



# Hot Topics in Intelligent User Interfaces

Cem Özcan

Karlsruher Institut für Technologie

# Inhaltsverzeichnis

1	Recommender Systems	4
1.1	Grundlagen	4
1.2	Content-based filtering	4
	Allgemein	4
	User Experience	5
1.3	Collaborative filtering	5
	Allgemein	5
	Context-specific trust [23]	6
1.4	Vergleich beider Ansätze	7
	Inhaltsbasiertes Filtern	7
	Kollaboratives Filtern	7
1.5	Hybrid Recommender Systems	7
	Personalized Explanations [18]	7
2	Ubiquitous Computing	9
2.1	Grundlagen	9
2.2	Wearable Computing	9
	Allgemein	9
	Assistive Intelligent User Interfaces	9
2.3	Mobile Computing	11
	Allgemein	11
	User feedback	11
3	Multimodal Interfaces	13
3.1	Grundlagen	13
3.2	Natural Language Interfaces	13
3.3	Touch gestures	14
4	Fazit	16
4.1	Themen	16
4.2	Gemeinsamkeiten und Unterschiede	16

Struktur ist  
nicht  
klar  
erkennbar

App.  
elt?

**Zusammenfassung.** Vor allem in den letzten Jahren wurden Computer immer mehr dazu eingesetzt, uns bei der Lösung komplexer Probleme zu assistieren. Aufgrund der **steigenden Komplexität** der Aufgaben gewinnt eine effiziente Interaktion zwischen Mensch und Maschine für Nutzer also immer mehr an Bedeutung.

Eine Forschungsrichtung, die dieses Problem untersucht, ist die Forschung im Bereich "Intelligent User Interfaces". In diesem Gebiet konzentriert **man** sich auf die Entwicklung von intuitiven Benutzerschnittstellen, die die Interaktion mit einem Intelligenten System erleichtern und intuitiver gestalten sollen [16,29].

Die aktuelle Forschung richtet sich dabei hauptsächlich auf die Lösung von Problemen im Gebiet "**Information retrieval**" durch den Einsatz von "Recommender Systems" (Siehe section 1), die nahtlose Integration von Mobilien und Tragbaren Geräten in den Alltag (siehe section 2) und die Bewältigung komplexer Aufgaben durch den Einsatz multimodaler Interaktion (Siehe section 3).

Im Folgenden werden Paper aus der Konferenz "ACM Intelligent User Interfaces" [2] vorgestellt, die repräsentativ für die aktuelle Forschung in diesem Bereich sind.

✓ **Schlüsselwörter:** Intelligent User Interfaces, Recommender Systems, Explainable Artificial Intelligence, Ubiquitous Computing, Multimodal Interfaces, Human-Centered Computing, Human-Computer-Interaction

Aufklärung des  
Inhalts vermeiden

↗ auch für  
das Erste?

↘ Begründung,  
warum  
repräsentativ

# 1 Recommender Systems

## 1.1 Grundlagen

Recommender Systems sind vor allem im Zeitalter der Digitalisierung effektive Dienste, die die Mensch-Maschine-Interaktion erleichtern sollen. Durch Analyse des Nutzerverhaltens versuchen Recommender Systems vorherzusehen, welche Inhalte (im Folgenden auch Items genannt) für den Nutzer interessant sein könnten, um diese dann dem Nutzer zu empfehlen. Das trägt zu einer verbesserten Wahrnehmung vom System von Seiten des Nutzers bei und verbessert so auch die User Experience [6,12,17].

Das Filtern von irrelevanten Informationen erwies sich, vor allem in den letzten Jahren, als effektiver Ausweg aus der Informationsüberflutung [23,28] und gewann für Nutzer zunehmend an Bedeutung. Diese haben durch den Einsatz von Recommender Systems Zugang zu mehr relevanten Informationen, die bei einer Recherche ohne Hilfe eines solchen Systems, teilweise in der Masse untergegangen wären. Wie in [13] beschrieben kann das Auslassen relevanter Informationen beispielsweise im Gesundheitswesen fatale Folgen nach sich ziehen.

Auch für Unternehmen, die ihre Inhalte im Internet anbieten, spielen Recommender Systems eine wichtige Rolle. Gesammelte Nutzerdaten werden dazu eingesetzt, Massenwerbung nach und nach durch personalisierte Werbung und Produktempfehlungen zu ersetzen, was sich auch im Konsumverhalten der Nutzer bemerkbar macht:

Nach eigenen Angaben haben Netflix und YouTube je 75 und 70 Prozent ihrer Ansichten und Amazon 35 Prozent ihrer Verkäufe ihrem Recommender System zu verdanken [15,31].

Die Genauigkeit der Empfehlungen sind demnach von großer Bedeutung sowohl für die Nutzer als auch für die Anbieter des Systems.

Für die Forschung in diesem Gebiet ergibt sich daher die Herausforderung, das Empfehlen und Filtern von Inhalten durch Personalisierung, zusätzlicher Parametrisierung und Kontextualisierung so präzise wie möglich zu gestalten, ohne dabei die User Experience und Benutzbarkeit des Systems zu komprimieren.

Im Folgenden werden verschiedene Klassen von Recommender Systems und zum Thema relevante Paper vorgestellt, die Ansätze zur Verbesserung der Empfehlungen untersuchen.

## 1.2 Content-based filtering

### Allgemein

Inhaltsbasiertes Filtern ist eines der am häufigsten verwendeten Ansätze bei der Implementierung von Recommender Systems. Grundlegende Informationen, die dem System bekannt sein müssen, sind frühere Itembewertungen der Nutzer und möglichst viele Attribute, die einen Item so präzise wie möglich beschreiben. Basierend auf diesen Daten werden Items kategorisiert und Nutzerprofile erstellt [28].

Folgende Voraussetzungen müssen erfüllt sein, damit ein Item I einem Nutzer U empfohlen wird:

Sprung  
unklar



These

Vielleicht  
Grafik?

- ✓ – I wurde noch nicht von Nutzer U bewertet
- I ist ähnlich zu anderen Items, die U in der Vergangenheit positiv bewertet hat

Häufig werden Item- und Nutzerprofile als Term Frequency-Inverse Document Frequency-Vektoren (TF-IDF-Vektoren) dargestellt und die Ähnlichkeit zweier Items mithilfe von Korrelationskoeffizienten oder Abstandsfunktionen berechnet [25,28].

### User Experience

Eines der Vorteile inhaltsbasierter Recommender Systems ist die Transparenz und die Möglichkeit für den Nutzer, Empfehlungen nachzuvollziehen. Der Grund, weshalb die Nachvollziehbarkeit von Empfehlungen so wichtig ist, ist, dass sie sich auf die Wahrnehmung auswirkt, die der Nutzer vom System hat [12].

In [12] wurde, mithilfe einer Bilddatenbank, ein Recommender System für Gemälde implementiert. Anschließend wurde in einer Nutzerstudie ( $N = 121$ ) die Wirkung, die das Erklären von Empfehlungen auf die Nutzer hat, untersucht.

Dabei wurden drei verschiedene Interfaces implementiert, die die Empfehlungen des Systems auf unterschiedliche Art und Weise erklären sollen:

- $I_1$  Ohne Erklärungen
- $I_2$  Zeigt dem Nutzer Bilder, die von diesem Positiv bewertet wurden und ähnlich zum empfohlenen Bild sind
- $I_3$  Zeigt dem Nutzer, welche visuellen Eigenschaften das empfohlene Bild mit von ihm positiv bewerteten Bildern teilt

Diese Interfaces wurden in Kombination mit zwei verschiedenen inhaltsbasierten Empfehlungsalgorithmen, Deep Neuronal Networks (DNN) und Attractiveness Visual Features (AVF), den Nutzern zum Testen bereitgestellt. DNN generiert präzisere Empfehlungen als AVF, dafür ist der Entscheidungsprozess bei AVF transparenter, was eine Kombination aus  $I_3$  und AVF ermöglicht (dies ist mit DNN nicht möglich).

Dominguez et. al. kamen dabei zum Ergebnis, dass die Kombination aus  $I_2$  und Deep Neuronal Networks von den Nutzern die positivsten Bewertungen erhalten hat. Die Kombination aus  $I_3$  und Attractiveness Visual Features ist zwar die transparenteste Methode, hat aber relativ unpräzise Empfehlungen im Vergleich zu anderen Kombinationen. Das hat eine schlechtere Wahrnehmung und Bewertung des Systems aus Sicht der Nutzer zur Folge.

Aus [12] kann man also folgern, dass bei der Implementierung eines Recommender Systems Wert auf Transparenz gelegt werden sollte, wenn diese die Qualität der Empfehlungen nicht abschwächt.

## 1.3 Collaborative filtering

### Allgemein

Anders als beim inhaltsbasierten Filtern werden beim kollaborativen Filtern keine Informationen über den Inhalt eines Items im System benötigt. Stattdessen

werden, je nach Ansatz, Nutzer- bzw. Itemprofile basierend auf deren Bewertungshistorie erstellt:

#### **User-based approach**

Es werden Nutzerprofile basierend auf Bewertungen, die die Nutzer in der Vergangenheit über Items abgegeben haben, erstellt. Typischerweise wird der k-Nearest-Neighbour-Algorithmus eingesetzt, um Nutzerprofile mit ähnlicher Bewertungshistorie zu finden.

Falls hinreichend viele Nutzer, die sich in der Nachbarschaft des Nutzers U befinden, ein Item I positiv bewertet haben, so bekommt Nutzer U eine Empfehlung für Item I [20,28].

#### **Item-based approach**

Es werden Itemprofile basierend auf Bewertungen, die die Items von Nutzern erhalten haben, erstellt.

Zwei Items sind genau dann ähnlich, wenn es hinreichend viele Nutzer gibt, die beide Items bewertet haben. Außerdem muss eine Korrelation im Bewertungsverhalten der Nutzer, bezogen auf genannte Items, existieren.

Diese Informationen werden dann dafür verwendet, die Bewertung eines Nutzers für ein Item abzuschätzen [20,21].

#### **Context-specific trust [23]**

Mit der Hypothese, dass kollaboratives Filtern hohes Verbesserungspotential besitzt, hat man in [23] und [14] versucht, mit Modifikationen am kollaborativen Filtern auf bessere Empfehlungen zu kommen.

Die grundlegende Idee des Ansatzes in [23] ist dabei, den Kontext bei der Wahl der Nachbarschaft eines Nutzers zu berücksichtigen. Dazu werden die in [4] eingeführten Definitionen für "Trust" (Vertrauenswürdigkeit) modelliert und bei der Wahl ähnlicher Profile als zusätzlicher Parameter berücksichtigt:

#### **Profile-Level trust**

Die Vertrauenswürdigkeit eines Nutzers U ist abhängig davon, wie präzise die Empfehlungen sind, die andere Nutzer, aufgrund ihrer Ähnlichkeit zu U, erhalten haben.

#### **Item-Level trust**

Sei I ein von Nutzer U bewertetes Item. Die Vertrauenswürdigkeit von U im Bezug auf I ist abhängig davon, ob I häufig erfolgreich anderen Nutzern empfohlen wurde, weil diese ähnlich zu U sind.

O'Donovan et. al. haben verschiedene kollaborative Filteralgorithmen mit ihren Modellen erweitert und die Qualität der Empfehlungen miteinander verglichen. Zur Evaluierung der Algorithmen wurde ein Datensatz mit 950 Profilen mit je durchschnittlich 105 Filmbewertungen verwendet.

Die Ergebnisse waren recht Eindeutig: Jeder Algorithmus, der um den zusätzlichen Parameter "Trust" erweitert wurde, hat bessere Empfehlungsergebnisse geliefert, als der unmodifizierte Algorithmus. Die modifizierten Algorithmen waren

zwischen 3 und 22 % weniger Fehleranfällig in ihren Empfehlungen, als der unmodifizierte Algorithmus.

#### 1.4 Vergleich beider Ansätze

##### Inhaltsbasiertes Filtern

Zum Einen ist es nicht notwendig, Persönliche Informationen über die Nutzer zu halten und zum Anderen löst die Kategorisierung der Items das sogenannte “New-Item-Problem“ [5,28]. Beim inhaltsbasierten Filtern können also, anders als beim kollaborativem Filtern, auch neue Produkte empfohlen werden, ohne zuvor eine Bewertung zu erhalten. Des Weiteren ist es für Nutzer einfacher nachzuvollziehen, weshalb sie bestimmte Empfehlungen bekommen, was sich positiv auf die User Experience auswirkt (siehe section 1.2) [12].

##### Kollaboratives Filtern

Das Hauptproblem inhaltsbasierten Filterns ist der, dass Nutzer nur Empfehlungen erhalten, die sich in dieselben Kategorien einordnen lassen, wie die bereits vom Nutzer bewerteten Items. Durch kollaboratives Filtern können Nutzer auch Empfehlungen aus Kategorien bekommen, die ihnen noch unbekannt waren. Ein Problem des kollaborativen Filterns ist das sogenannte “Cold-Start-Problem“. Neue Nutzer haben keine Bewertungshistorie, weswegen das System ihnen keine Items empfehlen kann.

#### 1.5 Hybrid Recommender Systems

Um von den Vorteilen mehrerer verschiedener Empfehlungsalgorithmen zu profitieren werden häufig “Hybrid Recommender Systems“ (hybride Empfehlungsdienste) verwendet. Dies sorgt in der Regel für präzisere Empfehlungen als traditionelles kollaboratives oder inhaltsbasiertes Filtern [5].

In [18] wurde beispielsweise ein Hybrid Recommender System mit traditionellem kollaborativem Filtern verglichen. Nicht nur waren die Empfehlungen des Hybrid Recommender Systems akkurater, es war außerdem dank inhaltsbasierter Aspekte möglich, Empfehlungen wie in [12] zu Begründen.

##### Personalized Explanations [18]

Wie auch schon in [12] (siehe section 1.2) wurde in [18] versucht, die Empfehlungen eines Recommender Systems mithilfe unterschiedlicher Methoden zu erklären.

Kouki et. al. haben dafür ein Hybrid Recommender System implementiert, das sieben verschiedene Arten von Erklärungen für die Empfehlungen anbietet und diese in verschiedenen Formaten (visuell oder textuell) darstellt.

Die verschiedenen Erklärungsansätze lassen sich dabei in zwei Kategorien einordnen. Inhaltsbasierte Erklärungen, die Empfehlungen basierend auf deren Inhalt begründen und nutzerbasierte Erklärungen, die Empfehlungen basierend auf der Präferenz ähnlicher Nutzer begründen.



Ziel der Studie [18] war es, durch Variation der Anzahl verschiedener Erklärungen für eine Empfehlung, herauszufinden, welchen Einfluss diese auf die Nutzer haben.

Zur Beantwortung dieser Fragen, wurde das Recommender System mithilfe einer Nutzerstudie ( $N = 198$ ) evaluiert. Basierend auf einem Datensatz der Musikplattform Last.fm wurden diesen Nutzern Musikinterpreten vorgeschlagen. Die Vorschläge wurden wie schon in [12] von Begründungen begleitet, die dem Nutzer helfen sollen, die Empfehlungen zu verstehen.

Kouki et. al. haben nach einer Befragung genannter Nutzer herausgefunden, dass

- 1) Nutzer inhaltsbasierte Erklärungen überzeugender als nutzerbasierte Erklärungen fanden
- 2) Nutzer im Durchschnitt drei bis vier verschiedene Erklärungsansätze für ihre Empfehlungen bevorzugen
- 3) Nutzer die textuelle Darstellung der Erklärungen der visuellen Darstellung vorziehen

Ein weiteres interessantes Ergebnis der Studie in [18] war, dass die Genauigkeit des Empfehlungsalgorithmus nicht zu Zwecken der Erklärbarkeit abgeschwächt werden sollte. Auf ein ähnliches Ergebnis sind auch Dominguez et. al. in [12] gekommen (siehe section 1.2).

## 2 Ubiquitous Computing

### 2.1 Grundlagen

Der Begriff “Ubiquitous Computing“ beschreibt die Eingliederung von Computern in den menschlichen Alltag und wurde bereits 1991 von Mark Weiser (1952-1999) verwendet und geprägt [36]. Das Ziel des “Ubiquitous Computing“ ist es, das alltägliche Leben des Menschen mithilfe von intelligenten Geräten, die aus dem Hintergrund heraus agieren, zu erleichtern. Weisers Vorstellung des 21. Jahrhunderts hat sich als richtig herausgestellt: Heutzutage suchen Nutzer die Interaktion mit Computern nicht länger aktiv auf, vielmehr ist die Mensch-Maschine-Interaktion allgegenwärtig ohne dabei im Vordergrund unseres Lebens zu sein.

Die Veröffentlichungen der Konferenz “Intelligent User Interfaces“ beschäftigen sich vor allem mit den Themen “Mobile Computing“ und “Wearable Computing“, da Smartphones und andere mobile bzw. tragbare Geräte zunehmend an Bedeutung in unserem Alltag gewinnen.

### 2.2 Wearable Computing

#### Allgemein

„Wearable devices“ sind am Körper getragene Geräte, die den Nutzer im Alltag unterstützen sollen, ohne dabei von diesem als störend empfunden zu werden. Ein typisches Beispiel für Wearables aus dem kommerziellen Bereich sind Fitnessarmbänder:

Durch die Positionierung am Handgelenk des Nutzers können diese den Nutzer durch das Messen der Vitalfunktionen und Körperbewegung (Pulsmesser und Schrittzähler) beim Sport unterstützen, ohne dessen Bewegungen einzuschränken.

In der Forschung hingegen, werden Wearables vor allem mit Blick auf die Verbesserung der Mensch-Maschine-Interaktion eingesetzt.

#### Assistive Intelligent User Interfaces

Eine Vielzahl der Arbeiten, die auf der Konferenz vorgestellt werden, richtet sich auf Unterstützungstechnologien zur Hilfe von Menschen mit kognitiven Beeinträchtigungen [10,22]. Speziell Wearables beispielsweise werden hauptsächlich dafür eingesetzt, um die Kommunikation und Interaktion zwischen seh- bzw. hörgeschädigten Personen mit Personen ohne jeweilige Einschränkungen zu erleichtern [8,26,27,37].

Paudyal et. al. haben sich sowohl in [26] als auch in [27] mit dem Einsatz von Wearables zur Erkennung von Gesten in der Gebärdensprache beschäftigt:

#### SCEPTRE [26]

Mit dem Ziel, die Kommunikation zwischen Personen zu erleichtern, von denen nur eine die “American Sign Language“ (ASL) beherrscht, haben Paudyal et. al. die Applikation SCEPTRE entwickelt, die ASL-Gesten erkennen

und übersetzen soll. Zur Erkennung von Gesten werden Myo-Armbänder [19] verwendet, die der Nutzer während der zweistufigen Interaktion mit dem System tragen muss:

*Training* : Die Gestenerkennung des Systems ist abhängig vom Nutzer. Aus diesem Grund ist es für den Nutzer erforderlich, dem System zu Beginn der Interaktion Gesten beizubringen. Dafür wählt der Nutzer über die App eine ASL-Geste und führt diese drei mal aus.

Durch die Sensoren im Myo-Armband (Gyroskop, Beschleunigungssensor, Elektromyographiesensor) hat das System nun Sensordaten zur ausgewählten ASL-Geste. Diese verwendet das System, um eine Vorlage (Template) von der neu erlernten ASL-Geste zu erstellen

*Gestenerkennung* : Zur Erkennung von Gesten wird “Template matching“ verwendet. Die Inputdaten einer Geste werden dafür mittels Dynamic Time Wrapping mit den im *Training* erstellten Templates verglichen. Die ASL-Geste, deren Template dem Input am ähnlichsten ist, wird vom System vorgeschlagen.

Anschließend wurde die Gestenerkennung in einer Nutzerstudie auf Genauigkeit, Reaktionszeit und Benutzbarkeit geprüft:

*Evaluation* : Dem System wurden 20 Wörter aus dem ASL beigebracht. Im Fall, dass derselbe Nutzer das System trainiert und getestet hat, hatte das System, unter Verwendung aller Sensoren, in eine Genauigkeit von 97.72%. Im Fall, dass das Testen und Trainieren des Systems von verschiedenen Nutzern übernommen wurden, waren die Ergebnisse deutlich schlechter. Dies wurde damit begründet, dass vor allem die Messwerte des Elektromyographiesensors sehr stark vom Nutzer abhängig sind.

### **Dynamic Feature Selection and Voting (DyFAV) [27]**

Um ihre Arbeit in [26] zu erweitern, haben Paudyal et. al. mit DyFAV ein weiteres System zur Gestenerkennung implementiert. Anders als SCEPTRE soll DyFAV allerdings keine ASL-Worte, sondern ASL-Buchstaben erkennen. Wie auch schon in SCEPTRE wurden Myo-Armbänder für den Input und die Sensordaten verwendet.

*Initialisierung* : Aufgrund der Beschränktheit des ASL-Alphabets (26 Buchstaben), ist es für den Nutzer nicht länger erforderlich, das System nach einer Initialisierung weiterhin zu trainieren. Zu Beginn der Interaktion mit DyFAV muss der Nutzer lediglich jeden Buchstaben im ASL-Alphabet fünf mal wiederholen, um die Gestenerkennung vollständig verwenden zu können.

*Gestenerkennung* : Zur Erkennung von Gesten wird Feature Engineering verwendet. Dafür werden auf Basis der Sensordaten aus der *Initialisierung* verschiedene Features miteinander Verglichen. Für jeden ASL-Buchstaben

werden dann die für die Klassifikation Signifikantesten Features ausgewählt und in Abhängigkeit ihrer Signifikanz gewichtet. Basierend auf diesen Gewichten wird dann versucht, die Input-Gesten in Echtzeit zu klassifizieren.

*Evaluation* : Im Gegensatz zu ASL-Worten unterscheiden sich ASL-Buchstaben hauptsächlich in der Positionierung der Finger und erfordern beim Gestikulieren kaum Bewegung im Arm. Das Erkennen von Gesten ist in DyFAV also viel mehr vom Elektromyographiesensor als von den anderen beiden Sensoren abhängig, was die Gestenerkennung zusätzlich erschwert.

Durch die Wahl hinreichend vieler Features, die für die Klassifikation eingesetzt werden, ist das System dennoch sehr präzise:

In einer Nutzerstudie ( $N = 9$ ) wurden 95.36% der Gesten vom System erkannt.

## 2.3 Mobile Computing

### Allgemein

Durch die steigende Beliebtheit von Smartphones seit 2007 [33] gewann auch die Forschung im Bereich “mobile Computing“ immer mehr an Bedeutung. Heutzutage sind Smartphones in unserem Alltag allgegenwärtig und ersetzen für viele Nutzer stationäre Geräte wie Desktop PCs.

Die folgenden Paper richten sich danach, den Entwicklungs- und Fehlerbehebungsprozess mobiler Software, unter Berücksichtigung von Nutzerinteressen, so effizient wie möglich zu gestalten.

### User feedback

Eine effektive Methode, um die Lebensdauer eines Softwareprodukts zu erhöhen, ist die Software auf Basis von Nutzerrezensionen zu Aktualisieren. App-Rezensionen werden jedoch häufig mit Kommentaren umgesetzt, was aufgrund der Unabhängigkeit der einzelnen Kommentare unstrukturiert ist. Aus diesem Grund ist es sehr schwierig für Entwickler, für die Entwicklung relevantes Feedback aus den Kommentaren zu extrahieren [11].

Zur Lösung dieses Problems haben Su’a et. al. QuickReview entwickelt [34]:

QuickReview ist ein Intelligent User Interface, das das Rezensierten von Apps benutzerfreundlicher gestalten soll und außerdem Entwicklern erlaubt, effizient für die Entwicklung relevante Informationen aus diesen Rezensionen zu gewinnen:

*Interaktion* : QuickReview stellt dem Nutzer eine Liste von Features der App, die dieser bewerten möchte, zur Verfügung. Der Nutzer kann dann ein Feature auswählen, das es kritisieren möchte, und bekommt dann eine Liste von Issues vorgeschlagen, die abhängig vom ausgewählten Feature ist. Um eine App zu rezensieren, muss der Nutzer also lediglich ein fehlerhaftes Feature auswählen und aus einer Liste von Issues diejenigen auswählen, die den Fehler am besten

beschreiben.

Um diese Adaptivität zu gewährleisten, werden vorhandene Kommentare über die zu Bewertende App, mittels Natural Language Processing nach Häufig vorkommenden Feature, Issue Paaren untersucht.

*Feedback* : Durch Rezensionen nach diesem Schema kann QuickReview eine nach Häufigkeit sortierte Liste aus Feature, Issue Paaren präsentieren. Entwickler können diese Liste dann zur Identifizierung und anschließender Behebung von Bugs verwenden.

*Evaluation* : Zur Evaluation des Systems wurde eine Nutzerbefragung ( $N = 20$ ) durchgeführt, bei der die Nutzer zunächst die App “MyTracks“ nutzen und später unter Verwendung von QuickReview rezensieren sollten. Anschließend haben die Nutzer die Benutzbarkeit und kognitive Überlastung von QuickReview bewertet. Als Kontrollinstanz wurde das traditionelle Rezensionssystem von Google Play verwendet, welches auf einfachen Kommentaren basiert.

QuickReview hat zwar in beiden Kategorien besser abgeschnitten als Google Play, jedoch waren die Unterschiede nicht eindeutig genug, um eine eindeutige Aussage über die Präferenz der Nutzer zu treffen.

Allerdings ist bei der Durchführung der Studie aufgefallen, dass die Nutzung von QuickReview deutlich weniger Zeit in Anspruch genommen hat, als das Schreiben von Kommentaren auf Google Play. Das führt zu der Vermutung, dass QuickReview aufgrund der Zeitersparnis Nutzer dazu anregen könnte, mehr Rezensionen abzugeben als traditionelle Systeme.

Ein weiterer Vorteil ist offensichtlicherweise die Zeitersparnis für die Entwickler, da QuickReview das Problem der Informationsüberflutung für diese löst.

### 3 Multimodal Interfaces

#### 3.1 Grundlagen

Traditionellerweise ist die Mensch-Maschine-Interaktion mit vielen Systemen unimodal, es wird also nur eine Art der Eingabe verwendet, um das System zu steuern. Mittlerweile werden aber neben Tastatur, Maus und Touchscreens auch andere Eingabemöglichkeiten wie Spracherkennung, Gestenerkennung und Eye Tracking immer präziser und beliebter und können für eine Effektivere Kommunikation zwischen Mensch und Maschine eingesetzt werden.[1]

Dieser Fortschritt motiviert dazu, die Mensch-Maschine-Interaktion an zwischenmenschliche Interaktionen anzunähern, indem die Interaktion multimodal gestaltet wird [24,30,35].

multimodale Interfaces kombinieren mehrere Interaktionsmöglichkeiten miteinander und helfen Nutzern somit, komplexe Aufgaben effizienter zu bewältigen [9]. Das System ist somit durch den Einsatz mehrerer Interaktionsmöglichkeiten weniger Fehleranfällig und sorgt durch die erhöhte Präzision in der Erkennung von Nutzereingaben für eine bessere Wahrnehmung des Systems.

Das Ziel multimodaler Interfaces ist es also, die Stärken der einzelnen Modalitäten zu kombinieren, ohne dass die Interaktion mit dem System vom Nutzer als konstraintuitiv wahrgenommen wird.

#### 3.2 Natural Language Interfaces

Ein großes Problem bei der Bedienung von Geräten mittels Sprachsteuerung ist, dass Nutzer häufig nicht wissen, wie genau sie Befehle formulieren müssen, damit sie das System versteht. Dies führt häufig zu Kommunikationsfehlern zwischen Nutzer und Gerät und hat daher die Auswirkung, dass Nutzer eine negative Wahrnehmung von der Sprachsteuerung oder sogar vom gesamten System haben. Um dem entgegenzuwirken, wird in [32] versucht, Sprachbefehle in die Benutzeroberfläche zu integrieren, um somit Nutzer mit der sprachlichen Interaktion mit dem System vertraut zu machen und Kommunikationsfehler zu vermeiden. Srinivasan et. al. implementierten dafür ein multimodales Interface für ein einfaches Programm zur Bildbearbeitung. Nutzer sollen dazu in der Lage sein, sowohl mittels Touchscreen, als auch mithilfe von Sprachbefehlen mit dem System zu interagieren.

In der Umsetzung wurden drei verschiedene Interfaces implementiert und miteinander verglichen:

- *Exhaustive* : Der Nutzer kann ein Fenster aufrufen, auf dem alle möglichen Sprachbefehle aufgelistet sind. Um die kognitive Belastung auf den Nutzer so gering wie möglich zu halten, werden nur Befehle eingeblendet, die in der jeweiligen Situation verwendbar sind.

- *Adaptive* : Es werden Situationsabhängige Sprachbefehle im Bezug auf einzelne UI-Objekte empfohlen. Der Nutzer kann durch einfaches Zeigen auf ein UI-Objekt eine Liste von Sprachbefehlen auftauchen lassen. Das System versucht auf Basis vergangener Befehle vorherzusehen, welche Befehle der Nutzer als nächstes verwenden wollen könnte. Diese werden dann in der Liste angezeigt.
- *Embedded* : In dem Teil der Benutzeroberfläche, mit dem der Nutzer interagiert, werden neben den UI-Elementen auch die zugehörigen Sprachbefehle angezeigt.

Zur Evaluation des Systems wurden die unterschiedlichen Interfaces in einer Nutzerstudie ( $N = 16$ ) getestet und miteinander verglichen. Um die Testbedingungen so realistisch wie möglich zu gestalten, wurde die Plattform UserTesting [3] verwendet. Die Teilnehmer haben kein ausführliches Training mit dem System erhalten und haben es auf ihren eigenen Geräten getestet.

Am ende der Studie wurde festgestellt, dass die Interfaces *Adaptive* und *Embedded* deutlich mehr zur sprachlichen Interaktion mit dem System angeregt haben, als das Interface *Exhaustive*.

Des Weiteren hatte die Empfehlung von Befehlen hauptsächlich zwei Auswirkungen auf die Nutzer: Zum einen hatten die Empfehlungen zur Folge, dass nur ein recht kleiner Anteil aller Spracherkennungsfehler (18%) daran lagen, dass Nutzer sich falsch ausgedrückt haben.

Zum anderen haben sich Nutzer dank der Empfehlungen dazu angeregt gefühlt, die Spracherkennungsfunktion zu nutzen und hatten trotz hoher Fehlerquote in der Spracherkennung (44%) eine positive Wahrnehmung vom Spracherkennungsaspekt des Systems.

### 3.3 Touch gestures

Dank ihrer Kompaktheit sind Smartphones und vor allem Tablets für das Lesen von Dokumenten wie geschaffen. Die Suche nach relevanten Dokumenten erfordert jedoch häufig die Eingabe von Schlüsselbegriffen von Seiten des Nutzers. Diese Methode mag zwar effektiv sein, wurde jedoch speziell unter der Annahme entwickelt, dass der Nutzer eine Tastatur zur Eingabe benutzt.

Mit der Hypothese, dass es eine für Smartphone- und Tablet-nutzer Benutzerfreundlichere Methode zur Suche relevanter Dokumente gibt, haben Beltran et. al. *BINGO* [7] entwickelt:

BINGO soll dem Nutzer die Suche nach relevanten Dokumenten mithilfe von Wischgesten zu ermöglichen. Nutzer Bewerten dafür Dokumente die ihnen vom System empfohlen werden, indem sie es an den rechten Bildschirmrand wischen, falls das Dokument relevant für ihre Recherche war, und an den linken, falls nicht.

Um dem Nutzer genaueres Feedback zu ermöglichen, generiert das System fünf Schlüsselbegriffe (“Reason bins“) aus dem Dokument, die auf beiden Seiten des

Displays angezeigt werden. Der Nutzer kann das Dokument dann in eines dieser “Reason bins” wischen, falls der darin enthaltene Schlüsselbegriff besonders (un-)nützlich für seine Recherche ist. Diese zusätzliche Information wird dann bei der Empfehlung von Dokumenten an den Nutzer berücksichtigt.

*Umsetzung* : Zur technischen Umsetzung wird das TF-IDF-Maß verwendet. Dabei wird die Vorkommenshäufigkeit von Worten in Dokumenten ermittelt, um dem Nutzer relevante Dokumente zu empfehlen und “Reason bins” für diese zu generieren.

*Evaluation* : BINGO wurde in verschiedenen Szenarien und Aspekten mit zwei weiteren Methoden verglichen:

In einer Nutzerstudie ( $N = 20$ ) sollten Nutzer aus einem Datensatz (je 2.035 bzw. 44.150 Dokumente) möglichst viele Dokumente passend zu einem bestimmten Thema finden und speichern. Die Teilnehmer haben dafür neben BINGO folgende Systeme verwendet:

- *SWP* : Wie BINGO ohne “Reason bins”. Nutzer wischen nach rechts, falls sie das Dokument relevant finden, sonst nach links.
- *KWD* : Nutzer beschreiben das Dokument, indem sie Schlüsselbegriffe selber schreiben und das Dokument anschließend bewerten.

Die Bewertung von Dokumenten mit *SWP* war im Vergleich zur Bewertung mit *KWD* deutlich schneller und intuitiver, jedoch waren die Dokumentempfehlungen von *KWD* deutlich relevanter für die Nutzer. Es hat sich herausgestellt, dass *BINGO* nicht nur ein Kompromiss zwischen *SWP* und *KWD* ist, sondern sich den Stärken beider Methoden annähert:

Während die Durchschnittliche Zeit zwischen einzelnen Bewertungen in *BINGO* sehr ähnlich zu den Zeiten in *SWP* waren, war die Präzision von *BINGO* (Anzahl gespeicherte Dokumente / Anzahl gelesene Dokumente) sehr ähnlich zu der von *KWD*. Auch in einer anschließenden Nutzerbefragung wurde das von den Nutzern bestätigt.

*BINGO* sei wie *SWP* sehr intuitiv (*BINGO* bewertet mit 7, *SWP* mit 7.6 von 10), während es wie *KWD* relevante Dokumente empfiehlt (*BINGO* bewertet mit 7.5, *KWD* mit 7.5 von 10).



## 4 Fazit

### 4.1 Themen

Ein Großteil der Forschung richtet sich auf die Lösung von Problemen im Bereich “Information Retrieval“ mithilfe von Recommender Systems. Der Grund, weshalb gerade Recommender Systems zentral zur Forschung in diesem Gebiet sind, liegt am weit verbreiteten Einsatz von Recommender Systems im elektronischen Handel.

Auf der Konferenz vorgestellte Paper beschäftigen sich dabei vor allem mit der Verbesserung der “User Experience“ bei der Interaktion mit dem System und kommen häufig zum Entschluss, dass die Genauigkeit und Nachvollziehbarkeit von Empfehlungen für die Wahrnehmung eines Recommender Systems von großer Bedeutung sind.

Anders verhält es sich bei der Forschung im Bereich “Ubiquitous Computing“ und “Multimodal Interfaces“:

Viele Paper versuchen eine sinnvolle Verwendung für tragbare Geräte und multimodale Interfaces zu finden, indem sie versuchen, diese in verschiedene Situationen zu integrieren. Durch die Allgegenwärtigkeit und intuitive Bedienbarkeit von Smartphones erweist es sich als schwer, Alltagssituationen zu finden, in denen Nutzer tragbare Geräte und multimodale Interaktion einem Smartphone mit Touchscreens vorziehen würden.

Aus diesem Grund konzentriert sich die Forschung auf spezifischere Situationen: Tragbare Geräte werden deshalb zur Präzisierung assistiver Technologien verwendet, während multimodale Interaktion häufiger zur Bewältigung komplexer Aufgaben eingesetzt wird.

### 4.2 Gemeinsamkeiten und Unterschiede

Gemeinsamkeiten und Unterschiede zwischen den Papern der Konferenz sind stark vom vorgestellten Thema abhängig. Die Paper zu Recommender Systems beispielsweise, stellen die technische Umsetzung neuer Methoden und Algorithmen vor. Das hat zur Folge, dass die Methoden und Algorithmen einfach unter Verwendung von Datensätzen evaluiert werden können.

In Gebieten wie “Ubiquitous Computing“ und “Multimodal Interfaces“ sind stattdessen Nutzerstudien nötig, da hier in erster Linie verschiedene Einsatzmöglichkeiten untersucht werden.

Die Notwendigkeit von Nutzerstudien ist besonders dann ein Problem, wenn sich die Forschung auf sehr spezifische Szenarien richtet:

Bei der Evaluation assistiver Systeme für hörgeschädigte Personen beispielsweise, ist es schwer, Studienteilnehmer zu finden, die die Gebärdensprache können.

## Literatur

1. Amazon sold tens of millions of echo devices in 2018. <https://www.cnet.com/news/amazon-sold-tens-of-millions-of-echo-devices-in-2018/>. Accessed: 2019-06-29.
2. Intelligent user interfaces. <https://iui.acm.org/2019/>. Accessed: 2019-06-18.
3. Usertesting. <https://www.usertesting.com/>. Accessed: 2019-06-30.
4. Alfarez Abdul-Rahman and Stephen Hailes. A distributed trust model. In *Proceedings of the 1997 workshop on New security paradigms*, pages 48–60. ACM, 1998.
5. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, (6):734–749, 2005.
6. Paritosh Bahirat, Yangyang He, Abhilash Menon, and Bart Knijnenburg. A data-driven approach to developing iot privacy-setting interfaces. In *23rd International Conference on Intelligent User Interfaces*, pages 165–176. ACM, 2018.
7. Juan Felipe Beltran, Ziqi Huang, Azza Abouzied, and Arnab Nandi. Don’t just swipe left, tell me why: Enhancing gesture-based feedback with reason bins. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 469–480. ACM, 2017.
8. Syed Masum Billah, Vikas Ashok, and IV Ramakrishnan. Write-it-yourself with the aid of smartwatches: A wizard-of-oz experiment with blind people. In *23rd International Conference on Intelligent User Interfaces*, pages 427–431. ACM, 2018.
9. Pradipta Biswas and Jeevithashree DV. Eye gaze controlled mfd for military aviation. In *23rd International Conference on Intelligent User Interfaces*, pages 79–89. ACM, 2018.
10. Cong Chen, Ajay Chander, and Kanji Uchino. Guided play: digital sensing and coaching for stereotypical play behavior in children with autism. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 208–217. ACM, 2019.
11. Ning Chen, Jialiu Lin, Steven CH Hoi, Xiaokui Xiao, and Boshen Zhang. Arminer: mining informative reviews for developers from mobile app marketplace. In *Proceedings of the 36th International Conference on Software Engineering*, pages 767–778. ACM, 2014.
12. Vicente Dominguez, Pablo Messina, Ivania Donoso-Guzmán, and Denis Parra. The effect of explanations and algorithmic accuracy on visual recommender systems of artistic images. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 408–416. ACM, 2019.
13. Ivania Donoso-Guzmán and Denis Parra. An interactive relevance feedback interface for evidence-based health care. In *23rd International Conference on Intelligent User Interfaces*, pages 103–114. ACM, 2018.
14. Lukas Eberhard, Simon Walk, Lisa Posch, and Denis Helic. Evaluating narrative-driven movie recommendations on reddit. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 1–11. ACM, 2019.
15. Ian MacKenzie et. al. How retailers can keep up with consumers. <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers>. Accessed: 2019-06-18.
16. Kristina Höök. Steps to take before intelligent user interfaces become real. *Interacting with computers*, 12(4):409–426, 2000.
17. Joseph A Konstan and John Riedl. Recommender systems: from algorithms to user experience. *User modeling and user-adapted interaction*, 22(1-2):101–123, 2012.

18. Pigi Kouki, James Schaffer, Jay Pujara, John O'Donovan, and Lise Getoor. Personalized explanations for hybrid recommender systems. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 379–390. ACM, 2019.
19. Thalmic Labs. Myo gestensteuerungsarmband - schwarz. <https://www.robotshop.com/de/de/myo-gestensteuerungsarmband-schwarz.html>. Accessed: 2019-06-26.
20. Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80, 2003.
21. Gregory D Linden, Jennifer A Jacobi, and Eric A Benson. Collaborative recommendations using item-to-item similarity mappings, July 24 2001. US Patent 6,266,649.
22. Leo Neat, Ren Peng, Siyang Qin, and Roberto Manduchi. Scene text access: a comparison of mobile ocr modalities for blind users. 2019.
23. John O'Donovan and Barry Smyth. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174. ACM, 2005.
24. Sharon Oviatt and Philip Cohen. Perceptual user interfaces: multimodal interfaces that process what comes naturally. *Communications of the ACM*, 43(3):45–53, 2000.
25. Martin Kuhl Max Leonhardt Christoph Schröer Achim Völz Lukas Weinel Christian Wigger Christof Wolke Marius Wybrands Patrick Bruns, Christian Eilts. *Datenstrombasierte Recommender-Systeme*. Carl von Ossietzky Universität Oldenburg.
26. Prajwal Paudyal, Ayan Banerjee, and Sandeep KS Gupta. Sceptre: a pervasive, non-invasive, and programmable gesture recognition technology. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 282–293. ACM, 2016.
27. Prajwal Paudyal, Junghyo Lee, Ayan Banerjee, and Sandeep KS Gupta. Dyfav: Dynamic feature selection and voting for real-time recognition of fingerspelled alphabet using wearables. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 457–467. ACM, 2017.
28. Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.
29. Edwina L Rissland. Ingredients of intelligent user interfaces. *International Journal of Man-Machine Studies*, 21(4):377–388, 1984.
30. Monica Sebillo, Giuliana Vitiello, and Maria De Marsico. *Multimodal Interfaces*, pages 1838–1843. Springer US, Boston, MA, 2009.
31. Joan E. Solsman. Youtube's ai is the puppet master over most of what you watch. <https://www.cnet.com/news/youtube-ces-2018-neal-mohan/>. Accessed: 2019-06-18.
32. Arjun Srinivasan, Mira Dontcheva, Eytan Adar, and Seth Walker. Discovering natural language commands in multimodal interfaces. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 661–672. ACM, 2019.
33. Statista. Endkundenabsatz von smartphones weltweit von 2007 bis 2018. <https://de.statista.com/statistik/daten/studie/12856/umfrage/absatz-von-smartphones-weltweit-seit-2007/>. Accessed: 2019-06-29.
34. Tavita Su'a, Sherlock A Licorish, Bastin Tony Roy Savarimuthu, and Tobias Langlotz. Quickreview: A novel data-driven mobile user interface for reporting problematic app features. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 517–522. ACM, 2017.
35. Alex Waibel, Minh Tue Vo, Paul Duchnowski, and Stefan Manke. Multimodal interfaces. *Artificial Intelligence Review*, 10(3-4):299–319, 1996.

36. Mark Weiser. The computer for the 21 st century. *Scientific american*, 265(3):94–105, 1991.
37. Qian Zhang, Dong Wang, Run Zhao, and Yinggang Yu. Myosign: enabling end-to-end sign language recognition with wearables. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 650–660. ACM, 2019.