Novelty Detection im Information Retrieval

DENNIS HOPPE
Bauhaus-Universität Weimar

Werden Suchergebnisse ausschließlich anhand der Relevanz zu einer Anfrage sortiert, so liefert dies in vielen Fällen unbefriedigende Resultate. Vor allem bei Nachrichten werden einem Nutzer sehr viele verschiedene Artikel präsentiert, die auf dem zweiten Blick inhaltlich jedoch größtenteils redundant sind. Sie sind somit nicht von Interesse, nicht relevant. Gegenwärtige Retrievalsysteme sollten demnach Verfahren bei der Generierung der Suchergebnisse anwenden, die sowohl die Relevanz zu einer Anfrage erhalten als auch neue Informationen in Dokumenten berücksichtigen. Beides mit dem Ziel, dem Nutzer keine redundanten Informationen zu präsentieren. In diesem Zusammenhang wird der Forschungsbereich *Novelty Detection* vorgestellt, um Suchergebnisse anhand neuer Informationen neu zu sortieren.

1. EINLEITUNG

Aufgrund der großen Informationsflut im Internet ist es heutzutage wichtig Systeme anzubieten, die dem Nutzer bei der Bewältigung dieser helfen. Internetdienste wie *Google News* und *Bing News* fassen hierzu automatisch Schlagzeilen verschiedenster Nachrichtenagenturen zusammen, siehe Abbildung 1. Indem Nutzer Anfragen an den jeweiligen Nachrichtendienst stellen, können sie ihren Informationsbedarf bezüglich konkreter Themen befriedigen. Bei oftmals mehr als einhundert Artikeln zu einem Thema ist es dem Nutzer aufgrund des hohen Zeitaufwands jedoch

Abbildung 1. Auszug aus Ergebnissen zur Suchanfrage `nobelpreis chemie` an *Bing News*. Die Suchanfrage befriedigt den Informationsbedarf des Nutzers bereits vollkommen. Es zeigt sich jedoch, dass bereits nach Lesen der ersten Nachricht „Nobelpreis für Chemie geht an Forscher aus USA und Israel“ die nachfolgenden keine weiteren Informationen bieten. Sie sind somit nicht mehr von Relevanz.

nicht möglich, jeden Artikel vollständig zu lesen. Dennoch ist er an den wichtigsten Informationen aus allen Artikeln interessiert. Durch das Anbieten von Zusammenfassungen von Artikeln ist der Informationsbedarf des Nutzers maximierbar, da der Zeitaufwand beim Lesen der einzelnen Artikel deutlich reduziert wird. *Automatic Text Summarization* beschäftigt sich eingehend mit der Aufgabe, für Dokumente Zusammenfassungen zu erzeugen, die ausschließlich relevante Informationen enthalten.

1.1 Automatic Text Summarization

Nutzer, die ein Nachrichtenthema über einen längeren Zeitraum verfolgen, sind ausschließlich an neuen Informationen interessiert, mit denen sie sich neues Wissen aneignen können. Gerade bei Nachrichten werden jedoch oftmals keine neuen Fakten präsentiert. Vielmehr wird, dem regelmäßigen Leser bereits bekanntes, Wissen neu formuliert. Es ist in diesem Fall unzureichend, wenn ein traditionelles Informationssystem seine Artikel nur anhand der Relevanz zur Anfrage sortiert. Auch eine automatische Textzusammenfassung verhindert nicht, dass für den Nutzer Artikel mit bereits bekanntem Wissen ausgeblendet werden, da der zeitliche Verlauf von Nachrichtenthemen nicht berücksichtigt wird. Es ist festzuhalten, dass Dokumente der Ergebnismenge nicht nur anhand ihrer Relevanz zur Anfrage sortiert werden sollten, sondern zusätzlich anhand neuer Informationen, die sie dem Nutzer bieten. *Novelty Detection*, die Erkennung neuer Informationen, hat sich als Teilbereich aus der *Text Summarization* herausgebildet, um sich dieser Aufgabe anzunehmen [7].

1.2 Novelty Detection

Novelty Detection ist nicht auf die Sortierung von Nachrichten begrenzt. Ein weiteres Anwendungsgebiet ist die Breitensuche. Hierbei sind stark differenzierende Ergebnisse gewünscht. Eine Suche nach allen Betriebssystemen für den PC bei *Google* zeigt, dass vor allem populäre Betriebssysteme wie *Microsoft Windows*, *Linux*-Derivate und *Apple Mac OS* auf den vorderen Plätzen anzutreffen sind. Weitere Betriebssysteme wie *FreeBSD*, *OS/2* oder *DOS* finden sich unter den ersten Ergebnisseiten nicht. Es sind vermehrt Ergebnisse zu finden, die *Betriebssysteme* als Begriff selbst behandeln. Zahlreiche Ergebnisse haben zudem dasselbe Betriebssysteme zum Thema, sind also in diesem Fall redundant. Ein vollständiger Überblick ist somit aufgrund mangelnder Abgrenzung in den Ergebnissen nicht gegeben bzw. erschließt sich erst nach intensiver Sichtung der Suchergebnisse. Wünschenswert wäre für dieses Szenario, dass jeder Treffer ein eigenständiges Betriebssystem behandeln würde. Traditionelle Suchmaschinen leisten dies gegenwärtig nicht.

In [39] unterscheidet Xu die zwei vorgestellten Teilgebiete, die Sortierung von Suchergebnissen und die Breitensuche, der *Novelty Detection*:

Directed: Ein Nutzer möchte mehr über einen neuen Aspekt eines ihm bereits bekannten Themas erfahren. Hierbei sollten neue Artikel möglichst ähnlich zu bereits gelesenen sein und zudem neue Informationen beinhalten.

Undirected: Ein Nutzer möchte möglichst viel über ein Thema erfahren. Hierbei sollten neue Artikel sich möglichst stark von zuvor gelesenen unterscheiden. Der Begriff der *Novelty* ist hier als Gegenteil von Redundanz aufzufassen.

Letztgenannte Interpretierung wird am häufigsten in der Forschung behandelt.

2. DEFINITION VON NOVELTY

Um etwas neues in einem Artikel festzustellen, sind intuitiv dessen Unterschiede zu anderen Artikeln hervorzuheben; u.a. durch Zählen neuer Wörter. Wenn sich jedoch keine neuen Fakten ergeben, werden – wie bereits angesprochen – Nachrichtenartikel oft neu formuliert. Zwei zeitlich aufeinanderfolgende Artikel übermitteln in diesem Fall trotz verschiedenem Vokabular denselben Informationsgehalt. Der zweite Artikel ist trotz neuer Wörter also nicht als neu anzusehen. Eine Definition von *Novelty* muss also über die Erfassung neuer Wörter hinausgehen und berücksichtigen, ob Dokumente (zuvor Artikel) inhaltlich redundant sind. Bevor der Begriff der *Novelty* stärker gefasst wird, zunächst eine Unterscheidung drei in diesem Zusammenhang auftretender Begriffe:

Redundanz: Ermittelt, wieviel *redundante* Informationen ein Dokument enthält. Informationen sind redundant, wenn diese zuvor von anderen Dokumenten abgedeckt wurden. Frage: Wie oft wiederholen sich Konzepte?

Novelty: Ermittelt, wieviel *neue* Informationen ein Dokument enthält. Informationen sind neu, wenn sie relevant zur Anfrage sind und nicht zuvor von anderen Dokumenten abgedeckt wurden. Frage: Welche Konzepte sind neu?

Diversität: Ermittelt, wieviel *verschiedene* Informationen ein Dokument enthält. Unter verschiedenen Informationen werden Informationen verstanden, die relevant sind und zuvor nicht von anderen Dokumenten abgedeckt wurden. Frage: Wie viele verschiedene Konzepte existieren?

Festzuhalten ist, dass alle drei Begriffe *zuvor* untersuchte Dokumente in die Bewertung miteinbeziehen. Es handelt sich um asymmetrische Maße. Redundanz und *Novelty* werden hier als zwei verschiedene Begriffe aufgefasst.

Zhang [43] formulierte *Novelty* erstmals mathematisch mit Hilfe der Redundanz. Der Autor fasst dabei *Novelty* und Redundanz als entgegengesetzte Endpunkte einer gemeinsamen Skala auf.

Ein Dokument $d \in D = \{d_1, \ldots, d_n\}$ wird hierbei durch die endliche Menge an Wörtern $W = \{w_0, \ldots, w_m\}$ aller in D enthaltenen Dokumente repräsentiert. W bildet das Vokabular aller Dokumente. Um die Redundanz zwischen zwei Dokumenten $d_i, d_j \in D$ zu berechnen, wird eine logische Sicht auf diese Dokumente benötigt. Diese wird durch ein Dokumentmodell $\mathbf{D}$ realisiert.

Definition 2.1. Die Redundanz ρ eines Dokumentes $\mathbf{d}_t \in \mathbf{D}(t) = \{\mathbf{d}_0, \ldots, \mathbf{d}_{t-1}\}$, welches zum Zeitpunkt t untersucht wird, hängt von allen zuvor betrachteten Dokumenten ab: $\rho(\mathbf{d}_t) = \rho(\mathbf{d}_t|\mathbf{D}(t))$.

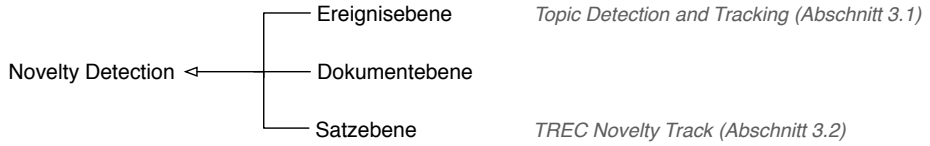


Abbildung 2. Taxonomie zur *Novelty Detection*. Kursive Begriffe benennen Forschungsbereiche, die sich explizit mit der Erkennung von *Novelty* auf der jeweiligen Ebene beschäftigen. Der Bereich der Dokumentebene findet kaum Berücksichtigung [2, 4, 11, 16, 17].

Definition 2.2. $\rho(\mathbf{d}_t|\mathbf{D}(t))$ hängt von allen relevanten Dokumenten $\mathbf{D}_r(t) \subseteq \mathbf{D}(t)$ ab, die vor $\mathbf{d}_t$ untersucht wurden: $\rho(\mathbf{d}_t|\mathbf{D}(t)) = \rho(\mathbf{d}_t|\mathbf{D}_r(t))$.

Definition 2.3. Seien A, B zwei Dokumentmengen, wobei gilt $B \subseteq A$. Wenn B dazu führt, dass $\mathbf{d}_t$ als redundant angenommen wird, so führt A ebenso dazu, dass $\mathbf{d}_t$ als redundant angenommen wird: $B \subseteq A \Rightarrow \rho(\mathbf{d}_t|A) \geq \rho(\mathbf{d}_t|B)$.

Um die Redundanz eines Dokuments zu messen, ist $\rho(\mathbf{d}_t|\mathbf{D}(t))$ zu berechnen. Kapitel 4 stellt Verfahren vor, um ρ zu berechnen.

3. NOVELTY DETECTION

Die Erkennung von *Novelty* wird gegenwärtig entsprechend Abbildung 2 auf drei verschiedenen Ebenen durchgeführt. Auf der Ereignisebene muss ein neues Dokument nicht nur relevant zur Anfrage sein, sondern auch ein neues Ereignis behandeln. Im Gegensatz dazu wird auf Dokument- und Satzebene ein Dokument auch als neu angesehen, wenn es relevante und neue Informationen zu einem bereits bekanntem Ereignis liefert.

3.1 Topic Detection and Tracking

Ereignisbasierte *Novelty Detection* ist eng verwandt mit dem Forschungsbereich *Topic Detection and Tracking*, kurz TDT [8, 16, 27, 42]. Ziel ist es, online neue Ereignisse (engl. *event detection*) als auch das erste Auftreten einer Nachricht zu einem neuen Thema (engl. *first story detection*) zu entdecken. Techniken zur Erkennung neuer Ereignisse basieren auf *Clustering*-Algorithmen. Jedes Cluster steht dabei für ein bestimmtes Thema. Neue Nachrichten werden entweder einem bereits bestehenden Cluster hinzugefügt oder dienen als Ausgangspunkt für ein neues Cluster. Im letzteren Fall ist dies äquivalent mit dem Auftreten einer ersten Nachricht zu einem neuem Thema.

3.2 TREC Novelty Track

Novelty Detection auf Satzebene wurde am stärksten erforscht. Im Zuge der Abspaltung von der *Automatic Text Summarization* entstand der *TREC Novelty Track* erstmals 2002 [10]. Ziel war die Optimierung von Suchergebnissen, die über eine einfache Relevanzsortierung hinausgeht.

Die grundlegende Aufgabe des *Novelty Track* ist es, zu einem gegebenen Thema

Korpus	Themen	Sätze (Anzahl und Verteilung)			
		Insgesamt	nicht-relevant	relevant	relevant & neu
TREC 2002	49	57.227	55.762	1.465	1.241
TREC 2003	50	39.820	15.557	15.557	10.226
TREC 2004	50	52.447	44.104	8.343	3.454

Abbildung 3. Statistik über die Korpora der TREC 2002, 2003 und 2004 Novelty Tracks [20]. Aus dem TREC 2002 wurde ein Thema entfernt, da von den Gutachtern kein einziger Satz als relevant eingestuft wurde. Die jeweils 25 Dokumente pro Thema sind in Sätze unterteilt. Die Spalte „relevante & neue Sätze“ ist in der Menge der relevanten Sätzen enthalten.

und einer sortierten Liste[1] relevanter Dokumente die relevanten und *neuen* Sätze zu finden, die dem Nutzer eines Retrieval-Systems ausschließlich präsentiert werden sollten. Die Erkennung von *Novelty* wurde dafür auf die Satzebene begrenzt. In [10] wird argumentiert, dass durch die Reduktion der Texteinheit eine einfachere Analyse möglich ist. Außerdem ist eine größere Texteinheit, wie beispielsweise ein Absatz, nicht einheitlich definiert.

Ein Thema, für das eine Menge an Suchergebnissen präsentiert wird, ist zum Beispiel Nummer 305 – *Most Dangerous Vehicles* (siehe Anhang auf Seite 14). Neben dem Thema werden eine kurze als auch eine ausführlichere Beschreibung zur Verfügung gestellt. Diese sind später von einem Verfahren zur Erkennung von *Novelty* als Informationsbedarf zu nutzen.

Bevor auf die konkret zu lösenden Aufgaben eingegangen wird, werden zunächst die Korpora der drei *TREC Novelty Tracks* von 2002 bis 2004 vorgestellt [10, 29, 28]. Sie bestehen jeweils aus 50 eben jener vorgestellten Themen inklusive der dazugehörigen relevanten Dokumente. Eine Übersicht wird in Abbildung 3 geboten.

TREC 2002 Novelty Track. Der *TREC Novelty Track* 2002 verwendete Dokumente aus den *TREC*s 6, 7 und 8. Aus 150 zur Verfügung stehenden Themen wurden 50 Themen ausgewählt, die zwischen 10 und 70 relevante Dokumente besaßen. Aus Effizienzgründen wurde die Zahl relevanter Dokumente pro Thema auf 25 begrenzt, die von einem System automatisch ausgewählt wurden. Existierten für ein Thema weniger als 25 relevante Dokumente, so wurden diese durch zufällig ausgewählte themenbezogene Dokumente ergänzt. Anschließend wurden die Dokumente in Sätze unterteilt. Spätere Verfahren zur Erkennung von *Novelty* sollten mit Hilfe eines externen Kriteriums evaluiert werden. Dazu wurden die Sätze der einzelnen Themen jeweils von zwei Gutachtern hinsichtlich der Kategorien *relevant* und *neu* unterschieden.

TREC 2003 Novelty Track. Es ergab sich jedoch das Problem, dass Gutachter in einem vom System als relevant eingestuften Dokument wenig bis keine relevante Sätze gefunden haben. In der Folge wurden nahezu alle relevanten Sätze als neu eingestuft. Die Dokumente wiesen in der Folge zu wenig Redundanz auf. Daher wurden im *Novelty Track* von 2003 alle 50 Themen ersetzt. Diese stammten aus

[1] Die Liste ist nur anhand der Relevanz sortiert, die von einem Retrieval-System vorgegeben wird.

verschiedenen Zeitungen[2] und behandelten zeitgeschichtliche Ereignisse als auch Meinungen über kontroverse Themen. Es wurde ebenso auf die Einhaltung der chronologischen Reihenfolge der Dokumente geachtet, welches beim ersten Korpus von 2002 unterschlagen wurde. Weiterhin gab ein Gutachter selbst das Thema vor und wählte manuell 25 relevante Dokumente aus. Durch diese Vorgehensweise wurden dem Korpus mehr Redundanz und weniger neue Informationen hinzugefügt.

TREC 2004 Novelty Track. Der letzte *TREC Novelty Track* erfuhr wiederum Änderungen im Aufbau des Korpus. Die Anzahl relevanter Sätze wurde deutlich reduziert, indem zusätzlich irrelevante Dokumente aufgenommen wurden.

Es ist festzuhalten, dass der Korpus von 2002 nicht verwendet werden sollte, da hier nahezu alle relevanten Dokumente neu sind. Der Korpus enthält zu wenig Redundanz und entspricht keinem realitätsnahem Szenario.

Von einem zukünftigen Retrieval-System wird erwartet, dass es die folgenden Aufgaben[3] automatisch löst [29]:

Aufgabe 1: Seien 25 relevante Dokumente eines Themas gegeben. Identifiziere alle relevanten und neuen Sätze.

Aufgabe 2: Seien alle relevanten Sätze aus 25 Dokumenten zu einem Thema gegeben. Identifiziere alle neuen Sätze.

Aufgabe 3: Seien alle relevanten und neuen Sätze aus den ersten 5 Dokumenten zu einem Thema gegeben. Identifiziere alle relevanten und neuen Sätze aus den verbleibenden 20 Dokumenten.

Aufgabe 4: Seien alle relevanten Sätze aus 25 Dokumenten eines Themas gegeben. Zusätzlich sind alle neuen Sätze aus den ersten 5 Dokumenten bekannt. Identifiziere alle neuen Sätze in den verbleibenden 20 Dokumenten.

Für ein Experiment wird jeweils ein Thema und eine Menge von Dokumenten zur Verfügung gestellt. Die chronologische Reihenfolge der Dokumente ist ebenfalls bekannt[4]. An den Experimenten beteiligten sich vor allem Universitäten, wodurch eine Vielzahl verschiedener Methoden zur Erkennung von *Novelty* entstanden [36, 37, 38]. Kapitel 4 stellt zunächst den allgemeinen Retrieval-Prozess vor, gefolgt von populären Ansätzen zur *Novelty Detection*.

4. RETRIEVAL-PROZESS UND ALGORITHMEN ZUR NOVELTY DETECTION

Ein traditionelles Informationssystem unterscheidet zwei Kategorien von Dokumenten: *relevant* und *nicht-relevant*. Dabei handelt es sich jeweils um binäre Entscheidungen. Ein Dokument ist relevant oder nicht. Die Relevanzentscheidung fällt zu

[2]Die Dokumente entstammten dem *AQUAINT* Korpus. Dieser enthält Artikel von drei Nachrichtenquellen aus sich überlappenden Zeiträumen: New York Times News Service, AP, Xinhua News Service [9].
[3]Im TREC 2002 *Novelty Track* wurde ausschließlich Aufgabe 1 gestellt.
[4]Im TREC 2002 *Novelty Track* war lediglich die Retrieval-Reihenfolge ausschlaggebend, mit der das System die Dokumente als relevant eingestuft hatte.

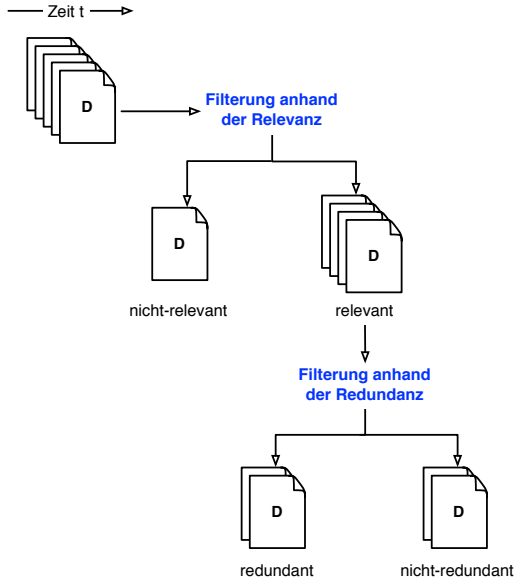


Abbildung 4. Dokumente werden in chronologischer Reihenfolge auf *Novelty* untersucht. Im ersten Schritt findet eine traditionelle Relevanzfilterung statt. Im zweiten Schritt werden aus der Menge relevanter Dokumente jene herausfiltert, die neue Informationen beinhalten (*nicht-redundant*).

jedem Zeitpunkt konsistent aus. Dies ist für die Erkennung von *Novelty* jedoch unzureichend.

Es sind vielmehr drei Kategorien von Dokumenten zu unterschieden: *nicht-relevant*, *redundant*, *nicht-redundant* [43]. Aus Definition 2.1 folgt, dass das Maß der Redundanz asymmetrisch ist. Die Entscheidung *redundant* oder *nicht-redundant* hängt also davon ab, zu welchem Zeitpunkt das Dokument dem Nutzer präsentiert wird. Es müssen Dokumente identifiziert werden, die ähnlich zu zuvor als relevant eingestuften Dokumenten sind, also relevant in Bezug auf die Anfrage. Gleichzeitig dürfen sie nicht ähnlich zu den zuvor als relevant eingestuften Dokumenten im Sinne neuer Informationen sein. Beide Teilaufgaben widersprechen sich. Daher wird ein zweigeteilter Retrieval-Prozess notwendig, der in Abbildung 4 veranschaulicht ist [43].

Relevanzmaße. Zur Ermittlung relevanter Dokumente werden im ersten Schritt traditionelle Retrievalverfahren wie die Euklidische Distanz oder die Kosinusähnlichkeit angewandt. Für zwei Dokumentmodelle $\mathbf{d}_i$ und $\mathbf{d}_j \in \mathbf{D}$ berechnet die Ähnlichkeitsfunktion $\varphi(\mathbf{d}_i, \mathbf{d}_j)$ einen reellen Zahlenwert aus dem Intervall $[0, 1]$ (vgl. Berechnung der Redundanz ρ). Liegt dieser Wert über einem benutzerdefinierten Schwellwert, so ist das Dokument relevant. Es wird folgend davon ausgegangen, dass bereits einer Menge relevanter Dokumente vorliegt.

Redundanzmaße. Basierend auf Definition 2.2 wird ein Maß $\rho(\mathbf{d}_t|\mathbf{D}_r(t))$ gesucht. Diese Forderung ist zu

$$\rho(\mathbf{d}_t|\mathbf{D}_r(t)) = \arg \max_{d_i \in D_r(t)} \rho(\mathbf{d}_t|\mathbf{d}_i) \tag{1}$$

zu vereinfachen. Es ist ausreichend, Redundanz zu messen, indem ein neues Dokument nur mit dem Dokument verglichen wird, welches am ähnlichsten zum neuen

Abbildung 5. Algorithmus, um eine Dokumentmenge D anhand der *Novelty* neu zu sortieren. Eingabe ist eine (nicht sortierte) Dokumentmenge D. Ausgabe ist eine nach *Novelty* sortierte Dokumentmenge R. Für $\rho(\mathbf{d}_i|\mathbf{d}_i)$ ist eines der in den Abschnitten 4.2 bis 4.4 vorgestellten Verfahren einzusetzen.

SortiereNachNovelty(D,n)
  $R \leftarrow \emptyset$
  for j 1 **to** $|D|$ **do**
    $d_j \leftarrow \arg\max_{d_i \in D_r(t)} \rho(\mathbf{d}_t|\mathbf{d}_i)$
    R $\leftarrow R \cup \{d_j\}$
    D $\leftarrow D \setminus \{d_j\}$
  end for

Dokument ist. Für $\mathbf{d}_t = \mathbf{d}_i$ wird $\rho(\mathbf{d}_t|\mathbf{d}_i)$ maximal. Um ρ zu messen, sind intuitiv wiederum Distanz- oder Ähnlichkeitsmaße anwendbar.

In jedem Schritt ist ein neues Dokument also mit dem ähnlichsten Dokument zu vergleichen, welches zuvor betrachtet wurde. Bei sehr vielen Dokumenten ist es aus Effizienzgründen sinnvoll, Kandidaten für das ähnlichste Dokument auf die N zum neuen Dokument zeitlich am Nähesten zu begrenzen. Annahme hierbei ist, dass Dokumente redundanter zu zeitnah erschienenen Dokumenten sind.

Ein einfacher Algorithmus, um relevante Dokumente anhand der *Novelty* zu sortieren, wird in Abbildung 5 vorgestellt.

Tabelle I auf Seite 9 bietet eine Übersicht über verbreitete Retrieval-Modelle, die in der *Novelty*-Forschung eingesetzt werden. In den folgenden Abschnitten werden einige Maße daraus vorgestellt. Für das Maß der Redundanz ρ sind entsprechend der Ähnlichkeitsfunktion φ Dokumentmodelle $\mathbf{d} \in \mathbf{D}$ zu verwenden.

4.1 Kosinusähnlichkeit

Für jedes Dokument $d_i \in D$ wird jedem Wort w_k ein Gewicht $w_{i,k} \in \{0, 1\}$ zugewiesen. Intuitiv wird w_k, $0 \leq k \leq m$, ein Gewicht von 1 zugeordnet, wenn es im Dokument enthalten ist, ansonsten 0. Fasst man die Gewichte zu einem Vektor $\mathbf{d}_i = \{w_{i,1}, w_{i,2}, \ldots, w_{i,m}\}$ zusammen, beschreibt er das Dokument d_i.

Aufgrund der Tatsache, dass Textdokumente durch Vektoren dargestellt werden, ist die Ähnlichkeit $\varphi(\mathbf{d}_t, \mathbf{d}_i)$ zweier Gewichtsvektoren durch den Kosinus berechenbar:

$$\varphi(\mathbf{d}_t, \mathbf{d}_i) = \frac{\langle \mathbf{d}_t, \mathbf{d}_i \rangle}{|\mathbf{d}_t| \times |\mathbf{d}_i|}$$

Je geringer der Winkel ausfällt, desto ähnlicher sind sich die beiden Dokumente.

Aufgrund der weiten Verbreitung der Kosinusähnlichkeit im Information Retrieval wird diese auch im Bereich der *Novelty Detection* vielfach angewandt und führt dabei zu überzeugenden Ergebnissen [1, 43]. Dabei wird entsprechend des vorgestellten Retrieval-Prozesses im Relevanzschritt $\varphi(\mathbf{d}_t, \mathbf{d}_i)$ berechnet. Im darauf folgenden Redundanzschritt

$$\rho(\mathbf{d}_t|\mathbf{d}_i) = 1 - \varphi(\mathbf{d}_t, \mathbf{d}_i).$$

Tabelle I. Klassifikation existierender Verfahren zur *Novelty Detection*

Document retrieval model $\mathcal{R}$			**Level**	**Evaluation**	**Notes**	**Reference(s)**
Repr. d	**Relevance filtering**	**Novelty filtering**		**corpus**		
LM	–	MM, KLD	Sentence	[35]		[41]
LM	TFISF	MM, KLD	Sentence	[10, 28, 29]	Based on [41]	[6]
LM	MBD	MBD	Sentence	[10]		[24]
NER+POS	CS	CS	Sentence	[28]		[44]
VSM, LM	CS+PRF	CS, SCMM, DS, SS, NW, SD, IAS	Sentence	[10]		[1, 43]
VSM	CS+RF	DN-Add, DN-Step, MMR-Add, MMR-Step	Document	own		[39]
VSM	CS	MMR	Document	–	One-Stage Task	[2]
VSM	CS	SD+LCA, NW+LCA	Sentence	[10, 28, 29]	No improvements compared to TREC 2002-2004	[5]
VSM	CS	CS	Sentence	[32]		[31]
	DE	DE	Sentence	[29]	TREC 2003	[23]
VSM	CS	CS, CA, TFIDF-based	Document	[32, 33, 34]		[4]
VSM	TFIDF-based	TFIDF-based	Document	own	No comparison to other approaches	[12]
VSM	CS	CFS, MMR	Sentence	[29]	TREC 2003	[26]
VSM	X2, CA	MMR	Sentence	[29]	TREC 2003	[30]
VSM	CS	CS	Sentence & Document	[10, 29]		[40]
VSM	–	NDF	Document	[19]		[14]
WW	TFISF	NE+IP	Sentence	[10, 28, 29]	Significant improvements	[20, 21, 22]
WW	–	TFIDF-based	Sentence	[28, 29]	No improvements over TREC 03	[8]
WW	TFIDF+RF, WB+RF	NID	Sentence	[29]	TREC 2003	[13]
WW	–	PRF	Sentence	[10, 28, 29]		[25]
WW	DFM	CA	Document	[3]	TDT	[15, 16, 17]
WW	PIRCS	DC	Sentence	[10]	TREC 2002	[18]

Tabelle II. Beschreibung der Abkürzungen von Tabelle I

ID	Description
Representation d	
NER	Named entity recognition
POS	Part-of-speech tagging
VSM	Vector Space Model
WW	Word Weights (Boolean, Frequency)
Relevance filtering	
CA	Cluster-based Algorithm
CS	Cosine Similarity
DE	Discourse entities and antecedents
DFM	Document Forgetting Model (probabilistic)
MBD	Multiple Bernoulli Distribution
PIRCS	a Network-Based Document Routing and Retrieval System
PRF	Pseudo-Relevance Feedback
RF	Relevance Feedback
TFIDF	Term frequency – inverse document frequency
TFISF	Term frequency – inverse sentence frequency
WB	Window-based Methods
X2	$\mathcal{X}^2$ statstic
Novelty filtering	
CA	Cluster-based Algorithm
CFS	Conceptual Fuzzy Sets
CS	Cosine Similarity
DC	Dice's Coefficient
DE	Discourse entities and antecedents
DN-Add	Additive Directed Novelty
DN-Step	Stepwise Directed Novelty
IAS	Interpolated Aggregate Smoothing
IP	Information Patterns
LCA	Local Context Analysis
KLD	Kullback-Leibler Diver
MBD	Multiple Bernoulli Distribution
MM	Mixture Model
MMR	Maximal Marginal Relevance
MMR-Add	Additive Maximal Marginal Relevance
MMR-Step	Stepwise Maximal Marginal Relevance
NDF	Novelty detector filter (machine learning approach)
NID	New Information Degree (NID)
NW	New Word Count
PRF	Pseudo-Relevance Feedback
SCMM	Sentence Core Mixture Model
SD	Set Difference
SS	Shrinkage Smoothing

4.2 Maximum-Marginal Relevance

Dieses Verfahren kombiniert mittels linearer Kombination die beiden erwähnten Berechnungsschritte:

$$\rho(\mathbf{d}_t|\mathbf{D}_r(t)) = \lambda * \varphi_1(\mathbf{d}_t, \mathbf{q}) - (1-\lambda) * \max_{d_i \in D(t)} \varphi_2(\mathbf{d}_t, \mathbf{d}_i)$$

Hierbei wird zunächst die Relevanz einer Anfrage[5] q zum Dokument d_i berechnet, gewichtet mit λ, und anschließend $\mathbf{d}_i$ auf Redundanz mit allen zuvor bereits gesehenen Dokumenten verglichen. Der Parameter λ steuert dabei die Präferenz von Relevanz oder *Novelty*. Wird $\lambda = 1$ gewählt, so stellt sich die traditionelle Retrievalsituation ein, bei der Dokumente anhand ihrer Relevanz sortiert werden. Ist $\lambda = 0$, so liegt das Augenmerk auf der Erkennung von *Novelty*. Dies resultiert in einer Ergebnisliste, die stark differenzierte Dokumente enthält (vgl. Breitensuche in Abschnitt 1.2).

MMR findet breite Verwendung sowohl im *Text Summarization* Bereich als auch in der *Novelty-Detection* und liefert gute Ergebnisse [2, 20]. In der Regel wird die Kosinusähnlichkeit sowohl für φ_1 als auch für φ_2 eingesetzt. Beide Funktionen müssen jedoch nicht zwingend dieselbe Ähnlichkeitsfunktion verwenden.

4.3 Sprachmodelle

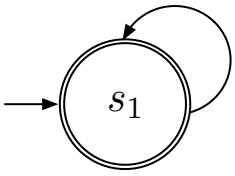
Ebenfalls häufig werden Sprachmodelle eingesetzt, um Dokumente zu repräsentieren. Ein Sprachmodell weißt aufgrund einer Wahrscheinlichkeitsverteilung jedem Wort w_k eines Dokumentes eine Wahrscheinlichkeit zu. Idee ist, dass ein Dokument d_i relevant zu einer Anfrage q ist, wenn die Wahrscheinlichkeit $P(q|M_d)$ hoch ist, das ein zugrundeliegendes Dokumentmodell M_d die Anfrage erzeugt hat. Dokumente sind demnach anhand von Sprachmodellen sortierbar.

Ein einfaches Dokumentmodell stellt das Unigramm Sprachmodell M_{uni} dar, welches in Abbildung 6 erklärt wird. Für das gezeigte Beispiel in Abbildung 6 ergibt sich für den Satz *Der Teufel steckt im Detail* unter M_{uni} durch Multiplikation der Einzelwahrscheinlichkeiten der Wörter folgende Wahrscheinlichkeit:

$$\begin{aligned} P(\text{„Der Teufel steckt im Detail“}) = &(0.1 * 0.01 * 0.03 * 0.05 * 0.01) * \\ &(0.8 * 0.8 * 0.8 * 0.8 * 0.2) \end{aligned}$$

Es sind nicht nur Dokument-Sprachmodelle M_d (engl. *query-likelihood model*) vorstellbar, die eine Anfrage erzeugen, sondern auch Anfrage-Sprachmodelle M_q

[5] Die Anfrage q wird ebenfalls wie Dokumente durch ein Modell $\mathbf{Q}$ repräsentiert



im	0.05
Detail	0.01
Teufel	0.01
steckt	0.03
der	0.1
...	...

Abbildung 6. Das Unigramm Sprachmodell ist als endlicher Automat mit einem Zustand s_1 darstellbar. s_1 besitzt eine Verteilungsfunktion, die bestimmt, mit welcher Wahrscheinlichkeit ein Term w_k erzeugt wird. Nachdem ein Term generiert wurde, wird mit einer Wahrscheinlichkeit $P(\text{STOP}|s_1)$ entschieden, ob der Automat in den Endzustand wechselt oder einen weiteren Term erzeugt.

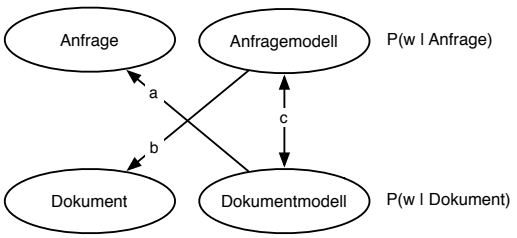


Abbildung 7. Die Abbildung verdeutlich drei Herangehensweisen, um ein Sprachmodell zu entwickeln: (a) Query likelihood, (b) Document likelihood, (c) Vergleich beider Modelle.

(engl. *document-likelihood model*), die Dokumente erzeugen. Weiterhin ist ein Sprachmodell für eine Anfrage als auch das Dokument definierbar. Diese Zusammenhänge sind nochmals in Abbildung 7 verdeutlicht. Im letzteren Fall ist relevant, wie stark sich beide Modelle voneinander unterscheiden.

Aus der Informationstheorie wird bekannte Kullback-Leibler (KL) Divergenz verwendet, die zwischen zwei Sprachmodellen berechnet wird:

$$\rho(\mathbf{d}_t|\mathbf{q}) = KL(M_d \| M_q) = \sum_{w \in W} P(w|M_q) \log \frac{P(w|M_q)}{P(w|M_d)}$$

Die KL-Divergenz wird zur *Novelty Detection* eingesetzt [6, 24, 43]. Die verschiedenen Ansätze unterscheiden sich in der Wahl der Sprachmodelle.

4.4 Simple New Word Count

Die Differenz zweier Dokumente $\mathbf{d}_t, \mathbf{d}_i \in \mathbf{D}$ ergibt sich mittels

$$\|\mathbf{d}_t \cap \overline{\mathbf{d}_i}\| + \|\overline{\mathbf{d}_t} \cap \mathbf{d}_i\|$$

und umfasst alle Terme, die jeweils nur in einem von beiden Dokumenten vorkommen. Für das Maß der *Novelty* ist jedoch die Menge $\|\overline{\mathbf{d}_t} \cap \mathbf{d}_i\|$ vernachlässigbar, da nur neue Wörter im aktuell zu untersuchenden Dokument d_t von Interesse sind. Die *Novelty*-Bewertung für das Dokument d_t ergibt sich also durch die Anzahl der Wörter, die zuvor in keinem anderen Dokument vorgekommen sind:

$$\rho(\mathbf{d}_t|\mathbf{D}_r(t)) = \|\mathbf{d}_t \cap \bigcup_{i=1}^{t-1} \overline{\mathbf{d}_i}\|$$

Dieses Maß ist intuitiv, liefert jedoch schlechtere Ergebnisse bei der Erkennung von *Novelty* im Vergleich zu zuvor vorgestellten Verfahren [1, 5].

4.5 Set Difference

Im Gegensatz zum *Simple New Word Count*-Maß beschränkt sich die *Set Difference* bei der Berechnung der *Novelty* auf den Vergleich eines einzigen Dokumentpaares (siehe Formel 1). Vergleicht wird ein neues Dokument $\mathbf{d}_t$ ausschließlich mit dem am ähnlichsten zuvor betrachteten $\mathbf{d}_i$, $j < t$:

$$\rho(\mathbf{d}_t|\mathbf{d}_i) = \|\mathbf{d}_t \cap \overline{\mathbf{d}_i}\|$$

Es gilt $w_k \in \mathbf{d}_t$ genau dann, wenn sein Beitrag $Count(w_k, d_t) > l$, $l \geq 0$. Der Beitrag eines Worts im Dokument berechnet sich durch

$$Count(w_k, d_t) = \alpha * \mathrm{tf}_{w_k, d_t} + \beta * \mathrm{df}_{w_k} + \gamma * \mathrm{rdf}_{w_k}$$

mit

tf_{w_k,d_t} : Frequenz des Terms w_k in Dokument d_t.
df_{w_k} : Anzahl gefilterter Dokumente, die den Term w_k enthalten.
rdf_{w_k} : Anzahl relevanter Dokumente, die den Term w_k enthalten.

Anstatt jeden Term gleichwertig zu betrachten (vgl. *Simple New Words Count*), wird jeder Term gewichtet.

Ein Nachteil dieses Verfahrens ist, dass es parametrisiert ist. Das heißt, die Variablen α, β, γ und l müssen bestimmt werden. Diese sind mit Hilfe von Trainingsdaten zu erlernen. In [43] werden sie wie folgt festgelegt: $\alpha = 0.8, \beta = 0.2, \gamma = 0.0, l = 2$.

5. ZUSAMMENFASSUNG

Zunächst wurde die Notwendigkeit einer Erkennung von *Novelty* und der daraus folgenden Neusortierung von Suchergebnissen motiviert. Im Zuge wurden zwei Szenarien skizziert, in denen *Novelty Detection* einem Nutzer helfen kann, seinen Informationsbedarf zu maximieren: *directed* und *undirected*.

Forscher beschäftigen sich insbesondere in den Jahren 2002 bis 2004 damit, neue Ansätze zur *Novelty Detection* zu entwickeln. Dies wurde vom *TREC Novelty Track* begleitet. Entsprechende zu lösende Aufgaben wurden vorgestellt, als auch die Zusammensetzung dreier Korpora zur Evaluierung der Verfahren.

Zuletzt wurde eine Übersicht über verschiedene Ansätze geboten und davon ausgewählte Verfahren vorgestellt.

Obwohl gezeigt werden konnte, dass der Einsatz von *Novelty Detection* in Retrieval-System sinnvoll und zu besseren Suchergebnissen führt, überrascht es, dass gegenwärtig kaum davon gebrauch gemacht wird.

ANHANG

<top>
<num> Number: 305
<title> Most Dangerous Vehicles

<desc> Description: Which are the most crashworthy, and least crashworthy, passenger vehicles?

<desc2> Description: Which are the most crashworthy, and least crashworthy, passenger vehicles?

<narr> Narrative:
A relevant document will contain information on the crashworthiness of a given vehicle or vehicles that can be used to draw a comparison with other vehicles. The document will have to describe/compare vehicles, not drivers. For instance, it should be expected that vehicles preferred by 16-25 year-olds would be involved in more crashes, because that age group is involved in more crashes. I would view number of fatalities per 100 crashes to be more revealing of a vehicle's crashworthiness than the number of crashes per 100,000 miles, for example.

<relevant>
LA031689-0177
FT922-1008
LA090190-0126
LA101190-0218
LA082690-0158
LA112590-0109
FT944-136
LA020590-0119
FT944-5300
LA052190-0048
LA051689-0139
FT944-9371
LA032390-0172
LA042790-0172
LA021790-0136
LA092289-0167
LA111189-0013
LA120189-0179
LA020490-0021
LA122989-0063
LA091389-0119
LA072189-0048
FT944-15615
LA091589-0101
LA021289-0208
</top>

Abbildung 8. Beispiel für die Beschreibung von Themen anhand von *Most Dangerous Vehicles*. 25 zum Thema relevante Dokument-IDs sind zuletzt angeführt.

REFERENZEN

[1] J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. *Proceedings of the 26th annual international ACM SIGIR ...*, Jan 2003.

[2] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. *Proceedings of the 21st annual international ACM SIGIR ...*, Jan 1998.

[3] C. Cieri, D. Graff, M. Liberman, N. Martey, and S. Strassel. The tdt-2 text and speech corpus. *Broadcast News Workshop'99 Proceedings*, Jan 1999.

[4] W. Dai and R. Srihari. Minimal document set retrieval. *Proceedings of the 14th ACM international conference on ...*, Jan 2005.

[5] R. Fernández and D. Losada. Novelty detection using local context analysis. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 725–726, 2007.

[6] R. Fernández and D. Losada. Novelty as a form of contextual re-ranking: efficient kld models and mixture models. *Proceedings of the second international symposium on Information interaction in context*, pages 27–34, 2008.

[7] E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. *Proceedings of the 13th international conference on ...*, Jan 2004.

[8] G. Gaughan and A. Smeaton. Finding new news: Novelty detection in broadcast news. *LECTURE NOTES IN COMPUTER SCIENCE*, 3689:583, 2005.

[9] D. Graff. The AQUAINT corpus of English news text. *Linguistic Data Consortium*, (Technical Report LDC20002T31), 2002.

[10] D. Harman. Overview of the trec 2002 novelty track. *The Eleventh Text REtrieval Conference, TREC 2002*, pages 46–56, 2002.

[11] Y. Ishikawa, Y. Chen, and H. Kitagawa. An on-line document clustering method based on forgetting factors. *LECTURE NOTES IN COMPUTER SCIENCE*, Jan 2001.

[12] F. Jacquenet and C. Largeron. Using the structure of documents to improve the discovery of unexpected information. *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1036–1042, 2006.

[13] Q. Jin, J. Zhao, and B. Xu. Nlpr at trec 2003: Novelty and robust. *NIST Special Publication: SP*, Jan 2003.

[14] R. Kassab and J. Lamirel. An innovative approach to intelligent information filtering. *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1089–1093, 2006.

[15] S. Khy, Y. Ishikawa, and H. Kitagawa. Incremental clustering based on novelty of on-line documents. *DBSJ Letters*, 5(1), 2006.

[16] S. Khy, Y. Ishikawa, and H. Kitagawa. Novelty-based incremental document clustering for on-line documents. *Proc. of International Workshop on Challenges in Web ...*, Jan 2006.

[17] S. Khy, Y. Ishikawa, and H. Kitagawa. A novelty-based clustering method for on-line documents. *World Wide Web*, 11(1):1–37, 2008.

[18] K. Kwok, P. Deng, N. Dinstl, M. Chan, and Q. C. F. N. D. O. C. SCIENCES. Trec2002 web, novelty and filtering track experiments using pircs. 2002.

[19] D. Lewis. Reuters-21578 text categorization test collection. 2004.

[20] X. Li. Sentence level information patterns for novelty detection. 2006.

[21] X. Li and W. Croft. Novelty detection based on sentence level patterns. *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 744–751, 2005.

[22] X. Li and W. Croft. Improving novelty detection for general topics using sentence level information patterns. *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 238–247, 2006.

[23] K. Litkowski. Use of metadata for question answering and novelty tasks. *Proceedings of the Twelfth Text REtrieval Conference (TREC ...*, Jan 2004.

[24] D. Losada. Language modeling for sentence retrieval: A comparison between multiple-bernoulli models and multinomial models. *Information Retrieval and Theory Workshop*, 2005.

[25] D. Losada and R. Fernandez. Highly frequent terms and sentence retrieval. *LECTURE NOTES IN COMPUTER SCIENCE*, 4726:217, 2007.

[26] R. Ohgaya, A. Shimmura, T. Takagi, and A. Aizawa. Meiji university web and novelty track experiments at trec 2003. *Proceedings of the Twelth Text Retrieval Conference (TREC)*, Jan 2003.

[27] R. Papka and J. Allan. Topic detection and tracking: Event clustering as a basis for first story detection. *Kluwer Academic Publishers*, Jan 2000.

[28] I. Soboroff. Overview of the trec 2004 novelty track. *Proceedings of the 13th Text REtrieval Conference (TREC 2004)*, 2004.

[29] I. Soboroff and D. Harman. Overview of the trec 2003 novelty track. *Proceedings of TREC-2003*, 2003.

[30] J. Sun, Z. Y. 0002, W. Pan, H. Zhang, B. Wang, and X. Cheng. Trec 2003 novelty and web track at ict. pages 138–146, 2003.

[31] M. Tsai, M. Hsu, and H. Chen. Approach of information retrieval with reference corpus to novelty detection. *Voorhees and Harman [13]*, 2003.

[32] E. Voorhees and D. Harman. Overview of the 6th text retrieval conference. *Proceedings of the 6th Text Retrieval Conference TREC-6*, Jan 1998.

[33] E. Voorhees and D. Harman. Overview of the seventh text retrieval conference. *Proceedings of the Seventh Text REtrieval Conference (TREC-7 . . .*, Jan 1999.

[34] E. Voorhees and D. Harman. Overview of the eighth text retrieval conference (trec-8). *NIST SPECIAL PUBLICATION SP*, Jan 2000.

[35] E. Voorhees and D. Harman. Overview of the ninth text retrieval conference (trec-9). pages 1–13, Aug 2001.

[36] E. Voorhees and D. Harman. Proceedings of the Eleventh Text REtrieval Conference (TREC 2002). *Gaithersbourg, MD*, 2002.

[37] E. Voorhees and D. Harman. Proceedings of the Twelfth Text REtrieval Conference (TREC 2003). *Gaithersbourg, MD*, 2003.

[38] E. Voorhees and D. Harman. Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004). *Gaithersbourg, MD*, 2004.

[39] Y. Xu and H. Yin. Novelty and topicality in information retrieval. 2006.

[40] Y. Yang, A. Lad, N. Lao, A. Harpale, B. Kisiel, and M. Rogati. Utility-based information distillation over temporally sequenced documents.

[41] C. Zhai, W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. *. . . on Research and development in informaion retrieval*, Jan 2003.

[42] J. Zhang, Z. Ghahramani, and Y. Yang. A probabilistic model for online document clustering with application to novelty detection. *Advances in Neural Information Processing Systems*, 17:1617–1624, 2005.

[43] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 81–88, 2002.

[44] Y. Zhang and F. Tsai. Combining named entities and tags for novel sentence detection. *Proceedings of the WSDM'09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 30–34, 2009.