Hybrides Clustering mit Rogers-Tanimoto- und Kosinusdistanz

Siegfried Gessulat & Simon Schwarzmeier
Betreuer: Gunar Maiwald

Seminar **Text Mining**

Zusammenfassung Wir möchten Cluster in einem Graphen aus mathematischen Artikeln finden. Die Idee ist Informationen aus zwei verschiedenen Blickwinkeln zu nehmen: Einmal dem Zitationsgraph und auf der anderen Seite die Zusammenfassungen der Artikel. Durch eine Mischung der Beiden hoffen wir ein besseres Clustering zu bekommen. Zusätzlich benutzen wir h-cores im Graphen, ein Verfahren dass auf dem h-index basiert. Durch diese Methode bestimmen wir die Anzahl der zu findenden Cluster und eine Initialbelegung für den verwendeten Clusteralgorithmus. Mit dieser hoffen wir deterministisch ein Optimum zu finden, was ohne die Belegung nur randomisiert möglich war.

1 Einleitung

Um Sammlungen von wissenschaftlichen Artikeln nutzbar zu machen, ist es sinnvoll die Artikel zu kategorisieren. Es ist daher nur möglich Sammlungen manuell zu kategorisieren, die nicht zu viele Dokumente beinhalten, sodass der Aufwand gerechtfertigt bleibt. Ist die Sammlung zu groß, muss die Kategorisierung automatisiert werden.

In dieser Ausarbeitung betrachten wir einen Datensatz von wissenschaftlichen Artikeln der - mit ca. 700 Tausend Dokumenten - zu groß für eine manuelle Kategorisierung ist. In einem solchen Fall, benutzt man Clusteringverfahren um eine Kategorisierung automatisch vorzunehmen. Ein Beispiel für ein solches Verfahren ist k-means Clustering.

In dieser Sektion wird die Problemstellung und die Besonderheiten des vorliegenden Datensatzes beschrieben. Außerdem wird die Wahl des Clusteringverfahrens erläutert und eine neue Methode zur Initialisierung des gewählten Verfahrens vorgeschlagen. Sektion 2 beschäftigt sich im Detail mit dem Aufbau des Experimentes. Danach werden in Sektion 3 die Ergebnisse des Experiment vorgestellt. Zu letzt (Sektion 4) folgt die Interpretation und ein Ausblick.

1.1 Motivation

Der Datensatz besteht aus 720.752 mathematischen wissenschaftlichen Artikeln in verschiedenen Sprachen. Allerdings ist der überwiegende Teil der Artikel (über 90%) auf Englisch verfasst. Die Daten stammen aus der Datenbank

von CiteSeer^{X}[1] und beinhalten zu jedem Dokument ein kurze Zusammenfassung (Abstract) und das Literaturverzeichnis (References) des Dokuments.

1.2 Hybrides Clustering

Üblicherweise spannen die Features eines Datenpunktes einen Raum für ein Clusterverfahren auf. In diesem Featureraum berechnet das Clustering Distanzen zwischen Punkten um anhand dessen eine Fehlerfunktion zu minimieren.

Der CiteSeer^{X} Datensatz gibt uns allerdings Informationen zu zwei unterschiedlichen Featureräumen - ein Featureraum über den Abstracts und ein Featureraum über den Referenzen.

1.2.1 Featureräume

- Die Abstracts liegen als unstrukturierte Textdaten vor. Ein klassisches Modell für Ähnlichkeiten oder Distanzen zwischen Textdaten ist das Vector Space Model (VSM), das seinen Ursprung im Information Retrieval hat.[1] Eine Distanz zweier Dokumente gibt informell die Ähnlichkeit der beiden Texte an. Sie beantwortet die Frage: Werden Wörter ähnlich verwendet? Die verwendete Distanz in diesem Raum ist die Kosinus-Distanz.
- Die Referenzen bilden einen gerichteten Graphen mit den Dokumente als Knoten und die Zitationen als Kanten: Ein Zitationsgraph. Die Distanz zwischen zwei Knoten im Zitationsgraphen gibt die Ähnlichkeit der Beiden Dokumente in Bezug auf gleiche Quellenverwendung an. Sie beantwortet die Frage: Werden ähnliche Quellen verwendet? Die Distanz in diesem Raum ist die Rogers-Tanimoto-Distanz

Eine formale Betrachtung beider Räume und des jeweiligen Distanzmaßes folgt in Sektion 2.

Würde man Dokumente per Hand kategorisieren, würden sowohl die Referenzen als auch die Abstracts die Entscheidung welche Zuordnung getroffen wird beeinflussen. Beide Informationsquellen haben einen Einfluss auf die Kategorisierung. Im vorgestellten Experiment wird eine Möglichkeit des Zusammenführens der Beiden Featureräume und deren Auswirkung auf die Güte des Clusterings untersucht. Da Featureräume aus zwei Informationsquellen betrachten, sprechen wir von hybridem Clustering.

Der Fokus des Experiments liegt auf den Auswirkungen, die die Verwendung von zwei Informationsquellen hat. Um den Versuchsaufbau nicht zusätzlich zu erschweren wird zum Clustern ein möglichst einfaches Verfahren verwendet. Allerdings können wir einige Algorithmen nicht verwenden: zum Beispiel k-means. Viele Clusteralgorithmen setzen einen euklidischen Raum vorraus.

[1] http://csxstatic.ist.psu.edu/about

1.2.2 k-medoids Die vorliegenden Featureräume sind nichteuklidisch. Ein Grund ist, dass ausschließlich die gegebenen Dokumente zugelassene Punkte im Raum sind. Punkte die nicht durch ein Dokument repräsentiert sein, müssten eine „durchschnittliches“ Abstract und ein „durchschnittliches“ Literaturverzeichnis haben. So ein durchschnittlicher Zusammenhang zu allen anderen Dokumenten ist für Textdokumente allerdings nicht definiert. Daraus ergibt sich, dass die Verschiebung eines Clusterzentrums im Raum ist nicht möglich wäre. Deshalb verwenden wir eine Variante von k-means für nichteuklidische Featureräume: k-medoids.

Als Eingabe bekommt k-medoids eine Matrix mit den Distanzen paarweise aller Dokumente. Zusätzlich werden k Dokumente als prototypische Clusterzentren, d.h. als Initialbelegung markiert.

Eine Schwäche von k-means und k-medoids ist, dass der Algorithmus nicht zwingend das globale Optimum, also das optimale Clustering findet. Zum Einen ist es möglich, dass der Algorithmus bei einem nicht-konvexen Problem in lokalen Optima hängen bleibt. Zum Anderen ist es auch bei konvexen Problemen möglich, dass der Algorithmus um das globale Optimum oszilliert, und abbricht, weil keine Verbesserung mehr stattgefunden hat.

Um dieses Problem zu umgehen, lässt man k-medoids deshalb mehrfach mit unterschiedlichen Initialbelegungen laufen. Die Initialbelegungen werden dabei typischerweise zufällig gewählt. Nach mehreren unterschiedlich initialisierten Durchläufen, wird dann das Beste bisherige Ergebnis ausgegeben und alle anderen Ergebnisse verworfen.

1.2.3 h-cores Für jede zufällige Initialisierung, muss k-medoids eine neue Optimierung durchführen. Es wäre daher eine intelligentere Initialisierungsmethode wünschenswert, die zu ähnlichen Ergebnissen führt, wie das beste Ergebnis der zufälligen Initialisierung.

Diese Ausarbeitung stellte eine solche Methode vor - sie basiert auf der Berechnung von Kerndokumenten, sogenannten h-cores. Die zusammengeführten Distanzen werden dafür als Indiz für die stärke der Vernetzung der Dokumente aufgefasst. Daraus werden dann die am stärksten vernetzten Dokumente ermittelt und für die Initialisierung verwendet.

In Sektion 2 wird die Methode formal vorgestellt und in Sektion 3 wird sie mit dem besten Ergebnis aus mehreren zufälligen Initialisierungen verglichen.

2 Experimentaufbau

Diese Sektion beschreibt den Experimentaufbau mit einem besonderen Augenmerk auf dem Preprocessing. Abbildung 1 gibt eine Übersicht über den gesamten Experimentaufbau.

Zunächst werden die Rohdaten im Preprocessing Schritt eingelesen und für eine weitere Verarbeitung vorbereitet. Zu dem genauen Prozess mehr in Sektion 2.4. Danach wird für jeden Featureraum je eine Distanzmatrix erzeugt. Für k-medoids müssen diese danach zu einer Distanzmatrix zusammengeführt werden.

Aus der resultierenden Matrix lassen sich dann die h-cores berechnen, die zur Initialisierung von k-medoids dienen sollen (siehe 2.1). Im Folgenden wird der Prozess im Detail betrachtet.

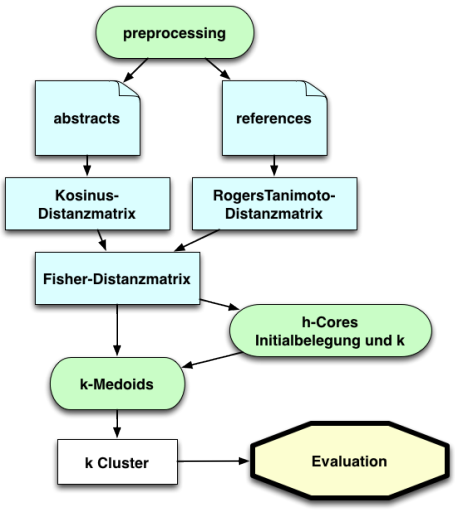

Abbildung 1. Experimentaufbau. Die einzelnen Schrittte werden in den folgenden Teilen näher behandelt.

2.1 Clustering

An dieser Stelle soll noch kurz der praktische Aspekt des Clusterings beleuchtet werden. Der Algorithmus, k-mediods, ist dem Leser bereits ein Begriff. Hat man die Distanzmatrix plus eine Initialisierung der Clusterzentren berechnet, kann dieser ohne weitere Zwischenschritte durchgeführt werden. In dieser Arbeit wurde PyCluster[2] verwendet. Bei der Größe unserer Distanzmatrix konnte ein Durchlauf in weniger als einer Minute terminieren. Das ist Interessant, da dies nur ein unterproportionalen Teil der gesamten Rechenzeit beansprucht. Den hauptsächliche Rechenaufwand macht die Berechnung der Distanzen sowie deren Zusammenführung aus.

2.2 Distanzen

k-Medoids clustert gegebene Daten anhand von Distanzen. Dazu wird für jedes Datenpunktpaar eine Distanz benötigt. In vorgestellten Fall ist ein Datenpunkt ein Dokument. Dieses Dokument liegt in zwei unterschiedlichen Featureräumen:

[2] PyCluster, Python-Bindings für die C-Clustering Library

Dem Zitationsgraphen und dem Vector Space Model. Deshalb werden zunächst die Distanzen in jedem Featureraum mit einem geeigneten Distanzmaß berechnet und danach durch eine Interpolationsmethode zusammengeführt. So ist es möglich mehrere unterschiedliche Mischungen zu betrachten um zu evaluieren welche Mischung der Featureräume das bestes Ergebnis erzielt.

2.2.1 Kosinusdistanz Die Kosinusdistanz ist ein räumliches Abstandsmaß im Vector Space Model. Sie wird hier benutzt um die Distanz zweier Abstracts voneinander zu bestimmen. Die Kosinusdistanz im Vector Space Model stammt aus dem Information Retrieval und ist das klassische Beispiel der Distanzberechnung zwischen zwei unstrukturierten Texten.

Zunächst muss dafür der verwendete Wortschatz aus der Dokumentengesamtheit ermittelt werden. Danach wird jedes Abstract als Vektor dargestellt - Die Länge des Vektors ist gleich, der Wörter im Wortschatz. Der Vektor stellt da wie häufig ein Wort des Wortschatzes im jeweiligen Abstract verwendet wird. Dies nennt man die Term-Frequenzdarstellung. Es sind auch andere Darstellungen möglich, die Untersuchung komplizierterer Darstellungen würde den Rahmen dieser Ausarbeitung allerdings sprengen.

Die Kosinusdistanz ist nun der Kosinusabstand zweier Abstract-Vektoren d_i, d_j zueinander. Formal:

$$sim_{VSM}(d_i, d_j) = \cos(d_i, d_j) = \frac{d_i \cdot d_j}{|d_i||d_j|}$$

Sind sich die Beiden Abstracts sehr ähnlich, d.h. sie verwenden die gleichen Wörter ähnlich Häufig, geht die Kosinusdistanz gegen 1 - im entgegengesetzen Fall gegen 0.

2.2.2 Rogers-Tanimoto-Distanz Die Rogers-Tanimoto-Distanz ist ein Distanzmaß zwischen zwei Dokumenten in einem Zitationsgraphen. Deshalb wird es in dieser Ausarbeitung für den Featureraum des Zitationsgraphen verwendet.

Die Rogers-Tanimoto-Distanz gibt das Verhältnis zwischen der Anzahl der Dokumente die von Beiden Dokumenten d_i, d_j zitiert werden und der Anzahl der Dokumente, die jeweils nur von einem der Beiden Dokumente zitiert werden an. Formal:

$I = \{x \mid d_i \text{ zitiert } x\}$
$J = \{x \mid d_j \text{ zitiert } x\}$

$$sim_{RT}(d_i, d_j) = \frac{|I \cap J|}{|I| + |J| - |I \cap J|}$$

Wenn zwei Dokumente die gleichen Dokumente im Literaturverzeichnis haben, sie also die genau die gleichen Dokumente zitieren und keine anderen, dann ist die Rogers-Tanimoto-Distanz 1. Wenn die Beiden Dokumente kein Dokument gemeinsam zitieren, dann ist die Distanz 0.

2.2.3 Zusammführung zweier Distanzmaße Zum Zusammenführen der beiden Distanzmaße wird Fisher's Inverse χ^2-Methode verwendet. Diese Methode interpoliert zwei Distanzmaße und bildet die neue Distanz auf dem Intervall $[0, 1]$ ab. Ihr Einsatz wurde von Glänzel, unter anderem, in [2] diskutiert. Sie ist wie folgt definiert:

$$sim_{fisher}(sim_{RT}, sim_{VSM}) = \cos(\lambda \cdot \arccos(sim_{RT}) + (1 - \lambda) \cdot \arccos(sim_{VSM}))$$

Der Parameter λ liegt in $[0, 1]$ und welches Distanzmaß mehr zum resultierenden Distanzmaß beitragen soll. Der Parameter λ ist frei wählbar. Es wurden die Werte 0.25, 0.50% und 0.75 für diesen Parameter evaluiert.

2.3 Die h-cores

Die h-cores wurden von Glänzel in [3] vorgestellt. Sie bilden eine auf dem Hirsch-Index basierende Methode für uns gute Prototypen für die Clusterzentren zu approximieren. Nicht-relevante, bzw. zu schwache Kanten in einem z.B. durch eine Distanzmatrix gegebenen Graphen werden in der Berechnung zunächst herausgefiltert, wenn sie nicht mindestens mit r gewichtet ist. Auf dem dadurch gefluteten Graphen werden dann die Knoten extrahiert, die die Cores darstellen. Formal kann das Vorgehen wie folgt dargestellt werden:

Def.: Core-Dokumente Core-Dokumente sind die Knoten, die mit minimal $n > 0$ Knoten mit einem Gewicht(Distanz/Ähnlichkeit) $r \in [0, 1]$ verbunden sind.

Vorraussetzung für die Definition ist, dass sie auf den gefluteten Graphen Graphen angewandt wird. Herauszustellen ist an dem Verfahren der *freie Parameter* r und dass dessen Wahl direkt die Anzahl der Clusterzentren bestimmt.

2.4 Preprocessing

Damit die später präsentierten Ergebnisse nachvollziehbar bleiben, ist es wichtig unseren Umgang mit den gegebenen Daten deutlich zu machen. Aus diesem Grund soll hier das konkrete Vorgehen beleuchtet werden. Wie bereits erwähnt beinhaltet dieser Schritt mehrere Teilschritt, die in folgender Reihenfolge durchgeführt wurden:

1. Einlesen und Trennen der rohen Daten
2. Getrenntes Filtern der Daten
3. Sampling
4. Stemming der Abstracts

Abbildung 2 zeigt die einzelnen Schritte auf einem Blick.

Ergebnis des Preprocessing sind drei getrennte Dateien. Eine, welche die gestemmten Abstracts beinhaltet und passend dazu ein Dokument, welches als Liste alle Wörter beinhaltet, die in allen Abstracts zusammen vorkommen. Sie wird für die Berechnung der Kosinusdistanzen benötigt. Der dritte Output ist eine Datei mit den Referenzen per Dokument, welche die Basis zur Berechnung der Tanimoto-Distanzen ist.

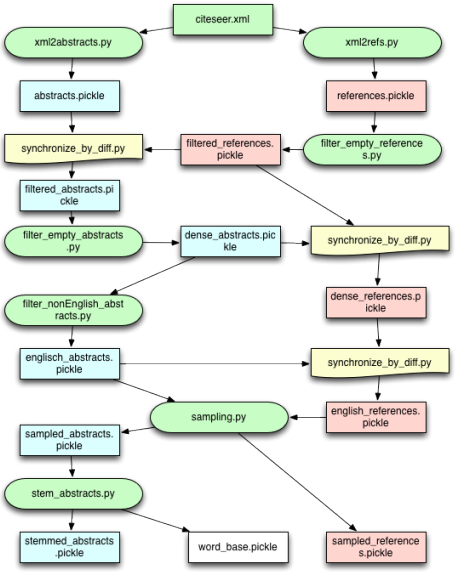

Abbildung 2. Preprocessing. Die Runden Elemente im Diagram sind Skripte welche in der durch die Pfeile bestimmten Reihenfolge abgearbeitet werden. Die Ergebnisse der Skripte sind Dateien (Rechtecke im Diagram) welche in der Abbildung für die Abstracts blau und für die References lachsfarben eingefärbt sind. Jedes mal, wenn für einen der beiden Zweige Artikel ausgesiebt werden, wird der zweite mit einem Synchronations-Skript auf den selben Stand gebracht. Ganz unten sieht man die drei Ergebnisdateien des Preprocessing-Schritts.

2.4.1 Einlesen und Trennen der Daten Der erste Schritt ist das Einlesen der Daten. Sind die Abstracts und Zitaltionslisten aus dem XML-Dokument extrahiert kann mit der eigentlichen Bearbeitung begonnen werden. Bei größeren Dokumenten, wie dem in unserem Beispiel vorhandenen, lohnt es sich nicht einen herkömmlichen XML-Parser zu nutzen. Grund ist, dass diese meist versuchen, das gesamte Dokument im Speicher abzubilden. Die alternative sind Event-basierte XML-Parser, die sich an der Simple API for XML (SAX) orientieren, wie z.B. die in diesem Projekt genutzte Python-Module der Standardbibliothek[3] . Dabei bindet man Funktionen an relevante Tags in denen dann die Informationen im Tag heraus gezogen werden. Der Aufbau der genutzten XML-Datenbank ist simpel. Es gibt pro Artikel einen record, welcher jeweils das Abstract als *description* beinhaltet und die Referenzen als *relation*.

2.4.2 Filtern Ein weiterer Schritt ist die Qualität des Datensatzes sicher zu stellen, indem unpassende Dokumente aussortiert werden. Als unpassend angesehen werden, zum Einen Dokumente mit nicht vorhandenen Abstracts, oder

[3] xml.sax

Zitationslisten. Der Grund hierfür ist, dass das Hauptinteresse die Zusammenführung zweier verschiedener Distanzmaße ist. Hat man also Distanzen, bei denen Teile der Datenbank nur mit einem der Distanzmaße miteinbezogen würden, könnte dies die Ergebnisse schon im Vornherein auf negative Weise beeinflussen. Das führt im schlimmsten Fall zu falschen Schlüssen was den gewonnenen Vorteil einer hybriden Nutzung von Distanzmaßen betrifft.

Bei der Distanzberechnung im VSM geht man davon aus, dass sich alle Abstracts in einem gemeinsamen Raum befinden. Englische und deutsche Dokumente lassen sich beispielsweise schlecht auf Ähnlichkeit prüfen. Wir begrenzen unsere Untersuchung auf englischsprachige Dokumente. Um Dokumente in anderen Sprachen zu eliminieren kann etwa eine einfache Bibliothek wie *guess-language*[4] für Python zum Einsatz kommen. Die Spracherkennung funktioniert schnell und zuverlässig. So hat es in dem hier beschriebenen Experiment ca. 90 Minuten gedauert, bis alle zu dem Zeitpunkt noch vorhandenen Dokumente zu klassifizieren. Es stellt im Arbeitsablauf also keinen Flaschenhals dar.

Selbstverständlich muss man bei jedem Siebvorgang aufpassen, die Dokumentenbasen beider Zweige konsinstent zu halten, indem die gelöschten Dokumente auch in dem zweiten Zweig entnommen werden. In dem Beispielprojekt, übernahm das ein sync-Skript, wie in Abbildung 2 zu sehen(*synchronize_ by_ diff.py*).

2.4.3 Sampling Praktisch macht es in unserem Kontext keinen Sinn zu versuchen den gesamten gefilterten Datensatz zum Clustern zu nutzen, da dieser spätestens bei der Distanzberechnung zu groß ausfallen würde. Die berechnete Distanzmatrix wäre zu groß für den Arbeitsspeicher normaler Rechner, wenn man das zu Anzahl der Artikel quadratische Wachstum dieser betrachtet, gilt das auch für größer bemessene Systeme. Aus diesem Grund wurde im Experiment nur mit Teilmengen der Daten gearbeitet. Die gewählte Größe betrug zehn Tausend der Artikel - ca. 1% der Daten. Jedoch kommt dabei die Frage auf, wie die Artikel ausgewählt werden, da man eine möglichst repräsentative Teilmenge der Artikel clustern möchte.

Hier wurde sich entschieden zwei verschiedene Samplingmethoden zu nutzen.

random Zum einen wurde eine mit zufällig ausgewählten Artikeln gefüllte Teilmenge genutzt. Diese Methode ist sehr simpel, wenn auch naiv, da man nicht wissen kann ob die so entnommenen Daten repräsentativ sind. Hier wurde diese Methode vor allem gewählt um die Ergebnisse mit der zweiten Samplingmethode kontrollieren zu können. Im Idealfall wird das zufällige Sampling mehrfach durchgeführt und im Ergebnis geclustert um Ausreißer erkennen zu können.

most references Die zweite Methode nutzt den Zitationsgraph um die Artikel auszuwählen, die am häufigsten zitiert wurden. Die Hoffnung dabei ist, dass dies die „wichtigsten“ Artikel sind und auch untereinander dichter vernetzt sind als die weniger Wichtigen. Dahinter steht die Vermutung, dass es legitim ist dort

[4] guess-language, Kent Johnson

die besten Clusterzentren zu vermuten. Dabei sei erwähnt, das diese Methode kein eigentliches Sampling ist, und nicht wiederholt werden braucht, da die entstehende Teilmenge immer die gleiche sein ist.

2.5 Stemming

Es macht Sinn die Abstracts zu stemmen, das heißt alle in einem Abstract enthaltenen Wörter auf deren Wortstamm zurückzuführen. So lässt sich die Anzahl der Dimensionen bei der Berechnung der Kosinusdistanzen stark reduzieren. Zusätzlich zählen alle Wörter, die die gleiche Bedeutung haben, auch nur für diese. Das Stemming lässt sich etwa mit dem Porter2 Stemming-Algorithmus [5] für englische Texte, ähnlich schnell wie die Sprachklassifizierung, durchführen. Hier ließ sich die Anzahl der Wörter auf, abhängig der Sample-Menge, zwischen 35.000 und 50.000 reduzieren.

3 Ergebnisse

In dieser Sektion werden die Ergebnisse mehrerer Experimente vorgestellt. Dabei sind unterschiedliche sampling-Methoden, verschiedene Zusammensetzungen der Beiden Featureräume und verschiedene Schwellwerte zu betrachten.

Die durch k-medoids zu minimierende Fehlerfunktion berechnet den Mittelwert über den Abstand jedes Dokumentes zu seinem zugeordneten Clusterzentrum. Der Abstand eines Dokumentes zu ihm nicht zugeordneten Clusterzentren, wird nicht betrachtet. Die beschriebene Fehlerfunktion ist durch die benutzte k-medoids Implementierung vorgegeben. Die Betrachtung weiterer Fehlerfunktionen übersteigt den Rahmen dieser Ausarbeitung.

3.0.1 Datensätze und Initialisierung Es wurden zwei verschiedene sampling-Methoden betrachtet:

- random references: 10000 zufällig aus dem CiteSeer^{X} Datensatz ausgewählte Dokumente.
- random references: Die 10000 Dokumente aus dem CiteSeer^{X} Datensatz mit den meisten Referenzen.

Zu jedem dieser Datensatz wurden Tests mit verschiedenen Parametern durchgeführt. Zusätzlich wurde für fixierte Parameter Vergleichswerte für eine Initialisierung durch h-cores und eine zufällige Initialisierung ermittelt.

3.0.2 Parameter Folgende Parameter wurden in den Experimenten betrachtet:

[5] The English (Porter2) stemming algorithm

– λ: Dieser Parameter bestimmt die Zusammensetzung der Distanzmatrizen (Rogers-Tanimoto und Kosinus) durch Fisher's Inverse χ^2-Methode. Da die Zusammenführung der Matrizen rechenintensiv ist wurden nur drei Werte betrachtet: $\lambda = 25\%$, $\lambda = 50\%$ und $\lambda = 75\%$. Dabei gibt die Prozentzahl den Anteil der Kosinusdistanz an der resultierenden Distanz an.

– r: Dieser Parameter bezeichnet den Schwellwert für die Berechnung der h-cores. Die Werte für r liegen in $[0, 1]$. Ein größerer Wert lässt mehr Kanten in der Berechnung zu, d.h. die Anzahl der h-cores steigt. Da r die Anzahl der h-cores beeinflusst, ist dies auch der Parameter der indirekt das k für k-medoids bestimmt. Für Beide sampling-Methoden und jeweils Beide Initialisierungen wurden h-cores für 12 verschiedene r-Werte berechnet: Von .05 bis .60. Größere Werte wurden nicht betrachtet, da die Zahl der h-cores schon bei .60 für alle λ-Werte über 2000 lagen.

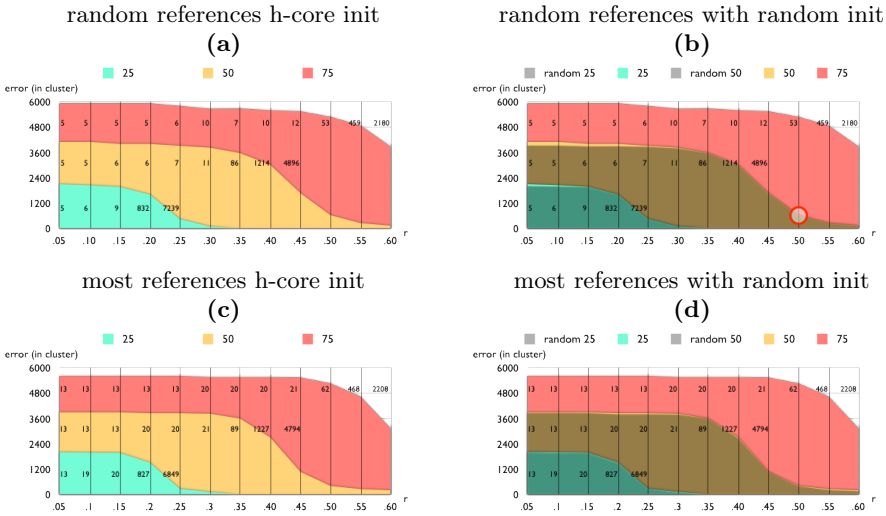

Abbildung 3. Die Abbildungen zeigen vier verschiedene Experimente: Abbildungen **(a)** und **(b)** zeigen die Ergebnisse mit den Daten die durch random sampling erzeugt wurden. Abbildung **c** und **d** zeigen die Ergebnisse für die Dokumente mit den meisten Referenzen. Die Farben stehen für unterschiedliche λ Werte für Fisher's Inverse χ^2-Methode in Prozent. Die Abbildungen **(a)** und **(c)** zeigen die Ergebnisse für die Initialisierung mit h-cores. Die grauen Zonen in **(b)** und **(d)** zeigen die Vergleichswerte mit zufälliger Initalisierung. Die Vertikale Achse bezeichnet den Fehler: den mittleren Abstand jedes Dokumentes zu seinem Clusterzentrum. Auf der horizontalen sind verschiedene Schwellwerte r der h-core Berechnung. Zusätzlich ist die Anzahl der Clusterzentren auf den Vertikalen für jeden r Wert im Bezug auf das verwendete λ gegeben.

Abbildung 3 **(a)**-**(d)** zeigt die Ergebnisse. Ein Diagram in 3 entspricht einem der zwei Datensätze mit einer der zwei Initialisierungen. Die Farben stehen für die unterschiedlichen λ-Werte. Auf der vertikalen Achse ist der Fehler nach k-medoids aufgetragen. Auf der Horizontalen sind verschiedene r-Werte.

Bei der Betrachtung ist zu beachten, dass die unterschiedlichen λ-Werte innerhalb eines Diagrammes nicht direkt miteinander vergleichbar sind. Da die λ-Werte für eine unterschiedliche Mischung der Beiden Distanzmaße stehen, sind die resultierenden Distanzmatrizen für unterschiedliche *lambda*-Werte verschieden. Dies kann zu einer anderen Distanzverteilung und damit unterschiedlichen Fehlerwerten führen.

Diagram 3 **(a)** zeigt die Ergebnisse für die zufällig ausgewählten Dokumente und eine Initialisierung mit h-cores. Mit steigenden h-core Schwellwerten r, steigt auch die Anzahl der gefundenen h-cores. Je größer die Anzahl der h-cores und damit die Anzahl der Kategorien, je niedriger ist der Fehler.

Diagram 3 **(b)** zeigt Vergleichswerte für die λ-Werte 25% und 50% mit zufälliger Initialisierung in grau. Die Vergleichswerte sind die jeweils besten k-medoids Ergebnisse aus 20 unterschiedlichen Initialisierungen. Generell sind ist der Fehler für mit zufälliger Initialisierung minimal niedriger als mit Initialisierung durch h-cores. Für den r-Wert 0.50% und λ ist der Fehler beider Initialisierungen identisch.

Diagram 3 **(c)** und **(d)** zeigen die Ergebnisse für den Datensatz mit den häufigsten Referenzen. Der Fehler bei zufälliger Initialisierung ist in diesem Fall immer niedriger als der Fehler bei h-core Initialsierung. Allerdings ist der Unterschied beider Fehler im Allgemeinen noch geringer als bei dem Datensatz mit zufällig ausgewählten Dokumenten.

3.1 Wortverteilung in Clustern

Zusätzlich zu den Error-Werten von k-medoids wurde außerdem die Wortverteilung in den einzelnen Clustern analysiert. Leider waren die Ergebnisse nicht aussagekräftig. Zunächst wurden alle Stopwörter aus den Clustern entfernt. Danach waren sich die häufigsten Wörter in den einzelnen Clustern immer noch sehr ähnlich - die häufigsten Wörter waren Standardbegriffe der Mathematik, die in allen Clustern häufig vorkamen. Die genauere Analyse der Wortverteilungen geht über den Rahmen dieser Arbeit hinaus

4 Fazit

Ziel der Arbeit war es die Zusammenführung zweier Distanzmaße und den Einsatz von h-cores auf dem Testdatensatz zu evaluieren.

h-cores als Approximation der Clusterzentren Widmen wir uns zunächst der Frage, ob die h-cores eine geeignete Approximation für die Clusterzentren darstellen. Die durchgeführten Experimente zeigen: die h-cores stellen tatsächlich eine gute Approximation dar. Zufällige Initialisierungen und h-core-Initialisierungen

bringen stark ähnliche Resultate - doch wo die Zufälligen mehrere Durchläufe benötigen, um auf ein Ergebnis zu kommen, benötigt man bei bei den h-cores nur einen Durchlauf.

Das bringt wiederum eine weitere Fragen auf.

Laufzeitvorteile durch die h-cores Bringt es einen Vorteil für die Laufzeit, wenn man schon weiß, dass man von ähnlichen Ergebnissen ausgehen kann? Diese Frage ist zugleich mit Ja und Nein zu beantworten. Denn auch wenn - wie eben erwähnt - Durchläufe des k-Mediod-Algorithmus spart, muss man dennoch zwei mal die Distanzmatrizen berechnen und diese ein weiteres mal zusammenführen. In unserem Fall übersteigt die Zeit die man zur Berechnung und Zusammenführung der Distanzen benötigt deutlich die Zeit, die das Clustering auch mit mehreren Durchläufen benötigt. Allerdings kann man aus unseren Beispiel keine direkten Schlüsse ziehen, wie sich das Verhältnis mit steigender Anzahl von Dokumenten entwickelt. Auch da sich damit nicht nur die Laufzeit des k-Mediods-Algorithmus verlängert, sondern auch z.B. die Wortbasis des VSM vergrößert und somit die Berechnung der Kosinusdistanzen länger dauert. Man muss außerdem bedenken: Mehrere Dokumente bedeuten, dass es ab einem bestimmten Punkt nicht mehr so einfach ist Distanzmatrizen im Zwischenspeicher zu behalten (was für uns der Grund war die Anzahl der Dokumente zu reduzieren) und so mindestens eine Zusammenführung von beiden Distanzen bei jedem Zugriff auf eine Distanz, während der Algorithmus läuft, neu berechnet werden müsste.

Höhere Qualität der Resultate durch die Kombination zweier Maße Steigt die Qualität bei der Nutzung zweier verschiedener Distanzen? Auch diese Frage ist nicht einfach zu beantworten. Qualität ist bei Clustern ein schwierig zu definierender Begriff. Der Schluss liegt nahe, anhand der gezeigten Grafiken zu sagen, dass eine Mischung, die die Tanimoto-Distanz deutlich stärker miteinbezieht, bessere Ergebnisse erzielt. Es also evtl. ganz ohne Abstracts besser wäre als die Mischung an sich. Doch muss man dabei bedenken, das die Güte der Cluster, in unserem Fall der durchschnittliche Abstand der Dokumente in ihrem Cluster, nicht unter verschiedenen Mischungen vergleichbar ist. Grund ist die Tatsache, dass der Parameter für die Mischung bereits jeweils in die genutzte Metrik mitbestimmt und verschiedene Clusterungen mit unterschiedlichen Parametern deshalb unvergleichbar macht. Eine ausführlichere Untersuchung der Qualität übersteigt den Rahmen dieser Arbeit.

Wahl der Samplingmethoden An den Ergebnissen sieht man, dass, vergleicht man die Ergebnisse mit den zufällig gewählten Dokumenten mit den most-reference-Artikeln, die zweite Methode wie gehofft bessere Ergebnisse lieftert. Wieviel besser, ist allerdings wieder schwer zu sagen.

4.1 Schwächen des vorgestellten Vorgehens

Hier möchten wir noch einmal zusammenfassend die Beschränkungen unseres Vorgehens aufzählen und Vorschläge für nähere Untersuchungen machen.

Es wäre wünschenswert den gesamten Datensatz zu Clustern, was den Aufwand stark erhöhen würde. Das würde stärkere Schlüsse ermöglichen. Auch wäre es von Interesse, die hier beschriebenen Ergebnisse genauer zu Untersuchen was statistische Qualität angeht. Dabei könnten z.B. weitere Evaluationsmaße wie der Silhouettenkoeffizient helfen. Zudem wäre es repräsentativer TF-IDF zum Finden der Labels für die Cluster zu nutzen. Bleibt man bei dem Ansatz nur eine Teilmenge zu clustern wäre die mehrmalige Ausführung von Samplingmethoden ratsam um Zweifel an der repräsentativität der Teilmengen zu minimieren. Des weiteren bleiben noch direkte Vergleiche zu anderen Methoden als hybrides Clustering mit k-Medoids offen.

4.2 Diskussion

Abschließend kann man sagen: Es ist nicht schwarz-weiß beantwortbar wie gut oder schlecht die hier untersuchten Methoden in der Praxis sind. Man kann sagen, dass wahrscheinlich noch zu wenige Dokumente genutzt wurden um die Möglichkeiten der h-cores mit gemischten Distanzen auszureizen.

Es ist die Frage, ob sich der Aufwand dieser Methode lohnt, oder ob weniger rechenintensive Methoden wie LDA nicht eventuell schneller aber dennoch befriedigend die Fragen der potentiellen Nutzer dieser Techniken beantworten können. Dazu kommt der Nachteil von h-cores, dass diese die Wahl möglicher Clustermethoden beschneidet - da es einen fast gänzlich auf eine Version von k-means beschränkt. Das kann natürlich nicht immer gut sein, denn k-means ist nicht in jeder Situation die beste Wahl. Im Großen und Ganzen kommt es also auch stark auf die am Anfang stehende Fragestellung an.

Literaturverzeichnis

[1] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. An Introduction to Information Retrieval. 2009.

[2] Wolfgang Glänzel and Bart Thijs. Using 'core documents' for the representation of clusters and topics. *Scientometrics*, 88(1):297–309, 2011.

[3] Wolfgang Glänzel. The role of core documents in bibliometric network analysis and their relation with h-type indices. *Scientometrics*, 93(1):113–123, 2012.

Abkürzungen

VSM Vector Space Model
SAX Simple API for XML