
Entwicklung und wirtschaftliche Evaluation eines personalisierten Nachrichtenaggregators

Masterthesis

Christian Kulas | 1410987

Wirtschaftsinformatik



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Christian Kulas
Matrikelnummer: 14010987
Studiengang: Wirtschaftsinformatik (M.Sc.)

Masterthesis
Thema: Entwicklung und wirtschaftliche Evaluation eines personalisierten
Nachrichtenaggregators

Eingereicht: 31. Oktober 2013

Prof. Dr. Alexander Benlian
Fachgebiet Information Systems and Electronic Services / Wirtschaftsinformatik
Fachbereich Rechts- und Wirtschaftswissenschaften

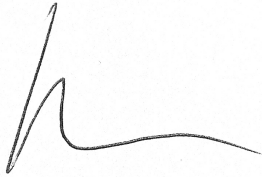
Prof. Dr. Johannes Fürnkranz
Fachgebiet Knowledge Engineering Group
Fachbereich Informatik

Technische Universität Darmstadt
Hochschulstraße 1
64289 Darmstadt

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig angefertigt habe. Sämtliche aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

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

A handwritten signature in black ink, consisting of a stylized 'h' followed by a horizontal line that tapers to the right.

Darmstadt, den 31. Oktober 2013

Inhaltsverzeichnis

Abbildungsverzeichnis.....	III
Tabellenverzeichnis.....	VI
Abkürzungsverzeichnis.....	VII
1 Einleitung.....	1
1.1 Motivation	2
1.2 Ziele und Anforderungen.....	2
1.3 Aufbau.....	4
2 Grundlagen	5
2.1 Grundlegende Begriffe: Individualisierung, Personalisierung, Customizing	5
2.2 Nutzen von Personalisierung im Allgemeinen und bei Nachrichten.....	6
2.3 Recommender Systems.....	8
2.3.1 TfIdf Repräsentation	9
2.3.2 kNN.....	10
2.3.3 Perceptron	10
2.3.4 Hybride Systeme	11
2.4 Nachrichtenfeeds	12
2.5 Evaluierung	12
2.5.1 Precision, Recall, F-Measure	14
2.5.2 Mean Squared Error, Root Mean Squared Error	15
2.5.3 Mean Average Precision.....	15
2.5.4 Normalized Discounted Cumulative Gain	16
2.5.5 Prequential (Interleaved Test-Then-Train).....	17
2.6 State-of-the-Art.....	17
2.6.1 Personalisierung bei Nachrichten.....	18
2.6.2 Zahlungsbereitschaft.....	18
2.6.3 Implizites Feedback	21
2.6.4 Personal Newsaggregators und Abgrenzung dazu	24
2.7 Beispiele aus der Praxis	26
2.7.1 Prismatic	26
2.7.2 Piano Media	29
2.7.3 Feedly	30
3 Konzeptuelles Design	32
3.1 Anforderungen des Nutzers an die persönliche Zeitung	32
3.2 Gesamtarchitektur	34
3.3 Auswertung von RSS-Feeds	36
3.4 Erfassung der Nutzerdaten	38
3.5 Strategien für implizites Feedback.....	39
3.6 Klassifikation von Nachrichtenartikeln	44
3.6.1 TfIdf Nutzer-Term-Repräsentation	44
3.6.2 Simple Scoring.....	47
3.6.3 Perceptron	48
3.6.4 kNN.....	50

3.6.5	Duplikateerkennung	51
3.6.6	Zusammenfassung	52
3.7	Ranking	53
3.7.1	Auswertung der Daten aus sozialen Netzwerken	54
3.7.2	Auswertung journalistisch kuratierter Daten.....	56
3.7.3	Andere Ranking-Faktoren	57
3.7.4	Kombination der Ergebnisse	59
4	Konkrete technische Umsetzung	61
4.1	Verwendete Infrastruktur	61
4.1.1	Programmiersprache.....	61
4.1.2	Datenbank	62
4.1.3	Frameworks, Tools und Bibliotheken.....	62
4.2	Erfassung der Nachrichtenartikel (Crawler)	63
4.3	Vorbereitung der Inhaltsanalyse	66
4.4	Nutzeroberfläche	67
5	Evaluation der technischen Umsetzung.....	70
5.1	Testdaten.....	70
5.2	Methodik der Evaluierung	74
5.3	Evaluierung des Nutzer Feedbacks	74
5.4	Evaluierung der Klassifizierungsalgorithmen.....	77
5.5	Evaluierung des Rankings.....	80
6	Daten für Feldstudie zur Zahlungsbereitschaft.....	84
6.1	Registrierungsprozess	85
6.2	Personalisierungsmöglichkeiten.....	86
6.3	Customizing.....	88
6.4	Nutzungsschranke für finale Umfrage	89
6.5	Ausführliche Umfrage nach Nutzung.....	90
6.6	Zusammenfassung	91
7	Resultate der Feldstudie.....	92
7.1	Deskriptive Auswertung der Zahlungsbereitschaft.....	92
7.1.1	Evaluation der Vorbefragung	94
7.1.2	Evaluation nach der Nutzung.....	95
7.1.3	Zusammenfassung	97
8	Fazit und Ausblick.....	98
8.1	Technische Fragestellung	98
8.2	Weiterentwicklung	99
8.3	Wirtschaftliche Fragestellung	100
8.4	Weitere Studien.....	102
	Anhang.....	VIII
	Literaturverzeichnis	XXVII

Abbildungsverzeichnis

Abbildung 3-2:Konzeption der Gesamtarchitektur	35
Abbildung 4-2: Crawler Architektur	63
Abbildung 4-3: Portal für den Evaluierungsexperten.	68
Abbildung 4-4: Artikeldetailansicht für Evaluierungsexperten.	68
Abbildung 5-1: Überblick zur Expertenbewertung.	72
Abbildung 5-2: Feedback Generierung durch Experten in absoluten Zahlen.....	73
Abbildung 5-3: Übersicht über Kombinationen aus impliziten Strategien für Chunks mit der Größe 70 und Bewertung größer 1.....	75
Abbildung 5-4: Übersicht ohne explizites Feedback.	75
Abbildung 5-5: Precision nach jedem Chunk mit der Größe 70 für alle Artikel, die nicht als uninteressant bewertet wurden.	76
Abbildung 5-6: Evaluation für kNN exklusiv, nur wortbasierte Features.....	77
Abbildung 5-7: kNN für Chunk-Größe von 70 und Bewertungen größer 0.	78
Abbildung 5-8:kNN exklusiv, alle Features mit Gewichtung von 0,5.....	78
Abbildung 5-9: Evaluation für Simple TfIdf exklusiv.....	79
Abbildung 5-10: Evaluation für Perceptron exklusiv, nur Wörter.....	79
Abbildung 5-11: Perceptron exklusiv, alle Features.	80
Abbildung 5-12: kNN exklusiv, sortiert ausschließlich nach startPageScore.	81
Abbildung 5-13: Simple TfIdf exklusiv, sortiert ausschließlich nach startPageScore.....	82
Abbildung 5-14: Perceptron exklusiv, sortiert ausschließlich nach startPageScore.	82
Abbildung 5-15: kNN exklusiv, Social Score Sortierung.....	83
Abbildung 6-1: Quellenauswahl im Registrierungsprozess.....	86
Abbildung 6-2: Explizites Feedback und Personalisierungselemente.....	87
Abbildung 6-3: Kanäle mit persönlichen Schwerpunkten.....	88
Abbildung 6-4: Oberfläche mit Customizing-Elementen	89
Abbildung 6-5: Fragebogenelement in Artikelliste	90
Abbildung 6-6: Zusammenfassung zu unterschiedlicher Personalisierung der Testgruppen. ..	91

Abbildung 7-1: Getrennte Zahlungsbereitschaften der Testgruppen, einschließlich Variationskoeffizienten (ZB) und Zufriedenheiten.	95
Abbildung 8-1: Newspaper Association of America, Einnahmen von US Zeitungen mit Print- und Onlinewerbung seit 2003	VIII
Abbildung 8-2: AGOF 2012, Nutzung von Themen und Angeboten des Internets.....	VIII
Abbildung 8-3: AGOF 2012, Top-Themen nach Geschlecht	IX
Abbildung 8-4: Boston Consulting Group, Willingness to pay for news online: Key findings from an international survey	IX
Abbildung 8-5: Boston Consulting Group (BCG), Willingness to pay for news online, German Customers.....	X
Abbildung 8-6: Boston Consulting Group (BCG), Willingness to pay for news online, European customers	X
Abbildung 8-7:Allensbachstudie, ACTA 2012.....	XI
Abbildung 8-8: PWC, WAN. Moving into Multiple Business Models: Outlook for Newspapers Publishing in the Digital Age, 2009	XII
Abbildung 8-9: Fragebogen bei Registrierungsprozess	XIII
Abbildung 8-10: Statistische Auswertung über alle Teilnehmer nach der Umfrage.....	XV
Abbildung 8-11: Statistische Auswertung über Gruppe A nach der Umfrage.	XVI
Abbildung 8-12: Statistische Auswertung über Gruppe B nach der Umfrage.	XVIII
Abbildung 8-13: Statistische Auswertung über Gruppe C nach der Umfrage.	XIX
Abbildung 8-14: Statistische Auswertung über Gruppe D nach der Umfrage.	XX
Abbildung 8-15: Implizites Feedback, Bewertung größer 0.....	XXI
Abbildung 8-16: Implizites Feedback, Bewertung größer 1.....	XXII
Abbildung 8-17:Implizites Feedback, Bewertung größer 2.....	XXIII
Abbildung 8-18: Ausschließlich implizites Feedback.....	XXIV
Abbildung 8-19: Implizite Feedback Strategien, Vergleich Precision / Chunks mit F=105 und Bewertung > 1	XXV
Abbildung 8-20: Implizite Feedback Strategien, Vergleich Precision / Chunks mit F=70 und Bewertung > 1	XXV



Abbildung 8-21: Implizite Feedback Strategien, Vergleich Precision / Chunks mit $F=70$ und
Bewertung = 3 XXVI

Abbildung 8-22: kNN, Sortierung nach socialScore mit socialScore größer 2. XXVI

Tabellenverzeichnis

Tabelle 1: Konfusionsmatrix.....	14
Tabelle 2: Implizite Feedback Strategien.	43
Tabelle 3: Herkunft und Häufigkeit der Evaluierungsartikel.....	71
Tabelle 4: ANOVA zu Zahlungsbereitschaft und Gruppen.....	XIV
Tabelle 5: Bonferroni-Test der Zahlungsbereitschaft nach Gruppen	XIV

Abkürzungsverzeichnis

PNA	Personalized News Aggregator
SVM	Support Vector Machine
IR	Information-Retrieval
kNN	k-Nearest Neighbor
Tfidf	Term Frequency Inverse Document Frequency
GTN	General Topic News

1 Einleitung

„Das Internet ist ein großer Misthaufen, in dem man allerdings auch kleine Schätze und Perlen finden kann.“ – Joseph Weizenbaum (1923-2008)

Die Entwicklung des Internets in den vergangenen Jahren hat dazu geführt, dass Nutzer heute Zugang zu mehr Informationen denn je haben. Sie werden überrollt von einer Informationsflut, bei der es für sie immer schwieriger und aufwändiger wird, zwischen Wichtigem und Unwichtigem zu unterscheiden.

In vielen Bereichen entgegnen wir dieser Problematik mit algorithmischen Lösungen. Das E-Commerce, dabei vor allem das Unternehmen Amazon, ist hier als einer der Vorreiter zu nennen, vorhandene Daten individuell maschinell für jeden Nutzer auszuwerten. Jeff Bezos, der Gründer von Amazon, sah es als seine Mission, für eine Millionen Kunden, eine Millionen verschiedene Shops anzubieten.¹

Diese Arbeit fokussiert sich auf die Medienbranche, speziell journalistische Inhalte. Dort wird bis heute die Relevanz eines Artikels über den sog. Nachrichtenwert bestimmt, der von der einem Redakteur oder Journalisten kuratiert wird.

Nachrichtenaggregatoren, wie Google News, führen Artikel aus diversen Quellen zusammen und geben dem Nutzer einen Überblick über das aktuelle Geschehen. Sie berücksichtigen dabei allerdings nur unzureichend die persönlichen Präferenzen eines jeden Nutzers. Dies übernimmt die spezielle, noch junge Form der personalisierten Nachrichtenaggregatoren (PNA).

In dieser Arbeit sollen zunächst Algorithmen des maschinellen Lernens vorgestellt werden, die eine Adaption der Nutzerbedürfnisse ermöglichen und dies anschließend validiert werden. In einem zweiten Teil soll über eine Feldstudie festgestellt werden, welchen Grad der Personalisierung von Nutzern gewünscht ist.

¹ Jeff Bezos, Washington Post, 1998: „If we have 4.5 million customers, we shouldn't have one store. We should have 4.5 million stores.“ (<http://www.washingtonpost.com/wp-srv/washtech/daily/nov98/amazon110898.htm>, Abruf 17.9.2013)

1.1 Motivation

Neben der **Fülle an Informationen**, die uns in den letzten Jahren mehr und mehr durch das Internet zum Abruf bereitgestellt wurden, haben sich weitere Trends entwickelt, die die aktuelle Brisanz von PNA unterstreichen.

Trend zu Mobilgeräten: Ein großer Trend ist die Verschiebung des Nachrichtenkonsums und der Reifegrad der heutigen Mobilgeräte (Smartphones, Tablets), wo die Ressourcen (Displaygröße, Rechenkapazität, Internetverbindung) immer noch weitaus beschränkter sind als am Heimcomputer, weshalb sich die Informationsüberladung noch stärker auswirkt. Diese Geräte bieten für Entwickler außerdem die Möglichkeit, den Kontext bzw. den wechselnden Standort miteinzubeziehen.

Verändertes Konsumverhalten: Ein weiterer großer Trend betrifft die Akzeptanz von Empfehlungssystemen, die besonders deutlich in der Musikbranche wird. Dort haben sich Musikempfehlungssysteme wie Pandora oder LastFM bei den Nutzern als sehr beliebt erwiesen. Diese Akzeptanz wird auch durch die erhöhte Bereitschaft begründet, persönliche Daten im Internet zu veröffentlichen bzw. solchen Anbietern zu überlassen.

Diese aktuellen Trends machen den Zeitpunkt für eine personalisierte digitale Zeitung optimal. In dieser Arbeit soll deshalb die Entwicklung eines Empfehlungssystems für journalistische Inhalte beschrieben werden, das dem Nutzer hilft, leichter zwischen Relevantem und Irrelevantem zu unterscheiden.

1.2 Ziele und Anforderungen

Ziel dieser Arbeit ist es, in einem gesamtheitlichen Ansatz, ein personalisiertes Nachrichtensystem zu entwickeln. Dies umfasst einen intelligenten Crawler, der Feeds einer Webseite automatisch findet und auf verschiedene Sprachen anwendbar ist. Er soll dabei Feeds mit den Standards Atom und RSS bis Version 2.0 unterstützen. Neben Website-Feeds soll der Crawler auch in der Lage sein, die jeweiligen Bilder zu einem Artikel zu finden, die oftmals nicht in den Feed-Elementen angegeben sind.

Die Anpassung an die Nutzerpräferenzen im Empfehlungssystem bzw. PNA soll vollständig automatisch möglich sein, sodass keinerlei explizite Nutzerinteraktion notwendig ist. Hierfür sollen Klassifikationsalgorithmen aus dem maschinellen Lernen zum Einsatz kommen. Optional sollen Nutzer jedoch die Möglichkeit haben, manuelle Anpassungen vorzunehmen. Das Empfehlungssystem soll außerdem nicht auf einem statischen Modell für alle Nutzer

beruhen, sondern für jeden Nutzer *individuell* trainiert werden. Das Ranking dabei soll anhand verschiedener Kriterien wie Popularität, Aktualität, persönlicher Relevanz etc. vorgenommen werden. Es soll hierbei eine Lösung angeboten werden, die auch den zuvor erwähnten Nachrichtenwert in Betracht zieht. Duplikate oder Beinahe-Duplikate (gleiche Meldung von anderer Quelle) sollen vermieden werden.

Der daraus entstandene PNA soll anschließend genutzt werden, um den Grad der gewünschten Personalisierung festzustellen. Hierzu sollen vier Testgruppen erstellt werden, die unterschiedliche Varianten des PNA erhalten. Dabei werden Personalisierungsmöglichkeiten zu jeder weiteren Gruppe erhöht, wobei die erste eine vollautomatische Personalisierung erhält und dabei keinerlei Änderungen vornehmen darf, während die letzte dies darf und zusätzlich noch Designeinstellungen der Oberfläche vornehmen kann (Customization).

Diese Arbeit soll im Wesentlichen Antwort auf die folgenden Fragen geben:

1. Wie entwickelt man gesamtheitlich einen PNA?

- **Kann man mit impliziten Feedback Strategien schneller und besser die Interessen eines Nutzers lernen?**
- **Können die vorhandenen Informationen aus sozialen Netzwerken oder auf Seiten der Nachrichtenanbieter besser genutzt werden?**
- **Wie schnell adaptiert ein PNA die Interessen eines Nutzers?**
- **Wie treffsicher ist ein PNA?**
- **Welcher Algorithmus ist geeignetste?**

2. Existiert eine Zahlungsbereitschaft für so einen Dienst?

- **Wird Personalisierung im Allgemeinen bzw. bei Nachrichten vom Nutzer gewünscht?**
- **Wie hoch ist die Zahlungsbereitschaft für die Dienstleistung eines PNA?**
- **Was ist der optimale Grad der Personalisierung?**
- **Welche Aussagen lassen sich über die Zielgruppe machen?**

1.3 Aufbau

Die Arbeit hat zwei große Schwerpunkte, die einerseits im wirtschaftlichen und andererseits im technischen Bereich liegen. Da diese Bereiche teilweise stark zusammenhängen ist eine strikte Trennung nicht sinnvoll und die Übergänge daher teilweise fließend. Im zweiten Kapitel sollen zunächst die Grundlagen der Personalisierung und Recommender Systemen erläutert werden. Im Anschluss daran folgt der Stand der Technik im wissenschaftlichen Bereich und getrennt davon, Beispiele aus der Praxis, wo die wirtschaftlichen und technischen Aspekte tatsächlich realisiert wurden.

Sowohl das dritte, vierte als auch das fünfte Kapitel sind auf der technischen Seite anzuordnen. Im dritten Kapitel wird zunächst das konzeptuelle Design eines PNA beschrieben. Hier wird als erstes versucht, die Ansprüche eines Nutzers an eine personalisierte Zeitung zu bestimmen. Anschließend wird die Aggregation von Nachrichtenartikeln erklärt und wie man implizit und explizit an die Nutzerpräferenzen gelangt. Darauf aufbauend werden Klassifikationsalgorithmen für die Domäne Nachrichten entworfen. Außerdem wird beschrieben, wie die daraus entstandene Menge an relevanten Nachrichtenartikeln mit verschiedenen Rankingansätzen in eine bessere Reihenfolge gebracht werden kann.

Das vierte Kapitel widmet sich der technischen Umsetzung des zuvor beschriebenen PNA. Dort sollen wichtige und interessante Implementierungsdetails erläutert werden. Im fünften Kapitel werden die zuvor entworfenen und implementierten Algorithmen und Strategien evaluiert.

Das sechste und siebte Kapitel sind rein wirtschaftlich orientiert und dienen, die Zahlungsbereitschaft für einen PNA Dienst zu bestimmen. Dort werden zunächst im sechsten Kapitel die Daten der Feldstudie beschrieben und im siebten Kapitel dann evaluiert.

Im achten Kapitel wird ein Fazit gezogen und ein Ausblick in weitere Entwicklungen und Forschung in diesem Bereich gegeben.

2 Grundlagen

Jüngst hat eine Umfrage der Bloomberg Business Week herausgestellt, dass 80% der Marktteilnehmer Personalisierung als Top-Priorität bewerten. Dies umfasst vor allem automatisierte Recommender Systeme sowie die traditionelle manuelle Customizing. Einer der ersten großen Anwendungsbereiche für ersteres waren Recommender Systeme im Bereich E-Commerce.

In diesem Kapitel sollen zunächst die Begrifflichkeiten in diesem Bereich geklärt werden. Anschließend wird der Nutzen von Personalisierung dargelegt, um die Motivation dieser Arbeit aufzuzeigen. Danach wird ein technischer Überblick über die Recommender Systeme gegeben und dabei besonderer Fokus auf Textklassifizierung gelegt.

Im Anschluss daran wird ein Überblick über den wissenschaftlichen Stand sowohl bezüglich Zahlungsbereitschaft als auch bezüglich personalisierten Nachrichtenaggregatoren gegeben. Schließlich werden hierzu noch reale Beispiele aus der Wirtschaft analysiert.

2.1 Grundlegende Begriffe: Individualisierung, Personalisierung, Customizing

Häufig werden die Begriffe Individualisierung, Customization und Personalisierung in der Literatur synonym angewandt – als Anpassung einer Form von Marktleistung an eine Gruppe bis hin zum einzelnen Kunden.² Tatsächlich stellen diese Begriffe jedoch sehr heterogene Auffassungen dar, was im Folgenden genauer erläutert werden soll.

Individualisierung wird oftmals als Gegenpol zur Standardisierung bezeichnet: Die „bewusste, gewollte Gestaltung eines Produkts im Hinblick auf die Nutzung durch ein Individuum“.³ Das Ziel wird hierbei als die maximale Befriedigung der individuellen Präferenzstruktur des Nachfragers durch die Leistungen des Anbieters.⁴ Die Einzigartigkeit sowie deren Erstellung stehen dabei im Vordergrund.

Der Begriff *Personalisierung* wird vor allem im Kontext der IT verwendet und beschreibt den Individualisierungsprozess. Dort wird Personalisierung definiert als der Prozess von Veränderung der Funktionalität, Nutzerschnittstelle oder des Informationsinhalts eines Systems, um die persönliche Relevanz eines Individuums zu erhöhen.⁵ Oftmals wird sie als

² Vgl. Rauscher & Hess (2005), S.5

³ Schneider (1998), S.13

⁴ Vgl. Reichwald et al. (2002), S.10

⁵ Vgl. Blom (2000), S.4

Vorstufe der Individualisierung angesehen, da bei ihr im Gegensatz zur Individualisierung nur eine Kundengruppe und nicht der einzelne angesprochen wird.⁶

Bei der Personalisierung wird system-initiiert, meistens algorithmisch versucht, die Relevanz für den Nutzer zu erhöhen. Es werden dabei Techniken des Data Mining sowie Clickstreams eines Nutzers analysiert, um eine Aussage über sein zukünftiges Verhalten zu erstellen. Nutzer finden dadurch Informationen effizienter und erhalten einen Hebel zu einer größeren Datenvielfalt. Derartige Systeme sind heute möglich und auch notwendig. Sie soll die Informationsüberladung hemmen und dem Nutzer einen zusätzlichen Service anbieten.⁷

Auch der Begriff *Customizing* findet häufig synonyme Verwendung mit den Vorherigen Begriffen. Von der Personalisierung zu unterscheiden ist das *Customizing*. Man unterscheidet es von der Personalisierung durch die Art der Interaktion, während Personalisierung system-initiiert ist, ist *Customizing* nutzer-initiiert. Der Nutzer ist bei der Personalisierung passiv, das System adaptiert implizit die Präferenzen. Beim *Customizing* hingegen muss der Nutzer selbst aktiv werden und seine Präferenzen explizit oftmals mit statischen Profilen auswählen.⁸ Die Grenzen zwischen *Customizing* und Personalisierung sind nicht strikt gesetzt, oftmals existieren auch Mischformen.

Im Rahmen dieser Arbeit wird Individualisierung im Sinne der klassischen Personalisierung durch Recommender Systeme, *Customizing* insbesondere bei der Nutzeroberfläche des PNA, jedoch auch eine Mischung davon angewandt.

2.2 Nutzen von Personalisierung im Allgemeinen und bei Nachrichten

Online Personalisierung beruht auf dem Prinzip, vorhandene Nutzerinformationen bspw. aus dem Browsing-Verlauf zu verwenden, um damit neue Online Empfehlungen zu generieren oder generell ein Angebot auf den Anwender zuzuschneiden. Dies führt zu einem Abbau der Informationsüberladung und kann mit neuen Diensten einen Zusatznutzen generieren. Die Personalisierungsfunktionen sind dabei höchst unterschiedlich und reichen von einfacher Namensanzeige des Anwenders bis hin zu komplexen Katalognavigationen und

⁶ Vgl. Runte (2000), S.8

⁷ Vgl. Li & Unger (2012), S.2

⁸ Vgl. Rauscher & Hess (2005), S.8

Produktempfehlungen.⁹ Der Nutzen einer solchen Personalisierung für den Anwender bzw. Konsumenten ist jedoch nicht erwiesen und wurde deshalb zahlreich untersucht.¹⁰ Die Untersuchungen hatten dabei unterschiedliche Ausrichtungen (Produkte, Musik, Werbung etc.), von denen die Erkenntnisse u.U. nicht übertragbar sind auf den Bereich der Nachrichten.

Der generelle Nutzen von solchen personalisierten Systemen für Anwender lässt sich insbesondere vom ökonomischen Argument ableiten, dass eine engere Bindung zwischen Präferenzen und Produktattributen zu einem größeren Vorteil für den Kunden wird (Preference Fit Theory).¹¹ Ausführliche Informationen zu diesem Thema sowie den Strategien hierfür lieferten auch Shapiro und Varian.¹²

Bei der Nutzung eines Personalisierungsdienstes muss der Anwender zwangsläufig persönliche Daten preisgeben, wobei er nicht weiß, was der Dienstanbieter mit diesen Daten letztendlich tut. Einer der größten Einwände daher ist der Trade-Off zwischen Datenschutz sowie erhöhter Qualität und wurde u.a. von Li & Unger aufgezeigt.¹³ Ein zusätzlicher Nutzen ist nach ihren Erkenntnissen unter bestimmten Umständen möglich und bietet neue Möglichkeiten der Monetarisierung, da auch eine Zahlungsbereitschaft besteht.

Tam & Ho haben ein Modell, basierend auf der sozialen Wahrnehmung und der Psychologie des Konsumenten entwickelt, um den Nutzen zu testen.¹⁴ Dabei haben sie festgestellt, dass der Anwender die Personalisierung wahrnimmt und auch empfänglich für sie ist. Sie also als nützliche Entscheidungshilfe auffasst.¹⁵ Ihre Untersuchungen umfassen jedoch nur ein personalisiertes Werbungs- bzw. Produktangebot sowie ein personalisiertes Musikangebot.

Außerdem wurde der Nutzen in der Arbeit von Liang et al. untersucht.¹⁶ Dabei kategorisierten die Autoren die psychologischen Haupttheorien zur Informationssuche von Nutzern in Aufwandstheorien sowie motivationsbegründete Theorien und begründen damit den Nutzen von Personalisierung.¹⁷

⁹ Eine genaue Analyse findet sich bei Kramer et al. (2000)

¹⁰ Komiak & Bensabat (2006), Awad & Krishnan (2006) Ho et al. (2008), Ho & Tam (2011)

¹¹ Vgl. Simonson (2005), S.32

¹² Shapiro & Varian (1998)

¹³ Li & Unger (2012)

¹⁴ Tam & Ho (2006)

¹⁵ Vgl. Tam & Ho (2006), S.1

¹⁶ Liang et al. (2006)

¹⁷ Vgl. Lian et al. (2007), S.46

Insgesamt lässt sich sagen, dass sich der Nutzen von Personalisierung für einige Bereiche erwiesen hat. Personalisierte Nachrichtenangebote für die breite Masse¹⁸ stellen noch ein sehr neues Phänomen dar, für dessen Nutzen es noch keine eindeutigen Belege gibt. Dieses Angebot hat jedoch große Ähnlichkeit mit bestätigten Bereichen (wie z.B. Musik) und könnte ebenfalls auf dem ökonomischen Argument der Preference Fit Theory beruhen, weshalb die Vermutung eines Zusatznutzens vorliegt. Für die Ermittlung wird oftmals die Zahlungsbereitschaft als aussagekräftige Kennzahl verwendet.¹⁹ Sie lässt sich leicht quantifizieren und bedingt den Nutzen der Personalisierung. Aus diesem Grund soll dies im Zusammenhang mit der Zahlungsbereitschaft und dem optimalen Grad der Personalisierung in dieser Arbeit geklärt werden.

2.3 Recommender Systems

Empfehlungssysteme (Recommender Systems) stellen die konkrete Realisierung von Personalisierungsmechanismen dar, für die verschiedene Techniken angewandt werden. Die bekannteste dabei sind das *Collaborative Filtering* sowie das *Content-based Filtering*. Beim Collaborative Filtering werden nutzerbasierte Vergleiche anhand ihrer Bewertungen (z.B. Like oder Dislike) angestellt. Beim Content-based Filtering werden die Interessensobjekte definiert durch ihre Eigenschaften (*Features*), aus denen für den Nutzer ein individuelles Profil erstellt wird.²⁰ Das Content-based Filtering ist insbesondere geeignet für die Recommender Systems mit Textinhalten, da hier sehr leicht eindeutige Features extrahiert werden können.

Diese genannten Techniken werden im Folgenden noch genauer vorgestellt. Weitere, weitaus weniger häufige, Techniken sind *Demographi-*, *Utility-based* und *Knowledge-based Filtering*.²¹

Für die oben genannten Techniken ist eine Sammlung der Nutzerdaten erforderlich (*Eingangsdaten*). Dies kann *implizit* oder *explizit* geschehen.²² Bei der *expliziten* Variante wird der Nutzer befragt oder gibt manuell Rückmeldung, womit ein Nutzerprofil von ihm erstellt wird und Empfehlungen für ihn generiert werden können. Hierzu kann beispielsweise eine Bewertungsskala von eins bis zehn nach der Einblendung einer Empfehlung angezeigt

¹⁸ Hiermit sind nicht die frühen Prototypen v.a. aus und für den Informatik-Bereich gemeint.

¹⁹ Z.B. bei Li & Unger (2012)

²⁰ Vgl. Burke (2002), S.334

²¹ Eine sehr gute Zusammenfassung hierfür findet sich bei Burke (2002), S.332ff

²² Vgl. Zhang & Seo (2001), S.666

werden. Allerdings scheuen Nutzer den zusätzlichen Aufwand, finden es umständlich, haben Datenschutzbedenken oder benutzen es aus anderen Gründen nicht.

Das Gegenstück zur expliziten Aufnahme ist die *implizite*. Sie beruht auf einer Verhaltensbeobachtung des Nutzers. Hierzu werden beispielsweise die Clickstreams eines Nutzers erfasst. Studien haben gezeigt, dass diese Methode vergleichbare Ergebnisse erzielt wie die explizite.²³ So können ohne weiteres Zutun des Nutzers seine Präferenzen schnell und effektiv festgestellt und ein Nutzerprofil von ihm erstellt werden.

2.3.1 TfIdf Repräsentation

Die TfIdf (*Term frequency Inverse Document Frequency*) Repräsentation ist ein Maß zur Beurteilung der Relevanz von Worten bzw. Termen in Dokumenten. Es wird primär im Rahmen von Textklassifikation verwendet, um Dokumente in eine vorgegebene Menge von Klassen einzuteilen.²⁴

Der wesentliche Aspekt dabei ist, dass bei der TfIdf Worte über ihre relative Häufigkeit gewichtet werden. Dabei spricht man von Termfrequenz (*term frequency*) und Dokumentfrequenz (*document frequency*). Die Termfrequenz $tf(w, d)$ gibt an, wie oft das Wort w im Dokument d auftritt. Die Dokumentfrequenz $df(w)$ hingegen beschreibt die Anzahl der Dokumente, in denen das Wort w mindestens einmal vorkommt. Unter der Annahme, dass Worte, die nur in wenigen Dokumenten vorkommen, aber häufig im untersuchten Dokument auftreten, repräsentativ sind für dieses Dokument. Die Formel für den TfIdf lautet deshalb wie folgt:²⁵

$$tfidf(w, d) = tf(w, d) * idf(w)$$

$$idf(w) = \log \frac{|D|}{df(w)}$$

Hierbei ist D die zugrundeliegende Dokumentensammlung und $|D|$ deren Anzahl. Folgendes Beispiel soll das Prinzip des TfIdf genauer erläutern. Im Satz „Dort steht ein grüner Baum, mit grünen Blättern.“ wird der Term „grün“ zweimal verwendet ($Tf=2$), während das Wort

²³ Vgl. Zhang & Seo (2001), S.666

²⁴ Vgl. Joachims (1996), S.3

²⁵ Vgl. Joachims (1996), S.3

„Baum“ einmal verwendet wird ($T_f=1$). Kommt dieser Satz innerhalb einer Reihe botanischer Dokumente vor, die hauptsächlich von Bäumen handeln, ist der TfIdf von „Baum“ vermutlich geringer als der von „grün“. Kommt dieser Satz jedoch in einer Reihe von Kunstdokumenten, über verschiedene Farbtöne vor, hat „Baum“ voraussichtlich einen höheren TfIdf.

2.3.2 kNN

Der kNN (*k-Nearest Neighbor*) ist einer der simpelsten (Online-)Klassifizierungsalgorithmen, der seine Entscheidung aufgrund seiner nächsten bzw. ähnlichsten k -Nachbarn trifft. Es wird die Klasse gewählt, die am häufigsten unter den k -Nachbarn auftritt. Hierfür wird ein Merkmalsvektor erstellt, der anhand einer Metrik (Manhattan-Metrik, Euklidischer Abstand etc.) die Nähe zu den anderen Trainingsbeispielen berechnet. Die tatsächliche Klassifikation ist dann im einfachsten Fall eine Mehrheitsentscheidung (*Voting*).²⁶

Zu kleine k führen dazu, dass das Klassifikationsergebnis durch Rauschen verschlechtert wird. Ein zu großes k kann dazu führen, dass zu unähnliche Beispiele bzw. zu weit entfernte Punkte herangezogen werden, was häufig bei wenigen oder nicht gleichverteilten Daten auftritt. Für letzteren Fall kann diese Problematik dadurch gemindert werden, dass eine gewichtete Abstandsfunktion verwendet wird. Ein weiteres Problem ist der hohe Rechenaufwand der mit der Anzahl der zu vergleichenden Dokumente steigt.²⁷

2.3.3 Perceptron

Der Perceptron (im Deutschen oft auch Perzeptron genannt) ist ein vereinfachtes neuronales Netzwerk, das von Rosenblatt 1958 vorgestellt wurde. In der einfachsten Variante besteht das neuronale Netzwerk aus einem einzigen Neuron mit variablen Gewichtungen und einem Schwellenwert. Das Prinzip des Perceptrons ist es, bei falschen Vorhersagen die Gewichte mit einer Lernrate leicht anzupassen, um so das gewünschte Resultat zu erhalten. Das Modell lässt sich mithilfe von $\phi : \mathcal{X} \rightarrow \mathbb{R}^D$ generalisieren, wobei $\phi(x)$ als Featurevektor verwendet wird. Das zu trainierende Modell lautet:

$$f_w(x) = \text{sign}(\langle w, \phi(x_n) \rangle)$$

²⁶ Vgl. Zhang et al. (2007) S.2

²⁷ Vgl. YU, Cui, et al. (2001), S.1

$$\text{sign}(a) = \begin{cases} -1 & \text{falls } a \leq 0 \\ 1 & \text{sonst} \end{cases}$$

Nach Bishop ist ein w gesucht, sodass für alle Trainingsbeispiele (x_n, y_n) gilt, dass $y_n \langle w, \phi(x_n) \rangle > 0$.²⁸ M ist Menge aller von f_w falsch vorhergesagten Beispiele. Korrekt klassifizierte Beispiele verändern das Modell. Die Fehlerfunktion sieht wie folgt aus:

$$e_p(w) = - \sum_{n \in M} y_n \langle w, \phi(x_n) \rangle$$

Wobei e_p minimiert werden soll. Beim Lernvorgang iteriert der Perceptron über alle Trainingsdaten und addiert die fehlklassifizierte Beispiele entsprechend ihrer korrekten Klassifikation mit der Lernrate η hinzu.

2.3.4 Hybride Systeme

Hybride Ansätze verbinden verschiedene Recommender Techniken, um so die Vorteile der jeweiligen Verfahren zu erhalten und die Nachteile auszubessern. In den meisten Fällen wird hierbei Collaborative Filtering mit einem anderen Verfahren kombiniert, um so die das Cold Start Problem zu beheben.²⁹

Für die Zusammenführung der Ergebnisse werden existieren fünf verbreitete Methoden:³⁰

- *Weighted*: Die einzelnen Scores oder Votes verschiedener Recommender werden in einer einzigen Empfehlung vereinigt.
- *Switching*: Das System wechselt zwischen verschiedenen Recommendern, abhängig vom Kontext.
- *Mixed*: Empfehlungen mehrerer Recommender werden zur gleichen Zeit angezeigt.
- *Feature Combination*: Features aus mehreren Verfahren werden vereinigt.
- *Cascade*: Ein Recommender verfeinert das Ergebnis eines anderen.
- *Feature Augmentation*: Das Ergebnis eines vorherigen Recommenders fließt als Feature mit in den nachfolgenden Recommendation Prozess ein.

²⁸ Vgl. Bishop (2007), S.2

²⁹ Vgl. Burke (2002), S.339

³⁰ Vgl. ders., S.340

-
- *Meta-Level*: Das gesamte Modell eines vorherigen Recommenders wird als Input für einen nachfolgenden Recommender genutzt.

2.4 Nachrichtenfeeds

Zuvor wurden Wege beschrieben, wie man aus den Eingangsdaten eines Nutzers Empfehlungen generieren kann. Diese Empfehlungen sind im vorliegenden Fall Nachrichtenartikel, die in einer einheitlichen Form gespeichert bzw. aggregiert werden müssen. Hierzu wird von vielen Nachrichtenaggregatoren die weitverbreiteten Formate Rich Site Summary³¹ (RSS) oder Atom genutzt, die beide auf XML basieren. Diese *RSS-Feeds* (bzw. Atom-Feeds) beinhalten Informationen zu mehreren Artikeln und werden oftmals von Nachrichtenseiten in Kategorien unterteilt. Zu einem Artikel werden dabei der Titel sowie teilweise Bilder, Kurzzusammenfassung (Abstract), Stichwörter und Kategorien mitgeliefert.

Nachrichtenseiten veröffentlichen ihre neuen Artikel auf ihrer Webseite und zeitgleich in den RSS-Feeds. Aggregatoren nutzen dies aus, in dem sie in regelmäßigen Zeitabständen die Feeds auf Aktualisierung untersuchen. Dazu müssen sie jedes Mal von einem Parser verarbeitet werden, was aufgrund des, im Verhältnis zu HTML-Webseite, geringen XML-Overheads sehr effizient gestaltet werden kann.

Die Links zu den wichtigsten RSS-Feeds werden von den Nachrichtenseiten in der Regel in den Meta-Informationen der Startseite eingetragen. Auf den Artikelseiten befinden sich wiederum Verweise in den Meta-Informationen auf andere, meistens Kategorie-Feeds. Diese Meta-Informationen können aufgrund einheitlicher Tags automatisiert ausgelesen werden, was auch als *Autodiscovery* bezeichnet wird.

2.5 Evaluierung

Im Folgenden soll ein Überblick über Verfahren und Metriken der Evaluierung gegeben werden, die für Information-Retrieval-Systeme (IR-Systeme) relevant sind und daher auch für diese Arbeit. Die zwei wesentlichen Fragen bei der Qualitätsbeurteilung lauten:

³¹ Oftmals wird RSS auch mit *Really Simple Syndication* ausgeschrieben.

-
- *Precision (Genauigkeit)*: „Wie viele gefundenen Dokumente sind tatsächlich relevant?“ – Dieser Wert stellt die Genauigkeit des Ergebnisses dar.
 - *Recall (Trefferquote)*: „Wie groß ist beim Ergebnis der Anteil der relevanten Dokumente im Verhältnis zur Gesamtzahl der relevanten Dokumente?“ – Dieser Wert kann als Vollständigkeit des IR-Systems aufgefasst werden.

In der vorliegenden Domäne der Nachrichtenaggregatoren ließe sich der Erfolg eines neuen Algorithmus automatisiert im Langzeittest und manuell über eine festgelegte Auswahl von Artikeln ermitteln. Manuell bedeutet, dass ein Korpus von Dokumenten bzw. Artikeln von menschlichen Experten bewertet (*labeled*) wird. Die Evaluierung wird anschließend ausschließlich auf diesem Datensatz durchgeführt.

Automatisiert messen ließe sich der Erfolg in diesem Fall nur über die Anzahl der Klicks eines Nutzers, wobei mehr Klicks auf eine höhere Güte hinweisen. Dies hätte jedoch den Makel, dass interessante Artikel, bei denen schon die Überschrift ausreichte, nicht erkannt werden können. Ein schwaches Signal liefern auch „übersprungene“ Artikel, die der Nutzer nicht angeklickt hat, wobei hier nicht eindeutig ist, ob sie zwar interessant, aber nicht lesenswert waren oder weniger interessant. Auf diese beiden Problematiken wird im Verlauf noch ausführlich eingegangen. Bei einer automatisierten Auswertung können sich außerdem große Schwankungen dadurch bilden, dass die Artikelauswahl mit der Zeit variiert, womit auch die Relevanz variieren kann. Die Evaluierung würde so an unterschiedlichen Tagen zu unterschiedlichen Ergebnissen kommen.

Im Folgenden werden die typischen Metriken zur Evaluierung genauer erläutert. Precision, Recall und F-Measure bilden dabei die Grundlage. Anschließend werden Metriken erläutert, die die Genauigkeit einer Ergebnisliste (*Mean Squared Error, Root Mean Squared Error*) sowie die Güte ihrer Sortierung ermitteln (*Mean Average Precision* und *Normalized Discounted Cumulative Gain*). Auf die ebenfalls weitverbreitete Metrik *Mean Reciprocal Rank* wird in dieser Arbeit verzichtet. Sie bezieht zusätzlich noch die Reihenfolge mit ein, was nicht für den Nachrichtenbereich geeignet scheint, weshalb sie auch nicht genauer erläutert wird.

2.5.1 Precision, Recall, F-Measure

Die Gesamtmenge betrachteter Dokumente lässt sich in die zwei Teilmengen der *relevanten* und *gefundenen* Dokumente unterteilen. Diese werden als Kombination aus „true“ bzw. „false“ und „positive“ und „negative“ in einer Konfusionsmatrix dargestellt (Tabelle 1: Konfusionsmatrix). Anschaulich erklären lassen sich „false negative“ als irrtümlich ausgefilterte Dokumente, wohingegen „false positive“ fälschlicherweise als relevant gezeichnete Dokumente beschreibt.

	Relevant	Nicht relevant
Gefunden	tp (true positive)	fp (false positive)
Nicht gefunden	fn (false negative)	tn (true negative)

Tabelle 1: Konfusionsmatrix

$$Precision = \frac{tp}{tp + fp} = \frac{|{\{Relevante\} \cap \{Gefundene\}}|}{|{\{Gefundene\}}|}$$

$$Recall = \frac{tp}{tp + fn} = \frac{|{\{Relevante\} \cap \{Gefundene\}}|}{|{\{Relevante\}}|}$$

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

Precision und Recall verhalten sich oftmals gegenläufig. Eine hohe Precision geht daher oft mit einem niedrigen Recall einher. Die F1-Metrik bringt diese Werte ins Verhältnis. Abhängig von der Domäne können die Prioritäten individuell auf Precision oder Recall gelegt werden. Bei Suchmaschinen beispielsweise ist ein Anwender nicht daran interessiert, möglichst *alle* relevanten Dokumente zu seinem Suchbegriff angezeigt zu bekommen, sondern unter den ersten eine hohe Relevanz zu erhalten, weshalb die Precision hier besonders wichtig ist. In der

professionellen Recherche hingegen nehmen Anwender eine niedrigere Precision zugunsten eines höheren Recalls in Kauf.³²

In der Praxis lässt sich die Precision problemlos berechnen, wohingegen die Ermittlung des Recalls oftmals, aufgrund der unbekannt Menge an relevanten Dokumenten, unmöglich ist. Häufig wird deshalb die Evaluierung anhand von Ausschnitten aus der Gesamtmenge vorgenommen oder die Ergebnisse mehrerer Suchalgorithmen vereinigt und dies als relevante Gesamtmenge angesehen.³³

2.5.2 Mean Squared Error, Root Mean Squared Error

Der *Mean Squared Error*³⁴(MSE) gibt die Abweichung eines Schätzers vom zu schätzenden Wert an. Für die Ermittlung wird der Durchschnitt der einzelnen quadratischen Differenzen zwischen dem tatsächlichen Wert y und dem geschätzten Wert \bar{y} berechnet. Große Abweichungen von den tatsächlichen Werten werden durch die Quadrierung stärker gewichtet als geringere.

Der *Root Mean Squared Error* (RMSE) zieht zum Zwecke der Glättung aus dem MSE zusätzlich die Wurzel.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_i)^2}$$

Der RMSE muss stets im absolut im Kontext der zugrundeliegenden Werte betrachtet werden, da allein aufgrund des RMSE keine generelle Beurteilung möglich ist.³⁵

2.5.3 Mean Average Precision

Die bisherigen Metriken Precision und Recall, sowie das daraus abgeleitete F-Maß beziehen sich ausschließlich auf Mengen. Dies lässt jedoch die Sortierung einer Ergebnisliste außer Betracht, wodurch keine Evaluation von Rankingverfahren möglich ist, bei denen man eine

³² Vgl. Manning et al. (2008), S.2

³³ Vgl. Stumme, Krause (2009)

³⁴ Im deutschen Raum auch als Mittlere quadratische Abweichung bezeichnet.

³⁵ Vgl. Riedel (2008)

Sortierung nach Relevanz anstrebt. Das im Folgenden dargestellte Evaluierungsmaß, *Mean Average Precision* (MAP), berücksichtigt genau diese Sortierung bei der Bewertung.

Die *Genauigkeit* (*Precision*) für einen Rang k (siehe folgende Formel) wird durch die Anzahl der relevanten bzw. irrelevanten Dokumente in der Ergebnisliste bestimmt. Hierbei nimmt die Funktion $rel(i)$ den Wert 1 an, falls es sich um ein relevantes Dokument handelt und ansonsten 0.³⁶ Eine Ergebnisliste mit zehn Dokumenten, $P(k=10)$, von denen 3 relevant sind, hat folglich eine Präzision von 0,3.

$$P(k) = \frac{\sum_{i=1}^k rel(i)}{k}$$

Die *Average Precision* (AP) beschreibt die Summe der einzelnen zuvor berechneten Genauigkeiten für die relevanten Treffer, im Verhältnis zur Gesamtzahl der relevanten Dokumente (R).

$$AP = \frac{\sum_{k=1}^n P(k)rel(k)}{R}$$

Die *Mean Average Precision* (MAP) nutzt die *Average Precision* und bezieht außerdem noch mehrere, verschiedene Anfragen (Queries) in Bewertung mit ein.

$$MAP = \frac{\sum_q AP_q}{|Q|}$$

2.5.4 Normalized Discounted Cumulative Gain

Der *nDCG* zieht ebenfalls, wie die MAP, die Sortierung einer Liste in Betracht. Er ist dabei jedoch feingranularer als MAP, da hier nur binäre Relevanzbewertungen (0|1) übernommen werden können. Außerdem ist er insbesondere für abgestufte Bewertungen auf Grundlage der Top- k Elemente ausgelegt.

$$nDCG(Q, k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} Z_k \sum_{m=1}^k \frac{2^{rel(j,m)} - 1}{\log(1 + m)}$$

$rel(j,m)$ stellt die Relevanzbewertung des m -ten Elements der j -ten Anfrage dar und muss dabei keine absolute Relevanz angeben, sondern kann auch mit Relevanzklassen genutzt werden. Z_k hingegen ist ein konstanter Wert, der das bestmögliche Ergebnis beschreibt.

³⁶ Vgl. University of Stanford, Evaluation of ranked retrieval results, 2009 (<http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html>), Abruf: 19.9.2013)

2.5.5 Prequential (Interleaved Test-Then-Train)

Im *Data Stream Mining* ist Prequential Testing das am häufigsten genutzte Messverfahren.³⁷ Hierbei wird für jedes Beispiel eine Vorhersage geliefert, die dann mit dem tatsächlichen Wert verglichen wird und dieser tatsächliche Wert der Trainingsmenge hinzugefügt. Der Vorteil ist, dass man so einen Überblick der Leistungsfähigkeit eines Lernverfahrens über die Zeit erhält.³⁸

Eine Abwandlung hiervon für eine Nachrichtenauswahl wurde von Bomhardt beschrieben.³⁹ Dabei wurden Chunks aus Nachrichtenartikeln mit der Größe 50 gebildet, auf die zunächst ein trainiertes Model evaluiert wurde. Anschließend wurde dieser Chunk der Trainingsmenge hinzugefügt. Chunk 1 wurde zum Training benutzt und auf Chunk 2 getestet. Anschließend wurden Chunk 1 und 2 zum Training genutzt und auf Chunk 3 getestet etc.

2.6 State-of-the-Art

Das Internet ist eine ideale Plattform, um den Kunden Produkte zu bieten, die zugeschnitten sind auf ihre Bedürfnisse und Präferenzen.⁴⁰ Dies gilt vor allem für den Nachrichtensektor, da die meisten Nutzer nur in einen bestimmten Teil der angezeigten Nachrichten und Berichte interessiert sind.⁴¹ Manche sind eher im politischen Geschehen interessiert, während andere sich mehr für Finanznachrichten interessieren. Ein individuell auf den Nutzer zugeschnittenes Angebot spart demnach Zeit, Aufwand und kann im besten Fall zu einer freudigen Überraschung führen (Serendipität). Aus diesem Grund haben große Anbieter wie Google News den Nutzen der Personalisierung erkannt und für ihre Nutzer implementiert. Der Grad der Personalisierung variiert jedoch zwischen den Anbietern stark, was im nächsten Unterkapitel noch genauer geklärt werden soll.

Der wesentliche Punkt dieser Arbeit ist jedoch zu klären, ob und wie groß die Zahlungsbereitschaft für PNA, unter Berücksichtigung des Personalisierungsgrades. Da in der Literatur PNAs in der Vergangenheit meistens nur aus technischer Sicht betrachtet wurden, sind die Informationen bezüglich PNAs als Geschäftsmodell und dabei die Zahlungsbereitschaft sehr spärlich.

³⁷ Vgl. Gama et al. (2009), S.2

³⁸ Vgl. Albert und Eibe (2010), S.4

³⁹ Bomhardt (2004)

⁴⁰ Vgl. Liang et al. (2007), S. 46

⁴¹ Vgl. ders., S.46

Im Folgenden soll der Fokus neben der verfügbaren Zahlungsbereitschaft auch auf die technische Entwicklung gelegt werden. Aufgrund des derzeitigen Vorhandenseins von Unternehmen, die ähnliche Ideen verfolgen, sollen diese wegen ihrer Relevanz im Folgenden auch analysiert werden. Der folgende Abschnitt soll dabei jedoch als Wettbewerbsanalyse zu verstehen sein, sondern viel mehr die Erkenntnisse dieser Unternehmen adäquat reflektieren. Dies soll zudem eine wirtschaftliche Sicht auf PNA bringen und wie man mit solchen bestehen kann.

2.6.1 Personalisierung bei Nachrichten

Blom hat in einer Studie web-basierte Personalisierung untersucht und dabei fünf Typen von Nutzern festgestellt, an denen er konkrete Richtlinien für den Entwurf einer solchen allgemeinen Personalisierung abgeleitet hat.⁴²

Seine Empfehlungen für den Entwurf eines solchen Systems wurden konkret am Beispiel einer Plattform für Filmempfehlungen durchgeführt, lassen sich jedoch auf andere Bereiche generalisieren:⁴³

- Personifiziere die Oberfläche mit den Kommentaren und Bewertungen anderer.
- Seriöses Auftreten nach außen (nicht zu spielerisch).
- Informationen zu einem empfohlenen Element hinzufügen, die Empfehlung unterstützen.
- Nutzern nicht vorschreiben was sie mögen, sondern diverse Alternativen bieten.
- Nutzern erklären, warum der Algorithmus ein bestimmtes Element für ihn empfohlen hat.

Die genannten Richtlinien eignen sich als Grundlage für die Entwicklung eines PNA und dienen fortan der Orientierung.

2.6.2 Zahlungsbereitschaft

In dieser Arbeit soll die Zahlungsbereitschaft für einen PNA und damit einhergehend auch der optimale Grad der Personalisierung mit der höchsten Zahlungsbereitschaft geklärt werden.

⁴² Vgl. Blom (2002), S.540

⁴³ Vgl. ders., S.541

Die Zahlungsbereitschaft für den Dienst eines PNA setzt sich aus zwei Komponenten zusammen. Einerseits aus der Zahlungsbereitschaft für digitale Nachrichten (Online News Content). Andererseits aus der Zahlungsbereitschaft für die tatsächliche Personalisierung bzw. individuelle Filterung der Nachrichten. Im Folgenden sollen deshalb diese Bereiche zunächst getrennt und anschließend in Bezug auf PNA erläutert werden.

Derzeit sind Online Inhalte der Nachrichtenportale größtenteils kostenlos, was zu einem stetigen Schrumpfen der Einnahmen führt, das durch Online-Verkäufe nicht kompensiert werden kann (Abbildung 8-1: Newspaper Association of America, Einnahmen von US Zeitungen mit Print- und Onlinewerbung seit 2003). Aus diesem Grund beschäftigen sich viele Quellen aus der freien Wirtschaft mit dem Thema der Zahlungsbereitschaft für digitale Nachrichten und sollen hier auch herangezogen werden.

Aus diesen Umfragen geht hervor, dass fast 70% der deutschen Internetnutzer Online-Nachrichten zum Weltgeschehen lesen.⁴⁴ Aus 50,34 Mio. Internetnutzern⁴⁵ interessieren sich also 35,24 Mio. Nutzer für Online-Nachrichten. Eine umfassende Studie der Boston Consulting Group (BCG) im Jahre 2009 hat sich explizit mit dem Thema Zahlungsbereitschaft beschäftigt. An dieser Studie nahmen 5083 Internetnutzer aus neun Ländern (1006 aus Deutschland) teil. Man kam dort zu dem Ergebnis, dass 63% der deutschen Konsumenten von Online-Nachrichten bereit wären, für diese Inhalte zu zahlen.⁴⁶ Dies entspricht potentiell 12,21 Mio. Interessierten in Deutschland. Die Preisspanne für die meisten zahlenden Konsumenten liegt dabei zwischen 1-3 Euro (29%), gefolgt von 4-6 Euro (15%) und 7-10 Euro (11%) pro Monat.⁴⁷ Nur 8% der Internetnutzer wären bereit, 11 Euro oder mehr monatlich für Nachrichten auszugeben.⁴⁸

In dieser Studie der BCG wurde außerdem abgefragt, für welche Ausprägungen von Online-Nachrichten sich die Teilnehmer interessieren. Dabei stellte man fest, dass das höchste Interesse für lokal- und regionalbezogene Nachrichten (67%) sowie Nachrichtenarchive (63%) besteht.⁴⁹ Das geringste Interesse hingegen existiert bei Wirtschafts- und

⁴⁴ Siehe Abbildung 8-2: AGOF 2012, Nutzung von Themen und Angeboten des Internets

⁴⁵ Quelle: TNS Infratest, April 2013

⁴⁶ Siehe Abbildung 8-5: Boston Consulting Group (BCG), Willingness to pay for news online, German Customers

⁴⁷ Siehe Abbildung 8-5: Boston Consulting Group (BCG), Willingness to pay for news online, German Customers

⁴⁸ Siehe Abbildung 8-5: Boston Consulting Group (BCG), Willingness to pay for news online, German Customers

⁴⁹ Siehe Abbildung 8-4: Boston Consulting Group, Willingness to pay for news online: Key findings from an international survey

Finanznachrichten (37%).⁵⁰ Für eine personalisierte Online-Zeitung aus verschiedenen Quellen würden sich nach dieser Umfrage 55% der Online-Nachrichtenleser interessieren.⁵¹ Der klassische Konsument von Online-Nachrichten lässt sich einordnen als tendenziell männlich und jünger als 29 Jahre.⁵² ⁵³ Diese Gruppe ist am meisten bereit für Online-Nachrichten zu zahlen (72% des Preises für Printmedien), wohingegen die Gruppe der über 50-jährigen eine deutlich geringere Zahlungsbereitschaft aufweist (45%).⁵⁴

Die Zahlungsbereitschaft für *allgemeine Personalisierung* wurde von Li und Unger untersucht.⁵⁵ Sie fanden dabei heraus, dass Menschen, die Online Personalisierung nutzen, wahrscheinlicher bereit sind, auch dafür zu zahlen. Das Augenmerk wurde hierbei insbesondere auf die Faktoren Qualität und Datenschutz gelegt, wozu jedoch keine starke Verbindung gefunden werden konnte. Die tatsächlich vorhandene Zahlungsbereitschaft machten sie von der Domäne abhängig, wofür sie auch Beispiele nannten. Das erste Beispiel bezog sich auf Nachrichtenpersonalisierung, wo die Autoren keine besondere Zahlungsbereitschaft vermuteten, da trotz des generierten Zusatzwertes, in diesem Bereich *großer Wettbewerb* besteht. Das zweite Beispiel hingegen zielte auf Finanzdienstleistungen, bei denen Kunden in einer Art Finanzberatung tiefere Informationen in real-time und außerdem erweiterte Analyse Tools erhalten. Hierfür machten die Autoren eine Zahlungsbereitschaft aus, da die *Informationen schwerer zugänglich* sind und *kein signifikanter Wettbewerb* besteht.⁵⁶

Oechslein und Hess haben in ihrer Arbeit untersucht, welche Geschäftsmodelle sich für die Verlagsindustrie im Hinblick auf die sinkenden Einnahmen beim Printgeschäft ergeben.⁵⁷ Da die Zahlungsbereitschaft für reinen Online-Inhalt gering ist legten sie dabei besonders den Fokus auf PNA, mit denen zusätzlicher Eigenwert durch die Analyse und Adjustierung von Informationen geschaffen werden kann. Die durchgeführte Studie ist aufgrund ihrer Zeitnähe

⁵⁰ Dieselbe.

⁵¹ Dieselbe.

⁵² Siehe Abbildung 8-7:Allensbachstudie, ACTA 2012

⁵³ Siehe Abbildung 8-3: AGOF 2012, Top-Themen nach Geschlecht

⁵⁴ Siehe Abbildung 8-8: PWC, WAN. Moving into Multiple Business Models: Outlook for Newspapers Publishing in the Digital Age, 2009

⁵⁵ Li & Unger (2012)

⁵⁶ Vgl. Li & Unger (2012), S.638

⁵⁷ Ochslein & Hess (2013)

besonders relevant und aussagekräftig, da sie dadurch wesentliche aktuelle Trends bzw. vergangene Trends miteinbezieht.

Es wurde eine repräsentative Umfrage mit 498 Teilnehmern gemacht, die einen Fragebogen zu einem PNA ausfüllen mussten. Von diesen konnten aber nur 116 verwertet werden, da die anderen noch keine Erfahrung mit PNA hatten. Mit der Studie sollten zwei Forschungsfragen geklärt werden:

1. Welche Eigenschaften sind für Nutzer relevant und wie beeinflussen sie die Nutzung?

Dabei haben die Forscher als signifikante Eigenschaften herausgestellt: *Die wahrgenommene Nützlichkeit, der Nutzungskomfort und die soziale Personalisierung*. Sie empfehlen daher einen PNA so zu fokussieren, dass der Nachrichtenkonsum der Nutzer vereinfacht wird. Design Aspekte bezüglich der Nutzungsoberfläche und der Inheldarstellung sollten dabei eine maßgebliche Rolle spielen. Ihrer Meinung ist Customizing ein Hauptelement des PNA und könnte zu einer höheren Zahlungsbereitschaft führen.

2. Was sind die Nutzer bereit für einen PNA zu zahlen?

Hierbei stellten die Forscher fest, dass sich mit dem zugrundeliegenden Modell ein optimaler Preis von 1,88 Euro pro Monat ergibt sowie ein akzeptables Preisspektrum von 0,42 Euro bis 6,83 Euro pro Monat. Der Indifferenzpreis betrug 2,83 Euro pro Monat.

2.6.3 Implizites Feedback

Aktuelle Recommender Systeme basieren vor allem auf explizitem Feedback (NewsWeeder, WebMate, NewsRec). Dies hat den Nachteil, dass der Nutzer so immer gezwungen ist, dem System aktiv Rückmeldung zu geben, was oftmals erschöpfend für ihn ist. Eine ergänzende Alternative hierzu ist die Nutzung von implizitem Feedback.

Typischerweise werden hierzu die angezeigten Webseiten aufgezeichnet, was jedoch häufig zu falsch zugeordneten Dokumenten führt, weil der Nutzer unwissentlich oder versehentlich uninteressante Webseiten aufruft.⁵⁸

Einige implizite Systeme schließen deshalb Anzeigedauer, Scrollverhalten, Mausaktivität, Tastaturaktivität, Lesezeichen sowie Speicher- und Druckbefehle als ein. Hierbei hat sich für viele Bereiche die Anzeigedauer als aussagekräftigstes Feature bewährt.⁵⁹

⁵⁸ Vgl. Burke & Gaul (2009), S.93

Bomhardt und Gaul haben sich mit einem hybriden System aus explizitem und implizitem Feedback für personalisierte Nachrichtensysteme beschäftigt.⁶⁰ Sie schätzten dabei die Anzeigedauer bei einem Artikel als unzureichend für das implizite Feedback ein. Sie begründeten dies damit, dass Nachrichtenartikel oftmals kurz sind und damit eine hohe Varianz einhergeht. Außerdem gibt ihrer Ansicht nach die Anzeigedauer keine Aussage über die Relevanz, da extrem kurze Meldungen („Sicherheitsupdate verfügbar“) auch interessant sein können und weil Ablenkungen schließlich auch das Ergebnis verfälschen könnten.⁶¹

Aus diesem Grund schlagen sie für ihr hybrides System die tatsächlich aufgerufenen Webseiten vor. Hierbei werden alle aufgerufenen Seiten positiv gewertet, außer der Nutzer hat explizit ein negatives Rating abgegeben. Wird ein Artikel nicht aufgerufen, wird dies als negatives Signal gewertet. Die eigentlichen Empfehlungen wurden dabei von einer SVM erstellt.

Die Evaluierung wurde anhand eines Datensatzes aus 1185 Artikeln von Heise Online⁶² gebildet, der zuvor komplett von einer Testperson bewertet wurde. Mit ihrem hybriden System erreichten sie im Durchschnitt über verschiedene Parameterschemata einen Recall von 38% und eine Precision von 67%. Damit liegen sie etwa im Bereich des reinen expliziten Feedbacks (46%, 65%), jedoch weitaus besser als mit einem reinen impliziten Modell (77%, 53%). Das hier angewendete implizite Feedback soll als Grundlage für diese Arbeiten dienen.

Aus einem Modell der Volkswirtschaftslehre entlehnten Joachims et al. die Idee, dass in Suchergebnissen der Klick auf ein Element, nicht nur Relevanz für dieses hat, sondern auch für die umgebenden Elemente. Sie nutzen als implizites Feedback demnach neben den geklickten Elementen auch die nicht geklickten und erzielen so zwar keine *absolute* Relevanz, jedoch eine *relative*. So ist beispielsweise in einer Reihe von Links, bei denen der erste und der dritte geklickt werden, der dritte von höherer Relevanz als der zweite, da dieser bewusst *nicht* geklickt wurde. Die Autoren geben hierfür elf Strategien an, die sie über Nutzerbeobachtungen erstellt haben, mit denen sie eine Genauigkeit von bis zu 82% erreichen konnten:⁶³

⁵⁹ Vgl. Kelly & Teevan (2003), S.19

⁶⁰ Bomhardt & Gaul (2009)

⁶¹ Vgl. Bomhardt & Gaul (2009), S.2

⁶² www.heise.de/ct (Abruf 13.9.2013)

⁶³ Vgl. Joachims et al. (2007), S.15ff.

-
1. **Click > Skip Above:** Alle übersprungenen Elemente vor einem geklickten Element haben eine geringere Relevanz als das geklickte.
 2. **Last Click > Skip Above:** Das letzte geklickte Element hat eine höhere Relevanz als die davor übersprungenen.
 3. **Click > Earlier Click:** Die Reihenfolge der Relevanz wird gegeben, durch die Klickreihenfolge.⁶⁴
 4. **Click > Skip Previous:** Die Relevanz des geklickten Elements ist größer als die des direkt übersprungenen Vorgängers.
 5. **Click > No-Click Next:** Die Relevanz des geklickten Elements ist größer als die des Nachfolgers, wenn danach kein weiteres Element geklickt wird.
 6. **Click > Skip Earlier Query Chain:** Das geklickte Element hat eine höhere Relevanz als die übersprungenen Elemente in vorherigen Suchanfragen (Analog zu 1., jedoch für multiple Suchanfragen).
 7. **Last Click > Skip Earlier Query Chain:** Analog zu 2., jedoch für multiple Suchanfragen.
 8. **Click > Click Earlier Query Chain:** Analog zu 3., jedoch für multiple Suchanfragen.
 9. **Click > Top One No-Click Earlier Query Chain:** Falls in der vorherigen Suchanfrage kein Element geklickt wurde, dann hat das geklickte Element höhere Relevanz als das erste in der vorherigen Anfrage.
 10. **Click > Top Two No-Click Earlier Query Chain:** Analog zu 9., allerdings für das zweite Element.
 11. **Top One > Top One Earlier Query Chain:** Das erste Element der neuen Anfrage ist immer höherwertiger, als das der vorhergehenden.

Die Strategien 1 – 5 generieren Signale innerhalb der Ergebnisse einer Suchanfrage, was ihrer Ansicht nach suboptimal ist, da nur die Top Ergebnisse verglichen werden und außerdem da sie nicht berücksichtigen, dass Nutzer oftmals mehrere Anfragen nacheinander stellen.

Strategien 6 – 11 beziehen sich auf multiple Suchanfragen, also solche, bei denen der Nutzer nicht sofort ein Ergebnis gefunden hat und deshalb seine Anfrage umformulierte.

⁶⁴ Anmerkung: Die Autoren schreiben dies zwar als eine ihrer Strategien aus, konnten jedoch in der Evaluation mit ihren Daten diese These nicht stützen.

Hierbei ist jedoch nochmals zu betonen, dass diese Strategien für Suchanfragen entwickelt wurden und *nicht* für die Nachrichtendomäne. In dieser Arbeit sollen deshalb ähnliche Strategien entwickelt werden, die sich an die von Joachims et al. anlehnen.

2.6.4 Personal Newsaggregators und Abgrenzung dazu

Die Idee, Informationsüberladung und –Filterung durchzuführen, existiert bereits seit den 90er Jahren. Dementsprechend gibt es in diesem Bereich viele Arbeiten, die jedoch teilweise sehr unterschiedliche Ansätze verfolgen. Außerdem sind der Aufbau der Evaluation und deren immer extrem unterschiedlich, was eine Vergleichbarkeit erschwert. Aus diesem Grund soll sich dieser Abschnitt nur auf wenige, prägnante von ihnen beziehen.

NewsWeeder ist eine der ersten webbasierten Anwendung für Usenet Newsgroups.⁶⁵ Der Nutzer bewertet Artikel explizit zwischen 1 und 5, worauf ein Lernmechanismus ein Nutzerprofil bildet. Hierfür werden sowohl inhaltsbasierte als auch kollaborative Ansätze genutzt, womit Nachrichtenzusammenstellungen generiert werden. Hierbei werden vor allem Tfidf-Ähnlichkeiten verwendet und eine Precision von 44% (Datensatz a) bzw. 59% (Datensatz b) für die oberen 10% der Artikel erreicht.

Ein weiterer Client, WebMate, basiert auf Ähnlichkeitsvergleichen zu explizit bewerteten Dokumenten und erzielte dort eine Precision von 31%.⁶⁶ Ist ein Dokument sehr ähnlich zu einem vorher positiv bewerteten, so wird es empfohlen. Ebenfalls ein bekanntes Projekt ist der Tour Guide WebWatcher.⁶⁷ Er erstellt Interessenprofile eines Nutzers anhand von Stichworten und empfiehlt Webseiten von Nutzern mit ähnlichen Interessen. Es basiert dabei sowohl auf implizitem (Link besucht) als auch explizitem Feedback. NewsRec spezialisiert sich auf Nachrichtenartikel von Heise Online und nutzt dabei explizites Feedback zusammen mit einer SVM.⁶⁸ Dabei erzielte NewsRec erzielte bereits nach 30 Trainingsdokumenten eine Precision von 54% und maximal 61%.

In der neueren Zeit sind die Projekte wie Necta RSS, RelevantNews, NewsSeecr und PersoNews hervorzuheben. NectaRSS nutzt verschiedene Algorithmen zur Filterung und Priorisierung von RSS-Feed Einträgen, basierend auf Nutzerprofilen.⁶⁹ RelevantNews haben

⁶⁵ Lang (1995)

⁶⁶ Checn & Sycara (1998)

⁶⁷ Joachims et al. (1996)

⁶⁸ Bomhardt (2004)

⁶⁹ Samper et al. (2008)

verschiedene Wege vorgeschlagen, Feeds nach semantischer Ähnlichkeit zu clustern.⁷⁰ NewsSeecr bildet ebenfalls Cluster aus Artikeln aus Nachrichtenfeeds, jedoch mithilfe von Named Entities und führt das Ranking basierend auf syntaktischen Features, popularitätsbasierten Features, zeitlichen Features etc. durch.⁷¹ PersoNews ist ein Projekt, bei dem ebenfalls eine größere Menge von Feeds aggregiert wird und mithilfe von Semantik und Taxonomien für den Nutzer personalisiert.⁷²

Die aufgezeigten Projekte haben Schwächen in vielerlei Hinsicht. Dies ist einerseits bedingt durch die technologische und gesellschaftliche Entwicklung der vergangenen Jahre, jedoch auch durch den kurzen Zeithorizont dieser Projekte. Im Folgenden werden deshalb die Schwächen aufgezeigt und anschließend die Abgrenzung vorgenommen.

(1) NewsWeeder und andere beschäftigten sich ihrer Zeit nur mit Usenet Newsgroups und nicht mit Nachrichtenartikeln. (2) Viele, z.B. auch WebWatcher, benötigen das Feedback von mehreren Nutzern. (3) Lediglich bei NewsSeecr bzw. Rankify werden Daten aus sozialen Netzwerken heran gezogen, um das Ranking zu verbessern. (4) Bei den Projekten speziell für den Nachrichtenbereich, beispielsweise NewsRec, wurden oftmals nur Artikel eine Quelle herangezogen gezogen, wodurch keinesfalls die breiten Interessen von Nutzern abgedeckt werden können. (5) Die bisherigen Projekte nutzen weder Heuristiken noch bieten sie effektive Strategien für Ausnutzung des impliziten Feedbacks. Die Strategien von Joachims et al. für implizites Feedback bieten zwar den richtigen Ansatz, wurden jedoch für Suchmaschinen konzipiert und lassen sich daher nicht auf PNA übertragen.

Aus diesem Grund ergeben sich folgende Abgrenzungsmerkmale:

- (1) Spezialisierung auf Nachrichtenartikel.
- (2) Generierung von Empfehlungen bereits ab dem ersten Nutzer.
- (3) Ausnutzung von Kontextinformationen, vor allem aus den sozialen Netzwerken.
- (4) Aggregation und Evaluierung mit unterschiedlichen Quellen.
- (5) Heuristiken und Ausnutzung von implizitem Feedback.

⁷⁰ Bergamaschi et al. (2007)

⁷¹ Katz (2010)

⁷² Banos et al. (2006)

2.7 Beispiele aus der Praxis

Die folgenden Analysen der beschriebenen Systeme beruhen zum größten Teil auf eigenen Beobachtungen und vor allem auf öffentlichen Informationen beispielsweise durch Interviews. Dies ist dadurch begründet, dass die Unternehmen möglichst wenig Informationen zu ihren Systemen preisgeben müssen, um in dieser noch jungen Disziplin ihren Wettbewerbern keinen Vorteil zu bieten. Die Analysen werden im nachfolgenden Kapitel als Grundlage zur Konzeption eines eigenen personalisierten Nachrichtenaggregators verwendet. Aus diesem Grund werden auch nur die hierfür relevanten Informationen erläutert.

2.7.1 Prismatic

Prismatic ist ein US-amerikanisches Startup, das das Ziel verfolgt den Nutzer von der Informationsüberladung zu befreien und bietet hierfür ein Webinterface sowie Apps für Mobilplattformen an. Aktuell aggregiert Prismatic ausschließlich Artikel in englischer Sprache und bietet dem Nutzer seine eigenen Quellen hinzuzufügen. Gegründet wurde das Unternehmen 2010 von drei PHD Absolventen, die maßgeblich für die Konzeption und Programmierung zuständig sind. Wie alle anderen Startups in dieser Branche sind sie sehr verschwiegen über ihre Vorgehensweise. Die folgenden Informationen beruhen deshalb auf spärlichen Interviews mit den Mitgründern.

Die verwendete Programmiersprache ist Clojure, eine moderne Variante von Lisp, die in Java kompiliert. Große Frameworks wie Hadoop werden nicht verwendet, sondern anstatt dessen kleinere, die nach eigenen Aussagen leichter zu debuggen sind. Prismatic gibt an maschinelles Lernen auf Nutzer sowie auf Dokumente durchzuführen.

Dateneingang

Prismatic nutzt RSS Feeds für die Verarbeitung von neuen Artikeln mit einer Feed Discovery Funktion, sodass lediglich die Startseite eingegeben werden muss. Es wird angegeben, dass Signale (Likes, Shares, Retweets, Kommentare etc.) aus sozialen Netzwerken wie Twitter, Facebook, Google Reader sowie anderen, ungenannten, Stellen im Internet extrahiert werden. Anschließend werden die eingegangenen Links einem Spamfilter unterworfen, worüber u.a. anstößige Inhalte gefiltert werden sollen. Mehr Information werden dazu jedoch nicht veröffentlicht. Die gefundenen Links werden in einem nächsten Schritt aufgerufen und ihr HTML Quellcode mit maschinellem Lernen verarbeitet, um den Artikeltext zu erhalten, das

beste Foto sowie den Autoren, Themenlabel und weitere, nicht näher spezifizierte Daten. Nach eigenen Angaben wurde viel Aufwand in diesen Prozess investiert, um ihn möglichst schlank und effizient zu gestalten. In einem finalen Schritt werden die gesammelten Dokumente von einem „Doc-Master“ in einem Online-System in Story-Cluster unterteilt.

Clusteringverfahren⁷³

Das Clustering wird bei Prismatic, wie zuvor erwähnt, online durchgeführt und muss deshalb performant geschehen. Es geht darum, verschiedene Artikel, von verschiedenen Quellen einem Thema zuzuordnen, damit der Nutzer keine Duplikate angezeigt bekommt, jedoch nur thematisch relevante Artikel unangetastet zu lassen. Im Blog von Prismatic wird erklärt, dass Natural-Language-Processing (NLP) hierbei ein wesentlicher Faktor ist.

In einem ersten Schritt wird eine Named Entity Recognition (NER) auf den Titeln und dem Artikeltext durchgeführt, mit denen Menschen, Firmen und Orte gefunden werden können. Daneben werden die Inhalte mit Stopword-Filterung und Stemming normalisiert. Anschließend werden Unigramme, Bigramme sowie N-Gramme mit Grad 3 bis 8 aus dem verbleibenden Text gebildet.

Es wird ein Korpus erstellt aus den gesamten Artikeln des letzten Tages, bei dem die Features eine an den Arbeitsspeicher angepasste Threshold-Häufigkeit überschreiten müssen. Bei allen darunterliegenden Features wird als Heuristik eine Häufigkeit von 1 angenommen, womit nach eigenen Angaben fast identische Ergebnisse erzielt werden und der Ressourcenaufwand stark reduziert werden kann. Die einzelnen Artikel-Features werden schließlich mit einem TfIdf-Score aus diesem Korpus gewichtet.

Für die Ähnlichkeitsfunktion wird das Skalarprodukt zweier Feature-Vektoren verwendet, nähere Informationen dazu sind nicht vorhanden. Um die vielen Stringmanipulationen zu optimieren wurde ein zweistufiges Verfahren entwickelt, das zunächst nur die Titel-Features (Named Entities sowie N-Gramme mit $n < 8$) verwendet und mit den bisherigen Artikeln vergleicht. Falls die Ähnlichkeit über einem gegebenen Threshold liegt, wird die Berechnung auf den gesamten Artikelinhalt erweitert. Mit diesem Verfahren ist eine ressourcenarme und präzise Art der Ähnlichkeitsberechnungen möglich, die laut Prismatic in 95% der Fälle richtig liegt.

⁷³ Vgl. Prismatic Blog über Clustering (2012, <http://blog.getprismatic.com/blog/2012/4/17/clustering-related-stories.html>, abgerufen 27.8.2013)

Es wird außerdem erwähnt, dass die Vergleiche nicht auf alle Dokumente durchgeführt werden, sondern auch hierbei Heuristiken angewandt wurden, mit denen der Ablauf effizienter gestaltet werden kann. Über den Clustering Algorithmus selbst werden keine Informationen veröffentlicht. Es wird lediglich darauf hingewiesen, dass auch hier mittels Heuristiken nach systematischen Hinweisen gesucht wird, die auf inkorrekt zusammengeführte Cluster deuten. Laut Prismatic ist dieses Verfahren geeignet, um die Dokumente sofort beim Eingang korrekt zu klassifizieren. Im Ergebnis haben bei Prismatic 62% der Cluster eine Größe von 1.

Initiale Empfehlungen

Es wird hervorgehoben, dass innerhalb weniger Sekunden ausreichend historische Daten zu einem Nutzer gesammelt und analysiert werden können, um dem Nutzer aussagekräftige Themen- und Quellenempfehlungen angeboten werden können. Dies funktioniert allerdings nur, wenn der Nutzer sich über sein soziales Profil (Facebook, Twitter, Google Reader) anmeldet und dort zuvor auch aktiv Inhalte verbreitet bzw. kuratiert hat. Nach eigenen Beobachtungen werden dabei, wahrscheinlich aus Performanzgründen, in erster Linie die Meta-Keywords im Header einer Webseite extrahiert. Diese Relevanz bildet sich dabei nicht durch die kummulierte Häufigkeit dieser Begriffe, sondern eher durch eine TfIdf Repräsentation. Dem Nutzer werden anschließend die passenden Keywords angezeigt, wodurch er sehr schnell einen positiven Eindruck und damit ein Vertrauen in die Lernmechanismen erhält. Dieser initiale Prozess dauerte im Anfangsstadium von Prismatic 24h und der Nutzer wurde anschließend über die Fertigstellung seines Profils via Email informiert. Ein wesentlicher Grund dafür könnte sein, dass früher der komplette Inhalt der Webseite analysiert wurden, was diesen zusätzlichen Zeitaufwand erklären würde. Die Abweichung davon könnte durch verbessertes Caching der analysierten Seiten, einen nur geringen Informationszuwachs gegenüber einfachen Keywords und anderen Heuristiken und der fehlenden Skalierbarkeit bedingt sein.

Lernprozess

Der Nutzer hat die Möglichkeit, dem System explizites Feedback in Form von „+“ und „remove“ zu geben. Diese Signale werden von Prismatic hoch priorisiert und sollen sich möglichst schnell im Nachrichtenfeed des Nutzers niederschlagen, was durch Heuristiken in einer Art Online-Learning erreicht wird. Als Beispiel wird hierfür ein Nutzer genannt, der fünf

Artikel von einer Quelle entfernt hat. Dies wird als starkes Signal gegen die Quelle interpretiert, wodurch diese Quelle fortan nicht mehr angezeigt wird. Alle Interaktionen des Nutzers werden hierbei persistent gespeichert.

Neben dem Online-Learning wird ein Offline-Batch-System am Rande erwähnt, das etwaige Korrekturen des Online-Systems vornimmt und feinere Modelle des Nutzers erstellen kann.

Global Newsfeed⁷⁴

Das Team von Prismatic kam zu der Erkenntnis, dass Nutzer nicht nur personalisierte Artikel zu ihren eigenen Themen wünschen, sondern auch informiert werden möchten, über die wichtigen Themen in einer weitaus breiteren Fassung. Aus diesem Grund hat Prismatic einen „Global Newsfeed“ eingeführt, der zwei wesentliche Faktoren berücksichtigt: Einerseits die Popularität in den sozialen Netzwerken, andererseits die allgemeine Häufigkeit zu einem Thema über ein Clustering. Ein vorheriges reines Ranking nach der Popularität hatte eine starke Verzerrung zur Folge, zugunsten von Themen über soziale Netzwerke. Die anschließende Ranking-Funktion ordnet die Cluster mit einer Mindestgröße von fünf zunächst nach ihrer Größe und addiert einen Bonus für die soziale Popularität.

Nutzeroberfläche

Der Nutzer erhält bei Prismatic einen Nachrichtenfeed, der seine Artikel aus verschiedenen Quellen widerspiegelt. Dieser Feed wird entweder als Liste oder in unterschiedlich großen Kacheln dargestellt. Der Nutzer kann hier verschiedene Elemente eines Artikels, wie die Quelle, Kategorie oder Tags mit „mehr“ oder „weniger“ gewichten.

2.7.2 Piano Media

Das slowakische Unternehmen Piano Media bietet eine Paywall für Verleger in der Slowakei, Slowenien und Polen an. Der Nutzer erlangt Zugang zum Paid Content mithilfe eines einzelnen Abonnements und die Verleger werden für ihre Beiträge nach dem Leseverhalten des Nutzers entlohnt. Der technologische Unterschied zu einem PNA besteht darin, dass die Nachrichten nicht aggregiert werden, sondern der Nutzer wie üblich die Seite jedes teilnehmenden Anbieters besuchen und sich ggf. dort einloggen muss. Es findet keinerlei Personalisierung statt. Die Verleger kümmern sich selbst um den Content, Piano Media

⁷⁴ Vgl. Prismatic Blog über den Global Newsfeed (2012, <http://blog.getprismatic.com/blog/2012/5/11/prismatic-global-newsfeed-1.html>, abgerufen am 27.8.2013)

übernimmt hierbei letztendlich die Rolle der Autorisierung. PNAs grenzen sich hierbei vor allem durch die vorhandene Nutzerbasis und der aktiven Akquirierung von Kunden innerhalb dieser Nutzerbasis ab, da sie einen Zusatznutzen bieten. Damit sind sie in der Lage, effizienter Umsätze zu generieren als ein reines Paywall-Modell.

2.7.3 Feedly

Feedly wurde zuerst 2008 vorgestellt und ist ein Nachrichtenaggregator, der dem Nutzer einen Nachrichtenfeed aus verschiedenen RSS-Feeds zusammenstellt. Aufgrund der Ähnlichkeit zu Google Reader hat Feedly stark von dessen Wegfall im Jahre 2013 profitiert und ist seitdem einer der größten Aggregatoren auf dem Markt.⁷⁵

Relevant ist Feedly einerseits für diese Arbeit, da Feedly einen hohen Grad an Personalisierung im Sinne von Customizing bietet. Die Nutzungsoberfläche spielt bei Feedly steht bei Feedly im Vordergrund, wie von Oechslein und Hess empfohlen.⁷⁶ Nutzer können zwischen verschiedenen Themen und Farben auswählen und zudem aus diversen Stilen für die Nachrichtenanzeige (Kacheln, Magazin, Liste mit wenig und viel Informationen) wählen. Dadurch sind die Nutzer in der Lage sich ihre Oberfläche genau nach ihren Wünschen zu gestalten. Eine systemische Personalisierung mithilfe von Recommender Systemen existiert bei Feedly nicht.

Andererseits ist Feedly relevant, da seit Mitte 2013 eine kostenpflichtige Pro-Version angeboten wird, nach eigenen Angaben, um die finale Version von Feedly finanzieren zu können. Bei dieser Pro-Version erhalten die Nutzer aktuell die Möglichkeit ihre Wunschartikel zu Evernote sowie Pocket zu übertragen und außerdem eine Suche in den Feeds durchzuführen. Feedly weist jedoch daraufhin, dass noch weitere Features folgen werden. Für diese Pro-Variante verlangt Feedly \$5 pro Monat, \$45 pro Jahr oder \$99 auf Lebenszeit. Nach eigenen Angaben wurden die ersten 5000 Probeverträge innerhalb weniger Stunden ausverkauft.⁷⁷ Damit decken sich die Preise mit der Zahlungsbereitschaft (0,42 Euro bis 6,83 Euro pro Monat) aus der aktuellen Forschung von Oechslein und Hess.⁷⁸

⁷⁵Vgl. The Next Web über Feedly (2013, <http://thenextweb.com/insider/2013/06/19/feedly-launches-standalone-web-version-for-all-users-offers-a-one-click-migration-path-from-google-reader/>, 30.8.2013)

⁷⁶Oechslein & Hess (2013)

⁷⁷Vgl. Internet World Business Artikel über Premium Dienst von Feedly (2013, <http://www.internetworld.de/Nachrichten/Medien/Medien-Portale/Feedly-plant-kostenpflichtige-Pro-Version-Premium-Dienst-ab-fuenf-US-Dollar-78648.html>, 30.8.2013)

⁷⁸Oechslein & Hess (2013)

Genauere Zahlen über die Conversion Rate oder die Anzahl der Abschlüsse veröffentlicht Feedly nicht. Es ist jedoch davon auszugehen, dass im Vorfeld Untersuchungen bei der Nutzergruppe durchgeführt wurden. Das Geschäftsmodell von Feedly stellt das klassische Freemium Modell dar, bei dem in der Regel eine Conversion Rate von 1 bis 10 Prozent erreicht wird, im Schnitt jedoch zwischen 2 und 4 Prozent.⁷⁹

⁷⁹Vgl. Maltz (2013), S.4

3 Konzeptuelles Design

Im vorherigen Kapitel wurde bereits erläutert, dass Nutzer an einer Personalisierung interessiert und sogar bereit sein könnten, hierfür zu zahlen. Dieses Kapitel widmet sich nun der Konzeption eines PNA.

Begonnen wird mit der Frage, was der Nutzer von solch einem System erwartet bzw. was seine Bedürfnisse dabei sind. Im darauffolgenden Abschnitt wird erläutert, wie man dies erreichen kann. Dabei werden zunächst Strategien vorgestellt, wie man das Nutzerfeedback, vor allem das implizite, besser ausnutzen kann. Anschließend werden verschiedene Klassifizierungsalgorithmen vorgestellt, deren Eignung für die Domäne Nachrichten zum Schluss getestet werden soll. Außerdem werden Vorschläge gemacht, wie man das Ranking anhand von sozialen Netzwerken und den Daten eines Nachrichtenanbieters verbessern kann.

3.1 Anforderungen des Nutzers an die persönliche Zeitung

In diesem Abschnitt soll zunächst geklärt werden, wofür sich der Leser eigentlich interessiert und was er wirklich möchte. Es ist dafür notwendig die Kriterien festzuhalten, die für den Leser dabei von Bedeutung sind. Nur so ist es möglich, adäquate Aussagen darüber zu treffen, welche Artikel ihn interessieren und welche nicht.

Für diese Kriterien wurde einerseits eine Nutzerbefragung in einer Gruppe (n=30) technikaffiner Internetnutzer gestartet, die ein Interesse für das tägliche Geschehen haben. Sie sollten ihre Wünsche an einen PNA möglichst einfach formulieren, zum Schluss wurden die Ergebnisse aggregiert. Andererseits wurde das Internet im nicht-wissenschaftlichen Blog-Bereich durchsucht, da hier teilweise Experten ihre Meinungen und Visionen zu diesem Thema äußern.^{80 81}

Die gesammelten Daten darüber lassen sich in zwei Gruppen unterteilen:

1. Kriterien zur Relevanz eines Artikels.
2. Richtlinien für das Nachrichtensystem aus Nutzersicht.

⁸⁰ Markus Breuer, „Die persönliche Zeitung – Die Zeit ist reif“ (<http://notizen.steingrau.de/2011/05/17/die-personliche-zeitung-die-zeit-ist-reif/>, Abruf: 28.9.2013)

⁸¹ Martin Weigert, „Das Streben nach Relevanz“ (<http://netzwertig.com/2011/07/25/personalisierung-das-streben-nach-relevanz/>, Abruf: 28.09.2013)

bewertet. Aus diesem Grunde sollen im Folgenden Aspekte erläutert werden, an denen sich die Entwicklung im weiteren Verlauf stets orientieren wird.

- **Nutzung von impliziten Feedback:** Nutzer sollen nicht nur über explizites Feedback das System anlernen können, sondern vor allem auch durch ihr Verhalten und damit implizites Feedback.
- **Explizites Feedback gewünscht:** Nutzer müssen bei der Artikelauswahl die Möglichkeiten zur Einflussnahme erhalten, um die Anwendung ihren Präferenzen anpassen zu können. Fehlt dieser Aspekt, fühlen sich Nutzer machtlos gegenüber den Algorithmen.
- **Schnelle Umsetzung der Nutzereingaben:** Expliziten Eingaben des Nutzers müssen sich schnell in den Empfehlungen niederschlagen, ansonsten könnte der Nutzer das Vertrauen an das System verlieren.
- **Erläuterung der Empfehlungen:** Um das Nutzervertrauen weiter zu stärken, sollen die Empfehlungen nachvollziehbar begründet werden.
- **Anbindung an die sozialen Netzwerke:** Das Teilen von Artikeln ist aktuell ein sehr wichtiger Faktor für Nutzer, weshalb dies auch im Vordergrund stehen soll.

3.2 Gesamtarchitektur

Dieser Abschnitt stellt die grobgranulare Architektur eines PNA vor. Dadurch soll zunächst ein Überblick über die wesentlichen Komponenten, ihre Einordnung im Gesamtsystem und deren jeweilige Aufgaben geschaffen werden (siehe hierzu auch Abbildung 3-2:Konzeption der Gesamtarchitektur). Eine genaue Beschreibung der einzelnen Komponenten erfolgt in den nachfolgenden Abschnitten.

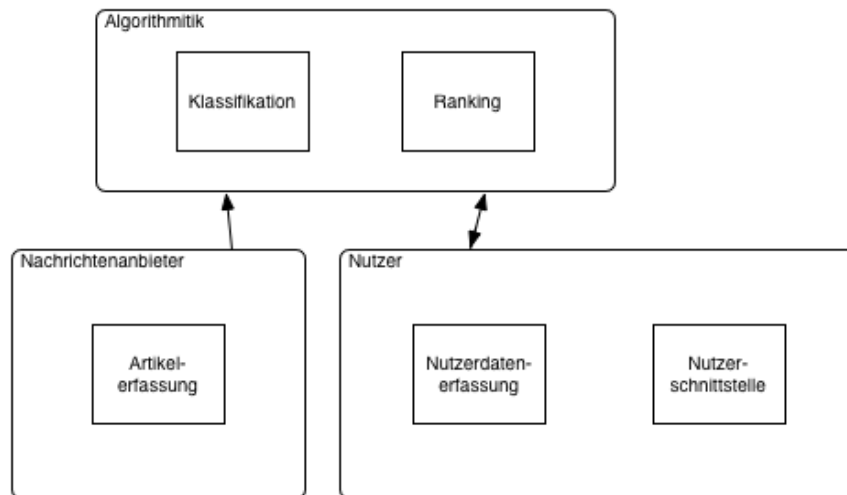


Abbildung 3-2:Konzeption der Gesamtarchitektur

Artikelerfassung (3.3): *Welche Daten werden später klassifiziert?*

Die Daten für den hier vorgestellten Nachrichtenaggregator kommen ausschließlich von RSS-Feeds der Nachrichtenseiten. Diese sind öffentlich abrufbar und die Haupt-Feeds von den Anbietern oftmals in den Meta-Informationen einer Startseite vermerkt, wodurch sie leicht ausgelesen werden können. Es sollen hierbei nicht nur Feeds von einer einzigen Nachrichtenseite bzw. Quelle genutzt werden, sondern ein breites Portfolio an *General Topic* Nachrichtenseiten und thematisch spezialisierten Seiten erstellt werden.

Nutzerdatenerfassung (3.4, 3.5): *Was ist über den Nutzer und seine Präferenzen bekannt?*

Zur Erstellung von präferenzbasierten Nachrichtenempfehlungen ist es von essentieller Bedeutung, die Interessen eines Nutzers *schnell und präzise* zu erfassen. Es soll gezeigt werden, dass ein System, das lediglich auf dem expliziten Feedback eines Nutzers beruht, schlechtere Ergebnisse erzielt als ein hybrides, das ebenso implizites Feedback nutzt. Auf alternative Wege, Nutzerpräferenzen zu ermitteln, wird im Ausblick hingewiesen.

Klassifikation (3.6): *Welche Artikel sind interessant und welche uninteressant?*

Nachdem die persönlichen Präferenzen eines Nutzers vorliegen, soll für ihn eine Auswahl an interessanten Artikeln zusammengestellt werden. Hierfür wird eine Klassifikation durchgeführt, die für jeden Artikel entscheidet, ob er für den Nutzer potentiell relevant ist oder nicht.

Ranking (3.7): *Was ist die richtige Reihenfolge für die Artikel?*

Für das Ranking sollen einerseits die verfügbaren Daten aus den *sozialen Netzwerken* genutzt werden. Diese geben Rückschlüsse über die Popularität der Artikel. Es wird auch angenommen, dass sich damit Aussagen zur Qualität eines Artikels machen lassen.

Andererseits sollen die kuratierten, journalistischen Informationen von den Nachrichten(start)seiten selbst hinzugezogen werden. Diese wurden von den Anbietern bzw. deren Redakteuren mittels eines Nachrichtenwertes⁸² manuell bestimmt und bieten eine wertvolle aggregierte Information.

Nutzerschnittstelle: *Wie werden die Artikel dem Nutzer präsentiert?*

Zuvor wurde bereits die zu überprüfende Annahme geäußert, dass ein System, ausschließlich basierend auf explizitem Feedback schlechtere Ergebnisse liefert. Nichtsdestotrotz hat sich der Wunsch nach persönlicher Einflussnahme deutlich gemacht. Es soll sich deshalb bei dem User Interface primär an den *Richtlinien für das System aus Nutzersicht* im vorherigen Kapitel 3.1 orientiert werden.⁸³

3.3 Auswertung von RSS-Feeds

In diesem Abschnitt soll erläutert werden, welche primären Informationen zu einem Artikel beim erstmaligen Abruf gesammelt werden sollen und wie dies geschieht. Diese gewonnenen Daten sollen primär für das spätere Content-Filtering verwendet werden.

Zunächst sollen RSS-Feeds automatisiert, allein durch Quellenangabe gefunden werden. Hierzu können die Verweise auf RSS-Feeds in den Meta-Daten der Nachrichtenstartseiten genutzt werden. Die dazugehörigen Feeds umfassen i.d.R. folgende Informationen:

- Titel (50-90 Zeichen)
- Kurzzusammenfassung (250-300 Zeichen)
- Link zum Vorschaubild

⁸² Weiterführende Informationen in Kepplinger (2011).

⁸³ Dieser Aspekt wird in diesem Kapitel nicht weiter beleuchtet, jedoch wird er der Vollständigkeit halber bei Gesamtarchitektur erwähnt.

-
- Kategorie und/oder Stichworte zum Artikel
 - Veröffentlichungsdatum
 - Link zum Artikel selbst (in den meisten Fällen maskiert)

Bei fehlenden Artikelinformationen sollen die Meta-Daten (Open-Graph Format) auf der einzelnen Artikelseite ergänzend wirken, die ursprünglich für die Vorschauanzeige bei Facebook von den Nachrichtenanbietern eingerichtet wurden.⁸⁴ Diese Meta-Daten beinhalten oftmals Werbung für den Nachrichtenanbieter, welche ggf. entfernt werden muss.

Aufbauend auf dem vorherigen Wissen durch die Feed-Extraktion wären zwei Strategien zur Volltext-Extraktion naheliegend:

Template-Extraktion: Die erste Variante basiert auf zuvor festgelegten Vorlagen (*Templates*), die feste Strukturen in der Darstellung der Inhalte nutzen. Dies wird dadurch möglich, dass Autoren von Artikeln für ihre Inhalte Content-Management-Systeme nutzen, die sie für sie in maschinenlesbaren HTML-Code transformieren. Konstante Teile einer Webseite (z.B. Header, Footer, oder Seitenleiste) können so erkannt und schnell entfernt werden. Problematisch hingegen sind dynamische Inhalte wie bspw. Werbung zum relevanten Inhalt zugeordnet werden, was zu Verfälschungen führt. Andererseits können mehrseitige Artikel nur schwer erkannt werden.

Ausnutzung des bekannten Wissens: Bei der zweiten Variante nutzt man das bereits bekannte Wissen zu einem Artikel über Titel und Kurzzusammenfassung und extrahiert damit das umschließende oder die nachfolgenden DOM-Elemente mit dem größten zusammenhängenden Text. Hierbei problematisch erweist sich, dass die bekannten Elemente teilweise nur in abgeänderter auftreten, meistens mit Werbung für den veröffentlichenden Verlag.

Neben der Verunreinigung durch irrelevante Inhalten haben beide Methoden ein konzeptuelles Problem. Artikel unterscheiden sich in ihrer Länge teilweise signifikant, wodurch etwaige Schlagwörter mit ansteigender Artikellänge verwässert werden können. Die Kurzzusammenfassungen bzw. *Abstracts* hingegen sind in den meisten Fällen 250-300 Zeichen lang. Da es sich hierbei in der Regel um Zusammenfassungen oder Teaser zum nachfolgenden

⁸⁴ Der verwendete Standard für diese Meta-Daten wird von Facebook Open-Graph genannt.

Text handelt, umfassen diese Texte nach eigener Untersuchung überdurchschnittlich viele relevante Stichworte. Zusammen mit dem Tokens aus dem Titel lässt sich so sehr schnell auf den Inhalt des Artikels schließen, was außerdem die Verarbeitung begünstigt, die diese so verkürzt wird. Aus diesem Grund werden im Folgenden auf die Extraktion der *gesamten Artikelinhalte verzichtet* werden.

3.4 Erfassung der Nutzerdaten

In diesem Abschnitt soll ein kurzer Überblick darüber gegeben werden, *was* vom Nutzer aufgezeichnet wird.⁸⁵ Es beschreibt also nur die Roheingangsdaten. Die *Verwendung* und etwaige Strategien der aufgezeichneten Daten wird in einem späteren Abschnitt erläutert.

Implizit

- Klicks auf Artikel, Rubriken und Einstellungen.
- Übersprungene Artikel im Nachrichtenstream.
- Gewichtete Häufigkeit der Impressionen.

Explizit

- „Daumen hoch, „Daumen herunter“ für ganze Artikel.
- *Mehr/weniger von einer Kategorie, einem Nachrichtenanbieter oder einem Keyword aus dem Artikel.*⁸⁶

Die beschriebenen Erfassungswege müssen nicht gemeinsam genutzt werden, können in Kombination jedoch *schneller* zu *besseren* Ergebnissen führen. Die Lernphase soll mit der ersten Nutzung beginnen und nicht enden, da sich auch Interessen des Lesers über die Zeit ändern können.

Die ausschließliche Nutzung von impliziten Daten wird nicht als vorteilhaft angesehen, da das Feedback der Pilotnutzer gezeigt hat, dass Benutzer auch manuell personalisieren möchten

⁸⁵ Neben den hier aufgezeigten Möglichkeiten existieren noch weitere, wie bspw. die Abfrage des Nutzerprofils bei Facebook oder Twitter. Diese werden in dieser Arbeit nicht genutzt und daher nur im Ausblick erwähnt.

⁸⁶ Diese Art der manuellen Personalisierung wird als äußerst sinnvoll erachtet, jedoch nicht im Rahmen dieser Arbeit weiter untersucht.

und nicht die gesamten Inhalte maschineller Auswertung überlassen wollen. Eine Kombination aus beiden Verfahren der richtige Weg zu zufriedenen Benutzern ist.

3.5 Strategien für implizites Feedback

Im folgenden Abschnitt sollen Strategien zum impliziten Feedback erläutert werden, die auf Beobachtungen, speziell im Nachrichtenbereich, beruhen. Daran geknüpft werden die getroffenen Annahmen. Ziel des impliziten Feedbacks ist es, mehr Relevanzbewertungen zu generieren und dadurch den Lernvorgang einerseits schneller und andererseits genauer zu gestalten.

Zu Beginn der Arbeit wurden implizite Feedback Strategien von Joachims et al. (2007) für den Bereich der Suchmaschinen erläutert. Sie sollen als Grundlage für die eigenen Strategien dienen, obwohl sie sich nicht direkt auf den Nachrichtenbereich übertragen lassen, da hier eine andere Zielsetzung gegeben ist, die das implizite Feedback erschwert.

- a) Bei Suchmaschinen möchte man in den allermeisten Fällen auf eine Webseite weitergeleitet werden, die eine Antwort auf die gesendete Suchanfrage liefert. Bei Nachrichten hingegen verbirgt sich die Antwort oftmals schon in der Schlagzeile oder im Abstract. Nachrichtenleser möchten in erster Linie informiert sein und um informiert zu sein, müssen sie aus diesem Grund nicht zwingend den dahinterliegenden Artikel lesen bzw. öffnen.
- b) Außerdem werden bei einer Suchanfrage Einträge genau zu einem Thema angezeigt, falls es sich nicht um Ambiguitäten handelt. Bei einem Nachrichtenaggregator versucht man thematische Duplikate jedoch weitestgehend zu vermeiden, wodurch man eine starke thematische Diversifikation erhält. Der Leser ist deshalb nicht auf der Suche nach einem bestimmten Eintrag, sondern auf der Suche nach einer unbestimmten Menge an relevanten Einträgen.

Im Folgenden sollen implizite Feedback Strategien erläutert werden, die direkt positive als auch negative Trainingsbeispiele generieren oder anderweitig genutzt werden. Auf eine feingranulare Verarbeitung des impliziten Feedbacks wird der Einfachheit halber im Rahmen dieser Arbeit verzichtet.

Sitzungen (Sessions)

Zunächst ist für das implizite Feedback die Definition einer Sitzung notwendig. Dies ist daher notwendig, da bestimmte Strategien mehrere Klicks eines Nutzers in Betracht ziehen. Es handelt sich hier bei einer Sitzung um alle Klicks, die mit einem Seitenaufruf, also einer einmaligen Zusammenstellung von Nachrichtenartikeln, zusammenhängen. Sie ist in ihrer zeitlichen Hinsicht auf maximal eine Stunde begrenzt. Führt der Nutzer einen Refresh der Seite durch, wird die alte Sitzung geschlossen und eine neue eröffnet. Die folgenden Strategien sollen erst dann umgesetzt werden und nicht schon vorher.

Feedback Strategien für eindeutig positive Beispiele

Die trivialste Möglichkeit ein aussagekräftiges positives Trainingsbeispiel zu erhalten, ist die angeklickten Artikel zu nutzen (*ArticleClick*). Wenn der Leser auf einen Artikel klickt um ihn zu lesen, ist mit einer großen Wahrscheinlichkeit davon auszugehen, dass er sich auch für diesen interessiert. Etwaige Fehlklicks sind hierbei völlig vernachlässigbar. Explizite positive Bewertungen im Anschluss verstärken das Signal, während bei negativen Bewertungen das implizite Feedback neutralisiert wird.

Wie bereits zuvor erwähnt, reicht es bei vielen Meldungen aus, nur die Überschrift oder den Abstract zu lesen, um informiert zu sein. Um diese Problematik miteinzuschließen, wäre zunächst denkbar, die Mausbewegung und das Scrollverhalten zu analysieren. Allerdings sind derartige Mechanismen sehr ungenau und die Wahrscheinlichkeit für ein falsches Signal zu hoch (Leser ist kurzzeitig nicht am Computer, unterschiedliche Länge der Artikel, manche Leser öffnen zunächst alle interessanten Artikel in Tabs etc.), weshalb sie in dieser Arbeit nicht zur Anwendung kommen.

Feedback Strategien für nichteindeutige Beispiele

Diese Kategorie bezieht sich auf Einträge, die nicht eindeutig zugewiesen werden können, von denen jedoch verwertbare Aussagen abgeleitet werden können. Wie zuvor erwähnt, reicht es bei manchen Artikeln schon aus, die Überschrift und den Abstract zu lesen. Außerdem ist die Reihenfolge der angezeigten Artikel bekannt. Klickt ein Leser auf den Artikel n in einer Listendarstellung, so ist davon auszugehen, dass er die vorangegangenen Artikel 1 bis $n-1$

ebenfalls wahrgenommen hat.⁸⁷ Daraus lassen sich zwei grundlegende Schlussfolgerungen ableiten:

1. Der Leser hat die vorherigen Einträge gesehen, jedoch nicht als relevant eingeschätzt (Hinweis auf negative Trainingsbeispiele).
2. Der Leser hat die vorherigen Einträge gesehen, ihm haben jedoch die verfügbaren Informationen ausgereicht (Hinweis auf positive Trainingsbeispiele).⁸⁸

Trotz der nicht eindeutigen Zuordnung dieser Einträge, lässt sich festhalten, dass der Nutzer diese Einträge zumindest teilweise gesehen haben sollte. Es ist also davon auszugehen, dass sie dem Leser keine neuen Informationen liefern und er sie auch in Zukunft wahrscheinlich nicht anklicken wird. Die Strategie hierfür ist demnach, die „übersprungenen“ Artikel in der nächsten Sitzung auszublenden (*SkipCountNeutral*). Um sicher zu gehen, dass der Leser die übersprungenen Artikel tatsächlich alle wahrgenommen hat, soll zusätzlich noch ein Schwellenwert t_j gesetzt werden. Mit dieser Strategie wird eine Dynamik in die Ergebnisliste gebracht, die weniger relevante Artikel mit kontinuierlicher Nutzung verschwinden lässt.

Ein weiterer häufiger Fall ist, dass ein Leser ohne einen Klick auf die Seite geht und sich vollständig mit den Überschriften zufrieden gibt. Eine zweite Strategie soll dafür sorgen, dass die Ergebnisliste nicht dauerhaft konstant bleibt, indem mit jeder Anzeige ein Decay-Wert inkrementiert wird (*NoClickCount*). Diese Erhöhung geschieht in Abhängigkeit von der Position des Artikels und soll degressiv abnehmen. Damit wird abgebildet, dass die Wahrscheinlichkeit einen oberen Eintrag zu sehen höher ist, als die eines späteren Eintrags. Wie zuvor soll hier ebenfalls ein Schwellenwert t_s gesetzt werden, nachdem sie verschwinden.

Die Kombination aus beiden Strategien lässt einen Artikel verschwinden die Bedingungen zum Teil erfüllt sind, wodurch der Schwellenwert schneller erreicht wird (*NoClickSkipCount*).

Feedback Strategien für eindeutig negative Beispiele

Explizites Feedback ist für den Leser sehr mühselig und wird auch nur von Wenigen konsequent durchgeführt. Zuvor wurde eine sehr aussagekräftige Strategie vorgestellt, wie man positive Trainingsbeispiele erhält. Negative Beispiele sind weitaus schwieriger zu

⁸⁷ Der Fall, dass der Nutzer an eine zufällige Stelle springt und deshalb die Artikel davor nicht wahrnimmt tritt so selten auf, dass er vernachlässigt werden kann.

⁸⁸ Ein Beispiel hierfür wäre die Schlagzeile, dass die Arbeitslosenquote auf 7,2% gesunken ist. Für viele Leser reicht dieser Fakt bereits aus und sie lesen nicht den gesamten Artikel, da oft die wesentlichen Informationen schon in den Abstract geschrieben wurden. Nichtsdestotrotz kann ein Interesse für solche Nachrichten bestehen.

erhalten. Aus diesem Grund sollen im Folgenden mehrere, jedoch schwächere Strategien vorgestellt werden.

Im vorherigen Abschnitt zu den neutralen Strategien wurde festgelegt, dass übersprungene Einträge nicht eindeutig zuordenbar sind und deshalb nach Überschreitung eines Schwellenwertes nicht mehr angezeigt werden sollen. Eine zu überprüfende Sichtweise wäre hingegen anzunehmen, dass übersprungene Einträge nach dem Schwellenwert als negative Beispiele anzusehen sind (*SkipCountNegative*). Dies geht mit der veränderten Sichtweise einher bezüglich der Frage, was der Leser eigentlich sehen möchte. Zuvor wurde angenommen, dass dem Leser bei einigen Artikeln ausreicht, nur Überschrift mit Abstract zu lesen, um informiert zu sein und deshalb auch diese Artikel eine wichtige Rolle einnehmen. Mit dieser Strategie ist nun jedoch das Primärziel die Anzahl der Klicks zu maximieren, da weniger relevante Artikel nun verworfen werden. Es ist zu erwarten, dass hierdurch jedoch einige Artikel verworfen werden, die der Leser allerdings interessant findet, weshalb sie unbedingt getrennt evaluiert werden muss.

Die weiteren Strategien sind stark an die von Joachmis et al. angelehnt, bei denen mittels Eyetracking bestätigt wurde, dass die ersten Ergebnisse, insbesondere *das Erste*, mehr Aufmerksamkeit vom Anwender erhalten als spätere. Es wird vermutet, dass dies trotz veränderter Zielsetzung bei Nachrichten ebenfalls der Fall ist. Daher wird eine Strategie vorgeschlagen, die das erste Element den negativen Beispielen zuordnet, wenn es übersprungen wurde (*TopNoClick*). Hierfür soll ebenfalls ein Schwellenwert t_{Top} eingesetzt werden, der jedoch niedriger ist als die vorherigen Schwellenwerte. Diese Strategie steht nicht im Widerspruch zur vorher beschriebenen, sondern soll allenfalls ergänzend wirken. Diese Strategie ließe sich auch auf die ersten fünf Einträgen erweitern, da das primäre Ziel bei diesen Einträgen eine hohe Klickrate sein sollte.

Hat ein Leser einen Artikel angeklickt, so ist davon auszugehen, dass er den direkt umgebenden Artikeln eine erhöhte Aufmerksamkeit schenkt. Aus diesem Grund lässt sich einerseits die Strategie ableiten, dass der Vorgänger weniger relevant ist und den negativen Beispielen zugeordnet werden soll (*ClickPredecessorNegative*). Andererseits soll auch der Nachfolger zu den Negativen gezählt werden. Diese Trennung der Strategien soll die

Unabhängigkeit gewährleisten, da der Vorgänger voraussichtlich mehr Aufmerksamkeit erhält als der Nachfolger. Bei Joachims et al. wurden Strategien sehr ähnlich zu diesen vorgestellt.⁸⁹

Ein weiterer Strategieansatz wäre es, Gemeinsamkeiten bei den übersprungenen Artikeln zu suchen und anhand dieser die Abneigungen des Lesers abzuleiten. Eine triviale Möglichkeit besteht darin, sich die ausgelassenen Nachrichtenanbieter, Kategorien oder RSS-Feeds anzusehen. Wird über eine Heuristik festgestellt, dass ein bestimmtes Merkmal immer wieder auftritt, könnte man dies nach einem Refresh nutzen. Dies wird jedoch im Rahmen dieser Arbeit nicht verfolgt.

Bei maschinellen Lernalgorithmen wird oftmals eine binäre Klassifikation durchgeführt. Es wird jedoch angenommen, dass dies den vorliegenden Sachverhalt nicht adäquat abbildet, da wie zuvor beschrieben auch eine Nutzerrückmeldung entsteht, die nicht eindeutig zuordenbar ist. Die genannten Strategien werden in nachfolgender Tabelle noch einmal kurz zusammengefasst (Tabelle 2).

Name der Strategie	Beschreibung	Verwertung
ArticleClick	Leser klickt Artikel an, um ihn zu lesen.	+
SkipCountNeutral	Übersprungene Artikel werden neutral gewertet, verschwinden jedoch aus der Ansicht.	0
NoClickCount	Wenn kein Klick ausgeführt wurde, dann sollen nach die obersten Artikel verschwinden.	0
NoClickSkipCount	Kombination aus den beiden vorangegangenen mit niedrigeren Schwellenwerten.	0
SkipCountNegative	Dasselbe wie die zweite Strategie, jedoch mit negativer Bewertung.	-
TopNoClick	Der erste Artikel wurde übersprungen.	-
ClickPredecessorNegative	Der Artikel vor einem geklickten ist nicht relevant.	-
ClickSuccessorNegative	Der Artikel nach einem geklickten ist nicht relevant.	-

Tabelle 2: Implizite Feedback Strategien.

⁸⁹ Joachims et al. (2007)

3.6 Klassifikation von Nachrichtenartikeln

In diesem Abschnitt soll erläutert werden, wie man herausfindet, ob ein Artikel für einen Nutzer relevant oder nicht relevant ist. Hierzu sollen mehrere Klassifikationsalgorithmen angewandt und gezeigt werden, in welchem Zusammenhang und auf welche Art diese Algorithmen einzusetzen sind. Das Ergebnis dieses Schrittes ist demnach eine *Menge* von Artikeln, die für den Nutzer *persönlich interessant* sind. Die Datengrundlage hierbei wurde durch das implizite und explizite Leseverhalten des Benutzers geschaffen.

Es soll im Folgenden auf Grundlage der *persönlichen Interessen* ein gewichteter, kombinierter Score berechnet werden. Wobei auch die Daten aus den sozialen Netzwerken bzw. von den Startseiten der Nachrichtenanbieter wieder Verwendung finden, die im weiteren Verlauf noch genauer erläutert werden. Außerdem sollen spezielle Heuristiken angewandt werden, die die Klassifikation noch weiter verbessern. Abschließend wird eine Umsortierung vorgenommen, damit dem Nutzer Artikel aus unterschiedlichen Interessengebieten angezeigt werden und er nicht in einen Tunnel mit ausschließlich einem Bereich gelangt. Dieser Aspekt wird abschließend im Kapitel zum Ranking behandelt.

3.6.1 TfIdf Nutzer-Term-Repräsentation

Die Berechnung des TfIdf ist ein zentrales Merkmal, was für alle weiteren Algorithmen benutzt wird. Es beschreibt, wie oft ein Begriff relativ einem speziellen Kontext vorkommt und wie oft in den Artikeln insgesamt.

Von diesem Prinzip soll Gebrauch gemacht werden. Die Besonderheit ist jedoch, dass hier für jeden Nutzer einmalig eine eigene TfIdf-Verteilung aus den gelesenen bzw. bewerteten Artikeln erstellt wird und diese dann im Folgenden genutzt wird.

Jedoch soll sich die hier verwendete Form an einer Stelle vom üblichen TfIdf unterscheiden: Hier wird für jeden Term, der innerhalb eine (über die Zeit) gewichtete Häufigkeit innerhalb der Nutzerartikel erstellt.

Der Grundgedanke ist, dass man alle Terme aus den Nutzerartikeln als eine große Menge betrachtet und zunächst die Häufigkeit dieser Terme zählt (subjektive Komponente). Damit erhält man eine noch nicht aussagekräftige Verteilung, da vor allem Stoppwörter bis hier her eine hohe Relevanz einnehmen. Jedoch sollten hier bereits Auffälligkeiten von bestimmten

Termen auffallen. Hat der Nutzer beispielsweise 20 Artikel über Darmstadt gelesen, so sollte der Term „Darmstadt“ bereits sehr weit oben in der Liste auftauchen.

Um diese Anomalien besser sichtbar zu machen, soll die inverse Dokumentfrequenz dienen. Dadurch werden nicht aussagekräftige Terme, wie Stoppwörter, sofort in ihrer Wichtigkeit gesenkt (objektive Komponente). Da anzunehmen ist, dass der Term „Darmstadt“ nicht besonders häufig in der zugrundeliegenden Dokumentensammlung auftritt, sollte er nunmehr ganz oben in der Häufigkeitsverteilung stehen.

Durch einen kurzen Blick auf die Gewichtungen wird schnell ersichtlich, welche Begriffe unter den Nutzerartikeln am häufigsten vorkommen. Daraus lässt sich sehr schnell schließen wofür er sich interessiert. Der Vorteil hierbei ist, dass einmalig die subjektive Häufigkeitsverteilung der Nutzerterme durchgeführt werden muss. Anschließend können die subjektiven Häufigkeitsverteilungen genutzt werden, ohne sie jedes Mal neugenerieren zu müssen. Auch Updates sind so leicht möglich, wodurch diese Architektur sehr skalierbar ist. Dabei ist es extrem wichtig, dass diese Werte richtig gebildet werden.

1. Annahme: Unterschiedliche Kategorien bedeuten unterschiedliche Wortverteilungen.

In der Kategorie Technik ist die Häufigkeit des Begriffes „Apple“ weitaus höher als in der Kategorie Wirtschaft. Die Terme in der Menge der Nutzerartikel werden deshalb einerseits jeweils für *alle Sprachen* unterteilt sowie für *einzelne Kategorien*. So wird bspw. das Privatleben Madonnas (Kategorie Panorama) nicht mit ihrem Musikleben vermischt.

Auf einer allgemeinen Nachrichtenseite wie Spiegel Online wird der Begriff „Apple“ ebenfalls weniger häufig vorkommen, als auf einer Technikseite wie Engadget. Eine weitere Unterteilung, die die Nachrichtenseite bei dem jeweiligen Wort mit in Betracht zieht, wäre ebenfalls denkbar.

2. Annahme: Mindesthäufigkeit im Gesamtkontext vermeidet sinnlose Werte.

Daneben ist es auch wichtig, dass eine minimale Häufigkeit für einen Begriff im Gesamtkontext festgelegt wird. Dies ist wichtig, da sonst ein Begriff, der nur einmal im Gesamtkontext vorkommt, einen sehr großen Wert erzeugen. Um dies zu vermeiden, soll eine Mindesthäufigkeit hierfür gesetzt werden. Ein zu hoher Wert verwirft unnötig wichtige Begriffe, ein zu niedriger verzerrt die Ergebnisse. Aus diesem Grund wird ein Wert von fünf zu Beginn als sinnvoll erachtet.

3. Annahme: Mindesthäufigkeit im Nutzerkontext vermeidet falsche Empfehlungen.

Es wird außerdem eine Mindesthäufigkeit im Nutzerkontext festgelegt. Dadurch soll vermieden werden, dass ein Wort aufgrund des einmaligen Auftretens bereits Empfehlungen erzeugt. Der Schwellwert sollte hier geringer gesetzt werden, da ansonsten der Lernvorgang unnötig verlängert wird. Deshalb wird hierfür zunächst ein Wert von zwei als sinnvoll angesehen. D.h. ein Wort muss mindestens zweimal im Nutzerkontext vorkommen, ansonsten wird die gesamte Score auf Null gesetzt.

4. Annahme: Interessen verändern sich über die Zeit.

Daneben wird angenommen, dass die Interessengebiete eines Nutzers sich verändern können. Daher wird eine über die Zeit gewichtete Termfrequenz innerhalb der geteilten Nutzerartikelmengen berechnet. Dazu wird für jedes Vorkommen eines Wortes (bzw. eines Terms) das Datum mit dem Artikel gespeichert, in dem es vorkam. Anschließend wird die Termfrequenz mit folgender Formel gewichtet (weighted frequency):

$$wf_u(w_u) = \begin{cases} \sum_k^w \frac{1}{1 + P + \log(1 + t_k)} & \text{falls } |w| > 1 \\ 0 & \text{sonst} \end{cases}$$

- k ist das einzelne Auftreten des Terms w .
- P beschreibt die Zeitpräferenzrate, sie sollte auf einen Wert nahe Null gesetzt werden.⁹⁰
- t ist die Zeit seit dem Auftreten in Stunden.
- u beschreibt den Nutzer in dessen gelesener bzw. bewerteter Artikelmenge wf berechnet wird.

Die gesamte Formel zur Berechnung des Scores lautet:

$$UserTermRelevance(wf_{ut}, df_t, N) = \begin{cases} \log(1 + wf_{ut}) * \log\left(\frac{|D_c|}{df_{ct}}\right) & \text{falls } df_{ct} > 4 \\ \log(1 + wf_{ut}) * \log\left(\frac{|D|}{df_t}\right) & \text{falls } df_{ct} < 5 \text{ und } df_t > 4 \\ 0 & \text{sonst} \end{cases}$$

⁹⁰ Im vorliegenden Fall hat sich 0,2 bewährt.

- wf_{ut} ist die zuvor beschriebene gewichtete Frequenz für den Term t von Nutzer u .
- df_{ct} ist die Dokumentfrequenz innerhalb der Dokumente von Kategorie c für den Term t .
- df_t ist die Dokumentfrequenz innerhalb aller Dokumente für den Term t .
- $|D|$ ist die Gesamtanzahl der Artikel in diesem Korpus.

Es wird zunächst versucht, die Dokumentenhäufigkeit innerhalb der Kategoriedokumente zu finden, ist dies nicht möglich, da sie nicht das Mindestkriterium erfüllen, wird auf die Häufigkeit im Gesamtkontext zurückgegriffen, um die Werte nicht vollständig herauszufiltern. Dies kann jedoch zur Folge, dass diese Werte stark nach oben ausbrechen.

Die Anwendung des Logarithmus führt zu einer gewollten Dämpfung, damit einzelne Werte kein zu großes Gewicht erhalten. Eine Alternative wäre es, einen maximalen Wert für ein Token zu setzen.

3.6.2 Simple Scoring

Das erste Verfahren zur Bestimmung der Relevanz eines Artikels ist ein einfaches Scoring mittels der *UserTermRelevance* zuvor. Die Werte werden dabei addiert und anschließend normalisiert. Hat der Benutzer Präferenzen in Form von Keywords⁹¹ angegeben, erhalten diese einen Boost, falls sie im Artikel vorkommen.

Die Berechnung findet wie folgt statt:

$$Simple\ Scoring(a, k) = \frac{\sum_w UserTermRelevance(a_w) * k_w B}{\sqrt{\sum_w (UserTermRelevance(a_w) * k_w B)^2}}$$

- a ist der Artikel, der untersucht wird, a_w den Term w aus dem Artikel a .
- k sind die Keywords des Nutzers, die positiv oder negativ, jedoch nicht null sein können, als Default-Wert wird 1 angenommen.
- B ist der Boost für das Keyword.

⁹¹ Dieser Aspekt wird im Rahmen dieser Arbeit jedoch nicht näher betrachtet.

Das Simple Scoring beruht auf der wesentlichen Annahme, dass die Anzahl der textbasierten Features bei allen Artikeln gleich ist, bis auf kleine, vernachlässigbare Abweichungen. Dies ist dadurch begründet, dass diese textbasierten Features ausschließlich aus den RSS-Feeds stammen und dort Nachrichtenanbieter stets in etwa dieselbe Länge bei Titel + Abstract nutzen.

Es ist zu erwarten, dass dieses Verfahren sehr spezialisierte Empfehlungen liefert. Dies könnte dadurch begründet, dass hier ausschließlich die Wörter unabhängig voneinander betrachtet werden.

Problematisch könnten diese spezialisierten Empfehlungen dann werden, wenn der Nutzer sich für mehrere Gebiete interessiert, die miteinander zusammenhängen (Apple, iPhone, Steve Jobs). Dieses Verfahren würde diese Bereiche nicht isoliert betrachten, sondern einen Artikel empfehlen, der möglichst viele dieser Begriffe enthält (z.B. einen Kommentar von Steve Jobs, Apple's Gründer, zum iPhone).

3.6.3 Perceptron

Der klassische Perceptron ist ein sehr einfacher, aber dennoch mächtiger Algorithmus. Perceptrons sind auf lineare Entscheidungsgrenzen beschränkt, was bei hochdimensionalen Textklassifikationsproblemen bereits für eine gute Performanz ausreicht. Da das vorliegende Problem vor allem als ein Textklassifikationsproblem angesehen wird, soll er hier verwendet werden. Der Perceptron entspricht jedoch nicht dem aktuellen Stand der Technik, weshalb ihm vor allem Support Vector Machines (SVM) überlegen sind, die auch bereits zur Klassifizierung von Nachrichtenartikeln eingesetzt wurden.⁹² Hier sollen jedoch die Vorzüge der Perceptrons ausgenutzt werden: Dass sie bei großen Datenmengen vergleichsweise ressourceneffizient sind und online ausgeführt werden können.

Zunächst sollen hierfür die genutzten Features erläutert werden:

- Tokens aus Titel und Zusammenfassung
- Bigramme aus Titel und Zusammenfassung
- Artikel-Keywords
- Kategorie

⁹² Bomhardt (2004)

-
- Nachrichtenseite (z.B. www.spiegel.de)
 - RSS Feed
 - Facebook Likes und Shares (eingeteilt in Blöcken)
 - Erwähnungen auf Twitter (eingeteilt in Blöcke, jedoch feingranularer als bei Facebook, da die Anzahl weitaus geringer ist)
 - Themenhierarchie, extrahiert aus der Adresse
 - Artikel auf Startseite des Anbieters
 - Artikel an erster Position auf Anbieterseite
 - Artikelbild vorhanden
 - Länge der Zusammenfassung (eingeteilt in Blöcke)
 - Zeitdifferenz zwischen Erscheinen und Lesen des Artikels (in 10min Schritten)
 - Bereits von dieser Quelle Artikel gelesen
 - *Bevorzugte Quelle*⁹³
 - *Bevorzugte Kategorie*
 - *Beinhaltet Nutzer-Keywords*

Unter der Annahme, dass spezielle Themenbereiche sich besser klassifizieren lassen als die Gesamtheit aller Artikel, wird ein *dynamisches Splitting* eingeführt. Ziel ist es dabei, die Menge der Artikel so groß zu halten, dass sich für Lerner einfach bestimmen lässt, ob der Inhalt des Artikels interessant ist oder nicht.

Beim dynamischen Splitting darf die Menge der Trainingsartikel nicht zu klein werden, da ansonsten keine aussagekräftigen Ergebnisse geliefert werden können. Deshalb wurde eine minimale Größe von 15 Artikeln für einen Split festgelegt.

Die Reihenfolge des Splittings wurde wie folgt festgelegt:

1. Sprachen
2. Kategorien
3. Nachrichtenseiten

⁹³ Die kursiven Einträge sind zwar für die Nutzer zugänglich, wurden in der Evaluation jedoch vernachlässigt.

4. Schlagzeilenartikel auf Startseite

Ein Leser interessiert sich oftmals für Artikel, die nicht allein durch die Wörter seiner vorherigen Artikel definiert sind. Aus diesem Grund werden Trainingssätze *mit* und *ohne* Wörter gebildet, deren Güte in der Evaluation verglichen werden soll.

Das Training dieser multiplen Perceptrons soll auf den obigen Splits mit einer maximalen Trainingssatzgröße von 300 Beispielen in höchstens 20 Iterationen erfolgen, um die Auslastung möglichst gering zu halten. Die Initialwerte für die Worte soll dabei mit ihrem *UserTermRelevance* besetzt werden.

Beim Training wird eine Konfidenz berechnet, die beschreibt wie viele Trainingsbeispiele relativ gesehen über den Lerner in der letzten Iteration richtig klassifiziert wurden. Später werden die Splits so lange vergrößert, wie die Konfidenz wächst. Dies unterliegt der Annahme, dass je kleiner die Splits werden, sie sich umso besser lernen lassen und dadurch die Konfidenz steigt.

Das Ergebnis des Perceptrons ist ebenfalls ein Score aus dem Gewicht der einzelnen Features. Die Gewichtsvektoren der Perceptrons werden anschließend im Zwischenspeicher abgelegt und dann gepflegt. So müssen bei den zukünftigen Iterationen keine neuen Berechnungen durchgeführt werden. Hat der Nutzer zwischenzeitlich einen neuen Artikel gelesen, werden nur die Perceptrons erneuert, zu deren Splits dieser Artikel gehört. Die Auslastung pro Nutzer kann so erheblich reduziert werden.

3.6.4 kNN

Unter der Annahme, dass sich ein Leser immer für dieselben Dinge interessiert, bietet sich die Nutzung von kNN auf den wortbasierten Features an. In der Regel ist die Anwendung des kNN sehr ressourcenintensiv, weshalb er für diesen Fall etwas abgeändert wurde. Dabei soll eine Architektur ausgenutzt werden, bei der zu jedem Auftreten eines Tokens die Artikel Id gespeichert wird. So lässt sich sehr schnell für jedes Token sofort eine Liste der gelesenen Artikel zurückgeben, in der das Token aufgetreten ist. Die Berechnung des kNN ist damit sehr effizient und beansprucht nur wenige Ressourcen. Die auftretenden wortbasierten Features im untersuchten Dokument werden dabei mit der *UserTermRelevance* gewichtet. Die Ähnlichkeit zu einem anderen Dokument soll sich aus der *UserTermRelevance* gewichteten Kosinus-Ähnlichkeit sowie einem fixen Anteil für andere Features ergeben, der bei Evaluation variieren soll. In verschiedenen Szenarien sollen die Features des Perceptrons in

unterschiedlichen Kombinationen überprüft werden. Die Anzahl Nachbarn soll dabei auf maximal fünf beschränkt werden.

Ausgehend von der Annahme, dass man im vorliegenden Fall nur daran interessiert ist, die interessanten Dokumente weiter oben anzuzeigen, soll eine einfache Summierung der vorherigen Kosinus-Ähnlichkeiten die finale Reihenfolge bestimmen.

Da der kNN voraussichtlich Empfehlungen zu Artikeln liefert, die den gelesenen Artikeln am ähnlichsten sind, soll darauf geachtet werden, dass identische Meldung nicht nochmals empfohlen werden. Daher sollte der gelesene Artikel ein Mindestalter überschreiten, bevor er für einen Ähnlichkeitsvergleich herangezogen wird.⁹⁴ Begründet ist dies durch die Annahme, dass eine Meldung zeitnah in einem Artikel veröffentlicht wird und es sich daher nach dem Mindestalter nicht um dieselbe Meldung handeln kann.

3.6.5 Duplikateerkennung

Eine der Nutzeranforderungen in Kapitel 3.1 war, dass der Nutzer Abwechslung in seiner Nachrichtenliste erhält. In diesem Abschnitt soll beschrieben werden, wie man thematische Duplikate erkennt.

Zunächst einmal entstehen thematische Duplikate, wenn mehrere Nachrichtenanbieter zu einer Meldung einen Artikel veröffentlichen, wie in folgendem Beispiel:⁹⁵

Titel A: „Bundesregierung will mit Snowden sprechen“

Abstract A: “Edward Snowden könnte bereit sein, vor einem Untersuchungsausschuss des Bundestages auszusagen. Innenminister Friedrich will ein Gespräch mit der Bundesregierung ermöglichen. Moskau darf Snowden aber wohl nicht verlassen.”

Titel B: “Bundesregierung will mit Snowden sprechen“

Abstract B: “Das Treffen des Grünen-Politikers Ströbele mit Snowden zeigt Wirkung: Nach Wochen des Mauerns deutet Innenminister Friedrich an, dass die Bundesregierung mit dem US-Informanten sprechen könnte. Eine Deutschland-Reise des Ex-NSA-Mitarbeiters ist aber unwahrscheinlich.”

⁹⁴ Hierfür hat sich im weiteren Verlauf ein Mindestalter von drei Tagen bewährt.

⁹⁵ Diese Beispiele stammen von FAZ und SPIEGEL Online am 31.10.2013.

Thematisch beschreiben diese Artikel dieselbe Meldung, stammen jedoch von unterschiedlichen Nachrichtenseiten. Ihre Titel sind zwar identisch, ihre Abstracts unterscheiden sich jedoch signifikant, wodurch ein einfacher Vergleich über alle Worte nicht das gewünschte Ergebnis liefert. Oftmals unterscheiden sich sogar die Titel in ihrer Wortwahl, sodass die Duplikate noch schwerer zu erkennen sind. Die Titel weisen jedoch die kleinste Varianz auf, weshalb sie zunächst bei der Duplikateerkennung ein höheres Gewicht erhalten sollen. Außerdem fällt auf, dass bestimmte Worte wie „Bundesregierung“ oder „Snowden“ im Gesamtkontext vermutlich weitaus weniger häufig vorkommen als die Worte „sprechen“ oder „mit“.

Konkret soll als Maß für die Ähnlichkeitsvergleiche die Kosinus-Ähnlichkeit dienen. Sie soll die Stoppwörter-bereinigten und gestemmtten Wortlisten mit den typischen TfIdf Gewichten erhalten, wobei Titel-Tokens höher bewertet werden als Abstract Tokens. Außerdem sollen neben den textuellen Features noch zu einem kleinen Teil die Nachrichtenquelle und die Kategorie miteinfließen, da nicht davon auszugehen ist, dass auf derselben Nachrichtenseite zwei Artikel zur selben Meldung veröffentlicht werden.⁹⁶

Dieser hat die Grundlage für die Erkennung von Duplikaten geliefert, der konkrete Umgang mit ihnen soll im Kapitel zum Ranking genauer erläutert werden.

3.6.6 Zusammenfassung

Es wurden in den vorhergehenden Abschnitten die drei Algorithmen vorgestellt, die der Klassifikation von Nachrichtenartikeln dienen. Sie sollen in der Evaluation einerseits unabhängig voneinander, andererseits in Kombination getestet werden. Dabei steht am Ende der Berechnung immer ein Score für das *persönliche Interesse* an einem speziellen Artikel.

Die vorgestellten Algorithmen hängen miteinander zusammen, jedoch unterscheiden sie sich signifikant in ihrer Berechnung, weshalb sie unterschiedlich angewandt werden sollten. Es ist zu erwarten, dass der Perceptron bei einer großen Trainingsmenge mit vielen positiven und negativen Signalen aussagekräftige Ergebnisse liefert und dabei den anderen klar überlegen ist. Wohingegen das Simple Scoring voraussichtlich bei wenigen Signalen bereits akzeptable schwach personalisierte Empfehlungen aussprechen kann, da er nur die Gewichte der einzelnen Worte aufaddiert. Der kNN hingegen verfolgt einen anderen Ansatz, kann aufgrund

⁹⁶ Im weiteren Verlauf hat sich hier ein Verhältnis von 90% Wortfeatures und 10% andere Features bewährt.

der Restriktion des Mindestalters aber nicht von Anfang genutzt werden. Dies soll in der Evaluierung überprüft werden.

Diese Klassifikation erfordert in der Regel einen großen Ressourcenbedarf. Je mehr Nutzer und je mehr Artikel verwaltet werden müssen, desto mehr Ressourcen sind erforderlich. Ein präzises Online Empfehlungssystem ist so nicht möglich und sollte ins Backend ausgelagert werden. Dadurch soll es möglich werden, die Empfehlungen iterativ für jeden Nutzer zu erneuern.

Dies hat den Vorteil, dass in einer Iteration eine große Menge von Nutzern effizient abgearbeitet werden kann (Batch-Betrieb). Dieser Prozess lässt sich leicht auf mehrere Server verteilen, sodass die Skalierbarkeit gewahrt ist. Eine höhere Anzahl an Nutzern führt dazu, dass die Dauer jeder Iteration sich erhöht. Somit würde bei einem rapiden Anstieg der Nutzerzahlen nicht das gesamte System abstürzen, sondern sich nur die Iterationsdauer erhöhen. Ein Backend-Server sollte bei einer effizienten Implementierung mindestens 50.000 aktive Nutzer bearbeiten können und eine Iterationsdauer von weniger als 30 Minuten haben.

Der Nachteil eines reinen Offline-Recommendern ist hingegen, dass einerseits Trends und Adhoc Meldungen unzureichend abgedeckt werden und andererseits die Anpassung an den Nutzer nicht sofort stattfinden kann, da dies im schlimmsten Fall eine gesamte Iteration dauert.

Ein Hybrid-System, das im Backend mit einem *Offline Recommender* präzise Empfehlungen und Modelle berechnet, während im Frontend ein schneller *Online Recommender* arbeitet, könnte den optimalen Weg darstellen. Hierfür bieten sich insbesondere etwaige Heuristiken wie die von Prismatic an, in der beispielsweise nach fünf negativen Signalen einer Quelle, diese gänzlich ausgeblendet wird. Der Nutzer erhält dadurch *sofortige Rückmeldung* über seine Aktionen und gewinnt Vertrauen zu dem System.

Im Rahmen dieser Arbeit wird ausschließlich der *Offline Recommender* beschrieben, jedoch werden konkrete Vorschläge und Ansätze zum *Online Recommender* im Ausblick geliefert.

3.7 Ranking

Im vorherigen Abschnitt wurden Artikel für den Nutzer als relevant oder nicht-relevant klassifiziert. Dieser Abschnitt steht unter der Annahme, der Benutzer interessiert sich für bestimmte Themen mehr und für andere weniger, weshalb die *Reihenfolge* der relevanten Artikel so angeordnet sein sollte, dass sie genau diese Vorlieben widerspiegeln.

Wenn man einen typischen Nachrichtennutzer anschaut, dann stellt man fest, dass er ein großes Interesse an den Artikeln auf der Startseite seines Nachrichtenanbieters hat und außerdem seine Nachrichten heute auch aus den sozialen Netzwerken konsumiert. Diese Aspekte sollen in das Ranking miteinfließen und werden im Folgenden genauer betrachtet.

3.7.1 Auswertung der Daten aus sozialen Netzwerken

Das Internet bringt einerseits eine ungeahnte Informationsvielfalt mit, andererseits hat es auch dazu geführt, dass mitunter Beiträge veröffentlicht werden, die falsche Informationen beinhalten oder schlicht schlecht geschrieben wurden. Oftmals werden Artikel mit demselben Inhalt von mehreren Quellen veröffentlicht, aus denen Leser den für sie geeigneten wählen müssen.

Indem dem Nutzer herausragende Artikel angezeigt werden, soll ihm diese Wahl erleichtert werden. Dazu soll der Abdruck jedes Artikels in den sozialen Netzwerken (Facebook, Twitter) überprüft werden. Wird beispielsweise festgestellt, dass ein Artikel stark überdurchschnittlich viele Likes bzw. Shares erhalten hat, deutet dies in jedem Fall auf seine Beliebtheit hin und ist ein starkes Indiz für seine Qualität.

Diese Informationen sollen deshalb zu den zuvor aggregierten Artikeln automatisiert gesammelt werden, als eindeutige Identifikation dient die URL des Artikels. Bei Facebook lässt sich direkt der Zahl der Likes und Shares im Batch-Betrieb abfragen. Die Rückgabe ist dabei insbesondere bei kleineren Seiten oftmals 0, kann bei den großen allgemeinen Nachrichtenseiten auch bis 100.000 gehen. Bei Twitter können die Retweets nur indirekt über eine Suchquery mit dem Artikellink abgefragt werden. Dabei gibt Twitter die Tweets und deren Nutzerprofile zurück. Die Rückgabe ist auf maximal 100 Elemente beschränkt. Aufgrund der restriktiven Nutzungspolitik von Twitter und dem fehlenden Batch-Betrieb wird die Abfrage stark erschwert. Da die Retweets bei Weitem nicht die Größenordnung von Facebook erreichen, wird ein Element daher nach 100 Einträgen nicht weiter untersucht.

Die sozialen Signale der verschiedenen Nachrichtenseiten sind nicht direkt miteinander vergleichbar. Auf einige wesentliche Besonderheiten muss hierbei Rücksicht genommen werden:

- Die Anzahl der sozialen Signale hängt stark von der Bekanntheit der Nachrichtenseite ab, so erreicht bspw. Spiegel Online weitaus höhere Werte als die Frankfurter Rundschau.
- Die Anzahl hängt auch mit der Leserschaft bzw. der Ausrichtung der Nachrichtenseite zusammen, technische Seiten haben z.B. tendenziell weniger Facebook Likes als General Topic Seiten.
- Innerhalb der Nachrichtenseiten unterscheidet sich die Anzahl in Abhängigkeit mit den Kategorien. So erhält bspw. Politik bei Spiegel Online weitaus mehr Signale als der Bereich Wissenschaft bei selbiger Seite.
- Artikel, die auf der Startseite eines Nachrichtenanbieters angezeigt werden, erhalten weitaus mehr Signale. Dieser Effekt ist umso stärker, je weiter oben die Artikel gelistet werden.

Unter Berücksichtigung dieser Merkmale ergibt sich folgende Formel zur Bestimmung des sozialen Faktors eines Artikels:

$$socialScore(p, c, t) = count - \omega(m_{pct}t + (1 - t)m_{pc}) + \alpha m_p + \beta m_{pc} + \gamma m_{pt}t$$

- p beschreibt die Nachrichtenseite.
- c ist die Kategorie des Artikels.
- t beschreibt, ob es sich um einen Startseitenartikel (*Topnews*) handelt und kann 0 oder 1 annehmen.
- m_{pct} ist der Mittelwert für Artikel der Nachrichtenseite p , der Kategorie c , für den Fall, dass es sich um einen Topnews-Artikel handelt. Die anderen Mittelwerte verhalten sich analog.
- ω , α , β , γ beschreiben Dämpfungsfaktoren zwischen 0 und 1, wobei ω tendenziell näher bei 1 sein sollte und α , β , γ hingegen eher bei 0, da sie ein fixes Premium darstellen.

Mit dieser Berechnung soll eine Vergleichbarkeit der verschiedenartigen Signale gewährleistet werden. Später sollen die Daten dann in das Content-Filtering (eingeteilt in Klassen) sowie das Ranking einfließen.

3.7.2 Auswertung journalistisch kuratierter Daten

Dieser Abschnitt bezieht sich auf allgemeine bzw. gesellschaftliche Relevanz. Sie beschreibt einerseits Ereignisse, über die der Leser in jedem Fall informiert sein möchte, auch wenn sie eigentlich nicht in sein persönliches Spektrum fallen.

Leser erwarten andererseits jedoch auch Abwechslung beim Lesen ihrer Nachrichten und möchten mit neuen Inhalten überrascht werden (Serendipität). Ansonsten kann es passieren, dass sie in eine Art Tunnel (Filter-Bubble-Effekt⁹⁷) fallen, in dem die gelesenen Artikel die Fokussierung auf ein Thema immer weiter verstärken, sodass nach einer gewissen Zeit nur noch Artikel zu einem Thema angezeigt werden. Mit der gesellschaftlichen Relevanz sollen diese Ziele erreicht werden und dem Leser breitgefächerte Inhalte angezeigt werden.

Um zu bestimmen, ob ein Artikel gesellschaftlich relevant ist, soll die Artikelreihenfolge auf den Startseiten der Nachrichtenanbieter genutzt werden. Diese Reihenfolge wird von Journalisten bzw. Redakteuren individuell gesetzt. Um eine objektive Berichterstattung zu ermöglichen, wird hierfür von ihnen ein sog. Nachrichtenwert bestimmt. Dieser umfasst verschiedene unabhängige Elemente, wie Konflikte, Prominenz, Überraschung etc.⁹⁸

Außerdem ist davon auszugehen, dass die Reihenfolge auf der Anbieterstartseite noch durch die Klickraten der Artikel sowie der Entwicklung in den sozialen Netzwerken beeinflusst wird. Bei kleineren Nachrichtenseiten sind oftmals keine Abdrücke in den sozialen Netzwerken vorhanden, weshalb die Beeinflussung durch die Klickraten hier ein positiver Effekt erzielt wird. Nichtsdestotrotz sollte dieser Zusammenhang mit der vorher beschriebenen Popularität nicht vernachlässigt werden.

Die Daten zur momentanen Reihenfolge sollen stets aktuell sein, weshalb sie in regelmäßigen Abständen aktualisiert werden sollen. Zur effizienteren Bearbeitung sollen Nachrichtenseiten mit einem größeren Veröffentlichungsvolumen häufiger besucht werden.

Die hier beschriebene Wichtigkeit eines Artikels soll nicht nur die punktuelle Auflistung auf der Startseite repräsentieren, sondern auch, *wie lange* ein Artikel dort gezeigt wird, denn je

⁹⁷ Umgangssprachlich auch Tunneleffekt genannt.

⁹⁸ Mehr dazu von Kepplinger (1998).

länger ein Artikel in den oberen Rängen verweilt, desto wichtiger ist er. Hierfür sollte man sich zunächst den Trendverlauf eines Artikels auf der Startseite genauer anschauen. Dieser wird, wie zuvor beschrieben, in der Regel von einem Redakteur gesetzt und bewegt sich in den meisten Fällen anschließend mit fortschreitender Zeit abwärts (a). In einigen Fällen steigt er zuvor noch auf, bedingt bspw. durch starke Klickraten (b). Um diesen Sachverhalt abzubilden, müssen aktueller Rang, Trend und zeitabhängiger Rang in eine Formel miteinbezogen werden:

$$startpageScore_i(a) = \frac{\omega}{rank_i(a)} + \delta(rank_i(a) - rank_{i-1}(a)) + \theta \sqrt{\frac{i^2}{\sum_{k=0}^i rank_k(a)}}$$

- i beschreibt die i -te Extraktion der Startseitenartikel.
- $rank_i(a)$ beschreibt die Position von Artikel a in der Iteration i .
- ω stellt den Anteil des punktuellen Rangs dar.
- δ gibt den Anteil des Trends wider. In dieser Form werden Bewegungen in beide Richtungen gleich bewertet.
- θ bestimmt die Wirkung des zeitabhängigen Rangs, bei dem die Anzahl der Iterationen mit dem Durchschnitt dividiert wird.

Damit kann man den Artikel in seiner gesellschaftlichen Relevanz gewichten und später in einem Clustering-Verfahren, die Themen der Nachrichtenanbieter untereinander verknüpfen. Das Ergebnis ist eine Liste mit Clustern zu Themen über die die meisten Nachrichtenanbieter berichten.

3.7.3 Andere Ranking-Faktoren

Bisher wurden in diesem Kapitel der popularitätsbasierten Faktor der sozialen Netzwerke sowie gesellschaftlich orientierte Faktor aus den Informationen seitens der Nachrichtenanbieter erläutert. Im Kapitel zuvor wurde über die vorgestellten Klassifizierungsalgorithmen ein Score für das *persönliche Interesse* eines individuellen Nutzers für einen Artikel berechnet.

Diese Informationen werden für das Ranking jedoch nicht als ausreichend erachtet, da die Nutzeranforderungen aus Kapitel 3.1 damit noch nicht erfüllt sind. In diesem Abschnitt sollen deshalb noch Regeln für die Bereiche *Aktualität* und *Abwechslung* vorgeschlagen werden.

Aktualität eines Artikels

Benutzer sind daran interessiert, immer auf dem neuesten Stand bei Nachrichten zu sein. Allerdings trifft dies bei Sportmeldungen mehr zu als auf Meldungen aus der Wissenschaft. Außerdem variieren Benutzer auch hier bei ihren Präferenzen. Es ist daher notwendig diesen Sachverhalt abzubilden und den richtigen Kompromiss aus Relevanz und Aktualität zu finden. Dies kann gelöst werden über eine Funktion, die fix mit steigendem Alter (in Minuten) des Artikels, dessen Score “verderben” langsam lässt. Die Geschwindigkeit des Verderbens wird durch β bestimmt. Artikel, die bereits drei Tage alt sind, sollten nicht dafür einen wesentlich schlechteren Score erhalten, als Artikel, die nur zwei Tage alt sind. Aus diesem Grund soll die Alterungsfunktion bis zu einem Maximalwert m berechnet werden. Außerdem sollten Artikel, die in den letzten n Minuten veröffentlicht wurden (bspw. Eilmeldungen), ein Premium τ erhalten, da der Alterungseffekt hier sich nicht adäquat niederschlägt.

$$timeAdjustedScore(score, article) = \begin{cases} \frac{score}{1+\beta*\log(1+m)} & falls\ age > m \\ \frac{score}{1+\beta*\log(1+article_{age})} + \tau & falls\ age < n \\ \frac{score}{1+\beta*\log(1+article_{age})} & sonst \end{cases}$$

Eine individuelle Koppelung an die Nachrichtenseite oder die Kategorie wäre ebenfalls denkbar und würde den einleitend erwähnten Sachverhalt mit der Sportmeldung abdecken.

Abwechslung in finaler Liste

Nutzer möchten nicht Themen immer nur zu einem Thema erhalten, sondern möglichst breit über ihre Interessen informiert werden. Deshalb soll in einem letzten Schritt eine finale Umsortierung stattfinden. Hierzu soll im Folgenden ein Algorithmus vorgestellt werden.

Der Algorithmus fügt der Reihe nach die einzelnen Elemente einer finalen Liste hinzu. Dabei wird für jeden Artikel einerseits die *maximale* und *andererseits* eine gewichtete Kosinus-Ähnlichkeit zu den vorherigen Artikeln bestimmt.

Mit der maximalen Ähnlichkeit lassen sich Artikel finden, die dieselbe Meldung beschreiben, jedoch bspw. von einer anderen Quelle veröffentlicht wurde. Falls die maximale Ähnlichkeit größer ist als ein festgelegter Grenzwert φ , dann soll der Artikel fortan ignoriert werden.⁹⁹

Liegt die maximale Ähnlichkeit unter dem Grenzwert wird nun die gewichtete Ähnlichkeit überprüft. Sie beschreibt die Ähnlichkeit von diesem Artikel zu den restlichen Artikeln in der Liste, wobei deren Ähnlichkeiten in Abhängigkeit ihrer Distanz, also ihrem Rang, gewichtet werden. Damit soll verhindert werden, dass zu viele Artikel zu einem Thema in der Liste auftauchen, ohne sie direkt abzulehnen.¹⁰⁰

Falls die gewichtete Ähnlichkeit eines Artikels über einem weiteren Grenzwert ρ liegt, wird der Artikel auf einen Stack geschrieben und dort aufbewahrt, andernfalls wird er direkt hinzugefügt. Da die gewichtete Ähnlichkeit maßgeblich von der Zusammensetzung der finalen Liste abhängt, wird nach jeder Iteration geprüft, ob die Elemente aus dem Stack nun qualifiziert sind, in die Liste aufgenommen zu werden. Dabei wird außerdem für jeden auf dem Stack für jede Iteration des Wartens die gewichtete Ähnlichkeit schrittweise mit dem Faktor ω reduziert. Dies liegt der Annahme zugrunde, dass der Nutzer zwar keine Häufung von thematisch gleichen Artikeln hinnimmt, jedoch nach einem gewissen Abstand wieder Artikel mit demselben Thema akzeptiert. Am Ende soll eine möglichst breitgefächerte und diversifizierte Liste mit Artikeln.

3.7.4 Kombination der Ergebnisse

In diesem Abschnitt soll beschrieben werden, die beschriebenen Ansätze miteinander kombiniert werden. Mit den folgenden Scores und Mechanismen soll ein finaler Rang für einen Artikel generiert werden:

- Persönliches Interesse (3.6.6)
- SocialScore (3.7.1)

⁹⁹ Hierbei hat sich im weiteren Verlauf eine Kosinus-Ähnlichkeit von 0,5 für φ als optimal erwiesen.

¹⁰⁰ Für die gewichtete Ähnlichkeit hat sich im weiteren Verlauf eine Kosinus-Ähnlichkeit von 0,1 bewährt.

-
- StartageScore (3.7.2)
 - TimeAdjustedScore (3.7.3)
 - Abwechslungsmechanismus (3.7.3)

Hierbei stellen die ersten drei Werte eine voneinander unabhängige Bewertung des Artikels dar. Während die beiden letzteren den vorhandenen Wert nur anpassen.

Die Grundidee ist es, dem Nutzer zunächst die Artikel anzuzeigen, die auf den Startseiten seiner präferierten Anbieter erscheinen. Falls der Artikel sich auf der Startseite des Anbieters befindet, wird deshalb der kombinierte Score um ein Premium erhöht. Ist der Artikel sogar an erster Stelle beim Anbieter, dann wird er nochmals erhöht. Das beruht auf der Annahme, dass Schlagzeilen für den Nutzer interessant sind, auch wenn dieser Artikel eigentlich nicht in sein Interessenspektrum fällt.

Außerdem soll der Rank eines Artikels aus dem Score zum persönlichen Interesse bestehen sowie aus dem *SocialScore*. Aus diesem Grund soll der *SocialScore* den bisherigen Score nochmals um einen gegebenen Anteil erhöhen, der in Abhängigkeit vom bisherigen Score berechnet wird. Zum Schluss soll dieser neue Score noch bezüglich des Artikelalters und der Themenähnlichkeit angepasst werden.

Alternativ wäre auch ein reines Ranking anhand des *SocialScores* oder des *StartpageScores* möglich, in dem vorgegebene Blöcke gemäß ihrer Beliebtheit in den sozialen Netzwerken umsortiert werden. Dies könnte jedoch zu verzerrten Ergebnissen führen.

4 Konkrete technische Umsetzung

Bei der Implementierung wurde die Architektur zunächst in vier Blöcke Crawler, Frontend, Model und Recommendation Engine (Backend) geteilt (siehe Abbildung 4-1: Gesamtarchitektur in der Implementierung). Physisch sind Frontend, Model und Crawler gemeinsam auf einem und das Backend auf einem separaten Server platziert. Dies ist möglich, da der Crawler gleichzeitig nur eine begrenzte Zahl an RSS-Feeds bearbeitet und dadurch keine kurzen Spitzen, sondern eine relativ gleichmäßige Last verursacht. Das Backend repliziert die Datenbank vom Frontend Server, wodurch auf der Hauptdatenbank lediglich Schreiboperationen beim Erneuern der Empfehlungen anfallen.

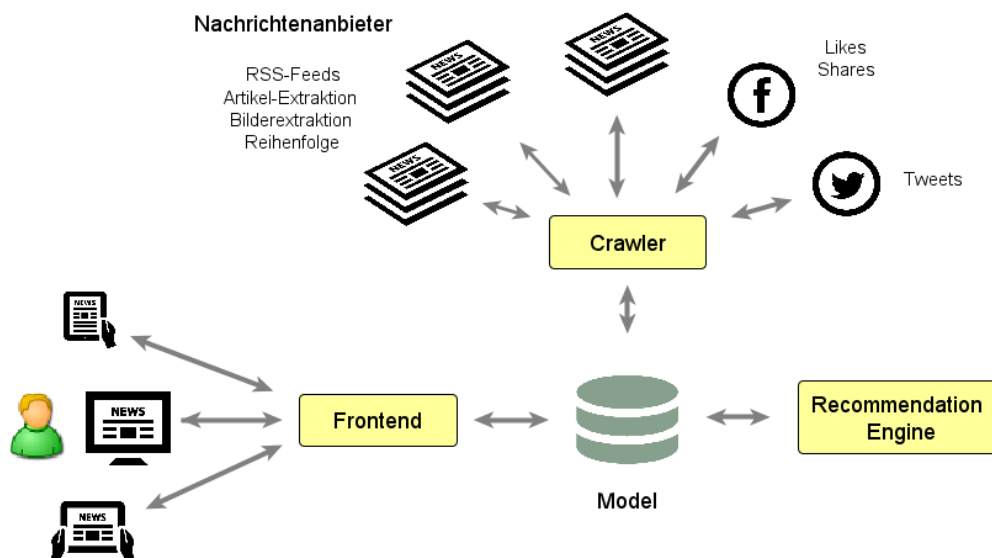


Abbildung 4-1: Gesamtarchitektur in der Implementierung

4.1 Verwendete Infrastruktur

In diesem Kapitel soll auf die unterliegende Infrastruktur eingegangen werden. Hierbei vor allem auf die verwendete Datenbank, Programmiersprache und die Frameworks bzw. Bibliotheken, die Verwendung finden. Dies soll dem Leser helfen, die Möglichkeiten und die Schwachstellen des Systems zu erkennen.

4.1.1 Programmiersprache

Die Programmierung wurde ausschließlich in **Java** vorgenommen. Einerseits, da der Autor im Studium und in persönlichen Projekten bereits häufiger mit Java gearbeitet hat. Andererseits

jedoch auch, weil zusätzlich zum Recommender System noch ein modernes Webprojekt erstellt werden soll und in Java einige sehr hilfreiche Bibliotheken und Frameworks für diesen Fall existieren. Zudem ist bei Java Threading sehr leicht möglich, wodurch sich Parallelisierung schnell implementieren lässt, was bei den maschinellen Lernalgorithmen ausgenutzt werden kann. Grundlegend kommen hier jedoch die Programmiersprachen Python, Java und C/C++ in Frage.

Python bietet viele fertige Pakete für linguistische Verarbeitung und vor allem auch viele gute für das maschinelle Lernen. Außerdem fördert das Design von Python die Lesbarkeit von Quellcode. Allerdings ist die Performanz hier ein großes Problem, weshalb es sich für den Produktiveinsatz nicht eignet.

Besser bei der Performanz ist aufgrund der Hardwarenähe vor allem **C/C++**, jedoch gibt es hier weitaus weniger Bibliotheken als bei Java und zudem sind diese oftmals nicht ausgereift und enthalten Fehler. Langfristig sollte ein Umstieg auf C/C++ von bestimmten Modulen jedoch durchaus in Betracht gezogen werden.

4.1.2 Datenbank

Für die Datenbank wurde Mysql gewählt, da auch diesbezüglich der Autor Erfahrungen mitbringt. Zudem ist Mysql Open Source, hat eine große Verbreitung und dementsprechend auch eine gute Community für Support und nicht zuletzt schnell auf allen gängigen Betriebssystemen einzurichten.

Langfristig sollte zusätzlich eine weitere Datenbank wie **MongoDB** herangezogen werden, die performanter mit großen Daten umgehen kann und besser skaliert als Mysql. Dies nützt vor allem bei der Auswertung mit maschinellem Lernen.

4.1.3 Frameworks, Tools und Bibliotheken

Die verwendeten Frameworks lassen sich in drei Kategorien einteilen: Allgemein, Content und Backend. Im Folgenden sollen diese genauer beleuchtet werden.

Allgemein

Hibernate, ein objekt-relationaler Mapper für Java, der die Mysql Datenbankbindung regelt.

Apache Maven, ein Build-Management-Tool für Java, mit dem man Programme standardisiert erstellen und verwalten kann.

Ehcache, ein allgemeiner Open Source Cache für Java.

Content

Apache Wicket, ein komponentenbasiertes Open Source Web-Framework für Java, das u.a. beim Discounter Walmart Verwendung findet.

Jakarta Feedparser, ein Open Source Parser für RSS/Atom, der u.a. heuristische Methoden zur Entdeckung von RSS Feeds bereitstellt und ist toleranter bei nicht-XML-konformen Dokumenten.

Backend

Apache Lucene, eine Programmbibliothek zur Volltextsuche für Java, notwendig für die Verarbeitung und Indizierung von Worten.

Zukünftig ist der Einsatz von Machine Learning Frameworks wie Mahout denkbar für bestimmte Anwendungsbereiche, bspw. für das Clustering. Das gesamte Recommender System über Mahout zu erstellen wird jedoch nicht als vorteilhaft erachtet, da so der Differenzierungsvorteil verloren geht.

4.2 Erfassung der Nachrichtenartikel (Crawler)

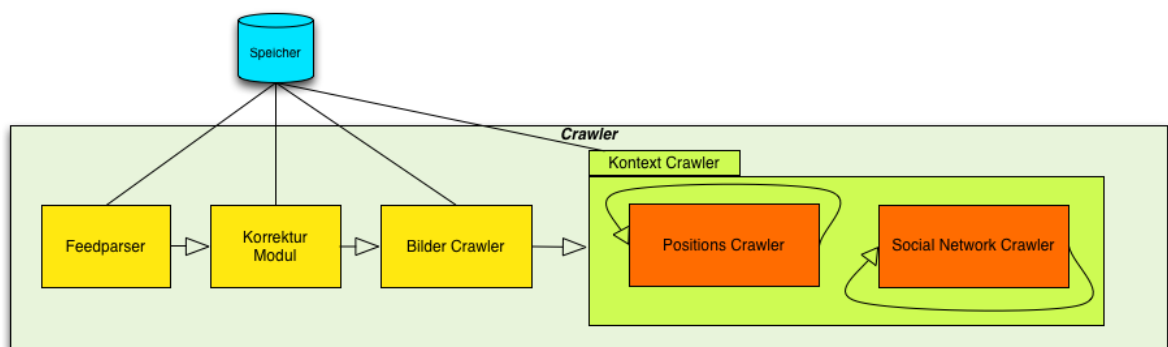


Abbildung 4-2: Crawler Architektur

Die Erfassung der Nachrichtenartikel ist wichtig für den weiteren Verlauf, um zu verstehen welche Informationen vorliegen. Sie geschieht iterativ für jeden RSS Feed im Abstand von wenigen Minuten, jeweils in mehreren Schritten (siehe Abbildung 4-2: Crawler Architektur).

So können Redundanzen vermieden und trotzdem die Daten immer aktuell gehalten werden. Im Folgenden soll der Ablauf kurz erläutert werden:

Feedparser

Der Jakarta Feedparser iteriert über die Inhalte im RSS-Feed und schreibt den Artikel ggf. in die Datenbank, ansonsten stoppt der Crawler hier. Jeder Feed wird dabei als unabhängiger Thread bearbeitet, sodass die Ausführung bei räumlich weit entfernten Quellen (bspw. Asien) nicht verzögert wird.

Korrektur Modul

Die Artikelinformationen sollten für die weitere Verarbeitung möglichst vollständig und richtig sein. Daher werden für jeden neuen Artikel in diesem Modul folgende Korrekturen vorgenommen:

- *Demaskierung der Links:* Nachrichtenseiten verwenden in ihren RSS-Feeds oftmals maskierte Links, um den Besucherstrom besser zuteilen zu können. Diese Links sind für die weitere Verarbeitung jedoch nicht nützlich. Daher werden diese Links demaskiert.
- *Ergänzung um Meta-Informationen:* Die ursprünglichen Informationen stammen nur aus den RSS-Feeds. Teilweise fehlen dort Informationen (z.B. keine Zusammenfassung vorhanden) oder sie sind inkorrekt (z.B. Titel mit Werbung der Verlage). Dies kann behoben werden durch das Auslesen der Meta-Informationen eines Artikels, hierbei erweist sich der Open-Graph von Facebook als besonders nützlich.
- *Kategorie Mapping:* Nachrichtenseiten bezeichnen ihre Rubriken unterschiedlich, beispielsweise meinen „Ausland“ und „International“ dasselbe. Dieses Problem wird hier gelöst.
- *Link Extraktion:* Bei manchen Artikeln existieren keine Informationen zur Kategorie, allerdings lässt sich dies im deutschsprachigen Raum oftmals durch ein scharfes Hinsehen auf den Artikellink beheben. Diese beinhalten häufig die hierarchische Einteilung des Artikels. Ein Artikel von Jan Fleischhauer (<http://www.spiegel.de/politik/deutschland/jan-fleischhauer.html>) lässt erkennen, dass dieser Artikel in der Kategorie Politik, speziell in der Unterkategorie Deutschland ist. Diese Zusatzinformationen werden auch in die Datenbank geschrieben.

-
- *Kategorien finden über kNN*: Falls die vorhergehenden Schritte zu keinem Ergebnis geführt haben, wird mithilfe von Lucene eine effiziente Suchanfrage über die Artikel der letzten Tage gesendet. Anschließend wird ein Majority Voting über die gefundenen Kategorien, gewichtet mit der Ähnlichkeit, durchgeführt.

Bilder Crawler

Ein ansprechendes Bild ist wichtig bei der Entscheidung, einen Artikel zu lesen. Die Bilder aus den RSS-Feeds sind heutzutage oftmals für Mobilgeräte ausgelegt und weisen daher eine schlechte Qualität auf. Handelt es sich um einen neuen Artikel, werden deshalb über einen separaten Crawler die Bilder zu diesem Artikel gesucht und abgespeichert. Die Suche nutzt die zuvor gesammelten Titel und Zusammenfassungen zum Artikel. Im XML-Baum des Artikels wird dann ihr gemeinsamer Vaterknoten gesucht und von dort aus der Baum nach außen traversiert. Das nächste Bild wird als Artikelbild abgespeichert, wenn eine Mindestgröße überschreitet und eine festgelegte Knotendistanz nicht überschreitet. Dies ist erforderlich um falsche Bilder bzw. Werbung zu umgehen. Dieser Mechanismus kann später auch genutzt werden, um den gesamten Artikelinhalt zu extrahieren.

Kontext Crawler

Im nächsten Schritt werden Kontext-Informationen zum Artikel gesucht. Dazu werden Separate Prozesse gestartet, die in regelmäßigen Abständen einerseits die Position des Artikels auf der Anbieterstartseite suchen (die Artikelreihenfolge dieser Seite wird zurückgegeben. Andererseits aber auch Twitter und Facebook nach dem „sozialen Abdruck“ dieses Artikels durchsuchen und die Anzahl der Retweets bzw. die Anzahl der Likes und Shares zum aktuellen Zeitpunkt persistent festhalten.

Ein Nachrichtenartikel hat demnach folgende Attribute:

- Titel
- Zusammenfassung
- Link
- Bild
- Quelle
 - RSS-Feed
 - Startseite

-
- Sprache
 - (Beschreibung der Quelle)
 - (Kategorie der Quelle)
 - Kategorie
 - Keywords
 - Veröffentlichungsdatum
 - Eingangsdatum
 - Anzahl der Retweets zu unterschiedlichen Zeitpunkten
 - Anzahl der Likes zu unterschiedlichen Zeitpunkten
 - Anzahl der Shares zu unterschiedlichen Zeitpunkten

Wie zuvor erwähnt, wird der komplette Inhalt des Artikels **nicht** extrahiert, da die Zusammenfassung in der Regel die wichtigsten Begriffe bereits enthält. Ein Lerner muss daher auf weniger Features lernen, was sich jedoch auf die Genauigkeit auswirken kann.

4.3 Vorbereitung der Inhaltsanalyse

Preloading

Anfragen an die Datenbank werden zum Flaschenhals bei der Ausführung der Lernalgorithmen. Deshalb ist es extrem wichtig, diese Anfragen so gering wie möglich zu halten.

Aus diesem Grund werden in einem ersten Schritt einmalig alle notwendigen Daten in den Arbeitsspeicher geladen, sodass jederzeit schneller Zugriff auf diese möglich ist (Over-Eager-Loading). Außerdem können Datenbankanfragen mit diesem Mechanismus gebündelt werden, wodurch in der Folge nur noch Schreiboperationen bei den finalen Empfehlungen anfallen. Anschließend werden bei jeder weiteren Iteration in mit einer Update-Strategie nur noch neu hinzugekommene bzw. geänderte Daten geladen.

Der Trade-Off des Preloadings ist es, dass hauptsächlich bedingt durch die Strings ein relativ hoher Bedarf im Arbeitsspeicher entsteht. Daher ist es wichtig, nur die Daten zu laden, die auch wirklich erforderlich sein könnten.

Um den Arbeitsspeicher jedoch zu schonen, werden alle Daten zentral verwaltet, sodass immer auf denselben Objekten gearbeitet wird und keine Redundanzen entstehen. Besonders wichtig hier ist, dass die Daten threadsafe abgefragt werden.

Das Ziel dieser Maßnahme ist es, eine möglichst konstant hohe Auslastung durch die Lernalgorithmen zu erzeugen und dabei die Wartezeiten für das Bereitstellen der Daten zu minimieren.

Indizierung der Artikel

In diesem Schritt werden die Nachrichtenartikel mit Lucene indiziert. Lucene übernimmt dabei auch die Übersetzung in Tokens, das Stemming der Worte sowie das Filtern der Stopwords. Die daraus resultierenden Directories sollen später genutzt werden, um die Tfidf der einzelnen Wörter zu berechnen. Aus diesem Grund sollen mehrere Directories angelegt werden, in denen Lucene sehr schnell nach Begriffen suchen kann. Die Unterteilung soll wie folgt geschehen:

- Artikel unterteilt nach Sprachen.
 - Artikel unterteilt nach Kategorien in der jeweiligen Sprache.

Da Lucene jeden Artikel als einzelnes Dokument speichert, ist es im Nachhinein leicht möglich, auf einen einzigen indizierten Artikel zuzugreifen.

Zu obigen Artikel werden in diesem Schritt folgende Felder indiziert:

- Titel
- Zusammenfassung
- NGramme (n = 2)
- Keywords
- Kombination aus Titel, Zusammenfassung und Bigrammen

4.4 Nutzeroberfläche

Im Folgenden soll kurz die Nutzeroberfläche für den Experten der Evaluation beschrieben werden. Diese wurde gemäß der Richtlinien aus Nutzersicht in Kapitel 3.1 umgesetzt.

Um nicht vom Nachrichtenvergnügen abzulenken, wurde ein minimalistischer Stil eingehalten, wodurch die Übersicht für den Experten wie folgt aussah (Abbildung 4-3):

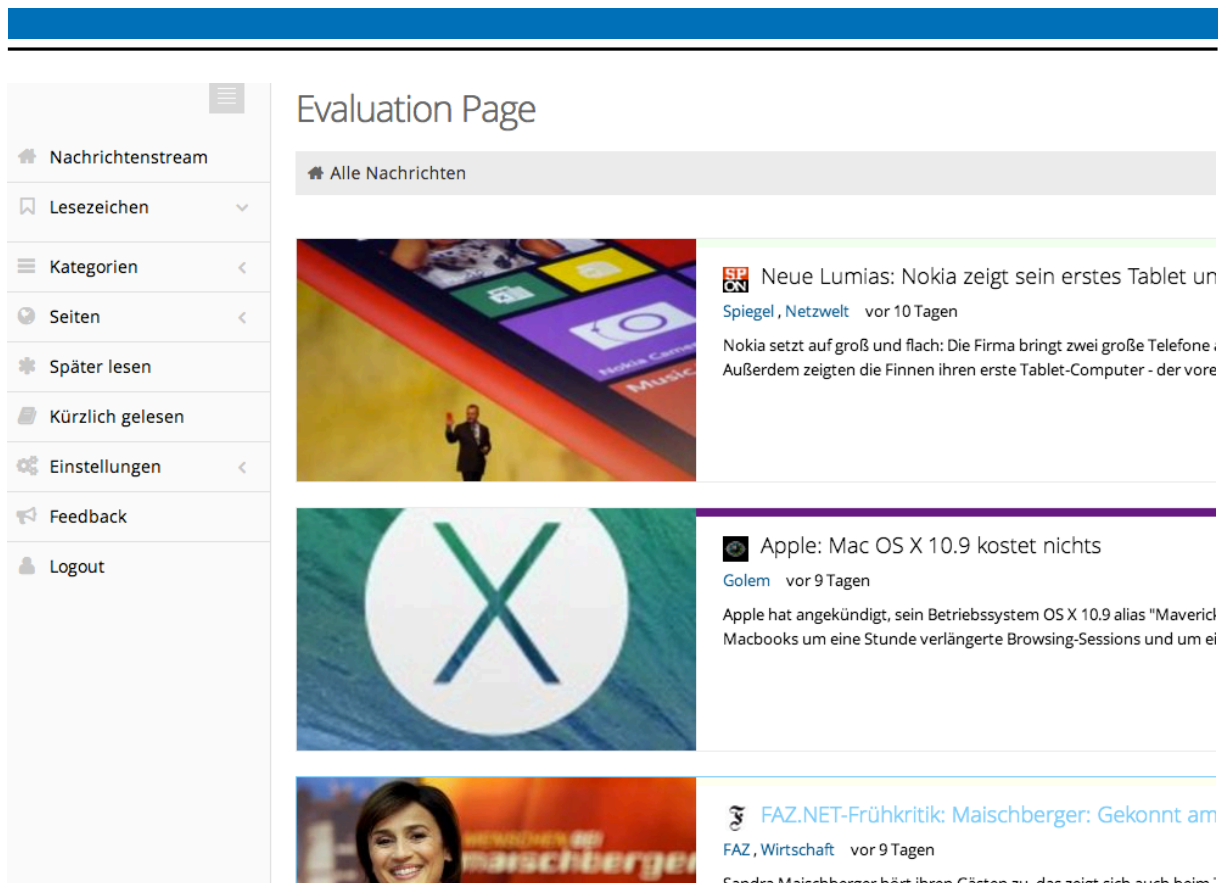


Abbildung 4-3: Portal für den Evaluierungsexperten.

Die Darstellung wurde in Listenform gewählt, damit keine Verzerrungen bei den vorgestellten impliziten Strategien auftreten. Alternativ dazu wurde auch eine Ansicht in Kachelform entwickelt, wobei die Teilnehmer der anschließenden Feldstudie, zwischen diesen Designs wählen konnten.

Klickte der Experte auf einen Artikel, so wurden ihm weitere Informationen und Bewertungsmöglichkeiten angezeigt (Abbildung 4-4):

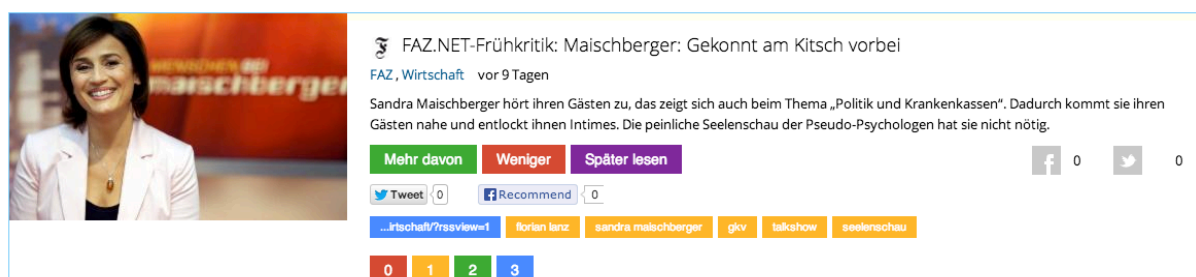


Abbildung 4-4: Artikeldetailansicht für Evaluierungsexperten.

Dort hatte er auch die Möglichkeit, einen Artikel positiv und negativ zu bewerten, ihn zu seiner Leseliste hinzuzufügen sowie Details über den sozialen Abdruck des Artikels. Ganz unten befand sich die Bewertungsskala für den Experten in den Farben (rot=0, gelb=1, grün=2 und blau=3). Diese Bewertung wird im 0. Kapitel noch genauer erläutert.

5 Evaluation der technischen Umsetzung

Die Evaluation bei Nachrichten-Recommendern stellt eine besondere Schwierigkeit dar, die nicht mit Suchmaschinen zu vergleichen ist. Zunächst sollen deshalb die Probleme kurz aufgezeigt werden:

1. Zu jeder Meldung existieren mehrere Artikel, von verschiedenen Quellen. Der Nutzer präferiert oftmals nicht nur eine dieser Quellen.
2. Ein Nutzer interessiert sich nicht ausschließlich für bestimmte Themenbereiche, sondern möchte häufig auch mit Neuem überrascht werden.
3. Ein Nutzer hat keine statischen Interessen, sondern verändert diese über die Zeit.
4. In vielen Fällen reicht es dem Nutzer außerdem, die Überschrift und die dazugehörige Zusammenfassung zu lesen. Eine Messung der Klicks kann dadurch ein verzerrtes Bild liefern.

Dadurch ergibt sich keine singuläre, perfekte Anordnung der Artikel, weshalb bestimmte Evaluationsmethoden nicht anwendbar sind. Vielmehr existiert eine extrem große Zahl an Permutationen, die sehr gut zum Nutzer passen.

Es ist im Folgenden auch angenommen, dass der Nutzer nicht an allen für ihn relevanten Artikeln interessiert ist, da er täglich nur eine begrenzte Zeit mit Medien verbringt und nicht in der Lage ist, jeden Artikel zu lesen. In der Folge sollen deshalb Ergebnislisten mit einer höheren Precision gegenüber denen mit höherem Recall präferiert werden. Es wird sogar angenommen, dass die F1 Metrik nicht aussagekräftig für den vorliegenden Fall ist. Im Grundlagenteil (siehe Kapitel 2.6.4) wurde bereits erwähnt, dass dies auch in vielen anderen Projekten in diesem Bereich so gehandhabt wurde.

5.1 Testdaten

Zunächst wurde ein Korpus von Dokumenten bzw. Artikeln von menschlichen Experten indiziert. Diese Methode ist jedoch mit einem größeren Aufwand verbunden, weshalb der

Einfachheit halber die Relevanz typischerweise binär ausgedrückt wird. Wegen der bereits erwähnten Ambiguitäten mit Nachrichtenartikeln wurde eine Bewertung jedoch in folgendem Schema durchgeführt:

- 3 – Lesenswert
- 2 – Sehr Interessant, aber Überschrift und Zusammenfassung eventuell ausreichend
- 1 – Marginal interessant
- 0 – Nicht interessant

Mit diesem Bewertungsschema wurden 840 Artikel (24 Chunks mit jeweils 35 Artikeln) aus acht verschiedenen Nachrichtenseiten in einem Zeitraum von *sieben Tagen* zufällig ausgewählt (siehe Tabelle 3: Herkunft und Häufigkeit der Evaluierungsartikel).

<i>Nachrichtenportal</i>	<i>Webseite</i>	<i>Anteil</i>	<i>Beschreibung</i>
Spiegel Online	www.spiegel.de	37%	General Topic News (GTN)
FAZ	www.faz.net	16%	GTN, Wirtschaft
Zeit	www.zeit.de	16%	GTN
Handelsblatt	www.handelsblatt.com	8%	Wirtschaft
Süddeutsche	www.sueddeutsche.de	8%	GTN
Golem	www.golem.de	6%	IT
Bild	www.bild.de	5%	GTN, Boulevard
Sport1	www.sport1.de	3%	Sport, Fußball

Tabelle 3: Herkunft und Häufigkeit der Evaluierungsartikel

Die acht Quellen hingegen sind nicht zufällig ausgewählt worden, sondern entsprechen in etwa den persönlichen Interessen und dem Leseverhalten des Experten, der sie bewerten sollte. Dadurch soll die Klassifizierungsaufgabe erschwert werden, da die Quellen zu einem großen Teil im Interessengebiet des Experten liegen und so eine tiefere Personalisierung

durchgeführt werden muss. Von vornherein uninteressante Quellen und Kategorien können so nicht einfach von den Algorithmen aussortiert werden, was die Ergebnisse stark positiv verzerren würde. Nichtsdestotrotz wurden zu einem kleinen Anteil Quellen hinzugezogen, die nicht das Interessengebiet des Experten adressieren (Bild und Sport1). Damit kann sichergestellt werden, dass der Experte auch bei der Evaluierung zufällig auf interessante Nachrichten von nicht präferierten Quellen stoßen könnte, die durch die Klassifizierung nicht abgelehnt werden sollten.

Es wurde Wert darauf gelegt, dass der Experte eine reale Umgebung für das Labeling vorfindet. Er sollte dieselben Möglichkeiten der Bewertung haben, wie der finale Nutzer des PNA und die Artikel auch in derselben Weise präsentiert bekommen wie ein typischer Nutzer. Insbesondere Letzteres ist von außerordentlicher Bedeutung, da eine erfolgreiche Evaluierung von einer neutralen und kontinuierlichen Bewertung der Sachverhalte abhängt. Es wird angenommen, dass eine unattraktive Präsentation der Artikel zu schlechteren bzw. verfälschten Ergebnissen führt. Aus diesem Grund soll die Evaluierungssoftware zumindest so gut gestaltet werden, dass dieser Effekt ausbleibt. Ebenfalls aus diesem Grund wurde die Anzahl der angezeigten Artikel pro Seite auf 35 gesetzt. Dies ist dadurch begründet, dass viele große Nachrichtenseiten auf ihrer Startseite zwischen 30 und 40 Artikel anzeigen, was mit der nachlassenden Aufnahmefähigkeit mit steigender Anzahl zusammenhängen könnte.

Bewertung	Absolut	Relativ	Kumuliert
3	141	16,79%	16,79%
2	168	20%	36,79%
1	245	29,15%	65,94%
0	286	34,05%	100%

Abbildung 5-1: Überblick zur Expertenbewertung.

Die Verteilung der Expertenbewertungen nach obigem Schema sind in Abbildung 5-1 beschrieben. Der Experte wurde angewiesen, die präsentierte Artikelliste mehrfach durchzugehen und bei jedem Durchgang nur eine Art von Bewertung zu verteilen. Also entweder 0, 1, 2 oder 3, wodurch er sein Ergebnis nicht noch im Nachhinein verändern

konnte, da er mit jedem Durchgang mehr Informationen aufnimmt, die auch seine Bewertungen beeinflussen. Bei der ersten Durchsicht sollte er zunächst alle Artikel anklicken, die er sofort lesen würde (Bewertung 3), weil sie so interessant auf ihn wirkten. Er durfte auch teilweise explizites positives Feedback geben, jedoch nicht übermäßig. Im zweiten Durchgang sollte er alle Artikel als negativ bewerten (Bewertung 0), die ihn überhaupt nicht interessieren. Auch hierbei sollte er nicht zu exzessiv mit dem expliziten Feedback umgehen, sondern nur, wenn er es für angebracht hält. Danach sollte er unter den übrigen diejenigen bewerten, die er interessant findet (Bewertung 2), aber zuvor nicht angeklickt hat und dabei hin und wieder ebenfalls explizites Feedback geben. Die verbleibenden Artikel sollten im letzten Schritt mit der Bewertung 1 versehen werden, da er sie zwar nicht uninteressant, aber auch nicht stark interessant. Die Vorgabe zur spärlichen Verwendung des expliziten Feedbacks ist dadurch begründet, dass nach eigener Beobachtung, ein typischer Nutzer davon ebenfalls seltener Gebrauch macht.

Basierend auf den zuvor festgelegten Strategien (siehe 3.5) wurde dabei folgendes Feedback wurde vom Experten generiert (Abbildung 5-2):

Feedback Art	Absolute Häufigkeit
Explizit positiv („Mehr“)	45
Explizit negativ („Weniger“)	69
Artikelklick (implizit)	139
Übersprungene Artikel (implizit)	391
Top Artikel nicht geklickt (implizit)	12
Vorgänger nicht geklickt (implizit)	143
Nachfolger nicht geklickt (implizit)	145

Abbildung 5-2: Feedback Generierung durch Experten in absoluten Zahlen.

5.2 Methodik der Evaluierung

Um den Lernfortschritt über die Zeit zu messen, wird eine Prequential Evaluation (2.5.5) durchgeführt, ähnlich wie bei Bomhardt.¹⁰¹ Hierbei wird zunächst der erste Chunk zum Training genutzt und das trainierte Model auf dem zweiten Chunk getestet. Anschließend wird das zweite Chunk zur Trainingsmenge hinzugefügt und damit der dritte Chunk getestet etc. Um reale Bedingungen herzustellen, sollen zur Trainingsmenge Beispiele nur hinzugefügt werden, wenn sie durch die vorgestellten Strategien so zugeordnet wurden und nicht, weil dies aus dem Expertenfeedback bekannt ist. Um etwaige Schwankungen und Verzerrungen zu vermeiden sollen die Chunks sich an die ursprüngliche Größe von 35 bzw. Vielfache davon anlehnen. Dadurch begründet ergeben sich für die Evaluation *drei Chunkgrößen von 35, 70 und 105 Artikeln*. Die zufällige Anordnung, die der Experte zu sehen bekam, dient im Folgenden als Baseline.

Ziel eines personalisierten Nachrichtenaggregators sollte es sein, die lesenswerten und interessanten Artikel möglichst weit oben anzuzeigen. Hierfür soll der *nDCG* zu jeder Liste berechnet werden (siehe auch 2.5.4), der die Reihung der Liste mit den einzelnen Artikelbewertungen ins Verhältnis setzt.

Die *Precision* wird außerdem wichtiger als der *Recall* eingeschätzt. Dies beruht einerseits auf der Annahme, dass Nutzer nur bereit ist, eine geringe Anzahl von Artikeln tatsächlich zu lesen und *nicht* daran interessiert ist, *alle zu lesen*. Außerdem wird angenommen, dass er relativ *indifferent innerhalb der höchstbewerten* Gruppe von Artikeln ist.

Allerdings sind diese Einschätzungen subjektiv und müssen im Kontext des Einsatzes gesehen werden, weshalb auch Metriken wie *Recall* und *F1* mit angegeben werden.

5.3 Evaluierung des Nutzer Feedbacks

Zuvor wurden in Kapitel 3.5 Strategien für die Verwendung des Nutzer Feedbacks vorgestellt, die auf diversen Annahmen beruhen. In diesem Abschnitt soll nun herausgefunden werden, welche Feedback Kombinationen die besten Ergebnisse liefern, um die vorherigen Annahmen zu bestätigen oder abzuweisen und um die beste Kombination bei den weiteren Evaluationen als Grundlage nutzen zu können. Es wurden hierfür Evaluierungen mit unterschiedlichen

¹⁰¹ Bomhardt (2004)

Kombinationen (für die Strategien, siehe 3.5) aus implizitem und explizitem Feedback durchgeführt (Abbildung 5-3). Die Ergebnisse müssen, wie zuvor erwähnt, auf den Kontext des Anwendungszwecks bezogen werden, einige Kombinationen sind jedoch anderen klar überlegen (siehe Evaluation Abbildung 8-15, Abbildung 8-16, Abbildung 8-17).

Rating > 1, Chunk 70	Micro-averaged			Macro-averaged			nDCG
	Precision	Recall	F1	Precision	Recall	F1	
ONLY_EXPLICIT	0,4811	0,6061	0,5364	0,5237	0,6132	0,5297	0,6233
EXPLICIT_CLICK	0,4424	0,8139	0,5732	0,4481	0,8110	0,5678	0,6525
EXPLICIT_CLICK_SKIP	0,5771	0,4372	0,4975	0,5703	0,4274	0,4787	0,5320
EXPLICIT_CLICK_TOP	0,4723	0,7749	0,5869	0,4738	0,7661	0,5783	0,6494
EXPLICIT_CLICK_PREDECESSOR	0,5814	0,4329	0,4963	0,5770	0,4238	0,4785	0,5323
EXPLICIT_CLICK_SUCCESOR	0,5814	0,4329	0,4963	0,5767	0,4238	0,4787	0,5272
EXPLICIT_CLICK_SKIP_TOP_PREDECESSOR_SUCCESOR	0,5303	0,4545	0,4895	0,5440	0,4723	0,4866	0,5561
EXPLICIT_CLICK_TOP_PREDECESSOR	0,5193	0,5238	0,5216	0,5252	0,5406	0,5198	0,5763
EXPLICIT_CLICK_PREDECESSOR_SUCCESOR_TOP	0,5245	0,4632	0,4920	0,5371	0,4810	0,4886	0,5593

Abbildung 5-3: Übersicht über Kombinationen aus impliziten Strategien für Chunks mit der Größe 70 und Bewertung größer 1.

Zusammengefasst ergeben sich folgende Erkenntnisse:

- Die besten Strategien in Bezug auf Precision waren SKIP, PREDECESSOR und SUCCESOR.
- Das ausschließliche explizite Feedback, auch in Kombination mit CLICK wurde immer bei der Precision durch die vorher genannten signifikant outperformed.

Hier ist festzustellen, dass ein System, allein basierend auf explizitem Feedback den hybriden Systemen klar unterlegen ist. Daher ist von einem PNA, basierend ausschließlich auf explizitem Feedback abzuraten.

Chunk: 70	Rating	Micro-averaged			Macro-averaged			nDCG
		Precision	Recall	F1	Precision	Recall	F1	
CLICK	> 0	0,7140	0,7749	0,7432	0,7279	0,7831	0,7353	0,6369
CLICK_SKIP	> 0	0	0	0	0	0	0	0
CLICK_TOP	> 0	0,7171	0,7749	0,7449	0,7308	0,7834	0,7371	0,6440
CLICK_PREDECESSOR	> 0	0	0	0	0	0	0	0
CLICK_SUCCESOR	> 0	0	0	0	0	0	0	0
CLICK_SKIP_TOP_PREDECESSOR_SUCCESOR	> 0	0,8837	0,4502	0,5965	0,8994	0,4486	0,5794	0,5477
CLICK_TOP_PREDECESSOR	> 0	0,9010	0,4313	0,5833	0,9072	0,4370	0,5810	0,5275
CLICK_PREDECESSOR_SUCCESOR_TOP	> 0	0,9000	0,4052	0,5588	0,9082	0,4111	0,5559	0,5209

Abbildung 5-4: Übersicht ohne explizites Feedback.

Bei der nächsten Evaluation wurde gänzlich *ohne explizites Feedback* durchgeführt (siehe Abbildung 8-18). Dabei gibt es zwei interessante Auffälligkeiten, die erwähnt werden sollten:

- Einige Feedback-Kombinationen funktionieren nicht und lieferten schlicht keine Einträge, vermutlich, weil zu viele widersprüchliche Signale generiert wurden.
- Andere Feedback-Kombinationen lieferten hingegen sehr stabile Werte, die in jedem Fall vergleichbar sind mit der hybriden Evaluation zuvor, so z.B. CLICK + TOP + PREDECESSOR.

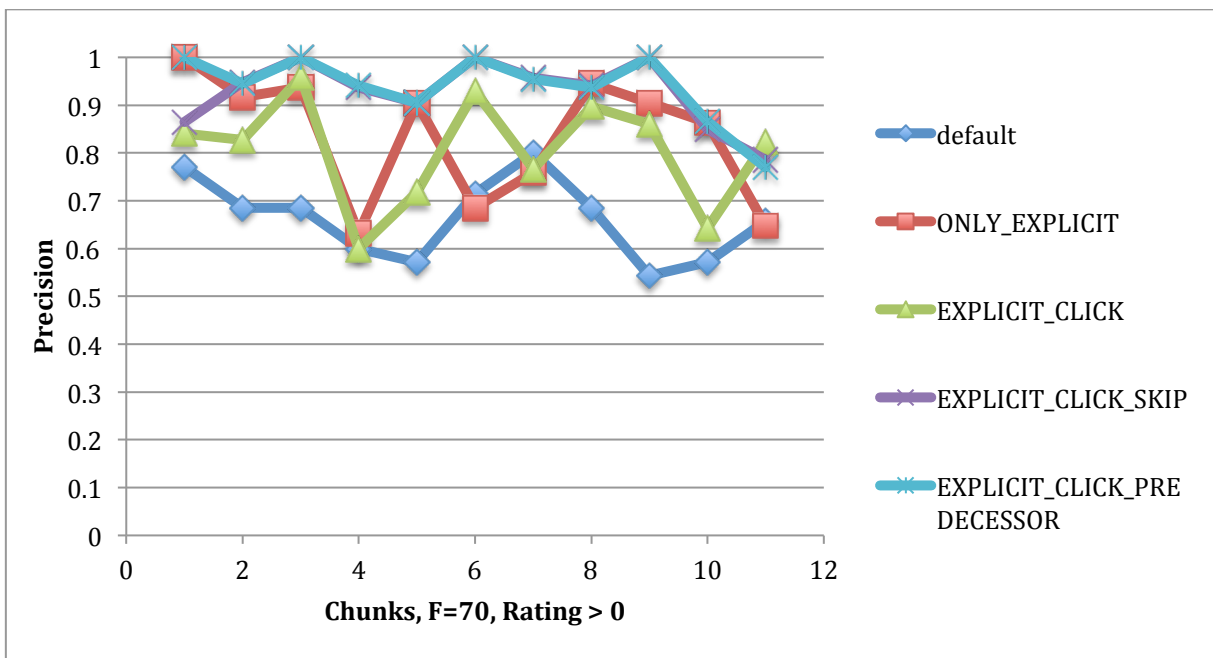


Abbildung 5-5: Precision nach jedem Chunk mit der Größe 70 für alle Artikel, die nicht als uninteressant bewertet wurden.

Der zeitliche Verlauf der Precision für Artikel, die als nicht uninteressant bewertet wurden, bestätigt die vorherigen Auffälligkeiten (Abbildung 5-5): EXPLICIT CLICK PREDECESSOR und EXPLICIT CLICK SKIP sind den anderen Strategien auch hier klar überlegen. Hier fällt vor allem auf, dass die Precision bei den anderen Strategien, vor allem beim ausschließlich expliziten Feedback und der Kombination mit den Artikelklicks stark schwankt. Die Glättung bei der SKIP ist vermutlich bedingt durch die vielen Rückmeldungen bei dieser Strategie. Dieser Effekt lässt sich auch bei anderen Auswertungen (siehe Abbildung 8-19, Abbildung 8-20, Abbildung 8-21) mit unterschiedlicher Chunk-Größe nachvollziehen, wobei zu

beobachtet ist, dass die Schwankungsbreite mit einer steigenden Mindestbewertung ebenfalls zunimmt.

5.4 Evaluierung der Klassifizierungsalgorithmen

Im Abschnitt zuvor wurde festgestellt, dass die SKIP Strategie aufgrund ihrer vielen Signale gute Ergebnisse liefert und daher als sinnvoll erachtet wird. Sie bietet sogar noch Potential, da sie in dieser Arbeit nur sehr simpel implementiert wurde. Auch die PREDECESSOR Strategie hat im vorangegangenen Abschnitt gute Ergebnisse geliefert, weshalb weiter mit diesen Kombinationen evaluiert werden soll. Zum Zwecke der Vergleichbarkeit werden außerdem noch EXPLICIT-ONLY sowie EXPLICIT-CLICK hinzugezogen.

In diesem Abschnitt sollen die Annahmen und Hypothesen bezüglich der Klassifizierungsalgorithmen aus Kapitel 3.6 überprüft werden. Zunächst sollen dabei die Algorithmen isoliert evaluiert und die Ergebnisse schließlich mit der Kombination verglichen werden.

kNN Strategy	Rating	Micro-averaged				Macro-averaged			
		Baseline	Precision	Recall	F1	Precision	Recall	F1	nDCG
ONLY_EXPLICIT	> 0	0,6594	0,8806	0,4673	0,6106	0,8907	0,4733	0,6057	0,6363
EXPLICIT_CLICK	> 0	0,6594	0,8677	0,5327	0,6601	0,8764	0,5407	0,6571	0,6468
EXPLICIT_CLICK_SKIP	> 0	0,6594	0,8810	0,4693	0,6124	0,8910	0,4750	0,6075	0,6431
EXPLICIT_CLICK_PREDECESSOR	> 0	0,6594	0,8810	0,4693	0,6124	0,8910	0,4750	0,6075	0,6431
ONLY_EXPLICIT	> 1	0,3679	0,5373	0,5125	0,5246	0,5586	0,5263	0,5225	0,6363
EXPLICIT_CLICK	> 1	0,3679	0,5290	0,5836	0,5550	0,5412	0,5995	0,5523	0,6468
EXPLICIT_CLICK_SKIP	> 1	0,3679	0,5390	0,5160	0,5273	0,5598	0,5293	0,5251	0,6431
EXPLICIT_CLICK_PREDECESSOR	> 1	0,3679	0,5390	0,5160	0,5273	0,5598	0,5293	0,5251	0,6431
ONLY_EXPLICIT	> 2	0,1679	0,2761	0,5781	0,3737	0,2892	0,6106	0,3661	0,6363
EXPLICIT_CLICK	> 2	0,1679	0,2548	0,6172	0,3607	0,2643	0,6454	0,3549	0,6468
EXPLICIT_CLICK_SKIP	> 2	0,1679	0,2788	0,5859	0,3778	0,2918	0,6163	0,3700	0,6431
EXPLICIT_CLICK_PREDECESSOR	> 2	0,1679	0,2788	0,5859	0,3778	0,2918	0,6163	0,3700	0,6431

Abbildung 5-6: Evaluation für kNN exklusiv, nur wortbasierte Features.

Der kNN trifft seine Entscheidung aufgrund der n-ähnlichsten Dokumente. Zunächst wurden hierfür nur wortbasierte Features genutzt (Abbildung 5-6). Obwohl der Algorithmus so simpel ist, erzielt er vergleichsweise gute Ergebnisse. Der Lernvorgang ist in nachfolgender Abbildung dargestellt (Abbildung 5-7).

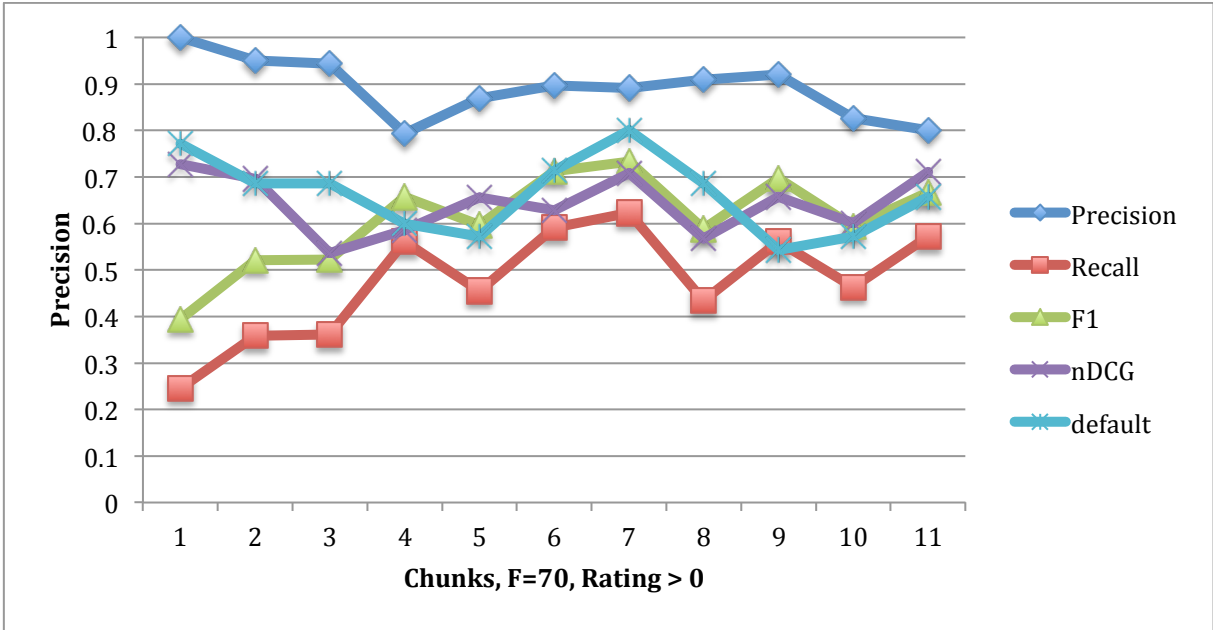


Abbildung 5-7: kNN für Chunk-Größe von 70 und Bewertungen größer 0.

Anschließend wurden andere Features mit einem Anteil von 0,5 hinzugezogen, wodurch sich das Ergebnis jedoch verschlechterte (Abbildung 5-8). Hierbei sind sowohl die Precision als auch Recall sowie nDCG gefallen.

kNN Strategy	Rating	Baseline	Micro-averaged			Macro-averaged			
			Precision	Recall	F1	Precision	Recall	F1	nDCG
ONLY_EXPLICIT	> 0	0,6594	0,78486	0,3901	0,5212	0,79633	0,3929	0,5189	0,5344
EXPLICIT_CLICK	> 0	0,6594	0,71159	0,5228	0,6027	0,73047	0,5318	0,5997	0,5497
EXPLICIT_CLICK_SKIP	> 0	0,6594	0,80591	0,3782	0,5148	0,81249	0,3816	0,5109	0,5191
EXPLICIT_CLICK_PREDECESSOR	> 0	0,6594	0,80894	0,3941	0,53	0,81268	0,3979	0,5254	0,5148
ONLY_EXPLICIT	> 1	0,3679	0,49801	0,4448	0,4699	0,51073	0,4521	0,4677	0,5344
EXPLICIT_CLICK	> 1	0,3679	0,43127	0,5694	0,4908	0,44807	0,5886	0,4894	0,5497
EXPLICIT_CLICK_SKIP	> 1	0,3679	0,49367	0,4164	0,4517	0,498	0,4212	0,4432	0,5191
EXPLICIT_CLICK_PREDECESSOR	> 1	0,3679	0,4878	0,427	0,4554	0,48706	0,4313	0,4459	0,5148
ONLY_EXPLICIT	> 2	0,1679	0,26693	0,5234	0,3536	0,27534	0,537	0,3439	0,5344
EXPLICIT_CLICK	> 2	0,1679	0,20485	0,5938	0,3046	0,21845	0,6271	0,3032	0,5497
EXPLICIT_CLICK_SKIP	> 2	0,1679	0,26582	0,4922	0,3452	0,26575	0,4984	0,3341	0,5191
EXPLICIT_CLICK_PREDECESSOR	> 2	0,1679	0,26829	0,5156	0,3529	0,26413	0,5206	0,3501	0,5148

Abbildung 5-8: kNN exklusiv, alle Features mit Gewichtung von 0,5.

Simple TfIdf		Micro-averaged				Macro-averaged			
Strategy	Rating	Baseline	Precision	Recall	F1	Precision	Recall	F1	nDCG
ONLY_EXPLICIT	> 0	0,6594	0,86901	0,5386	0,665	0,88016	0,5478	0,6587	0,6367
EXPLICIT_CLICK	> 0	0,6594	0,84656	0,6337	0,7248	0,8505	0,641	0,7232	0,6578
EXPLICIT_CLICK_SKIP	> 0	0,6594	0,86943	0,5406	0,6667	0,88046	0,5495	0,6604	0,6396
EXPLICIT_CLICK_PREDECESSOR	> 0	0,6594	0,86943	0,5406	0,6667	0,88046	0,5495	0,6604	0,6396
ONLY_EXPLICIT	> 1	0,3679	0,5016	0,5587	0,5286	0,52096	0,5736	0,5239	0,6367
EXPLICIT_CLICK	> 1	0,3679	0,49206	0,6619	0,5645	0,49498	0,6753	0,5611	0,6578
EXPLICIT_CLICK_SKIP	> 1	0,3679	0,50318	0,5623	0,5311	0,52247	0,5767	0,5263	0,6396
EXPLICIT_CLICK_PREDECESSOR	> 1	0,3679	0,50318	0,5623	0,5311	0,52247	0,5767	0,5263	0,6396
ONLY_EXPLICIT	> 2	0,1679	0,2492	0,6094	0,3537	0,26432	0,6403	0,348	0,6367
EXPLICIT_CLICK	> 2	0,1679	0,2328	0,6875	0,3478	0,23512	0,7167	0,3422	0,6578
EXPLICIT_CLICK_SKIP	> 2	0,1679	0,25159	0,6172	0,3575	0,26675	0,646	0,3515	0,6396
EXPLICIT_CLICK_PREDECESSOR	> 2	0,1679	0,25159	0,6172	0,3575	0,26675	0,646	0,3515	0,6396

Abbildung 5-9: Evaluation für Simple TfIdf exklusiv.

Der vorgestellte Simple TfIdf Score, der lediglich auf der Addition der TfIdf Gewichte beruht, erzielt bei Precision lediglich ein leicht schlechteres Ergebnis als der kNN zuvor (Abbildung 5-9). Er erreicht jedoch einen leicht besseren nDCG als der kNN.

Perceptron nur Wörter		Micro-averaged				Macro-averaged			
Strategy	Rating	Baseline	Precision	Recall	F1	Precision	Recall	F1	nDCG
ONLY_EXPLICIT	> 0	0,6594	0,90341	0,3149	0,467	0,90537	0,3198	0,4671	0,4919
EXPLICIT_CLICK	> 0	0,6594	0,91515	0,299	0,4507	0,92015	0,3045	0,451	0,5095
EXPLICIT_CLICK_SKIP	> 0	0,6594	0,90341	0,3149	0,467	0,90537	0,3198	0,4671	0,4919
EXPLICIT_CLICK_PREDECESSOR	> 0	0,6594	0,90341	0,3149	0,467	0,90537	0,3198	0,4671	0,4919
ONLY_EXPLICIT	> 1	0,3679	0,48864	0,306	0,3764	0,49924	0,3141	0,3753	0,4919
EXPLICIT_CLICK	> 1	0,3679	0,50909	0,2989	0,3767	0,51932	0,3074	0,3754	0,5095
EXPLICIT_CLICK_SKIP	> 1	0,3679	0,48864	0,306	0,3764	0,49924	0,3141	0,3753	0,4919
EXPLICIT_CLICK_PREDECESSOR	> 1	0,3679	0,48864	0,306	0,3764	0,49924	0,3141	0,3753	0,4919
ONLY_EXPLICIT	> 2	0,1679	0,24432	0,3359	0,2829	0,25107	0,3698	0,2777	0,4919
EXPLICIT_CLICK	> 2	0,1679	0,25455	0,3281	0,2867	0,25787	0,3641	0,2806	0,5095
EXPLICIT_CLICK_SKIP	> 2	0,1679	0,24432	0,3359	0,2829	0,25107	0,3698	0,2777	0,4919
EXPLICIT_CLICK_PREDECESSOR	> 2	0,1679	0,24432	0,3359	0,2829	0,25107	0,3698	0,2777	0,4919

Abbildung 5-10: Evaluation für Perceptron exklusiv, nur Wörter.

Der Perceptron hingegen konnte in diesem Zusammenhang überraschenderweise nicht überzeugen. Er erzielte ein schlechteres Ergebnis bei Precision unter den und nDCG (Abbildung 5-10). Gemeinsam mit den anderen Features lieferte der Perceptron jedoch schlechtere Werte (Abbildung 5-11). Die nächste Vermutung war, dass das Trainingsset nicht ausreichend war, sodass der Perceptron nicht genügend Beispiele für den Lernvorgang hatte. Auch dies konnte ausgeschlossen werden. Testweise wurden hierfür *alle* vom Experten

bewerteten Artikel nach jedem Chunk hinzugefügt¹⁰² und nicht nur das generierte Nutzerfeedback. Die Ergebnisse des Perceptrons wurden daraufhin sogar noch schlechter. Der Trick mit der Verkleinerung der Trainingsmenge (das zuvor beschriebene dynamische Splitting) machte in diesem Fall – womöglich aufgrund einer zu kleinen Trainingsmenge keinen nennenswerten Unterschied.

Perceptron alle Features Strategy	Rating	Micro-averaged				Macro-averaged			nDCG
		Baseline	Precision	Recall	F1	Precision	Recall	F1	
ONLY_EXPLICIT	> 0	0,6594	0,6992	0,695	0,6971	0,73566	0,7132	0,6782	0,6588
EXPLICIT_CLICK	> 0	0,6594	0,70255	0,7624	0,7312	0,72512	0,7724	0,7187	0,6747
EXPLICIT_CLICK_SKIP	> 0	0,6594	0,70321	0,7366	0,7195	0,72856	0,751	0,7073	0,6653
EXPLICIT_CLICK_PREDECESSOR	> 0	0,6594	0,70038	0,7267	0,7133	0,72631	0,7414	0,7013	0,6558
ONLY_EXPLICIT	> 1	0,3679	0,40837	0,7295	0,5236	0,45371	0,757	0,515	0,6588
EXPLICIT_CLICK	> 1	0,3679	0,40328	0,7865	0,5332	0,42806	0,8056	0,5248	0,6747
EXPLICIT_CLICK_SKIP	> 1	0,3679	0,40832	0,7687	0,5333	0,43706	0,7922	0,5265	0,6653
EXPLICIT_CLICK_PREDECESSOR	> 1	0,3679	0,40649	0,758	0,5292	0,43557	0,7809	0,5225	0,6558
ONLY_EXPLICIT	> 2	0,1679	0,18725	0,7344	0,2984	0,22271	0,7875	0,3027	0,6588
EXPLICIT_CLICK	> 2	0,1679	0,18796	0,8047	0,3047	0,20961	0,8431	0,3038	0,6747
EXPLICIT_CLICK_SKIP	> 2	0,1679	0,18904	0,7813	0,3044	0,21447	0,826	0,3066	0,6653
EXPLICIT_CLICK_PREDECESSOR	> 2	0,1679	0,18893	0,7734	0,3037	0,21479	0,8159	0,3063	0,6558

Abbildung 5-11: Perceptron exklusiv, alle Features.

Der Autor vermutet abschließend, das schlechte Abschneiden könnte an zwei Dingen gelegen haben, die jedoch nicht genauer in dieser Arbeit untersucht wurden. Wie zuvor beschrieben, wurden keine gesamten Artikelinhalte als Features an den Perceptron übergeben, sondern inhaltlich nur die Textfragmente aus den RSS Feeds. In der Folge könnte der Perceptron zu wenige Features für adäquate Empfehlungen gehabt haben, was das Ergebnis verzerrt bzw. beeinträchtigt hat. Der andere Grund könnte sein, dass der Perceptron zu restriktiv mit den vorhandenen Features umgeht, was zu einem Overfitting führt, während dies bei den anderen Algorithmen aufgrund ihrer Einfachheit nicht passiert ist. Für eine derartige Anwendung ist deshalb die Empfehlung, einen kNN oder den vorgestellten Simple TfIdf zu nutzen.

5.5 Evaluierung des Rankings

In diesem Abschnitt sollen die zuvor berechneten Empfehlungen verglichen werden mit der Evaluierung nach dem Ranking. Hierzu wurden zwei verschiedene Szenarios getestet:

¹⁰² Dabei wurden in mehreren Szenarien alle Bewertungen größer als 0 als positive Beispiele, danach alle größer als 1 etc., während die restlichen als negative Beispiele gewertet wurden.

1. Die Ergebnisliste ausschließlich sortiert nach dem *startPageScore* (wobei Artikel, die nicht auf der Startseite vorkamen, ausgelassen wurden).
2. Die Ergebnisliste ausschließlich sortiert nach dem *socialScore* (wobei auch hier Artikel mit einem *socialScore* von 0 ausgelassen wurden).

Im ersten Szenario wurden die Werte in Abbildung 5-12 erzielt. Hierbei fällt auf, dass die Precision im Vergleich zu vorher (Abbildung 5-6) 3-5% besser ist, jedoch zu Lasten des Recalls, wodurch sich auch die F1-Metrik verschlechtert. Ausgehend von der früheren Annahme, dass ein Leser ohnehin nicht interessiert ist, alle relevanten Artikel zu erhalten, kann dies als Vorteil gewertet werden. Es ist denkbar, dass weitere Schwellenwerte bezüglich der Maximalposition ebenfalls zu Verbesserungen führen könnten. Anzumerken ist, dass die Reihenfolge strikt nach dem *startPageCount* zu einer Verschlechterung des nDCGs geführt hat.

kNN Strategy	Rating	Micro-averaged			Macro-averaged			nDCG
		Precision	Recall	F1	Precision	Recall	F1	
ONLY_EXPLICIT	> 0	0,9143	0,3802	0,5371	0,9182	0,3832	0,5352	0,5536
EXPLICIT_CLICK	> 0	0,8669	0,5287	0,6568	0,8753	0,5369	0,6526	0,5710
EXPLICIT_CLICK_SKIP	> 0	0,9319	0,3525	0,5115	0,9359	0,3549	0,5077	0,5340
EXPLICIT_CLICK_PREDECESSOR	> 0	0,9365	0,3505	0,5101	0,9408	0,3530	0,5070	0,5415
ONLY_EXPLICIT	> 1	0,5762	0,4306	0,4929	0,5832	0,4368	0,4902	0,5536
EXPLICIT_CLICK	> 1	0,5195	0,5694	0,5433	0,5289	0,5846	0,5392	0,5710
EXPLICIT_CLICK_SKIP	> 1	0,5864	0,3986	0,4746	0,5871	0,4041	0,4674	0,5340
EXPLICIT_CLICK_PREDECESSOR	> 1	0,5873	0,3950	0,4723	0,5900	0,4000	0,4660	0,5415
ONLY_EXPLICIT	> 2	0,3143	0,5156	0,3905	0,3128	0,5268	0,3747	0,5536
EXPLICIT_CLICK	> 2	0,2500	0,6016	0,3532	0,2551	0,6296	0,3453	0,5710
EXPLICIT_CLICK_SKIP	> 2	0,3141	0,4688	0,3762	0,2982	0,4769	0,3532	0,5340
EXPLICIT_CLICK_PREDECESSOR	> 2	0,3122	0,4609	0,3722	0,2957	0,4617	0,3501	0,5415

Abbildung 5-12: kNN exklusiv, sortiert ausschließlich nach *startPageScore*.

Das gleiche Bild ergibt sich für den Simple TfIdf (Abbildung 5-13) und dem Perceptron (Abbildung 5-14).

Simple TfIdf		Micro-averaged				Macro-averaged			
Strategy	Rating	Baseline	Precision	Recall	F1	Precision	Recall	F1	nDCG
ONLY_EXPLICIT	> 0	0,6594	0,9030	0,4238	0,5768	0,9091	0,4282	0,5729	0,5659
EXPLICIT_CLICK	> 0	0,6594	0,8564	0,6139	0,7151	0,8589	0,6215	0,7141	0,5854
EXPLICIT_CLICK_SKIP	> 0	0,6594	0,9358	0,3465	0,5058	0,9389	0,3491	0,5020	0,5373
EXPLICIT_CLICK_PREDECESSOR	> 0	0,6594	0,9379	0,3287	0,4868	0,9384	0,3308	0,4829	0,5348
ONLY_EXPLICIT	> 1	0,3679	0,5612	0,4733	0,5135	0,5690	0,4790	0,5083	0,5659
EXPLICIT_CLICK	> 1	0,3679	0,4917	0,6335	0,5537	0,4936	0,6485	0,5512	0,5854
EXPLICIT_CLICK_SKIP	> 1	0,3679	0,5829	0,3879	0,4658	0,5775	0,3898	0,4550	0,5373
EXPLICIT_CLICK_PREDECESSOR	> 1	0,3679	0,5876	0,3701	0,4541	0,5853	0,3727	0,4452	0,5348
ONLY_EXPLICIT	> 2	0,1679	0,2996	0,5547	0,3890	0,3041	0,5661	0,3776	0,5659
EXPLICIT_CLICK	> 2	0,1679	0,2293	0,6484	0,3388	0,2315	0,6836	0,3345	0,5854
EXPLICIT_CLICK_SKIP	> 2	0,1679	0,3369	0,4922	0,4000	0,3200	0,5011	0,3964	0,5373
EXPLICIT_CLICK_PREDECESSOR	> 2	0,1679	0,3277	0,4531	0,3803	0,3099	0,4581	0,3823	0,5348

Abbildung 5-13: Simple TfIdf exklusiv, sortiert ausschließlich nach startPageScore.

Perceptron		Micro-averaged				Macro-averaged			
Strategy	Rating	Baseline	Precision	Recall	F1	Precision	Recall	F1	nDCG
ONLY_EXPLICIT	> 0	0,6594	0,9091	0,2772	0,4249	0,9080	0,2813	0,4250	0,4447
EXPLICIT_CLICK	> 0	0,6594	0,9136	0,2931	0,4438	0,9186	0,2993	0,4443	0,4750
EXPLICIT_CLICK_SKIP	> 0	0,6594	0,9231	0,2614	0,4074	0,9227	0,2647	0,4070	0,4366
EXPLICIT_CLICK_PREDECESSOR	> 0	0,6594	0,9231	0,2614	0,4074	0,9227	0,2647	0,4070	0,4366
ONLY_EXPLICIT	> 1	0,3679	0,5065	0,2776	0,3586	0,5049	0,2831	0,3553	0,4447
EXPLICIT_CLICK	> 1	0,3679	0,5062	0,2918	0,3702	0,5208	0,3013	0,3699	0,4750
EXPLICIT_CLICK_SKIP	> 1	0,3679	0,5105	0,2598	0,3443	0,5040	0,2647	0,3392	0,4366
EXPLICIT_CLICK_PREDECESSOR	> 1	0,3679	0,5105	0,2598	0,3443	0,5040	0,2647	0,3392	0,4366
ONLY_EXPLICIT	> 2	0,1679	0,2403	0,2891	0,2624	0,2407	0,3065	0,2529	0,4447
EXPLICIT_CLICK	> 2	0,1679	0,2593	0,3281	0,2897	0,2670	0,3641	0,2844	0,4750
EXPLICIT_CLICK_SKIP	> 2	0,1679	0,2448	0,2734	0,2583	0,2364	0,2894	0,2533	0,4366
EXPLICIT_CLICK_PREDECESSOR	> 2	0,1679	0,2448	0,2734	0,2583	0,2364	0,2894	0,2533	0,4366

Abbildung 5-14: Perceptron exklusiv, sortiert ausschließlich nach startPageScore.

Im zweiten Szenario wurde die Liste rein nach dem socialScore sortiert. Die angenommenen Parameter waren hierbei $\omega = 0,5$ und für die anderen Werte 0,1. SocialScores von 0 wurden ausgelassen, jedoch traten diese nicht häufig auf, da in jedem Fall noch ein fixer Teil addiert wurde.

kNN socialScore Sortierung		Micro-averaged				Macro-averaged			nDCG
Strategy	Rating	Baseline	Precision	Recall	F1	Precision	Recall	F1	nDCG
ONLY_EXPLICIT	> 0	0,6594	0,8806	0,4673	0,6106	0,89074	0,4733	0,6057	0,5673
EXPLICIT_CLICK	> 0	0,6594	0,86774	0,5327	0,6601	0,87636	0,5407	0,6571	0,5859
EXPLICIT_CLICK_SKIP	> 0	0,6594	0,88104	0,4693	0,6124	0,89098	0,475	0,6075	0,5712
EXPLICIT_CLICK_PREDECESSOR	> 0	0,6594	0,88104	0,4693	0,6124	0,89098	0,475	0,6075	0,5712
ONLY_EXPLICIT	> 1	0,3679	0,53731	0,5125	0,5246	0,55865	0,5263	0,5225	0,5673
EXPLICIT_CLICK	> 1	0,3679	0,52903	0,5836	0,555	0,54117	0,5995	0,5523	0,5859
EXPLICIT_CLICK_SKIP	> 1	0,3679	0,53903	0,516	0,5273	0,55984	0,5293	0,5251	0,5712
EXPLICIT_CLICK_PREDECESSOR	> 1	0,3679	0,53903	0,516	0,5273	0,55984	0,5293	0,5251	0,5712
ONLY_EXPLICIT	> 2	0,1679	0,27612	0,5781	0,3737	0,28917	0,6106	0,3661	0,5673
EXPLICIT_CLICK	> 2	0,1679	0,25484	0,6172	0,3607	0,26429	0,6454	0,3549	0,5859
EXPLICIT_CLICK_SKIP	> 2	0,1679	0,27881	0,5859	0,3778	0,29181	0,6163	0,37	0,5712
EXPLICIT_CLICK_PREDECESSOR	> 2	0,1679	0,27881	0,5859	0,3778	0,29181	0,6163	0,37	0,5712

Abbildung 5-15: kNN exklusiv, Social Score Sortierung.

Auch hier zeigt sich, dass keine Verbesserungen bezüglich des nDCG erzielt werden konnten. Es wurde außerdem ein Mindestscore gesetzt, um ausschließlich populäre Artikel zu erhalten, jedoch hat dies eher zu einer Verschlechterung der Precisions und des Recalls geführt (Abbildung 8-22).

Das beste Ergebnis hierfür lieferte bisher eine reine Sortierung nach dem Score für das persönliche Interesse. Aufgrund dessen sollte dieser Score auch für das Ranking in diesem einfachen Modell herangezogen werden. Obwohl der *socialScore* hier nicht zur besseren Ordnung der Reihenfolge beiträgt, wird dennoch vermutet, dass er eine wichtige Rolle bei Generierung von Serendipitätsartikeln einnimmt. Durch ihn werden einerseits Trends abgedeckt und andererseits beliebte Artikel gefunden, die den Nutzer überraschen können.

6 Daten für Feldstudie zur Zahlungsbereitschaft

Neben der zuvor beschriebenen technischen Implementierung eines personalisierten Nachrichtenaggregators (PNA) sollte auch der wirtschaftliche Aspekt einer solchen Idee genauer beleuchtet werden. Hierzu wurde eine Feldstudie unter mehreren hundert Teilnehmern mit dem Ziel durchgeführt, die Zahlungsbereitschaft für PNA und damit verbunden, den optimalen Grad der Personalisierung zu bestimmen. Um die Zahlungsbereitschaft jedoch adäquat abzufragen, sollten die Testpersonen selbst Erfahrungen mit einem PNA sammeln. Aus diesem Grund wurde für sie ein vierstufiges Framework mit ansteigender Personalisierung entwickelt:

- a. Vollständige Automatisierung – der Nutzer hat kann weder zu Beginn noch im späteren Verlauf die Personalisierung beeinflussen.
- b. Hohe Automatisierung – der Nutzer kann zu Beginn Nachrichtenseiten auswählen und später durch „Daumen hoch“ und „Daumen herunter“ Feedback über die Personalisierungsgüte geben.
- c. Teilweise Automatisierung – der Nutzer hat die vorherigen Möglichkeiten und kann sich zudem manuell mit Tags (Quellen, Stichworte, Kategorien) seine *eigenen, themenbasierten Zeitungen* erstellen.
- d. Teilweise Automatisierung mit Customizing – vorherige Möglichkeiten sowie Customizing anhand von Farbe, Darstellungsstil etc.

Für diese vier Gruppen wurde eine Gleichverteilung (25%) gewählt. Je nachdem, um welche Gruppe es sich handelt, müssen die Teilnehmer unterschiedlicher Registrierungsprozesse durchlaufen. Am Ende des Registrierungsprozesses steht für alle Gruppen eine kurze Abfrage der demographischen Daten und ihrer aktuellen Zahlungsbereitschaft.

Anschließend werden ihnen, ebenfalls gruppenabhängig, unterschiedliche Personalisierungsmöglichkeiten geboten. Ab Gruppe d) kommt zudem das Customizing hinzu. Die genannten Elemente sollen im Folgenden beschrieben werden.

Nach Erreichung einer gesetzten Nutzungsschranke werden die Teilnehmer aufgefordert, einer zweiten, ausführlicheren Umfrage beizuwohnen.

6.1 Registrierungsprozess

Für das Experiment musste sich jeder Teilnehmer zu Beginn mit seiner Emailadresse anmelden, eine anonyme Nutzung wurde aufgrund der unzureichenden Eindeutigkeit nicht zugelassen. Alle Teilnehmer müssen zum Abschluss der Registrierung personenbezogene Daten angeben, bevor sie den PNA zum ersten Mal nutzen, wobei folgende Daten abgefragt werden:

- Geschlecht
- Alter
- Bildungsgrad
- Berufsstand
- Aktuelle Zahlungsbereitschaft für einen derartigen Dienst

Diese Daten dienen ausschließlich dem Umfragezweck und werden nicht weiterverwendet.

Gruppe a) erhält nach dieser Anmeldung sofort eine Zusammenstellung aus populären Artikeln verschiedener Quellen, wohingegen den Teilnehmern ab Gruppe b) die Möglichkeit geboten wird aus einer Liste von Nachrichtenseiten ihre Favoriten zu wählen (siehe Abbildung 6-1).

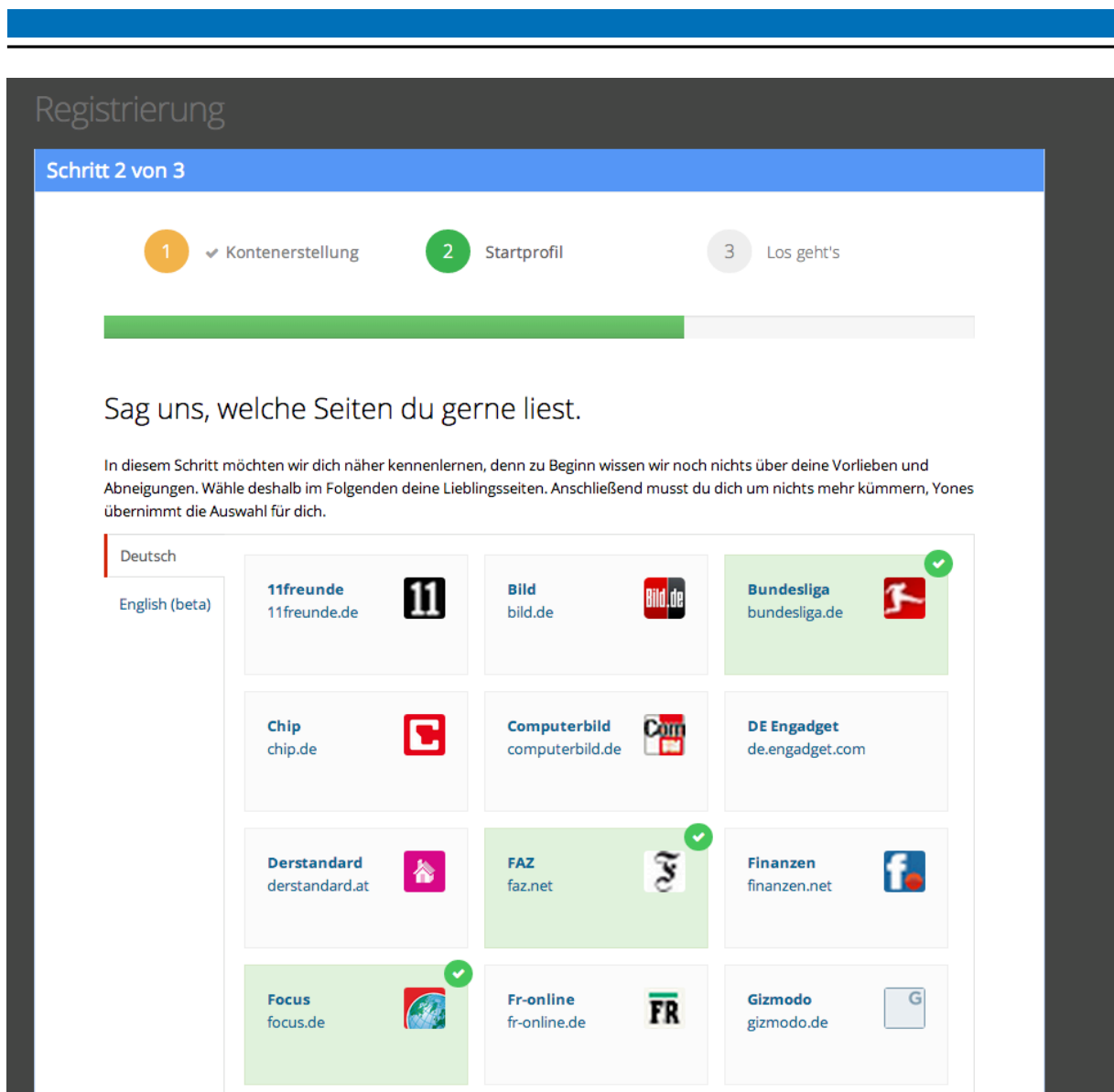


Abbildung 6-1: Quellenauswahl im Registrierungsprozess.

Wählt ein Teilnehmer Seiten aus dieser Übersicht, erhält er in einem ersten Schritt populäre Artikel dieser Quellen aus den letzten 48 Stunden.

Anschließend werden für alle Nutzer die maschinellen Lernalgorithmen im Hintergrund gestartet.

6.2 Personalisierungsmöglichkeiten

Die Teilnehmer können über die automatische Anpassung an ihre Interessen korrigieren und unterstützen, indem sie dem System Rückmeldung über die Relevanz eines Artikels angezeigten Artikels geben. Hierzu dienen in erster Linie die Knöpfe „mehr davon“ und „weniger“ (Abbildung 6-2: Explizites Feedback und Personalisierungselemente). Nutzt ein

Teilnehmer diese Form des expliziten Feedbacks, fließen die Features dieses Artikels vollständig in den Personalisierungsalgorithmus ein. Diese Form des Feedbacks wird oftmals auch durch „Daumen hoch“ bzw. „Daumen herunter“ repräsentiert.

Daneben können die Teilnehmer auch bestimmte Eigenschaften eines Artikels nutzen, um damit ihre *eigene, themenbasierte Zeitung* zu erstellen. Hierdurch sollen die Teilnehmer eine Gruppierung der Artikel in bestimmte Bereiche (z.B. Technik und Wirtschaft) vornehmen können, sodass die Artikel nicht ungeordnet im Nachrichtenstream erscheinen. Sie können dabei beliebig viele eigene Zeitungen (im Folgenden „Kanäle“) erstellen, diese auch nach ihren Wünschen benennen und mit folgenden Attributen (Abbildung 6-2: Explizites Feedback und Personalisierungselemente) akzentuieren:

- Schlagworte zum Artikel
- Artikelkategorie
- Nachrichtenquelle (Hauptseite)
- RSS-Feed zum Artikel

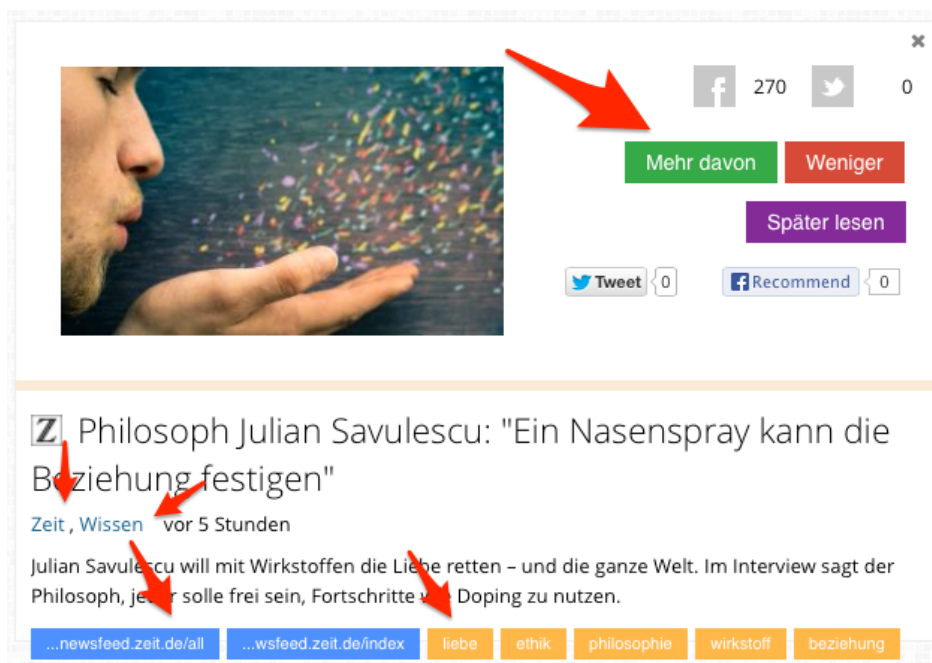


Abbildung 6-2: Explizites Feedback und Personalisierungselemente.

Diese Elemente müssen vom Nutzer über Drag & Drop in die erstellten Kanäle (Abbildung 6-3: Kanäle mit persönlichen Schwerpunkten) gezogen werden und verstärken dort die ausgewählten Attribute der zuvor berechneten Articlempfehlungen. Bei Hinzunahme bspw. der Quelle *Spiegel Online* und des Stichworts *Apple* werden übereinstimmende (empfohlene) Artikel höher gewichtet, wodurch sie weiter oben im Ranking gelangen.

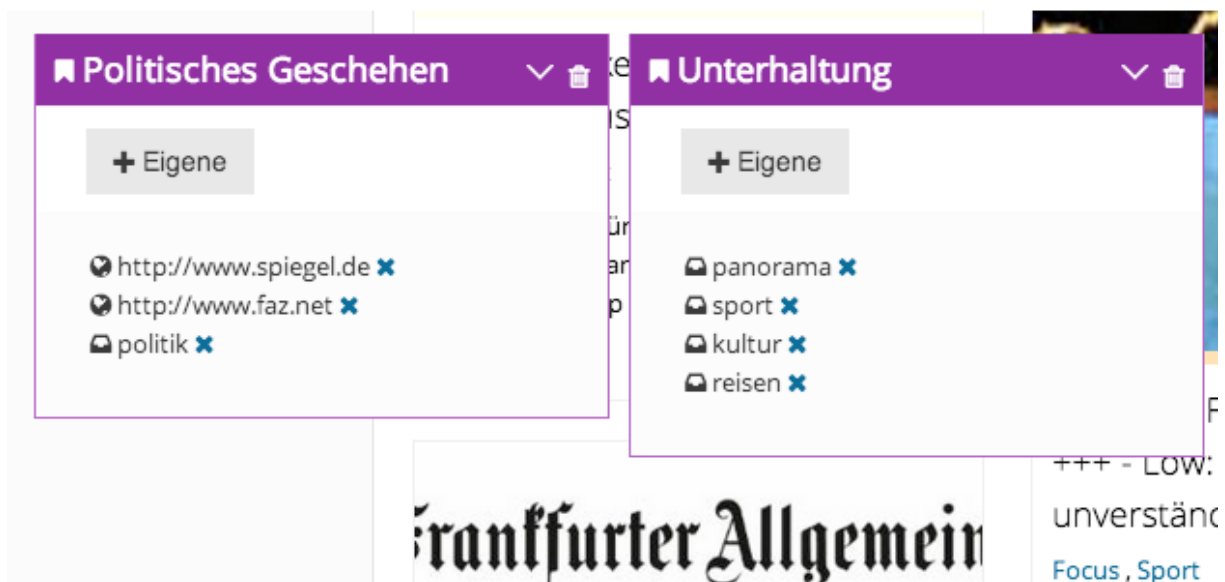


Abbildung 6-3: Kanäle mit persönlichen Schwerpunkten.

6.3 Customizing

Das Customizing verfolgt den Ansatz, dass die Teilnehmer sich ihr Nachrichtenportal nach ihren Wünschen gestalten können und betrifft ausschließlich die Gruppe d). Dieser Ansatz wird z.B. auch von Feedly (siehe Kapitel 2.7.3) verfolgt. Folgende Bereiche der Weboberfläche sind vom Nutzer veränderbar:

- Artikellayout (Kachel- und Listendesign; 1)
- Hintergrundmuster im Inhaltsbereich (fünf verschiedene Muster; 2)
- Farbstil für Navigationselemente und Seitenleiste (schwarz, braun, lila, blau und hellgrau; 3)

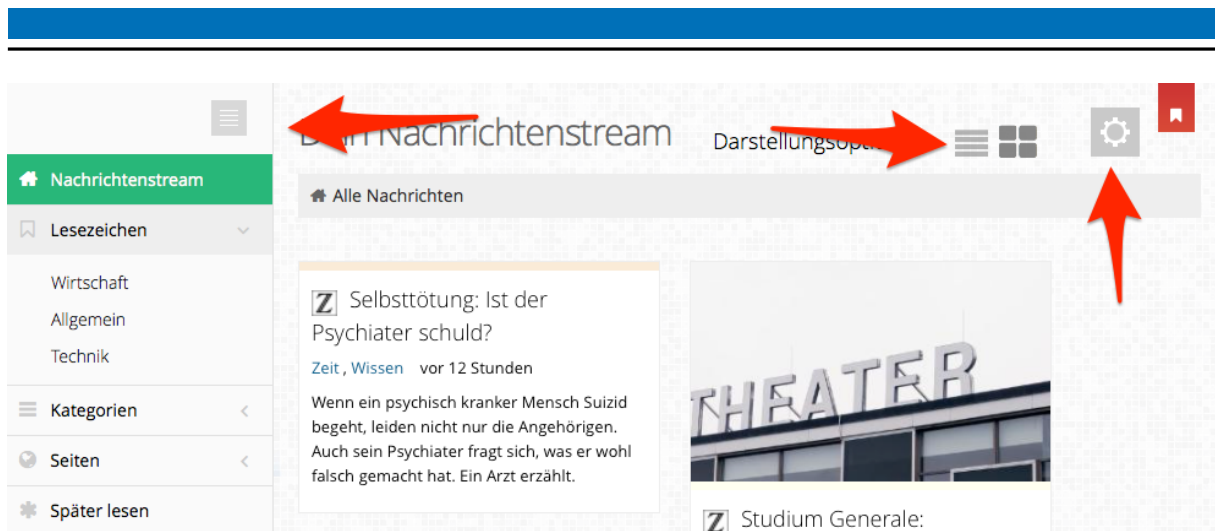


Abbildung 6-4: Oberfläche mit Customizing-Elementen

6.4 Nutzungsschranke für finale Umfrage

Von den Teilnehmern wird erwartet, dass sie ihre persönlichen Erfahrungen mit dem Prototypen des PNA sammeln. Um dies sicherzustellen, wurde eine Nutzungsschranke gesetzt, nach der ein Teilnehmer den finalen Fragebogen angezeigt bekommt.

Hierzu wurde einerseits die Anzahl der Sitzungen verwendet, wobei eine Sitzung eine Laufzeit von einer Stunde hat. Hiermit werden unabhängige Logins festgehalten, deren Dauer der Teilnehmer nicht manipulieren kann. Andererseits wurde die Anzahl der gelesenen bzw. bewerteten Artikel herangezogen. Die Kriterien für den finalen Fragebogen müssen beide erfüllt sein und lauten:

1. Mindestens drei Sitzungen
2. Mindestens zehn gelesene bzw. bewerte Artikel

Der Aufruf zum finalen Fragebogen wird als zusätzliches Element, ähnlich wie ein Artikel, an erster Stelle eingeblendet (Abbildung 6-5).



Abbildung 6-5: Fragebogenelement in Artikelliste

6.5 Ausführliche Umfrage nach Nutzung

Nachdem die Nutzungsschranke erreicht wurde, konnten die Teilnehmer an der finalen Umfrage teilnehmen. Dort wurden 30 Fragen zu ihrer Nutzung mit dem PNA sowie zu demographischen Gesichtspunkten gestellt. Unter den Fragen waren Kontrollfragen, um ein zufälliges Durchklicken zu vermeiden.

Die Nutzungsfragen bezogen sich auf folgende Bereiche:

- Kundenzufriedenheit (Homburg et al., 2006)
- Zahlungsbereitschaft (Voelckner, 2006)
- Preference Fit (Franke et al., 2010)
- Wahrgenommener Personalisierungsaufwand (Dellaert & Stremersch, 2005)
- Wahrgenommener Personalisierungsnutzen (Franke & Schreier, 2010)
- Psychologischer Besitz / Grad der Selbsteinbringung (Franke et al., 2010)
- Preference Insight (Franke et al., 2009)
- Product Involvement (Franke et al., 2009)

6.6 Zusammenfassung

Die Teilnehmer der Umfrage wurden bei der Umfrage bei der Registrierung gleichmäßig in vier Gruppen unterteilt. Der Registrierungsprozess beinhaltete für alle Gruppen außer a) bereits erste Personalisierungsmechanismen und endete mit einer kurzen Erfassung der demographischen Daten sowie der Zahlungsbereitschaft. Im Anschluss konnten die Teilnehmer den PNA mit unterschiedlichen Personalisierungsgraden nutzen. Nach einer vorher festgelegten Nutzungsdauer (drei Sitzungen, mindestens zehn bewertete Artikel) konnten die Teilnehmer an der finalen Umfrage teilnehmen.

Im Folgenden soll noch einmal ein kurzer Überblick über die Unterschiede bei den vier Testgruppen gegeben werden (Abbildung 6-6).

Gruppe / Eigenschaft	A	B	C	D
Quellenauswahl im Registrierungsprozess	–	✓	✓	✓
Relevanzfeedback über Daumen	–	–	✓	✓
Relevanzfeedback über Tags (Schlüsselwörter, Kategorie, Seite, Feed)	–	–	✓	✓
Erstellung eigener, themenbasierter Zeitung	–	–	–	✓
Customizing (Hintergrundfarbe, Themenfarbe, Artikelstil)	–	–	–	✓

Abbildung 6-6: Zusammenfassung zu unterschiedlicher Personalisierung der Testgruppen.

7 Resultate der Feldstudie

7.1 Deskriptive Auswertung der Zahlungsbereitschaft

Die Erhebung der Zahlungsbereitschaft wurde unter 195 Teilnehmern aus dem deutschsprachigen Raum durchgeführt (n=195). Diese Teilnehmer wurden, wie zuvor beschrieben, zufällig in vier Gruppen eingeteilt, wobei eine gleichmäßige Verteilung angestrebt wurde (Kapitel 6). Die durchgeführten Erhebungen unterteilen sich in:

- Kurze Online-Befragung vor der Nutzung (im Registrierungsprozess)
- Ausführliche Online-Befragung nach definierter Nutzung (Kapitel 6.4)

Über die Online-Befragung zu Beginn wurden 394 Teilnehmer zu demographischen Details sowie zu ihrer aktuellen Zahlungsbereitschaft für digitale Nachrichten befragt („Wie viel wärst du generell bereit, für elektronische Nachrichten im Monat zu bezahlen?“). Die Fragebogen waren vom 15.7.2013 bis zum 31.8.2013 im Internet zugänglich und wurden über verschiedene Kanäle (Universitäten in Darmstadt und München, soziale Netzwerke, Foren) beworben.

Nachdem eine definierte Nutzung (Kapitel 6.4) festgestellt werden konnte, wurden die Teilnehmer zu einer zweiten, ausführlichen Umfrage eingeladen. Die hohe Rücklaufquote betrug etwa 50%. Die hohe Teilnehmerquote wurde einerseits erreicht, da der Fragebogen schnell zu beantworten war (etwa 5-10min) und außerdem, da die Teilnehmer Aussicht auf den Gewinn eines Tablets, E-book-Readers sowie mehrerer Gutscheine hatten. Andererseits lässt die Teilnehmerquote auf ein Interesse der Internetnutzer an der Thematik schließen. Von den 394 Teilnehmern mussten bei der Vorbefragung aufgrund fehlender Daten (missing values) ein Teil ausgenommen werden. Bei der finalen Umfrage mussten abermals Datensätze ausgeschlossen werden, da sie Ausreißer, alternative Konten etc. darstellten.

Die 195 Einzelergebnisse verteilen sich sehr gleichmäßig über die vier Testgruppen wie folgt:

- a) 50 Teilnehmer
- b) 49 Teilnehmer

c) 47 Teilnehmer

d) 49 Teilnehmer

Die demographischen Charakteristika der Untersuchung über alle Gruppen können wie folgt zusammengefasst werden:

- Die Befragung wurde überwiegend von Männern beantwortet (68,5%).
- Die Teilnehmer verfügten über ein Bildungsniveau von mindestens einer Fachhochschulreife und die Mehrheit ein Abitur (85,8%).
- Im Durchschnitt waren die Teilnehmer genau 25 Jahre alt, wobei das Minimum 19 und das Maximum 40 Jahre waren.
- Bei der Tätigkeit ist festzustellen, dass die meisten Teilnehmer Studenten sind (75,3%), gefolgt von Vollzeitarbeitnehmern (16,8%) im Durchschnitt 1,57. Median 1.

Es handelt sich demnach um eine sehr junge, gebildete Teilnehmergruppe mit überwiegend Männern, was durch die Bekanntmachung vor allem im Kreise der TU Darmstadt sowie der LMU München und dort vor allem im technischen und wirtschaftlichen Bereich begründet ist. Dadurch kann mit großer Sicherheit gesagt werden, dass das vorliegende Sample nicht repräsentativ ist.

Diese Kerngruppe ähnelt sehr stark der Beschreibung der *Digital Natives*, besonders den *Digital Souveränen*, die zu 41% bzw. 15% der Bevölkerung gehören.¹⁰³ Deshalb werden die Ergebnisse dennoch für wertvoll erachtet, da die in den Daten vorherrschende Gruppe bzw. die Digital Natives durchaus als Interessent für digitale Nachrichten aus verschiedenen Quellen gilt.^{104 105}

¹⁰³ Vgl. Deutsches Institut für Vertrauen und Sicherheit im Internet (DIVSI), <https://www.divsi.de/publikationen/studien/divsi-milieu-studie/internet-milieus-zu-vertrauen-und-sicherheit-im-netz/510-digital-souveraene/>, 2013 (Abruf 16.09.2013)

¹⁰⁴ Dieselbe.

¹⁰⁵ Siehe Zahlungsbereitschaft

7.1.1 Evaluation der Vorbefragung

Bei der Registrierung bereits wurde jeder Teilnehmer einer unveränderbaren Testgruppe zugeteilt. Die erste kurze Online-Vorbefragung war dabei immer der letzte Schritt der Anmeldung. Aufgrund der ansteigenden Personalisierung innerhalb der Testgruppen, war es für Gruppe C und D notwendig, zuvor aus einer Liste die favorisierten Quellen auszuwählen (Abbildung 6-1: Quellenauswahl im Registrierungsprozess). Damit haben die Gruppen C und D bereits eine manuelle Personalisierungsvorleistung erbracht, die ihre anschließende Angabe der Zahlungsbereitschaft hätte beeinflussen können. Damit verbundene, mögliche Auffälligkeiten bei den Gruppen B, C und D sind jedoch nicht festzustellen. Aus diesem Grund werden die Zahlungsbereitschaften aller Gruppen vor der Nutzung zusammengefasst.

Zahlungsbereitschaft

Die Zahlungsbereitschaft vor der Nutzung des PNA liegt über alle Gruppen (n=195) bei 0,78 Euro, mit einem Minimum von 0 und einem Maximum von 10 Euro. Hierbei lag die Standardabweichung mit 1,52 Euro relativ hoch, was auf eine große Unsicherheit schließen lässt.

Effekte mit anderen Variablen

Zwischen der ausgeübten beruflichen Tätigkeit und der Zahlungsbereitschaft konnte keine bzw. nur eine extrem schwach negative und nicht signifikante Korrelation festgestellt werden. Auch beim Bildungsgrad und der Zahlungsbereitschaft konnte nur eine sehr schwach positive, nicht signifikante, Korrelation festgestellt werden.

Beim Alter und der Zahlungsbereitschaft hingegen wird eine positive Korrelation vermutet, da der Korrelationskoeffizient nur knapp unter dem kritischen Wert für Signifikanz liegt. Dies würde bedeuten, dass ältere Nutzer tendenziell eine höhere Zahlungsbereitschaft haben. Diese Aussage ließe sich jedoch nicht generalisieren, da das Sample nicht repräsentativ ist und im Widerspruch zu vorherigen Erkenntnissen aus anderen Umfragen steht.¹⁰⁶ Allerdings ließe sich die Aussage möglicherweise auf die Gruppe der Digital Natives übertragen, da wie zuvor erläutert (siehe Kapitel 7.1), eine starke Ähnlichkeit zu dieser Gruppe vorliegt.

¹⁰⁶ Siehe bspw. Abbildung 8-8: PWC, WAN. Moving into Multiple Business Models: Outlook for Newspapers Publishing in the Digital Age, 2009

7.1.2 Evaluation nach der Nutzung

Bei der Auswertung nach der Nutzung fällt auf, dass die Zahlungsbereitschaften für die einzelnen Testgruppen, wie zu erwarten, stark voneinander abweichen. Aus diesem Grunde sollen sie im Folgenden auch getrennt erläutert werden.

Zahlungsbereitschaft

Zunächst die aggregierten Ergebnisse über alle Gruppen (n=195). Die bereinigte Zahlungsbereitschaft ist *nach* der Nutzung signifikant auf durchschnittlich 2,48 Euro gestiegen, mit einem Minimum weiterhin bei 0 und einem Maximum bei 5,99 Euro. Die Standardabweichung ist mit 1,68 Euro absolut etwas höher als zuvor. In Anbetracht der Tatsache, dass die durchschnittliche Zahlungsbereitschaft nunmehr dreimal so hoch ist wie zuvor, ist sie relativ gesehen jedoch stark gesunken. Dies lässt auf eine geringere Unsicherheit schließen, was darauf zurückzuführen ist, dass die Nutzer ausreichend Erfahrung mit dem PNA gesammelt haben und so besser ihre Entscheidung bewerten konnten.

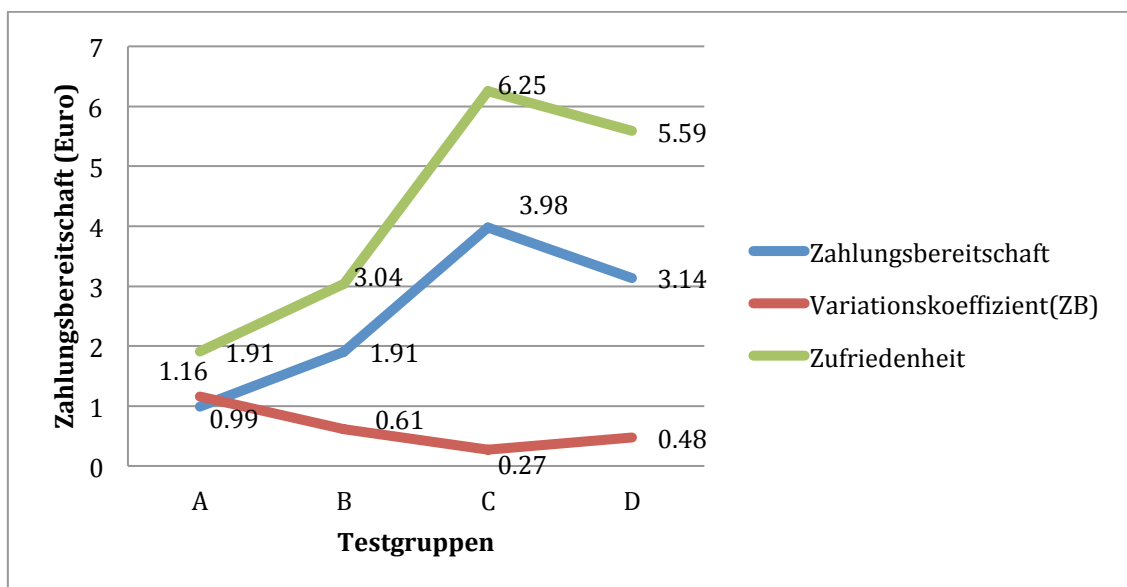


Abbildung 7-1: Getrennte Zahlungsbereitschaften der Testgruppen, einschließlich Variationskoeffizienten (ZB) und Zufriedenheiten.

Die Zahlungsbereitschaft für Gruppe A liegt nunmehr bei 1,16 Euro, für Gruppe B bei 1,91 Euro, für C und D, bei 3,98 Euro sowie 3,14 Euro (die weiteren Auswertungen befinden sich im Anhang). Innerhalb der Gruppen zeichnet sich demnach eine signifikant unterschiedliche

Zahlungsbereitschaft ab (Abbildung 6-1: Quellenauswahl im Registrierungsprozess). So steigt diese gleichsam mit der Personalisierung bis zur Gruppe C an und fällt bei Gruppe D wieder stark ab.

Hierfür wurde zunächst eine Varianzanalyse (ANOVA) über die Gruppen und ihrer Zahlungsbereitschaft durchgeführt. Die Nullhypothese, dass alle Mittelwerte gleich sind, musste verworfen werden (siehe Tabelle 4: ANOVA zu Zahlungsbereitschaft und Gruppen). Daher kann man sagen, dass die Mittelwerte sich stark signifikant unterscheiden. Die einzelnen Personalisierungsfunktionen haben also mit hoher Sicherheit einen Einfluss.

Es wurde außerdem ein Bonferroni-Test durchgeführt, der als Ergebnis hatte, dass die Verteilung der Mittelwerte in den Gruppen nicht zufällig ist, sondern mit der Personalisierung steigt und bei Gruppe D wieder leicht fällt. Hierfür wurde ebenfalls eine starke Signifikanz gefunden (siehe Tabelle 5: Bonferroni-Test der Zahlungsbereitschaft nach Gruppen). Personalisierung ist demnach bis zu einem bestimmten Maß förderlich.

Mit dem Abfall bei Gruppe D einhergehend, sinkt der Variationskoeffizient bis zur Gruppe C und steigt danach wieder an, allerdings ist er in allen Fällen weitaus geringer als bei der Vorbefragung. Dieser Intergruppentrend des Variationskoeffizienten lässt darauf schließen, dass die Teilnehmer sich bei Entscheidung bei Gruppe C am sichersten waren, dies also am besten einschätzen konnten. Wohingegen sie die zusätzlichen Customizing-Elemente von Gruppe D eher nicht quantifizieren konnten oder als unnützlich erachteten, daher die Unsicherheit anstieg und die Zahlungsbereitschaft fiel. Auch die Zufriedenheit lässt in dieser Testgruppe nach, was diesen Eindruck bestätigt. Gruppe C stellt also mit großer Sicherheit den optimalen Grad der Personalisierung mit der höchsten Zahlungsbereitschaft dar.

Effekte mit anderen Variablen

Die Zahlungsbereitschaft in der Vorbefragung korreliert schwach positiv mit der in der ausführlichen Umfrage, was wenig verwunderlich ist, da dies die persönliche Wertschätzung für journalistische Inhalte widerspiegelt.

Ein ebenso wenig verwunderlicher, doch sehr starker Zusammenhang besteht, wie bereits erwähnt, zwischen Zufriedenheit und Zahlungsbereitschaft. Über alle Gruppen lässt sich dieser Effekt wie folgt ausdrücken: Steigt die Zufriedenheit der Nutzers um einen Punkt, erhöht sich seine Zahlungsbereitschaft um 0,75 Euro. Aufgrund der maximal erreichten

Zufriedenheit von 6,25/7 in Gruppe C könnte zudem noch Raum nach oben für die Zahlungsbereitschaft sein.

Ein signifikanter Zusammenhang zwischen dem Geschlecht und der Zahlungsbereitschaft konnte auch hier nicht festgestellt werden. Auch der Einfluss von Bildung und beruflicher Tätigkeit ist nur marginal.

7.1.3 Zusammenfassung

In erster Linie sollte mit der durchgeführten Umfrage die Zahlungsbereitschaft für digitale Nachrichten, speziell in der Darstellung durch einen PNA erörtert werden. Außerdem sollte in diesem Zusammenhang der optimale Personalisierungsgrad mit der höchsten Zahlungsbereitschaft gefunden werden.

Die Ergebnisse der Umfrage haben gezeigt, dass vor der Nutzung mit dem PNA Prototypen nur eine geringe Zahlungsbereitschaft für digitale Nachrichten bestand und damit eine große Unsicherheit der Bewertung einherging. Nach einer vorbestimmten Nutzung haben die Teilnehmer im Durchschnitt die dreifache Zahlungsbereitschaft gezeigt. Diese stieg gleichsam mit dem Personalisierungsgrad an und fand bei der Testgruppe C (siehe Kapitel 6) ihren Höchststand. Dabei verhielt sich die Zufriedenheit proportional zur Zahlungsbereitschaft. Eine erhöhte Zahlungsbereitschaft für das Customizing eines PNA (siehe Kapitel 6.3) konnte nicht festgestellt werden. Customizing führte sogar zu einer 20% geringeren Zahlungsbereitschaft und einem Anstieg in der Unsicherheit. Weitere Zusammenhänge zwischen Geschlecht, Bildung oder beruflicher Tätigkeit und der Zahlungsbereitschaft konnten nicht festgestellt werden. Es liegt jedoch die Vermutung nahe, dass ein positiver Zusammenhang zwischen der tendenziellen Zahlungsbereitschaft vor der Nutzung und dem Alter des Nutzers vorliegt. Dies konnte jedoch nicht endgültig bestätigt werden.

8 Fazit und Ausblick

In diesem Fazit soll auf die Erkenntnisse dieser Arbeit eingegangen werden und gleichzeitig ein Ausblick auf kommende, spannende Fragen gegeben werden. Zunächst sollen jedoch die eingangs gestellten Forschungsfragen beantwortet werden.

8.1 Technische Fragestellung

Wie entwickelt man gesamtheitlich einen PNA?

Kann man mit impliziten Feedback Strategien schneller und besser die Interessen eines Nutzers lernen?

In dieser Arbeit wurde versucht, die vorhandenen Signale eines Nutzers besser zu verwenden. Hierzu wurden diverse Strategien vorgestellt, mit denen man teilweise eine signifikant höhere Genauigkeit erzielen konnte als nur mit explizitem Feedback. Besonders positiv ist hierbei die Skip-Strategie aufgefallen, die für übersprungene Artikel einen Zähler hochgesetzt hat. Diese Strategie wurde hier nur in einer sehr einfachen Form implementiert und kann noch stark verfeinert werden. So wurden hier die letzten zehn Artikel vor einem geklickten Element ganzzahlig hochgezählt. Dies könnte zukünftig degressiv gestaltet werden, weil nähere Elemente vermutlich eine höhere Aufmerksamkeit erhalten. Außerdem positiv aufgefallen sind die Predecessor und Successor Strategie, die auf derselben Annahme wie die Skip Strategie beruhen, jedoch nur für ein einziges Element. Bei der Entwicklung eines PNA sollten diese Strategien Verwendung finden.

Können die vorhandenen Informationen aus sozialen Netzwerken oder auf Seiten der Nachrichtenanbieter besser genutzt werden?

Dieser Punkt konnte nicht vollständig nachvollzogen werden. Es wird zwar weiterhin angenommen, dass diese Informationen dazu genutzt werden können, die Abwechslung zu erhöhen und verhindern, dass ein Leser schnell den Tunneleffekt erhält, allerdings führte es beim Ranking zu keinen merklichen Verbesserungen.

Die Startseiteninformationen konnten dazu beitragen, dass die Precision sich verbessert hat, jedoch zu Lasten des Recalls. Im Rahmen dieser Anwendung wird jedoch angenommen, dass dies keine Einschränkung darstellt, da der Nutzer ohnehin nicht an allen relevanten Ergebnissen interessiert ist.

Wie schnell adaptiert ein PNA die Interessen eines Nutzers?

Die Erkenntnis dieser Arbeit ist, dass die Interessen des Nutzers sehr schnell vom System bis zu einem bestimmten Grad adaptiert werden können. So war bereits nach dem ersten Chunk eine Precision von etwa 53% für interessante und stark interessante Artikel vorzufinden. Das erste Chunk entspricht in etwa dem Nutzerfeedback, das bereits nach ein bis zweimal Aufrufen der Übersichtsseite und Lesen der interessanten Artikel generiert werden kann.

Wie treffsicher ist ein PNA?

Der hier vorgestellte PNA empfiehlt Artikel, die nicht uninteressant sind, mit einer Precision von etwa 94%, während die Baseline bei 67% liegt. Artikel, die interessant oder stark interessant sind, werden mit 58% Precision empfohlen, während die Baseline hier bei 37% liegt.

Welcher Algorithmus ist geeignetste?

In dieser Arbeit wurden die Algorithmen Simple TfIdf, kNN und Perceptron angewandt. Wobei der kNN am besten abgeschlossen hat, geringfügig besser als der Simple TfIdf. Der Perceptron wies vor allem bei den stark interessanten Artikeln große Schwächen im Vergleich zu den anderen auf. Das Hinzufügen von weiteren Features hat zu Verschlechterung bei den Algorithmen geführt, wofür eine Begründung Overfitting sein könnte.

8.2 Weiterentwicklung

Das hier vorgestellte System hat aufgrund des hohen Ressourcenbedarfs den Nachteil, dass im schlimmsten Fall eine gesamte Iteration vergeht, bis die Empfehlungen für einen Nutzer aktualisiert werden können. Hinzu kommt noch die Vorlaufzeit des Crawlers von wenigen Minuten. Dadurch verpasst der Nutzer Eilmeldungen und Trends, wodurch er schnell das Gefühl erhält, alte Nachrichten zu lesen. Nachrichten sind so volatil, dass sich dies hier stärker auswirkt, als in anderen Bereichen. Nutzer erwarten, dass sich ihre Aktionen direkt niederschlagen. Vor allem bei neuen Nutzern kann es passieren, dass ihre Personalisierungsaktionen nicht zeitgerecht übernommen werden können, wodurch der Nutzer das Vertrauen in den PNA verlieren könnte.

Zukünftige Arbeiten könnten sich damit beschäftigen, wie man ein Online-Lernsystem erstellt, mit dem es möglich ist, mit wenig Ressourceneinsatz, schnell aber grobgranular

Empfehlungen zu generieren. Die Aufgabe des Offline-Lernsystems wäre es dann, feingranulare Empfehlungen in selteneren Abständen zu erstellen oder das Modell für den Online-Rec recommender zu verbessern.

In dieser Arbeit wurde das *Collaborative Filtering* vernachlässigt, bietet sich jedoch gerade für diesen Bereich an. Hierbei sollte jedoch zunächst ein guter Mechanismus zur Duplikateidentifikation bzw. zum Clustering von Artikelmeldungen erstellt werden, sodass verschiedene Artikel aus mehreren Quellen auf eine einzige Meldung projiziert werden können. Anschließend könnte das Collaborative Filtering auf die Meldungen ausgeführt werden, wodurch das Sparsity Problem gemindert würde.

Ebenso vernachlässigt wurden die Nutzerdaten aus sozialen Netzwerken. Ein großer Trend im Jahre 2011 waren sog. Social Reader, die Facebook Profildaten (einschließlich Likes und Shares eines Nutzers) dafür verwendeten, um ihm interessante Nachrichten anzuzeigen.

Im Rahmen dieser Arbeit wurden textuelle Artikel-Features lediglich aus den RSS Feeds sowie Meta-Informationen der Nachrichtenanbieter gewonnen. Auf ein Auslesen der gesamten Artikelinhalte wurde verzichtet. Vor allem das schlechte Abschneiden des Perceptrons lässt den Wunsch offen, eine zusätzliche Evaluation mit den gesamten Artikelinhalten durchzuführen.

Die Reihung der Artikel wurde hier im Wesentlichen durch die Ergebnisse der Klassifikation bestimmt. Spezialisierte Methoden wie *Learning to Rank* oder *Preference Learning* wurden in diesem Zusammenhang nicht durchgeführt. Es wird jedoch erwartet, dass mit *Instance Ranking* oder *Pairwise Ranking* eine relevantere Reihung der Ergebnislisten generiert werden kann, was unbedingt überprüft werden sollte.

8.3 Wirtschaftliche Fragestellung

Existiert eine Zahlungsbereitschaft für so einen Dienst?

Wird Personalisierung im Allgemeinen bzw. bei Nachrichten vom Nutzer gewünscht?

Die Ergebnisse der Umfrage und die generierte Zahlungsbereitschaft insbesondere bei Testgruppe C (siehe Kapitel 6) lassen den Schluss zu, dass ein derartiger Service von den Nutzern gewünscht ist. Die Zahlungsbereitschaft nimmt nach diesen Erkenntnissen steigender Personalisierung zu, erreicht jedoch auch einen Sättigungspunkt, ab dem die

Nutzer keine weitere Personalisierung wünschen. Auch die Tatsache, dass die Nutzerzufriedenheit für Gruppe C bei 6,25/7 lag, deutet daraufhin, dass es den Nutzern gefallen hat. Es muss zwar nicht so sein, da die Wünsche der Nutzer oftmals mit den tatsächlichen Aktionen divergieren, jedoch zeigen andere Bereiche (Musik, E-Commerce etc.), dass Personalisierung bei Nutzern gerne angenommen und durchaus genutzt wird.

Was ist der optimale Grad der Personalisierung?

In der durchgeführten Umfrage (n=195), wurde wie zuvor beschrieben, ein Anstieg der Zahlungsbereitschaft bis zum einem Sättigungspunkt festgestellt. Dieser Sättigungspunkt wurde bei Gruppe C erreicht. Die Personalisierungsfunktionen dieser Gruppe umfassen das initiale Auswählen der präferierten Quellen, Relevanzfeedbacks mit Daumen sowie feingranularen Tags (siehe Kapitel 6.6). Die Gruppe danach hatte zusätzlich noch die Möglichkeit, Farben, Stile etc. (Customizing) anzupassen. Mit ihr nahm jedoch die Zahlungsbereitschaft leicht ab, was verschiedene Ursachen haben kann. Eine Erklärung wäre es, dass die Nutzer sich überladen mit Funktionen fühlten und nicht recht wussten, was sie tatsächlich verändern sollen. Ausgehend von der Annahme, dass ihre Zahl der aufgewendeten Klicks für eine neue Plattform eine Toleranzschwelle besitzt, waren sie vielleicht zu sehr mit dem Customizing beschäftigt und nicht mit der Anpassung an ihre Interessen. Dies könnte dazu geführt haben, dass die Empfehlungen eher mäßig ausgefallen sind und sie daher die Experimentierfreunde verloren haben. Ein weiterer Grund könnte sein, dass die Plattform durch das Customizing zu spielerisch und zu wenig seriös auf den Nutzer gewirkt hat. Dies wurde durch Oechslein (2013) ebenfalls so festgestellt. Die Konsequenz sollte sein, dass man nicht zu viel Personalisierung bzw. Customizing einführt und sich eher auf die Kernkompetenzen, in diesem Fall, das adäquate Filtern von Nachrichten, beschränkt.

Wie hoch ist die Zahlungsbereitschaft für die Dienstleistung eines PNA?

In der Umfrage (n=195) wurde eine unterschiedliche Zahlungsbereitschaft für die unterschiedlichen Testgruppen festgestellt (siehe auch Kapitel 7.1.2). Diese weicht ungefähr um das dreifache von der ursprünglichen Zahlungsbereitschaft vor der Nutzung des PNA ab. Vorher lag diese relativ gleichverteilt im Durchschnitt über alle Gruppen bei 0,78 Euro, obwohl hier schon Unterschiede bezüglich der Anpassungsfähigkeit des PNA vorgenommen wurden. Danach variierte sie stark zwischen den Gruppen. Für die Zahlungsbereitschaft für die Gruppe mit der niedrigsten Personalisierung wurde nach der

Nutzung 1,16 Euro, für Gruppe B 1,91 Euro, für C und D, 3,98 Euro sowie 3,14 Euro festgestellt.

Welche Aussagen lassen sich über die Zielgruppe machen?

Die Zielgruppe ist im vorliegenden Fall nicht eindeutig zu ermitteln, da die Teilnehmergruppe nicht repräsentativ ist. Jedoch ähnelte die Gruppe stark einer Untergruppe der *Digital Natives*, besonders den *Digital Souveränen*, die einen beträchtlichen Teil der Bevölkerung stellen.¹⁰⁷ Diese Gruppe ist deshalb interessant, weil sie bisher wenig oder gar nicht für digitale Nachrichten zahlt. Für die Print-Wirtschaft, deren Umsätze jährlich schrumpfen, wäre es also erstrebenswert, Angebote für diese Gruppe zu schaffen, da ihnen so ein völlig neuer Kundenkreis geöffnet wird. Mit dieser Arbeit wurde gezeigt, dass innovative Nachrichtenplattformen, wie ein PNA, ein Weg dahin sein könnten.

8.4 Weitere Studien

Diese Arbeit hat einen kurzen Abriss zur Zahlungsbereitschaft für personalisierte Dienste gegeben. In zukünftigen Arbeiten wäre es erstrebenswert zu erfahren, welche Modelle konkret von den Nutzern getragen würden.

In diesem Zusammenhang wäre zunächst die Eignung eines Flatrate Modells in der Art von Spotify, jedoch für Nachrichten, zu prüfen. Hierbei würden die Nutzer einen fixen monatlichen Preis zahlen und dafür so viele Artikel lesen, wie sie möchten. In Anbetracht der Tatsache, dass die Musikindustrie die große Schrumpfung bereits einige Jahre vor der Verlagsindustrie durchgemacht hat, wären solche Modelle womöglich auch auf Anbieterseite potentiell durchführbar.

Mit dem Flatrate Modell könnte außerdem parallel ein Pay-per-Use Ansatz getestet werden, bei dem Nutzer pro Artikel bezahlen und zwar nur für diejenigen, die tatsächlich auch gelesen werden.

Ein großer Trend ist auch die Verschiebung der Internetnutzung in den Mobilbereich. Die implizite und explizite Erfassung der Nutzerdaten wurde hier ausschließlich über die Webanwendung beschrieben. Gänzlich vernachlässigt wurde der Aspekt der mobilen

¹⁰⁷ Vgl. Deutsches Institut für Vertrauen und Sicherheit im Internet (2013)

Smartphone und Tablet Apps. Sie bieten mächtige Frameworks, die weitaus detaillierte Informationen zum Nutzungsverhalten offenbaren, die im Webbereich nur sehr schwer oder gar nicht möglich sind. Zudem kann immer nur eine dieser Apps aktiv im Sichtfeld des Nutzers laufen, wodurch die erhaltenen Signale auch aussagekräftig sind. Dies könnte im Rahmen einer zukünftigen Studie ebenfalls geprüft werden.

Ein letzter Wunsch betrifft die Güte der Personalisierungsalgorithmen. In dieser Arbeit durften die Teilnehmer bereits nach kurzer Nutzungszeit die finale Umfrage durchführen. Interessant wäre es in einer weiteren Arbeit herauszufinden, wie sich die Güte der Personalisierungsalgorithmen auf die Zufriedenheit und die Zahlungsbereitschaft auswirkt. Hiermit könnten auch verschiedene Personalisierungsmechanismen wie Schieberegler genutzt werden, mit denen die Verteilung der Kategorien und Themen bestimmt wird.

Anhang

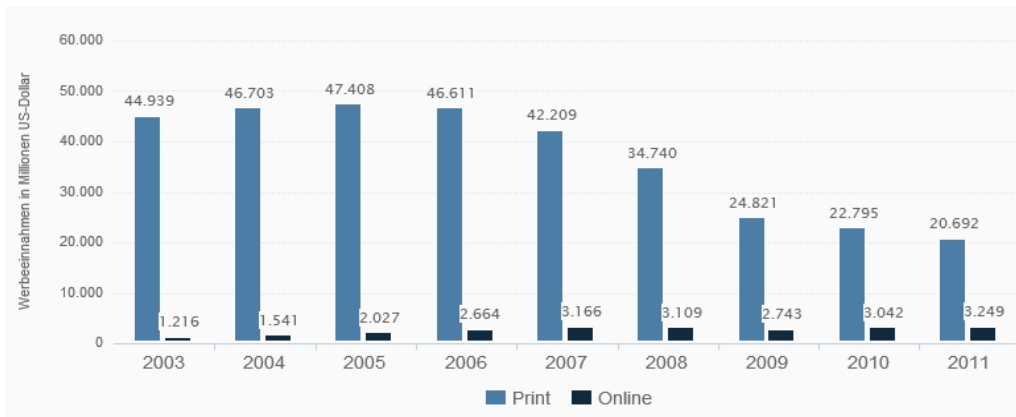
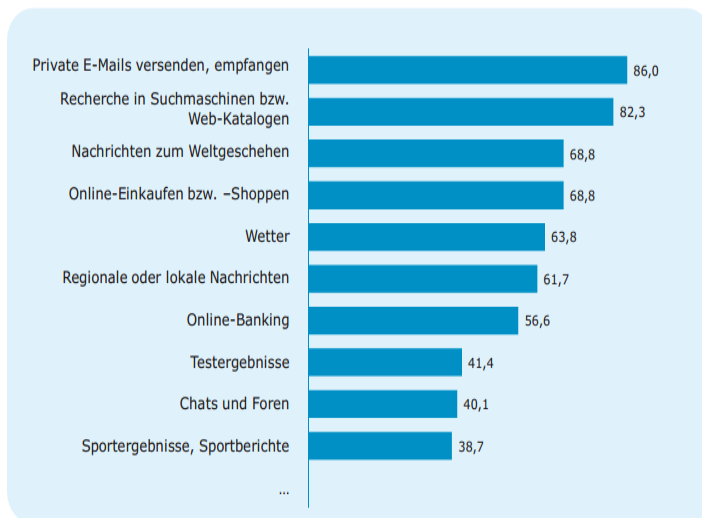


Abbildung 8-1: Newspaper Association of America, Einnahmen von US Zeitungen mit Print- und Onlinewerbung seit 2003

Thematische Schwerpunkte: Top 10

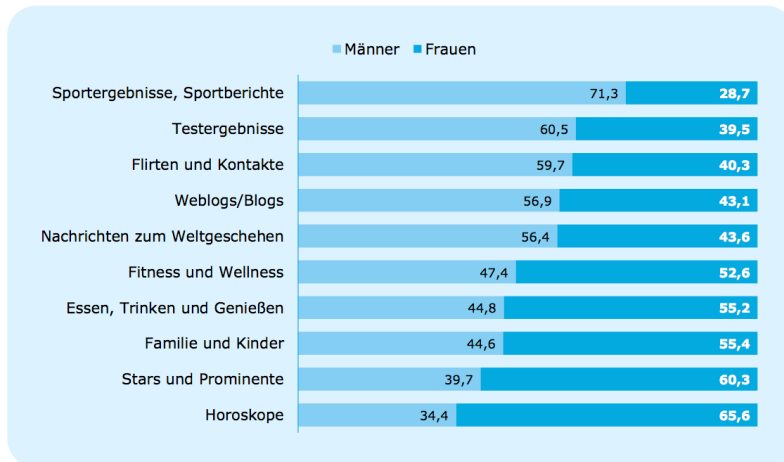


Basis: 101.306 Fälle (Internetnutzer letzte 3 Monate) / „Nutzen Sie diese Themen und Angebote häufig, gelegentlich, selten oder nie?“ / Top Two-Box: häufige oder gelegentliche Nutzung / Darstellung der Top 10 von insgesamt 22 Themen / Angaben in Prozent / Quelle: AGOF e.V. / internet facts 2012-02

Seite 13

Abbildung 8-2: AGOF 2012, Nutzung von Themen und Angeboten des Internets

Top-Themen nach Geschlecht

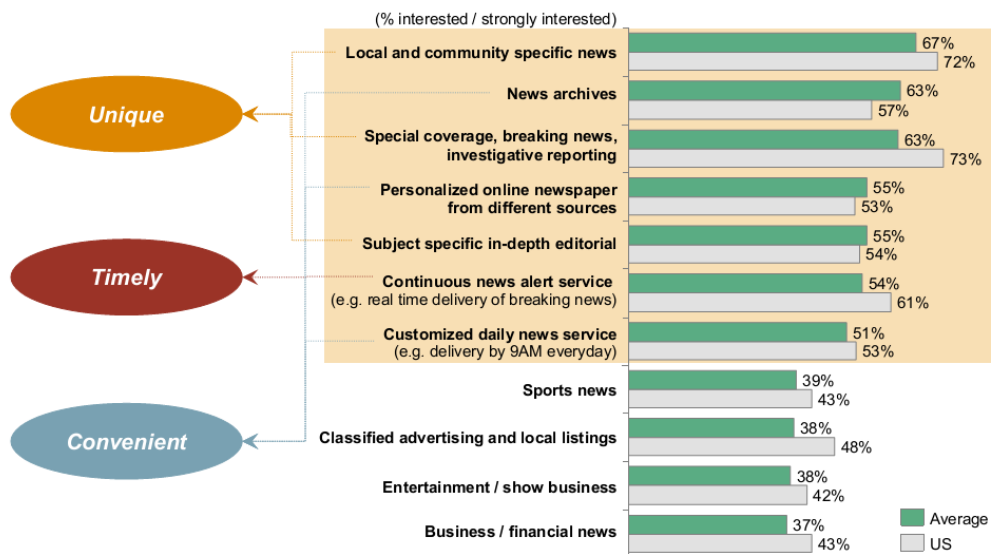


Basis: 101.306 Fälle (Internetnutzer letzte 3 Monate) / „Nutzen Sie diese Themen und Angebote häufig, gelegentlich, selten oder nie?“ / Top Two-Box: häufige oder gelegentliche Nutzung / Darstellung der Top 5 Themen pro Geschlecht / Angaben in Prozent / Quelle: AGOF e.V. / internet facts 2012-02

Seite 14

Abbildung 8-3: AGOF 2012, Top-Themen nach Geschlecht

How interested would you be in accessing each of these type of online news?



Note= Average is the average score (% interested/strongly interested) of 9 countries
Source: BCG Multi-Country Survey on Online Paid Content (Oct. 2009, sample size=5,083), BCG analysis

THE BOSTON CONSULTING GROUP

6

Abbildung 8-4: Boston Consulting Group, Willingness to pay for news online: Key findings from an international survey

63% are willing to pay for online news

How much per month are you willing to spend to get online news?

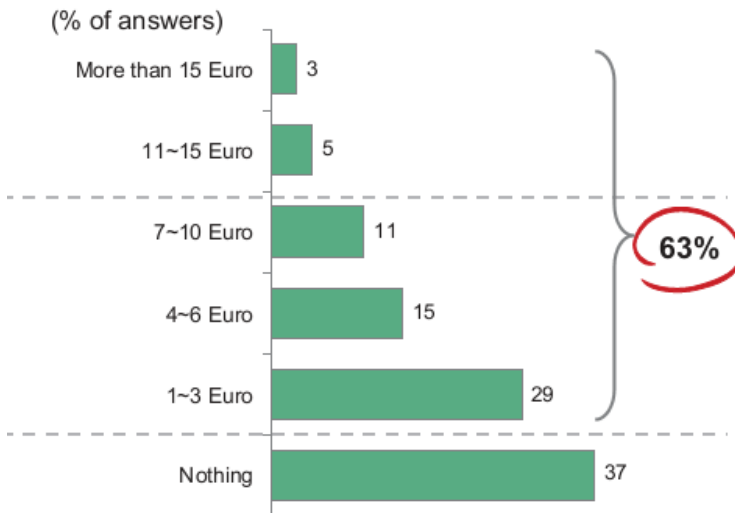


Abbildung 8-5: Boston Consulting Group (BCG), Willingness to pay for news online, German Customers

How much per month are you willing to spend to get online news on your PC or mobile?

Average amount that consumers would be willing to pay for online news (\$)

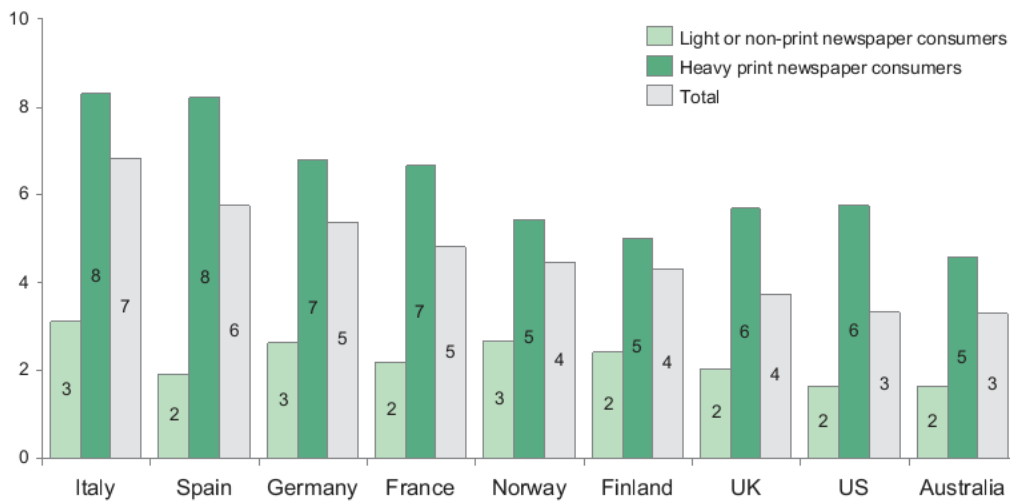


Abbildung 8-6: Boston Consulting Group (BCG), Willingness to pay for news online, European customers



Mediennutzung allgemein: Um mich zu informieren, brauche ich keine Zeitung, mir reichen die Informationen, die ich im Fernsehen und im Internet bekomme					
	Fälle (ungew.)	%	Mio	Anteil %	Index
Gesamt	2.063	100,0	12,53	38	100
Geschlecht					
Männer	1.166	57,3	7,18	39	102
Frauen	897	42,7	5,35	37	98
Alter					
Alter					
40-49 Jahre	333	18,4	2,31	31	81
30-39 Jahre	482	20,5	2,56	41	108
50-59 Jahre	140	10,2	1,28	24	63
25-29 Jahre	289	14,0	1,76	50	130
18-24 Jahre	499	21,3	2,66	51	133
14-17 Jahre	255	10,8	1,35	56	147
60-64 Jahre	35	3,2	0,40	26	68
65-69 Jahre	30	1,7	0,21	20	51
Nettoeinkommen des Hauptverdieners					
2.500 - unter 3.000 Euro	229	10,9	1,37	34	89
2.000 - unter 2.500 Euro	315	14,7	1,84	35	91
1.250 - unter 1.500 Euro	261	12,6	1,57	44	114
1.000 - unter 1.250 Euro	169	8,2	1,02	42	110
1.750 - unter 2.000 Euro	241	12,6	1,58	41	108
3.000 - unter 3.500 Euro	137	6,4	0,81	32	83
1.500 - unter 1.750 Euro	210	11,2	1,41	40	104
750 - unter 1.000 Euro	117	5,1	0,64	41	108
4.000 - unter 5.000 Euro	67	3,0	0,38	31	81
3.500 - unter 4.000 Euro	72	3,5	0,43	34	88
5.000 - unter 6.000 Euro	26	0,9	0,12	19	50
500 - unter 750 Euro	113	5,5	0,69	55	143
Unter 500 Euro	89	4,4	0,55	55	142
7.500 - unter 10.000 Euro	4	0,3	0,04	34	88
6.000 - unter 7.500 Euro	10	0,4	0,05	26	67
10.000 Euro und mehr	3	0,1	0,01	8	21
Nutzung kostenpflichtiger Inhalte von Zeitungen und Zeitschriften im Internet					
Käme eher nicht in Frage	1.667	80,9	10,14	39	102
Käme in Frage	180	8,3	1,05	36	94
Nutze es bereits	28	1,1	0,14	20	53
Einstellung zum Lesen am Bildschirm					
Lieber auf Papier	641	31,0	3,88	28	73
Genauso gerne wie auf Papier	937	45,1	5,66	48	125
Unentschieden	455	21,9	2,75	42	109

Abbildung 8-7: Allensbachstudie, ACTA 2012



Abbildung 8-8: PWC, WAN. Moving into Multiple Business Models: Outlook for Newspapers Publishing in the Digital Age, 2009



Alter

Geschlecht Männlich
 Weiblich

Welchen höchsten allgemeinen Schulabschluss hast du?

- Schüler, besuche eine allgemeinbildende Vollzeitschule
- Von der Schule abgegangen ohne Hauptschulabschluss (Volksschulabschluss)
- Hauptschulabschluss (Volksschulabschluss)
- Realschulabschluss (Mittlere Reife)
- Fachhochschulreife, Abschluss einer Fachoberschule
- Allgemeine oder fachgebundene Hochschulreife
- Abitur über den zweiten Bildungsweg nachgeholt
- Einen anderen Schulabschluss, und zwar:

Was ist deine aktuelle Tätigkeit?

- Schüler
- Student
- Auszubildender
- Arbeitnehmer (Vollzeit)
- Arbeitnehmer (Teilzeit)
- Selbständig
- Arbeitssuchend
- Hausfrau/Hausmann
- Rentner/Pensionär

Wie viel wärst du generell bereit, für elektronische Nachrichten im Monat zu bezahlen?

Euro Cent

Abbildung 8-9: Fragebogen bei Registrierungsprozess

Anzahl der Beobachtungen = 190

R-quadrariert = 0,521

Root MSE = 1,17

Adj R-quadriert = 0,513

Variations- quelle	Quadrat- summe	Freiheits- grade	Mittlere Quadrat-	F	Wahrschein- lichkeit > F
-----------------------	-------------------	---------------------	----------------------	---	-----------------------------

	summe				
Modell	276,71	3	92,24	67,34	0,0000
Gruppe	276,71	3	92,24	67,34	0,0000
Residuen	254,79	186	1,37		
Total	531,50	189	2,81		

Tabelle 4: ANOVA zu Zahlungsbereitschaft und Gruppen

Reihendurchschnitt/ Zeilendurchschnitt	1	2	3
2	1,045 0,000		
3	3,226 0,000	2,181 0,000	
4	2,276 0,000	1,231 0,000	-0,95 0,002

Tabelle 5: Bonferroni-Test der Zahlungsbereitschaft nach Gruppen

Grundlegende Statistiken, mit Beobachtungen 1 - 195
(Fehlerte wurden ausgelassen)

Variable	arith. Mittel	Median	Minimum	Maximum
zufriedenheit1	4,22564	5,00000	1,00000	7,00000
zufriedenheit2	4,10256	5,00000	1,00000	7,00000
zufriedenheit3	4,17436	5,00000	1,00000	7,00000
preferenceit1	4,22564	5,00000	1,00000	7,00000
preferenceit2	3,98974	4,00000	1,00000	7,00000
preferenceit3	4,08718	5,00000	1,00000	7,00000
interesse	4,74872	5,00000	1,00000	7,00000
purchaseintenti	3,83590	4,00000	1,00000	7,00000
involvement1	4,65128	5,00000	2,00000	7,00000
involvement2	4,68718	5,00000	1,00000	7,00000
involvement3	4,63077	5,00000	1,00000	7,00000
involvement4	4,40000	5,00000	1,00000	7,00000
involvement5	4,53846	4,00000	1,00000	7,00000
incomeclass	2,54872	2,00000	1,00000	6,00000
processeffort1	3,10000	3,00000	1,00000	7,00000
processeffort2	5,38621	6,00000	1,00000	7,00000
processeffort3	4,31034	5,00000	1,00000	7,00000

processeffort4	4,30345	5,00000	1,00000	7,00000
perceivedenjoyment	4,98621	5,00000	1,00000	7,00000
perceiva	5,00000	5,00000	1,00000	7,00000
psychownership1	6,38000	7,00000	4,00000	7,00000
psychownership2	5,01379	6,00000	1,00000	7,00000
psychownership3	5,05517	6,00000	1,00000	7,00000
prefinsight1	4,63590	5,00000	1,00000	7,00000
prefinsight2	4,52308	5,00000	1,00000	7,00000
prefinsight3	4,61538	5,00000	1,00000	7,00000
satisfaction	4,16752	5,00000	1,00000	7,00000
wtp	2,48041	2,50000	0,000000	5,99000
wtp	9,73303	10,0000	0,000000	50,0000
sex	0,310526	0,000000	0,000000	1,00000
job	1,57368	1,00000	0,000000	5,00000
education	5,10000	5,00000	4,00000	6,00000
age	25,0053	24,0000	19,0000	40,0000
priorwtp	0,780526	0,150000	0,000000	10,0000
Variable	Std. Abw.	Var'koeff.	Schiefe	èberwùlung
zufriedenheit1	2,01550	0,476968	-0,204323	-1,45995
zufriedenheit2	2,13216	0,519715	-0,130931	-1,41634
zufriedenheit3	2,20598	0,528460	-0,192974	-1,50323
preferenceit1	1,96630	0,465327	-0,182166	-1,42220
preferenceit2	2,00769	0,503213	-0,108522	-1,39835
preferenceit3	2,11952	0,518577	-0,179687	-1,43170
interesse	1,34449	0,283128	-0,211662	-0,454613
purchaseintenti	2,04429	0,532936	0,128240	-1,35293
involvement1	1,29290	0,277966	-0,0633119	-0,619891
involvement2	1,45361	0,310124	-0,162096	-0,597898
involvement3	1,36112	0,293930	-0,147683	-0,636158
involvement4	1,28993	0,293166	-0,312626	0,0669767
involvement5	1,28925	0,284073	0,00881865	-0,393730
incomeclass	1,22324	0,479945	1,06422	0,914503
processeffort1	1,32865	0,428596	1,02836	1,32360
processeffort2	1,35496	0,251561	-1,06074	0,898481
processeffort3	1,95618	0,453833	-0,0487492	-1,44215
processeffort4	2,01840	0,469020	-0,103478	-1,46687
perceivedenjoyment	1,61153	0,323198	-0,596958	-0,617682
perceiva	1,77951	0,355903	-0,519381	-0,841066
psychownership1	0,779586	0,122192	-1,29196	1,47027
psychownership2	1,73600	0,346245	-0,732521	-0,619535
psychownership3	1,81345	0,358732	-0,671186	-0,879152
prefinsight1	1,32608	0,286047	-0,347823	-0,534262
prefinsight2	1,40442	0,310502	-0,0641179	-0,558623
prefinsight3	1,33578	0,289418	-0,0531162	-0,575476
satisfaction	2,06547	0,495612	-0,192305	-1,56328
wtp	1,67656	0,675920	0,206663	-1,12184
wtp	6,44999	0,662691	1,93952	8,22450
sex	0,463932	1,49402	0,818975	-1,32928
job	1,09464	0,695593	1,63140	1,53294
education	0,364423	0,0714555	1,23645	3,32207
age	4,06136	0,162420	1,41628	1,70007
priorwtp	1,52759	1,95712	2,87034	9,15115

Abbildung 8-10: Statistische Auswertung über alle Teilnehmer nach der Umfrage.

Grundlegende Statistiken, mit Beobachtungen 1 – 50 (Fehlwerte wurden ausgelassen)

Variable	arith. Mittel	Median	Minimum	Maximum
zufriedenheit1	2,00000	2,00000	1,00000	7,00000
zufriedenheit2	1,92000	1,50000	1,00000	7,00000
zufriedenheit3	1,82000	1,00000	1,00000	7,00000
preferencefit1	2,02000	2,00000	1,00000	6,00000
preferencefit2	1,80000	2,00000	1,00000	6,00000
preferencefit3	1,70000	1,00000	1,00000	6,00000
interesse	4,24000	4,00000	2,00000	7,00000
purchaseintenti	2,38000	2,00000	1,00000	7,00000
involvement1	4,34000	4,00000	2,00000	7,00000
involvement2	4,24000	4,00000	1,00000	7,00000
involvement3	4,24000	4,00000	1,00000	7,00000
involvement4	4,20000	4,00000	2,00000	7,00000
involvement5	4,24000	4,00000	2,00000	7,00000
incomeclass	2,38000	2,00000	1,00000	6,00000
processeffort1	3,10000	3,00000	1,00000	7,00000
psychownership1	6,38000	7,00000	4,00000	7,00000
prefinsight1	4,34000	4,50000	2,00000	7,00000
prefinsight2	4,02000	4,00000	1,00000	7,00000
prefinsight3	4,22000	4,00000	1,00000	7,00000
satisfaction	1,91333	1,33333	1,00000	7,00000
wtp	0,986200	0,525000	0,000000	5,00000
wtp	7,70200	5,00000	0,000000	25,0000
sex	0,319149	0,000000	0,000000	1,00000
job	1,65957	1,00000	0,000000	5,00000
education	5,21277	5,00000	5,00000	6,00000
age	25,0426	23,0000	20,0000	39,0000
priorwtp	0,617234	0,250000	0,000000	5,00000
Variable	Std. Abw.	Var'koeff.	Schiefe	ëberwölbung
zufriedenheit1	1,22890	0,614452	2,33269	6,11249
zufriedenheit2	1,33768	0,696708	2,26571	5,98057
zufriedenheit3	1,35059	0,742080	2,48997	6,53720
preferencefit1	1,02000	0,504952	1,70835	4,07672
preferencefit2	0,968904	0,538280	1,90383	5,41399
preferencefit3	1,05463	0,620371	1,88753	4,10454
interesse	1,25454	0,295883	-0,0231536	-0,820004
purchaseintenti	1,49680	0,628907	1,28549	1,09488
involvement1	1,25536	0,289253	0,211229	-0,574981
involvement2	1,40785	0,332040	0,187523	-0,615096
involvement3	1,45069	0,342143	-0,100868	-0,796936
involvement4	1,19523	0,284578	-0,0289767	-0,581224
involvement5	1,25454	0,295883	0,290071	-0,557708
incomeclass	1,25990	0,529370	1,17088	1,16478
processeffort1	1,32865	0,428596	1,02836	1,32360
psychownership1	0,779586	0,122192	-1,29196	1,47027
prefinsight1	1,30321	0,300280	-0,203511	-0,831760
prefinsight2	1,43555	0,357103	0,0483661	-0,625751
prefinsight3	1,46092	0,346189	0,00955256	-0,681071
satisfaction	1,23775	0,646906	2,75369	7,97102
wtp	1,14691	1,16296	2,24902	4,69906
wtp	5,49568	0,713539	1,18392	0,939915
sex	0,471186	1,47638	0,775940	-1,39792
job	1,20283	0,724784	1,36147	0,734205
education	0,413688	0,0793606	1,40366	-0,0297297
age	4,72255	0,188581	1,42142	1,19419
priorwtp	1,17347	1,90117	2,84572	7,34950

Abbildung 8-11: Statistische Auswertung über Gruppe A nach der Umfrage.

Grundlegende Statistiken, mit Beobachtungen 1 – 49 (Fehlwerte wurden ausgelassen)

Variable	arith. Mittel	Median	Minimum	Maximum
zufriedenheit1	3,18367	3,00000	1,00000	7,00000
zufriedenheit2	2,91837	3,00000	1,00000	7,00000
zufriedenheit3	3,02041	3,00000	1,00000	7,00000
preferencefit1	3,30612	3,00000	2,00000	6,00000
preferencefit2	3,02041	3,00000	1,00000	6,00000
preferencefit3	3,12245	3,00000	1,00000	7,00000
interesse	4,63265	5,00000	1,00000	7,00000
purchaseintenti	2,89796	3,00000	1,00000	7,00000
involvement1	4,42857	4,00000	2,00000	7,00000
involvement2	4,57143	5,00000	1,00000	7,00000
involvement3	4,32653	4,00000	2,00000	7,00000
involvement4	4,14286	4,00000	1,00000	7,00000
involvement5	4,28571	4,00000	2,00000	7,00000
incomeclass	2,38776	2,00000	1,00000	6,00000
processeffort2	4,48980	4,00000	1,00000	7,00000
processeffort3	2,42857	2,00000	1,00000	7,00000
processeffort4	2,42857	2,00000	1,00000	6,00000
perceivedenjoyment	3,48980	3,00000	1,00000	7,00000
perceiva	3,28571	3,00000	1,00000	7,00000
psychownership2	3,26531	3,00000	1,00000	7,00000
psychownership3	3,18367	3,00000	1,00000	7,00000
prefinsight1	4,55102	5,00000	2,00000	7,00000
prefinsight2	4,28571	4,00000	1,00000	7,00000
prefinsight3	4,26531	4,00000	2,00000	7,00000
satisfaction	3,04082	2,66667	1,33333	7,00000
wtp	1,90816	1,99000	0,000000	5,00000
wtp	8,18490	8,00000	0,000000	20,0000
sex	0,326531	0,000000	0,000000	1,00000
job	1,53061	1,00000	1,00000	5,00000
education	5,06122	5,00000	4,00000	6,00000
age	25,3673	24,0000	19,0000	40,0000
priorwtp	1,03776	0,150000	0,000000	6,29000
Variable	Std. Abw.	Var'koeff.	Schiefe	èberwùlbung
zufriedenheit1	1,34897	0,423716	1,10244	0,858652
zufriedenheit2	1,42649	0,488796	0,797288	0,317959
zufriedenheit3	1,49290	0,494270	0,724039	-0,116893
preferencefit1	1,32609	0,401100	0,668656	-0,782685
preferencefit2	1,34613	0,445679	0,428651	-0,676589
preferencefit3	1,49489	0,478755	0,545077	-0,513050
interesse	1,42440	0,307469	-0,380533	-0,295460
purchaseintenti	1,37303	0,473791	0,672950	0,0106957
involvement1	1,24164	0,280370	0,0659784	-0,302045
involvement2	1,48605	0,325073	-0,274892	-0,666607
involvement3	1,23132	0,284598	0,0338675	-0,752161
involvement4	1,27475	0,307699	-0,209036	0,0700197
involvement5	1,17260	0,273608	0,212613	-0,0895317
incomeclass	1,25526	0,525708	1,15109	1,02516
processeffort2	1,55620	0,346609	-0,349445	-0,489108
processeffort3	1,13652	0,467977	2,19997	5,59157
processeffort4	1,09924	0,452629	1,56209	2,94946
perceivedenjoyment	1,27675	0,365854	0,356717	-0,299168
perceiva	1,27475	0,387969	0,487750	0,677515
psychownership2	1,51073	0,462662	0,640343	-0,537316
psychownership3	1,53668	0,482676	0,906307	0,0710509
prefinsight1	1,24267	0,273052	-0,0847563	-0,730383
prefinsight2	1,17260	0,273608	-0,0223803	0,546832

prefinsight3	1,15064	0,269768	0,545821	0,0330814
satisfaction	1,32048	0,434252	0,952526	0,527906
wtp	1,16513	0,610603	1,30780	1,74576
wtp	4,92829	0,602120	0,481427	-0,134319
sex	0,473804	1,45102	0,739830	-1,45265
job	1,00212	0,654721	1,67214	1,78694
education	0,316765	0,0625867	1,39346	6,24296
age	4,10637	0,161876	1,54477	2,28312
priorwtp	1,76686	1,70258	1,72345	1,53576

Abbildung 8-12: Statistische Auswertung über Gruppe B nach der Umfrage.

Grundlegende Statistiken, mit Beobachtungen 1 - 47
(Fehlwerte wurden ausgelassen)

Variable	arith. Mittel	Median	Minimum	Maximum
zufriedenheit1	6,10638	6,00000	4,00000	7,00000
zufriedenheit2	6,27660	6,00000	3,00000	7,00000
zufriedenheit3	6,36170	7,00000	3,00000	7,00000
preferencefit1	6,04255	6,00000	2,00000	7,00000
preferencefit2	5,89362	6,00000	2,00000	7,00000
preferencefit3	6,19149	6,00000	3,00000	7,00000
interesse	5,06383	5,00000	2,00000	7,00000
purchaseintenti	5,48936	6,00000	1,00000	7,00000
involvement1	4,95745	5,00000	2,00000	7,00000
involvement2	4,97872	5,00000	1,00000	7,00000
involvement3	5,02128	5,00000	2,00000	7,00000
involvement4	4,72340	5,00000	2,00000	7,00000
involvement5	4,91489	5,00000	2,00000	7,00000
incomeclass	2,76596	2,00000	1,00000	6,00000
processeffort2	5,95745	6,00000	3,00000	7,00000
processeffort3	4,95745	5,00000	2,00000	7,00000
processeffort4	5,04255	5,00000	1,00000	7,00000
perceivedenjoym	6,12766	7,00000	2,00000	7,00000
perceiva	6,25532	7,00000	2,00000	7,00000
psychownership2	6,14894	6,00000	4,00000	7,00000
psychownership3	6,19149	6,00000	3,00000	7,00000
prefinsight1	4,85106	5,00000	1,00000	7,00000
prefinsight2	5,02128	5,00000	2,00000	7,00000
prefinsight3	4,91489	5,00000	2,00000	7,00000
satisfaction	6,24823	6,33333	3,33333	7,00000
wtp	3,97936	4,00000	0,000000	5,99000
wtp	11,8296	10,0000	0,000000	50,0000
sex	0,355556	0,000000	0,000000	1,00000
job	1,46667	1,00000	1,00000	5,00000
education	5,04444	5,00000	4,00000	6,00000
age	24,5682	24,0000	21,0000	33,0000
priorwtp	0,706000	0,150000	0,000000	5,00000
Variable	Std. Abw.	Var'koeff.	Schiefe	èberwùlung
zufriedenheit1	0,786684	0,128830	-0,728573	0,359572
zufriedenheit2	0,877301	0,139773	-1,54167	2,95760
zufriedenheit3	0,845076	0,132838	-1,63377	3,50340
preferencefit1	1,08262	0,179166	-1,64320	3,36351
preferencefit2	1,18384	0,200868	-1,62070	2,93186
preferencefit3	1,05580	0,170524	-1,73243	2,87616
interesse	1,37373	0,271283	-0,420721	-0,295965
purchaseintenti	1,93225	0,351999	-1,08023	-0,125894
involvement1	1,36664	0,275675	-0,387645	-0,444120
involvement2	1,55325	0,311977	-0,386526	-0,400905

involvement3	1,42172	0,283138	-0,267139	-0,848609
involvement4	1,31384	0,278156	-0,233670	-0,259826
involvement5	1,34859	0,274388	-0,0592313	-0,702615
incomeclass	1,08773	0,393257	1,39696	1,68971
processeffort2	0,931512	0,156361	-1,05756	1,15793
processeffort3	1,21508	0,245101	-1,09416	0,954248
processeffort4	1,38246	0,274159	-1,27435	1,19055
perceivedenjoym	1,24441	0,203081	-1,81739	3,13547
perceiva	1,29326	0,206746	-2,06952	3,62026
psychownership2	0,859189	0,139730	-0,704264	-0,286746
psychownership3	1,05580	0,170524	-1,50838	1,85383
prefinsight1	1,47411	0,303873	-0,769578	-0,0231144
prefinsight2	1,34309	0,267479	-0,0933917	-0,872680
prefinsight3	1,36461	0,277648	-0,260291	-0,847608
satisfaction	0,723879	0,115853	-2,01055	4,47221
wtp	1,06746	0,268249	-1,15540	2,82508
wtp	7,54937	0,638177	2,79513	12,1792
sex	0,484090	1,36150	0,603510	-1,63578
job	1,05744	0,720983	2,01277	2,63821
education	0,298142	0,0591031	1,27885	7,85912
age	3,31591	0,134968	1,11304	0,255305
priorwtp	1,38303	1,95896	2,36950	4,34749

Abbildung 8-13: Statistische Auswertung über Gruppe C nach der Umfrage.

Grundlegende Statistiken, mit Beobachtungen 1 - 49
(Fehlerte wurden ausgelassen)

Variable	arith. Mittel	Median	Minimum	Maximum
zufriedenheit1	5,73469	6,00000	4,00000	7,00000
zufriedenheit2	5,42857	5,00000	1,00000	7,00000
zufriedenheit3	5,63265	6,00000	1,00000	7,00000
preferencefit1	5,65306	6,00000	4,00000	7,00000
preferencefit2	5,36735	5,00000	2,00000	7,00000
preferencefit3	5,46939	5,00000	3,00000	7,00000
interesse	5,08163	5,00000	3,00000	7,00000
purchaseintenti	4,67347	5,00000	1,00000	7,00000
involvement1	4,89796	5,00000	2,00000	7,00000
involvement2	4,97959	5,00000	2,00000	7,00000
involvement3	4,95918	5,00000	2,00000	7,00000
involvement4	4,55102	5,00000	1,00000	7,00000
involvement5	4,73469	5,00000	1,00000	7,00000
incomeclass	2,67347	2,00000	1,00000	6,00000
processeffort2	5,73469	6,00000	2,00000	7,00000
processeffort3	5,57143	6,00000	2,00000	7,00000
processeffort4	5,46939	6,00000	1,00000	7,00000
perceivedenjoym	5,38776	6,00000	1,00000	7,00000
perceiva	5,51020	6,00000	1,00000	7,00000
psychownership2	5,67347	6,00000	1,00000	7,00000
psychownership3	5,83673	6,00000	2,00000	7,00000
prefinsight1	4,81633	5,00000	2,00000	7,00000
prefinsight2	4,79592	5,00000	2,00000	7,00000
prefinsight3	5,08163	5,00000	2,00000	7,00000
satisfaction	5,59864	5,66667	2,00000	7,00000
wtp	3,13959	3,00000	0,000000	5,99000
wtp	11,3427	10,0000	0,000000	35,0000
sex	0,244898	0,000000	0,000000	1,00000

job	1,63265	1,00000	1,00000	5,00000
education	5,08163	5,00000	4,00000	6,00000
age	25,0000	24,0000	20,0000	37,0000
priorwtp	0,748367	0,150000	0,000000	10,0000
Variable	Std. Abw.	Var'koeff.	Schiefe	Überwülbung
zufriedenheit1	0,638211	0,111289	-0,204373	0,0537502
zufriedenheit2	0,978945	0,180332	-1,34621	6,94612
zufriedenheit3	0,950743	0,168791	-2,14943	9,62499
preferencefit1	0,663043	0,117289	0,0729041	-0,309405
preferencefit2	0,858630	0,159973	-0,779021	3,52972
preferencefit3	0,738863	0,135091	-0,363577	1,35742
interesse	1,16970	0,230182	0,156386	-0,960821
purchaseintenti	1,62516	0,347742	-1,02054	-0,0532714
involvement1	1,22890	0,250901	-0,212741	-0,642582
involvement2	1,26639	0,254316	-0,0858832	-0,584413
involvement3	1,17188	0,236305	-0,155972	-0,255311
involvement4	1,32384	0,290888	-0,873302	1,21958
involvement5	1,28737	0,271901	-0,560066	0,444951
incomeclass	1,26471	0,473059	0,883581	0,385297
processeffort2	0,995312	0,173560	-1,24327	3,08673
processeffort3	1,76777	0,317292	-0,953668	-0,423733
processeffort4	1,91596	0,350307	-0,976218	-0,480976
perceivedenjoyment	0,975028	0,180971	-2,06508	6,50133
perceiva	1,20973	0,219543	-1,34385	3,28228
psychownership2	1,10657	0,195042	-1,75104	5,15048
psychownership3	0,986301	0,168982	-1,51192	3,91048
prefinsight1	1,25289	0,260133	-0,290889	-0,568412
prefinsight2	1,45745	0,303893	-0,252655	-0,906728
prefinsight3	1,15175	0,226650	-0,160093	-0,181008
satisfaction	0,716795	0,128030	-2,32364	11,2431
wtp	1,50486	0,479317	-0,349996	0,0339038
wtp	6,68675	0,589523	1,49597	3,57414
sex	0,434483	1,77414	1,18645	-0,592342
job	1,13089	0,692670	1,54388	1,28150
education	0,399830	0,0786814	0,689626	2,83120
age	4,02596	0,161038	1,22900	1,17511
priorwtp	1,70103	2,27298	4,02904	17,6318

Abbildung 8-14: Statistische Auswertung über Gruppe D nach der Umfrage.

Rating > 0	Chunk	Micro-averaged			Macro-averaged			nDCG
		Precision	Recall	F1	Precision	Recall	F1	
ONLY_EXPLICIT	35	0,7949	0,6086	0,6894	0,8336	0,6077	0,6664	0,5937
EXPLICIT_CLICK	35	0,7378	0,8108	0,7725	0,7571	0,8100	0,7641	0,6564
EXPLICIT_CLICK_SKIP	35	0,9535	0,3527	0,5149	0,9566	0,3551	0,5070	0,5519
EXPLICIT_CLICK_TOP	35	0,8065	0,7441	0,7740	0,8327	0,7385	0,7652	0,6526
EXPLICIT_CLICK_PREDECESSOR	35	0,9524	0,3441	0,5055	0,9547	0,3471	0,4981	0,5375
EXPLICIT_CLICK_SUCESSOR	35	0,9521	0,3419	0,5032	0,9544	0,3445	0,4957	0,5313
EXPLICIT_CLICK_SKIP_TOP_PREDECESSOR_SUCESSOR	35	0,9072	0,4624	0,6125	0,9161	0,4703	0,6035	0,5748
EXPLICIT_CLICK_TOP_PREDECESSOR	35	0,9064	0,5204	0,6612	0,9098	0,5280	0,6540	0,5876
EXPLICIT_CLICK_PREDECESSOR_SUCESSOR_TOP	35	0,9091	0,4731	0,6223	0,9179	0,4818	0,6134	0,5782
ONLY_EXPLICIT	70	0,7869	0,5427	0,6424	0,8477	0,5501	0,6291	0,6233
EXPLICIT_CLICK	70	0,7835	0,7891	0,7863	0,7937	0,7874	0,7830	0,6525
EXPLICIT_CLICK_SKIP	70	0,9371	0,3886	0,5494	0,9502	0,3863	0,5381	0,5320
EXPLICIT_CLICK_TOP	70	0,7995	0,7180	0,7566	0,8231	0,7192	0,7543	0,6494
EXPLICIT_CLICK_PREDECESSOR	70	0,9360	0,3815	0,5421	0,9497	0,3799	0,5311	0,5323
EXPLICIT_CLICK_SUCESSOR	70	0,9360	0,3815	0,5421	0,9499	0,3793	0,5305	0,5272
EXPLICIT_CLICK_SKIP_TOP_PREDECESSOR_SUCESSOR	70	0,9040	0,4242	0,5774	0,9132	0,4315	0,5739	0,5561
EXPLICIT_CLICK_TOP_PREDECESSOR	70	0,9056	0,5000	0,6443	0,9084	0,5074	0,6435	0,5763
EXPLICIT_CLICK_PREDECESSOR_SUCESSOR_TOP	70	0,9069	0,4384	0,5911	0,9161	0,4462	0,5878	0,5593
ONLY_EXPLICIT	105	0,7959	0,5639	0,6601	0,8422	0,5667	0,6473	0,5930
EXPLICIT_CLICK	105	0,7287	0,7639	0,7459	0,7420	0,7681	0,7432	0,6568
EXPLICIT_CLICK_SKIP	105	0,9524	0,3373	0,4982	0,9513	0,3397	0,4965	0,5080
EXPLICIT_CLICK_TOP	105	0,7885	0,7277	0,7569	0,8025	0,7247	0,7515	0,6226
EXPLICIT_CLICK_PREDECESSOR	105	0,9075	0,3783	0,5340	0,9271	0,3746	0,5218	0,5262
EXPLICIT_CLICK_SUCESSOR	105	0,9514	0,3301	0,4902	0,9510	0,3320	0,4891	0,5058
EXPLICIT_CLICK_SKIP_TOP_PREDECESSOR_SUCESSOR	105	0,9031	0,4265	0,5794	0,9095	0,4342	0,5792	0,5648
EXPLICIT_CLICK_TOP_PREDECESSOR	105	0,9032	0,4723	0,6203	0,9083	0,4806	0,6189	0,5682
EXPLICIT_CLICK_PREDECESSOR_SUCESSOR_TOP	105	0,9055	0,4386	0,5909	0,9124	0,4463	0,5905	0,5662

Abbildung 8-15: Implizites Feedback, Bewertung größer 0

Rating > 1	Chunk	Micro-averaged			Macro-averaged			nDCG
		Precision	Recall	F1	Precision	Recall	F1	
ONLY_EXPLICIT	35	0,4579	0,6367	0,5327	0,5111	0,6475	0,5314	0,5937
EXPLICIT_CLICK	35	0,4207	0,8398	0,5606	0,4363	0,8420	0,5555	0,6564
EXPLICIT_CLICK_SKIP	35	0,5872	0,3945	0,4720	0,6029	0,4123	0,4664	0,5519
EXPLICIT_CLICK_TOP	35	0,4709	0,7891	0,5898	0,4855	0,7840	0,5839	0,6526
EXPLICIT_CLICK_PREDECESSOR	35	0,5833	0,3828	0,4623	0,5989	0,3982	0,4561	0,5375
EXPLICIT_CLICK_SUCESSOR	35	0,5868	0,3828	0,4634	0,6007	0,4016	0,4581	0,5313
EXPLICIT_CLICK_TOP_PREDECESSOR_SUCESSOR	35	0,5401	0,5000	0,5193	0,5515	0,5240	0,5117	0,5748
EXPLICIT_CLICK_TOP_PREDECESSOR	35	0,5281	0,5508	0,5392	0,5369	0,5687	0,5318	0,5876
EXPLICIT_CLICK_PREDECESSOR_SUCESSOR_TOP	35	0,5331	0,5039	0,5181	0,5443	0,5290	0,5102	0,5782
ONLY_EXPLICIT	70	0,4811	0,6061	0,5364	0,5237	0,6132	0,5297	0,6233
EXPLICIT_CLICK	70	0,4424	0,8139	0,5732	0,4481	0,8110	0,5678	0,6525
EXPLICIT_CLICK_SKIP	70	0,5771	0,4372	0,4975	0,5703	0,4274	0,4787	0,5320
EXPLICIT_CLICK_TOP	70	0,4723	0,7749	0,5869	0,4738	0,7661	0,5783	0,6494
EXPLICIT_CLICK_PREDECESSOR	70	0,5814	0,4329	0,4963	0,5770	0,4238	0,4785	0,5323
EXPLICIT_CLICK_SUCESSOR	70	0,5814	0,4329	0,4963	0,5767	0,4238	0,4787	0,5272
EXPLICIT_CLICK_SKIP_TOP_PREDECESSOR_SUCESSOR	70	0,5303	0,4545	0,4895	0,5440	0,4723	0,4866	0,5561
EXPLICIT_CLICK_TOP_PREDECESSOR	70	0,5193	0,5238	0,5216	0,5252	0,5406	0,5198	0,5763
EXPLICIT_CLICK_PREDECESSOR_SUCESSOR_TOP	70	0,5245	0,4632	0,4920	0,5371	0,4810	0,4886	0,5593
ONLY_EXPLICIT	105	0,4762	0,6306	0,5426	0,4979	0,6236	0,5328	0,5930
EXPLICIT_CLICK	105	0,4161	0,8153	0,5510	0,4256	0,8144	0,5501	0,6568
EXPLICIT_CLICK_SKIP	105	0,5646	0,3739	0,4499	0,5628	0,3740	0,4444	0,5080
EXPLICIT_CLICK_TOP	105	0,4334	0,7477	0,5488	0,4432	0,7435	0,5459	0,6226
EXPLICIT_CLICK_PREDECESSOR	105	0,5491	0,4279	0,4810	0,5521	0,4177	0,4637	0,5262
EXPLICIT_CLICK_SUCESSOR	105	0,5694	0,3694	0,4481	0,5687	0,3684	0,4437	0,5058
EXPLICIT_CLICK_SKIP_TOP_PREDECESSOR_SUCESSOR	105	0,5459	0,4820	0,5120	0,5612	0,4905	0,5114	0,5648
EXPLICIT_CLICK_TOP_PREDECESSOR	105	0,5346	0,5225	0,5285	0,5488	0,5321	0,5271	0,5682
EXPLICIT_CLICK_PREDECESSOR_SUCESSOR_TOP	105	0,5373	0,4865	0,5106	0,5530	0,4949	0,5101	0,5662

Abbildung 8-16: Implizites Feedback, Bewertung größer 1

Rating > 2	Chunk	Micro-averaged			Macro-averaged			nDCG
		Precision	Recall	F1	Precision	Recall	F1	
ONLY_EXPLICIT	35	0,2219	0,6695	0,3333	0,2492	0,6964	0,3395	0,5937
EXPLICIT_CLICK	35	0,1977	0,8559	0,3211	0,2010	0,8725	0,3121	0,6564
EXPLICIT_CLICK_SKIP	35	0,3314	0,4831	0,3931	0,3125	0,4684	0,3842	0,5519
EXPLICIT_CLICK_TOP	35	0,2261	0,8220	0,3547	0,2275	0,8237	0,3432	0,6526
EXPLICIT_CLICK_PREDECESSOR	35	0,3214	0,4576	0,3776	0,3004	0,4450	0,3643	0,5375
EXPLICIT_CLICK_SUCCESOR	35	0,3174	0,4492	0,3719	0,2945	0,4334	0,3678	0,5313
EXPLICIT_CLICK_SKIP_TOP_PREDECESSOR_SUCCESOR	35	0,2700	0,5424	0,3606	0,2728	0,5514	0,3138	0,5748
EXPLICIT_CLICK_TOP_PREDECESSOR	35	0,2584	0,5847	0,3584	0,2540	0,6272	0,3395	0,5876
EXPLICIT_CLICK_PREDECESSOR_SUCCESOR_TOP	35	0,2686	0,5508	0,3611	0,2720	0,6014	0,3419	0,5782
ONLY_EXPLICIT	70	0,2543	0,6981	0,3728	0,2839	0,7150	0,3742	0,6233
EXPLICIT_CLICK	70	0,2118	0,8491	0,3390	0,2158	0,8460	0,3333	0,6525
EXPLICIT_CLICK_SKIP	70	0,3200	0,5283	0,3986	0,2940	0,5121	0,0000	0,5320
EXPLICIT_CLICK_TOP	70	0,2243	0,8019	0,3505	0,2187	0,7903	0,3357	0,6494
EXPLICIT_CLICK_PREDECESSOR	70	0,3140	0,5094	0,3885	0,2878	0,4966	0,3730	0,5323
EXPLICIT_CLICK_SUCCESOR	70	0,3140	0,5094	0,3885	0,2861	0,4966	0,3730	0,5272
EXPLICIT_CLICK_SKIP_TOP_PREDECESSOR_SUCCESOR	70	0,2778	0,5189	0,3618	0,2837	0,5599	0,3519	0,5561
EXPLICIT_CLICK_TOP_PREDECESSOR	70	0,2618	0,5755	0,3599	0,2670	0,6169	0,3537	0,5763
EXPLICIT_CLICK_PREDECESSOR_SUCCESOR_TOP	70	0,2745	0,5283	0,3613	0,2794	0,5784	0,3517	0,5593
ONLY_EXPLICIT	105	0,2313	0,6733	0,3443	0,2436	0,6589	0,3419	0,5930
EXPLICIT_CLICK	105	0,1954	0,8416	0,3172	0,1951	0,8385	0,3123	0,6568
EXPLICIT_CLICK_SKIP	105	0,2925	0,4257	0,3468	0,2796	0,4248	0,3305	0,5080
EXPLICIT_CLICK_TOP	105	0,2089	0,7921	0,3306	0,2158	0,7858	0,3308	0,6226
EXPLICIT_CLICK_PREDECESSOR	105	0,2832	0,4851	0,3577	0,2667	0,4558	0,3309	0,5262
EXPLICIT_CLICK_SUCCESOR	105	0,2847	0,4059	0,3347	0,2716	0,4002	0,3183	0,5058
EXPLICIT_CLICK_SKIP_TOP_PREDECESSOR_SUCCESOR	105	0,2704	0,5248	0,3569	0,2799	0,5400	0,3543	0,5648
EXPLICIT_CLICK_TOP_PREDECESSOR	105	0,2535	0,5446	0,3459	0,2651	0,5623	0,3451	0,5682
EXPLICIT_CLICK_PREDECESSOR_SUCCESOR_TOP	105	0,2687	0,5347	0,3576	0,2781	0,5519	0,3550	0,5662

Abbildung 8-17: Implizites Feedback, Bewertung größer 2

Chunk: 70	Rating	Micro-averaged			Macro-averaged			nDCG
		Precision	Recall	F1	Precision	Recall	F1	
CLICK	> 0	0,7140	0,7749	0,7432	0,7279	0,7831	0,7353	0,6369
CLICK_SKIP	> 0	0	0	0	0	0	0	0
CLICK_TOP	> 0	0,7171	0,7749	0,7449	0,7308	0,7834	0,7371	0,6440
CLICK_PREDECESSOR	> 0	0	0	0	0	0	0	0
CLICK_SUCESSOR	> 0	0	0	0	0	0	0	0
CLICK_SKIP_TOP_PREDECESSOR_SUCESSOR	> 0	0,8837	0,4502	0,5965	0,8994	0,4486	0,5794	0,5477
CLICK_TOP_PREDECESSOR	> 0	0,9010	0,4313	0,5833	0,9072	0,4370	0,5810	0,5275
CLICK_PREDECESSOR_SUCESSOR_TOP	> 0	0,9000	0,4052	0,5588	0,9082	0,4111	0,5559	0,5209
CLICK	> 1	0,3908	0,7749	0,5196	0,3996	0,7980	0,5124	0,6369
CLICK_SKIP	> 1	0	0	0	0	0	0	0
CLICK_TOP	> 1	0,3947	0,7792	0,5240	0,4037	0,8021	0,5168	0,6440
CLICK_PREDECESSOR	> 1	0	0	0	0	0	0	0
CLICK_SUCESSOR	> 1	0	0	0	0	0	0	0
CLICK_SKIP_TOP_PREDECESSOR_SUCESSOR	> 1	0,5395	0,5022	0,5202	0,5270	0,4900	0,4921	0,5477
CLICK_TOP_PREDECESSOR	> 1	0,5099	0,4459	0,4758	0,5070	0,4584	0,4697	0,5275
CLICK_PREDECESSOR_SUCESSOR_TOP	> 1	0,5158	0,4242	0,4656	0,5174	0,4357	0,4597	0,5209
CLICK	> 2	0,1856	0,8019	0,3014	0,1919	0,8479	0,2956	0,6369
CLICK_SKIP	> 2	0	0	0	0	0	0	0
CLICK_TOP	> 2	0,1864	0,8019	0,3025	0,1928	0,8479	0,2967	0,6440
CLICK_PREDECESSOR	> 2	0	0	0	0	0	0	0
CLICK_SUCESSOR	> 2	0	0	0	0	0	0	0
CLICK_SKIP_TOP_PREDECESSOR_SUCESSOR	> 2	0,2884	0,5849	0,3863	0,2748	0,5542	0,3533	0,5477
CLICK_TOP_PREDECESSOR	> 2	0,2723	0,5189	0,3571	0,2618	0,5339	0,3211	0,5275
CLICK_PREDECESSOR_SUCESSOR_TOP	> 2	0,2737	0,4906	0,3514	0,2672	0,4957	0,3288	0,5209

Abbildung 8-18: Ausschließlich implizites Feedback

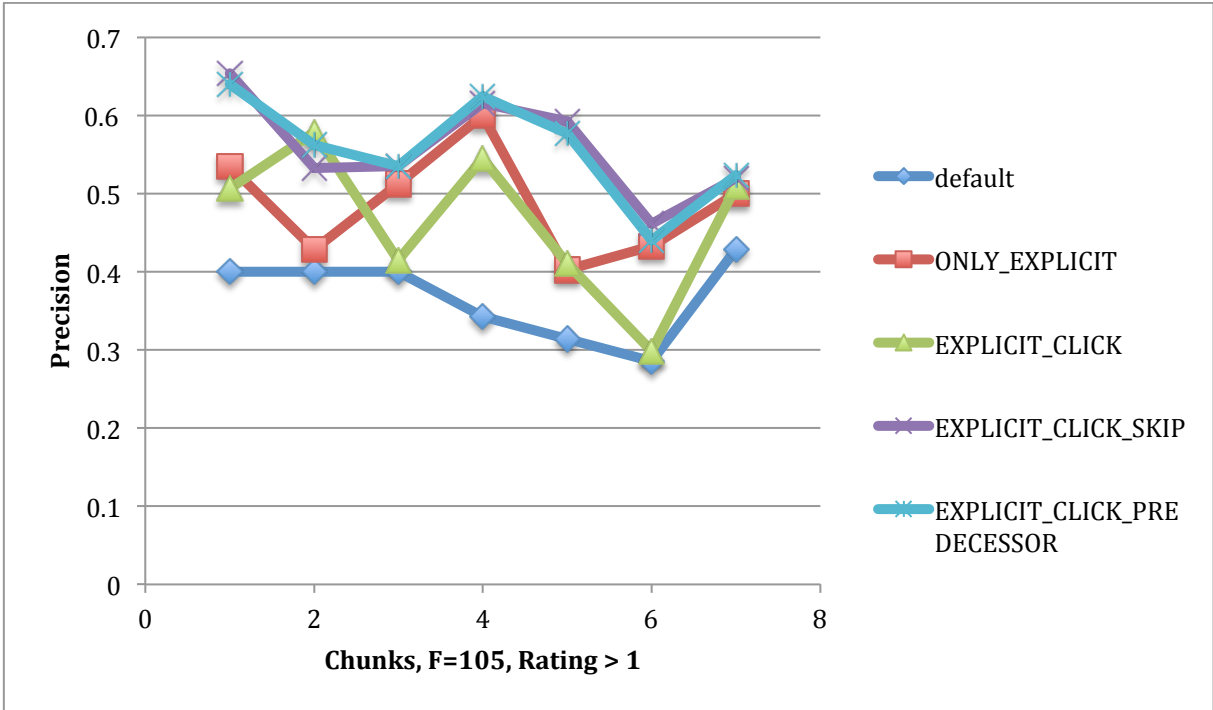


Abbildung 8-19: Implizite Feedback Strategien, Vergleich Precision / Chunks mit F=105 und Bewertung > 1

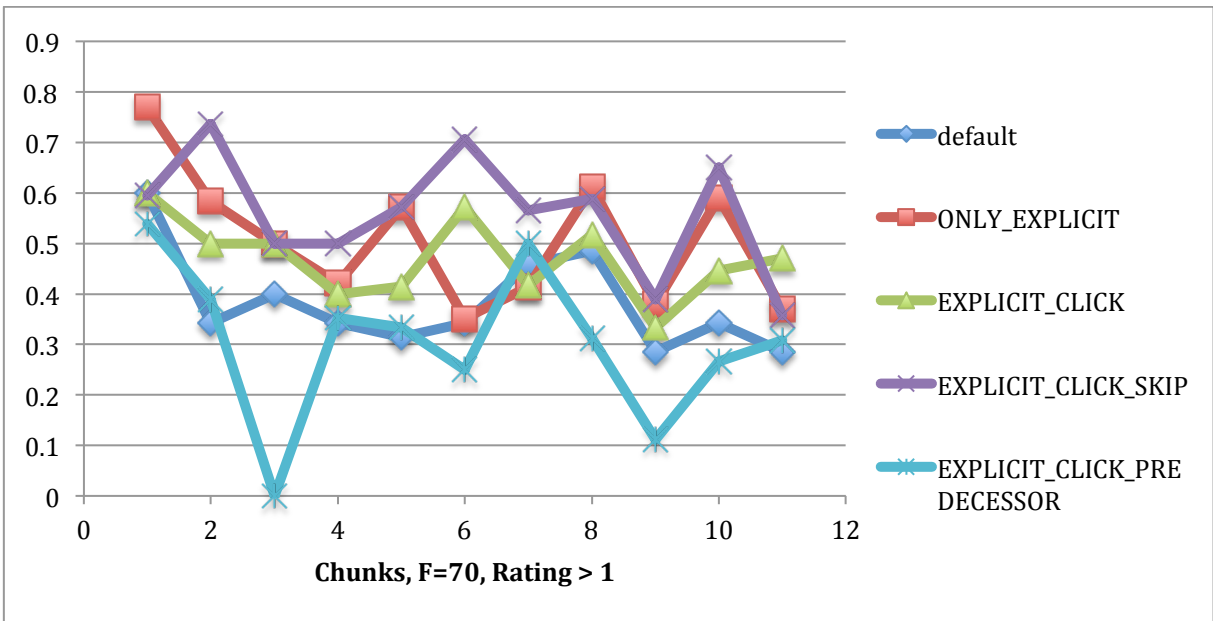


Abbildung 8-20: Implizite Feedback Strategien, Vergleich Precision / Chunks mit F=70 und Bewertung > 1

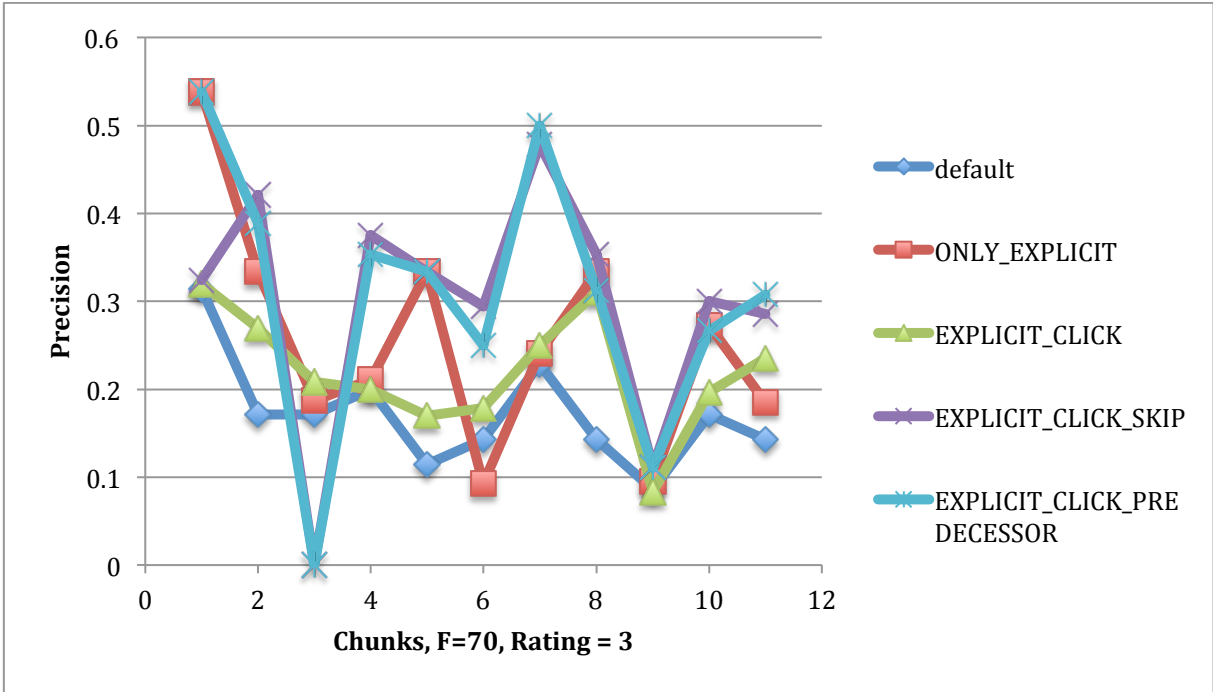


Abbildung 8-21: Implizite Feedback Strategien, Vergleich Precision / Chunks mit F=70 und Bewertung = 3

kNN, socialScore > 2		Micro-averaged				Macro-averaged			
Strategy	Rating	Baseline	Precision	Recall	F1	Precision	Recall	F1	nDCG
ONLY_EXPLICIT	> 0	0,6594	0,84615	0,1525	0,2584	0,85758	0,1539	0,2583	0,4163
EXPLICIT_CLICK	> 0	0,6594	0,84783	0,1545	0,2613	0,85758	0,1556	0,2611	0,4262
EXPLICIT_CLICK_SKIP	> 0	0,6594	0,84783	0,1545	0,2613	0,85758	0,1556	0,2611	0,4214
EXPLICIT_CLICK_PREDECESSOR	> 0	0,6594	0,84783	0,1545	0,2613	0,85758	0,1556	0,2611	0,4214
ONLY_EXPLICIT	> 1	0,3679	0,46154	0,1495	0,2258	0,48842	0,1552	0,2303	0,4163
EXPLICIT_CLICK	> 1	0,3679	0,46739	0,153	0,2306	0,49448	0,1588	0,2357	0,4262
EXPLICIT_CLICK_SKIP	> 1	0,3679	0,46739	0,153	0,2306	0,49448	0,1582	0,235	0,4214
EXPLICIT_CLICK_PREDECESSOR	> 1	0,3679	0,46739	0,153	0,2306	0,49448	0,1582	0,235	0,4214
ONLY_EXPLICIT	> 2	0,1679	0,26374	0,1875	0,2192	0,28351	0,2147	0,2229	0,4163
EXPLICIT_CLICK	> 2	0,1679	0,27174	0,1953	0,2273	0,28957	0,2204	0,2299	0,4262
EXPLICIT_CLICK_SKIP	> 2	0,1679	0,27174	0,1953	0,2273	0,28957	0,2204	0,2299	0,4214
EXPLICIT_CLICK_PREDECESSOR	> 2	0,1679	0,27174	0,1953	0,2273	0,28957	0,2204	0,2299	0,4214

Abbildung 8-22: kNN, Sortierung nach socialScore mit socialScore größer 2.

Literaturverzeichnis

BANOS, Evangelos, et al. PersoNews: a personalized news reader enhanced by machine learning and semantic filtering. In: *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*. Springer Berlin Heidelberg, 2006. S. 975-982.

BERGAMASCHI, Sonia, et al. RELEVANT News: a semantic news feed aggregator*. SEMANTIC WEB APPLICATIONS AND PERSPECTIVES (SWAP 2007), 2007, S. 150.

BIFET, Albert; FRANK, Eibe. Sentiment knowledge discovery in twitter streaming data. In: *Discovery Science*. Springer Berlin Heidelberg, 2010. S. 1-15.

Christopher M. Bishop: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. Corr. 2nd printing Auflage, October 2007.

Blom, J. Personalization - a taxonomy. Extended Abstracts of CHI'00, 313-314, New York, ACM.

Blom, J. A Theory of Personalized Recommendations. Doctoral Consortium of CHI 2002, 540-541, Minnesota, ACM.

Bomhardt und Gaul, Feedback Options for a Personal News Recommendation Tool, Cooperation in Classification and Data Analysis, Studies in Classification, Data Analysis, and Knowledge Organizations. Springer, 2009

BURKE, Robin. Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 2002, 12. Jg., Nr. 4.

Doerr, J., Benlian, A., Vetter, J. and Hess, T. (2010) Pricing of Content Services – An Empirical Investigation of Music as a Service, in Proceedings of the sixteenth Americas Conference on Information Systems, August 12-15, Lima, Peru, ACM Press, 13-24.

J. Gama, R. Sebastião, and P. P. Rodrigues. Issues in evaluation of stream learning algorithms. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 329-338, 2009.

JOACHIMS, Thorsten, et al. Accurately interpreting clickthrough data as implicit feedback. In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2005. S. 154-161.

JOACHIMS, Thorsten, et al. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. ACM Transactions on Information Systems (TOIS), 2007, 25. Jg., Nr. 2.

JOACHIMS, Thorsten. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1996.

Kelly und Teevan, Implicit Feedback for Inferring User Preference: A Bibliography, ACM, 2003

KEPPLINGER, Hans Mathias. Der Nachrichtenwert der Nachrichtenfaktoren. In: *Journalismus als Beruf*. VS Verlag für Sozialwissenschaften, 2011. S. 61-75.

Hans Mathias Kepplinger: „Der Nachrichtenwert der Nachrichtenfaktoren“ In: Bacha/Scherer/ Waldmann (Hrsg.): „Wie die Medien die Welt erschaffen und wie die Menschen darin leben“, Westdeutscher Verlag, Opladen/Wiesbaden, 1998

KATZ, Dipl-Medieninf Philipp. NewsSeecr – Clustering und Ranking von Nachrichten zu Named Entities aus Newsfeeds, Diplomarbeit, TU Dresden, 2010

KRAMER, Joseph; NORONHA, Sunil; VERGO, John. A user-centered design approach to personalization. *Communications of the ACM*, 2000, 43. Jg., Nr. 8, S. 44-48.

Li T., Unger T. Willing to pay for quality personalization? Trade-off between quality and privacy. *European Journal of Information Systems* 2012, 621, Operational Research Society Ltd.

LIANG, Ting-Peng; LAI, Hung-Jen; KU, Yi-Cheng. Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings. *Journal of Management Information Systems*, 2007, 23. Jg., Nr. 3, S. 45-70.

Maltz, Barney: Freemium Software: A Guide for Startups, Institutional Venture Partners, 2012

MANNING, Christopher D.; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. Introduction to information retrieval. Cambridge: Cambridge University Press, 2008.

RAUSCHER, Barbara; HESS, Thomas. Kontextsensitive Inhaltebereitstellung: Begriffsklärung und Analysegrundlagen. Ludwig-Maximilians-Universität München, Institut für Wirtschaftsinformatik und Neue Medien, WIM-Arbeitsbericht, 2005, Nr. 3

REICHWALD, Ralf; MEIER, Roland; FREMUTH, Natalie. Die mobile Ökonomie-Definition und Spezifika. *Mobile Kommunikation*, Wiesbaden, 2002

Riedel, Doris: Modellierung in der Geoinformation. http://homepage.univie.ac.at/doris.riedel/model0910/tut_geostat/RMS.htm. 2008 (Abruf: 19.9.2013)

ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 1958, 65. Jg., Nr. 6, S. 386.

Runte, M.: Personalisierung im Internet: individualisierte Angebote mit Collaborative Filtering, Wiesbaden, 2000

SAMPER, Juan J., et al. NectarRSS, an intelligent RSS feed reader. *Journal of Network and Computer Applications*, 2008, 31. Jg., Nr. 4, S. 793-806.

Schneider, P.: Produktindividualisierung als Marketing-Ansatz, Schesslitz, 1998
SHAPIRO, Carl; VARIAN, Hal. Information Rules: A Strategic Guid. Harvard Business Press, 1998

Simonson, I. (2005) Determinants of Customers' Responses to Customized Offers: Conceptual Framework and Research Propositions. *Journal of Marketing*, 69, 32-45

Stumme und Krause: Vorlesung „Internet-Suchmaschinen“ 2009, <http://www.kde.cs.uni-kassel.de/lehre/ss2009/IR>, 2009, Abruf 13.9.2013

TAM, Kar Yan; HO, Shuk Ying. Understanding the impact of web personalization on user information processing and decision outcomes. *Mis Quarterly*, 2006, S. 865-890.

WEN, Hao; FANG, Liping; GUAN, Ling. A hybrid approach for personalized recommendation of news on the Web. *Expert Systems with Applications*, 2012, 39. Jg., Nr. 5, S. 5806-5814.

YU, Cui, et al. Indexing the distance: An efficient method to knn processing. In: *VLDB*. 2001. S. 421-430.

ZHANG, Min-Ling; ZHOU, Zhi-Hua. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 2007, 40. Jg., Nr. 7, S. 2038-2048.

Zhang, B., and Seo, Y. Personalized Web-document filtering using reinforcement learning. *Applied Artificial Intelligence*, 15, 7 (2001)

