen Eingabevektor.

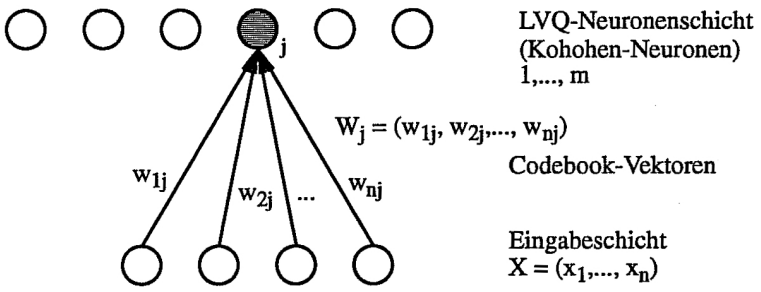


Abbildung 13: Netzstruktur der lernenden Vektorquantisierung. Quelle: [Zel94, S.172]

Zur Klassifizierung und zum Lernen gibt es bei LVQ zahlreiche Verfahren, beispielsweise LVQ1 [Zel94, S.172], das die Ähnlichkeit von Eingabe- und Codebook-Vektoren mit der euklidischen Distanz misst und das gefundene *Gewinner-Neuron* dem Eingabevektor ähnlicher macht.

Selbstorganisierende Karten sind eine Weiterentwicklung der lernenden Vektorquantisierung und führen eine lokale Nachbarschaftsbeziehung der Neuronen ein. Wie schon bei LVQ handelt es sich auch hier um einschichtige neuronale Netze mit einer Schicht aktiver Neuronen. Sie werden jedoch mit unüberwachten Lernverfahren trainiert. [Zel94, S.179]

Wie bei der LVQ gibt es eine Menge von Eingabevektoren $X_1, X_2, ..., X_p$, sowie eine Menge von Codebook-Vektoren, die diesen Raum möglichst gut abdecken sollen. Zusätzlich kommt nun noch eine lokale Nachbarschaftsbeziehung zwischen den Neuronen in einem m-dimensionalen Gitter hinzu.

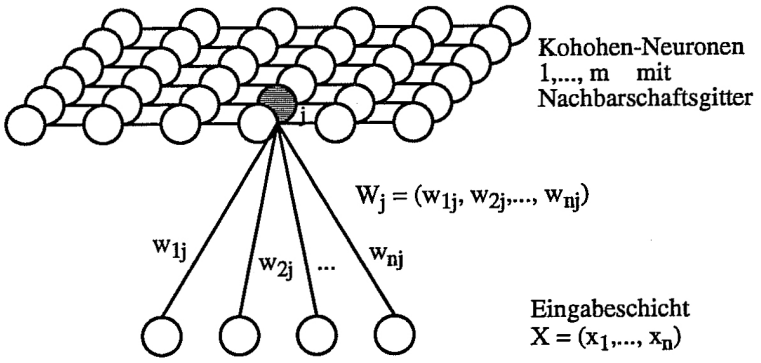


Abbildung 14: Netzstruktur der selbstorganisierenden Karten. Quelle: [Zel94, S.180]

Beim Training werden die Codebook-Vektoren durch eine spezielle Art der Vektorquantisierung so im n-Dimensionalen Eingaberaum angeordnet, dass sie die Eingabevektoren

räumlich gut repräsentieren. Das Netz der Codebook-Vektoren kann man sich gut als *elastisches Gitternetz* [Zel94, S.179] vorstellen. Durch die lokalen Nachbarschaftsbeziehungen hat eine Änderung eines Codebook-Vektors auch Auswirkungen auf die Nachbarn dieses Vektors. Je weiter (zwei Vektoren von) zwei Neuronen im Netz voneinander entfernt sind, desto weniger beeinflussen sie sich gegenseitig.

Bei den SOM wird durch besagtes *Gitternetz* eine nichtlineare Regression der Referenzvektoren durch die Datenpunkte der Eingabevektoren definiert. Gleichzeitig gibt es eine topologieerhaltende Abbildung der n-dimensionalen Eingabevektoren aus R^n auf das m-dimensionale Gitter der Neuronen. Diese Abbildung wird durch das unüberwachte Lernverfahren der SOM iterativ berechnet. Mit Hilfe dieses Lernverfahrens können Cluster in den Eingabedaten erkannt sowie neue Eingabemuster schon erkannten Clustern zugeordnet werden.

Das Lernverfahren läuft bei SOM folgendermaßen ab: Es gibt einen n-dimensionalen Eingabevektor $X = (x_1, ..., x_n)$, der mit allen Gewichtsvektoren $W_j = (w_{1j}, ..., w_{nj})$ in einer beliebigen Metrik[9] verglichen wird. Gesucht ist das Neuron c, dessen Gewichtsvektor W_c dem Eingabevektor X am ähnlichsten ist:

$$\|X - W_c\| = min_j(\|X - W_j\| \tag{23}$$

Eine Alternative ist die Funktion arg, die den Index der Minimumsuche liefert:

$$c = arg\ min_j(\|X - W_j\|) \tag{24}$$

Während des Lernens werden außer dem Gewichtsvektor des *Gewinner-Neuron* selbst auch die Gewichtsvektoren von dessen topologischen Nachbarn im Neuronengitter modifiziert. Die Form und Größe der Nachbarschaft kann sich dabei mit der Zeit ändern. In der Regel wird mit einer sehr großen Umgebung begonnen, die dann immer kleiner wird:

$$W_j(t+1) = W_j(t) + \eta(t)h_{cj}(t)[X(t) - W_j(t)] \tag{25}$$

Hier ist $h_{cj}(t)$ eine Distanzfunktion (oder auch Nachbarschaftsfunktion) auf dem Gitter der Neuronen. $\eta(t)$ ist die zeitlich veränderliche Lernrate. Üblicherweise ist sie eine monoton fallende Funktion mit $0 < \eta(t) < 1$ und es gilt

$$h_{cj}(t) = h(\|r_c - r_j\|, t) = h(z, t) = h'(z, d) \tag{26}$$

r_c und r_j sind die Vektoren der Positionen der Neuronen c und j im Gitter. Je größer die Distanz $z = \|r_c - r_j\|$ wird, desto kleiner wird die Distanzfunktion ($h_{cj}(t) \to 0$). Aus praktischen Gründen kann auch die Distanzfunktion statt mit dem Zeitparameter t mit dem Distanzparameter d formuliert werden. d gibt die Größe der Umgebung an (mit $d \to 0$ für $t \to \infty$).

Es gibt viele verschiedene Nachbarschaftsfunktionen [Zel94, S.183]. Der erste Vorschlag von Kohonen ist konstant für eine bestimmte Anzahl von Nachbarn des *Gewinner-Neurons* c und sonst Null:

$$h_{cj,bubble}(t) = \begin{cases} 1, & \text{falls } j \in N_c(t) \\ 0, & \text{sonst} \end{cases} \tag{27}$$

[9] Beispielsweise die euklidische Norm. Bei normalisierten Vektoren wird oft das Skalarprodukt verwendet.

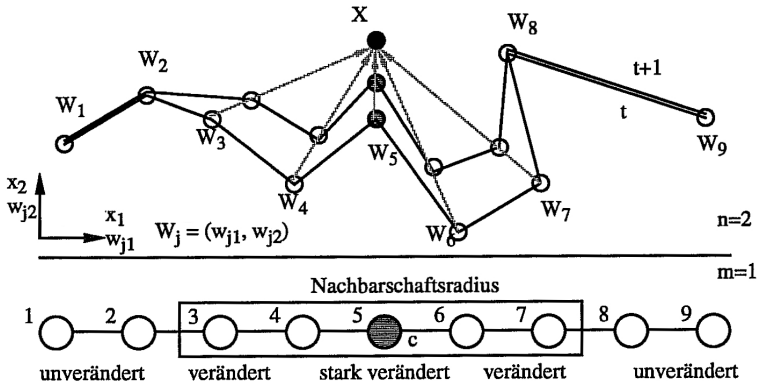


Abbildung 15: Modifikation der zweidimensionalen Gewichtsvektoren der Neuronen einer selbstorgansierenden Karte während des Lernvorgangs. Quelle: [Zel94, S.182]

Kohonens zweiter Vorschlag ist die normalisierte Gaußfunktion, wobei die Varianz $\sigma(t)$ der Gaußfunktion die Breite des Nachbarschaftskerns definiert:

$$h_{cj,gauss}(t) = \frac{1}{\sigma(t)\sqrt{2\pi}} \cdot exp\left(-\frac{\|r_c - r_j\|^2}{2\sigma^2(t)}\right) \tag{28}$$

3.3.4 Anwendung

Aus diesen drei Verfahren soll im Folgenden das am besten geeignete ausgewählt werden.

Das hierarchische Clustering hat durch die während des Gruppierungsvorgangs gebildete Baumstruktur den Vorteil, dass keine initiale Anzahl an Clustern vorgegeben werden muss; der Clustering-Vorgang kann vorgenommen werden und erst danach muss der Anwender anhand der Ergebnisse entscheiden, wie viele Cluster sinnvoll wären. Zudem ist es in sämtlichen Datenräumen nutzbar, sofern sich ein geeignetes Distanzmaß definieren lässt. Die Nachteile des hierarchischen Clusterings rühren von den Eigenschaften der verschiedenen Distanzmaße her, die dazu führen, dass je nach verwendetem Distanzmaß andere Arten von Clustern entstehen: Beispielsweise entstehen bei Complete Linkage tendenziell viele kleine Clustern, während bei Single Linkage oftmals unerwünscht längliche Cluster entstehen [Bac96, S.146]. Zudem müsste zusätzliches Wissen in den Algorithmus eingebracht werden, um zu entscheiden, wie viele Cluster es geben soll.

Hierarchisches Clustering ist aufgrund der vorher nicht festgelegten Cluster-Anzahl vielseitig einsetzbar, liefert jedoch je nach Abstandsmaß unterschiedliche Ergebnisse. Um für verschiedenste Verteilungen von Daten vernünftige Ergebnisse erzielen zu können, muss daher das optimale Abstandsmaß vor jedem Clustering automatisch gewählt werden. Ein weiterer Nachteil ist die im Vergleich zum C-Means-Clustering hohe Zeitkomplexität[10]. Speziell der letzte Aspekt, der in dieser Abschlussarbeit eine wichtige Rolle spielt, macht das hierarchische Clustering für das geplante Verfahren unbrauchbar.

Selbstorganisierende Karten liefern qualitativ gute Ergebnisse, wenn eine hinreichend große Datenbasis sowie ausreichend Zeit für eine Trainingsphase zur Verfügung stehen.

[10] Mindestens $O(n^2)$.

Ein Nachteil besteht zudem darin, dass während der Anwendungsphase ein hoher Rechenaufwand anfällt.

Aufgrund der Voraussetzung der Echtzeitfähigkeit und Interaktivität für das geplante Verfahren lassen sich selbstorganisierende Karten nicht verwenden, da immer eine Trainingsphase durchgeführt werden muss, die eine große Zahl von Rechenschritten erfordert. [Zel94, S.187]

C-Means-Clustering ist einfach zu implementieren und benötigt eine geringe Anzahl von Iterationsschritten bei der Anwendung. Nachteilig wirken sich hier Ausreißer beziehungsweise Rauschen auf den Clustering-Vorgang aus, da alle Datenelemente in die Berechnung der Cluster-Zentren mit einfließen. Ein weiteres Problem ist die Suche nach einer guten Initialisierung der Cluster-Zentren, von denen die Güte des Ergebnisses sowie die Laufzeit maßgeblich abhängen.

C-Means-Clustering liefert in der Regel auch bei zufällig initialisierten Cluster-Zentren und beliebigen Mengen von Datenelementen solide Ergebnisse und ist, im Vergleich zu den anderen Verfahren, schnell [TKS05, S.510-513]. Da letzteres ein Hauptaspekt des geplanten Gruppierungsverfahrens ist, soll C-Means-Clustering verwendet werden, selbst wenn die Qualität der Ergebnisse geringfügig schlechter ist als bei anderen Verfahren. Zudem gibt es Verfahren wie den Silhouetten-Koeffizienten und das Quick-Clustering-Verfahren, um nichtdeterministische Elemente zu entfernen und die Qualität der Ergebnisse des C-Means-Clusterings zu verbessern.

4 Interaktionstechniken

Dieses Kapitel beschreibt, wie ein Benutzer ein Clustering-Verfahren steuern kann. In den ersten Abschnitten wird der Kontext zwischen Interaktion und Cluster-Analyse erläutert, gefolgt von einer kurzen Anmerkung zur Visualisierung der Clustering-Ergebnisse. Der Hauptteil des Kapitels beschäftigt sich mit einer Möglichkeit, wie Interaktionstechniken objektiv bewertet werden können, um aus einer großen Menge eine qualitative Auswahl zu treffen. Nachdem einige Interaktionstechniken vorgestellt wurden, werden die zuvor beschriebenen Kriterien auf diese angewendet, um zu bestimmen, welche Techniken implementiert werden. Ziel des Kapitels ist es, drei geeignete Interaktionstechniken zu finden.

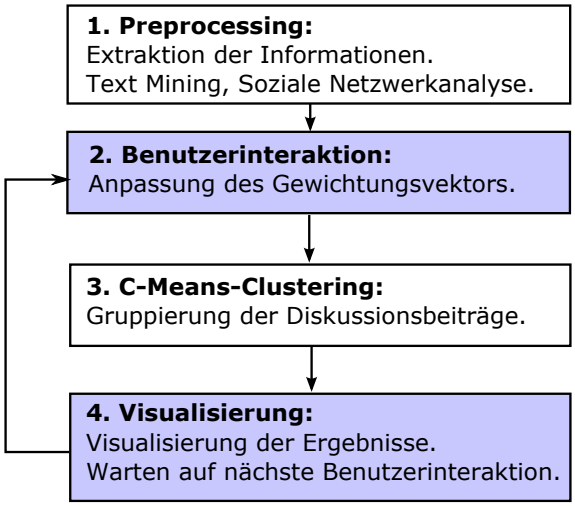


Abbildung 16: Vier Phasen des interaktiven Clustering-Verfahrens.

Im Hinblick auf den Ablauf des Clustering-Verfahrens (siehe Abbildung 16), werden in diesem Kapitel die Techniken beschrieben, die für die Schritte 2 und 4 notwendig sind.

4.1 Interaktion und Clustering

Die Interaktion zwischen dem Moderator und dem Moderator's Interface findet in den beschriebenen Systemen hauptsächlich Maus-basiert statt. Um beispielsweise einen Diskussionsbeitrag zu verschieben, kann der Benutzer diesen mit der linken Maustaste *aufnehmen* und an seiner Zielposition wieder *loslassen*. Er kann auch mehrere Beiträge auswählen, indem er vorher (ebenfalls mit der linken Maustaste) einen Rahmen um seine Auswahl zieht, und diese danach verschieben.

Nehmen wir an, im Clustering-Algorithmus gibt es einen *Gewichtungsvektor*, der zu jedem Parameter eine Aussage trifft, wie stark dieser in den aktuellen Gruppierungsvorgang mit einfließen soll. Dieser Gewichtungsvektor sollte grundsätzlich neutral eingestellt sein, so dass jeder Parameter die gleiche Gewichtung erhält.

Abbildung 17 zeigt ein abstraktes Beispiel für die Anpassung des Gewichtungsvektors durch eine bestimmte Benutzerinteraktion. Der Benutzer kann in diesem Fall Diskussionsbeiträge auf der Ebene mit der Maus verschieben. Gegeben ist ein Gewichtungsvektor

w mit 2 Werten: Der erste für die Farbe (rot / grün) und der zweite für den Inhalt (A / B / C) eines Knotens. Zunächst bewegt der Benutzer einen Knoten zu einem anderen (Schritt 1). Danach werden der aktivierte Knoten sowie seine Nachbarknoten auf Ähnlichkeiten untersucht (Schritt 2). Die Suche nach den Nachbarknoten wurde in diesem Beispiel durch eine kreisförmige, räumliche Abtastung um den aktivierten Knoten herum dargestellt. Der Gewichtungsvektor wird angepasst, Parameter 1 abgeschwächt und Parameter 2 erhöht (Schritt 3). Der Algorithmus sollte daran erkennen, dass nicht die Farbe, sondern der Inhalt der Knoten der für den Moderator wichtige Parameter ist. Danach wird eine Cluster-Analyse für die anderen Knoten durchgeführt (Schritt 4).

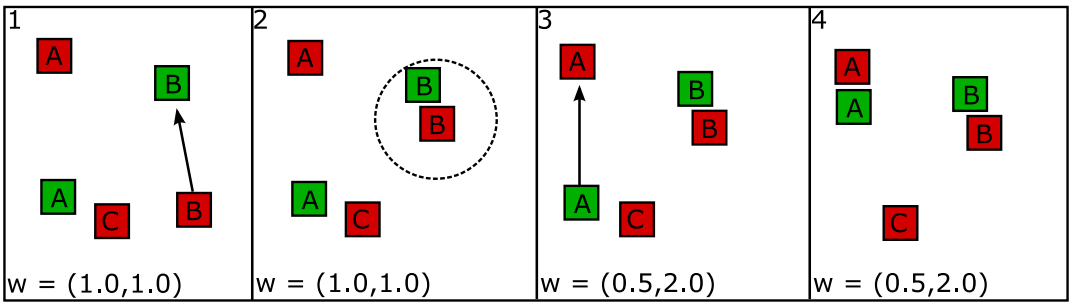


Abbildung 17: Beispiel für die Anpassung des Gewichtungsvektors durch eine bestimmte Benutzerinteraktion.

Bei der oben beschriebenen Interaktionstechnik ist es für den Clustering-Algorithmus wichtig zu wissen, ob und wie der Moderator mit den Diskussionsbeiträgen interagiert. Hat er diese Daten erhalten, kann er versuchen, sie zu interpretieren, was eine Anpassung des Gewichtungsvektors zur Folge hat. Beispielsweise muss er den Parameter *Autor* höher bewerten, wenn der Benutzer mehrere Beiträge gruppiert hat, die alle von dem selben Autoren stammen. Umgekehrt muss er eventuell diesen Parameter abschwächen, wenn der Moderator (wenn auch möglicherweise zufällig) nur Beiträge von unterschiedlichen Autoren zusammen gelegt hat, da der Autor offensichtlich keine Rolle in der Gruppenzuordnung spielt.

Zur Modifikation des Gewichtungsvektors werden also die Interaktionsdaten sowie der ursprüngliche Gewichtungsvektor benötigt. Nachdem dies berechnet wurde, kann der neue Gewichtungsvektor für die Cluster-Analyse verwendet werden.

Die Benutzerinteraktion ist der wichtigste Faktor für die Erzeugung des Gewichtungsvektors: Die Entscheidung, welcher Parameter *wichtig* ist, wird, im Gegensatz zu einem statischen Verfahren, nicht dem Algorithmus überlassen, sondern vom Benutzer getroffen. Es fließt also externes Wissen in den Gruppierungsvorgang mit ein. Ziel dieses Verfahrens ist es, zu überprüfen, ob dieses externe Wissen das Clustering verbessern kann, so dass der Benutzer anhand der Ergebnisse eine bessere Übersicht über die Diskussion gewinnen und gegebenenfalls leichter Informationen aus dieser extrahieren kann.

4.2 Visualisierung der Clustering-Ergebnisse

Ein Faktor, der bei der Bewertung der Interaktionstechniken zu beachten ist, ist die Visualisierung der Clustering-Ergebnisse. Im Moderator's Interface gibt es eine *Springy Physics*

Visualisierung, die die Kräfte von Sprungfedern simuliert. Dieses Konzept wurde bereits 1981 von Platt und Badler [PB81] zur Simulation von Bewegungen eingesetzt.

Ursprünglich bezweckte diese Visualisierungsform im Moderator's Interface, dass Knoten, die mit Kanten verbunden sind, zueinander hingezogen und Knoten, die nicht miteinander verbunden sind, voneinander weggestoßen werden. Je weiter zwei miteinander verbundene Knoten voneinander entfernt sind, desto größer wird die Kraft der Sprungfeder, die sie zueinander zieht.

Da diese Visualisierungsform gut für den Benutzer nachvollziehbar ist, wurde sie für die Darstellung der Clustering-Ergebnisse angepasst. Der Unterschied zu der ursprünglichen Variante besteht darin, dass nicht mehr die Kanten, sondern die Cluster-Zugehörigkeiten entscheidend für die Berechnung der Kräfte ist. Knoten desselben Clusters ziehen einander an, während sich Knoten aus unterschiedlichen Clustern abstoßen.

4.3 Bewertungskriterien

Die Norm ISO 9241 mit dem deutschen Titel *Ergonomie der Mensch-System-Interaktion* beschäftigt sich mit dem ergonomischen Aufbau eines Büroarbeitsplatzes unter Benutzung eines Bildschirms. In dieser Arbeit wird ein Teil dieser Norm betrachtet: Die deutsche Fassung DIN EN ISO 9241-110 mit dem Titel *Grundsätze der Dialoggestaltung* beinhaltet, wie eine sinnvolle Schnittstelle für einen Benutzer zu einem Software-System beschaffen sein sollte. In der Norm werden einige Ziele formuliert, die eine Benutzerschnittstelle erfüllen sollte. In dieser Arbeit wird die Norm dazu verwendet, um Interaktionstechniken mit den dort vorgestellten Kriterien zu bewerten. [din06]

In der DIN EN ISO 9241-110 Norm wird häufig der Begriff *Dialog* verwendet. Damit ist eine Folge von Benutzereingaben und Ausgaben eines interaktiven Systems gemeint, die notwendig sind, um ein bestimmtes Ziel zu erreichen [din06, S.6]. DIN EN ISO 9241-110 definiert sieben allgemeine Ziele, die ein Dialog zwischen Benutzer und Anwendung erfüllen soll [din06, S.7]:

1. Aufgabenangemessenheit
2. Selbstbeschreibungsfähigkeit
3. Erwartungskonformität
4. Lernförderlichkeit
5. Steuerbarkeit
6. Fehlertoleranz
7. Individualisierbarkeit

Die Reihenfolge der Ziele hat keine Bedeutung; sie sind als gleichwertig anzusehen. Im folgenden werden die Ziele im Detail und mit Hinblick auf das gegebene Anwendungsszenario betrachtet.

4.3.1 Aufgabenangemessenheit

Bei der Aufgabenangemessenheit geht es darum, zu prüfen, ob eine Benutzerschnittstelle geeignete Funktionalitäten bietet, um die Zielaufgabe ausführen zu können. Zudem wird die Einfachheit betrachtet, also ob der Programmablauf simpel und funktional oder zu kompliziert und mit zu vielen überflüssigen Aktionen und Informationen versehen ist. [din06, S.8]

In dieser Arbeit wird das Kriterium dazu verwendet, um zu überprüfen, wie einfach beziehungsweise kompliziert eine Interaktionstechnik für den Benutzer auszuführen ist und ob sie den gewünschten Erfolg bringt. Da es sich bei der Zielgruppe des Moderator's Interface um Pädagogen handelt, kann keine Expertise im Umgang mit komplexen Programmen vorausgesetzt werden. Daher würden bei diesem Kriterium einfache, intuitive und effektive Techniken höher bewertet und komplexere, mit vielen technischen Informationen versehene Verfahren, die möglicherweise den gleichen Effekt haben, niedriger.

4.3.2 Selbstbeschreibungsfähigkeit

Der Begriff der Selbstbeschreibungsfähigkeit bewertet, wie verständlich eine Benutzerschnittstelle gestaltet ist. Dazu wird auch berücksichtigt, welche Hilfen der Benutzer während der Anwendung durch das Programm erhält, ohne erst in einem Benutzerhandbuch nachlesen zu müssen. [din06, S.10]

Dieser Punkt ist differenziert zu betrachten, da der Programmierer (in diesem Falle der Autor dieser Arbeit) es selbst in der Hand hat, welche Hilfen der Benutzer bei der Anwendung einer Interaktionstechnik zugesprochen bekommt. Jedoch besteht unter der Annahme, dass grundsätzlich keine Hilfe gegeben ist, die Möglichkeit, dieses Kriterium zu benutzen. Dabei wird bewertet, ob ein Benutzer, der zuvor nicht mit dieser Technik gearbeitet hat, wissen kann, was er zu tun hat oder nicht.

4.3.3 Erwartungskonformität

Erwartungskonformität macht eine Aussage darüber, ob eine Benutzerschnittstelle an die Bedürfnisse und Erwartungen der typischen Benutzergruppe des jeweiligen Systems angepasst ist und ob die Struktur und Organisation konsistent gehalten sind. [din06, S.11]

Mit diesem Kriterium wird in dieser Arbeit bewertet, wie gut eine Technik die Erwartungen der typischen Benutzergruppe (Pädagogen) erfüllt. Es ähnelt den beiden vorherigen Kriterien, unterscheidet sich jedoch davon, indem auch bewertet wird, ob die Möglichkeit besteht, die Interaktionstechnik an die Bedürfnisse der Pädagogen anzupassen. Konsistenz hingegen wird in dieser Bewertung nicht berücksichtigt, da jede Technik für sich betrachtet wird und kein Vergleich zwischen ihnen stattfindet. Konsistenz sollte jedoch, abgesehen von dieser Evaluation, im Hinblick auf spätere Anwendungen gegeben sein, um die Benutzbarkeit zu verbessern.

4.3.4 Lernförderlichkeit

Eine Benutzerschnittstelle ist lernförderlich, wenn sie den Benutzer bei dem Erlernen der Benutzung unterstützt. Dies sollte mit einem Hilfe-System und konstruktiven Hinweisen geschehen. [din06, S.12]

Auch dieses Kriterium liegt in der Regel zu einem großen Teil in der Verantwortung des Programmierers, da dieser bestimmen kann wie stark der Benutzer bei dem Erlernen der Benutzung unterstützt wird. Auf diese Arbeit bezogen wird es so interpretiert, dass eine Bewertung vorgenommen wird, wie einfach, ohne weitere Hilfen, das Erlernen einer Interaktionstechnik für einen Pädagogen sein kann.

4.3.5 Steuerbarkeit

Die Steuerbarkeit bestimmt, wie gut ein Benutzer die Interaktion mit einer Benutzerschnittstelle steuern kann. Dazu zählen beispielsweise die Richtung und Geschwindigkeit der Interaktion. [din06, S.13]

Dieser Punkt kann nahezu vollständig aus der Norm übernommen werden. Dieser Aspekt soll bewerten, ob der Benutzer mit der jeweiligen Interaktionstechnik den Ablauf gut steuern kann, beziehungsweise, ob er die Kontrolle über die Anwendung besitzt oder ob der restliche Programmablauf dazu führt, dass die Steuerung mit dieser Interaktionstechnik nicht zufriedenstellend funktioniert.

4.3.6 Fehlertoleranz

Fehlertoleranz bewertet, ob ein Benutzer trotz fehlerhafter Eingaben des Benutzers immer noch sein Ziel erreichen kann. Daher sollen zum einen fehlerhafte Eingaben vermieden werden, was beispielsweise dadurch realisiert wird, dass der Benutzer vor Abschluss seiner Aktion darauf hingewiesen, dass seine Eingabe nicht korrekt ist. Zum anderen soll verhindert werden, dass eine fehlerhafte Eingabe im Nachhinein eine schwerwiegende Unterbrechung des Programmablaufes zur Folge hat. Stattdessen soll dem Benutzer mit konstruktiven Vorschlägen dabei geholfen werden, Fehler zu vermeiden und / oder zu korrigieren. [din06, S.14]

Auf diese Arbeit bezogen bedeutet dies, dass bewertet wird, welche Interaktionstechnik robuster und welche anfälliger für Fehler ist. Natürlich sollen die Techniken grundsätzlich so gestaltet und implementiert werden, dass Fehler vermieden werden, jedoch sind manche Techniken dafür besser geeignet als andere. Mit Fehlern sind schließlich nicht nur Programmabstürze gemeint, sondern auch Aktionen, die nicht im Sinne des Benutzers waren und ihn dabei behindern, sein Ziel, die sinnvolle Gruppierung der Diskussionsbeiträge, zu erreichen.

4.3.7 Individualisierbarkeit

Individualisierbarkeit bedeutet, dass der Benutzer die Interaktion mit dem System und die Visualisierung von Informationen seinen Bedürfnissen anpassen kann. [din06, S.15]

Dieses Kriterium wird bei der Bewertung der Interaktionstechniken für das Moderator's Interface nicht berücksichtigt. Es wird eine feststehende Anzahl von Interaktionstechniken geben, die nicht weiter angepasst werden. Zudem ist die Visualisierung der Ergebnisse kein Hauptziel dieser Arbeit.

Name	AA	SBF	EK	LF	ST	FT	Summe
1 Beitrag verschieben							
x Beiträge verschieben							
x Beiträge markieren							
Manuelle Modifikation von w							
Schlüsselparameter							
Wichtigste Parameter							
History mit UNDO-Funktion							
History mit Random Access UNDO							

Tabelle 3: Übersicht der Interaktionstechniken. AA = Aufgabenangemessenheit. SBF = Selbstbeschreibungsfähigkeit. EK = Erwartungskonformität. LF = Lernförderlichkeit. ST = Steuerbarkeit. FT = Fehlertoleranz.

4.4 Evaluation der Kriterien

In den folgenden Abschnitten werden die zuvor beschriebenen Bewertungskriterien auf acht verschiedene Interaktionstechniken angewendet. Bei den ersten drei Interaktionstechniken kann der Benutzer Diskussionsbeiträge verschieben oder markieren. Bei den Techniken vier, fünf und sechs wird der Gewichtungsvektor eher explizit angegeben. Bei den letzten beiden Interaktionstechniken erhält der Benutzer, aufbauend auf Technik eins, eine UNDO-Funktion.

Die Bewertung der Kriterien erfolgt so, dass jeweils ein Wert zwischen 0 und 5 zugewiesen wird, wobei 0 für das jeweilige Kriterium *sehr negativ* und 5 *sehr positiv* bedeutet. Eine 5 für das Kriterium *Lernförderlichkeit* bedeutet demnach, dass die Benutzung der Interaktionstechnik sehr leicht zu erlernen ist, während eine 0 bei *Selbstbeschreibungsfähigkeit* auf eine extrem komplizierte Benutzung hindeutet. Alle Kriterien werden als gleichwertig betrachtet, da jedes einzelne Kriterium einen wichtigen Aspekt der Interaktionstechniken bewertet und an dieser Stelle keines davon besonders hervorgehoben werden soll. Im letzten Abschnitt dieses Kapitels wird die Summe über alle Kriterien für jede Interaktionstechnik gebildet. Drei Techniken mit der höchsten Bewertung werden ausgewählt und implementiert.

Die Bewertungen der einzelnen Kriterien sind keine Fakten oder Ergebnisse, sondern Hypothesen, die ich aufstelle, um die Güte der Interaktionstechniken vorab abschätzen zu können.

4.4.1 Verschieben eines einzelnen Diskussionsbeitrages

Bei der ersten und einfachsten Interaktionstechnik hat der Benutzer die Möglichkeit, einen Diskussionsbeitrag mit der linken Maustaste zu *greifen* und an einer beliebigen Stelle der Interaktionsebene wieder *abzulegen*. Danach wird die Umgebung des verschobenen Beitrages nach anderen Beiträgen abgesucht. Alle gefundenen Beiträge (einschließlich des verschobenen) werden auf Ähnlichkeiten untersucht. Parameter mit ähnlichen Werten werden danach im Gewichtungsvektor erhöht, während unähnliche verkleinert werden.

Abbildung 18 zeigt ein Beispiel für diese Interaktionstechnik. Der rote Knoten mit dem Inhalt B wird zu dem grünen Knoten mit dem Inhalt B verschoben. Daraus resultierend

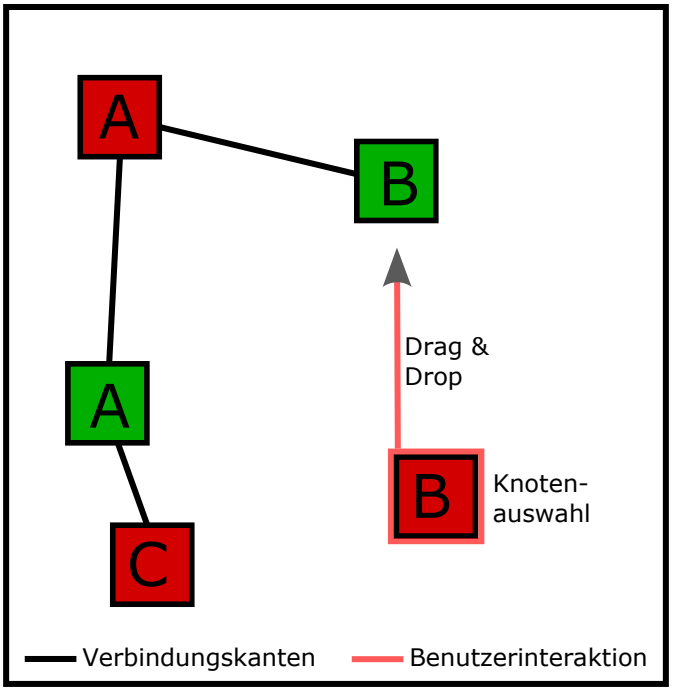


Abbildung 18: Verschieben eines einzelnen Diskussionsbeitrages.

wird die inhaltliche Ähnlichkeit der beiden Knoten erkannt und die restlichen Knoten hauptsächlich nach dem Inhalt gruppiert.

Aufgabenangemessenheit: 2
Die Benutzung dieser Technik ist sehr einfach, da der Benutzer immer nur einen einzelnen Diskussionsbeitrag mit der linken Maustaste *greifen* und dann verschieben muss. Der Effekt auf den Gewichtungsvektor ist jedoch fragwürdig; dies soll am folgenden Beispiel verdeutlicht werden: Wenn der Benutzer zwei Beiträge aus dem Grund zusammenzieht, weil der Inhalt gleich ist und zufällig auch Autor und Knotentyp dieser Beiträge übereinstimmen, dann werden neben *Inhalt* auch die Parameter *Autor* und *Knotentyp* höher bewertet, was in diesem Fall nicht in der Absicht des Benutzers lag.

Selbstbeschreibungsfähigkeit: 5
Die Selbstbeschreibungsfähigkeit dieser Technik ist sehr hoch. Auch ohne Benutzerhandbuch und Hilfe im Programm selbst sollte dem Benutzer klar sein, was zu tun ist. Dies liegt vor allem daran, da das Verschieben von Diskussionsbeiträgen ohnehin schon als Feature des Moderator's Interface vorhanden und auch in Tools zur graphbasierten Diskussion üblich ist.

Erwartungskonformität: 2
Die Erwartungskonformität dieser Technik ist aus folgenden Gründen niedrig: Das Verschieben von Beiträgen ist, wie bereits bei *Selbstbeschreibungsfähigkeit* erläutert, üblich bei Werkzeugen wie dem Moderator's Interface und anderen Programmen zur graphbasierten Diskussion. Die Technik an sich ist demnach schon stark in den Anwendungsbereich der graphbasierten Diskussionen integriert und muss nicht weiter angepasst werden. Der Effekt der Interaktion kann jedoch für den Benutzer verwirrend sein, da es ungewohnt sein

kann, dass das Verschieben eines Elementes die Visualisierung aller anderen Elemente beeinflusst.

Lernförderlichkeit: 5
Aufgrund der Einfachheit dieser Interaktionstechnik ist die Lernförderlichkeit sehr hoch. Wie die Technik zu benutzen ist, sollte dem Benutzer auch ohne jegliche Hilfen[11] klar sein.

Steuerbarkeit: 3
Die Steuerbarkeit ist, auch bei einer dynamischen Visualisierung der Clustering-Ergebnisse, gut. Vor dem ersten Verschieben eines Diskussionsbeitrages ist der Diskussionsgraph bewegungslos, weshalb an dieser Stelle keine Probleme entstehen können. Nach dem Verschieben eines Diskussionsbeitrages verändert sich dies jedoch und die Knoten bewegen sich auf der Ebene und bilden Gruppen. Diese Visualisierung in Kombination mit der Interaktionstechnik kann sich für manche Benutzer so auswirken, dass die Knoten sich zu schnell bewegen, als dass sie diese einfach greifen und weiter verschieben können.

Fehlertoleranz: 3
Die Fehlertoleranz dieser Interaktionstechnik ist hoch. Selbst wenn der Benutzer unabsichtlich einen Diskussionsbeitrag an die falsche Stelle verschiebt, kann er die gewünschte Gruppierung immer noch durch weiteres Verschieben von anderen Beiträgen erreichen. Diese Technik kann im Zusammenspiel mit der schon beschriebenen Visualisierung jedoch zu weiteren (unkritischen) Fehlern führen.

4.4.2 Verschieben mehrerer Diskussionsbeiträge

Um mehrere Beiträge auszuwählen, kann der Benutzer im Moderator's Interface einen Rahmen um die Beiträge ziehen, die er auswählen möchte. Danach kann er diese mit einer Mausbewegung auf einmal verschieben. Wie schon bei der Technik zum Verschieben eines einzelnen Diskussionsbeitrages wird danach die Umgebung der verschobenen Beiträge nach anderen Beiträgen abgesucht. Alle gefundenen Beiträge (einschließlich der verschobenen) werden auf Ähnlichkeiten untersucht. Parameter mit ähnlichen Werten werden danach im Gewichtungsvektor erhöht, während unähnliche verkleinert werden.

Abbildung 19 zeigt ein Beispiel für diese Interaktionstechnik. Der rote Knoten mit dem Inhalt B und der rote Knoten mit dem Inhalt C wurden vom Benutzer ausgewählt und werden gleichzeitig um die gleiche Strecke nach oben verschoben. Die Diskussionsbeiträge, die örtlich nah zueinander platziert wurden, haben die gleiche Farbe. Nach diesem Kriterium werden auch die restlichen Beiträge gruppiert.

Aufgabenangemessenheit: 2
Die Benutzung dieser Technik ist nur minimal schwieriger als die Technik aus Abschnitt 4.4.1 (Verschieben einzelner Diskussionsbeiträge), da lediglich ein Rahmen um die jeweiligen Beiträge gezogen werden muss, bevor sie verschoben werden können. Was die Effektivität angeht, bestehen ähnliche Probleme wie bei der vorangegangenen Technik. Je mehr Beiträge ausgewählt und verschoben werden, desto größer ist die Fläche, die nach dem Verschieben nach weiteren Beiträgen abgesucht wird, was wiederum bedeutet, dass in der Regel mehr Beiträge gefunden werden. Je mehr Beiträge analysiert werden, desto schwieriger wird es für den Algorithmus, aussagekräftige Ähnlichkeiten zwischen diesen

[11] Siehe auch Selbstbeschreibungsfähigkeit und Erwartungskonformität dieser Interaktionstechnik.

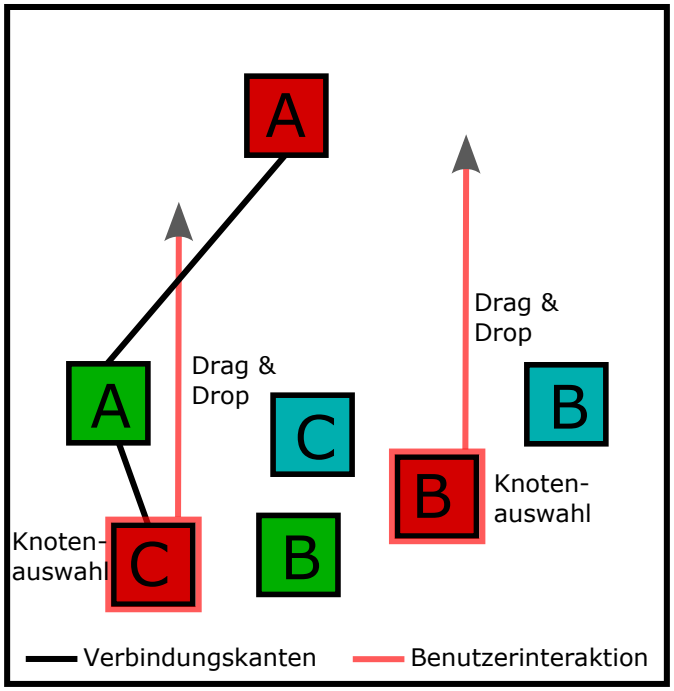


Abbildung 19: Verschieben mehrerer Diskussionsbeiträge.

festzustellen, da es bei steigender Anzahl schwieriger wird, Muster innerhalb der betrachteten Beiträge zu erkennen.

Selbstbeschreibungsfähigkeit: 5
Die Selbstbeschreibungsfähigkeit ist sehr hoch. Wie schon bei der letzten Technik ist das Verschieben von Diskussionsbeiträgen auf einer Ebene eine intuitive und etablierte Aktion. Hinzu kommt das Ziehen eines Rahmens um eine Auswahl von Beiträgen. Auch dies ist eine übliche Aktion in weit verbreiteter Standard-Software. In vielen Programmen hat der Benutzer die Möglichkeit, um eine Menge von Objekten einen Rahmen zu ziehen, um diese auszuwählen, weshalb dieses Feature von vielen Benutzern als intuitiv beziehungsweise fast schon selbstverständlich angesehen werden wird.

Erwartungskonformität: 2
Die Erwartungskonformität ist vergleichbar zu der Technik aus Abschnitt 4.4.1. Die Auswahl und das Verschieben mehrerer Beiträge durch eine Maus-Aktion ist für den Benutzer einfach, jedoch ist es fragwürdig, ob die Bedeutung der Aktion für den Benutzer nachvollziehbar ist.

Lernförderlichkeit: 5
Die Lernförderlichkeit ist aufgrund der bereits sehr einfachen und intuitiven Handhabung der Interaktionstechnik sehr hoch. Das Verschieben von Beiträgen und die Auswahl mehrerer Beiträge, indem ein Rahmen um diese gezogen wird, ist so einfach gehalten, dass sie für den Benutzer sehr einfach zu erlernen sein sollten.

Steuerbarkeit: 1
Die Steuerbarkeit beinhaltet die selben Vor- und Nachteile der Interaktionstechnik von Ab-

schnitt 4.4.1. Bei dieser Technik wird sie noch schwieriger, da der Benutzer um eine sich bewegende Menge von Beiträgen einen Rahmen ziehen muss. Danach muss er einen der selektierten Beiträge mit der Maus *greifen*, um sie zu verschieben. Ein weiteres Problem kann sich durch die Bewegung der Beiträge ergeben, da der Benutzer dadurch eventuell nicht mehr in der Lage ist, genau die Beiträge auszuwählen, die er auswählen will. Es kann beispielsweise ein Beitrag in den Auswahlbereich geraten, den der Benutzer nicht verschieben will. In diesem Fall muss er mindestens zwei Gruppen von Beiträgen markieren und diese separat verschieben, wobei nach dem Verschieben der ersten Gruppe unklar ist, ob er die zweite Gruppe in Ruhe an ihre Zielposition verschieben kann.

Fehlertoleranz: 0
Die Fehlertoleranz ist deutlich niedriger als bei dem Verschieben einzelner Diskussionsbeiträge. Wie schon bei *Steuerbarkeit* beschrieben, kann es leicht zu Situationen kommen, die der Benutzer nicht mehr kontrollieren kann und in denen Fehler provoziert werden. Versucht der Benutzer beispielsweise, eine Menge von Beiträgen auszuwählen, während ein anderer Beitrag e in den Auswahlbereich gerät. Bemerkt der Benutzer e, so muss er den Auswahlbereich verkleinern und mehrere Gruppen von Beiträgen verschieben. Dabei können ihm wiederum Fehler unterlaufen, da sich alle Beiträge ständig weiter bewegen. Er kann die Beiträge, die er noch verschieben will, aus den Augen verlieren oder die falschen verschieben. Bemerkt der Benutzer e nicht, so verschiebt er einen Beitrag, den er gar nicht verschieben wollte. Diese (durch die Interaktionstechnik kombiniert mit der Visualisierungsform begünstigten) Fehlverhalten können einen falschen Gewichtungsvektor und somit falsche Cluster-Ergebnisse zur Folge haben.

4.4.3 Markierung ähnlicher Diskussionsbeiträge

Zusätzlich zum bereits beschriebenen *Rahmen um Diskussionsbeiträge ziehen* gibt es im Moderator's Interface die Möglichkeit, durch das Halten der SHIFT-Taste und die wiederholte Verwendung der linken Maustaste mehrere Beiträge auszuwählen. Danach wird die Auswahl mit einem Mausklick auf einen Knopf bestätigt und das Clustering durchgeführt. Die Idee dieser Interaktionstechnik ist es also, mehrere möglicherweise räumlich voneinander getrennte Beiträge auszuwählen und diese als Referenz-Cluster vorzugeben. Danach kann der Clustering-Algorithmus diese Beiträge auf Ähnlichkeiten untersuchen und entsprechend ähnliche Parameter im Gewichtungsvektor erhöhen und unähnliche abschwächen.

Nachdem eine erste Gruppierung berechnet wurde, kann der Benutzer diese modifizieren. Durch erneutes markieren von einem oder mehreren Diskussionsbeiträgen und einem Rechtsklick kann er ein Kontextmenü öffnen. Dort hat er die Möglichkeit, durch das Auswählen eines Menüeintrages die markierten Beiträge aus ihren bisherigen Gruppen zu entfernen und aus ihnen eine neue Gruppe zu erstellen. Damit kann der Benutzer beispielsweise einzelne Beiträge aus Clustern entfernen oder zwei Cluster zu einem neuen Cluster verbinden. Auch bei diesem Vorgang muss die Interaktion interpretiert und auf den Gewichtungsvektor angewendet werden.

Abbildung 20 zeigt ein Beispiel für diese Interaktionstechnik. Der rote Knoten mit dem Inhalt B und der blaue Knoten mit dem Inhalt B wurden vom Benutzer ausgewählt. Danach wird das Clustering gestartet und die markierten Knoten als Vorgabe-Cluster angesehen. Da sie als Gemeinsamkeit den Inhalt haben, werden die restlichen Knoten nach dem Inhalt gruppiert.

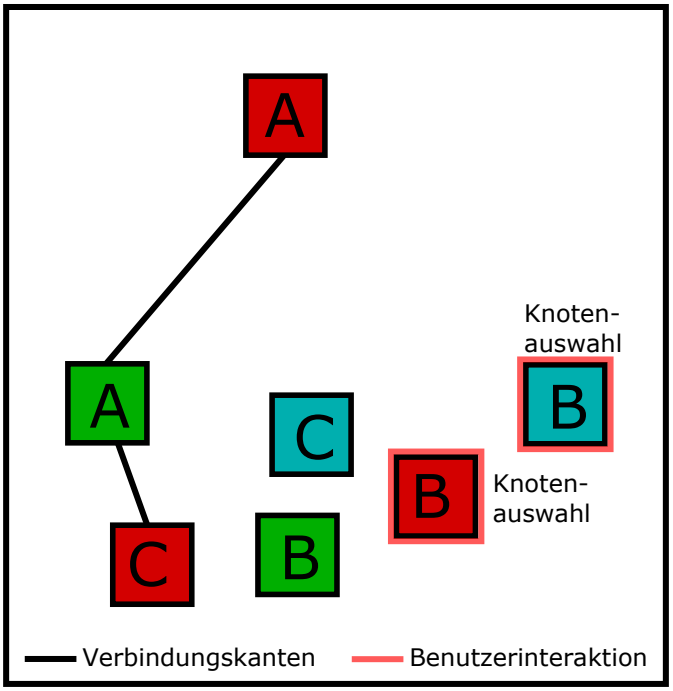


Abbildung 20: Markierung ähnlicher Diskussionsbeiträge.

Aufgabenangemessenheit: 4
Die Benutzung dieser Interaktionstechnik ist einfach. Der Benutzer hält die SHIFT-Taste (bei anderen Programmen, die diese Funktion unterstützen, in der Regel die STRG-Taste) gedrückt und wählt mit der linken Maustaste die Diskussionsbeiträge aus, die er in einem Cluster haben möchte. Hat er alle Beiträge ausgewählt, die in seinem *Referenzcluster* liegen, so drückt er einen Knopf, um die ausgewählten Beiträge analysieren und die restlichen Beiträge gruppieren zu lassen. Die Effektivität ist besser als die der letzten Interaktionstechniken, da die Auswahl der Beiträge präziser erfolgt und somit der Gewichtungsvektor besser angepasst werden kann.

Selbstbeschreibungsfähigkeit: 4
Die Selbstbeschreibungsfähigkeit dieser Technik ist hoch. In vielen Standard-Programmen gibt es die Möglichkeit, durch das Halten einer Taste (beispielsweise SHIFT oder STRG) mehrere Objekte auszuwählen. Falls der Benutzer also weiß, dass er diese Aktionsmöglichkeit hat, so ist die Benutzung als einfach und intuitiv anzusehen. Es muss jedoch unmissverständlich klargemacht werden, warum der Benutzer diese Beiträge markieren soll, damit er nicht den Eindruck bekommt, er würde sie markieren, um sie, wie in den ersten beiden Interaktionstechniken, zu verschieben.

Erwartungskonformität: 3
Die Erwartungskonformität ist mittel. Der Benutzer benötigt das Vorwissen, dass er Diskussionsbeiträge mit der linken Maustauste und der SHIFT-Taste markieren kann. Ist dies der Fall, sollte der Knopf zum Start des Clusterings so angepasst sein, dass er eine Aussage darüber macht, was mit den markierten Beiträgen passiert. Dies kann beispielsweise durch eine Beschriftung wie *Markierte Beiträge als Vorlage verwenden* verdeutlicht werden.

Allerdings kann es passieren, dass die Markierung des Benutzers darin resultiert, dass die restlichen Beiträge nach Kriterien gruppiert werden, die der Benutzer bei seiner Selektion nicht beachtet hat.

Lernförderlichkeit: 5
Die Lernförderlichkeit dieser Interaktionstechnik ist sehr hoch. Falls dem Benutzer bekannt ist, dass mit der SHIFT-Taste mehrere Beiträge markiert werden und welche Wirkung der Knopf, der das Clustering startet, hat, sind bereits alle Funktionen bekannt. Falls nicht, sind sie nach wenigen Anwendungen leicht zu erlernen.

Steuerbarkeit: 5
Die Steuerbarkeit dieser Technik wird als sehr gut eingeschätzt. Der Benutzer hat bei dem ersten Versuch soviel Zeit wie er möchte, um die Beiträge des Referenz-Clusters zu bestimmen. Ist er mit der folgenden Gruppierung unzufrieden, kann er erneut eine Menge von Beiträgen definieren, indem er diese wieder durch das Halten der SHIFT-Taste und der linken Maustaste auswählt. Da die Position der Beiträge keine Rolle spielt, kann er sich auch hier Zeit lassen und muss nicht so schnell wie möglich eine Auswahl treffen. Falls er ungewollt einen Beitrag auswählt, kann er dies durch einen Klick darauf wieder rückgängig machen. Durch loslassen der SHIFT-Taste und einem Klick *ins Leere* kann die gesamte bisherige Auswahl aufgehoben werden.

Fehlertoleranz: 4
Die Fehlertoleranz dieser Interaktionstechnik ist hoch. Der Benutzer kann den Fehler machen, nur einen Diskussionsbeitrag auszuwählen und dann den Start-Knopf zu drücken, was bei dieser Interaktionstechnik keinen Sinn machen würde. Dieser Fall muss also in der Programmlogik abgedeckt sein. Zudem kann der Benutzer ungewollt einen Beitrag auswählen, was er jedoch durch einen erneuten Klick wieder rückgängig machen kann.

4.4.4 Manuelle Modifikation des Gewichtungsvektors

Bei dieser Interaktionstechnik geht es darum, den Gewichtungsvektor w manuell anzupassen. Dies könnte so umgesetzt werden, indem Zahlenwerte in ein Feld eintragen werden; da jedoch ein fester Wertebereich (0.0 bis 1.0) gegeben ist, ist es sinnvoll, für jeden Parameter einen Schieberegler zu verwenden, der genau diesen Bereich abdeckt. Neben jedem Schieberegler steht eine kurze, treffende Erklärung des jeweiligen Parameters. Nachdem die Schieberegler an ihre Zielposition verschoben wurden, muss per Mausklick ein Knopf aktiviert werden, um das Clustering zu starten. Bei dieser Technik ist keine Interpretation der Benutzeraktion notwendig, da direkt auf den Daten des Gewichtungsvektors gearbeitet wird.

Abbildung 21 zeigt ein Beispiel für diese Interaktionstechnik. Der Benutzer kann die Clustering Parameter (in diesem Beispiel *Inhalt* und *Farbe*) mit den Schiebereglern links verändern. Wenn er den Start-Knopf betätigt, werden die Diskussionsbeiträge entsprechend der Vorgabe gruppiert.

Aufgabenangemessenheit: 3
Die Benutzung dieser Technik ist sehr einfach. Schieberegler sind bereits seit langer Zeit bekannte Steuerelemente in grafischen Oberflächen und als selbsterklärend anzusehen. Ob sie jedoch Ergebnisse liefert, die dem Benutzer Nutzen verschaffen, ist fragwürdig. Die direkte Manipulation des Gewichtungsvektors ist insofern abstrakt, da der Benutzer bei

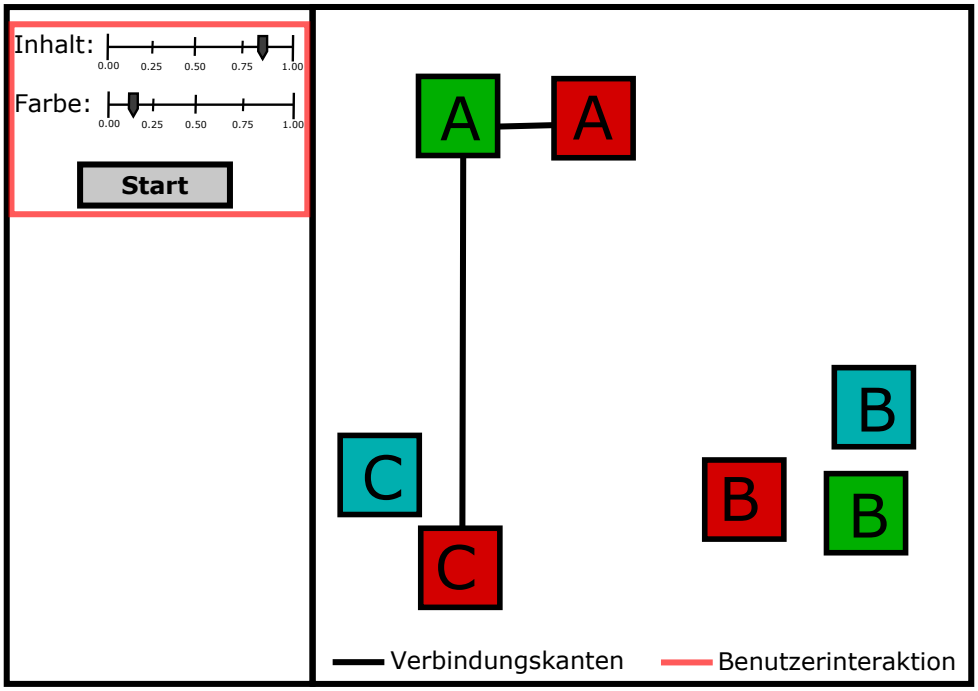


Abbildung 21: Manuelle Modifikation des Gewichtungsvektors.

größeren Diskussionsgraphen kaum vorhersagen kann, welche Einstellung der Schieberegler welches Ergebnis liefern wird. Er wird also zwangsläufig mehrere Versuche unternehmen müssen, um sich einem sinnvollen Resultat anzunähern.

Selbstbeschreibungsfähigkeit: 5
Die Selbstbeschreibungsfähigkeit dieser Technik ist sehr hoch. Grundsätzlich sind Schieberegler in der heutigen Zeit ein viel verwendetes GUI-Steuerelement und die Anwendung wird von vielen Benutzern als intuitiv angesehen.

Erwartungskonformität: 4
Schieberegler werden, von vielen Benutzern als einfach und intuitiv angesehen. Die Bedeutung der Regler kann jedoch verwirrend für einen Benutzer sein. Es gibt einen festen Wertebereich für jeden Regler, der durch Zahlen an der linken und rechten Seite gekennzeichnet ist, und der Benutzer hat die Möglichkeit einen Wert aus diesem Bereich zu setzen. Dem Benutzer werden jedoch die genauen Zahlen in der Regel nicht wichtig sein und auch kaum zum Verständnis beitragen. Wenn er den Schieberegler auf die Position von 0.75 oder 1.00 bewegt, so wird er erwarten, dass der jeweilige Parameter als *wichtig* angesehen wird. Um den Benutzer also nicht durch die internen Werte des Gewichtungsvektors zu verwirren, sollten anstatt der Zahlen 0.00 und 1.00 Worte wie *unwichtig* und *sehr wichtig* verwendet werden.

Lernförderlichkeit: 4
Da die Steuerelemente dieser Technik einfach sind, wird die Lernförderlichkeit als sehr hoch eingeschätzt. Der Benutzer muss sich jedoch mit den Einstellungen der Schieberegler vertraut machen und beobachten, welche Einstellung welche Effekte auf das Clustering-Ergebnis hat. Da es gegebenenfalls für den Benutzer unverständlich sein kann, dass be-

stimmte Reglereinstellungen bestimmte, eventuell ungewollte Effekte auf das Ergebnis haben.

Steuerbarkeit: 5
Die Steuerbarkeit dieser Interaktionstechnik wird als sehr gut eingeschätzt. Die Elemente, mit denen der Benutzer interagiert (die Schieberegler), werden niemals vom Programm verändert. Er behält während des gesamten Ablaufes die Kontrolle über die Anwendung. Sollte ein Ergebnis nicht seinen Erwartungen entsprechen, so kann er die Einstellung der Schieberegler verändern und ein erneutes Clustering durchführen.

Fehlertoleranz: 5
Die einzige Möglichkeit, wie der Benutzer einen kritischen Fehler im Programm hervorrufen könnte, besteht darin, dass er alle Schieberegler auf den Wert 0.00 setzt. Das Clustering würde in diesem Fall keinen Parameter berücksichtigen, daher muss dieser Fall in der Programmlogik abgefangen werden. Unkritische Fehler (falsche Einstellung der Schieberegler) des Benutzers führen dazu, dass das Clustering-Ergebnis nicht seinen Anforderungen entspricht. In diesem Fall muss er die Regler neu positionieren und ein weiteres Clustering vornehmen.

4.4.5 Angabe eines Schlüsselparameters

Der Benutzer hat die Möglichkeit, genau einen Parameter anzugeben, nach dem das Clustering durchgeführt werden soll. Die Auswahl wird durch eine Drop-Down-Liste getroffen. Danach muss er per Mausklick das Clustering starten. Auch bei dieser Interaktionstechnik, muss keine Aktion des Benutzers interpretiert werden. Stattdessen wird der Parameterwert des ausgewählten Parameters auf den Höchstwert 1.00 gesetzt, während alle anderen Parameterwerte auf den Wert 0.00 gesetzt werden.

Abbildung 22 zeigt ein Beispiel für diese Interaktionstechnik. Der Benutzer kann in der Drop-Down-Liste einen der Parameter auswählen (in diesem Beispiel *Inhalt*). Wenn er den Start-Knopf betätigt, werden die Diskussionsbeiträge nach diesem Parameter gruppiert.

Aufgabenangemessenheit: 2
Diese Interaktionstechnik spezialisiert die Technik aus Abschnitt 4.4.4. Anstatt der Angabe von kontinuierlichen Werten für jeden beliebigen Parameter, hat der Benutzer die Möglichkeit einen Parameter mit 1.00 (*sehr wichtig*) und alle anderen mit 0.00 (*unwichtig*) zu bewerten. Die Auswahl erfolgt über ein übliches GUI-Element wie eine Drop-Down-Liste. Dies macht die Benutzung einfach, schränkt jedoch die Mächtigkeit und Effektivität der Technik ein, da der Benutzer keine Clustering-Parameter miteinander kombinieren kann. So kann es vorkommen, dass der Benutzer sein Ziel niemals erreichen kann, wenn das Clustering-Ergebnis die Einbeziehung von mehreren Clustering-Parametern erfordert.

Selbstbeschreibungsfähigkeit: 5
Die Selbstbeschreibungsfähigkeit dieser Technik ist sehr hoch. Die Funktionsweise einer Drop-Down-Liste sollte jedem Benutzer bekannt und die Auswahl eines Parameters demnach intuitiv sein. Dies ist neben dem Start-Knopf das einzige Steuerelement dieser Technik.

Erwartungskonformität: 4
Die Erwartungskonformität dieser Technik ist hoch. Die Benutzung einer Drop-Down-Liste

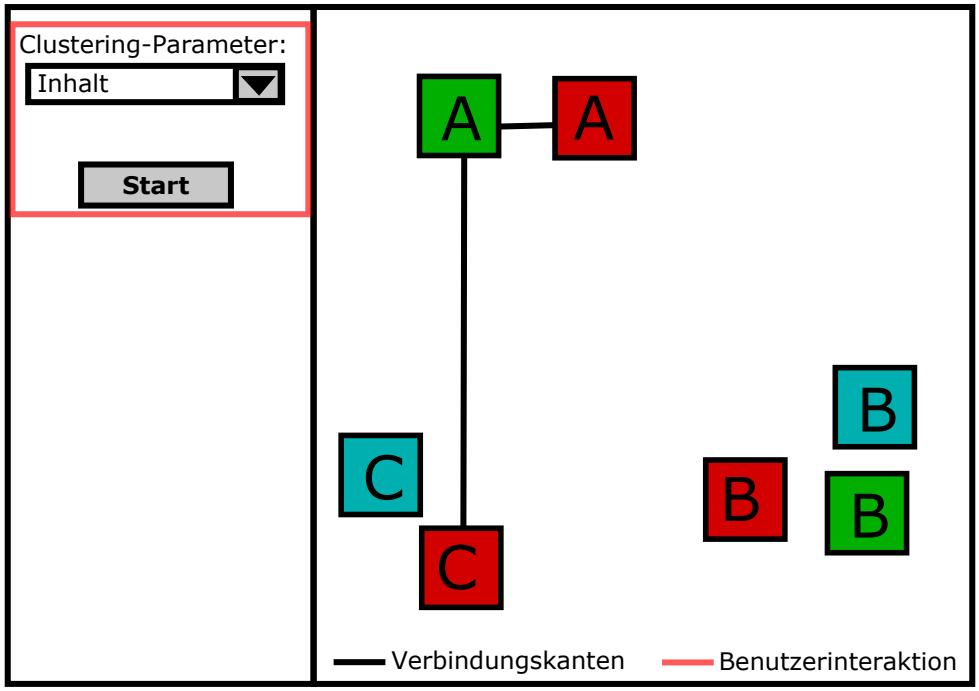


Abbildung 22: Angabe eines Schlüsselparameters.

ist in vielen Programmen mit grafischer Benutzeroberfläche üblich und die Anwendung wird von vielen Benutzern als intuitiv und selbsterklärend angesehen. Auch der Effekt der Auswahl eines Parameters sollte durch treffende Bezeichnungen der Parameter leicht zu verstehen sein. Eine weitere Anpassung der Benutzerschnittstelle muss daher nicht erfolgen.

Lernförderlichkeit: 5
Die Lernförderlichkeit ist aufgrund der Einfachheit dieser Technik sehr hoch. Der Benutzer kann nur einen einzigen Wert manipulieren und der Effekt auf das Clustering-Ergebnis sollte daher deutlicher ausfallen als bei der letzten betrachteten Technik. Dort konnte es gegebenenfalls unverständlich sein, warum eine geringfügige Veränderung eines Schiebereglers das Clustering-Ergebnis stark beeinflusst, während bei dieser Technik bei jeder Interaktion die Clustering-Parameter auf $0.00/1.00$ gesetzt werden. Der Benutzer würde demnach erwarten, dass sich das Ergebnis bei unterschiedlichen Parametern deutlich voneinander unterscheidet.

Steuerbarkeit: 4
Die Steuerbarkeit dieser Interaktionstechnik ist sehr hoch. Die Steuerelemente, die der Benutzer bedient, werden nie vom Programm manipuliert. Der Benutzer hat somit zwar eine sehr beschränkte Auswahl von Interaktionsmöglichkeiten[12], jedoch behält er im Rahmen dieser Interaktionstechnik immer die Kontrolle über den Programmablauf. Entspricht das Clustering nach einem Parameter nicht seinen Vorstellungen, so kann er ein Clustering nach einem anderen Parameter durchführen. Ein Nachteil besteht darin, dass immer nur nach genau einem Parameter gruppiert werden kann.

[12] Auswahl eines Parameters

Fehlertoleranz: 5
Der Benutzer hat bei dieser Technik keine Möglichkeit, einen Fehler zu begehen. Standardmäßig sollte in der Drop-Down-Liste ein Parameter ausgewählt sein, so dass nicht nach *keinem* Parameter gruppiert werden kann.

4.4.6 Angabe der wichtigsten Parameter

Diese Interaktionstechnik ähnelt semantisch der Technik aus Abschnitt 4.4.4. Sie gibt dem Benutzer die Möglichkeit, mehrere Parameter auszuwählen und diese mit Prioritäten zu versehen. Die Auswahl der Priorität jedes Parameters wird über eine Drop-Down-Liste getroffen, wobei der Standardwert für jeden Parameter *unwichtig* sein sollte. Auf den Gewichtungsvektor bezogen entspricht dies einem Nullvektor. Die Grundidee ist, dass der Benutzer nur an den Parameterwerten etwas verändern muss, die ihm wichtig sind; alle anderen muss er nicht modifizieren. Nach seiner Auswahl muss er das Clustering per Mausklick auf einen Knopf starten. Denkbare Werte für die Drop-Down-Liste sind zum Beispiel *sehr wichtig*, *wichtig*, *normal*, *weniger wichtig* und *unwichtig*. Diese sollten bezüglich des Gewichtungsvektors dem Wertebereich 1.00, 0.75, 0.50, 0.25 und 0.00 entsprechen.

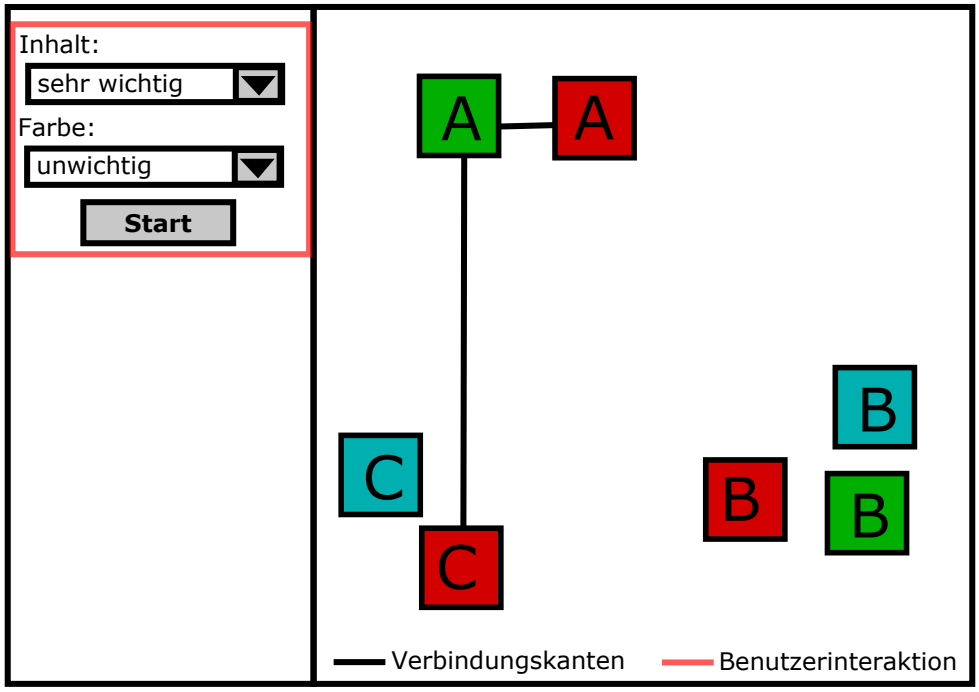


Abbildung 23: Angabe der wichtigsten Parameter.

Abbildung 23 zeigt ein Beispiel für diese Interaktionstechnik. Der Benutzer kann in der Drop-Down-Liste für jeden Parameter dessen Wichtigkeit auswählen (in diesem Beispiel wurde der *Inhalt* als sehr wichtig angegeben). Wenn er den Start-Knopf betätigt, werden die Diskussionsbeiträge nach diesen Gewichten gruppiert.

Aufgabenangemessenheit: 3
Auch diese Interaktionstechnik spezialisiert die Technik aus Abschnitt 4.4.4, jedoch wird hier kein Schieberegler verschoben, sondern ein Eintrag aus einer Drop-Down-Liste aus-

gewählt, um das Gewicht für einen Parameter zu verändern. Auch diese Technik ist eher abstrakt, da der Benutzer kaum voraussagen kann, ob eine Einstellung der Prioritäten ein zufriedenstellendes Ergebnis liefert. Daher muss er in der Regel mehrere Versuche unternehmen, um ein *gutes* Ergebnis zu erzielen.

Selbstbeschreibungsfähigkeit: 5
Die Benutzung eines Standard-GUI-Elementes wie einer Drop-Down-Liste ist für die meisten Benutzergruppen intuitiv.

Erwartungskonformität: 4
Bei dieser Interaktionstechnik müssen keine exakten Werte angegeben, sondern eine Auswahl aus vorgegebenen Prioritäten getroffen werden. Benutzergruppen wie Computerexperten würden dies eventuell als Nachteil ansehen, da sie genug Expertise besitzen, um exakte Werte einzugeben. Da es sich bei der typischen Benutzergruppe des Moderator's Interface jedoch um Pädagogen handelt, ist diese Vereinfachung eindeutig als Vorteil zu betrachten. Der Benutzer muss sich keine Gedanken darum machen, auf welchen Wert genau er einen Parameter setzen muss; stattdessen kann er einfach die Parameter auswählen, die ihm wichtig sind.

Lernförderlichkeit: 5
Die Lernförderlichkeit ist aufgrund der Einfachheit und intuitiven Benutzbarkeit der Interaktionstechnik sehr hoch. Sämtliche Steuerelemente sind selbsterklärend und die beabsichtigte Wirkung der Bewertung der Parameter offensichtlich.

Steuerbarkeit: 5
Die Steuerbarkeit dieser Interaktionstechnik ist, wie bei den beiden vorherigen Techniken, sehr gut. Die Steuerelemente, die der Benutzer bedient, werden zu keinem Zeitpunkt von der Anwendung verändert. Der Benutzer hat somit zwar eine beschränkte Auswahl von Interaktionsmöglichkeiten (Auswahl von Prioritäten), jedoch behält er im Rahmen dieser Interaktionstechnik immer die Kontrolle über den Programmablauf. Entspricht das Clustering mit den festgelegten Prioritäten nicht seinen Vorstellungen, so kann er diese verändern und ein erneutes Clustering durchführen.

Fehlertoleranz: 5
Auch bei dieser Technik besteht kaum die Möglichkeit, dass der Benutzer kritische Fehler begehen kann. Standardmäßig sollten in den Drop-Down-Listen der Parameter alle Prioritäten als *unwichtig* markiert sein, damit der Benutzer nur die Parameter verändern muss, die ihm wichtig sind. Wird das Clustering gestartet, ohne das eine Priorität verändert wurde, so entsteht ein Fehler, der abzufangen ist. Ansonsten kann der Benutzer, solange er sein Ziel-Clustering nicht erreicht hat, solange die Parameter anpassen und neu gruppieren wie er es für sinnvoll hält.

4.4.7 History mit UNDO-Funktion

Diese Technik stellt eine Erweiterung dar und soll die Interaktionstechnik aus Abschnitt 4.4.1 unterstützen. Zusätzlich gibt es eine Liste mit Benutzeraktionen (History) und einen UNDO-Knopf. Durch das Betätigen des UNDO-Knopfes hat der Benutzer die Möglichkeit, die jeweils letzte Aktion rückgängig zu machen. Dabei müssen die Werte des Gewichtungsvektors auf die alten Werte zurückgesetzt werden.

Die Bewertung der Interaktionstechnik, die für diese Technik vorausgesetzt wird, fließt in die aktuelle Bewertung mit ein.

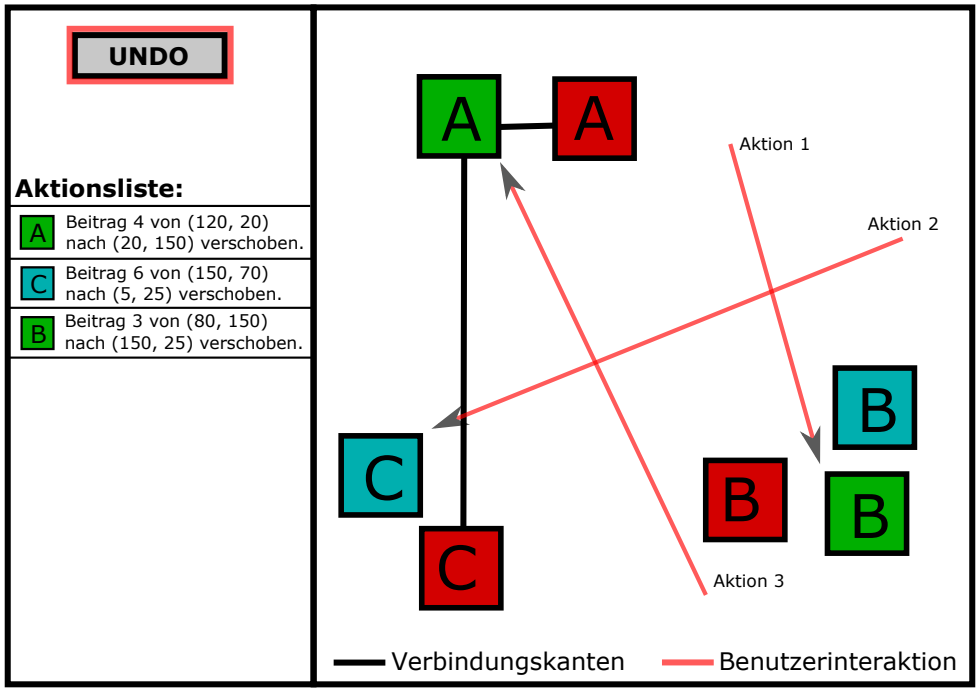


Abbildung 24: History mit UNDO-Funktion.

Abbildung 24 zeigt ein Beispiel für diese Interaktionstechnik. Auf der linken Seite sieht ist eine Liste mit drei Benutzerinteraktionen (Verschiebungen von Diskussionsbeiträgen) zu sehen. In der Liste steht, welcher Beitrag von welcher Startposition an welche Zielposition bewegt wurde. Mit der UNDO-Taste kann der Benutzer die oberste Aktion in der Liste rückgängig machen.

Aufgabenangemessenheit: 3
Durch das Hinzufügen des UNDO-Knopfes verbessert sich die Effektivität der ursprünglichen Interaktionstechnik. Der Benutzer hat die Möglichkeit, Fehler sofort zu korrigieren, indem er die letzte Aktion rückgängig macht. Die Einfachheit der Technik bleibt jedoch wie sie ist, da UNDO-Funktionen in vielen Standard-Anwendungen eingesetzt und von vielen Benutzern als selbstverständlich angesehen werden.

Selbstbeschreibungsfähigkeit: 4
Die Selbstbeschreibungsfähigkeit verringert sich durch die neue UNDO-Funktion, da durch diese die Komplexität steigt. Dennoch wird davon ausgegangen, dass der Benutzer die Funktionsweise der UNDO-Funktion auch ohne Handbuch oder Hilfen im Programm selbst versteht und diese anwenden kann.

Erwartungskonformität: 2
In der Bewertung der Technik zu der Technik aus Abschnitt 4.4.1 wurde bereits erläutert, warum die Erwartungskonformität dort so niedrig ist. Nach Hinzufügen der UNDO-Funktion bleibt dieser Wert gleich, da sich durch die neue Funktion zwar die Komplexität

der Interaktionstechnik erhöht, deren Bedeutung jedoch für den Benutzer verständlich sein sollte.

Lernförderlichkeit: 5
Sowohl das Erlernen des Verschiebens von Diskussionsbeiträgen, als auch die Verwendung der UNDO-Funktion, sind als so einfach anzusehen, dass das Erlernen der Benutzung dieser Funktionen sehr einfach sein sollte.

Steuerbarkeit: 3
Die Steuerbarkeit der ursprünglichen Technik wird durch die UNDO-Funktion geringfügig verbessert. Der Benutzer hat mehr Kontrolle über die Anwendung, da er seine Fehler sofort rückgängig machen kann und nicht mehr durch anderweitige Aktionen (weiteres Verschieben von Diskussionsbeiträgen) korrigieren muss. Dennoch ist die grundsätzliche Steuerung für die typische Benutzergruppe der Anwendung nicht als trivial anzusehen.

Fehlertoleranz: 4
Die Fehlertoleranz der Basisinteraktionstechnik wird durch die UNDO-Funktion deutlich verbessert. Falls der Benutzer einen Beitrag an die falsche Stelle verschiebt, kann er seinen Fehler durch die neue Funktion korrigieren.

4.4.8 History mit wahlfreier UNDO-Funktion

Diese Technik ist eine Verallgemeinerung der Interaktionstechnik aus Abschnitt 4.4.7. Sie erlaubt dem Benutzer nicht nur die letzte, sondern eine beliebige in der History enthaltene Aktion rückgängig zu machen. Der Benutzer muss daher zusätzlich zum Betätigen des UNDO-Knopfes die jeweilige Aktion in der Liste auswählen, damit genau diese gelöscht werden kann.

Die Bewertung der Interaktionstechnik, die für diese Technik vorausgesetzt wird, fließt in die aktuelle Bewertung mit ein.

Abbildung 25 zeigt ein Beispiel für diese Interaktionstechnik. Siehe Beschreibung der Interaktionstechnik aus Abschnitt 4.4.7. Der einzige Unterschied besteht darin, dass der Benutzer eine Aktion in der Liste anklicken kann. Nach Betätigung der UNDO-Taste wird die ausgewählte Aktion rückgängig gemacht.

Aufgabenangemessenheit: 2
Die Aufgabenangemessenheit der Technik mit der neuen Erweiterung ist aus folgenden Gründen schlechter als die der ursprünglichen Technik. Zwar hat der Benutzer durch die wahlfreie UNDO-Funktion deutlich mehr Interaktionsmöglichkeiten, die auch einfach zu bedienen sind, jedoch kaum effizient einsetzbar sind. Wenn der Benutzer die letzte Aktion rückgängig macht, kann er sich höchst wahrscheinlich noch an die Cluster-Zuordnung zu diesem Zeitpunkt erinnern. Wenn er jedoch irgendeine beliebige Aktion aus der Liste löscht, kann er nie wissen, welchen Effekt dies auf den Gewichtungsvektor und somit das Clustering-Ergebnis haben wird. Dies hat zur Folge, dass die Effizienz gleich bleibt und die Benutzbarkeit komplexer wird.

Selbstbeschreibungsfähigkeit: 2
Die Anwendung ist durch die neue Funktion komplexer geworden. Es kann zwar davon ausgegangen werden, dass ein *normaler* Benutzer mit der Verwendung einer UNDO-

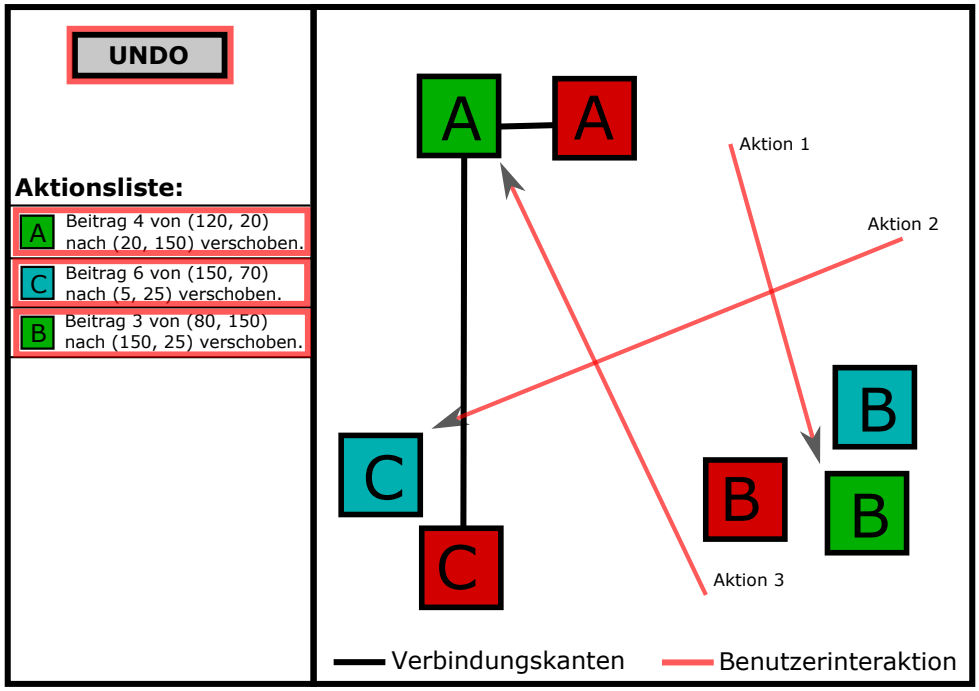


Abbildung 25: History mit wahlfreier UNDO-Funktion.

Funktion vertraut ist, nicht jedoch mit einer wahlfreien UNDO-Funktion. Daher muss zumindest im Programm die Liste der Benutzeraktionen so gestaltet sein, dass der Benutzer sie verstehen und gezielt Einträge löschen kann.

Erwartungskonformität: 0
Die hier beschriebene wahlfreie UNDO-Funktion mit einer Liste von Aktionen ist kompliziert zu benutzen und nicht für jede Benutzergruppe gleich verständlich. Ein weiteres Problem sind die Einträge der Liste, die so dargestellt werden müssen, dass der Benutzer auch nach längerer Zeit noch weiß, welche Aktion bei welchem Eintrag durchgeführt wurde. Dies kann durch eine kurze Zusammenfassung wie zum Beispiel *Knotennummer, Alte Position, Neue Position* oder eventuell eine grafische Darstellung gelöst werden. Das Verständnis der Aktionsliste, die Auswahl eines Eintrags und die Benutzung der UNDO-Funktion ist jedoch für einen Benutzer schwer zu verstehen.

Lernförderlichkeit: 2
Auch die Lernförderlichkeit wird von den schon erläuterten Nachteilen in Mitleidenschaft gezogen. Das Erlernen der Benutzung dieser Technik wird durch folgende Probleme erschwert: Der Benutzer könnte nach längerer Zeit nicht mehr wissen, welcher Eintrag in der Liste für welche Aktion steht. Auch durch geschickte Visualisierung der Daten der Interaktion ist dieses Problem schwer zu lösen. Zudem kann der Benutzer sich nicht sicher sein, welche Auswirkung das Löschen eines bestimmten Beitrages auf das Clustering-Ergebnis hat.

Steuerbarkeit: 3
Die Steuerbarkeit erhält durch die neue Erweiterung geringfügige Vorteile und einige Nachteile, weshalb die Bewertung letztlich gleich bleibt. Ein Vorteil besteht darin, dass

kurz zurück liegende Aktionen für den Benutzer gut überschaubar sind und bei einer Fehleingabe effizient gelöscht werden können. Bei länger zurück liegenden Aktionen ist dieser Vorteil jedoch nicht mehr gegeben, da der Benutzer nicht mehr genau wissen kann, welche Aktion durchgeführt wurde und wie diese das Clustering-Ergebnis beeinflusste.

Fehlertoleranz: 3
Die Fehlertoleranz hat sich durch die Erweiterung einerseits verbessert, da fehlerhafte Aktionen der näheren Vergangenheit effizient gelöscht werden können. Sie hat sich jedoch andererseits verschlechtert, was länger zurück liegende Aktionen betrifft. Da der Benutzer keinen (detaillierten) Überblick über diese Aktionen hat, werden neue Fehler provoziert, wenn er versucht diese Fehler zu korrigieren und eventuell die falschen Aktionen aus der Liste löscht.

4.5 Zusammenfassung

Es folgt eine Tabelle, die die Ergebnisse der Evaluation der Interaktionstechniken übersichtlich darstellt:

Name	**AA**	**SBF**	**EK**	**LF**	**ST**	**FT**	**Summe**
1 Beitrag verschieben	2	5	2	5	3	3	**20**
x Beiträge verschieben	2	5	2	5	1	0	**15**
x Beiträge markieren	4	4	3	5	5	4	**25**
Manuelle Modifikation von w	3	5	4	4	5	5	**26**
Schlüsselparameter	2	5	4	5	4	5	**25**
Wichtigste Parameter	3	5	4	5	5	5	**27**
History mit UNDO-Funktion	3	4	2	5	3	4	**21**
History mit Random Access UNDO	2	2	0	2	3	3	**12**

Tabelle 4: Übersicht der Interaktionstechniken und ihrer Bewertungen. AA = Aufgabenangemessenheit. SBF = Selbstbeschreibungsfähigkeit. EK = Erwartungskonformität. LF = Lernförderlichkeit. ST = Steuerbarkeit. FT = Fehlertoleranz.

Ziel der Evaluation war es, die drei besten Interaktionstechniken für eine Implementierung im Moderator's Interface zu finden. Da sich jedoch *Schlüsselparameter* und *x Beiträge markieren* den dritten Platz teilen, werden beide Techniken implementiert. Daher werden insgesamt folgende vier Techniken implementiert:

- Wichtigste Parameter
- Manuelle Modifikation von w
- Schlüsselparameter
- x Beiträge markieren

5 Konzept und Implementierung

In diesem Kapitel wird die Konzeption und Entwicklung der für die Abschlussarbeit benötigten Software-Module dokumentiert. Im ersten Abschnitt wird ein abstraktes Grundkonzept und ein praxisnaher Ablaufplan für den Clustering-Algorithmus vorgestellt. Danach wird die Implementierung der Module, die für die einzelnen Phasen benötigt werden, beschrieben. Dazu zählen die Clustering-Parameter, Interaktionstechniken des Benutzers mit dem Moderators Interface, das C-Means-Clustering-Verfahren sowie die Visualisierung der Ergebnisse. Anschließend folgt ein Abschnitt über externe Software und Frameworks, die zur Implementierung verwendet wurden.

5.1 Grundkonzept

5.1.1 Mathematische Betrachtung

In diesem Abschnitt geht es um eine allgemeine Betrachtung des zu lösenden Strukturierungsproblems.

Gegeben ist eine Menge $X = \{x_1, ..., x_n\}$, die alle Diskussionsbeiträge x_i der aktuellen Diskussion enthält. Jeder Beitrag x_i beinhaltet Daten, die seine Eigenschaften, wie zum Beispiel Beitragstyp, Inhalt oder ein- und ausgehende Kanten, innerhalb des Diskussionsgraphen beschreiben.

Zudem gibt es einen Vektor $w = (a_1, ..., a_m)$, der für jeden Clustering-Parameter einen Gewichtungswert enthält. Ein Gewichtungswert a_j liegt zwischen 0 und 1 und soll angeben, wie wichtig der jeweilige Parameter j für den Strukturierungsvorgang ist, wobei 1 in diesem Fall *sehr wichtig* und 0 *absolut unwichtig* bedeutet. w wird zunächst mit gleichen Werten für jeden Parameter (beispielsweise 0.50) initialisiert und je nach Art und Grad der Benutzerinteraktion modifiziert.

X repräsentiert den Diskussionsgraphen, x_i die jeweiligen Diskussionsbeiträge und w die Kriterien, nach denen die Diskussionsbeiträge gruppiert werden sollen. w wird durch die Interaktion des Benutzers modifiziert - je besser also die Interaktion interpretiert und auf Werte von w übertragen werden kann, desto besser kann die "Absicht" des Benutzers im späteren Clustering umgesetzt werden.

Gesucht ist eine Aufteilung von X in disjunkte Teilmengen $C_1, ..., C_c$, anhand der Parameterwerte der Diskussionsbeiträge x_i sowie des Gewichtungsvektors w. Diese Aufteilung wird durch den C-Means-Clustering-Algorithmus berechnet.

5.1.2 Ablaufplan

Dieser Abschnitt beschreibt das Konzept, wie das geplante Clustering-Verfahren praktisch umgesetzt werden soll. Grundsätzlich wird der Ablauf in vier Phasen aufgeteilt:

1. *Preprocessing:* In diesem Schritt werden alle Informationen der Diskussionsbeiträge x_i extrahiert, so dass diese in ein Datenformat gebracht werden, das eine korrekte und leichte Weiterverarbeitung erlaubt. Danach werden diese Werte normalisiert und als Parameter für jeden einzelnen Diskussionsbeitrag gespeichert.

2. *Benutzerinteraktion:* Der Benutzer interagiert mit Elementen der Diskussionskarte. Die daraus resultierenden Informationen werden verwendet, um den Gewichtungsvektor w anzupassen.

3. *C-Means-Clustering:* Die Diskussionsbeiträge werden anhand der Daten von X und w gruppiert, woraus folgt, dass jeder Diskussionsbeitrag x_i einem bestimmten Cluster C_k zugeordnet wird.

4. *Visualisierung und Abschluss:* Die Gruppenzugehörigkeiten werden visualisiert, indem Beiträge der selben Gruppe zueinander hingezogen werden, während sich Beiträge aus verschiedenen Gruppen abstoßen. Die Bewegung der Diskussionsbeiträge darf nicht zu schnell erfolgen, da der Benutzer die Gelegenheit haben soll, bei nicht zufriedenstellenden Ergebnissen nochmals einzugreifen. In diesem Fall wird bei Punkt 2. fortgefahren.

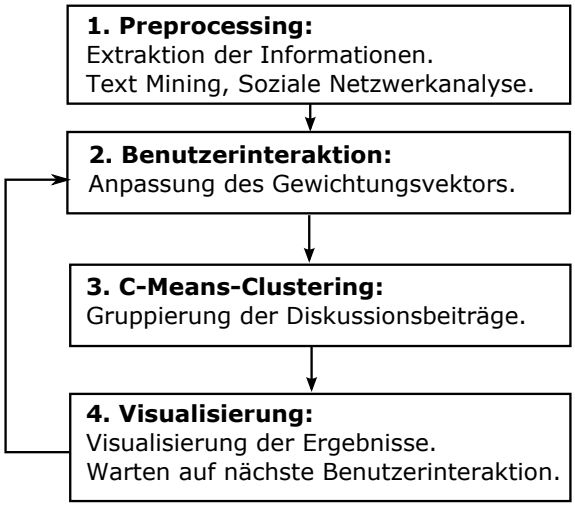


Abbildung 26: Vier Phasen des interaktiven Clustering-Verfahrens.

5.1.3 Einbettung in das Moderator's Interface

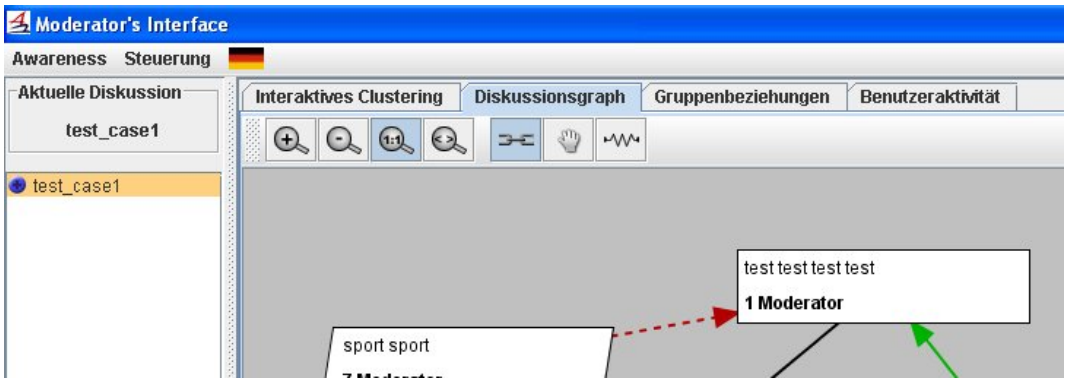


Abbildung 27: Ausschnitt des Moderator's Interface (Screenshot).

Das Moderator's Interface ist aufgrund seiner Architektur sehr variabel konfigurierbar. Es besteht aus einer Kernanwendung, die mit Hilfe eines Untermoduls beliebig viele Plugins einbinden kann. Grundsätzlich hat das Moderator's Interface wenige Funktionen, was jedoch durch zusätzliche Plugins ausgeglichen werden kann. Dies bringt zudem den weiteren Vorteil, dass der Benutzer nur die Plugins einbinden muss, die er auch wirklich benötigt.

In Abbildung 27 ist ein Ausschnitt des Moderator's Interface zu sehen, das in diesem Fall so konfiguriert ist, dass vier Plugins verwendet werden. Dabei handelt es sich um *Interaktives Clustering*, *Diskussionsgraph*, *Gruppenbeziehungen* und *Benutzeraktivität*.

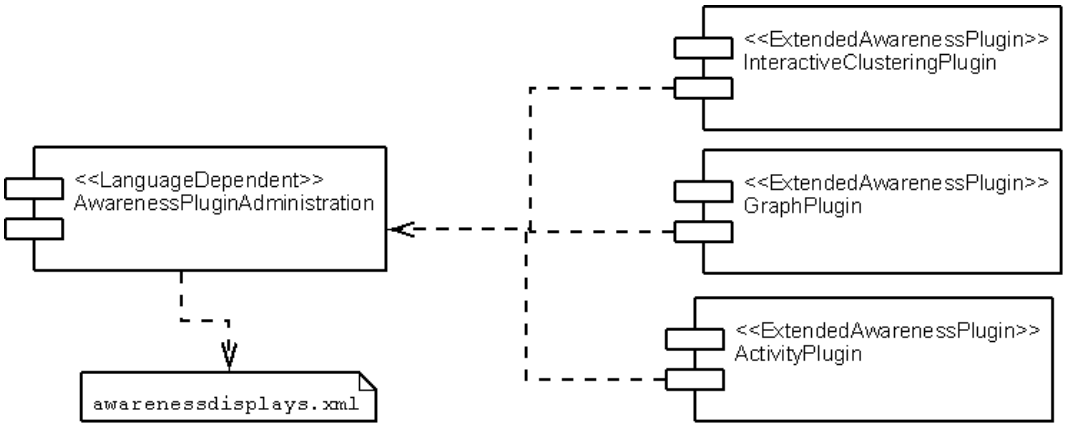


Abbildung 28: Komponentendiagramm zur Plugin Struktur des Moderator's Interface.

Abbildung 28 zeigt, wie die Einbindung von Plugins im Moderator's Interface intern strukturiert ist. Es gibt ein Modul zur Administration von Plugins (Klassenname *AwarenessPluginAdministration*), in dem auf die XML-Datei *awarenessdisplays.xml* zugegriffen wird. In dieser XML-Datei befinden sich Informationen darüber, welche Plugins (in diesem Beispiel *GraphPlugin*, *ActivityPlugin* und *InteractiveClusteringPlugin*) beim Start des Moderator's Interface geladen werden sollen. Jedes Plugin muss eine Schnittstelle *ExtendedAwarenessPlugin* implementieren, die bestimmte Methoden beinhaltet, die zur korrekten Darstellung in der graphischen Benutzeroberfläche notwendig sind.

Das Modul zur Verwaltung von Plugins selbst ist nur eine von mehreren Unterkomponenten des Gesamtsystems und wird vom Hauptprogramm instanziiert.

Die für diese Arbeit benötigte Funktion des interaktiven Clusterings wurde entsprechend als Plugin für das Moderator's Interface implementiert und kann dort als Zusatzwerkzeug ausgewählt werden. Die entsprechenden GUI-Elemente für die jeweilige Interaktionstechnik wurden in der Klasse *InteractiveClusteringPluginPanel* implementiert. Diese verweist wiederum auf eine Klasse *CMeansHighDimensionalClustering*, in der sich ein Großteil der Programmlogik des Cluster-Analyse-Vorgangs befindet.

5.2 Clustering-Parameter

Ein Diskussionsgraph kann im Laufe einer Diskussion eine komplexe Struktur erreichen. Zudem besteht jeder einzelne Diskussionsbeitrag aus vielen verschiedenen Daten, die zu

einer formalen Beschreibung herangezogen werden könnten. Davon werden folgende Parameter für den Clustering-Algorithmus verwendet:

- Zeit
- Autor
- Knotentyp
- Inhalt
- Länge des Inhaltes
- Struktur
- Knotengrad

Die Parameter wurden so gewählt, dass möglichst viele verschiedene Aspekte der graphbasierten Diskussion bei einer Gruppierung berücksichtigt werden können. Die Liste ist nicht vollständig, da noch viele weitere Parameter aus dem Diskussionsgraph extrahiert beziehungsweise durch Kombination der vorhandenen Parameter neu erschlossen werden können. Da der Fokus dieser Arbeit hauptsächlich auf der Kombination von Clustering und Interaktion liegen soll, sollen diese sieben Parameter exemplarisch ausreichen. Die Parameter haben die folgende Bedeutung:

Zeit: Jeder Diskussionsbeitrag enthält mehrere Zeitstempel, die zum Beispiel den Entstehungszeitpunkt oder den Zeitpunkt der letzten Änderung bezeichnen. Der Parameter *Zeit* wird mit dem Wert des Entstehungszeitpunkt des jeweiligen Beitrages belegt. Würde ein Benutzer ein rein zeitliches Clustering vornehmen, könnte er durch die entstehenden Gruppen eine Art Chronologie erkennen, welche Diskussionsbeiträge in welchen Intervallen entstanden sind.

Autor: Für jeden Diskussionsbeitrag werden auch Informationen über den Autor gespeichert. Um diese in ein für den Algorithmus verwertbares Format zu bringen, werden die Namen der Autoren als Zahlen kodiert.

Diese Kodierung bringt jedoch ein semantisches Problem mit sich: Gibt es beispielsweise die Autoren 1, 2, 3 und es wird ein Clustering, wie in Abschnitt 3.3 beschrieben, durchgeführt, so würde die (euklidische) Distanz zwischen Autor 1 und 2 geringer sein als die Distanz zwischen 1 und 3. Da die Zahlen jedoch völlig wertfrei sein sollen, ist dieser Effekt nicht wünschenswert. Wie das Problem gelöst wird, wird in Abschnitt 5.4 weiter erläutert.

Knotentyp: Jeder Knoten eines Diskussionsgraphen besitzt einen Typ wie *Frage*, *Behauptung*, *Idee* oder *Information*. Die genannten Knotentypen sind keineswegs vollständig; in Programmen für graphbasierte Diskussionen gibt es für den Benutzer in der Regel die Möglichkeit, Knotentypen selbst zu definieren. Diese Informationen werden, wie schon beim Parameter *Autor*, als Zahlen kodiert, womit auch hier ein besonderes Ähnlichkeitsmaß verwendet werden muss.

Inhalt: Für den Parameter *Inhalt* kommen die Techniken aus Abschnitt 3.1 zum Einsatz. Mit Hilfe des Java Frameworks Lucene (siehe Abschnitt 5.6) werden zunächst die Term-Frequenz-Vektoren aus den Feldern *Titel* und *Notiz* der Beiträge, extrahiert. Diese geben

nun an, welche Wörter wie oft in einem Diskussionsbeitrag vorkommen. Danach werden diese Werte normalisiert sowie die inverse Dokumentfrequenz für jeden Term bestimmt.

Aus diesen Werten wird nun die Term-Dokument-Matrix berechnet. Anschließend wird mit Hilfe der entstandenen Dokumentvektoren und dem Cosinus-Maß als Ähnlichkeitsmaß die Dokument-Dokument-Matrix / Ähnlichkeitsmatrix berechnet. Letztere gibt zu jedem Diskussionsbeitrag an wie ähnlich sein Inhalt zu den anderen Beiträgen ist. Für einen Diskussionsgraphen mit k Elementen ist der Parameter *Inhalt* also für jeden Diskussionsbeitrag ein Vektor mit k Werten.

Länge des Inhaltes: Dieser Parameter repräsentiert die Textlänge jedes Diskussionsbeitrages. Er ergibt sich aus der Länge der beiden Felder *Titel* und *Notiz*.

Struktur: Zur Extraktion der Struktur des Diskussionsgraphen wird eine Adjazenzmatrix generiert. Für jeden Diskussionsbeitrag ist der Parameter *Struktur* demnach eine Zeile der Adjazenzmatrix. Wie bereits in Abschnitt 3.2 beschrieben, reichen diese Informationen aus, um Positionen innerhalb eines Graphen ausfindig zu machen. Beim C-Means-Clustering soll die euklidische Distanz als Distanzmaß verwendet werden. Ein Clustering ausschließlich nach diesem Parameter würde darin resultieren, dass jedes Cluster eine Gruppe von strukturell ähnlichen Beiträgen repräsentiert.

Knotengrad: Dieser Parameter gibt an, wie viele ein- und ausgehende Kanten ein Diskussionsbeitrag besitzt.

Es folgt eine tabellarische Übersicht der Parameter:

Name	**Beschreibung**	**Datentyp**
Zeit	Entstehungszeitpunkt	double
Autor	Autor	double
Knotentyp	Diskussionsbeitragstyp	double
Inhalt	Ähnlichkeitsvektor basierend auf Term-Frequenz-Vektoren	double[]
Länge des Inhaltes	Länge der Felder *Titel* und *Notiz*	double
Struktur	Vektor basierend auf Adjazenzmatrix: Ähnlichkeit ein- und ausgehender Kanten	double[]
Knotengrad	Anzahl der ein- und ausgehenden Kanten	double

Tabelle 5: Übersicht der verwendeten Clustering-Parameter.

Um eine korrekte Weiterverarbeitung der Parameter zu ermöglichen werden diese normalisiert, so dass deren Werte in einen Bereich zwischen 0 bis 1 überführt werden. Dieser Schritt ist besonders im Hinblick auf die Berechnungen mit dem Gewichtungsvektor w notwendig, da sich die Wertebereiche der Clustering-Parameter teilweise stark voneinander unterscheiden. Würden die Werte nicht normalisiert werden, so würden Parameter, die tendenziell hohe Werte haben, einen dominanten Einfluss auf den Clustering-Algorithmus haben, während Parameter mit tendenziell niedrigen Werten kaum ins Gewicht fallen. Diese ungewollten Einflüsse würden das Clustering-Ergebnis verfälschen, wenn beispielsweise aufgrund der Daten des Gewichtungsvektors nach einem Parameter gruppiert wer-

den soll, der eher niedrige Werte beinhaltet. Denn auch in einem solchen Fall würde die hohe Bewertung des Parameters im Gewichtungsvektor nicht ausreichen, um den Einfluss der Parameter mit hohen Werten zu übertreffen.

Folgende Formel wird zur Normalisierung verwendet:

$$v' = \frac{v - min}{max - min} \cdot (max_{norm} - min_{norm}) + min_{norm} \tag{29}$$

Dabei ist v der Ursprungsvektor, v' der normalisierte Vektor, sowie min und max der minimale beziehungsweise maximale Wert der alten Wertspanne. min_{norm} und max_{norm} bezeichnen die Extremwerte der neuen Wertspanne und werden für diese Anwendung 0 und 1 sein.

Abbildung 29: PreProcessing: Übersicht der Klassen der Vorverarbeitung.

Abbildung 29 zeigt die Klassen, die für die Extraktion der Clustering-Parameter verantwortlich sind. *PreProcessor* ist die Hauptklasse des PreProcessing. Sie stellt für jeden Parameter eine öffentliche Getter-Methode[13] bereit und wird von *CMeansHighDimensional-Clustering*, die Klasse in der das Clustering selbst stattfindet, instanziiert.

PreProcessor verwendet für die inhaltlichen und strukturellen Parameter die Zusatzklassen *TextMiner* und *NetworkAnalyzer*. Dies hat den Vorteil, dass diese Module beliebig gegen andere Verfahren ausgetauscht werden können, solange letztlich die Ergebnisse mit den richtigen Datentypen ($double[]$, $double[][]$) geliefert werden.

Die Parameter *Zeit*, *Autor* und *Knotentyp* werden nicht in separaten Klassen, sondern direkt

[13]Es handelt sich natürlich um keine einfache Getter-Methode, die nur einen Wert zurückgibt, da einige Clustering-Parameter zunächst berechnet werden müssen.

in *PreProcessor* aus den Variablen des Moderator's Interface ausgelesen. Da dieser Vorgang relativ einfach ist und sich mit hoher Wahrscheinlichkeit in der Zukunft nicht wesentlich ändern wird, gibt es für jeden dieser Parameter lediglich eine eigene Methode und keine eigene Klasse.

5.3 Interaktionstechniken

In diesem Abschnitt folgt eine Beschreibung, wie die in Kapitel 4 vorgestellten und ausgewählten vier Interaktionstechniken implementiert wurden und wie die Ergebnisse im Moderator's Interface aussehen. Jede Technik wurde separat implementiert, so dass eine einzelne Betrachtung der Funktionsweise und Leistungsfähigkeit möglich ist. Dabei werden die Interaktionstechniken in folgender Reihenfolge vorgestellt:

- Angabe eines Schlüsselparameters
- Angabe der wichtigsten Parameter
- Manuelle Modifikation des Gewichtungsvektors
- Markierung ähnlicher Diskussionsbeiträge

Die GUI-Elemente für diese vier Interaktionstechniken wurden in der Klasse *InteractiveClusteringPluginPanel* implementiert. Jede Interaktionstechnik besitzt eine *boolean*-Variable, die bestimmt, ob die jeweilige Technik verwendet werden soll. Für die Technik, deren Variable *TRUE* ist, wird eine Methode aufgerufen, die entsprechende GUI-Elemente einer Werkzeugleiste hinzufügt.[14]

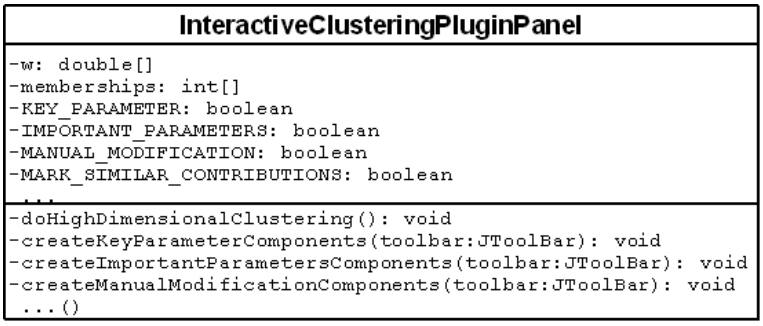


Abbildung 30: Klassendiagramm: *InteractiveClusteringPluginPanel*.

5.3.1 Angabe eines Schlüsselparameters

Zur Steuerung gibt es in der linken Kontrollleiste eine Drop-Down-Liste, die die Auswahl eines Clustering-Parameters ermöglicht, sowie direkt darunter einen Knopf, mit dem das Clustering gestartet wird.[15]

[14] Anmerkung: Für die Interaktionstechnik *Markierung ähnlicher Diskussionsbeiträge* gibt es zwar eine *boolean*-Variable, damit das Plugin im Falle eines Clusterings die Interaktionsdaten korrekt auswerten kann, jedoch werden keine weiteren GUI-Elemente benötigt.

[15] Alle anderen Steuerelemente, die in Abbildung 31 zu sehen sind, sind Standard-Steuerelemente des Moderator's Interface und gehören nicht zu dieser speziellen Interaktionstechnik.

Der Gewichtungsvektor wird so manipuliert, dass der ausgewählte Parameter den Wert 1.00 zugewiesen bekommt, während alle anderen Parameter auf den Wert 0.00 gesetzt werden. Die Werte werden erst in den Gewichtungsvektor geschrieben, nachdem der Start-Knopf betätigt wurde.

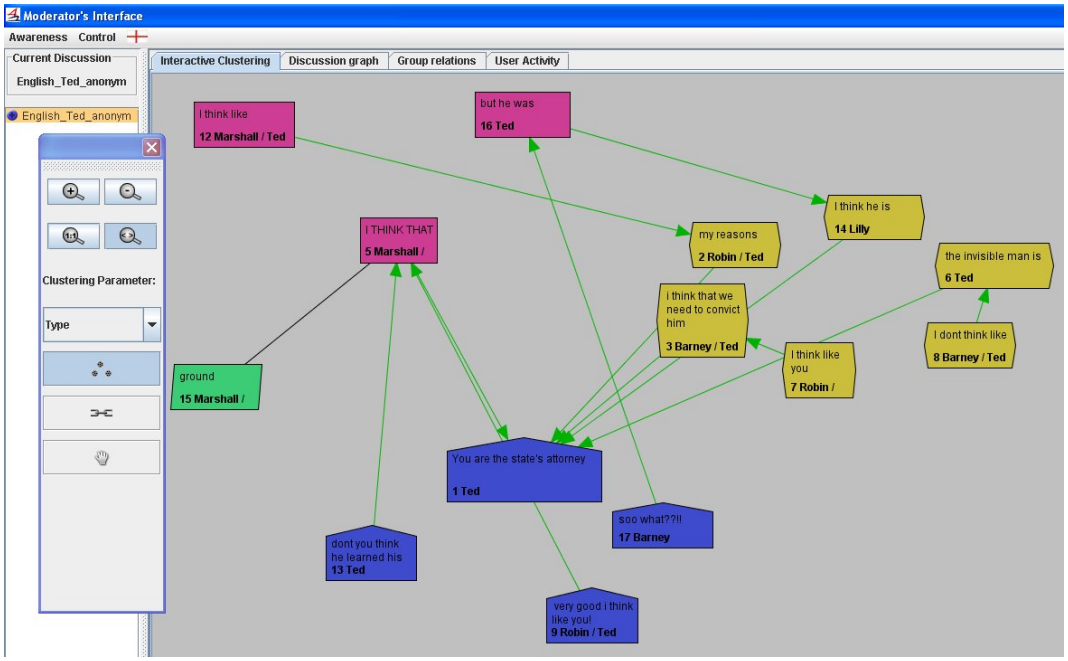


Abbildung 31: Angabe eines Schlüsselparameters (Screenshot).

Abbildung 31 zeigt einen Screenshot, wie diese Interaktionstechnik im Moderator's Interface implementiert wurde. In diesem Beispiel findet eine Gruppierung nach dem *Knotentyp*-Parameter statt, so dass vier Cluster entstehen.

5.3.2 Angabe der wichtigsten Parameter

Der Benutzer hat die Möglichkeit, für jeden einzelnen Parameter aus einer Drop-Down-Liste einen Wert zu bestimmen. Er hat dabei die Auswahl aus den Werten *unwichtig*, *weniger wichtig*, *normal*, *wichtig* sowie *sehr wichtig*. Es ist dabei nicht von Bedeutung, ob er alle Werte auf *unwichtig* oder *sehr wichtig* setzt, auch wenn dies offensichtlich keine sinnvollen Ergebnisse liefern kann. Unter den Drop-Down-Listen befindet sich ein Knopf, um das Clustering zu starten.

Jedem der Werte der Drop-Down-Liste ist ein fester Wert zwischen 0.00 und 1.00 zugeordnet. *Unwichtig* entspricht 0.00, *weniger wichtig* 0.25, *normal* 0.50, *wichtig* 0.75 und *sehr wichtig* 1.00. Diese Werte werden nach Betätigen des Start-Knopfes in den Gewichtungsvektor geschrieben. Vorher kann der Benutzer beliebig lange die Werte verändern.

Abbildung 32 zeigt einen Screenshot, wie diese Interaktionstechnik im Moderator's Interface implementiert wurde. In diesem Beispiel findet eine Gruppierung nach dem *Autor-*

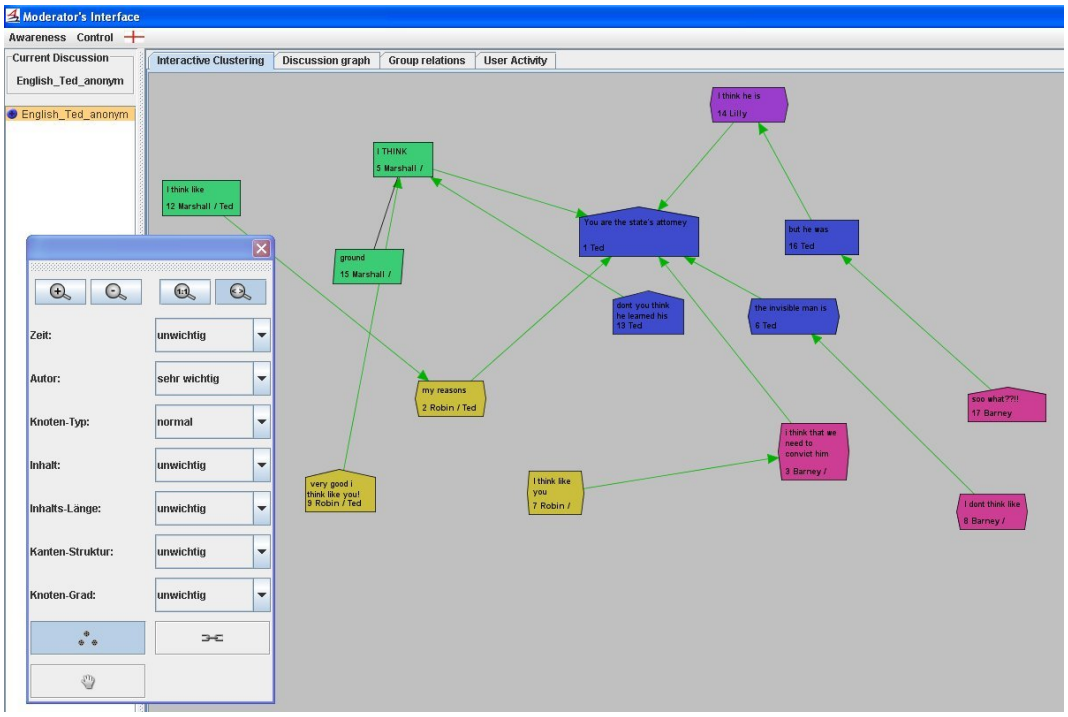


Abbildung 32: Angabe der wichtigsten Parameter (Screenshot).

Parameter (sehr wichtig), sowie nach dem *Knotentyp*-Parameter (normal) statt, so dass fünf Cluster entstehen.

5.3.3 Manuelle Modifikation des Gewichtungsvektors

Bei dieser Interaktionstechnik gibt es für jeden Parameter einen Schieberegler mit einem Intervall von 0 bis 100. Diese Werte sollen in etwa den prozentualen Einfluss eines Parameters bezüglich des Clustering-Vorgangs angeben. Unter den Schiebereglern befindet sich ein Knopf zum Starten des Clusterings.

Die Werte der Schieberegler müssen nach dem Auslesen minimal angepasst werden. Sie werden mit der selben Formel, die für die Normalisierung der Parameterwerte der Diskussionsbeiträge verwendet wurde, umgeformt und auf einen Bereich zwischen 0.00 und 1.00 übertragen.

Abbildung 33 zeigt einen Screenshot, wie diese Interaktionstechnik im Moderator's Interface implementiert wurde. In diesem Beispiel findet eine Gruppierung nach dem *Inhalt*-Parameter (Wert nahe 1.00) und dem *Knotentyp*-Parameter (Wert nahe 0.25) statt, so dass zehn Cluster entstehen.

5.3.4 Markierung ähnlicher Diskussionsbeiträge

Die einzigen grafischen Steuerelemente dieser Interaktionstechnik sind der Knopf zum Starten des Clustering-Vorgangs sowie ein Kontextmenü, das per Rechtsklick geöffnet werden kann. Bevor das Clustering gestartet wird, muss der Benutzer eine Auswahl treffen, wie eine Gruppierung nach seinen Vorstellungen auszusehen hat. Dazu kann er mehrere

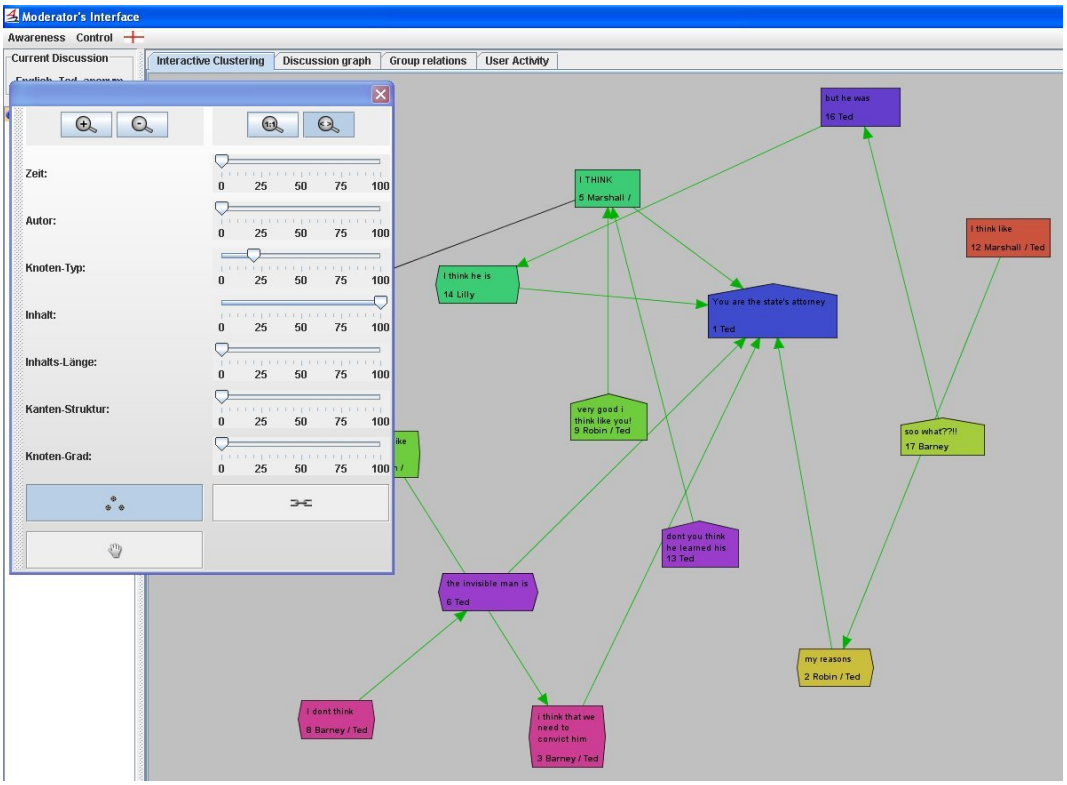


Abbildung 33: Manuelle Modifikation des Gewichtungsvektors (Screenshot).

Diskussionsbeiträge markieren, entweder, indem er mit der Maus einen Rahmen um mehrere Beiträge zieht, oder, indem er mit gehaltener SHIFT-Taste und der Maus per Linksklick mehrere Beiträge auswählt.

Nachdem der Benutzer seine Auswahl getroffen und den Start-Knopf betätigt hat, müssen die ausgewählten Diskussionsbeiträge analysiert und interpretiert werden, was durch folgendes Verfahren realisiert wurde: Intuitiv gesprochen wird ein Clustering mit zwei Clustern simuliert, wobei die markierten Knoten ein Cluster repräsentieren und die restlichen, nicht markierten Knoten das andere. Dabei werden die Ähnlichkeiten der markierten Knoten untereinander aufsummiert, was den Wert des jeweiligen Parameters im Gewichtungsvektor w erhöht. Zusätzlich werden die Ähnlichkeiten zwischen markierten und nicht markierten Beiträgen auf einen anderen Wert aufsummiert. Dieser Wert verringert die Wertigkeit des Parameters in w.

Es folgt eine formale Beschreibung der verwendeten Variablen sowie der Ablauf des Verfahrens in Pseudocode.

Es gibt einen temporären Gewichtungsvektor tmp_w, der in jedem Durchlauf des folgenden Verfahrens genau einen Clustering-Parameter maximal (1.00) und alle anderen minimal (0.00) bewertet. Der Wert von tmp_w ist demnach im dritten Iterationsschritt $(\ 0.00,\ 0.00,\ 1.00,\ 0.00,\ 0.00,\ 0.00,\ 0.00\ )$. Die Zähler-Variable w_count gibt an, bei welchem Parameter $(0$ bis $6)$ der Algorithmus sich gerade befindet. w ist der reale Gewichtungsvektor, der entsprechend der Analyseergebnisse modifiziert wird.

Es gibt zwei Knoten i und j aus der Gesamtknotenmenge N. Im Laufe der Prozedur werden alle möglichen Kombinationen von i und j miteinander verglichen.

Die Variable $count_selected$ zählt, wie oft zwei Knoten miteinander verglichen wurden, die jeweils vom Benutzer markiert wurden, während $count_diff$ zählt, wie oft zwei Knoten miteinander verglichen wurden, von denen nur genau einer markiert wurde. In $diff_w$ werden die Ähnlichkeiten von markierten und nicht markierten Knoten gespeichert.

Die vorgestellten Variablen werden folgendermaßen dazu verwendet, um die Benutzerinteraktion auszuwerten:

Prozedur: $calc_weights$
Zweck: Algorithmus zur Berechnung der signifikanten Parameter von mehreren Diskussionsbeiträgen
Parameter: tmp_w, w_count, w, i, j, N, $count_selected$, $count_diff$, $diff_w$

```
1  Setze w_count = 0.
2  Solange w_count<tmp_w.length wiederhole
3      Setze tmp_w = { 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00 }.
4      Setze tmp_w[w_count] = 1.00.
5      Vergleiche fuer alle i, j aus N
6          Falls i markiert und j markiert, dann
7              Setze w[w_count] = w[w_count] + 1 / (1 + EuklDist(i,j)).
8              Setze count_selected = count_selected + 1.
9          Falls i markiert und j nicht markiert oder i nicht markiert und j
               markiert, dann
10             Setze diff_w = diff_w + 1 / (1 + EuklDist(i,j)).
11             Setze count_diff = count_diff + 1.
12     Falls count_selected > 0, dann
13         Setze w[k] = w[k] / count_selected.
14         Setze diff_w = diff_w / count_diff.
15         Setze w[k] = w[k] - diff_w.
16     Setze w_count = w_count + 1.
```

Ergebnis: w

Der Aspekt, dass die Ähnlichkeiten zwischen markierten und nicht markierten Beiträgen die Werte des Gewichtungsvektors verringern müssen, muss berücksichtigt werden, da es für ein Clustering nicht ausreicht, lediglich die Elemente des eigenen Clusters zu betrachten. Stellen wir uns beispielsweise eine Diskussionskarte vor, bei der genau ein Benutzer alle Beiträge erstellt hat. Nun markiert der Moderator mehrere Beiträge, die den gleichen Knotentyp haben, mit der Absicht, dass die restlichen Beiträge nach diesem Aspekt gruppiert werden. Was ihm jedoch nicht aufgefallen ist, da es ihm offensichtlich nicht wichtig war, ist, dass die markierten Beiträge den selben Autor haben. Würde der Algorithmus nur Ähnlichkeiten von Elementen des selben Clusters berücksichtigen, so würde der Parameter *Autor* in diesem Fall überbewertet werden.

Nachdem alle Werte des Gewichtungsvektors berechnet wurden, werden diese mit der selben Formel, die für die Normalisierung der Parameterwerte der Diskussionsbeiträge verwendet wurde, umgeformt und auf einen Bereich zwischen 0.00 und 1.00 übertragen.

Um diese Interaktionstechnik interaktiver zu gestalten, wurde eine zusätzliche Funktion implementiert, die dem Benutzer die Möglichkeit bietet, nachdem ein initiales Clustering berechnet wurde, einen oder mehrere Diskussionsbeiträge per Linksklick (und gegebenen-

falls Halten der SHIFT-Taste) auszuwählen und danach per Rechtsklick ein Kontextmenü zu öffnen. In diesem Menü kann er die Auswahl treffen, dass aus den aktuell ausgewählten Diskussionsbeiträgen ein neues Cluster entsteht. Dadurch kann der Benutzer das Ergebnis weiter verfeinern bis es seinen Wünschen entspricht. Diese Funktion soll, wie schon die ursprüngliche Technik, nicht nur die Gruppenzugehörigkeit der ausgewählten Beiträge, sondern das gesamte Clustering-Ergebnis verändern.

Die Funktion zum Bilden neuer Cluster aus ausgewählten Diskussionsbeiträgen hat je nach Anzahl der ausgewählten Beiträge unterschiedliche visuelle Effekte für den Benutzer:

- *Genau ein Beitrag wird ausgewählt:* Genau ein Diskussionsbeitrag wird vom Ursprungscluster abgespalten und bildet ein eigenständiges Cluster.
- *Mehrere Beiträge des selben Clusters werden ausgewählt:* Mehrere Diskussionsbeiträge werden vom selben Ursprungscluster abgespalten und bilden ein eigenständiges Cluster.
- *Mehrere Beiträge aus jeweils unterschiedlichen Clustern werden ausgewählt:* Mehrere Diskussionsbeiträge werden von ihren Ursprungsclustern abgespalten und zu einem neuen Cluster vermischt.

Technisch gesehen erfolgt, wie schon bei der ursprünglichen Interaktionstechnik, eine Manipulation des Gewichtungsvektors w und danach wird ein erneutes Clustering durchgeführt. Dabei wird gewährleistet, dass die Diskussionsbeiträge, die der Benutzer ausgewählt hat, immer dem selben Cluster zugeordnet werden.

Bei der Manipulation des Gewichtungsvektors werden zunächst alle Diskussionsbeiträge miteinander verglichen. Dabei müssen folgende Fälle beim Vergleich von jeweils zwei Diskussionsbeiträgen bei der Berechnung berücksichtigt werden:

- *Von den zu vergleichenden Beiträgen ist genau einer aktiv ausgewählt und beide gehören aktuell zum selben Cluster:* Die beiden Beiträge werden auf Ähnlichkeiten hin untersucht und diese werden von dem Gewichtungsvektor subtrahiert, da die beiden betrachteten Beiträge absichtlich vom Benutzer voneinander getrennt werden. Aufgrund dessen können Ähnlichkeiten zwischen diesen Beiträgen kein für den Benutzer sinnvolles Clustering-Kriterium darstellen.
- *Keiner der beiden Beiträge ist aktiv ausgewählt, jedoch gehören beide zum selben Cluster. Zudem enthält dieses Cluster zumindest einen ausgewählten Diskussionsbeitrag:* Die beiden betrachteten Beiträge werden in diesem Fall nicht direkt manipuliert, jedoch werden ihre Cluster verändert, indem einer oder mehrere Diskussionsbeiträge daraus entfernt werden. Daher wird die Tatsache, dass der Benutzer diese nicht ausgewählten Beiträge in ihrem ursprünglichen Cluster lässt, so bewertet, dass Ähnlichkeiten zwischen diesen Beiträgen sich positiv auf den Gewichtungsvektor auswirken.
- *Beide Beiträge sind aktiv ausgewählt:* Die beiden Beiträge werden auf Ähnlichkeiten hin untersucht und diese werden auf den Gewichtungsvektor addiert, da die beiden betrachteten Beiträge absichtlich vom Benutzer zu einem neuen Cluster zusammengelegt werden. Daher stellen Ähnlichkeiten zwischen diesen Beiträgen sinnvolle Clustering-Kriterien dar.

Nachdem der Gewichtungsvektor nach oben beschriebenen Kriterien manipuliert und danach normalisiert wurde, erfolgt ein erneutes C-Means-Clustering. Danach werden die selektierten Diskussionsbeiträge dem selben Cluster zugeordnet, falls dies durch das Clustering nicht ohnehin schon der Fall war.

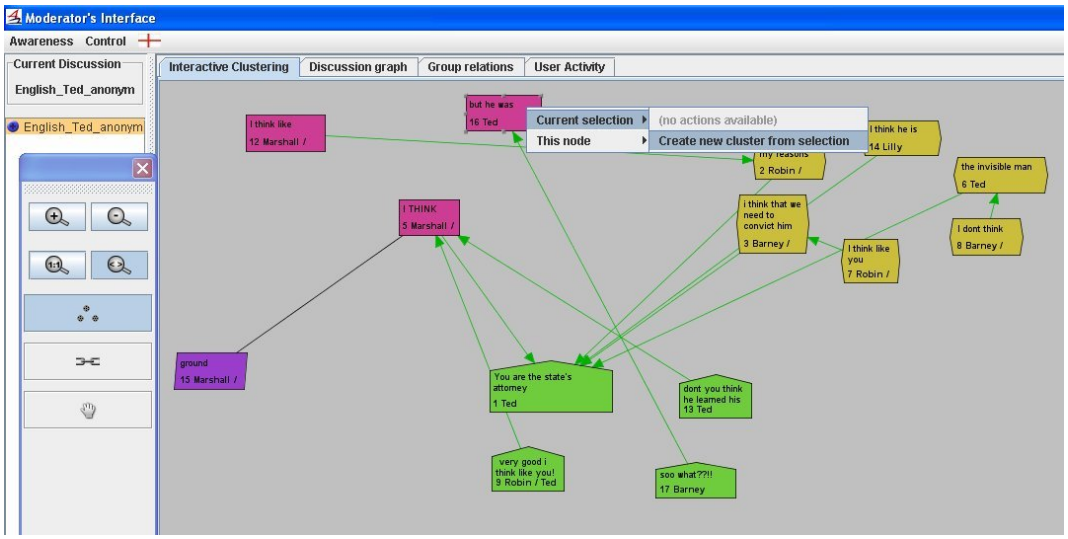


Abbildung 34: Markierung ähnlicher Diskussionsbeiträge (Screenshot).

Abbildung 34 zeigt einen Screenshot, wie diese Interaktionstechnik im Moderator's Interface aussieht. Der Benutzer hat mehrere Diskussionsbeiträge markiert und danach das Clustering gestartet. Die Auswahl wurde offensichtlich so interpretiert, dass ein Clustering nach dem Parameter *Knotentyp* durchgeführt wurde, so dass vier Cluster entstehen. Dies erkennt man daran, dass Beiträge mit der gleichen Form nah beieinander platziert und farblich gleich markiert wurden. Zudem ist in der Abbildung zu sehen, dass ein Diskussionsbeitrag ausgewählt und danach per Rechtsklick ein Kontextmenü geöffnet wurde. Dort kann der Beitrag mit der Auswahl *Create new Cluster from Selection* aus seinem aktuellen Cluster entfernt werden.

5.4 C-Means-Clustering

In diesem Abschnitt geht es um den Ablauf des C-Means-Clustering-Algorithmus. Dieser wurde etwas vom Grundalgorithmus an bestimmte anwendungsspezifische Bedingungen angepasst. Dazu zählt die Modifizierung der Initialisierungsphase des Verfahrens, die Verwendung verschiedener Distanzmaße für die Clustering-Parameter, sowie die Einbindung und Auswertung des Gewichtungsvektors.

5.4.1 Bestimmung von C

Der Algorithmus startet mit der Suche nach C, also der Anzahl der verwendeten Cluster-Prototypen. C wird durch den Silhouetten-Koeffizienten bestimmt, indem jeweils verschiedene Clusterings mit 2 bis $N-1$ Prototypen getestet werden und schließlich das Beste ausgewählt wird. Bei diesem Verfahren fließt bereits der Gewichtungsvektor mit in die Berechnung ein, was jedoch in Abschnitt 5.4.3 ausführlicher erläutert wird.

Im gegebenen Anwendungsszenario musste ich bezüglich des Silhouetten-Koeffizienten beachten, dass es auch sein kann, dass die durchschnittliche Distanz eines Objektes zu allen anderen Objekten innerhalb seines eigenen Clusters gleich 0 ist ($a_i = 0$), ohne dass es sich um ein 1-elementiges Cluster handelt. Dies ist immer dann der Fall, wenn Beiträge in einem Cluster zusammen sind, die in bestimmten Parameterwerten identisch sind und das Clustering nach genau diesen Parametern durchgeführt wird. In diesem Fall wird der Silhouetten-Koeffizient nicht mit 0 (sehr schlecht), sondern mit 1 (sehr gut) bewertet. Speziell bei einem Clustering nach den Parametern *Knotentyp* oder *Autor* treten solche Fälle auf, da es in Diskussionsgraphen wahrscheinlich ist, dass ein Autor mehrere Diskussionsbeiträge generiert hat oder, dass es Beiträge vom gleichen Typ gibt.

5.4.2 Platzierung der Prototypen

Der nächste Schritt besteht darin, die C Cluster-Prototypen *gut* auf die gegebenen Datenelemente zu verteilen. Beim C-Means-Clustering können die Prototypen zufällig oder nach einem bestimmten Gittermuster verteilt werden, was jedoch zur Folge haben kann, dass das Clustering-Ergebnis nicht optimal ist. Zudem könnte ein Zufallselement in diesem Verfahren den Benutzer verunsichern, da er nicht nachvollziehen könnte, warum die gleiche Interaktion mit dem Moderator's Interface zu unterschiedlichen Ergebnissen führen würde. Die Prototypen werden daher mit dem Quick-Clustering-Verfahren so über alle Diskussionsbeiträge verteilt, dass sie in Abhängigkeit des Gewichtungsvektors w die größte euklidische Distanz zueinander haben.

5.4.3 Gruppierung der Diskussionsbeiträge

Nachdem die Daten erschlossen, der Gewichtsvektor berechnet, sowie C Prototypen korrekt platziert wurden, erfolgt die Gruppierung der Diskussionsbeiträge. Grundsätzlich folgt dieser Ablauf dem Standard C-Means-Algorithmus: Es werden abwechselnd die Gruppenzugehörigkeiten der Diskussionsbeiträge und die Positionen der Prototypen angepasst.

Als Distanzmaß wird für die meisten Parameter *(Zeit, Inhalt, Länge des Inhaltes, Struktur, Knotengrad)* die euklidische Distanz verwendet.

Bei den Parametern *Knotentyp* und *Autor* ist es jedoch nicht sinnvoll, dieses Maß zu verwenden. Die Werte dieser Parameter werden aus dem Diskussionsgraphen ausgelesen und zu Zahlen kodiert. Dabei wird keine Aussage dazu getroffen, welcher Wert ähnlicher zu welchen anderen Werten ist. Eine Distanzmessung mit der euklidischen Distanz würde jedoch ergeben, dass Autoren, die zu Werten kodiert wurden, die ähnlich zueinander sind, eine geringe Distanz zueinander haben, obwohl dies durch die Zahlen nicht ausgesagt werden sollte. Dieses Problem könnte gelöst werden, indem die Daten der Autoren analysiert, interpretiert und zu Zahlen kodiert werden, so dass jede Zahl eine inhaltliche Bedeutung hat. Da eine solche Analyse jedoch aufwendig und nicht Hauptthema dieser Arbeit ist, wurde zur Lösung dieses Problems ein anderes Distanzmaß, die *diskrete Metrik*, für die Parameter *Knotentyp* und *Autor* verwendet:

$$dist_{Diskret}(x, y) = \begin{cases} 0, & \text{falls } a = b \\ 1, & \text{sonst} \end{cases} \tag{30}$$

Zusätzlich wurde in dem Clustering-Algorithmus der Gewichtungsvektor w ausgewertet. Dieser enthält für jeden Clustering-Parameter einen Wert zwischen 0.00 und 1.00, der an-

gibt wie stark der Parameter in den Clustering-Vorgang mit einfließt. 0.00 bedeutet, dass der Parameter gar nicht und 1.00, dass der Parameter vollständig berücksichtigt wird.

Der Gewichtungsvektor wird bei jeder Distanzberechnung einbezogen; bei der euklidischen Distanz oder der diskreten Metrik. Zusätzlich zu der normalen Berechnung der Distanz werden die jeweiligen Werte demnach mit dem entsprechenden Wert aus dem Gewichtungsvektor multipliziert.

Beispiel: Nehmen wir an es gibt drei Datenelemente $x_1 = (0.2, 0.28)$, $x_2 = (1.0, 0.33)$ und $x_3 = (0.3, 1.0)$. Des Weiteren ist der Gewichtungsvektor $w = (1.0, 0.2)$ gegeben. Die euklidische Distanz zwischen x_1 und x_2 sowie x_2 und x_3 unter Berücksichtigung von w ergibt:

$$d(x_1, x_2) = \sqrt{1.0 \cdot (0.2 - 1.0)^2 + 0.2 \cdot (0.28 - 0.33)^2} = 0.80 \tag{31}$$

$$d(x_1, x_3) = \sqrt{1.0 \cdot (0.2 - 0.3)^2 + 0.2 \cdot (0.28 - 1.0)^2} = 0.34 \tag{32}$$

Wird der Gewichtungsvektor verändert, so dass $w = (0.1, 1.0)$, so verändern sich die Distanzen wie folgt:

$$d(x_1, x_2) = \sqrt{0.1 \cdot (0.2 - 1.0)^2 + 1.0 \cdot (0.28 - 0.33)^2} = 0.25 \tag{33}$$

$$d(x_1, x_3) = \sqrt{0.1 \cdot (0.2 - 0.3)^2 + 1.0 \cdot (0.28 - 1.0)^2} = 0.72 \tag{34}$$

Durch die Verringerung des ersten und die Erhöhung des zweiten Parameters im Gewichtungsvektor haben sich die Distanzwerte stark verändert. x_1 liegt nun näher an x_2 als an x_3.

Insgesamt wirkt sich diese Gewichtung in weiten Teilen des Clustering-Algorithmus aus. Bei der Anwendung der Silhouetten-Methode werden unter anderem Distanzen von Diskussionsbeiträgen zu Diskussionsbeiträgen des eigenen beziehungsweise nächstgelegenen fremden Clusters berechnet. Beim Quick-Clustering-Verfahren werden Distanzen zwischen Diskussionsbeiträgen und Prototypen sowie Prototypen untereinander bestimmt. Beim Clustering selbst hat die Gewichtung Einfluss auf die Bestimmung der minimalen Distanz eines Diskussionsbeitrages zum nächsten Prototypen, sowie die Distanz eines Prototyps von seiner *alten* zu seiner *neuen* Position, nachdem er verschoben wurde.

Die Werte des Gewichtungsvektors haben Einfluss auf alle wichtigen Schritte des Clustering-Algorithmus. Jede Veränderung des Gewichtungsvektors kann daher die Bestimmung der optimalen Anzahl der Prototypen, sowie die Gruppenzuordnung der Diskussionsbeiträge stark beeinflussen.

Die beiden Screenshots (Abbildung 35 und 36) zeigen, wie sich die Clustering-Ergebnisse verändert haben, nachdem die selbe Diskussionskarte mit verschiedenen Gewichtungsvektoren gruppiert wurden. Im ersten Beispiel wird hauptsächlich der Parameter *Knotentyp* berücksichtigt, woraus drei Cluster entstehen. Im zweiten Beispiel wird nach dem Parameter *Autor* gruppiert, woraus fünf Cluster resultieren. Die Werte der Gewichtungsvektoren entsprechen Zeit, Autor, Knotentyp, Inhalt, Länge des Inhaltes, Struktur und Knotengrad.

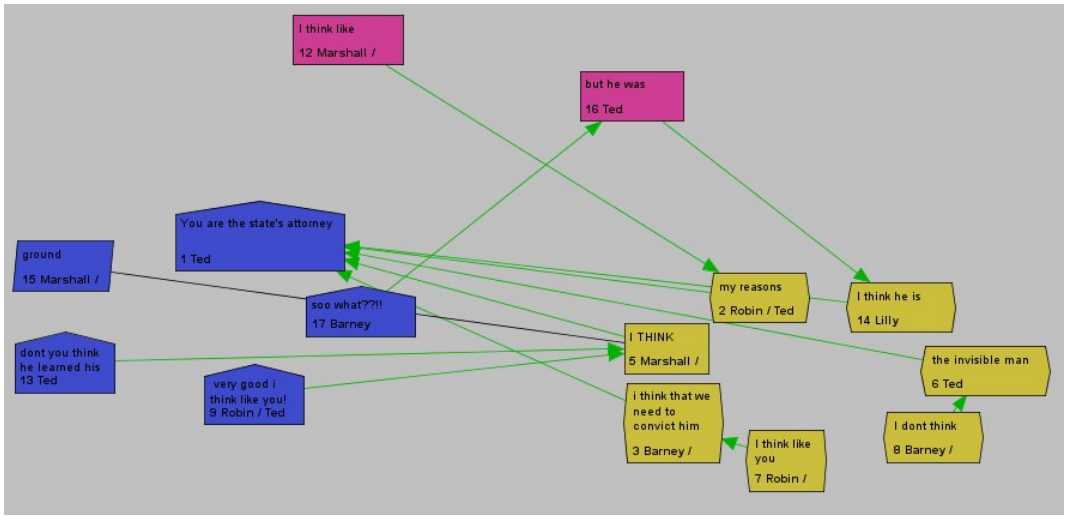


Abbildung 35: Gewichtungsvektor: [0.00, 0.39, 1.00, 0.26, 0.12, 0.60, 0.11]

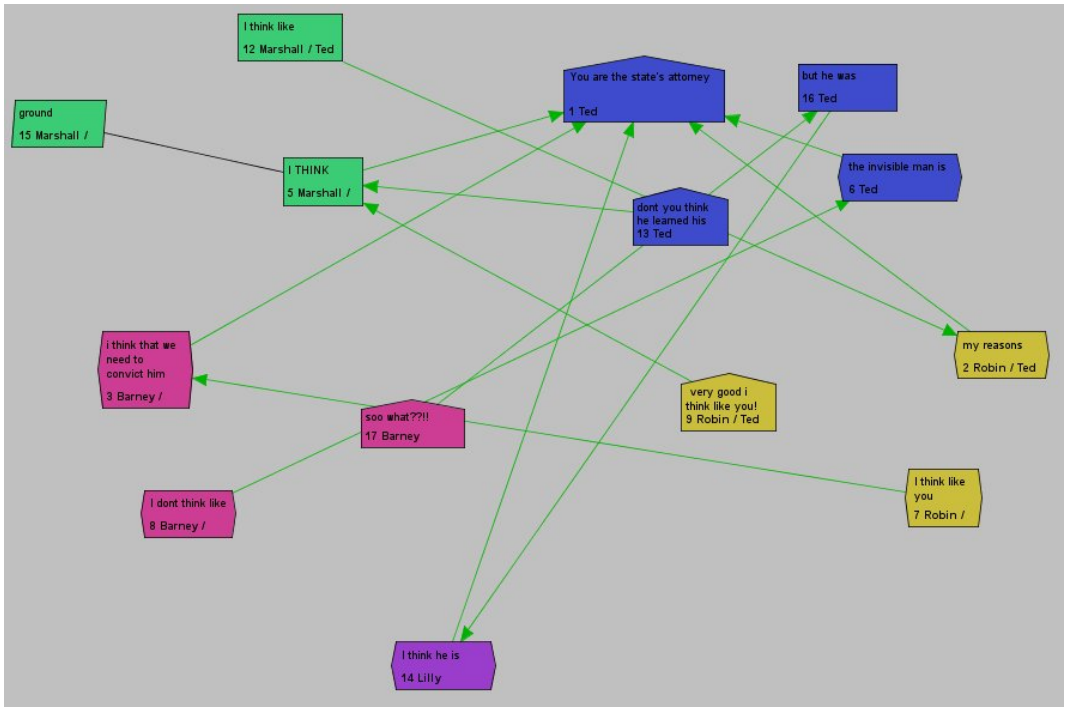


Abbildung 36: Gewichtungsvektor: [0.21, 1.00, 0.13, 0.12, 0.17, 0.00, 0.17]

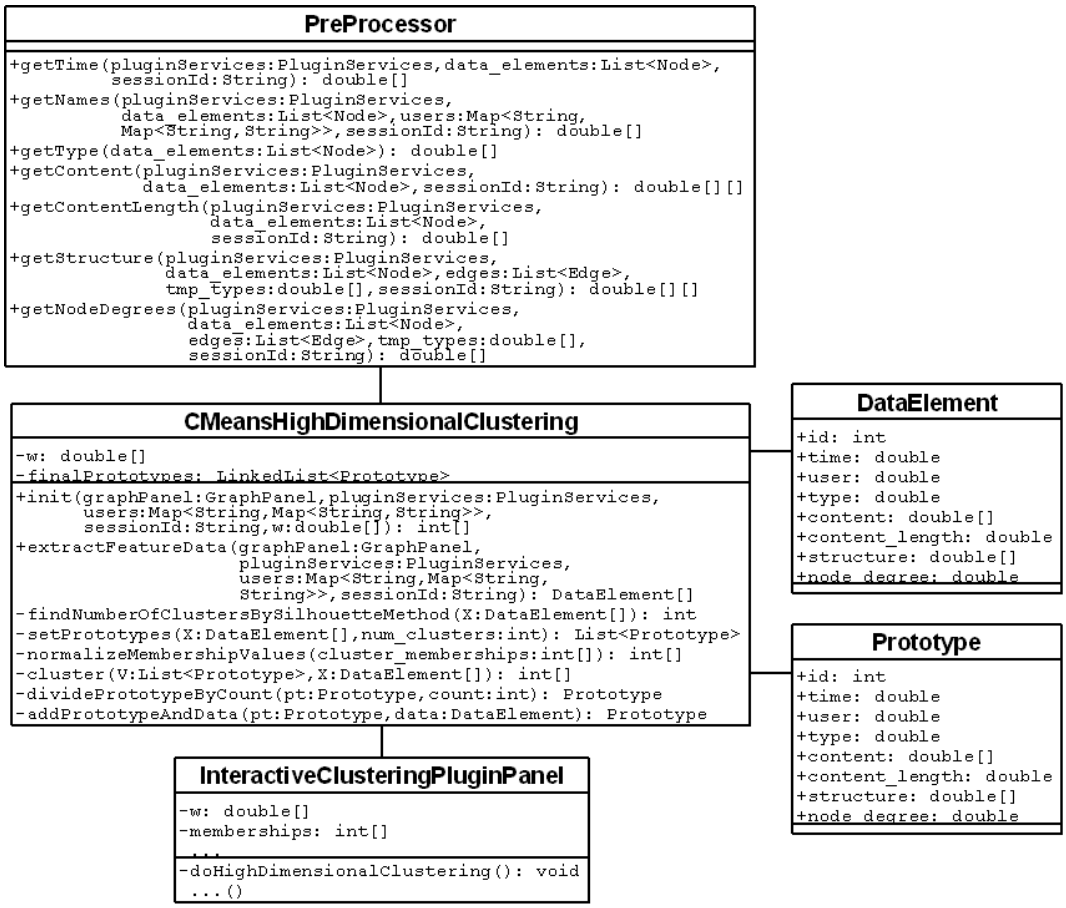


Abbildung 37: Klassenübersicht: *CMeansHighDimensionalClustering* ist die Hauptklasse der Clustering-Anwendung und verbindet alle wichtigen Komponenten.

Die Klasse *CMeansHighDimensionalClustering* implementiert die beschriebenen Funktionen. Durch die Klasse *PreProcessor* liest sie die Clustering-Parameter aus den Diskussionsbeiträgen aus und speichert diese in jeweils einer Instanz der Klasse *DataElement* für jeden Beitrag. Danach erfolgt die Berechnung der Anzahl der Cluster-Prototypen mit der Silhouetten-Methode (*findNumberOfClustersbySilhouetteMethod()*) sowie deren Initialisierung mit dem Quick-Clustering-Verfahren (*setPrototypes()*). Jeder Prototyp wird durch eine Instanz der Klasse *Prototype* repräsentiert.

Anschließend erfolgt das Clustering selbst, wodurch das *Integer*-Array *memberships* entsteht, das die Cluster-Zugehörigkeiten der Knoten repräsentiert. Beispielsweise bedeutet $memberships[2] = 4$, dass der dritte Diskussionsbeitrag (der Index startet bei 0) zum vierten Cluster gehört. Dieses Array wird an die Klasse *InteractiveClusteringPluginPanel* übergeben.

5.5 Visualisierung

Die Klasse *InteractiveClusteringPluginPanel* repräsentiert die grafische Darstellung des Plugins für interaktives Clustering im Moderator's Interface. Sie interpretiert die Benutzerinteraktion und modifiziert je nach Technik und Aktion den Gewichtungsvektor w. Danach wird die Klasse *CMeansHighDimensionalClustering* instanziiert, w an diese übergeben und das Clustering gestartet, wodurch das bereits oben erwähnte Array mit Cluster-Zugehörigkeiten als Ergebnis geliefert wird.

Die Daten dieses Arrays werden so interpretiert, dass die Diskussionsbeiträge beziehungsweise ihr Hintergrund eingefärbt werden. Jedem Cluster wird genau eine Farbe zugeordnet. Somit erhalten die Beiträge, die sich in diesem Cluster befinden, die gleiche Farbe.

Zusätzlich zur Färbung wird die *Springy Physics* Visualisierung, die bereits im Moderator's Interface implementiert war, aufgegriffen und an den aktuellen Anwendungsfall angepasst. Die vorhandene Implementierung sah grundsätzlich vor, dass mit Kanten verbundene Diskussionsbeiträge einander anziehen, während nicht verbundene Beiträge einander abstoßen.

Je weiter zwei miteinander verbundene Beiträge voneinander entfernt werden, desto größer wird die Kraft, mit der sie einander anziehen. Je näher sie zusammen sind, desto schwächer wird diese Kraft. Bei nicht verbundenen Beiträgen ist es genau umgekehrt. Je weiter sie voneinander entfernt sind, desto schwächer ist die abstoßende Kraft und je näher sie zueinander platziert sind, desto größer ist abstoßende Kraft.

Diese Visualisierung wurde so angepasst, dass nicht mehr die Kanten entscheidend für die Berechnung der anziehenden und abstoßenden Kräfte sind, sondern die Gruppenzugehörigkeiten. Diskussionsbeiträge des selben Clusters ziehen einander an, während Beiträge aus unterschiedlichen Clustern einander abstoßen. Auf die ursprüngliche Visualisierung, die auf der Kantenstruktur basiert, übertragen wäre dies so, als wenn jedes Cluster ein vollständig verbundener Subgraph, ist. Denn jeder Diskussionsbeitrag zieht jeden anderen Beitrag des eigenen Subgraphen an, während er alle anderen Knoten wegstößt.

Das Verfahren konvergiert, wenn alle Kräfte so klein geworden sind, dass kein Knoten mehr signifikant seine Position verändert. Dies ist immer dann der Fall, wenn Knoten des

selben Clusters nahe beieinander sind, während die Cluster selbst relativ weit entfernt voneinander sind, sich also kaum abstoßen.

In Abbildung 38 ist links oben die gegebene Anwendung der Springy Physics Visualisierung zu sehen. Rechts oben kommt das Clustering hinzu, wobei Knoten des selben Clusters gleich eingefärbt wurden. Zur Verdeutlichung wurden auch Kanten zwischen Knoten des selben Clusters gezogen, um zu verdeutlichen, dass dort später anziehende Kräfte wirken. Links unten ist zu sehen, wie die Ausgangslage der Visualisierung direkt nach dem Clustering-Vorgang ist. Einige Knoten liegen noch weit von den anderen Knoten des eigenen Clusters entfernt und werden daher stark von diesen angezogen, während sie gleichzeitig stark von den Knoten der anderen Cluster abgestoßen werden. Rechts unten ist die Lage der Knoten stabil. Alle Knoten liegen nahe an den anderen Knoten des eigenen Clusters und relativ weit weg von Knoten der anderen Cluster.

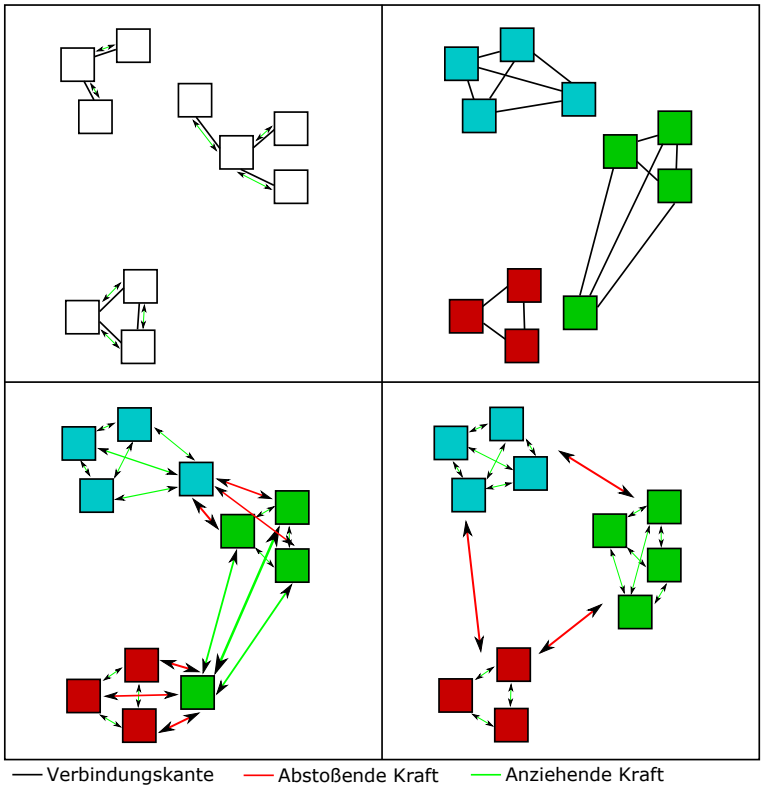


Abbildung 38: Anwendung der *Springy Physics* Visualisierung.

5.6 Verwendete Software

In diesem Abschnitt werden externe Software-Module und -Bibliotheken vorgestellt, die verwendet wurden, um das interaktive Clustering-Verfahren zu implementieren.

5.6.1 Java Frameworks

Zur inhaltlichen Analyse der Diskussionsgraphen wurde das Suchmaschinen-Framework *Apache Lucene* verwendet. Dabei wurden die Lucene-Klassen so eingesetzt, dass aus den

Textinformationen der Diskussionsbeiträge Term-Frequenz-Vektoren generiert werden. Der genaue Ablauf ist wie folgt.

Zunächst wird der Text mit einem Lucene *Analyzer* analysiert. Dabei werden überflüssige Wörter und Satzzeichen entfernt beziehungsweise gekürzt. Der Lucene *StandardAnalyzer* verwendet sogenanntes Stop-Word-Filtering, so dass bedeutungslose (englische) Worte wie *the* und *a* entfernt werden. Zudem werden überflüssige Apostrophe, sowie Punkte von Abkürzungen (Beispielsweise *A.B.C.* wird zu *ABC*) entfernt. Die Klasse *GermanAnalyzer* hat im Prinzip den gleichen Effekt, nur dass eine Stop-Word-Liste verwendet wird, die der deutschen Sprache gerecht wird. Zudem wird ein *Stemming*-Verfahren durchgeführt, bei dem verschiedene Varianten von Wörtern auf ihre Grundform zurückgeführt werden.

Im Moderator's Interface gibt es die Möglichkeit, die Sprache auf *Englisch*, *Deutsch*, *Spanisch* und *Hebräisch* zu stellen. Ändert der Benutzer die Sprache, während das Plugin für interaktives Clustering verwendet wird, so wird auch implizit der Analyzer umgestellt. Da es für *Spanisch* und *Hebräisch* jedoch noch keinen Analyzer gibt, wird in diesen Fällen der *StandardAnalyzer* verwendet. Sollte in Zukunft für diese beiden Sprachen ein geeigneter Analyzer erscheinen, kann dieser zusätzlich in die Anwendung eingebunden werden.

Nachdem der Text eines Diskussionsbeitrages von dem entsprechenden Analyzer prozessiert wurde, wird er in ein Feld einer Instanz der Lucene Klasse *Document* geschrieben. Diese repräsentiert nun den Diskussionsbeitrag und wird in einen temporären Suchindex geschrieben, der jedoch nur im Hauptspeicher der lokalen Maschine angelegt wird. Der Index wird an dieser Stelle nicht zum Suchen verwendet, sondern um die Term-Frequenz-Vektoren aller Diskussionsbeiträge zu berechnen. Nachdem also alle Diskussionsbeiträge im Index vorhanden sind, kann durch den Aufruf einer Methode (*FilterIndexReader.getTermFreqVector(index, feld)*) der Term-Frequenz-Vektor jedes Dokumentes ausgelesen werden. Diese Vektoren werden danach so weiterverarbeitet, dass letztlich die Dokument-Dokument-Matrix entsteht. Der Ablauf folgt dabei dem beschriebenen Vorgehen aus Abschnitt 5.2.

6 Evaluation

In diesem Kapitel wird das in dieser Arbeit implementierte interaktive Clustering-Verfahren auf verschiedene Arten ausgewertet. Zum einen wird eine stichprobenartige Analyse anhand einiger Gewichtungsvektoren und Diskussionsgraphen durchgeführt, um Probleme des Verfahrens aufzuzeigen. Zum anderen wird eine empirische Benutzerstudie vorgestellt, die die Effektivität und Intuitivität des Verfahrens und vor allem der Interaktionstechniken bewerten und vergleichen soll.

6.1 Validierung der Clustering Ergebnisse

Dieser Abschnitt beschäftigt sich mit einer eher statistischen Betrachtung der Clustering-Ergebnisse. Es zeigt einige prinzipielle Probleme des Verfahrens auf. Die Sicht des Benutzers bezüglich der Güte der Clustering-Ergebnisse wird in diesem Abschnitt nicht berücksichtigt.

Wie schon in den Abschnitten 3.3 und 5.4 erläutert wurde, sollte das hier implementierte C-Means-Clustering Verfahren unter den gegebenen Umständen[16] optimale Ergebnisse liefern. Demnach wird basierend auf den zuvor extrahierten Clustering-Parameter, dem Gewichtungsvektor, der Cluster-Anzahl, sowie den ursprünglichen Positionen der Cluster-Prototypen eine Gruppierung berechnet, die den zuvor berechneten maximalen Silhouetten-Koeffizienten besitzt.

Im Folgenden wurden verschiedene realistische Diskussionsgraphen mit jeweils den gleichen Gewichtungsvektoren durch den Clustering-Algorithmus gruppiert. Als Ergebnis erhält man das Clustering sowie den Silhouetten-Koeffizienten, der in diesem Fall eine Aussage dazu trifft, wie sich die statistische Güte des Clusterings unter Veränderung des Gewichtungsvektors auswirkt. Zudem soll festgestellt werden, ob bei bestimmten Gewichtungsvektoren besonders gute oder schlechte Ergebnisse gehäuft auftreten.

Wert	Bedeutung
0.00 bis 0.25	Keine Struktur
0.26 bis 0.50	Schwache Struktur
0.51 bis 0.75	Vernünftige Struktur
0.76 bis 1.00	Starke Struktur

Tabelle 6: Legende für den Silhouetten-Koeffizienten

Folgende Tabellen zeigen, wie sich der Silhouetten-Koeffizient für die Gruppierung eines bestimmten Diskussionsgraphen verändert, wenn der Gewichtungsvektor manipuliert wird. Da es sich hier nur um drei Tabellen handelt, kann dies sicher nicht als umfassende statistische Analyse bezeichnet werden, jedoch sollten sie ausreichen, um wesentliche Merkmale des Verfahrens zu verdeutlichen.

[16] Unter Berücksichtigung von Clustering-Parametern, Gewichtungsvektor, Bestimmung der Cluster-Anzahl C durch den Silhouetten-Koeffizienten, sowie der Verteilung der Prototypen durch das Quick-Clustering-Verfahren.

Gewichtungsvektor	Cluster Anzahl	Silhouetten-Koeffizient
[1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]	2	0.86
[0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0]	5	0.64
[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0]	4	0.71
[0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]	8	0.03
[0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]	2	0.58
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0]	5	0.36
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]	4	0.71
[0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.0]	4	0.52
[1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]	2	0.70
[0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0]	2	0.05
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0]	5	0.35
[0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0]	5	0.44
[0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0]	5	0.44
[0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0]	5	0.28
[0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0]	10	0.29
[1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0]	5	0.44
[0.0, 0.0, 1.0, 0.0, 1.0, 1.0, 1.0]	10	0.22
[0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0]	2	0.12
[0.0, 1.0, 0.0, 1.0, 1.0, 1.0, 0.0]	4	0.07

Tabelle 7: Diskussionsgraph: English_Ben

Gewichtungsvektor	Cluster Anzahl	Silhouetten-Koeffizient
[1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]	2	0.58
[0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0]	5	0.69
[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0]	4	0.75
[0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]	11	0.03
[0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]	2	0.68
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0]	4	0.44
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]	4	0.75
[0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.0]	4	0.58
[1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]	2	0.47
[0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0]	5	0.04
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0]	5	0.34
[0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0]	3	0.33
[0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0]	3	0.33
[0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0]	2	0.33
[0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0]	5	0.20
[1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0]	4	0.34
[0.0, 0.0, 1.0, 0.0, 1.0, 1.0, 1.0]	4	0.25
[0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0]	2	0.09
[0.0, 1.0, 0.0, 1.0, 1.0, 1.0, 0.0]	2	0.10

Tabelle 8: Diskussionsgraph: literature_ron

Gewichtungsvektor	Cluster Anzahl	Silhouetten-Koeffizient
[1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]	2	0.88
[0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0]	3	0.82
[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0]	4	0.76
[0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]	8	0.04
[0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]	2	0.67
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0]	4	0.27
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]	4	0.76
[0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.0]	4	0.55
[1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0]	2	0.69
[0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0]	2	0.08
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0]	2	0.17
[0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0]	4	0.49
[0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0]	9	0.37
[0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0]	3	0.15
[0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0]	4	0.36
[1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0]	5	0.47
[0.0, 0.0, 1.0, 0.0, 1.0, 1.0, 1.0]	4	0.31
[0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0]	10	0.03
[0.0, 1.0, 0.0, 1.0, 1.0, 1.0, 0.0]	2	0.09

Tabelle 9: Diskussionsgraph: biology_experiments2

Aus diesen Kombinationen von Diskussionsgraphen und Gewichtungsvektoren, wurden folgende Beobachtungen gemacht.

Ein Clustering, das den Parameter *Inhalt* mit einbezieht, wird meistens schlecht; dies erkennt man daran, dass der Silhouetten-Koeffizient einen Wert nahe 0.00 erreicht. Daraus kann man schließen, dass keine inhaltliche Struktur in den Daten vorhanden ist. Probleme bei realistischen Diskussionen treten bei diesem Parameter auf, da in der Regel viele verschiedene Wörter jeweils selten auftreten. Dadurch werden die Vektoren für jedes Datenelement sehr hochdimensional und beim Bilden der euklidischen Distanz ergeben sich meist eher hohe Werte, da es mit steigender Dimensionalität unwahrscheinlicher wird, dass hochdimensionale Vektoren ähnlich zueinander sind. Für einen Clustering-Algorithmus wäre es einfacher, wenn einige wenige Schlüsselwörter jeweils sehr häufig auftreten würden, da dies eine Reduzierung der Dimensionalität der Datenvektoren zur Folge hätte und somit leichter Strukturen in den Daten erkannt werden könnten. Ein experimenteller Ansatz, um dieses Problem abzuschwächen, könnte darin bestehen, dass die Textanalyse der Diskussionsbeiträge nicht alle Wörter, sondern nur die häufigsten berücksichtigt.

Eine weitere Beobachtung ist, dass das Clustering umso schlechter wird, je mehr Clustering-Parameter verwendet wurden. Der Silhouetten-Koeffizient wird mit steigender Parameteranzahl immer geringer. Die besten Ergebnisse werden hingegen erzielt, wenn nach einem einzigen oder maximal zwei Parametern gruppiert wird. Die Erklärung zu diesem Problem ist ähnlich wie bei dem Clustering zum Parameter *Inhalt*. Bei steigender Anzahl berücksichtigter Parameter, steigt gleichzeitig die Dimensionalität der Datenelemente und

damit auch die Wahrscheinlichkeit, dass Diskussionsbeiträge nicht zu einem Cluster zusammengelegt werden können. Entweder steigt also die Anzahl der Cluster oder es liegen Beiträge im selben Cluster, obwohl sie nach dem Gewichtungsvektor nicht sehr ähnlich zueinander sind. Beides hat den Effekt, dass der Silhouetten-Koeffizient kleiner wird. Bei Datenelementen mit kleiner Dimension ist es dementsprechend einfacher, ähnliche Elemente und damit Cluster zu finden. Ein Lösungsansatz für dieses Problem könnte in der Analyse des Gewichtungsvektors bestehen, so dass die Clustering-Parameter am höchsten bewertet werden, nach denen am wahrscheinlichsten gruppiert werden soll, während alle anderen Parameter verringert werden.

6.2 Empirische Benutzerstudie des interaktiven Verfahrens

Um eine Bewertung der Interaktionstechniken in Bezug auf die Diskussionsgraphen vornehmen zu können, wurde eine kleine empirische Studie durchgeführt.

Die Studie beschäftigt sich mit dem Forschungsproblem [Die02, S.162], ob mit dem in dieser Arbeit implementierten Gruppierungsverfahren und den gegebenen Interaktionstechniken bestimmte Strukturierungsprobleme von graphbasierten Diskussionen[17] gelöst werden können. Zudem wird untersucht, ob die Clustering-Ergebnisse für den Benutzer hilfreich sind und es ihm erlauben, effizienter Informationen aus einer graphbasierten Diskussion abzulesen.

Dabei wurden Daten erhoben, die erkennen lassen, wie effektiv und intuitiv die implementierten Interaktionstechniken bei der Gruppierung von unstrukturierten Diskussionsgraphen anwendbar sind. Zudem lassen sie Rückschlüsse zu, wie die Techniken in Zukunft verbessert werden können, um die Faktoren Effektivität und Intuitivität zu erhöhen.

Das Adäquationsproblem[18] wurde hier größtenteils vernachlässigt, da nicht der Clustering-Algorithmus und dessen Leistungsfähigkeit, sondern die Interaktion im Vordergrund steht. Daher wurden drei künstliche Diskussionsgraphen bereitgestellt, bei denen im Voraus bekannt war, dass für diese gute Strukturierungsergebnisse erzielt werden können. Die Testdaten sind daher für bestimmte Clustering-Parameter gut strukturiert. Bei einem realistischen Diskussionsgraph könnte das Problem auftreten, dass die Daten für jeden einzelnen der hier betrachteten Clustering-Parameter unstrukturiert sind und keine sinnvolle Gruppierung zulassen. So könnte ein Benutzer bei einem realistischen Graphen auf Probleme stoßen, die nicht zum Forschungsproblem gehören. Wenn der Diskussionsgraph beliebig ungünstig gewählt wird und der Clustering-Algorithmus keine guten Gruppierungen liefern kann, so kann der Benutzer diese Nachteile auch nicht durch gute Interaktionstechniken ausgleichen.

Der Test durch die Benutzer wurde jeweils an dem selben Computer innerhalb des *Fraunhofer IAIS* durchgeführt. Die Benutzer sollten dabei die in dieser Arbeit implementierten Interaktionstechniken dazu verwenden, um bestimmte Gruppierungen von Diskussionsbeiträgen bei einem gegebenen Diskussionsgraphen herbeizuführen. Bei den Benutzern handelt es sich um wissenschaftliche Mitarbeiter oder studentische hilfswissenschaftliche Mitarbeiter der Abteilung *Virtual Environments* des *Fraunhofer IAIS*.

[17] Oder von allgemeinen Graphen

[18] Das Adäquationsproblem beschreibt die Schwierigkeit, wie realistisch die Testdaten einer empirischen Studie sind. Sie treffen daher eine Aussage dazu, wie sehr die Testdaten den Weltausschnitt, in dem sich die Studie bewegt, repräsentieren.

Insgesamt haben zwölf Benutzer im Alter von 23 bis 53 Jahren an dem Testverfahren teilgenommen, davon zwei Frauen und zehn Männer. Der Altersdurchschnitt liegt bei 30.4 Jahren, da die Mehrzahl der Benutzer im Alter von 23 bis 28 Jahren war und es lediglich drei andere Benutzer im Alter von 33, 43 und 53 gab.

Mit Online-Diskussionen beschäftigten sich die meisten der Benutzer eher selten. Von Diskussionsgraphen hatten die meisten noch nie etwas gehört. Fast alle wussten, worum es bei der Cluster-Analyse geht und etwa ein Drittel der Benutzer kannte sogar einige Verfahren zu diesem Thema.

6.2.1 Forschungsdesign

Das Forschungsdesign [Die02, S.168] war folgendermaßen aufgebaut:

Es gibt mehrere Diskussionsgraphen, die zwar künstlich modelliert wurden, jedoch nah an realistischen Diskussionen angelehnt sind. Die Modellierung der Diskussionen bestand darin, dass ich jeweils eine Diskussion mit der Anwendung Digalo erstellt habe und mich danach abwechselnd mit verschiedenen Benutzern angemeldet habe, um die Diskussion zu führen. So konnte ich sicherstellen, dass in den Daten der Diskussionen bezüglich verschiedener Parameter Strukturen vorhanden sind. Diese waren jedoch nicht leicht zu erkennen und die Graphen ähnelten auch optisch (unübersichtlichen) realistischen Diskussionen.

Der Benutzer muss während des Tests so schnell wie es ihm möglich ist bestimmte Aufgaben erfüllen, wobei er den Graphen nach gewissen Kriterien gruppieren soll, um spezifische Informationen aus den Diskussionen zu extrahieren. Die Datensätze der Diskussionsgraphen sind zwar strukturiert, jedoch spiegelt sich dies nicht in der räumlichen Anordnung des Diskussionsgraphen wider, so dass sich die Struktur nicht direkt erkennen lässt. Also muss der Benutzer eine sinnvolle Gruppierung vornehmen.

Der erste Diskussionsgraph stellt eine Debatte mit zwei festen Meinungsgruppen dar, bei der es die Aufgabe des Benutzers ist, die Anhänger der beiden Lager nah beieinander zu platzieren. Die Teilnehmer der Diskussion verleihen ihren Meinungen mit den verschiedenen Kantenarten wie *Zustimmung* und *Widerspruch* Ausdruck. Dementsprechend kann eine Gruppierung nach der *Kantenstruktur* und dem *Knotengrad* die beiden Gruppen gut voneinander separieren.

Diskussionsgraph zwei ist komplexer aufgebaut als der erste Graph. Es ist nicht mehr offensichtlich wie viele Meinungsgruppen es gibt und welche Meinungen die einzelnen Gruppen vertreten. Die Aufgabe des Benutzers ist es diesmal, herauszufinden, wie viele verschiedene Meinungsgruppen es gibt und für welche Meinung diese jeweils stehen. Der Graph ist diesmal inhaltlich gut strukturiert, wodurch ein Clustering nach dem *Inhalt* zur Lösung der Aufgabe beiträgt.

Der dritte Diskussionsgraph wurde so modelliert, dass die Teilnehmer der Diskussion mehrmals unterbrochen wurden, um über den bisherigen Diskussionsverlauf zu reflektieren. Die Aufgabe des Benutzers besteht darin, herauszufinden, wann die Pausen gemacht wurden und welche Diskussionsbeiträge jeweils zu einer Etappe der Diskussion gehören. Diese Aufgabe kann am leichtesten mit einem zeitlichen Clustering gelöst werden.

Zur Gruppierung der Diskussionsbeiträge kann der Benutzer jeweils eine von zwei verschiedenen Interaktionstechniken verwenden - zum einen die *Markierung ähnlicher Diskussionsbeiträge*, im folgenden *Interaktionstechnik 1* genannt, und zum anderen die *Manuelle Modifikation des Gewichtungsvektors*, im folgenden *Interaktionstechnik 2* genannt. Der Benutzer sollte die Aufgaben für die drei Graphen entweder zuerst mit *Interaktionstechnik 1* und danach mit *Interaktionstechnik 2* durchführen oder umgekehrt. Bei jedem Benutzer wurde per Zufall entschieden in welcher Reihenfolge der Test ablaufen sollte, um Sequenzeffekte zu kompensieren. Von jedem der drei beschriebenen Diskussionsgraphen gibt es zwei Versionen, jeweils eine für jeden Testdurchlauf, die sich strukturell, aber nicht inhaltlich, ähneln. Dadurch soll vermieden werden, dass der Benutzer sich beim Test der zweiten Interaktionstechnik an die Ergebnisse des ersten Testdurchlaufs erinnert und die Aufgabe zu leicht lösen kann. Während des Tests werden die Zeiten zur Lösung der Aufgabenstellungen gemessen.

Nachdem der Benutzer / die Benutzerin alle Aufgaben erfüllt hat, füllt er oder sie einen Fragebogen aus, mit dessen Hilfe ermittelt wird, inwiefern es einfach oder schwer war, die Aufgaben zu lösen.

6.2.2 Auswertung

Im folgenden Abschnitt wird der Benutzertest nach verschiedenen Gesichtspunkten ausgewertet. Zunächst werden die statistischen Daten des Fragebogens, danach die Freitext-Antworten und schließlich die während der Tests gemessenen Zeiten vorgestellt und interpretiert.

Tabelle 10 fasst die Ergebnisse des Fragebogens zusammen. Die zwölf Benutzer können in drei Gruppen aufgeteilt werden. Gruppe *1* hat angegeben, lieber die Markierungs-Interaktionstechnik zu benutzen, Gruppe *2* war unentschlossen und Gruppe *3* hätte die Schieberegler zur Lösung der Aufgaben bevorzugt. Die Gruppen unterscheiden sich hauptsächlich bezüglich der Vorkenntnisse der Benutzer.

Die Benutzer aus Gruppe *1* besitzen von allen die größten Vorkenntnisse bezüglich Online-Diskussionen und Cluster Analyse. Diese Benutzer fanden die Interaktionstechnik, mit der sie die Diskussionsbeiträge markieren sollten, leichter zu benutzen.

Dass die Benutzer dieser Gruppe die Markierungs-Interaktionstechnik bevorzugen würden, könnte mit den grundsätzlichen Eigenschaften der Interaktionstechniken zusammenhängen. Um eine sinnvolle Gruppierung mit der Markierungs-Interaktionstechnik vorzunehmen, muss der Benutzer den Graph wesentlich eingehender untersuchen, als dies mit den Schiebereglern der Fall ist. Daher könnte es den Benutzern aus dieser Gruppe einen Vorteil verschafft haben, dass sie bereits Erfahrung mit Online-Diskussionen haben, was es ihnen erlaubt hat, einige zur Lösung der Aufgabenstellungen wichtige Informationen leichter zu erkennen.

Die Benutzer aus Gruppe *2* haben von allen die wenigsten Vorkenntnisse bezüglich Online-Diskussionen, besitzen jedoch solides Grundwissen über Cluster Analyse. Diese Benutzer fanden die Bedienung und die Effektivität der Schieberegler deutlich besser als die der Markierungs-Interaktionstechnik, jedoch waren sie mit den Ergebnissen beider Verfahren gleichermaßen nicht sehr zufrieden. Speziell diese Benutzergruppe gab zusätzlich in ihren

Frage	**1**	**2**	**3**	**Durchschnitt**
Vorkenntnisse (Online-Diskussionen)	0.50	0.31	0.38	0.40
Vorkenntnisse (Diskussionsgraphen)	0.25	0.25	0.17	0.22
Vorkenntnisse (Cluster Analyse)	0.58	0.41	0.33	0.44
Bedienung (Markieren)	0.88	0.63	0.75	0.75
Effektivität (Markieren)	0.69	0.63	0.44	0.58
Erwartungskonformität (Markieren)	0.88	0.63	0.44	0.65
Bedienung (Schieberegler)	0.88	0.75	0.75	0.79
Effektivität (Schieberegler)	0.56	0.75	0.81	0.71
Erwartungskonformität (Schieberegler)	0.56	0.63	0.69	0.63
Bevorzugte Interaktionstechnik (1: Eher Markieren, 0: Eher Schieberegler)	0.75	0.50	0.19	0.48
Gruppierung hilfreich? (1: Sehr hilfreich, 0: Überhaupt nicht)	0.81	0.81	0.75	0.79

Tabelle 10: Ergebnisse des Fragebogens

Freitext-Antworten an, dass die Steuerung der Schieberegler zwar etwas klarer und einfacher war, sie jedoch die Markierungs-Techniken unter bestimmten Bedingungen vorziehen würden.

Diese Aussagen werden später noch eingehender diskutiert, wenn speziell die Freitext-Antworten betrachtet werden.

Benutzergruppe *3* liegt bezüglich ihrer Vorkenntnisse zu Online-Diskussionen zwischen Gruppe *1* und *2*, besitzt jedoch deutlich weniger Vorkenntnisse über Diskussionsgraphen und Cluster Analyse.

Speziell die Kombination von Unerfahrenheit in den Punkten Online-Diskussionen und Diskussionsgraphen könnten in dieser Gruppe entscheidend für das Ergebnis sein. Da die Benutzer dieser Gruppe in diesem Test zum ersten Mal mit einem Diskussionsgraphen konfrontiert wurden und auch sonst eher selten an Online-Diskussionen beteiligt waren, könnte dies die Benutzung der Markierungs-Interaktionstechnik erschweren, da der Benutzer mit dieser Technik deutlich mehr mit dem Diskussionsgraph selbst interagieren muss, als dies mit den Schiebereglern der Fall ist.

In den folgenden Abschnitten wird auf die Freitext-Antworten der Benutzer eingegangen. Dabei wurden besonders die Kommentare betrachtet, die gehäuft auftraten.

Speziell die Personen aus Gruppe *2* haben angemerkt, dass sie in den ersten Versuchen die Interaktionstechnik mit den Schiebereglern besser, jedoch mit fortschreitender Dauer (und Erfahrung) die Markierungs-Interaktionstechnik leichter zu bedienen fanden. Zudem wäre eine Nutzung letzterer Technik sinnvoller gewesen, wenn die dargestellten Informationen des Diskussionsgraphen vollständiger gewesen wären. Es sollte beispielsweise zusätzlich die Entstehungszeit jedes Diskussionsbeitrages angezeigt und die Kantenstruktur deutlicher visualisiert werden. Dieser Aspekt löste bei einigen Benutzern während des Tests Unverständnis aus, da die genannten Parameter zwar in den Clustering-Algorithmus

mit einfließen, dies jedoch nicht unbedingt für den Benutzer ersichtlich war und er somit nur schwer abschätzen konnte, nach welchen Parametern die aktuelle Gruppierung vorgenommen wurde.

Die Benutzer dieser Gruppe hatten also anfangs Probleme mit den Diskussionsgraphen und deren Visualisierung und konnten nicht gut mit der Markierungs-Interaktionstechnik umgehen, da der Benutzer bei dieser Technik direkt mit dem Graphen interagieren muss. Letzteres ist bei den Schiebereglern nicht der Fall, so dass jemand, der sich nicht mit Diskussionsgraphen auskennt, durch das Austesten der Schieberegler leichter zu einem Ergebnis kommt. Daraus schließe ich, dass Benutzer, die Erfahrung mit dem Moderationswerkzeug und Diskussionsgraphen haben, die Markierungs-Interaktionstechnik bevorzugen, da sie den Graph besser überblicken und leichter mehrere Diskussionsbeiträge finden können, die sich als sinnvolles Vorgabe-Cluster verwenden lassen. Die Schiebereglerinteraktionstechnik ist dagegen besser für unerfahrene Benutzer geeignet.

Ein weiterer Aspekt, der von etwa der Hälfte aller Benutzer aus jeglichen Gruppen angesprochen wurde, besagt, dass eine Kombination aus den beiden Interaktionstechniken sinnvoll wäre, da beide individuelle Vor- und Nachteile besitzen, die sich je nach Aufgabenstellung positiv oder negativ auf den Lösungsvorgang auswirken können.

Basierend auf den Erkenntnissen, dass die Markierungs-Interaktionstechnik besser für erfahrene und die Schieberegler-Technik besser für unerfahrene Benutzer geeignet ist, schließe ich auch diesem Kommentar, dass eine Kombination der beiden Techniken die Benutzbarkeit für beide Benutzergruppen verbessern würde. Der Benutzer hätte die Wahl, ob er die Schieberegler benutzt oder die Diskussionsbeiträge markiert. So wäre das Verfahren besser für unterschiedliche Benutzer und Aufgabenstellungen anwendbar.

Ein anderer Kritikpunkt von drei Benutzern aus verschiedenen Gruppen bestand darin, dass der Einfluss der Schieberegler auf den Clustering-Vorgang besonders bei mehreren Parametern schwer nachvollziehbar wirkte.

Dies ist dadurch zu erklären, dass ein Clustering mit dem hier implementierten Verfahren sich nicht sofort ändert, wenn eine kleine Änderung des Gewichtungsvektors vorgenommen wird. Speziell bei Benutzern, die weniger Vorkenntnisse über Cluster Analyse besitzen, löste dies Verwirrung aus, da sie bei den Schiebereglern davon ausgingen, dass sich jede minimale Änderung so auswirkt, dass eine komplett neue Gruppierung vorgenommen wird.

Weitere Kommentare der Benutzer, die hier nicht weiter erläutert werden, da sie meistens nur von jeweils einer Person genannt wurden und / oder an dieser Stelle als nicht so wichtig erachtet wurden, sind im Anhang zu finden. Dort befindet sich ebenfalls der komplette Fragebogen.

Wie bereits in Abschnitt 6.2.1 erwähnt, wurde während des Benutzertests die Zeiten zur Lösung der Aufgaben gemessen. Diese sollten den Eindruck des Benutzers bezüglich seiner Meinung zur Effektivität der Interaktionstechniken unterstützen.

Im Test hat sich dies jedoch nicht bestätigt: In den gemessenen Zeiten sind bezüglich der Präferenz der Benutzer keine Muster zu erkennen. Daraus schließe ich, unter Berück-

sichtigung der bereits erläuterten Folgerungen, dass die Benutzer nicht nach der Bearbeitungsdauer entschieden haben, welche Interaktionstechnik sie besser fanden, sondern danach, wie gut sie mit den Diskussionsgraphen und deren Visualisierung zurecht kamen. Sie entschieden sich demnach entsprechend ihrer Erfahrungen und Vorkenntnisse und nicht aufgrund dessen, dass sie mit einer Technik die Aufgabe schneller gelöst haben.

7 Zusammenfassung und Ausblick

In dieser Abschlussarbeit wurde ein Verfahren zur Unterstützung von Moderatoren von Online-Diskussionen, speziell von graphbasierten Diskussionen, vorgestellt und implementiert. Dieses soll Moderatoren einen besseren Überblick über die von ihnen beobachteten Diskussionen verschaffen und ein schnelleres Ablesen von spezifischen Informationen ermöglichen.

Um diese Ziele zu erreichen, wurde ein Clustering-Algorithmus implementiert, der sieben signifikante Parameter der graphbasierten Diskussionen wie Inhalt, Kantenstruktur oder Knotentyp analysieren und in den Gruppierungsvorgang einbeziehen kann. Zudem wurden verschiedene Interaktionstechniken realisiert, die einem Moderator eine möglichst intuitive Steuerung des Clustering-Algorithmus ermöglichen sollten. Zwei dieser Interaktionstechniken wurden innerhalb einer empirischen Benutzerstudie ausgewertet.

Die empirische Benutzerstudie hat gezeigt, dass die Gruppierung der Diskussionsbeiträge von vielen Benutzern als sinnvoll angesehen wird, um eine bessere Übersicht über eine Diskussion zu erlangen. Zudem wurden die bereitgestellten Interaktionstechniken mit Abstrichen als gut bedienbar und effektiv bewertet. In diesem Abschnitt werden einige Punkte erläutert, die das Verfahren in Zukunft verbessern können.

Der erste Ansatzpunkt besteht darin, das Verfahren, basierend auf den vorhandenen Interaktionstechniken, zu verbessern. Da es je nach Anwendungsfall sinnvoll sein kann, die Schieberegler- oder die Markierungs-Interaktionstechnik zu verwenden, liegt eine Kombination dieser Techniken nahe. Beispielsweise kann der Benutzer mit Hilfe einer Markierung ein Vorgabe-Cluster bestimmen, so dass danach die Werte des Gewichtungsvektors auf die Schieberegler übertragen werden. Danach kann der Benutzer eine Feinjustierung der Parameter über die Schieberegler vornehmen oder das Clustering weiter modifizieren, indem er einzelne Knoten aus den Clustern abtrennt oder neue entstehen lässt.

Ein weiterer Vorteil der Kombination besteht darin, dass der Benutzer nicht mehr auf eine Interaktionstechnik eingeschränkt wird. Er hat beide Techniken zur Verfügung und kann entsprechend seines Vorwissens, seiner Erfahrung und der zu lösenden Aufgabe entscheiden, welche Technik er verwenden will.

Zwei weitere zu verbessernde Punkte sind die Visualisierung und Extraktion der Clustering-Parameter. Speziell bei der Markierungs-Interaktionstechnik ist es unerlässlich, dass der Benutzer vollständige Informationen über die Diskussionsbeiträge des Graphen erhält, damit er diese Technik sinnvoll nutzen kann. Zudem sollte die Extraktion der Parameter verbessert werden. Zusätzlich sollte eine semantische Analyse der Parameter *Autor* und *Knotentyp* durchgeführt werden, damit diese Werte zu Zahlen kodiert werden, die die Abstände der Werte zueinander besser abbilden. Weiterhin sollte die Analyse der inhaltlichen und strukturellen Informationen vertieft werden. Gerade bei diesen Parametern besteht im Hinblick auf die Datenerschließung großes Potenzial.

Des Weiteren könnte die Erwartungskonformität der Interaktionstechniken verbessert werden, indem die Interpretation der Interaktionstechniken modifiziert wird. Bei den aktuell implementierten Interaktionstechniken konnten einige Benutzer in manchen Fällen nicht nachvollziehen, warum bestimmte Gruppierungen entstanden. Hier sollte angesetzt wer-

den, um die Techniken so zu gestalten, dass sie besser auf die Benutzereingabe reagieren. Als Beispiel sei an dieser Stelle das Verschieben der Schieberegler genannt, das so modifiziert werden könnte, dass sich die Gruppierung der Diskussionsbeiträge auch bei relativ kleinen Änderungen der Schieberegler verändert.

Literatur

[Bac96] BACHER, Johann: *Clusteranalyse*. Oldenbourg, 1996

[Die02] DIEKMANN, Andreas: *Empirische Sozialforschung. Grundlagen, Methoden, Anwendungen.* Rowohlt Taschenbuch Verlag GmbH, 2002

[din06] Norm DIN EN ISO 9241-110 August 2006. *Grundsätze der Dialoggestaltung*

[ER60] ERDOS, P. ; RÉNYI, A.: *On the Evolution of Random Graphs*. Publ. Math. Inst. Hungar. Acad. Sci., 1960

[Gas03] GASSNER, Katrin S.: *Diskussionen als Szenario zur Ko-Konstruktion von Wissen mit visuellen Sprachen*. 2003

[HQW06] HEYER, Gerhard ; QUASTHOFF, Uwe ; WITTIG, Thomas: *Text Mining: Wissensrohstoff Text*. W3l, 2006

[Jan03] JANSEN, Dorothea: *Einführung in die Netzwerkanalyse*. Leske + Budrich, 2003

[Koh90] KOHONEN, T.: The self-organizing map. In: *Proceedings of the IEEE* 78 (Sep 1990), Nr. 9, S. 1464–1480

[KSC03] KIRSCHNER, Paul A. (Hrsg.) ; SHUM, Simon J. B. (Hrsg.) ; CARR, Chad S. (Hrsg.): *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*. London, UK : Springer-Verlag, 2003

[LK06] LOTAN-KOCHAN, E.: Analyzing Graphic-Based Electronic Discussions: Evaluation of Students' Activity on Digalo. In: *Proceedings of EC-TEL*. Crete, 2006, S. 652–659

[Mik07] MIKŠÁTKO, Jan: *Using Machine Learning Techniques to Analyze and Recognize Complex Patterns of Student E-Discussions*, Charles University, Prague, Diplomarbeit, 2007

[PB81] PLATT, Stephen M. ; BADLER, Norman I.: Animating Facial Expressions. In: *SIGGRAPH Comput. Graph.* 15 (1981), Nr. 3, S. 245–252

[Pet62] PETRI, Carl A.: *Kommunikation mit Automaten.* Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962

[Run00] RUNKLER, Thomas A.: *Information Mining. Methoden, Algorithmen und Anwendungen intelligenter Datenanalyse*. Vieweg Verlagsgesellschaft, 2000

[RWA+08] ROSÉ, C. P. ; WANG, Y. C. ; ARGUELLO, J. ; STEGMANN, K. ; WEINBERGER, A. ; FISCHER, F.: Analyzing Collaborative Learning Processes Automatically: Exploiting the Advances of Computational Linguistics in Computer-Supported Collaborative Learning. In: *International Journal of Computer-Supported Collaborative Learning* 3 (2008), September, Nr. 3

[SWCP95] SUTHERS, Daniel ; WEINER, Arlene ; CONNELLY, John ; PAOLUCCI, Massimo: Belvedere: Engaging Students in Critical Discussion of Science and Public Policy Issues. In: *In AI-Ed 95, the 7th World Conference on Artificial Intelligence in Education*, 1995, S. 266–273

[TKS05] TAN, Pang-Ning ; KUMAR, Vipin ; STEINBACH, Michael: *Introduction to Data Mining*. Addison Wesley, 2005

[Tou69] TOULMIN, Stephen E.: *The uses of argument*. Cambridge : Cambridge university press, 1969

[Wat99] WATTS, Duncan J.: *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton, NJ : Princeton University Press, 1999

[Web02] WEBB, Andrew: *Statistical Pattern Recognition*. Wiley & Sons, 2002

[WF94] WASSERMAN, Stanley ; FAUST, Katherine: *Social Network Analysis*. Cambridge University Press, 1994

[Zel94] ZELL, Andreas: *Simulation Neuronaler Netze*. Addison-Wesley, 1994

A Empirische Benutzerstudie

A.1 Freitext-Antworten

Dieser Abschnitt fasst die Kommentare der Benutzer zusammen, die nach der empirischen Benutzerstudie in die Freitext-Antwortmöglichkeit geschrieben wurden.

Kommentare zu den Interaktionstechniken:

- Mit mehr Trainingszeit und Informationen (Zeit, Inhalt, ...) zum Graphen wäre Markieren effektiver
- Je nach Anwendungsfall / Aufgabe Schieberegler oder Markieren besser einsetzbar (individuelle Vor- und Nachteile)
- Kombination aus Schieberegler und Markieren wäre hilfreich (Zum Beispiel Werte von Markierung auf Schieberegler übertragen)
- Einfluss der Schieberegler manchmal schwer nachvollziehbar bei mehreren Parametern
- Markieren sehr gewöhnungsbedürftig
- Bei Schieberegler hat man eine bessere Kontrolle über die Clustering-Parameter
- Besseres Feedback der Markierungs-Interaktionstechnik (Wonach wurde gruppiert?)

Kommentare zum Moderationstool:

- Zu wenig Informationen der Diskussionsbeiträge sichtbar (zu wenig Text, keine Zeitangabe, jedoch überflüssige ID)
- Fenster für Graph ist zu klein, besonders wenn Schieberegler benutzt wird
- Formen der Diskussionsbeiträge sehr gewöhnungsbedürftig
- Moderator sollte deutlicher gekennzeichnet werden
- Bessere Kantenvisualisierung (Welcher Knoten ist mit welchen anderen Knoten verbunden?)

A.2 Fragebogen

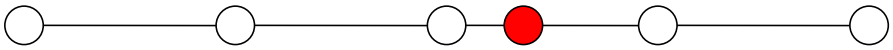
Alter: _____ **Geschlecht:** _____

Fragebogen - Dynamic Real-time Interactive Clustering

Vorkenntnisse

1) Wie schätzen Sie Ihre Erfahrung mit Online-Diskussionen ein?

Durchschnittswert: 0.40, Standardabweichung: 0.39



Regelmäßige Nutzung Seltene Nutzung Keine Erfahrung

2) Wussten Sie vor diesem Test bereits etwas über Diskussionsgraphen und / oder haben diese sogar schon aktiv genutzt?

Durchschnittswert: 0.22, Standardabweichung: 0.16

Vorwissen vorhanden & Aktive Nutzung | Vorwissen vorhanden | Schonmal gehört | Noch nie gehört

3) Wie schätzen Sie Ihr Wissen zur Cluster-Analyse und deren Verfahren ein?

Durchschnittswert: 0.44, Standardabweichung: 0.25

Experte | Grundwissen & einige Verfahren bekannt | Grundwissen | Noch nie gehört

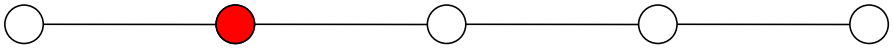
Benutzer-Test

[Interaktionstechnik 1 = Markieren]
[Interaktionstechnik 2 = Schieberegler]

4) Wie fanden Sie die Bedienung von

(a) Interaktionstechnik 1?

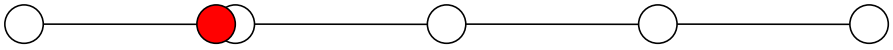
Durchschnittswert: 0.75, Standardabweichung: 0.18



Sehr einfach Sehr schwierig

(b) Interaktionstechnik 2?

Durchschnittswert: 0.79, Standardabweichung: 0.17

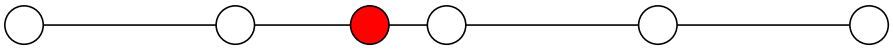


Sehr einfach Sehr schwierig

5) Wie gut konnten die Aufgaben gelöst werden

(a) mit Interaktionstechnik 1?

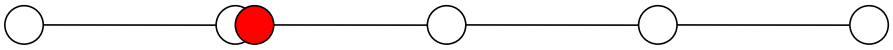
Durchschnittswert: 0.58, Standardabweichung: 0.21



Sehr effektiv Sehr ineffektiv

(b) mit Interaktionstechnik 2?

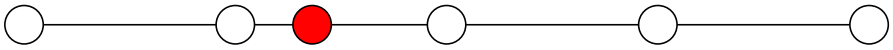
Durchschnittswert: 0.71, Standardabweichung: 0.22



Sehr effektiv Sehr ineffektiv

6) Interaktionstechnik 1: Hat das System auf die Eingabe so reagiert (Gruppen bilden) wie von Ihnen erwartet?

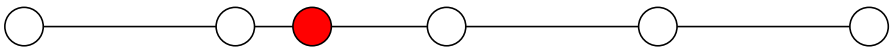
Durchschnittswert: 0.65, Standardabweichung: 0.22



Erwartungsgemäß Nicht erwartungsgemäß

7) Interaktionstechnik 2: Hat das System auf die Eingabe so reagiert (Gruppen bilden) wie von Ihnen erwartet?

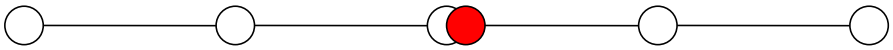
Durchschnittswert: 0.63, Standardabweichung: 0.24



Erwartungsgemäß Nicht erwartungsgemäß

8) Welche der Interaktionstechniken würden Sie eher zur Lösung der Aufgabenstellung bevorzugen?

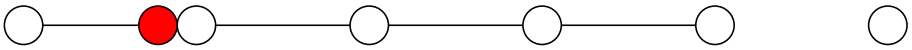
Durchschnittswert: 0.48, Standardabweichung: 0.24



Interaktionstechnik 1 Interaktionstechnik 2

9) Wie sehr hilft Ihnen die Gruppierung der Diskussionsbeiträge dabei, eine bessere Übersicht / besseres Verständnis über die Diskussion zu erlangen?

Durchschnittswert: 0.79, Standardabweichung: 0.09



Sehr Überhaupt nicht Weiß nicht

10) Freitext-Antwortmöglichkeit für weitere Anmerkungen, Ideen und Empfehlungen. Was war besonders gut oder schlecht? Was könnte man verbessern?