

# Topic tracking - Verfolgen von aufkommenden Themen

Robert Kubiak

13. Februar 2010

Tolle Sache das

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Problemfelder</b>	<b>4</b>
2.1	Event Detection . . . . .	4
2.1.1	Zu einem Event Dokumente ermitteln . . . . .	4
2.1.2	Aus Dokument Event ermitteln (Texterkennung) . . . . .	4
2.1.3	Neue Events entdecken . . . . .	4
2.2	Segmentierung . . . . .	5
2.3	Neuheitserkennung . . . . .	5
<b>3</b>	<b>Theoretischer Hintergrund</b>	<b>6</b>
3.1	BAYES Modell . . . . .	6
3.2	LDA . . . . .	6
3.3	Mengenverfahren . . . . .	7
<b>4</b>	<b>Themenverfolgung im Web</b>	<b>9</b>
4.1	Problemfelder . . . . .	9
4.2	Vorhandene Lösungen . . . . .	9
4.2.1	Themenerkennung (Event Tracking) . . . . .	9
4.2.2	Neuheitserkennung . . . . .	11
4.2.3	Benutzeroberfläche . . . . .	11
4.3	Auswertung der Lösungen . . . . .	12
<b>5</b>	<b>Zusammenfassung</b>	<b>14</b>

# 1 Einleitung

Die *Verfolgung von aufkommenden Themen* hat in der heutigen Wissensgesellschaft eine herausragende Bedeutung. Es gibt nicht wenige Experten, die zu einem Thema etwas schreiben, sondern viele Wissenschaftler an allen Universitäten veröffentlichen Artikel, Studien oder Bücher zu einem Themengebiet. Die Übersicht zu wahren, ist nicht mehr möglich. es wäre eine fülle von Menschen nötig, um dieser Flut Herr zu werden.

Dies gilt nicht nur im wissenschaftlichen Bereich, gerade im Web 2.0 gibt es Millionen von Autoren, die eigene Inhalte erzeugen, die mehr oder weniger lesenswert sind. Mit Hilfe von Online-Suchmaschinen kann man zu einem bestimmten Thema Seiten oder einzelne Artikel finden, jedoch ist die Übersicht nicht die Beste. Außerdem werden die Seiten geordnet nach der Anzahl der Links, die darauf verweisen. Es wird keine Sortierung getroffen, welche Seite wahrscheinlich am meisten auf die gesuchten Begriffe passen.

*Topic Tracking*<sup>1</sup> kann genau diese Aufgabe erfüllen. Dabei gibt es aber mehrere Problemfelder, die sich dabei ergeben. Diese werde ich in Kapitel 2 vorstellen. Zur Lösung dieser Probleme haben sich schon mehrere Techniken etabliert, die verwendet und miteinander kombiniert werden. Den theoretischen Hintergrund einiger dieser Ansätze werde ich im Kapitel 3 einführen. Viele Paper, die ich dazu gelesen haben, führen die Ansätze leider nicht so ausführlich ein. Diese Seminararbeit richtet sich also an jenen Leser, der in diesem Gebiet neu ist. Im Anschluss (Kapitel 4) werde ich einige praktische Lösungen vorstellen, um die Themenentwicklung im Internet zu analysieren.

---

<sup>1</sup>Meine Literaturverzeichnis listet nur englische Dokumente auf und auch Fachbegriffe in der Informatik sind durchgängig englisch. Ich werde im folgenden also viele englische Begriffe verwenden ohne sie ins deutsche zu Übersetzen, da es nur zu Verwirrungen führen würde.

## 2 Problemfelder

Die Probleme, die sich beim Topic Tracking ergeben, unterscheiden sich in der Art der Quelldaten, den Wissenstand über das gesuchte Thema und de, gewünschten Ausgabedaten.

### 2.1 Event Detection

Themen, Events beziehungsweise Topics innerhalb eines Dokumentes zu erkennen, ist das Hauptproblemfeld des Topic Trackings. Dies lässt sich aus verschiedenen Blickwinkel betrachten. Im folgenden sei  $E$  eine Menge von Events (bzw. Themen, Topics) und  $D$  eine Menge von Dokumenten (Artikel, Webseiten, Audio- oder Videomaterial). Des weiteren ist  $e_i \in E$  eine Event mit dem Index  $i$  und  $d_t \in D$  ein Dokument, das zum Zeitpunkt  $t$  aufgetaucht ist und eindeutig ist.

#### 2.1.1 Zu einem Event Dokumente ermitteln

Wenn ein bestimmtes Thema  $e_i$  gegeben ist, dann wird dazu alle Dokumente im Korpus  $D$  ermittelt, die das Thema beinhalten. Dies war die erste Definition, die ich zu Topic Tracking erhalten habe. Es ist sehr ähnlich mit dem Problem, dem man sich tagtäglich stellt, wenn man im Internet zu einem bestimmten Thema passende Seiten sucht. Dafür kann man Suchmaschinen verwenden. Diese geben aber nur Aussagen derart: "Dokument (beziehungsweise URL)  $d_t$  enthält die Suchanfrage *key*". Beim Topic Tracking wird das Thema  $e$  analysiert und abstrahiert, sodass das Event auch in Dokumenten gefunden wird, die nicht durch einfache keyword-Suche gefunden werden. Die Dokumente werden dann danach geordnet, wie genau die Anfrage auf das jeweilige Dokument passt. die Abstraktion ist eine Aufgabe des Topic Tracking.

#### 2.1.2 Aus Dokument Event ermitteln (Texterkennung)

Diese Aufgabe erfordert noch mehr künstliche Intelligenz. Diesmal ist ein Dokumente  $d_t$  gegeben und man will dazu das  $e$  ermitteln. Dabei sind schon Events bekannten, das Dokument muss also nur einem der Events zugeordnet werden. Dabei muss natürlich der gesamte Inhalt analysiert werden.

#### 2.1.3 Neue Events entdecken

Noch spezifischer wird es, wenn geprüft werden soll, ob das Dokument zu einem neuen Thema erschienen ist. Hierbei stellt sich zu erst die Frage, ob das Dokument zu einem

vorhanden Thema erschienen ist und falls nicht, wie das neue Thema spezifiziert werden soll. Das System arbeitet hier sehr autark, da es viele Freiheiten bekommen muss, um neue Themen definieren zu können.

## **2.2 Segmentierung**

Die Segmentierung ist nötig für Quelldaten, die nicht unterteilt sind. Dies können Nachrichtenmeldungen sein, die in einem andauernden Strom ankommen. Die Segmentierung fällt aber eher bei audiovisuellen Daten an, da diese selten durch bestimmten Trennzeichen unterteilt sind. Zum eigentlich Topic Tracking gehört dieses Problemfeld nicht, jedoch ist die Lösung dieses Problems nötig um ungetrennte Daten analysieren zu können.

## **2.3 Neuheitserkennung**

Wenn man die Themenentwicklung über die Zeit hinweg betrachten will, ist auch vom Vorteil darauf zu achten, das neu erschienene Dokumente auch einen neuen Inhalt haben. Deshalb lohnt es sich im Rahmen des Topic Trackings neue Dokumente mit bekannten Dokumenten zu einem Thema zu vergleichen. Dokumente die einfach Neuerscheinungen oder Ausschnitte von alten bekannten Dokumenten sind, sollten als solche markiert werden.

## 3 Theoretischer Hintergrund

Für die verschiedenen Problemfelder haben sich in der wissenschaftlichen Gemeinschaft bestimmte Ansätze aus der *künstlichen Intelligenz* etabliert. Dabei wird nicht pro Problemfeld einer der folgenden Ansätze verwendet, viel mehr werden die Ansätze in allen Problemfeld gleichberechtigt verwendet.

### 3.1 Bayes Modell

BAYES hat viel beigetragen zur Wahrscheinlichkeitsrechnung. Unter anderem hat er ein Baummodell entworfen um die Wahrscheinlichkeit von bestimmten Ereignissen mit mehrere Variablen zu berechnen. Dazu wird ein Baum aufgespannt. Jede Ebene des Baumes repräsentiert eine Variable im System. Die Anzahl der Knoten auf der Ebene gibt die verschiedene Werte der Variable an. auf den Kanten zwischen den Knoten stehen die Wahrscheinlichkeiten. Will man nun die Wahrscheinlichkeit eines Ereignisses, das mit dem Belegen der Variablen mit Werten gleichzusetzen ist, ermitteln, nimmt man die Pfad von der Wurzel bis zum Blatt über die Knoten, die das Ereignis beschreiben. Die Wahrscheinlichkeit, dass das Ereignis eintritt ist das Produkt aller Wahrscheinlichkeiten der Kanten.

Bei der Dokumentanalyse ist jetzt die Frage wie wahrscheinlich ist das Thema (Modell) eines Dokumentes ( $d_t$ ) das angenommene Modell ( $M$ ) entspricht.

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}$$

### 3.2 LDA

[BNJ03] beschreibt *Latent Dirichlet Allocation* als ein dreischichtiges, hierarchisches BAYES Modell. Ich werde diese Quelle verwenden, um einen Einblick in die Idee von LDA zu geben. Wie das Paper werde ich LDA anhand eines Beispielsproblems erklären und zwar der *Texterkennung* (Kapitel 2.1.2). Dabei wird das Modell als ein *Hidden Markov Model* (Abb. 1) konstruiert, bei dem das einzig beobachtbare Verhalten die Wörter  $w$  im Dokument sind, die von anderen Dingen abhängig, zum Beispiel vom Thema  $z$ , sind.

In der Abbildung ist  $N$  das Dokument und  $M$  die Menge aller Dokumente. Die Symbol  $\alpha$ ,  $\beta$  und  $\Theta$  sind Variablen, die die Wahrscheinlichkeiten von  $z$  und  $w$  beeinflussen. Die Variablen werden durch einen generativen Prozess optimiert. Um das LDA-Verfahren zu verstehen, benötigt man gute Kenntnisse in der Theorie und den Verfahren der Wahr-

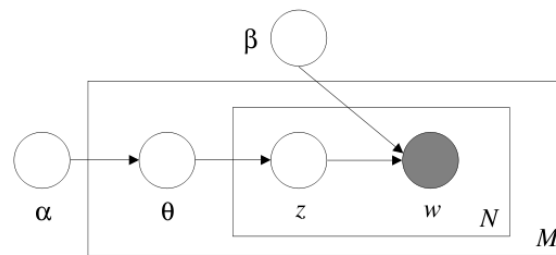


Abbildung 1: LDA als Hidden Markov Model [BNJ03, S. 997]

scheinlichkeitsrechnung besitzen. Ich kann das Verfahren nur soweit beschreiben, welche Variablen das Ergebnis beeinflussen.

Gibbs Sampling ist ein Verfahren um Anfangswahrscheinlichkeiten von Variablen abzuschätzen. Dieser Algorithmus kann also bei LDA dazu verwendet werden, um die Variablen  $\alpha$ ,  $\beta$  und  $\Theta$  sinnvoll vor zu belegen.

### 3.3 Mengenverfahren

Das Mengenverfahren kommt nicht aus der Wahrscheinlichkeitsrechnung. Es versucht die Übereinstimmung zweier Dokumente, beziehungsweise ihrer Themen dadurch herzu-leiten, dass es die Menge der Wörter betrachtet. Dabei werden bestimmte Füllwörter nicht mit in die Menge aufgenommen. Alle relevanten Wörter werden in die Menge aufgenommen. Nun lässt sich eine Prozentzahl bestimmen, wie viele Wörter in beiden Dokumenten vorkommen und wie viele nur in einem von beiden. Um so höher der erste Wert ausfällt, desto größer ist die Übereinstimmung. Erweitern lässt sich die zu einem Vektoren-Ansatz, bei dem nicht nur das pure Vorhandensein sondern auch die Anzahl des jeweiligen Wortes notiert wird.

Bei allen erwähnten Verfahren kann es so gut wie nie eine Antwort geben: "Dokument  $d_t$  hat genau das Thema  $e$ ." Vielmehr ist es jeweils nur ein Wert zwischen 0 und 1. Es muss also jeweils ein Schwellwert ermittelt werden, ab dem 2 Dokumente dasselbe Thema haben oder doch verschieden sind. Diese Schwellwert ist geradezu entscheidend für die Qualität des Algorithmus. Mit Hinsicht auf das Ziel der Analyse setzt man den Schwellwert wohl auch hoch und runter. Wenn man das Thema eines Dokuments aus einer bekannten Menge von Themen bestimmen will, wird man den Schwellwert niedrig ansetzen, sodass wenigstens ein Thema genommen wird. Hier lässt sich auch nach dem Wert sortieren und den höchsten nehmen. Auf der anderen Zeit muss man

beim Ermitteln ob ein Dokument ein neues Thema hat, den Schwellwert höher setzen. Hier sollte wahrscheinlich öfter herauskommen, dass das Dokumententhema nicht mit einem Thema aus der bekannten Themenmenge übereinstimmt. Um die Schwellwert zu optimieren, bieten sich Lernalgorithmen an.

## 4 Themenverfolgung im Web

Dem Problemfeld, dem ich mich widmen will ist die Unorganisiertheit des Internets. Es gibt viele tausend Seiten zu einem Thema und noch viele mehr, die nur hin und wieder zu einem Thema etwas berichten. Will man auf dem laufenden bleiben, reicht es nicht eine Seite anzusteuern, die versucht die Informationen zu sammeln. Auch diese wird sich auf wenige Quellen beziehen und viele außer acht lassen. Und wenn es nun solch eine Seite gar nicht gibt, ist man allein gelassen mit der Sichtung und Gewichtung der verfügbaren Daten.

### 4.1 Problemfelder

Hier kann das Topic Tracking eine große Hilfe sein. Die Problematik lässt sich in 4 Teile teilen. Zu erst müssen mögliche Dokumente ermittelt werden. Es ist nicht möglich jede Internetseite zu validieren, es muss eine Vorauswahl getroffen werden, die schnell und unkompliziert ist. Danach muss herausgefunden werden, welches Dokument nach unseren Kriterien das Thema behandelt. Dies ist eine zweite Filterung. Nun sollten auch Dokumente ausgeschlossen werden, die nicht neues liefern. Im letzten Schritt sollten die ermittelten Daten in einer ansprechenden Benutzeroberfläche präsentiert werden.

### 4.2 Vorhandene Lösungen

#### 4.2.1 Themenerkennung (Event Tracking)

Die erste Filterung kann durch eine einfache *keyword*-Suche gelöst werden. Dazu muss das Thema in verschiedene keywords aufgeteilt werden. Dies können zum Beispiel alle Wörter des Themas sein und zusätzlich vom Benutzer selbst angegebene Wörter sein. Diese können dann bei Suchdiensten eingegeben werden, um eine lange Liste von Internetseiten zu erhalten. Im folgenden sollte zu jeder URL vermerkt werden, wann diese besucht wurden, um in zukünftigen Schritten Seiten nicht mehrfach besuchen zu müssen, ohne das es eine Änderung des Inhalts gab.

Erst im zweiten Schritt wird eine „intelligente“ Filterung angewendet. Grundlage ist eine manuelle Sichtung von ein paar Internetseiten um eine Basis von Dokumenten zu haben, die auf jeden Fall mit dem Thema etwas zu tun haben. diese werden als Trainingsdaten verwendet.

[ACD<sup>+</sup>98] thematisiert verschiedene Ansätze von den 3 Forschungsgruppen: CMU (Carnegie Mellon University), UMass (University of Massachusetts in Amherst) und Dragon (Dragon System). Dabei ist das Testfeld für jede Gruppe dasselbe. Aus einem Korpus

von 16.000 Dokumenten werden 1, 2, 4, 8 beziehungsweise 16 Dokumente zu einem Thema als Testdaten bereitgestellt. Des Weiteren wurden auch Dokumente bereitgestellt, die das Thema nicht behandeln. Nach der Trainingsphase müssen die Programme aus den restlichen 15.984 Dokumenten diejenigen ermitteln, die das gleiche Thema behandeln. Die Vorgehensweise unterscheidet sich erwartungsgemäß je nach Gruppe voneinander.

**UMass** Die UMass entwickelt aus den Trainingsdaten keywords, die das Thema beschreiben sollen. Die Dokumente, die nicht das Thema behandeln, haben sie verwendet um bestimmte keywords auszuschließen. Insgesamt haben sie pro Thema ([ACD<sup>+</sup>98] hat die Dokumente in 25 Themen geteilt) 10 bis 100 keywords produziert und diese danach gegen die positiven Testdaten laufen lassen um ein Schwellwert zu ermitteln. Ein weiterer Ansatz geht ähnlich vor, verwendet aber nur Nomen, die nur in den positiven Testdaten auftauchen. Umso öfter ein Nomen in verschiedenen Dokumenten der positiven Testdaten auftaucht, desto wichtiger ist es.

**CMU** Auch die CMU verfolgt gleich 2 Ansätze. Der erste Ansatz evaluiert alle positiven und zu Verfügung stehenden negativen Dokumenten und gibt ihnen jeweils ein Index-Vektor. Die Entscheidung ob ein Dokument zu einem Thema gehört, wird über den Unterschied bzw. Entfernung des Dokumentvektors mit dem Themenvektors entschieden. Der zweite Ansatz ist eine Implementierung eines Entscheidungsbaums. Bei der Analyse der Trainingsdaten wird dieser konstruiert und bei der Anwendung auf den gesamten Korpus muss jedes Dokumenten den Entscheidungsbaum durchlaufen. Landet der Algorithmus in einem Blatt, das positiv ist, wird das Dokument als zu dem Thema gehörend angenommen.

**Dragon** Dragon verwendet ein Ansatz, den sie auch verwendet haben um das Segmentierungsproblem zu lösen. Dieses verwendet ein HMM und erinnert somit an LDA. Der Ansatz analysiert den Text um ein bestimmten Hintergrundmodell aufzustellen. Damit dieses Modell kein Wörter beachtet, die in der Sprache einfach oft auftauchen, verwenden sie ein Sprachmodell um die Wörter gewichten zu können. Das Themenmodell ist somit eine Zusammenfügung der Hintergrundmodelle der einzelnen positiven Testdokumente.

**Verknüpfungen** [HCP<sup>+</sup>09] stellt einen weiteren Aspekt vor, unter dem bestimmte Dokumente zusammengehören: Zitate. Die Autoren beziehen sich dabei auf Zitate in wissenschaftlichen Texten. Es ist leicht zu erahnen, dass Texte, die ein Dokument zitiert, mit dem Thema des Dokumentes zu tun haben. Dies lässt sich auf Internetseiten besonders gut anwenden. Die meisten Internetseiten präsentieren Hyperlinks zu anderen Seiten, die

das selbe Thema behandeln oder die zu dem jeweiligen Internetartikel den Anstoß gegeben haben. Es spannen sich somit Fäden zwischen den Seiten und es kommen teilweise neue Internetseiten hinzu. Es lassen sich auch Entwicklungen erkennen, den wenn eine Seite eine andere zitiert, dann kann die Seite eine gedankliche Weiterentwicklung der zitierten Seite sein. Es lässt sich verfolgen, welche Meinungen von welcher abhängt und wo der Ursprung ist.

**Annotationen** Die Autoren von [BEZG09] und [BEG09] halten sich nicht lange mit den Wörtern in den Dokumenten auf, sondern verwenden dessen Annotationen. Damit ist es nicht nötig das gesamte Dokument zu analysieren. Im Web gibt es auch Annotationen in den Kopfzeilen der HTML-Seiten. Viele Seiten setzen dort korrekte Meta-tags. Dieser Ansatz lässt sich also auch für Internetdokumente verwenden.

#### 4.2.2 Neuheitserkennung

[ZCM02] widmet sich der Neuheitserkennung. Dabei werden gleich 3 Ansätze und eine Mixtur aus diesen vorgestellt. Der erste Ansatz vergleicht die Menge aller Wörter in den Dokumenten. Zu erst müssen dabei Wörter definiert werden, die allgemein oft gebraucht werden (Artikel, Hilfsverben, etc.) und bei der Analyse ignoriert werden. Danach werden die restlichen Wörter gewichtet. Wörter die häufig im untersuchten Dokument auftauchen werden höher gewichtet als diejenigen die nicht so oft auftauchen. Ist das Wort jedoch in allen anderen gefilterten Dokumenten oft vorhanden, dann wird es auch höher bewertet. Das Ziel ist bei diesem Schritt unwichtige Wörter raus zu lassen. Ein Wort ist also nur wichtig, wenn es häufig im untersuchten Dokument vorkommt oder häufig in den anderen gefilterten Dokumenten.

Ein weiterer Ansatz berechnet den geometrischen Abstand zwischen 2 Vektoren, die die Dokumente darstellen. dabei werden einfach alle Wörter in den bekannten Dokumenten als eine Dimension angenommen. Der Wert der Stelle im Vektor entspricht der Anzahl des Wortes in im Dokument. Der Abstand wird über die Kosinus-Distanz ermittelt (Cosine distance). Der dritte Ansatz beruht auf der Sprachmodell der Dokumente. Die genaue Logik hinter diesen Wahrscheinlichkeitsmodell erfordert ausreichend Kenntnisse in der Wahrscheinlichkeitstheorie. Darum kann ich den dritten Ansatz nur nennen und verweisen auf den Text [ZCM02].

#### 4.2.3 Benutzeroberfläche

Was mich verwirrt ist, dass sich bis jetzt wenige Texte mit der Schnittstelle zum Benutzer beschäftigt haben. Es ist Ansichtssache, ob ein bestimmtes Dokument ein bestimmtes

Thema hat. Jeder Benutzer wird es anders sehen, ob das Dokument noch relevant ist oder nicht. Grenzt ein Algorithmus die Auswahl soweit ein, weil das Ziel der Entwickler ist, auf keinen Fall Dokumente zu präsentieren, die nichts mit dem Thema zu tun haben, werden auch Dokumente abgeschnitten, die das Thema behandeln. Es muss also für den Benutzer möglich sein, die getroffenen Entscheidungen des Algorithmus zu verfolgen und wenn nötig Werte anpassen zu können.

Nur [MDF<sup>+</sup>08] beschäftigt sich auch mit der Benutzeroberfläche. Über das von den Autoren besprochene Programm kann der Benutzer eine Sortierung der Dokumente einsehen und prüfen zu wie viel Prozent die einzelnen Dokumente dem gesuchten Thema entsprechen. Durch eine längere Benutzung des Programms kann der Benutzer selbst ermitteln bei wie viel Prozent Übereinstimmung er die Texte noch für relevant empfindet. Ein Schwellwert ist nicht mehr nötig, den kann der Benutzer selbst bestimmen. Bei den hier vorgestellten Algorithmen könnte man ähnlich vorgehen. Der Benutzer kann bestimmte Werte anpassen und einen Testdurchlauf starten. Entsprechen die Ergebnisse nicht seinen Erwartungen, kann er die Werte anpassen. Um ungeübte Benutzer nicht zu verschrecken, sollte dies natürlich nur angeboten werden und keine Pflicht sein.

Eine weiteres Ziel der Benutzeroberfläche muss eine geeignete Präsentation der gewonnenen Daten sein. Es ist wenig sinnvoll die bei der Berechnung der Dokumentenanalyse anfallenden Daten einfach zu verwerfen und sie dem Benutzer nicht zur Verfügung zu stellen. Wenn ein Dokument ein anderes zitiert oder Bezug darauf nimmt und das analysiert wurde, dann sollte das auch dargestellt werden durch Pfeile. Bei der Evolution ist es auch interessant wie viele Seiten in einem bestimmten Zeitraum erstellt wurde. Es lassen sich damit Aufmerksamkeitsentwicklungen feststellen (siehe auch [MDF<sup>+</sup>08]).

### 4.3 Auswertung der Lösungen

Die Themenerkennung wurde schon vielfach gelöst, aber die Fehlerquoten unterscheiden sich stark. [ACD<sup>+</sup>98] nennt bei den Lösung von *Dragon* von 71 %, bei *CMU* 29 % und bei einer Lösung von *UMass* von 13 %. Dabei fällt auf, dass der Ansatz der neben der Themenerkennung auch die Segmentierung lösen kann, am schlechtesten abschneidet. Was darauf schließen lässt, dass die Lösung am besten ist, wenn die Algorithmen und die Variablen auf ein Ziel ausgerichtet sind. Eine Fehlerquote von 13 % ist sehr klein und brauchbar. Die Ausführungszeit der Algorithmen hat der Text auch bewertet, hat aber keine Zahlen genannt. Es ist nur zu sehen, dass je nach Algorithmus die Ausführung mehr oder weniger steigt in Abhängigkeit von der Anzahl der verwendeten Daten. Mehr Trainingsdaten bedeutet eine längere Dauer in der Trainingsphase. Des weiteren kann dadurch auch ein komplexeres Modell entstehen, gegen das jedes Dokument geprüft wird, die Dauer der Analyse dauert also auch länger. Die Analyse der anderen Ansätze ergeben

ähnliche Erkenntnisse.

[ZCM02] hat die Analyse ihres Algorithmus mit Testdaten aus verschiedenen Newsstreams evaluiert. Dabei zeigte die Methode des geometrischen Abstands die besten Werte. Mit größerem Abstand folgt der Algorithmus, der die Sprachmodell in eine Wahrscheinlichkeitsbeziehung setzt. Auf dem letzten Platz liegt die Mengenmethode. Diese Evaluierung wundert mich, da der Unterschied zwischen der Mengenmethode und dem Abstand nicht groß ist. Beide basieren auf dem Vorkommen der einzelnen Worte und beide beziehen auch die Häufigkeit mit ein. Die Idee dahinter ist also relativ gleich. Der Unterschied ist die Messung des Unterschieds der gewonnen Menge beziehungsweise Vektors. Es zeigt sich daran, dass es 3 Schritte gibt, die ein Algorithmus zur Neuheitserkennung braucht: Abbilden des Dokuments in eine geordnete Struktur, Gewichtung der Element, Messung der Unterschiede.

Zu Benutzeroberfläche lässt sich wenig analysieren. Ich konnte noch keine Studie zur Benutzerzufriedenheit lesen, da die meisten Ansätze den normalen Benutzer außen vor lassen. Wahrscheinlich lässt sich aber das Gesamtergebnis eines Algorithmus, der Text- und Neuheitserkennung bietet, stark verbessern durch eine benutzerfreundliche und funktionsmäßige Oberfläche. Besonders im Internet boomt zur Zeit das Interesse an neuer User-Experience.

## 5 Zusammenfassung

Topic Tracking ist schon ziemlich stark erforscht und es gibt schon eine Menge bewährte Techniken. Dennoch hat es noch nicht den Sprung zum Benutzer gefunden. Zwei Gründe fallen mir dafür ein:

- Die Algorithmen arbeiten noch nicht perfekt. Es gibt viele Stellschrauben, die das Ergebnis stark beeinflussen. Bevor nicht ein ausgereiftes System vorhanden ist, wird es keine Firma geben, die es stabil implementieren will. vorhandene Lösungen bieten bis jetzt nur ein Nischendasein, da sie sich nur auf bestimmte Themen beschränken. Meist im wissenschaftlichen Bereich (Digitale Bibliotheken)
- Eine ansprechende intuitiv zu steuernde Oberfläche fehlt. Vom heutigen Computerbenutzer kann nicht mehr erwartet werden, dass er das innere des Programmes versteht, das er benutzt. Jedoch sind viele vorhandene Lösungen im alpha-Stadium, in denen viele Debuginformationen mit ausgeworfen werden.

Ich denke aber mit Suchmaschinenriesen wie Google oder Microsoft/Yahoo werden auch diese leistungsfähigen Suchdienste ihren Siegeszug antreten. Die Informationen, die man durch eine so komplexe Analyse erhält ist gegenüber den jetzigen Ergebnissen soviel besser, dass es irgendwann kommen wird und eine breite Akzeptanz finden wird.

Ich würde meinen Ansatz zum Topic Tracking nicht bei den Modellen der Wahrscheinlichkeit suchen und das nicht nur, weil deren Theorie für mich undurchdringbar ist. Mit einfacheren Mitteln über die Aufstellung und Analyse von Themen- und Wortvektoren lassen sich effektive Lösungen zusammenstellen, bei denen man noch viel optimieren kann durch die Berechnung des Abstands und der Höhe des Schwellwertes. Des weiteren kann die eine Lösung leicht abgewandelt auch gleich zum Testen von der Neuheit eines Dokuments verwenden. Mein Augenmerk würde ich auf die Benutzeroberfläche legen um Topic Tracking zu einer plastischen Erfahrung zu machen.

## Literatur

- [ACD<sup>+</sup>98] James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, Yiming Yang, James A. Umass, Brian A. Cmu, Doug B. Cmu, Adam B. Cmu, Ralf B. Cmu, Ira C. Dragon, George D. Darpa, Alex H. Cmu, John L. Cmu, Victor L. Umass, Xin L. Cmu, Steve L. Dragon, Van Mulbregt Dragon, Ron P. Umass, Thomas P. Cmu, Jay P. Umass, and Mike S. Umass. Topic detection and tracking pilot study final report. In *In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998.
- [BEG09] Levent Bolelli, Seyda Ertekin, and C. Lee Giles. Topic and trend detection in text collections using latent dirichlet allocation. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval April 06-09, 2009*.
- [BEZG09] Levent Bolelli, Seyda Ertekin, Ding Zhou, and C. Lee Giles. Finding topic trends in digital libraries. In *JCDL '09: Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 69–72, New York, NY, USA, 2009. ACM.
- [BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
- [HCP<sup>+</sup>09] Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and Lee Giles. Detecting topic evolution in scientific literature: how can citations help? In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 957–966, New York, NY, USA, 2009. ACM.
- [MDF<sup>+</sup>08] Fabian Mörchen, Mathäus Dejori, Dmitriy Fradkin, Julien Etienne, Bernd Wachmann, and Markus Bundschuh. Anticipating annotations and emerging trends in biomedical literature. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 954–962, New York, NY, USA, 2008. ACM.
- [ZCM02] Yi Zhang, Jamie Callan, and Thomas Minka. Novelty and redundancy detection in adaptive filtering. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 81–88, New York, NY, USA, 2002. ACM.