Peter Grasch

Speech-based Recommender Systems

Master's Thesis

Graz University of Technology

Institute for Software Technology

Supervisor: Univ.-Prof. Dipl-Ing. Dr.techn. Alexander Felfernig

Graz, April 2015

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, ____________________ ____________________

Date Signature

Eidesstattliche Erklärung[1]

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ____________________ ____________________

Datum Unterschrift

[1] Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Abstract

The aim of this thesis is to examine the application of spoken natural language input for knowledge-based recommender systems.

A prototype of a speech-based unit critiquing system, ReComment, was developed and compared to a traditional, baseline system, using an empirical study. It was shown, that the more precise preference articulation afforded by spoken language input allowed ReComment to recommend significantly better fitting products in substantially fewer interaction cycles than the baseline system.

Expanding on the hypothesis that speech-based interfaces facilitate more efficient preference elicitation, the impact of meta-information naturally present in spoken language, such as intonation and phrasing, was further analyzed. The developed prototype, SpeechRec, a conversational, knowledge-based recommender system, engages users in a human-like, mixed-initiative, spoken natural language dialog and incorporates lexical and paralinguistic polarity into its recommendation strategy. An empirical study was conducted, comparing SpeechRec with a reduced version of itself, which retained the novel user interface but limited its recommendation strategy to that of a traditional knowledge-based recommender. It was shown, that identifying and utilizing the additional nuances of spoken natural language input can further improve recommendation efficiency.

It was concluded, that harnessing natural language input, and the rich meta-information it naturally contains, can facilitate a more precise user preference model, and therefore greatly enhance recommendation performance.

Kurzfassung

Die vorliegende Arbeit untersucht die Nutzung von natürlichsprachlicher Interaktion für wissensbasierte Empfehlungssysteme.

Ein Prototyp eines sprachbasierten Unit Critiquing Systems, ReComment, wurde entwickelt und in einer empirischen Studie mit einem traditionellen System verglichen. Es wurde gezeigt, dass die von der Spracheingabe erleichterte Präferenzspezifikation es Nutzern erlaubte, mit ReComment signifikant besser passende Produkte in wesentlich weniger Interaktionszyklen zu finden, als mit dem Vergleichssystem.

Aufbauend auf der Hypothese, dass sprachbasierte Benutzerschnittstellen effizientere Spezifikation von Präferenzen ermöglichen, wurde weiters die Nützlichkeit von Zusatzinformationen von gesprochener, natürlicher Sprache, wie etwa Aussprache oder Formulierung, untersucht. Der entwickelte Prototyp, SpeechRec, ein interaktives, wissensbasiertes Empfehlungssystem, verwickelt Nutzer in einen natürlichsprachlichen, gesprochenen, mixed-initiative Dialog, der dem eines menschlichen Verkäufers ähnelt. SpeechRec integriert dabei lexikalische und paralinguistische Polarität in die Empfehlungsstrategie. In einer empirischen Studie wurde das System mit einer reduzierten Version von sich selbst verglichen, welche die neuartige Benutzerschnittstelle beibehielt, aber eine auf den Funktionsumfang traditioneller wissensbasierter Systeme eingeschränkte Empfehlungsstrategie verwendete. Es wurde gezeigt, dass das Identifizieren und Verwenden der zusätlichen Nuancen von gesprochener, natürlichsprachlicher Eingabe die Effizienz des Empfehlungssystems weiter steigern kann.

Zusammenfassend wurde geschlussfolgert, dass die Nutzung von natürlichsprachlicher Interaktion, und der darin enthaltenen Metainformationen, ein informationsreicheres Präferenzmodell ermöglicht, welches die Empfehlungsleistung substantiell verbessern kann.

Acknowledgments

This thesis would not have been possible without the continued support and invaluable expertise of my advisor, Prof. Dr. Alexander Felfernig, for which I am deeply grateful.

I also want to express my gratitude towards Florian Reinfrank, for his help during the development of ReComment, as well as towards all participants of the various studies conducted in the course of this research.

I would further like to thank my family and friends, whose feedback and enduring encouragement helped shape this work into what it has become.

Finally, I am sincerely thankful towards Valeriya Zakharova for proof-reading drafts of this thesis.

Contents

Abstract v

1. Introduction and Motivation 1
1.1. Introduction 1
1.2. Motivation 4
1.3. Related Work 4

2. Speech-based Unit Critiquing 7
2.1. System Description 7
2.1.1. Product Database 7
2.1.2. Spoken Language Input 8
2.1.3. User Interface 11
2.1.4. Recommender 15
2.2. Study Design 21
2.2.1. Compared Systems 21
2.2.2. Test Demography 21
2.2.3. Task Definition 22
2.2.4. Evaluation 23
2.3. Results 23
2.3.1. Input Processing 24
2.3.2. Usability 26
2.3.3. Recommendation Performance 27
2.4. Conclusion 28

3. Towards Human-like Recommender Systems 29
3.1. System Description 30
3.1.1. Product Database 30
3.1.2. Spoken Language Input 33
3.1.3. User Interface 40

3.1.4. Recommender 42
3.2. Study Design 46
3.2.1. Compared Systems 47
3.2.2. Test Demography 48
3.2.3. Task Definition 48
3.2.4. Evaluation 49
3.3. Results 50
3.3.1. Input Processing 50
3.3.2. Usability 51
3.3.3. Recommendation Performance 51
3.4. Conclusion 52

4. Conclusion **55**

5. Future Work **59**

A. Speech-based Unit Critiquing **65**
A.1. Language Definition 65
A.2. Product Attributes 68
A.3. Questionnaire 69

B. Towards Human-like Recommender Systems **75**
B.1. Language Definition 75
B.2. Product Attributes 82
B.3. Aspects 83
B.4. Questionnaire 85

Bibliography **93**

1. Introduction and Motivation

1.1. Introduction

The information age has given rise to ever larger collections of data, which have become increasingly difficult to navigate. In order to meet this challenge, a multitude of systems have been proposed to structurize large volumes of information, and to ascertain, codify and act on a given user's request. One notable category of such systems is that of recommender systems.

Recommender systems elicit a user's preferences, marshal this insight to form a user model, and then employ this model to recommend items. As the system handles the translation of high-level preferences to concrete products, recommender systems can facilitate efficient navigation in large and complex domains, which may even be unknown to the user.

First recommender systems were outlined as early as 1989, and a multitude of different approaches have since been proposed, many of which have found footing in a variety of domains [19]. In general, the following archetypes can be distinguished [5, 15, 28].

Collaborative filtering systems, or social filtering systems, constitute the arguably most well known and most common class of recommender systems. These systems recommend products based on the actions of "similar" users, where similarity is determined by comparing the users' interaction histories.

Content-based systems, or content-based filtering systems, use a user's previous interaction with the system to recommend items that are most "similar" to the products that the user has expressed an interest in previously, often explicitly by rating them, or implicitly by interacting with them. Product similarity is determined by (commonly rudimentary) domain knowledge of the system.

Knowledge-based systems employ comparatively extensive domain knowledge, as well as structured information about the user's true, hidden preferences to enable informed recommendations.

Collaborative filtering and content-based systems commonly see wider deployments because they largely avoid the potentially laborious knowledge engineering and even more crucially do not usually require structured user input. On the other hand, knowledge-based approaches suffer less from ramp up issues for new users and products for which no interaction history has yet been recorded. Additionally, their transparent and intelligible recommendation strategy makes them viable even for application domains where trust is crucial [4, 5, 9, 16]. Most importantly, knowledge-based recommenders base their recommendation strategy on the factual realities of the application domain and the users articulated preferences, instead of virtually relying on the observed decisions of other (human) users. It seems therefore reasonable to assume that a fictional, perfect knowledge-based recommender would outperform its equally perfect collaborative filtering and content-based counterparts.

However, creating effective knowledge-based recommender systems remains hard in practice. Building accurate domain and user models is difficult for all but trivial domains, especially because a user's preferences are commonly incomplete, often contradictory and subject to rapid change [31]. As a result, most practical knowledge-based recommenders are conversational recommender systems, meaning that their recommendations are iteratively refined through continued user input. Optimizing preference elicitation has nevertheless remained a topic of ongoing research [4, 9].

Generally, one distinguishes between the following type of feedback elicitation. Search based systems, where the user model is composed of answers to domain questions posed by the recommender, and example based systems, which use sample items to guide user interaction [25, 29, 39]. A particularly notable subtype of the latter employs directional user feedback on recommendations, so called "critiques", to navigate the domain. For example, in an e-commerce setting the user may respond to a recommendation by articulating the critique "cheaper", which the system would process to the constraint $price < x$, where x refers to the price of the currently displayed product [7, 9]. Critiquing-based recommender systems have received significant academic interest over the last decade, because they have repeatedly shown to provide good recommendation quality while requiring relatively little cognitive effort from the user [8, 24, 29]. However, simple critiques reveal comparatively little information about a user's hidden preferences to the user model, often resulting in overly long interaction sessions [8, 24, 29]. In an effort to curb this problem, several approaches have been proposed, which aim at increasing the utility of critiquing input.

Some systems incorporate the critiquing history into their recommendation strategy. Incremental critiquing systems iteratively refine their understanding of the user's needs by selecting products, which best satisfy the current, as well as past critiques [33]. Experienced-based critiquing systems even hark back to collaborative filtering approaches by searching for similar critiquing sessions of potentially different users and suggesting products that were ultimately accepted during these sessions [22].

Orthogonally to drawing conclusions from the critiquing history, the selection of critiquing options to present to the user has been heavily scrutinized. Simple critiques, targeting a single product attribute are often referred to as "unit critiques" [5]. Conversely, "compound critiques" target more than one attribute at a time, such as "cheaper and lighter" [21]. In order to provide a user the option of issuing some of the exponentially many compound critiques in resource constrained traditional user interfaces, significant research has been devoted to determining an optimal set of compound critiquing options to present to a user. Noteworthy examples include the Apriori method, that suggest critiques based on the remaining unexplored search space [23], and especially the concept based on multi-attribute utility theory (MAUT) presented in [45], which tries to build compound critiques that lead to products that would match the user's earlier critiques. While compound critiques can mitigate the problem of excessively long critiquing sessions to an extend, they also increase cognitive load [21].

The use of spoken natural language input in recommender systems could potentially sidestep some of the issues in knowledge-based recommender systems outlined above, yet has surprisingly remained largely unexplored.

1.2. Motivation

Surveying the current state of the art of knowledge-based recommender systems, it becomes apparent that the inherent tradeoff of maximizing the utility of user input, while minimizing the required cognitive effort, has been the subject of much research, yet has rarely been openly acknowledged.

In part, this tradeoff is an inherent property of preference elicitation. As the most crucial information for a recommender system is generally that, which maximally reduces the remaining relevant product space, the most valuable type of feedback is therefore commonly also a user's most decisive compromise. Such decisions naturally afford higher cognitive involvement than easier choices. However, while the trend towards simpler feedback is understandable, traditional recommender systems arguably reinforce this behavior by making it exceedingly difficult or even impossible to specify more complex types of feedback, even if that would more accurately reflect the user's true preferences.

To address this problem, I propose the use of speech-based natural language interfaces for knowledge-based recommender systems, which allow users to specify arbitrarily complex preferences in a more natural, and therefore more efficient manner, than systems employing traditional interfaces.

1.3. Related Work

When describing their seminal FindMe system, and discussing a potential thought processes ultimately responsible for issuing a critique in a movie recommender system, Burke et al, mentioned that a future user may think "That would be good, but it is too violent for my kids." Using their system, this notion would ultimately be codified as a "lower" critique on the "level of violence"-attribute. It is remarkable to note, that this very abstraction step would not only be unnecessary in a natural-language based interface, potentially decreasing cognitive load, but that it also misses other, potentially valuable information about the user's true preferences to find a movie acceptable for their kids [7].

While the use of spoken natural language input in the domain of recommender systems has been envisaged before, the various implementations commonly had limitations that diminished their potential advantages over traditional interfaces in often drastic ways.

An early prototype of a speech-based recommender system, the Adaptive Place Advisor, can be found in [42]. However, as the speech input in the Adaptive Place Advisor is primarily limited to providing concrete attribute values in response to questions posed by the system, the system draws little benefit from it. A similarly constrained system is discussed in [2].

Shimazu's ExpertClerk outlined in [39] uses written natural language input. The system engages the user in a dialog by asking domain questions until the search space is narrowed down to a sufficiently short list of viable products. The system then presents the user with a set of three maximally different options. Further navigation is possible by critiquing any of the displayed items. A textual natural language recommender system was further discussed by Wärnestål in [44], focusing on optimizing the dialog system, with equally promising results.

The first prototype presented in this thesis, outlined in Chapter 2, uses unit critiquing, building on the basic recommendation strategy of Burke's et al. FindMe systems [7]. The system recommends a given item, upon which the user can issue a critique, completing the feedback cycle and causing the system to recommend a new item. Critiques articulated in earlier interaction cycles are stored and respected when possible, following the incremental critiquing approach proposed in [33]. Instead of permanently removing previously recommended items from the search space, the recommender introduces a temporary bias against them to avoid the problem of diminishing choices while addressing the unreachability problem [24, 26]. The developed prototype additionally uses products sales ranks sourced from a popular online retailer to slightly favor popular products. This could be compared to utilizing information of other users' recommendation sessions as described in [22], but foregoing all but the information that a given product had or had not been accepted by other users.

As early studies showed that users would, given no restrictions, routinely also give feedback on product attributes not commonly incorporated in traditional recommender systems, such as how nice a product looked, we followed the principal algorithm presented by Moghaddam et al. in [27] to extract user sentiment from customer reviews to augment the product data used by the second prototype, presented in Chapter 3, with information extracted from customer reviews. In contrast to Dong's et al system outlined in [12], our prototype uses customer sentiment only to enrich the database, and not to replace factual information.

2. Speech-based Unit Critiquing

In this chapter, ReComment[1], a speech-based, critiquing-driven recommender system is introduced.

The developed prototype for the domain of digital compact cameras is discussed, and compared against an identical recommender system employing a traditional user interface, using an empirical study. It is shown, how spoken natural language input leads to a significant reduction in critiquing cycles, while increasing recommendation quality.

2.1. System Description

In this section, the developed prototype of ReComment is discussed.

2.1.1. Product Database

A selection of over 600 digital compact cameras currently on the market was collected. For every product, ReComment's database stores at least one product image, the actual retail price[2], as well as a range of 12 attributes. After collecting feedback from early testers, 9 of these attributes were selected for inclusion in the final prototype. More information can be found in Table 2.1. A complete list of recorded product attributes is shown in Section A.2.

[1] ReComment is a portmanteau of "Recommend" and "Comment".
[2] The popular Austrian price comparison website `http://geizhals.at` was used to determine product prices.

Attribute	Included in Final Prototype
Model	Yes
Manufacturer	Yes
Price (€)	Yes
Resolution (megapixel)	Yes
Sensor size (inches)	Yes
Sensor type	Yes
Size (w×h×d)	Yes
Weight (gram)	Yes
Internal memory (megabyte)	No
Digital zoom (times)	No
Optical zoom (times)	Yes
External storage	No
Product sales rank	Internally, see Section 3.1.4

Table 2.1.: Product attributes. (Translated from German.)

2.1.2. Spoken Language Input

Decoding spontaneous speech is still a largely open problem. Repetitions, false starts or self-interruptions, filler words, as well as dialectal and emotive speech are all common [40]. Additionally, any successful natural language speech-based interface further has to handle non-speech data such as laughs, coughs, lip smacks, and other kinds of background noise.

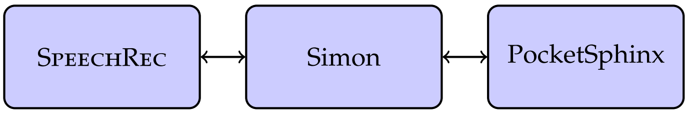

Figure 2.1.: Speech processing architecture.

In an effort to cope with such realities, a custom, task and domain specific automatic speech recognition (ASR) system for Austrian German was developed, using Simon[3] and the CMU SPHINX speech recognition framework[4]. An overview of the used architecture can be found in Figure 2.1.

[3] http://simon.kde.org
[4] http://cmusphinx.sourceforge.net/

In general, any speech model consists of two major components. The language model (LM) contains the phonetic dictionary of all words to be recognized, as well as a representation of context-dependent statistical word observation probabilities (N-Gram). The acoustic model (AM) describes how the individual sounds (phonemes) are expected to be pronounced by users of the system. The following subsections describe the realization of these components in the speech input layer of ReComment.

Language Model

A custom trigram language model, specifically tailored to the domain of digital cameras, was created. Internally, this model was constructed from a combination of three separate submodels, built from the following sources.

Written German, a corpus of standard German, was collected from a database dump of the German Wikipedia. Common abbreviations were expanded. The corpus was limited to its 25,000 most common words and heavily pruned to about one million 3-grams.

Non-numeric critiquing sentence fragments, a synthetic corpus of sentence fragments that the natural language parser described in Section 2.1.2 would accept, limited to those fragments which do not refer to concrete numbers, such as "cheaper" or "a bit more zoom", was generated.

Numeric critiquing sentence fragments, another synthetic corpus of understood critiquing sentence fragments, this time limited to exactly those fragments which do refer to explicit values, such as "less than 300 euros", was also collected.

The combination of the numeric and non-numeric critiquing sentence fragments corpora represent the entirety of parsable sentence fragments that ReComment's natural language processing layer accepts. They were split into two distinct corpora as the number of fragments involving concrete values is naturally significantly higher than of those that don't ("less than 100 euros", "less than 101 euros", etc.) and their collection in the same corpus would therefore significantly bias the language model in its favor.

The three individual language models were combined to a final model using linear interpolation. The mixing coefficients (mixture weights), represented as λ, with $\sum_i \lambda_i = 1$, were selected based on experiments on recorded user interactions from a pilot study. The selected coefficients can be found in Table 2.2.

λ	**Corpus**
0.1	Written German
0.5	Non-numeric critiquing sentence fragments
0.4	Numeric critiquing sentence fragments

Table 2.2.: Selected language model mixture weights.

Acoustic Model

ReComment's acoustic model is based on the GPL licensed Voxforge corpus[5], a selection of 19 free audio books from the LibriVox project[6] and the Austrian German data set from the ADABA database[7], kindly made available by the institute of Austrian German[8].

A 3-state continuous hidden Markov model (HMM) was estimated from this data set, which was then adapted to manually transcribed interaction sessions of pilot testers using maximum likelihood linear regression (MLLR) and maximum a posteriori probability (MAP) speaker adaption [11, 20].

Parser

A custom parser was developed that recognizes several hundred different key phrases, a list of which can be found in Table A.1. Details about the parser's implementation can be found in [18].

[5] http://voxforge

[6] http://librivox.org

[7] http://www-oedt.kfunigraz.ac.at/ADABA/

[8] http://www-oedt.kfunigraz.ac.at

2.1.3. User Interface

A simple unit-critiquing user interface (UI) was developed for ReComment. Except for the feedback controls, the traditional, mouse based interface is identical to the speech-based interface. In addition to the graphical user interface discussed below, both interfaces additionally use auditory notifications on recommending a new product, or detecting unfulfillable requirements (no matching product). The speech-based interface furthermore includes a sound effect informing a user, that the system is currently decoding their voice input.

Because ReComment does not prompt the user for an initial search, a user is immediately presented with the first recommendation upon starting the interaction section. Given a newly initialized user model, the prior recommendation probability described in Section 2.1.4 dominates the recommendation process. The initially shown product is therefore the current top seller.

Traditional User Interface

In an effort to draw meaningful conclusions about the potential advantages of a speech-based user interface over a mouse-based one, ReComment's UI closely resembles other traditional critiquing-based knowledge recommender systems, such as the FindMe systems and the Qwikshop system [6, 33].

A screenshot of ReComment's mouse-based user interface is shown in Figure 2.2[9].

As discussed in Section 2.1.4, triggering the "Back" button undoes the user's last completed feedback cycle.

[9]The original interface is in German, text in screenshots has been translated to English.

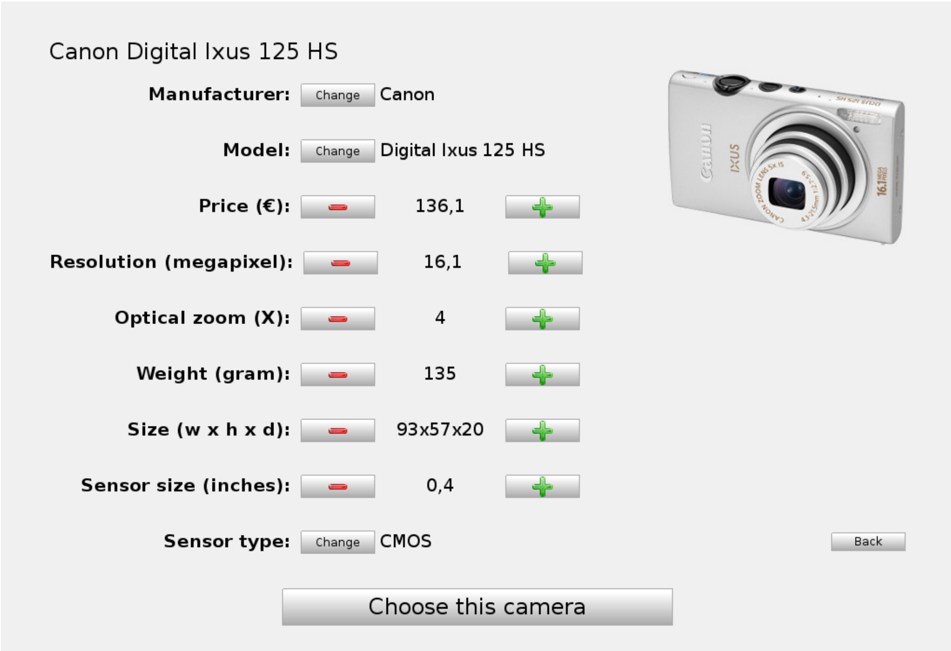

Figure 2.2.: Mouse-based user interface.

Speech-based User Interface

RECOMMENT's speech-based user interface uses a push-to-talk (PTT) interaction paradigm. Instead of continuously listening to user input, the user presses and holds a specific button while they speak, similarly to using a walkie-talkie. While an alternative, automatic voice activity detection was implemented, some users tended to think aloud when presented with a new interface, or ask questions directed at the person conducting the study. As the system's limited natural language understanding subsystem could not reliably distinguish these utterances from those aimed at RECOMMENT, the PTT system was used when conducting the empirical study presented in Section 2.2.4.

The interface of RECOMMENT's speech-based prototype is shown in Figure 2.3[9].

Underneath the PTT control element, an indicator shows the current input

Figure 2.3.: Speech-based user interface.

level of the used microphone. This was included to assure users that ReComment was actively listening to them. The functionality of the traditional user interface's "Back" button is replaced with an equivalent voice command.

As discussed in Section 2.2, particpants of the user study were not told which type of interaction ReComment was designed to understand. Instead, ReComment shows purposefully sparse hints as needed, so as not to influence the user's further interaction with the system more than necessary. System provided example sentences were tailored to formulations the user had already used, whenever possible. Figure 2.4 outlines this process. Any correction hints shown are automatically hidden by the system after at most 3.5 seconds.

The aforementioned measures were taken to minimize user instructions, in an effort to gain insight into what kind of instructions users would naturally use when interacting with a speech-based recommender system. More information on this topic can be found in [18].

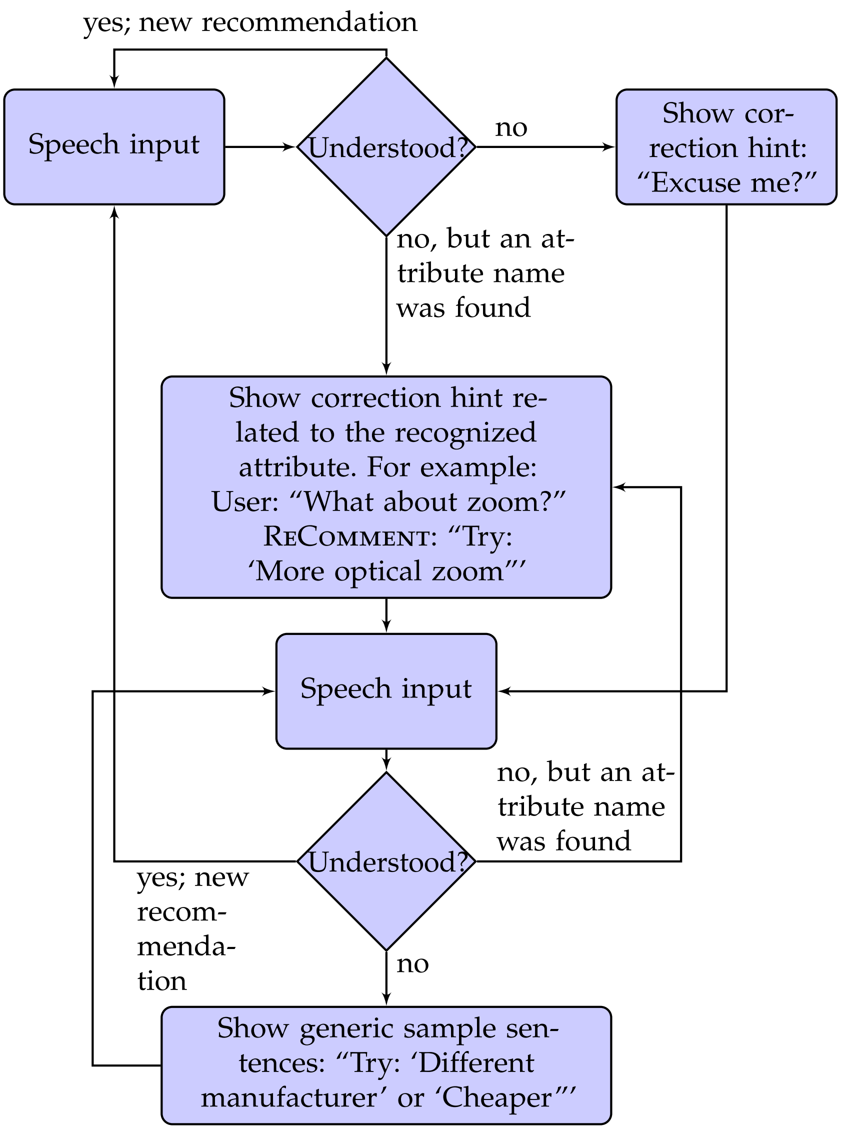

Figure 2.4.: Guiding user input.

2.1.4. Recommender

ReComment is based on a conversational, incremental unit-critiquing recommendation strategy.

User Preference Model

ReComment maintains a user preference model throughout each recommendation session. Issued critiques are stored for up to 15 feedback cycles and their influence on the recommendation strategy is linearly discounted based on their age. Newly introduced critiques are compared to the preference model and replace any earlier critiques that they contradict or refine as suggested in [33].

Some knowledge-based systems may suffer from what is commonly referred to as the "unreachability problem", where items that are "better" in all respects recorded in the domain model prevent the recommender system from ever recommending a factually inferior but potentially more desirable given product. This is especially problematic as the recommender system's view of the domain is inherently an incomplete abstraction. In order to address this problem, ReComment records an additional $product \neq currentProduct$ constraint during each completed feedback cycle as suggested in [24], encoding the assumption that the user issuing a critique is a result of them not being satisfied with the current recommendation. Such added constraints are subject to the aforementioned aging process and thus are eventually removed. As a result, ReComment does not suffer from the so called "diminishing choices" problem, where continued domain exploration limits the potentially viable products from the search space [26].

In response to feedback from pilot testers, ReComment includes the option of undoing feedback cycles by selecting the "Back" option in the traditional interface or giving an equivalent voice command in the speech-based interface, as described in Section 2.1.3. Reverting a feedback cycle restores ReComment's state before the last user input, causing any removed or replaced critiques to be restored, as well as any critiques added in the now undone feedback cycle to be discarded.

Prior Recommendation Probability

The 100 top sellers of the selected product domain of digital compact cameras were sourced from a popular online retailer[10], and their respective sales rank added to the products in ReComment's database. The recommendation strategy treats this sales rank as a prior recommendation probability, and thus slightly favors popular products.

In lieu of an initial search, ReComment therefore defaults to the current top selling digital compact camera for its initial recommendation.

Rate of Change

Speech-based natural language input allows user to specify more expressive forms of feedback than traditional mouse-based user input. For example, users may articulates critiques such as "a little cheaper" or "much cheaper" as part of their natural interaction with ReComment, which both carry more information than a basic "cheaper" critique traditional systems are usually limited to. This additional information was integrated in ReComment's user model with the aim, of enabling the recommender system to more accurately ascertain the user's real preferences and therefore ultimately recommend better fitting products.

In order to distinguish critiques such as "a little cheaper", "cheaper", and "much cheaper", ReComment considers directional critiques not as binary conditions that are either met or not, but as implicitly expressing an attribute's target value. For example, a critique of "more than 50 Euros" would cause ReComment to search for a product which costs around 100 Euros, the deduced implicit goal. The utility score of a given critique is proportional to the distance from the implicit target value expressed by the critique, positive when the constraint is met, negative if it is not. A plot of the utility function of such a critique can be found in Figure 2.5. Critiques on non-numeric attributes such as "Brand", are treated as binary statements with utility 1 if the constraint is met, and 0 otherwise.

[10] http://www.amazon.com

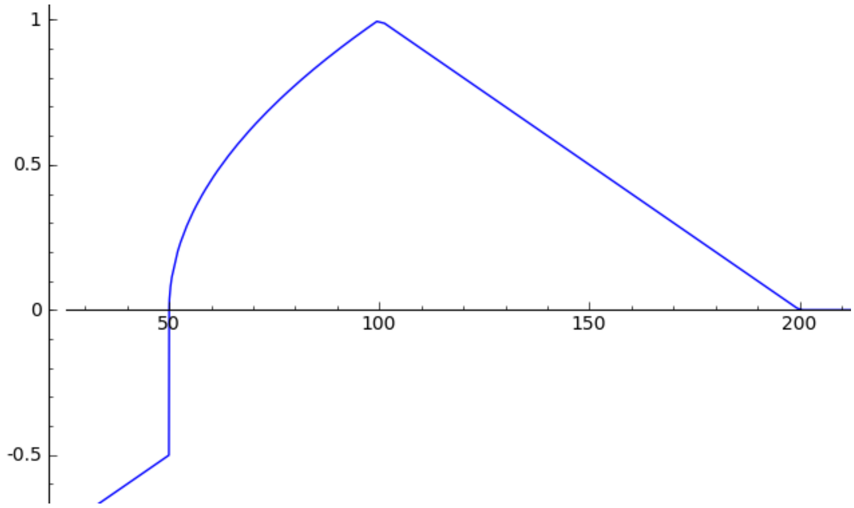

Figure 2.5.: Utility function of the critique $x > 50$.

The distance of the assumed goal from the current or specified attribute value is defined through the critique's "modifier factor", with $distance = modifierFactor * 50\%$[11] and a default modifier factor of 1.0. Sample adjectives that a user may use to affect the modifier factor of the resulting critique are shown in Table 2.3.

Adjective	Modifier factor	Deduced desired change of attribute value
"slightly"	0.2	+10 %
"very"	2.0	+100 %
"not"	-1.0	-50 %

Table 2.3.: Sample adjectives that affect the modifier factor.

In pilot tests, it was found that traditional, linear or triangular acceptance functions would create an optimal region of equal score when given both upper and lower bound for an attribute, such as $(x > 50) \wedge (x < 100)$. ReComment therefore uses a non-linear piecewise function shown in Formula 2.1, which fulfills all aforementioned properties and which the author

[11]Percentages refer either to the attribute value or the deduced goal, whichever is larger. This is done to ensure symmetry between larger-than and smaller-than critiques.

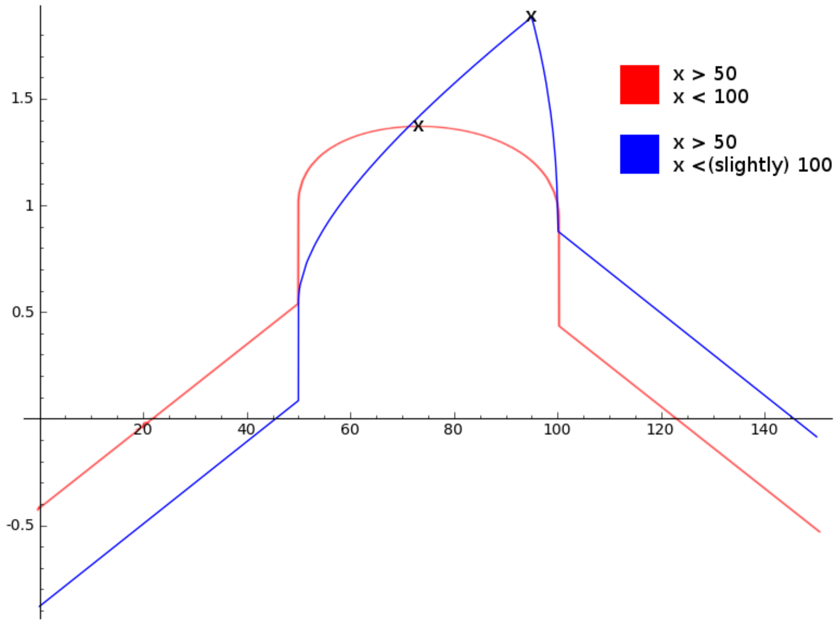

Figure 2.6.: Utility functions of the subsequent critiques $x > 50$ and $x < 100$, as well as $x > 50$ and $x <_{slightly} 100$.

feels better represents a user's intention. Figure 2.6 shows a plot of the effective utility function given the sample critiques above.

$$distance(a,b) = \begin{cases} -distance(b,a) & if\ a < b \\ a & if\ b = 0 \\ \frac{a}{b} - 1 & else \end{cases} \tag{2.1}$$

The algorithm for calculating the utility of a product in respect to a given critique is outlined in Algorithm 1.

Recommendation Strategy

ReComment displays its current recommendation to the user, who either accepts the product, ending the recommendation session, or responds by supplying at least one critique, completing the feedback cycle (iteration).

Input: Product p, relationship r, attribute a, modifier factor m.
Output: Utility u.
$distance = distance(a.value, p[a.id].value) * r.direction$;
$perfectDistance = m * 0.5$;
if *critiqueViolated* **then**
  return $-abs(distance - perfectDistance)$;
else
  if $distance < perfectDistance$ **then**
    return $\sqrt{\frac{distance}{perfectDistance}}$;
  else
    return $max(perfectDistance - distance + 1, 0.0001)$;
  end
end

Algorithm 1: ReComment's principal utility calculation.

Upon completion of a feedback cycle, ReComment adds all specified critiques to the user model. Previously articulated critiques are aged by one cycle, and feedback that has exceeded its lifetime is removed from the user model. ReComment then calculates the utility of all products in the search space, scored against the user model, and returns the item with the overall highest utility as the next recommendation. It is ensured that this product will fulfill at least one of the critiques of the immediately preceding feedback cycle. If no such item can be found, because, for example, the user requested a smaller product than the smallest known product, a warning message is displayed and the unfulfillable critiques added during the last cycle are removed from the user model. The principal recommendation strategy is sketched in Algorithm 2. Refer to Subsection 2.1.4 for details about the user preference model and a description of the used utility function.

Many practical critiquing-based recommender systems employ compound critiques to reduce session length [23]. These critiques act on multiple attributes simultaneously, such as "cheaper and larger sensor". Including compound critiques in ReComment was considered but ultimately rejected for the following reason. As pointed out by Reilly et al in [32], compound critiques are not only more efficient ways of providing feedback, they also reveal information about the domain. Naturally, a mouse-based interface

Input: Known products P, list of critiques C, current recommendation r_{old}.
Output: Next recommendation r_{new}.
$P' \leftarrow \{p \in P | p\ satisfies\ last\ given\ critique\}$;
if P' *is empty* **then**
  show warning and return r_{old};
end
$maxUtility \leftarrow -\infty$;
$bestOffer \leftarrow r_{old}$;
for $p \in P'$ **do**
  $thisUtility \leftarrow \infty$;
  for $c \in C$ **do**
    $thisUtility \leftarrow thisUtility + (1 - \frac{c.age}{MaxAge}) * c.utility(p)$;
  end
  if $thisUtility > maxUtility$ **then**
    $maxUtility \leftarrow thisUtility$;
    $bestOffer \leftarrow p$;
  end
end
return $bestOffer$;

Algorithm 2: Recommendation strategy.

can only hold a limited number of feedback options. The exponential nature of critiquing combinations therefore necessitates the selection of what the system, or the involved knowledge engineer, considers the most useful compound critiques for display. Such selected compound critiquing options often reveal the selection criteria employed. For example, choosing compound critiques which have high support from the product database may produce a compound critiquing option such as "CCD sensor and cheap". Displaying this option in the user interface tells the user that there is a link between the camera using a CCD sensor and it being cheap. Because displaying the same critiquing options in the speech-based system would color user-interaction, and not including them would unfairly bias the evaluation towards the traditional user-interface because of the aforementioned additional information, compound critiquing was not included in ReComment. Compound critiques specified through speech input are instead merely treated as independent unit critiques on the attributes they affect.

In an effort to keep the speech-based prototype as similar to the traditional mouse-based interface used for comparison as possible, ReComment, in contrast to many traditional critiquing-based recommender systems, such as the original FindMe systems, does not use an initial search [7]. Instead, the implicit similarity constraint to the previous recommendation is removed to allow for efficient exploration of the product space. Because of the incremental nature of the recommendation strategy, the recommender's focus is naturally narrowed as the recommendation session progresses and the user model becomes more meaningful. Moreover, the used utility function outlined in Section 2.1.4) discourages erratic jumps in the product space.

2.2. Study Design

To evaluate the hypothesis, that a speech-based, natural language driven critiquing-based recommender system could outperform an equivalent traditional mouse-based one, an empirical study was conducted.

2.2.1. Compared Systems

The viability of ReComment's speech-based interface was assessed by comparing it to a traditional, mouse-based equivalent. Both systems are described in detail in Section 2.1.3.

2.2.2. Test Demography

80 participants, mostly students, were split into two groups of 40 people each. Group A used ReComment's speech-based interface, Group B the traditional interface. Additional demographic information about the study participants can be found in Table 2.4.

Characteristics	Group A	Group B
Male	35	33
Female	5	7
Total	40	40
Median age	24	22
Personally own camera	67.5 %	60 %
Sought help when buying this camera	19.4 %	26.7 %
Consider themselves interested in digital cameras	32.5 %	35 %

Table 2.4.: Demography of the user groups of the empirical study.

2.2.3. Task Definition

Study participants were instructed to imagine themselves looking for a new digital compact camera based on their personal needs, and to use ReComment to help find such a product. They were told to stop when they found a product they could picture themselves purchasing, or when it became clear, that no such product could be found. As only the limited set of attributes described in Section 2.1.1 were known to ReComment, users were asked to ignore any product attributes not shown on screen.

As to the respective ReComment interface, participants received purposefully little instructions. Group B, using the mouse-based interface, was told to "use the buttons" to find a product that fit their requirements. Group A, using the speech-based interface, was equivalently instructed to "use voice commands". Additionally, a small note explained the PTT system as outlined in Section 2.1.3. It is important to note, that users of the speech-based interface did *not* receive any instructions about supported commands or example sentences. In case participants asked the person conducting the study for further instructions, they were told to simply "try it out". As mentioned in Section 2.1.3, these measures were taken to identify what interaction patterns users would naturally use, without being influenced by ReComment's potential limitations.

2.2.4. Evaluation

Immediately after completion of the recommendation session, each participant of the study was presented with a comprehensive questionnaire, which, included questions about their subjective impression of the performance of the used speech recognition subsystem, the system's usability, and the quality of the final recommendation. The full questionnaire can be found in Section A.3.

To evaluate ReComment's usability, the standard system usability survey (SUS) presented in [3] was adapted as follows. Questions referring to "inconsistencies" caused by "various functions" were removed, as they are hardly applicable in a system with such a singular focus. The influence of the remaining questions was uniformly increased to maintain the original scale of at most 100 points. The questions referring to "frequent" repeated use of the system were adapted to instead ask if the participants could see themselves using ReComment before purchasing a digital compact camera.

Where applicable, participants' responses were verified with an inverse control question. Responses where sentiment expressed by question and control question are contradicting were excluded from the evaluation. Study participants were allowed to abstain from any question they did not feel comfortable answering.

All user interaction sessions were logged, and recorded audio was stored for later analysis.

2.3. Results

This section outlines the results[12] of the conducted empirical study.

[12] Reported statistical significance of results of comparative analysis was calculated using Welch's t-tests.

2.3.1. Input Processing

In order to draw meaningful conclusions from the results of the study, it is necessary to ensure that the implementation, especially its challenging natural language processing component, performed adequately. ReComment's input processing was thus analyzed with both objective and subjective performance indicators.

After completion of the study, all participants' interaction sessions were manually described. In addition to the logged recognition results from the study itself, all recordings were also transcribed using Google's speech recognition API[13] for comparison. The transcriptions logged from ReComment's Simon and CMU SPHINX based ASR subsystem discussed in Section 2.1.2, were compared against Google's recognition results as well as the manual reference transcriptions. An error was reported if parsing the recognition hypothesis did not yield identical output to parsing the reference transcription[14]. Hypothesis, whose interpretation was found to be partly correct are reported as "partially correct". As shown in Figure 2.7, our custom speech recognition layer outperforms Google's off-the-shelf online service in both metrics.

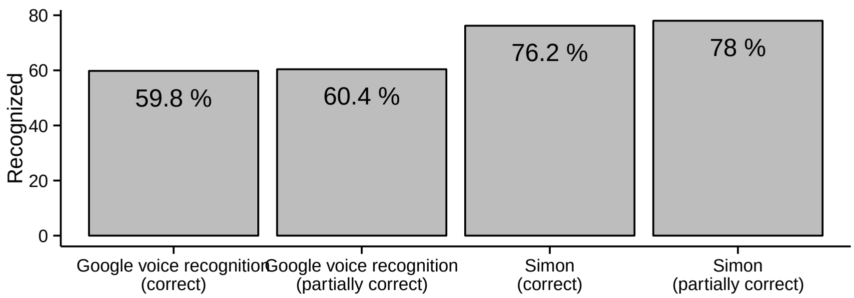

Figure 2.7.: Comparison of the ReComment's speech recognition system with a state of the art off-the-shelf online service.

Study participants that used the speech-based interface of ReComment

[13] http://www.google.com/speech-api/v1/recognize?lang=de

[14] Therefore, e.g., "cheaper, please" and "cheaper, peas" are treated as "identical" recognition results, as both parse to the same critique and their difference has therefore no bearing on ReComment's performance.

were additionally asked to rate the recognition rate of the system as part of the questionnaire. The overall positive result is shown in Figure 2.8.

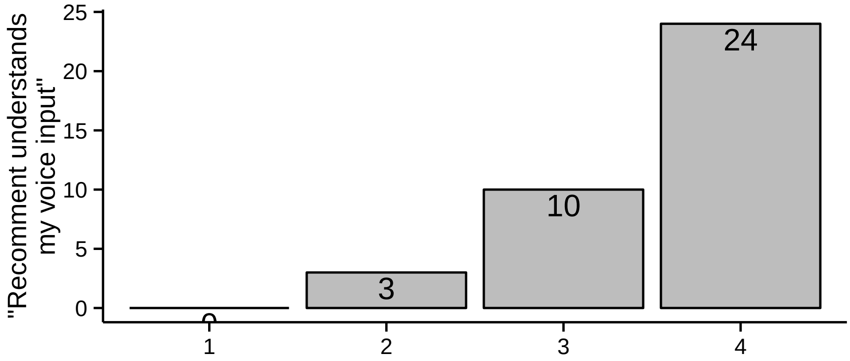

Figure 2.8.: Participants' perception of speech recognition accuracy ([1, 4], higher numbers signal agreement).

To assess the quality of ReComment's parser, the system's interpretation of the study participants' input was verified manually. Out of a total of 384 manually transcribed sentences, 21 sentences, or 5.5 %, failed to parse correctly. Most of these errors can be attributed to unexpected formulations such as "7 optical zoom", but complex constraints like "highest resolution for 120 Euros", expressions, such as "credit card size" or unresolved references like "even less" were also observed.

Sentence
I am looking for a camera with 12 megapixel and a weight of around 200 gram.
This camera with the same properties just smaller.
An even smaller camera.
Optical zoom of 14 times would be better.
More optical zoom.
...

Table 2.5.: Sample user interaction. (Translated from German.)

The initial turns of a transcribed recommendation session can be found in Table 2.5. More details about study participants' preference articulation when interacting with ReComment can be found in [18].

2.3.2. Usability

As proposed in Section 2.2.4, a modified SUS evaluation was used to assess ReComment's usability. SUS scores are reported on three groups. Users of the speech-based interface, users of the mouse-based interface and the subset of users of the speech-based interface that reported good speech recognition performance. The results for the usability evaluation are shown in Figure 2.9.

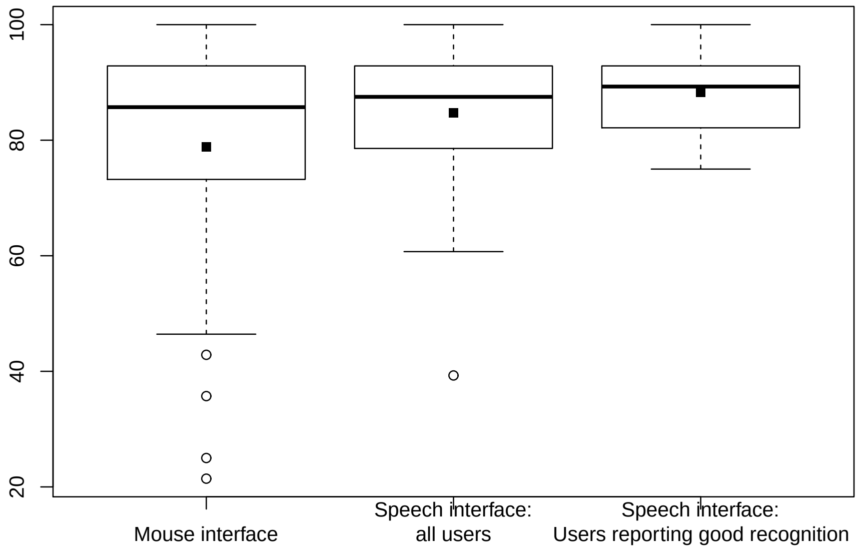

Figure 2.9.: ReComment: Usability evaluation (modified SUS scores; $[25, 100]$, higher is better. Black squares indicate the arithmetic mean.).

While ReComment's speech-based user interface appears to score higher than the traditional interface, this increase is not statistically significant ($p \approx 0.13$). However, the group of users who reported the speech recognition subsystem to work well for them rate the system's usability as significantly better than that of the traditional mouse-based interface ($p < 0.02$). Although cross-correlation between subjective speech recognition performance and reported usability is possible, it seems reasonable to conclude that continued advances in automatic speech recognition would facilitate more user friendly speech-based recommender systems.

Users of the novel interface were further asked if they thought, that the speech-based interface made ReComment easier to use than a traditional mouse-based interface. A graph of the responses is shown in Figure 2.10.

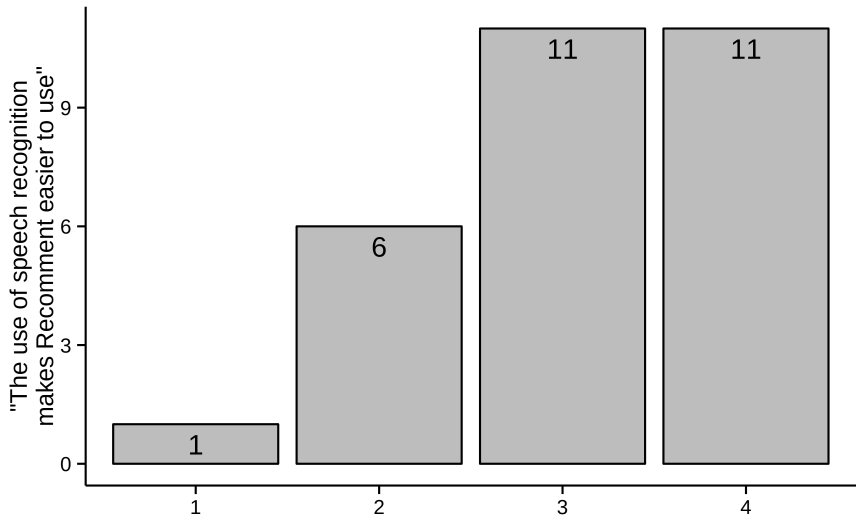

Figure 2.10.: Evaluation: “Did the speech recognition make ReComment easier to use?” ([1, 4], higher numbers signal agreement).

2.3.3. Recommendation Performance

As part of the questionnaire, study participants were asked to rate the accepted, final product on a scale of one to four (higher is better). Users of the novel speech-based user interface rated their final recommendation significantly higher than those using the traditional interface ($p < 0.05$). User's responses are summarized in Figure 2.11.

This improvement in recommendation quality becomes particularly notable when taking into account, that users of ReComment's speech-based interface used substantially fewer interaction cycles to arrive at a better fitting product ($p \ll 0.001$). Refer to Figure 2.12 for more details.

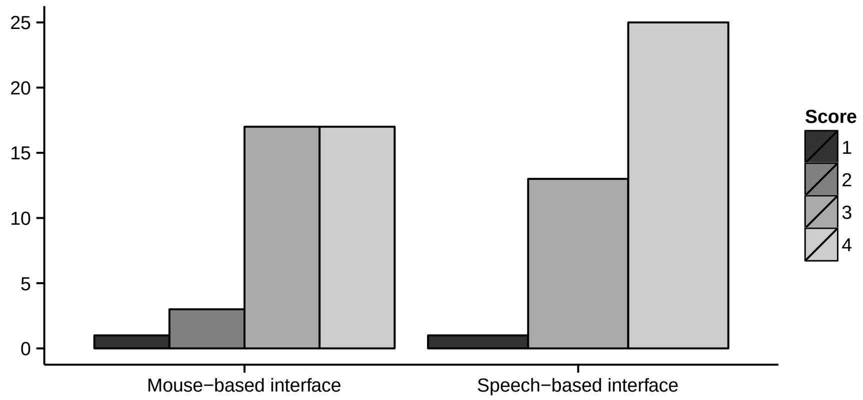

Figure 2.11.: User score of last recommended item ([1, 4], higher numbers are better).

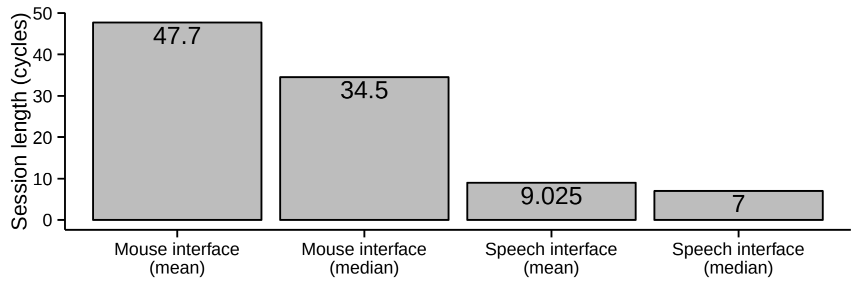

Figure 2.12.: Session length (lower is better).

2.4. Conclusion

This chapter outlined ReComment, a speech-based natural language approach to critiquing-based recommender systems. An empirical study was presented, which shows, that users of the speech-based interface required substantially fewer interaction cycles to find better fitting products than with a comparable traditional, mouse-base based interface.

3. Towards Human-like Recommender Systems

Building on the results of Chapter 2, where it was shown how a spoken natural language interface can allow users to more quickly and more accurately specify their true preferences, this chapter presents a conversational, knowledge-based recommender system, employing a human-like spoken natural language sales dialog. The major contribution of this chapter centers around exploiting additional information, extracted from spoken input, to increase recommendation performance without increasing users' cognitive load. To this end, the developed prototype for the domain of consumer laptops, SpeechRec, not just parses user input lexically, identifying commonly used qualifiers present in human speech (e.g., "slightly cheaper"), but also goes beyond spoken words by integrating paralinguistic features into the recommendation process. In an effort to best facilitate natural language communication, SpeechRec also sheds ReComment's restriction to critiquing-style user input and presents a first foray towards the application of human-like spoken dialog systems in the domain of recommender systems.

To evaluate the approach, an empirical study was conducted, comparing two versions of SpeechRec. A basic version, which uses the speech-based interface but ignores any of the aforementioned additional meta-information, reducing the system's recommendation strategy to one very similar to that of traditional knowledge-based recommenders, and the full version of SpeechRec, which further incorporates, for example, lexical and acoustical polarity information.

3.1. System Description

The implementation of the envisioned human-like sales dialog posed a multitude of challenges. In an effort to identify typical interaction patterns, a wizard-of-oz pilot study was conducted with an early prototype of the system. The following major means of feedback articulation were identified.

Explicitly specified attribute values, for example "A MacBook, please.", can be treated as constraints on the referenced product attributes and processed similarly to input in constraint-based recommender systems [13].

Relative attribute statements, such as "Show me a cheaper one.", can be interpreted as unit critiques of the current recommendation [5].

Use case statements, such as "I need a laptop for University.", can be resolved to individual constraints using the domain model.

Orthogonally to the various types of feedback articulation outline above, it became clear that given unrestricted interaction methods, users would naturally also discuss attributes not normally known to traditional recommender systems, such as "I am looking for a laptop that stays cool under load" or even "I want a good looking laptop". In order to adequately process such requests, additional product information would need to be extracted from consumer descriptions of the product.

The following sections discuss how these challenges were addressed during the development of SpeechRec.

3.1.1. Product Database

In an effort to create a more challenging, and therefore more revealing recommendation situation, the significantly more complex problem domain of consumer laptops was selected over the digital compact camera domain explored with ReComment.

For the prototype of SpeechRec, information was collected on a selection of 632 consumer notebooks currently on the market.

Factual Attributes

For every product, information on a core set of 40 attributes, ranging from the product's weight to the amount and type of cells used in its battery, were collected. A full list of factual attributes included can be found in Table B.1. Additionally, every item is associated with at least one product photo.

User Sentiment

In an effort to source product information beyond their manufacturer supplied specifications, sentiment information was extracted on a variety of aspects from a total of 3246 customer reviews collected from a popular online retailer[1].

Aspects To define the set of relevant product aspects to collect sentiment on, a list of around 100 aspects was compiled manually. This list was then extended based on what aspects reviewers actually commented on, by analyzing the collected reviews as follows. First, all review data was segmented into individual words using OpenNLP's German parser[2], and this collection of words was clustered to 64 cluster centers using Clark's POS induction algorithm [10]. Then, clusters of which more than 10 % of the contained words were already known aspects, were identified. From these, all other words with high membership functions were selected. The resulting list was manually pruned to arrive at a total of 304 aspect keywords, which were mapped to 41 distinct aspects.

Aspects were organized in a hierarchical structure, such that comments on, for example, the "Viewing Angles" aspect would also influence the "Display" aspect. The full hierarchy is shown in Figure B.1.

[1] `http://www.amazon.com`
[2] `https://opennlp.apache.org`

Polarity Lexicon To extract sentiment from plain text statements, a range of key words and phrases were annotated with their inherent polarity, similar to the process Andreevskaia et al. presented in [1].

A list of 40 positive (e.g., "great", "premium", or "sturdy") and 40 negative seed tokens (e.g., "catastrophic", "flimsy", or "heavy") were manually collected and assigned a fixed polarity of +1 and -1, respectively. Using the German OpenThesaurus web service[3], these initial groups of tokens were then recursively extended for two iterations with their elements' respective synsets. The polarity of newly added elements was scaled in relation to their distance from their respective seed. As suggested in [43], the distances from both the positive and the negative seed tokens were taken into account when computing a token's final polarity score.

Using the technique outlined above, a comprehensive polarity dictionary for the domain of consumer laptops was collected, containing 4628 polarity laden words and phrases.

Sentiment Extraction A lexical approach, roughly based on Shakih's et al. process outlined in [38], is used to extract sentiment. Reviews are parsed individually. For every sentence, a parse tree is generated using Parzu, the Zurich Dependency Parser for German [36, 37]. In the resulting tree, aspect and polarity laden phrases are identified using the resources described above, and their respective nodes are marked. Polarity nodes are then attached to their closest aspect node, where closeness is determined as the shortest path through the parse tree. The sentiment of an aspect expressed by a given review is determined as the sum of the polarity scores of all associated nodes. The final sentiment score for an aspect of a product is the median of all such aspect scores of the item's customer reviews. In an effort to reduce noise, only aspects that were commented on by at last two customers are included in the final evaluation of the product.

[3] https://www.openthesaurus.de

3.1.2. Spoken Language Input

As has been discussed in Section 2.1.2, conversational spoken language systems are inherently difficult. Note, that SpeechRec poses a significantly greater challenge to its speech-recognition component than even ReComment did, as SpeechRec is expected to handle a much broader range of user inputs in a decidedly more complex domain.

SpeechRec's speech processing subsystem is based on the same principal setup as outlined in Section 2.1.2. However, in addition to Simon and CMU SPHINX, SpeechRec also integrates the openEAR framework for paralinguistic analysis as discussed in Section 3.1.2. An overview of the resulting architecture can be found in Figure 3.1.

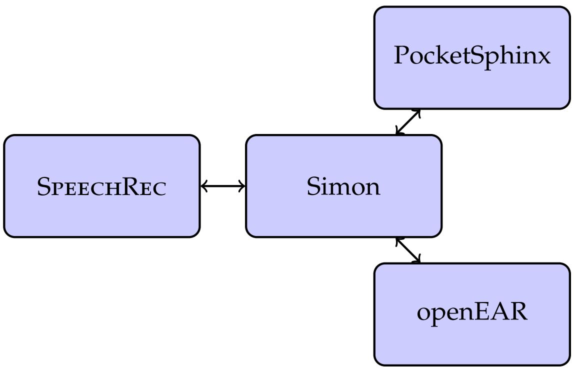

Figure 3.1.: Speech processing architecture.

Speech Model

Building on the successful setup developed for ReComment described in Section 2.1.2, a task dependent speech model was created for SpeechRec.

SpeechRec's even more challenging decoding situation posed by the significantly more complex domain, coupled with the more conversational setting, necessitated some revisions as described below.

Language Model Similar to the setup described in Section 2.1.2, SpeechRec's language model combines a large, generic language model with a smaller, task dependent one. However, in contrast to Grasch's et al. earlier work, a conscious effort was made to also align the larger "background" model with the task at hand. To that effect, text corpora were collected from laptop descriptions from e-commerce websites, user reviews, transcripts of user interactions of the pilot study and a crawled collection of texts from Austrian websites. Additionally, an artificial text collection was built of all sentence fragments SpeechRec's natural language understanding component recognizes, as described in Section 3.1.2. Individual language models were built for each corpus, which were then combined to create our final 3-gram language model. The mixing coefficients were chosen based on perplexity optimization on a held out portion of user transcripts from the pilot study.

Of this task related corpus, the portion that refers to explicit numerical values was again split off, so as not to inadvertently bias the language model by over representing these phrases as explained in Section 2.1.2. The two resulting task dependent corpora are referred to "NLU" and "NLU Numbered", respectively. Individual 3-gram language models were built for all corpora listed above, with all non task-related language models being limited to their 3500 most common words.

Based on perplexity tests on a held out portion of the transcripts from the pilot study, the mixing coefficients listed in Table 3.1 were selected for the recombination of the individual models, creating the final 3-gram language model.

λ	**Language models**
0.35	NLU
0.29	Pilot Transcripts
0.15	NLU Numbered
0.10	Austrian Websites
0.10	Product Reviews
0.01	Product descriptions

Table 3.1.: Selected language model mixtures.

SpeechRec's dictionary is based on Schuppler's et al. phonetic dictionary for conversational Austrian German [35]. As only a limited corpus of audio data was available to train the acoustic model as described in Section 3.1.2, the dictionary's phoneme set was slightly reduced by mapping all instances of the voiced "s" ('z'), a rare sound in Austrian German, to its unvoiced equivalent. The phonetic dictionary was further extended, in part through manual transcriptions and in part by using the Phonetisaurus grapheme to phoneme system[4], to include task-relevant vocabulary, and the words of the aforementioned background corpus.

Acoustic Model In an effort to maximize SpeechRec's recognition performance, the acoustic model for Austrian German described in Section 2.1.2, was revised.

The original audio corpus was extended to include audio collected in ReComment's evaluation as well as data sourced from the SpeechRec pilot tests. All recordings of spontaneous speech were manually orthologically transcribed, and audible breaths, flicks, clicks, coughs, laughs, various filler words, and other background noise were tagged with explicit disfluency markers.

Based on this data, and the updated phonetic dictionary, a continuous hidden Markov model was estimated using SphinxTrain of the CMU SPHINX speech recognition framework. This acoustic model was then used to force-align the training corpora, in order to introduce noise markers and to combat minor alignment problems during early stages of the training procedure. Based on these aligned transcripts, a final, continuous acoustic model was created, using 2000 tied states, 32 Gaussian mixtures, and a 29-dimensional LDA feature transformation.

Paralinguistic Analysis

In traditional human to human sales dialogs, a trained sales clerk will commonly not only listen to the words spoken by the customer. Intonation, hesitations, even body language all convey considerable information that

[4] https://code.google.com/p/phonetisaurus

ultimately allows for better recommendations. While recognizing such social cues is obviously a sizable undertaking, it seems reasonable to assume, that integrating even limited insight into the recommendation strategy can already prove beneficial.

In his seminal publication about emotion detection and classification, [34], James Russel mapped out human emotion on the following two dimensions. Valence, a measure of an emotion's positiveness, and arousal, the strength of the observed emotion. It stands to reason, that the emotional investment of an articulated preference, expressed by such an arousal score, would generally correlate with the user's investment in its meaning, and could therefore greatly aid a recommender system in its conflict resolution.

SpeechRec uses the openEAR framework, employing SVM based regression trained on the SAL corpus to calculate an arousal score for every recorded user statement. Refer to Section 3.1.4 on how this paralinguistic information is used in SpeechRec's recommendation strategy.

Natural Language Understanding

SpeechRec's natural language processing is centered around the concept of identifying statements in the speech recognition component's hypothesis. To enable processing of the types of preference articulation observed in the pilot study outlined in Section 3.1, the following statement archetypes were implemented in SpeechRec.

Aspect statements are recognized references to a sentiment carrying product aspect as described in Section 3.1.1.

Constraint statements encapsulate an uttered absolute or relative constraint.

Use case statements encode the user's expressed main use case of the product.

Command statements are directed at SpeechRec and express either "Yes", "No", or "Back".

In order to ultimately form such high-level statements, SpeechRec's parser first tokenizes the user input. More than 800 key phrases are known to the parser, a full list of which can be found in B.1. Superfluous input is ignored. The following types of tokens are distinguished.

Attributes are tokens referring to a product attribute of the domain. Matching key phrases include, for example, "display" or "warranty".

Modifiers define one relationship for constraints. While such tokens usually require an associated attribute to form a statement, some modifiers may imply a default attribute. For example, the modifier "larger" can be combined with an attribute, but when used without further qualification, it is assumed to refer to the laptops overall size.

Meta modifiers encapsulate lexical polarity that affects the strength of modifiers or some commands. For example, "slightly", a meta modifier, would act on the modifier "larger" to express "slightly larger". Polarity of meta modifiers ranges from -1 (e.g., "not") to 2 (e.g., "very").

Commands are self-contained statement tokens, such as "Yes", "Back", but also "cheaper" or "for university" (a commonly specified use case).

Tokens may also carry an inherent polarity expressed by the used wording. For example, the user input "less so" generally expresses the same meaning as "no", but carries additional polarity information.

A simple parse often produces many possible, potentially overlapping tokens which can in turn be combined in a multitude of ways to produce even more statements. For example, a user input of "larger" would be parsed to a statement reflecting the critique "increase screen size". The input "hard drive" alone would produce a constraint statement expressing "good hard drive"[5]. The combined input "larger hard drive" can therefore have at least two different parses by either interpreting the tokens independently or combining them to form a single constraint. Other, more subtle cases include for example the German word "Speicher", which can be used to roughly mean "storage", and is used to refer both to main memory and persistent storage. Only in the context of "more than 4 gigabytes", it becomes clear that in this particular instance the user was most likely referring to the laptops main memory. To cope with these issues, a parser was created, which extracts statements from all potential parse trees (token combinations) and

[5] Lone attributes in a recognition hypothesis default to the "good" modifier, given no other information.

selects the end result that maximizes the amount of tokens per statement (preferring one statement "larger hard drive" over two based on "larger" and "hard drive"), while minimizing unassigned (dropped) tokens. Despite this approach's significant computational complexity, performance is not an issue in practice, as the parser is only run against the relatively short sentences, or even sentence fragments, returned by the speech recognition subsystem.

All extracted statements are annotated with an "influence" score, computed from the following three components. Statement quality, expressing the parser's confidence in capturing a user's intended meaning ($[0,1]$), a statement importance score deduced from the arousal estimated by the paralinguistic analysis ($[0,2]$), and lexical polarity from extracted qualifiers ($[-1,2]$). These values are multiplied to arrive at the final statement influence score as shown in Equation 3.1.

$$statementInfluence = quality * arousal * polarity \tag{3.1}$$

Dialog Strategy

SpeechRec's dialog strategy is designed to allow users to specify their preferences as naturally as possible. In case the user does not yet have fully formed preferences, the system may take initiative and guide the user through their purchasing decision. On overview of SpeechRec's mixed-initiative dialog strategy is shown in Figure 3.2.

At the start of a new interaction session, SpeechRec will introduce itself and ask the user to either describe their ideal product or to let SpeechRec know, if they needed more guidance. In the latter case, the system would take initiative and ask one of the following domain questions. "What are you gonna be using your laptop for?", "Which attribute is most important to you?", "Do you need a very fast laptop?", "Do you need a laptop that is very portable?", and "Is a cheap price very important to you?". SpeechRec can be seen taking initiative in Figure 3.3.

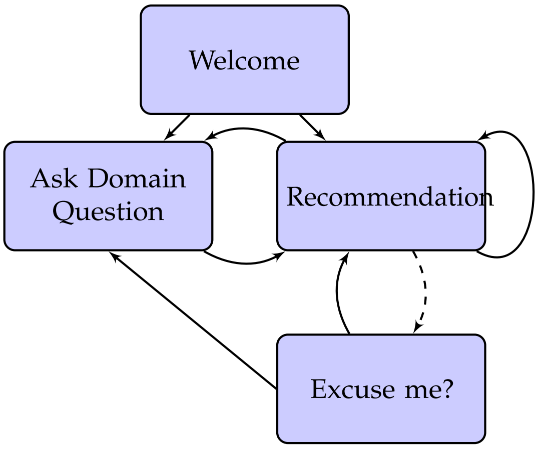

Figure 3.2.: SpeechRec's dialog strategy: An overview.

As suggested by Shimazu in [39], SpeechRec automatically selects the domain question of which it expects the user's reply to maximally constrain the remaining search space.

When a dialog turn produces no actionable statements, SpeechRec may ask the user to rephrase their input. However, as recommended in [41], the system will never ask this question more than once in succession and not more than twice during a single user's interaction session. Instead of asking for further clarification, the system will instead take initiative and ask one of the aforementioned domain questions. This naturally moves the conversation back to an interaction paradigm the system understands.

SpeechRec uses a simple, semantic approach for end-of-turn detection. After a user stops talking, the spoken utterance is processed and parsed. If it contains actionable statements, SpeechRec will wait for 1.5 seconds before considering the dialog turn complete. If the user starts talking again within that time, the timer resets and the new utterance is considered part of the same turn. If a dialog turn has produced no actionable statements and the user stops speaking, SpeechRec waits for up to 6 seconds before completing the turn. Every unactionable user input within a turn reduces this time limit by two seconds down to a minimum of 1.5 seconds. This seemingly long initial timeout was introduced as a response to some pilot user's excessive use of filler words ("uhm.... <long pause> <statement>").

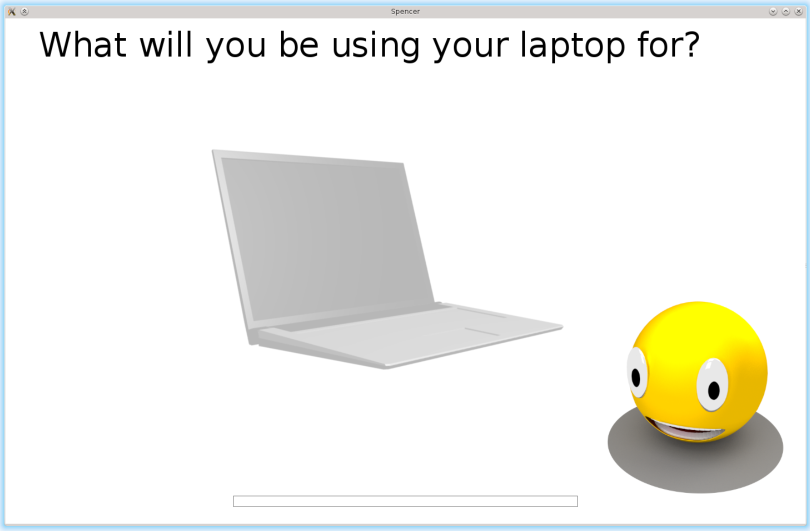

Figure 3.3.: SpeechRec taking initiative after the user expressed no particular preferences. (Text enlarged for readability and translated from German.)

3.1.3. User Interface

SpeechRec's user interface was designed to minimize distractions from the task at hand. Except for a visual indicator of the current input volume that was included to make SpeechRec look responsive while a user was talking, it includes no traditional controls.

Avatar

Holding a conversation with a faceless computer program can feel cumbersome for some users [30]. In the earlier study evaluating ReComment, it was found that many users, when presented with a simple, traditional looking interface, would restrict themselves to fairly primitive command-like interaction, and almost none would communicate with the system like they would with a human salesclerk.

In an effort to instill confidence in the system and to encourage users to use a natural, conversational tone when interacting with SpeechRec, an animated, speaking avatar was designed and implemented using the 3D animation suite Blender[6] and the MaryTTS[7] framework for text-to-speech synthesis. The avatar can be seen in, for example, Figure 3.3.

Product Display

Even though SpeechRec employs an extensive domain database as described in Section 3.1.1, only a very limited selection of ten core attributes is usually shown[8], as can be seen in Figure 3.4. Additionally, SpeechRec will show all other attributes that the user expresses an interest in either explicitly by mentioning them, or implicitly by mentioning a use case where they become relevant. During the course of the study presented in Section 3.2, it was found that users did not hesitate to articulate preferences about product attributes that were not (yet) shown on screen, highlighting an important advantage of speech-based interfaces. They allow expert users to efficiently work with a comprehensive domain model and a wide range of complex constraints, while keeping the user interface trivially simple.

Product attributes are set in the font weight directly representing their relevance to the current user, determined using the system's user preference model. Attributes referenced by constraints in the user model are furthermore shown with their values colored in a shade between red and green depending on how well they fulfill the user's requirement.

Underneath the factual product attributes, SpeechRec shows extracted user sentiment on aspects of the current recommendation. Because this information is usually sparse, all known sentiment information is shown. Aspects with positive sentiment are shown on the right, those with negative sentiment on the left. Red and green bars are used to express the polarity of the extracted sentiment.

[6] `http://www.blender.org`

[7] `http://mary.dfki.de/`

[8] The set of attributes that SpeechRec shows per default for every product are manufacturer and product name, price, user rating, screen size and resolution, processor name, core count and frequency, graphics card name, main memory capacity, storage description, operating system and average runtime on battery.

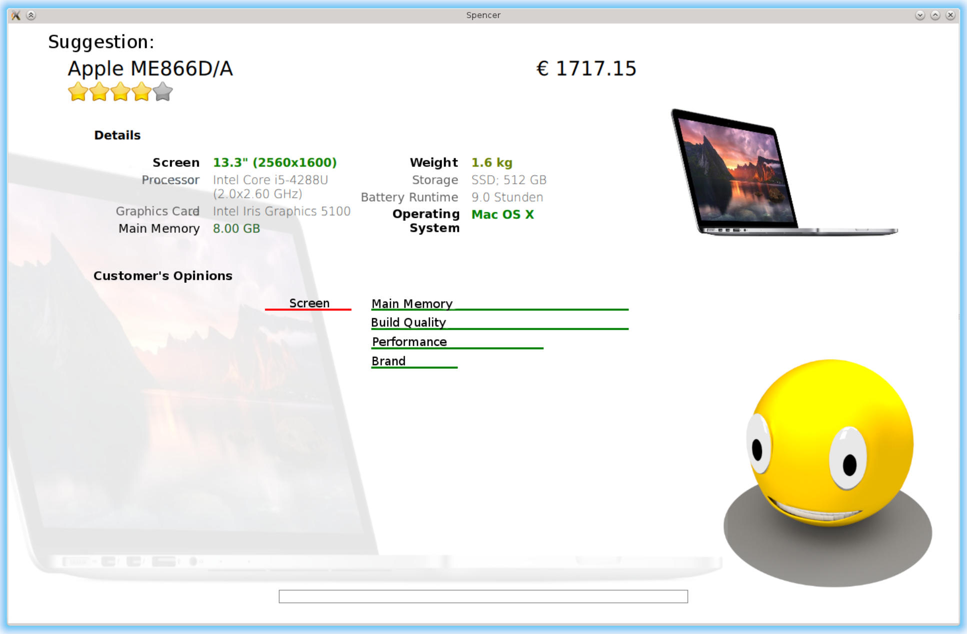

Figure 3.4.: SpeechRec recommending a product. (Text translated from German.)

Figure 3.4 shows SpeechRec's interface after the initial turn in an interaction session, the transcript of which can be found in Table 3.2.

3.1.4. Recommender

SpeechRec is based on the knowledge-based approach to recommender systems.

User Preference Model

SpeechRec maintains a rich user preference model, consisting of constraints on attributes and a list of aspects the user has expressed interest in. Both

Actor	Statement
SpeechRec	Hello! I'm here to help you find your new laptop. Ready? Then let's start! Please describe your ideal laptop. If you don't yet know what you are looking for, feel free to let me know.
User	okay ... I'd like to get a Mac, if that's possible. Actually ... I need a lot of ... main memory. That's pretty much the only important thing; Uhm... Oh, it shouldn't be too big, you know. About 15 inches tops. Also not too heavy. Because I'll need to carry it around. Uhm, yeah.
SpeechRec	What do you think of this laptop?

Table 3.2.: Sample user interaction. (Translated from German.)

these types of encoded requirements are referred to as "recommender items"[9].

The impact of a recommender item on the recommendation strategy depends on both the age of the recommender item, and its influence, derived from the influence of the statement that spawned it as described in Equation 3.1. The influence calculation is further described in Equation 3.2.

$$influence = \left(1 - \frac{age}{timeToLive}\right)^2 * statementInfluence \qquad (3.2)$$

In SpeechRec, the utility of a given item is the sum of the individual recommender item's utilities from the user's preference model, scaled by their current influence, discounted by a measure of dissimilarity from the current product. SpeechRec distinguishes between constraint recommender items and aspect recommender items.

[9]Please note, that the term "recommender item" refers *not* to an element of the product space, but to an encoded user preference.

Constraint Recommender Items Constraint recommender items consist of an attribute identifier, a concrete value and their relationship type describing the nature of constraint, such as "larger" or "equal to". Critiquing input grounds its created constraints in the current recommendation's attribute value. General statements, for example "a large screen" or "a good price" are treated as referring to the attribute's median value across the product space.

As described in Section 3.1.2, SpeechRec's introduction encourages users to describe their ideal product in broad strokes. While the reply to this prompt is not parsed differently than other feedback cycles, in practice this fulfills a roughly equivalent function as traditional knowledge-based system's initial search. This allowed to move away from ReComment's goal driven utility function outlined in Section 2.1.4, and towards a more conventional one described below.

To calculate the support of a constraint recommender item for a given product, SpeechRec measures a relative distance between the product's attribute value and the value recorded with the constraint, linearly scaled to a range of $[-1, 1]$ based on the attribute's value domain. Depending on the constraint's relationship type, this distance is then processed as outlined below.

$>$ The distance is used as is.
$<$ The distance is multiplied by -1.
$=$ The absolute value of the distance is substracted from 0.5, then doubled to maintain the scale from -1 to 1.
$\neq$ The distance reported from the equality operator is inverted.
"better than" This relationship type is resolved to one of the aforementioned types based on a manually defined optimality criteria for each domain attribute. For example, a *"better than"* constraint on *price* would resolve to a $<$ constraint. For attributes where relative comparisons are not defined (nonnumerial attributes), these relationships are grounded in equality constraints. For example, a *"better than"* constraint on *storageType* would resolve to a $storageType = SSD$ constraint. The distance reported is that of the deduced constraint.
"worse than" The distance reported by the *"better than"* relationship type is inverted.

To arrive at the final constraint utility the derived distance measures are scaled using the logistic function described in Equation 3.3.

$$cUtility = \left(\frac{2}{1 + exp(-6 * distance)} - 1 \right) * polarity \tag{3.3}$$

Additional information about the used utility function and a more detailed discussion about the reasoning supporting it can be found in [17].

Aspect Recommender Items Aspect recommender items reference a product aspect and encode a user's interest in it. An aspect recommenders item's utility, given s specific product, is equivalent to the product's customer sentiment on the aspect in question. As such, they represent the desire to maximize review sentiment on given aspects and could be compared with constraints expressing "the referenced aspect is good".

Aspect items enable SpeechRec to handle user requests that reference product attributes where information is either scarce, unavailable, or inaccurate, such as display quality, looks, or battery life. The base influence of aspect statements is scaled to a tenth of that of constraint recommender items, so as not to overrule factual constraints expressed by the user. Aspect recommender items and constraint recommender items function principally independently, but naturally also exhibit synergetic effects. For example, consider the user input "Good battery life is important." SpeechRec's domain model includes the manufacturer provided average runtime. However, as no standardized testing regiment is imposed to measure this attribute, practically observed battery runtime is known to often differ significantly from a manufacturer's estimate. Given the aforementioned input, SpeechRec would record a constraint, requesting products whose quoted runtime on battery is at least above average, preferably higher. In addition to the factual constraint, SpeechRec would also recognize the user's interest in the product's battery and store an aspect recommender item referring to the "Battery" aspect. As a result, SpeechRec would look for a product with a good quoted battery life, that customers have expressed positive sentiment on. Products with long quoted battery life, which had this claim discredited by customers in their reviews, would therefore see their scores discounted.

Prior Recommendation Probability

Every item in SpeechRec's database is assigned a prior recommendation probability based on its Amazon Sales Rank and its customer rating. This calculation is shown in Equation 3.4.

$$priorRecommendationProbability = rating + 1 - \frac{salesRank}{MaxSalesRank} \tag{3.4}$$

Recommendation Strategy

Given the utility calculation of recommender items detailed above, the recommendation strategy is defined as follows. As described in Section 2.1.4, the recommendation strategy employed by SpeechRec ensures, that it respects at least one of the constraints added in the preceding interaction cycle. This avoids the issue of earlier specified preferences dominating the interaction, giving the impression of SpeechRec being unresponsive. From the set of products which fulfill at least one of the most recently added constraints, the one with the overall highest utility is selected, where utility is calculated based on the product's prior recommendation probability as described in Section 3.1.4, and the sum of the user's recommender item's utility scores as described in Section 3.1.4, and finally discounted based on a measure of distance from the system's previous recommendation. The recommendation strategy is shown in more detail in Algorithm 3.

3.2. Study Design

An empirical study was conducted to assess the effect of incorporating more nuanced user preferences, afforded by the spoken natural language based interface's rich preference articulation, into SpeechRec's recommendation strategy.

Input: Known products P, user preference model (recommender items) R, current recommendation r_{old}.
Output: Next recommendation r_{new}.

$newC \leftarrow \{c \in R | c \text{ is constraint item} \wedge c.age = 0\}$;
$P' \leftarrow \{p \in P | \exists c \in newC : c.utility(p) > 0\}$;
if P' *is empty* **then**
  $P' \leftarrow \{p \in P | \exists c \in newC : c.utility(p) \geq 0\}$;
end
if P' *is empty* **then**
  $P' \leftarrow P$;
end
$bestUtility \leftarrow -\infty$;
$bestOffer \leftarrow r_{old}$;
for $p \in P'$ **do**
  $utility \leftarrow o.priorRecommendationProbability()$;
  for $ri \in R$ **do**
    $utility \leftarrow utility + ri.influence() * ri.utility(p)$;
  end
  $utility \leftarrow utility - o.distance(p)$;
  if $utility > bestUtility$ **then**
    $bestUtility \leftarrow utility$;
    $bestOffer \leftarrow p$;
  end
end
return $bestOffer$;

Algorithm 3: Schematic recommendation strategy. Scaling factors omitted for brevity.

3.2.1. Compared Systems

Two versions of the developed system were evaluated. An additional comparison of SpeechRec with the traditional, mouse-based recommender system WeeVis presented in [14] is shown in [17].

Reduced SpeechRec

A reduced configuration of SpeechRec, referred to as "SpeechRec reduced", was evaluated, which retains the novel user interface but does not act on the richer feedback which that affords. All recommender item's utility functions are thresholded to -1, 0, or 1 (whichever value is closest). Detected lexical and paralinguistic nuances are ignored. Hence, a recommender item's influence on the recommendation strategy solely depends on that item's age. These changes effectively reduce SpeechRec's recommendation strategy to that of most traditional knowledge-based recommender systems.

SpeechRec

The second configuration under test, simply called "SpeechRec", incorporates the full, rich user model discussed in 3.1.4. Compared to SpeechRec reduced, this introduces the following additional information. Statement quality determined by the parser's confidence score, lexical polarity deduced from any descriptive adjectives used, a paralinguistic measure of importance deduced from the observed arousal, and a nuanced utility score for each recommender item.

3.2.2. Test Demography

44 study participants, mostly students and post-graduate researchers, were split into two groups, each testing one system. A demographic overview of both groups can be found in Table 3.3.

3.2.3. Task Definition

Study participants were asked to imagine that their laptop, should they own one, had been stolen and that they were looking for a replacement. They were informed that they would be using a "virtual shopping assistant", whom they could converse with like a human sales clerk. No other

Characteristics	SpeechRec Reduced	SpeechRec Full
Male	15	18
Female	7	4
Total	22	22
Mean age	25	28
Personally own a laptop	91 %	91 %
Sought help when buying their last laptop	36 %	32 %

Table 3.3.: Demography of the participants of the empirical study.

instructions were given regarding SpeechRec's user interface. Participants were instructed to notify the person overseeing the experiment when they found a product that they were ready to accept, or when it became apparent that no such product could be found[10].

3.2.4. Evaluation

After completion of the recommendation session, participants were asked to fill out a questionnaire, assessing the participant's opinion on the usability of the system and the last recommended product. Additionally, basic demographic information was collected. Responses were verified using control questions. Results reported are of the form $positiveQuestionResponse - negativeQuestionResponse$. The complete questionnaire is shown in Section B.4.

All interactions with the systems were logged, and recorded user input was stored for analysis.

[10]When a user commended a recommendation, SpeechRec would remind the user to notify the supervisor, if they should want to accept the product.

3.3. Results

This section presents the results[11] of the empirical study discussed in Section 3.2.

3.3.1. Input Processing

During the course of the study a total of 579 dialog turns consisting of 811 utterances were observed.

All system interaction was manually reviewed after completion of the study. An error is reported for every turn, where a human operator would describe the system's interpretation of the input as incorrect. This includes, for example, faults caused by the speech-recognition component, faults introduced by the parser, or faults that occur as a result of limitations of the dialog strategy[12].

A total of 294 turns were found to be correct (50.78 %), with 55 more partially correct turns (9.5 %), leaving 230 erroneous turns (39.72 %). While these numbers may look alarming initially, it is important to note, that most errors resulted merely in SpeechRec taking initiative for one turn. As a result, users rarely even noticed that SpeechRec did not correctly handle their previous input. This is well reflected in the participant's assessment of the system's recognition accuracy, where SpeechRec's natural language processing received a median score of 2 on a scale from -3 to +3 (higher is better).

Given manually corrected transcriptions, the number of correct turns increased substantially to 442 (76.34 %), with 56 more (9.67 %) partially correct ones. Approximately half of SpeechRec total interpretation errors are therefore a direct result of speech recognition errors. While it can be argued, that the speech recognition component proved sufficiently capable for the purpose of this experiment, it stands to reason, that improvements to the

[11] Reported statistical significance of results of comparative analysis was calculated using Welch's t-tests.

[12] For example, one participant replied to the domain question, asking if they were looking for a particularly "fast" laptop with "How fast is *fast*?".

speech recognition subsystem would further enhance the recommendation quality of SpeechRec in practice.

3.3.2. Usability

Study participants were asked to rate the usability of the system under test using a modified system usability survey. Refer to Section 2.2.4 for details about the used questionnaire.

Perhaps unsurprisingly, the two versions of SpeechRec showed no statistically significant differences in SUS scores. In fact, both groups share the same median response of approximately 85.7 points, very close to ReComment's speech-based interface median score of 87.5.

3.3.3. Recommendation Performance

Both the recommendation quality, and the recommendation efficiency are assessed, by asking participants to rate the recommended product, and timing the interaction sessions, respectively.

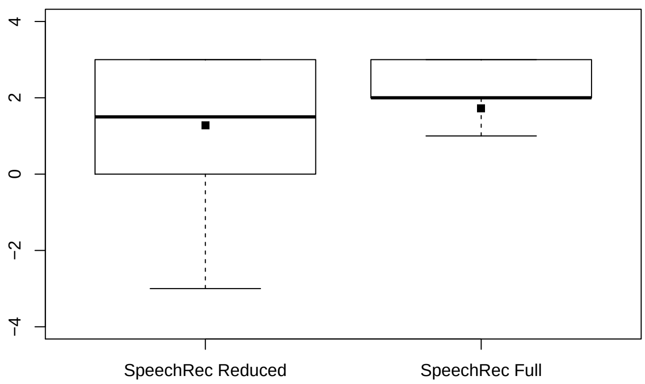

Figure 3.5.: Participant's subjective score of the last shown product of the interaction session. ($[-3, 3]$, higher is better. Black squares indicate the arithmetic mean.)

Figure 3.5 shows study participant's scores of the system's final recommendation. While the full version appears to outperform the reduced baseline, this difference does not reach statistical significance ($p \approx 0.07$).

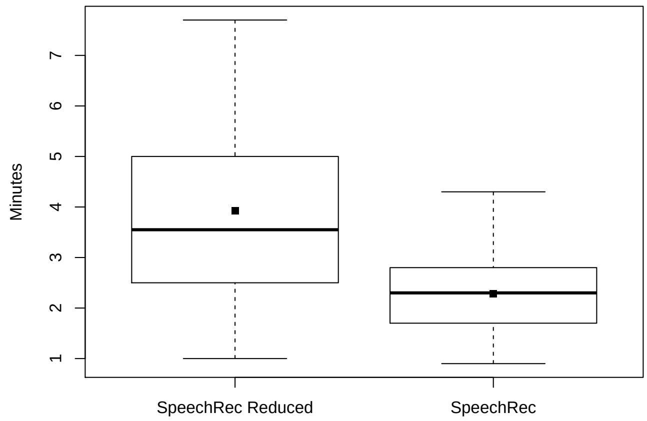

Figure 3.6.: Recommendation session duration. (Lower is better. Black squares indicate the arithmetic mean.)

However, our evaluation shows, that interaction sessions involving the full version of SpeechRec were substantially more efficient than those of participants using the reduced version ($p < 0.001$). Figure 3.6 shows an overview of recorded session lengths in minutes, Figure 3.7 further breaks this down to the amount of completed dialog turns and intermediate recommendations.

A more thorough evaluation, including a comparison of SpeechRec with a traditional knowledge-based recommender system using a comparable product database can be found in [17].

3.4. Conclusion

This chapter outlined SpeechRec, a speech-based natural language user interface, employing an exceedingly rich user preference model, facilitated by harnessing meta-information of spoken natural language input. It was

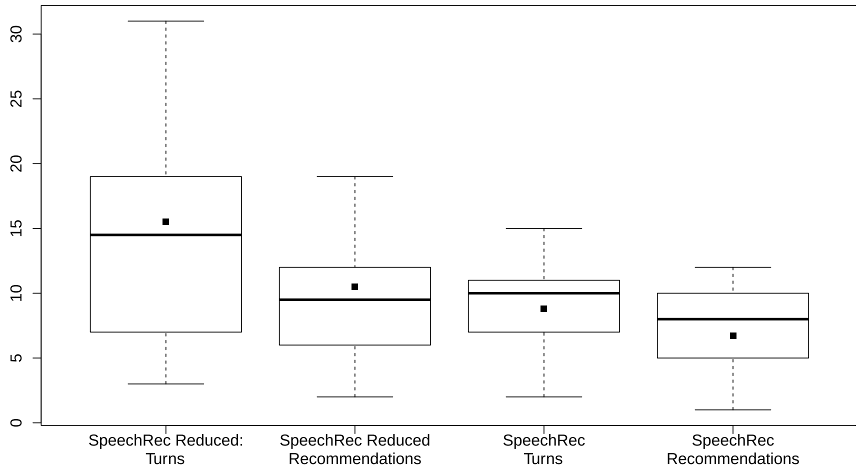

Figure 3.7.: Recommendation session length in completed iterations. (Lower is better. Black squares indicate the arithmetic mean.)

shown, how incorporating such more nuanced information into the recommendation strategy drastically increased SpeechRec recommendation efficiency, without impacting usability and potentially even slightly increasing recommendation quality.

4. Conclusion

This master's thesis examines the application of spoken natural language input for knowledge-based recommender systems.

To this end, the following two prototype systems were discussed: ReComment, a conversational, speech-based unit critiquing system, and SpeechRec, a knowledge-based recommender system, using mixed-initiative, human-like spoken natural language input, incorporating rich meta-information in its recommendation strategy. Both developed prototypes were evaluated using empirical studies. ReComment was shown to significantly surpass the recommendation performance of a comparable system employing a traditional user interface, reducing session length and increasing reported recommendation quality, establishing that spoken language input driven recommender systems can outperform traditional implementations. Based on the hypothesis, that this advantage was rooted primarily in allowing users to specify their requirements more naturally, and hence more efficiently, SpeechRec expands on this aspect by further analyzing the user's phrasing and intonation to afford more nuanced, and thus more precise user feedback. SpeechRec was compared to a reduced version of itself, which retains the novel user interface but limits its recommendation strategy to that of a traditional knowledge-based recommender system. It was shown, that identifying and utilizing meta-information naturally present in spoken natural language facilitates a more precise user preference model without demanding additional effort from users, which in turn can greatly enhance recommendation performance.

Usability In the course of the conducted empirical studies the usability of each system was evaluated using a modified system usability survey questionnaire.

Curiously, even though speech-based interaction presents an entirely different paradigm to traditional mouse-based interaction, the usability evaluation showed no statistically significant differences between any of the systems under tests compared to their respective baselines.

However, when limiting the analysis to participants who reported good speech recognition performance, the speech-based interface of ReComment handily outperformed its traditional counterpart. This suggests, that the developed prototypes may have been held back by their speech recognition component's accuracy, and that speech-based recommender systems may become increasingly viable as automatic speech recognition technology matures.

Recommendation Performance The evaluation of ReComment shows that using a speech-based user interface can facilitate more precise preference articulation, and therefore significantly improve recommendation performance by recommending better products ($p < 0.05$) in fewer interaction cycles ($p \ll 0.001$), when compared to a traditional, mouse-based system. As shown in the evaluation of SpeechRec, extracting additional meta-information from the user's articulated preferences can further increase recommendation efficiency over a simple, speech-based baseline ($p < 0.001$).

While further study is necessary to pinpoint the individual advances' contributions to the overall vastly improved performance, the following compounding factors allow speech-based recommender systems to maintain a more accurate user preference model, and thus summarize the main contributions of this thesis. Whereas traditional interfaces need to consider which options should be made available so as not to crowd the user interface or overwhelm novice users, speech-based interfaces allow direct articulation of even complex preferences. This leads to interfaces that require less abstraction on the part of the user. For example, a complex preference such as "I want a laptop for university" can easily be specified and, given a sufficient domain model, potentially be handled better, than if the user were forced to reformulate such a constraint to fit a restricted interface. In addition to more precise feedback, speech-based interaction also facilitates the extraction of a wide range of meta-information inherent in spoken language, expressed through the user's phrasing or intonation. Harnessing such information allows spoken language recommender systems to better capture the users true preferences, and thus to recommend better fitting products.

5. Future Work

This thesis demonstrates that speech-based recommender systems are worth further exploration in future research.

While human-like spoken dialog systems are still a topic of active research, prototypes like SpeechRec show that even comparatively simple implementations can enable practical systems to outperform traditional interfaces by facilitating more efficient interactions patterns. Further advances in automatic speech recognition and natural language understanding will continue to make speech-based recommender systems increasingly viable.

In future work, the application of paralinguistic analysis for spoken language recommender systems should be further studied. While SpeechRec already showed the integration of a measure of emotional arousal to be promising, a wide array of other paralinguistic phenomena are yet to be explored. For example, detected uncertainty could be used to make a recommender system more pro-actively react to a hesitant user. Furthermore, measured user frustration could potentially help a recommender system detect and react to its mistakes, as well as fuel efficient conflict resolution.

Interactions of study participants with SpeechRec showed that users would comfortably accept a human-like conversational style when talking to a speech-based recommender system. As such, the substantial body of research on effective sales dialogs can and should be exploited to improve future systems' dialog strategies. Since codifying such approaches in a virtual assistant facilitates testability, speech-based recommender systems could further present a fertile proofing ground for various psychological theories surrounding purchasing behavior.

Conversations with unrestricted, speech-based recommender systems such as SpeechRec additionally provide valuable insight into users' true, hidden preferences and their preferred articulation. Thorough review of interaction sessions could therefore help accelerate knowledge engineering and improve even traditional user interfaces.

Finally, it is worth mentioning that many of the core findings of this thesis are not generally limited to the domain of recommender systems. Speech-enabled applications are rapidly gaining popularity, yet many if not most still treat speech input as essentially text input. It is the authors believe, that the coming years will see a paradigm shift in processing spoken natural language, driven by an effort to utilize the vast amount of additional information human speech provides, which has so far been widely ignored.

Appendix

Appendix A.

Speech-based Unit Critiquing

This chapter provides additional supporting material surrounding ReComment, and its evaluation.

A.1. Language Definition

The following list shows all key phrases detected by ReComment expressed as regular expressions. Alternate (dialectal) spellings have been omitted.

AGFAPHOTO
ALTERNATIVE
ANDERE
ANDEREN
ANDERER
ANDERES
ÄNDERN
AUFLÖSUNG
AUSFÜHRUNG
AUSSER
BAUWEISE
BEIBEHALTEN
BENQ
BESSERE
BESSEREN PREIS
BESSERER PREIS
BESSERES
BILLIGE
BILLIGER
BILLIGERE
BILLIGERES
BLEIBEN
BREITER
BREITERE
CANON
CASIO
CCD
CMOS
DESIGN
DEUTLICH
DICKER
DICKERE
DIGITALEM ZOOM
DIGITALEN ZOOM
DIGITALER ZOOM
DIGITALE ZOOM
DOCH NICHT
DÜNNE
DÜNNER
DÜNNERE
ECHTEN ZOOM
EIN BISSCHEN
EIN BISSERL
EIN WENIG
ERHÖHEN
ETWAS
FARBE
FLACHER
FUJIFILM
GEFÄLLT MIR.* NICHT
GERINGEREN
GERINGERER
GERINGERERE
GERINGERES

GEWICHT
GLEICHE
GLEICHEM
GLEICHEN
GLEICHER
GLEICHES
GRAMM
GRÖSSE
GRÖSSER
GRÖSSERE
GRÖSSEREM
GRÖSSEREN
GRÖSSERER
GÜNSTIGER
GÜNSTIGERE
GÜNSTIGERES
GÜNSTIGES
HERSTELLER
HOCHWERTIGER
HOCHWERTIGERE
HOCHWERTIGEREN
HOCHWERTIGERES
HÖHER
HÖHERE
HÖHEREN
HÖHERER
HÖHERERE
HÖHERES
INTEGRIERTEM SPEICHER
INTEGRIERTEN SPEICHER
INTERNEM SPEICHER
INTERNEN SPEICHER
INTERNER SPEICHER
INTERNE SPEICHER
KAMERA
KEINE
KLEINE
KLEINER
KLEINERE
KLEINEREM
KLEINEREN
KLEINERER
KLOBIG
KODAK
KOMPAKT
KOMPAKTE
KOMPAKTER
KOMPAKTERE
KOMPLETT
KOSTEN
LEICA
LEICHT
LEICHTER
LEICHTERE
LETZTE KAMERA
LÖSCHEN
MARGINAL
MARKE
MEGABYTE
MEHR
MINDESTENS
MIT
MODELL
NICHT
NIEDRIGER
NIEDRIGERE
NIEDRIGEREN
NIEDRIGERER
NIEDRIGERES
NIKON
OLYMPUS
OPTISCHEN ZOOM
OPTISCHER ZOOM
ORDENTLICH
PANASONIC
PENTACON
PENTAX
PREIS
PREISWERTER
PREISWERTERE
PREISWERTERES
PRODUKT
RICOH
ROLLEI
SAMSUNG
SCHMALE
SCHMALER
SCHMALERE
SCHWER
SCHWERER
SCHWERERE
SELBE
SELBEN
SELBER
SELBES
SENSOR
SENSOR GRÖSSE
SENSOR TYP
SIGMA
SIGNIFIKANT
SONSTIGE
SONY
STABIL
STABILER
STABILERE
TEUER
TEURER
TEURERE
TEURERES
ÜBER
UM
VERGESSEN
VERGISS DAS
VERRINGERN
VIEL
VORHERIGE KAMERA
VORHERIGEN KAMERA
VORHERIGES MODELL
VORIGE KAMERA
VORIGEN KAMERA
VORIGES MODELL
WECHSELN
WEITER
WENIGER
WENIGSTENS
WERTIG
WERTIGE
WERTIGEN
WERTIGERE
WERTIGEREN
WESENTLICH
ZIERLICHER

```
ZIERLICHERE
ZOOM
ZU BILLIG
ZU BILLIGE
ZU BREIT
ZU DICK
ZU DÜNN
ZU GROSS
ZU GÜNSTIG
ZU HOCH
ZU KLEIN
ZU KLOBIG
ZU KOMPAKT
ZU KOMPAKTE
ZU LEICHT
ZUMINDEST
ZU NIEDRIG
ZU RIESIG
ZURÜCK
ZU SCHMAL
ZU SCHWER
ZU TEUER
ZU VIEL
ZU WENIG
ZU ZIERLICH
```

```
(\d+ ?[,.]? ?\d*)
(\d+ ?[,.]? ?\d*)? ?EURO
(\d+ ?[,.]? ?\d*)? ?FACHE?N? DIGITALEM ZOOM
(\d+ ?[,.]? ?\d*)? ?FACHE?N? DIGITALEM ZOOM
(\d+ ?[,.]? ?\d*)? ?FACHE?N? OPTISCHEN ZOOM
(\d+ ?[,.]? ?\d*)? ?FACHE?N? OPTISCHEN ZOOM
(\d+ ?[,.]? ?\d*)? ?FACHE?N? ZOOM
(\d+ ?[,.]? ?\d*)? ?FACHE?N? ZOOM
(\d+ ?[,.]? ?\d*)? ?GRAMM
(\d+ ?[,.]? ?\d*)? ?MEGABYTE
(\d+ ?[,.]? ?\d*)? ?MEGABYTE
(\d+ ?[,.]? ?\d*)? ?MEGAPIXEL
(\d+ ?[,.]? ?\d*)? ?MEGAPIXEL
(\d* ?[,.]? ?\d*)? ?ZOLL
GLEICH ?(\d+ ?[,.]? ?\d*)?
HÖCHSTENS ?(\d+ ?[,.]? ?\d*)?
HÖHER ALS (\d+ ?[,.]? ?\d*)
HÖHER WIE (\d+ ?[,.]? ?\d*)
MAXIMAL ?(\d+ ?[,.]? ?\d*)?
MEHR ALS (\d+ ?[,.]? ?\d*)
MEHR WIE (\d+ ?[,.]? ?\d*)
NIEDRIGER ALS (\d+ ?[,.]? ?\d*)
NIEDRIGER WIE (\d+ ?[,.]? ?\d*)
UNGLEICH ?(\d+ ?[,.]? ?\d*)?
UNTER ?(\d+ ?[,.]? ?\d*)?
WENIGER ALS (\d+ ?[,.]? ?\d*)
WENIGER WIE (\d+ ?[,.]? ?\d*)
```

A.2. Product Attributes

Table A.1 lists the product attributes recorded in ReComment's product database.

Attribute	Type
Model	String
Manufacturer	String
Price (€)	Number
Resolution (Megapixel)	Number
Sensor size (inches)	Number
Sensor type	Number
Size (w×h×d)	Number
Weight (gram)	Number
Internal memory (megabyte)	Number
Digital zoom (times)	Number
Optical zoom (times)	Number
External storage	String
Product Sales Rank	Number

Table A.1.: ReComment: Product attributes.

A.3. Questionnaire

The following section lists the questionnaires for participants of the empirical study conducted to evaluate ReComment. The formatting has been adapted to the style of this thesis. Questions shown below are in their original German form.

Every participant received both the demographic questions shown in Table A.2 and a set of general questions shown in Table A.3. The group evaluating the speech-based interface further answered the questions listed in Table A.4, whereas participants of the group testing the mouse-based interface further received the questions shown in Table A.5.

E-Mail (für die Verlosung eines Amazon Gutscheins):	______
Geschlecht:	□M □W
Alter:	______
Beruf:	______
Ich besitze eine Digitalkamera:	□Ja □Nein
Jahr des Kaufes:	______
Ich verwende meine Digitalkamera für:	□Beruf □Hobby □Freizeit / Alltag Sonstiges: ______

Table A.2.: ReComment questionnaire: Demographic questions.

Bitte markieren Sie das zutreffende Feld neben der jeweiligen Frage mit einem ×. Können Sie eine Frage nicht beantworten, so kreuzen Sie stattdessen bitte 'Keine Angabe' an.

	Stimme gar nicht zu			Stimme voll zu	Keine Angabe
Ich bin technisch versiert.	☐	☐	☐	☐	☐
Ich beschäftige mich aktuell auch privat mit Digitalkameras.	☐	☐	☐	☐	☐
Ich fühle mich unsicher beim Kauf einer Digitalkamera.	☐	☐	☐	☐	☐
Beim Kauf meiner letzten Digitalkamera habe ich mich von einem Fachhändler beraten lassen.	☐	☐	☐	☐	☐
Ich benötige viel Zeit, bevor ich eine Digitalkamera kaufe.	☐	☐	☐	☐	☐
Ich kann mir vorstellen, Recomment vor dem nächsten Kauf einer Digitalkamera zu nutzen.	☐	☐	☐	☐	☐
Das zuletzt vorgeschlagene Produkt entspricht meinen Anforderungen an eine Digitalkamera.	☐	☐	☐	☐	☐
Ich empfinde Recomment als unnötig komplex.	☐	☐	☐	☐	☐
Ich denke, dass ich technischen Support brauchen würde, um Recomment zu nutzen.	☐	☐	☐	☐	☐
Ich empfinde Recomment als einfach zu nutzen.	☐	☐	☐	☐	☐
Ich empfinde die Bedienung als sehr umständlich.	☐	☐	☐	☐	☐
Ich habe mich bei der Nutzung von Recomment sehr sicher gefühlt.	☐	☐	☐	☐	☐

continued on next page

Bitte markieren Sie das zutreffende Feld neben der jeweiligen Frage mit einem ×. Können Sie eine Frage nicht beantworten, so kreuzen Sie stattdessen bitte 'Keine Angabe' an.

	Stimme gar nicht zu			Stimme voll zu	Keine Angabe
Ich kann mir vorstellen, dass die meisten Leute Recomment schnell beherrschen werden.	☐	☐	☐	☐	☐

Table A.3.: ReComment questionnaire: General questions.

Bitte markieren Sie das zutreffende Feld neben der jeweiligen Frage mit einem ×. Können Sie eine Frage nicht beantworten, so kreuzen Sie stattdessen bitte 'Keine Angabe' an.

	Stimme gar nicht zu			Stimme voll zu	Keine Angabe
Die Spracheingabe vereinfacht die Nutzung von Recomment.	☐	☐	☐	☐	☐
Recomment versteht meine Spracheingaben.	☐	☐	☐	☐	☐
Durch die Spracheingabe wird die Nutzung von Recomment erschwert.	☐	☐	☐	☐	☐
Vorrausgesetzt die Spracheingabe in Recomment funktioniert zuverlässig, könnte ich mir vorstellen sie einem traditionelleren, Maus-basiertem Interface vorzuziehen.	☐	☐	☐	☐	☐
Die Spracheingabe in Recomment funktioniert nicht zuverlässig.	☐	☐	☐	☐	☐
Selbst wenn Recomment mich perfekt versteht, würde ich lieber nicht mit einem Computer reden.	☐	☐	☐	☐	☐

continued on next page

Bitte markieren Sie das zutreffende Feld neben der jeweiligen Frage mit einem ×. Können Sie eine Frage nicht beantworten, so kreuzen Sie stattdessen bitte 'Keine Angabe' an.

	Stimme gar nicht zu			Stimme voll zu	Keine Angabe
Ich habe bereits mit Systemen mit Spracheingabe gearbeitet.	☐	☐	☐	☐	☐
Am Recomment Interface würde ich folgendes ändern:					
Bitte geben Sie mindestens drei Domänen (außer Digitalkameras) an, in denen Sie sich die Nutzung von Recomment ebenfalls vorstellen könnten:					
Bitte geben Sie mindestens drei Vorzüge an, die die Spracheingabe in Recomment im Vergleich zu einem Maus-basiertem Interface haben könnte:					

Table A.4.: ReComment questionnaire: Questions for participants reviewing the speech-based interface.

Bitte markieren Sie das zutreffende Feld neben der jeweiligen Frage mit einem ×. Können Sie eine Frage nicht beantworten, so kreuzen Sie stattdessen bitte 'Keine Angabe' an.

	Stimme gar nicht zu			Stimme voll zu	Keine Angabe
Mit den angezeigten Buttons kann ich meine tatsächlichen Präferenzen gut abbilden.	□	□	□	□	□
Ich könnte mir vorstellen, dass es bessere Interfaces (z.B. Spracheingabe) für Recomment gibt.	□	□	□	□	□
Mit den derzeitigen Interaktionsmöglichkeiten ist es nicht möglich, meine Präferenzen in Recomment ausreichend gut abzubilden.	□	□	□	□	□
Bitte geben Sie mindestens drei Domänen (außer Digitalkameras) an, in denen Sie sich die Nutzung von Recomment ebenfalls vorstellen könnten:					
Nennen Sie bitte mindestens 3 Situationen, in denen Sie Empfehlungssysteme mit Mausinteraktion gegenüber alternativen Interfaces (z.B. Spracheingabe) vorziehen würden.					

Table A.5.: ReComment questionnaire: Questions for participants reviewing the mouse-based interface.

Appendix B.

Towards Human-like Recommender Systems

This chapter provides additional supporting material surrounding SpeechRec, and its evaluation.

B.1. Language Definition

The following list shows all key phrases detected by SpeechRec expressed as regular expressions. Alternate (dialectal) spellings have been omitted.

```
ABDECKUNG
ABMESSUNGEN
ACER
AKKU
AKKUART
AKKUKAPAZITÄT
AKKU KAPAZITÄT
AKKULADEZEIT
AKKULAUFZEIT
AKKU-LAUFZEIT
AKKULEISTUNG
AKKUTYP
AKKUZEIT
AKKUZELLEN
ALTERNATIVE
ALU
ALUMINIUM
ALUMINUM
AMD
ANBINDUNGEN
ANDERE
ANDEREN
ANDERER
ANDERES
ÄNDERN
ANLEITUNG
ANLEITUNGEN
ANSCHLÜSSE
ANSCHLUSSMÖGLICHKEITEN
ANSICHT
ANZEIGE
APPLE
APPS
ARBEIT
ARBEITEN
ARBEITSGERÄT
ARBEITSGESCHWINDIGKEIT
ARBEITSSPEICHER
ARBEITSSPEICHERTYP
ARBEITSZEIT
ARTIKELGEWICHT
ASIN
ASUS
AUDIO AUSGANG
AUFLÖSUNG
AUFRÜSTUNG
AUSREICHEND
AUSSEHEN
```

AUSSER
AUSSTATTUNG
BACKUP
BASS
BATTERIE
BATTERIEART
BATTERIELAUFZEIT
BATTERIETYP
BATTERIEZELLEN
BAUWEISE
BEDIENUNG
BEIBEHALTEN
BELEUCHTETE TASTATUR
BELEUCHTUNG
BENCHMARKS
BERECHNUNGEN
BERUFLICH
BERUFLICHE
BESSER
BESSERE
BESSEREN
BESSERER
BESSERES
BETRACHTUNGSWINKEL
BETRIEBSSYSTEM
BETRIEBSZEIT
BEWERTUNG
BILD
BILDQUALITÄT
BILDSCHIRM
BILDSCHIRMAUFLÖSUNG
BILDSCHIRM AUFLÖSUNG
BILDSCHIRMDIAGONALE
BILDSCHIRMGRÖSSE
BILDSCHIRMHELLIGKEIT
BILLIG
BILLIGE
BILLIGEN
BILLIGER
BILLIGERE
BILLIGEREN
BILLIGERES
BIN ZUFRIEDEN
BISSCHEN
BISSERL
BLEIBEN
BLICKWINKEL
BODY
BOOTVORGANG
BOOTZEIT
BOXEN
BRAUCHE ICH NICHT
BRAUCH ICH NICHT
BREITE
BREITER
BREITERE
BREITES
BRENNER
BS
CARDREADER
CD
CD-LAUFWERK
CHIPSATZ
CHIPSATZHERSTELLER
COMPAQ
COMPUTERSPIELE
CONVERTIBLE
COOL
CORES
CPU
DARSTELLUNG
DAS GEWÜNSCHTE
DAS LETZTE
DAS RICHTIGE
DAS VORHERIGE
DAS VORIGE
DATENSPEICHER
DATENVERARBEITUNG
DECKEL
DELL
DESIGN
DETAILS
DEUTLICH
DIAGONALE
DICKE
DICKER
DICKERE
DICKES
DIGITIZER
DISPLAY
DISPLAYHELLIGKEIT
DISPLAYPORT
DISPLAYRAHMEN
DISPLAYTYP
DOKUMENTATION
DRUCKPUNKT
DUALCORE
DÜNNEN
DÜNNER
DÜNNERE
DÜNNES
DVD
DVD-BRENNER
DVD-LAUFWERK
DVI
EHER
EINGABEGERÄTE
EINRICHTUNG
EMAILS
ENTSPIEGELT
ENTSPIEGELTEM
ENTSPIEGELTER
ENTSPRECHENDE
ENTSPRECHENDEM
ENTSPRECHENDEN
ENTSPRECHENDES
ENTWICKLUNGSMASCHINE
ERHÖHEN
ESATA
ETHERNET
ETWAS
ETWAS BILLIG
ETWAS BREIT
ETWAS DICK
ETWAS DÜNN
ETWAS GROSS
ETWAS HOCH
ETWAS KLOBIG
ETWAS LAHM
ETWAS LANGSAM
ETWAS NIEDRIG
ETWAS SCHMAL

```
ETWAS SCHWER
ETWAS TEUER
ETWAS VIEL
ETWAS WENIG
EURO
EURO KOSTEN
EXPRESS CARD
FAN
FARBE
FARBEN
FESTPLATTE
FESTPLATTEN
FESTPLATTENGRÖSSE
FESTPLATTENINTERFACE
FESTPLATTENKAPAZITÄT
FESTPLATTENSPEICHER
FESTPLATTEN-TECHNOLOGIE
FILM
FILME
FINGERABDRUCK
FINGERABDRUCKLESER
FINGERABDRUCKSCANNER
FIREWIRE
FIRMA
FLACHEN
FLACHER
FLACHES
FORMAT
FORMFAKTOR
FOTOS
FREQUENZ
FUJITSU
FUNKTIONALITÄT
FUNKTIONEN
FUNKTIONSTASTEN
FUNKTIONSUMFANG
GAMEN
GARANTIE
GARANTIEART
GARANTIEDAUER
GARANTIEZEIT
GARNATIE
GEFÄLLT MIR
GEHÄRTETER BILDSCHIRM
GEHÄUSE
GENUG
GENÜGEND
GERÄUSCH
GERÄUSCHENTWICKLUNG
GERINGEREN
GERINGERER
GERINGERERE
GERINGERES
GESAMTEINDRUCK
GESCHEIT
GESCHWINDIGKEIT
GESTEN
GESTENSTEUERUNG
GEWICHT
GIGABYTE
GIGABYTE ARBEITSSPEICHER
GIGABYTE HAUPTSPEICHER
GIGABYTE RAM
GLEICH
GLEICHE
GLEICHEM
GLEICHEN
GLEICHER
GLEICHES
GORILLA GLAS
GRAFIK
GRAFIKBEARBEITUNG
GRAFIKCHIP
GRAFIKKARTE
GRAFIKKARTEN
GRAFIKKARTENMARKE
GRAFIKKARTENSPEICHER
GRAFIKKARTENTYP
GRAFIKLEISTUNG
GRAFIKSPEICHER
GRAFIKTREIBER
GROSSE
GRÖSSE
GROSSEN
GROSSER
GRÖSSER
GRÖSSERE
GRÖSSEREM
GRÖSSEREN
GRÖSSERER
GRÖSSERES
GÜNSTIG
GÜNSTIGE
GÜNSTIGEN
GÜNSTIGER
GÜNSTIGERE
GÜNSTIGEREN
GÜNSTIGERES
GÜNSTIGES
GUT
GUTE
GUTEN
GUTER
GUTES
GUTES GERÄT
HANDBALLENAUFLAGE
HANDBUCH
HANDHABUNG
HANDLING
HAPTIK
HAUPTSPEICHER
HAUPTSPEICHERTYP
HDD
HDMI
HDMI-ANSCHLUSS
HELLIGKEIT
HERAUSRAGEND
HERSTELLER
HERSTELLERGARANTIE
HERSTELLERN
HINTERGRUNDBELEUCHTUNG
HOCH
HÖCHSTENS
HOCHWERTIGER
HOCHWERTIGERE
HOCHWERTIGEREN
HOCHWERTIGERES
HÖHE
HOHEM
HOHEN
HOHER
HÖHER
```

HÖHER ALS
HÖHERE
HÖHEREN
HÖHERER
HÖHERERE
HÖHERES
HÖHER WIE
HOHES
HORIZONTALE AUFLÖSUNG
HOTLINE
HP
HUB
I3
I5
I7
ID
INBETRIEBNAHME
INKLUDIERTE SOFTWARE
IN ORDNUNG
INSTALLATION
INTEL
INTERNETRECHERCHEN
IST ETWAS KLEIN
IST OK
IST VON VORTEIL
JA
KABEL
KAMERA
KAMERAS
KANTE
KANTEN
KARTENLESER
KAUFEN
KEIN
KEINE
KEINEN
KERNE
KEYBOARD
KILO
KILOGRAMM
KLANG
KLEINE
KLEINEM
KLEINEN
KLEINER
KLEINERE
KLEINEREM
KLEINEREN
KLEINERER
KLEINES
KLINKE
KLOBIG
KOMMT HIN
KOMMUNIKATION
KOMPAKT
KOMPAKTE
KOMPAKTER
KOMPAKTERE
KOMPAKTES
KOMPLETT
KONNEKTIVITÄT
KONTRAST
KONTRASTE
KONVERTIERBAR
KOSTEN
KRATZFEST
KRATZFESTEN
KÜHLUNG
KUNDENBEWERTUNG
KUNDENSERVICE
KUNSTSTOFF
LACK
LADEGERÄT
LADEKABEL
LAN
LANG
LANGE
LÄNGE
LANGEN
LANGER
LÄNGERE
LÄNGEREN
LAUFWERK
LAUFZEIT
LAUTSPRECHER
LAUTSPRECHERN
LAUTSTÄRKE
LEBENSDAUER
LEERTASTE
LEICHT
LEICHTER
LEICHTERE
LEICHTIGKEIT
LEISTUNG
LEISTUNGEN
LEISTUNGSAUFNAHME
LEISTUNGSFÄHIGKEIT
LEISTUNGSSTARK
LEISTUNGSSTARKEN
LEISTUNGSSTARKER
LEISTUNGSSTARKES
LENOVO
LINUX
LINUX BETRIEBSSYSTEM
LISTENPREIS
LÖSCHEN
LÜFTER
LÜFTERGERÄUSCH
LÜFTERGERÄUSCHE
LÜFTUNG
MAC
MACBOOK
MAC OS
MAC OS X
MAGNESIUM
MAINBOARD
MARGINAL
MARKE
MASSE
MATERIAL
MATERIALIEN
MATT
MATTEM
MATTER
MATTER BILDSCHIRM
MATTES
MATTES DISPLAY
MAUS
MAUSPAD
MAUSTASTE
MAUSTASTEN
MAVERICKS

MAXIMAL
MEDION
MEHR
MEHR ALS
MEHR WIE
METALL
MICRO HDMI
MICROSOFT
MINDESTENS
MINI DISPLAYPORT
MINI HDMI
MINI VGA
MIT
MITNEHMEN
MOBILITÄT
MODELL
MONITOR
MOTHERBOARD
MOUSE
MOUSEPAD
MSI
MUSIK
NAJA
NAME
NAMEN
NEHME ICH
NEIN
NETZKABEL
NETZSTECKER
NETZTEIL
NICHT
NICHT BESONDERS
NICHT NOTWENDIGERWEISE
NICHTS
NICHT SCHLECHT
NICHT SO SCHLECHT
NICHT SO TEUER
NICHT SO ÜBEL
NICHT ÜBEL
NICHT UNBEDINGT
NICHT ZU GROSS
NICHT ZU GROSSEN
NICHT ZU GROSSER
NICHT ZU GROSSES
NICHT ZU KLEIN
NICHT ZU KLEINEM
NICHT ZU KLEINEN
NICHT ZU KLEINER
NICHT ZU KLEINES
NICHT ZU SCHWER
NICHT ZU TEUER
NIEDEREM
NIEDEREN
NIEDERER
NIEDERES
NIEDRIGEM
NIEDRIGEN
NIEDRIGER
NIEDRIGER ALS
NIEDRIGERE
NIEDRIGEREN
NIEDRIGERER
NIEDRIGERES
NIEDRIGER WIE
NIEDRIGES
NIMM ICH
NIPPEL
NOCH EINMAL ZEIGEN
NOCH MAL ZEIGEN
NOTWENDIG
NUMMERNBLOCK
OBERFLÄCHE
OBERFLÄCHEN
OFFICE
OHNE
OK
OPTIK
OPTISCHE MEDIEN
OPTISCHER SPEICHER
OPTISCHES LAUFWERK
OPTISCHES MEDIUM
ORDENTLICH
OS
OS X
PAD
PAKET
PANASONIC
PANEL
PCMCIA
PERFORMANCE
PERMANENTEN SPEICHER
PFEILTASTEN
PLASTIK
PLATTE
PLATTFORM
POINTING STICK
PORTABEL
PORTABELE
PORTABELEN
PORTABELER
PORTS
PREIS
PREISKLASSE
PREIS LEISTUNGS VERHÄLTNIS
PREIS-LEISTUNGSVERHÄLTNIS
PREIS-/LEISTUNGSVERHÄLTNIS
PREIS-LEISTUNGS-VERHÄLTNIS
PREIS/LEISTUNGSVERHÄLTNIS
PREISLICH
PREISSEGMENT
PREISWERTER
PREISWERTERE
PREISWERTEREN
PREISWERTERES
PRODUKTABMESSUNGEN
PROGRAMME
PROGRAMMIEREN
PROZESSOR
PROZESSORGESCHWINDIGKEIT
PROZESSORKERNE
PROZESSORLEISTUNG
PROZESSORMARKE
PROZESSOR MARKE
PROZESSOR NAME
PROZESSORTAKT
PROZESSORTYP
QUADCORE
QUALITÄT
RAM
RATING
RECHENLEISTUNG
RECHERCHEN

RECHT
RELATIV
ROBUST
ROBUSTEREN
SAMSUNG
SCHARNIERE
SCHMALE
SCHMALEN
SCHMALER
SCHMALERE
SCHNELL
SCHNELLEN
SCHNELLER
SCHNELLES
SCHNELLIGKEIT
SCHNITT
SCHNITTSTELLEN
SCHÖN
SCHRAUBEN
SCHRIFT
SCHWER
SCHWERER
SCHWERERE
SCREEN
SEHR
SELBE
SELBEN
SELBER
SELBES
SERVICE
SICHER
SIGNIFIKANT
SMART CARD
SOFTWARE
SOFTWAREENTWICKLUNG
SONY
SOUND
SPALTMASSE
SPEICHER
SPEICHERKARTEN
SPEICHERPLATZ
SPIELE
SPIELEN
SPUR
SSD
SSD FESTPLATTE
SSD-FESTPLATTE
SSDS
SSD-SPEICHER
STABIL
STABILER
STABILERE
STABILES
STABILITÄT
STANDBY-ZEIT
STARK
STARKE
STARKEN
STARKER
STÄRKER
STÄRKERE
STÄRKEREN
STÄRKERER
STÄRKERES
STARKES
STECKER
STIFT
STROMQUELLE
STROMVERBRAUCH
SUBWOOFER
SUPPORT
SURFEN
TAKT
TAKTFREQUENZ
TAROX
TASTATUR
TASTATURBELEUCHTUNG
TASTATUREN
TASTE
TASTEN
TEMPERATUR
TERRA
TEUER
TEURER
TEURERE
TEUREREN
TEURERES
TEXTVERARBEITUNG
THUNDERBOLT
TON
TOSHIBA
TOUCH
TOUCHDISPLAY
TOUCHPAD
TOUCHSCREEN
TOUCHSCREEN TYP
TRACKING POINT
TRACKPAD
TRACKPOINT
TRANSPORTABEL
TRANSPORTIEREN
TREIBER
TREIBERN
ÜBER
UM
UNGEFÄHR
UNGLEICH
UNI
UNIVERSITÄT
UNIVERSITÄTS
UNTER
USB
USB 2
USB 3
USB-ANSCHLUSS
USB-ANSCHLÜSSE
USB-BUCHSEN
VERARBEITUNG
VERARBEITUNGSQUALITÄT
VERBINDUNG
VERBINDUNGEN
VERFÜGBAR SEIT
VERGESSEN
VERGISS DAS
VERHÄRTETEN
VERPACKUNG
VERRINGERN
VERTIKALE AUFLÖSUNG
VGA
VIDEO
VIDEOS
VIEL

```
VIRTUELLE MASCHINE
VORHERIGES MODELL
VORIGES MODELL
WÄRE VON VORTEIL
WÄRME
WÄRMEENTWICKLUNG
WEBCAM
WEBCAM AUFLÖSUNG
WECHSELN
WEITER
WENIG
WENIGER
WENIGER ALS
WENIGER WICHTIG
WENIGER WIE
WENIGSTENS
WERTIG
WERTIGE
WERTIGEN
WERTIGERE
WERTIGEREN
WESENTLICH
WILL ICH HABEN
WILL ICH NICHT
WINDOWS
WINDOWS 7
WINDOWS 8
WINDOWS 8.1
WINDOWS BETRIEBSSYSTEM
WIRELESS
WLAN
WORKSTATION
WORTMANN
WÜRDE MIR ZUSPRECHEN
WÜRDE PASSEN
ZIEMLICH WICHTIG
ZIERLICHER
ZIERLICHERE
ZIFFERNBLOCK
ZOCKEN
ZOLL
ZUBEHÖR
ZU BILLIG
ZU BILLIGE
ZU BILLIGEN
ZU BILLIGER
ZU BREIT
ZU BREITEN
ZU BREITER
ZU BREITES
ZU DICK
ZU DICKEN
ZU DICKER
ZU DICKES
ZU DÜNN
ZU GROSS
ZU GROSSEN
ZU GROSSER
ZU GROSSES
ZU GÜNSTIG
ZU GÜNSTIGEN
ZU GÜNSTIGER
ZU HOCH
ZU KLEIN
ZU KLEINEM
ZU KLEINEN
ZU KLEINER
ZU KLEINES
ZU KLOBIG
ZU KLOBIGEN
ZU KLOBIGER
ZU KLOBIGES
ZU LAHM
ZU LANGSAM
ZU LEICHT
ZUMINDEST
ZU NIEDRIG
ZU RIESIG
ZURÜCK
ZU SCHMAL
ZU SCHWER
ZU TEUEREN
ZU VIEL
ZU WENIG
ZU ZIERLICH
```

```
DURCHSCHNITTLICHE LAUFZEIT
ENTSPRICHT( \w+){0,3} ERWARTUNGEN
ERWARTUNGEN( \w+){0,2} ENTSPRECHEN
GEFÄLLT MIR( \w+){0,3} NICHT
HÄTTE ICH( \w+){0,1} GERNE
HÄTT ICH( \w+){0,1} GERNE
KÖNNTE MICH( \w+){0,1} MIT DEM( \w+){0,2} ANFREUNDEN
MUSS( \w+){0,3} NICHT SEIN
NICHT( \w+){0,3} SPIEGELND
NICHT( \w+){0,3} SPIEGELT
NICHT( \w+){0,3} WICHTIG
PRODUKTGEWICHT INKLUSIVE VERPACKUNG
UM( \w+){0,4} ZU TEUER
WINDOWS 7 BETRIEBSSYSTEM
```

```
WINDOWS 8.1 BETRIEBSSYSTEM
WINDOWS 8 BETRIEBSSYSTEM
WÜRDE MIR AUCH ZUSPRECHEN
```

B.2. Product Attributes

Table B.1 lists all factual product attributes recorded in SpeechRec's product database.

Attribute	Type
Model name	String
Manufacturer name	String
User rating	Number
Price	Number
Operating system	(Android | Mac OS X | Windows 7 | Windows 8 | Chrome OS | FreeDOS | Linux | None)
CPU brand	(Intel | AMD | ARM)
CPU frequency per core	Number
Number of CPU cores	Number
CPU name	String
Main memory capacity	Number
Graphics card brand	(Intel | AMD | Nvidia | ARM)
Graphics card name	String
Graphics memory	Number
Screen size in inches	Number
Horiz. screen resolution	Number
Vertical screen resolution	Number
Touch screen type	(None | Single | Multi)
Anti glare coating	Boolean
Screen panel technology	(TN | IGZO | IPS)
Screen hardened	Boolean

continued on next page

Attribute	Type
Digitizer support	Boolean
Storage media	[((SSD | SSD-Cache | HDD); Number)]
Supported optical Media	[(DVD-RW | BluRay | BluRay-RW)]
Connectivity options	[(Wifi-A | Wifi-AC | Wifi-B | Wifi-G | Wifi-N | Bluetooth | NFC | Celluar | GPS)]
Battery technology	(LiIon | LiPolymer)
Battery cells	Number
Available ports	[(USB 2 | USB 3 | Lan 100M | Lan 1000M | Thunderbolt | Firewire | ExpressCard 54 | SmartCard | PCMCIA | E-SATA | Micro HDMI | Mini HDMI | Mini DisplayPort | HDMI | DisplayPort | VGA | DVI)]
Weight	Number
Width	Number
Length	Number
Height	Number
Warranty duration	Number
Warranty type	(SendIn | PickUp | OnSite)
Webcam resolution	Number
Supported memory cards	[(SD | MMC | MMCPlus | MS | MSPro | XD)]
Number block	Boolean
Pointing stick	Boolean
Backlit keyboard	Boolean
Finger print reader	Boolean
Convertible	Boolean

Table B.1.: SpeechRec: Product attributes.

B.3. Aspects

Figure B.1 shows an overview of the known aspects of the domain. Please note that activation keywords of these aspects are not necessarily limited to synonyms of the aspect's names. For example, "Mavericks" triggers the "OS" aspect.

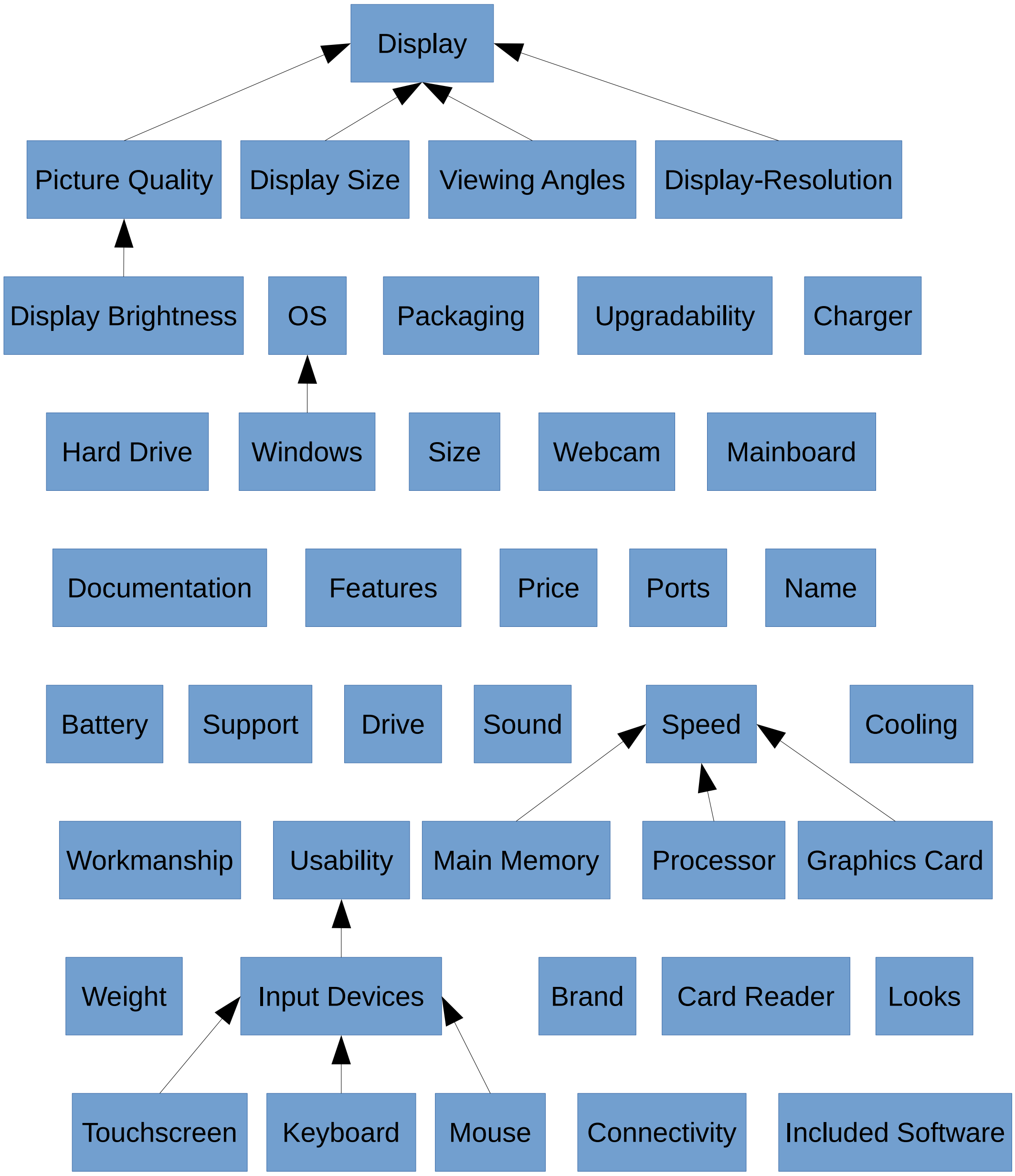

Figure B.1.: SpeechRec's recognized product aspects.

B.4. Questionnaire

The following section shows questionnaires for participants of the empirical study conducted to evaluate SpeechRec. The formatting has been adapted to follow the style of this thesis. Questions shown below are in their original German form.

Every study participant received the demographic questions shown in Table B.2 and the questions evaluating SpeechRec's performance shown in Table B.3.

Geschlecht:	□M □W
Alter:	____
Beruf:	____
Ich besitze einen Laptop:	□Ja □Nein
Jahr des Kaufes:	____
Ich verwende meinen Laptop für:	□Beruf / Universität
	□Hobby
	□Freizeit / Alltag
	Sonstiges: ____

Table B.2.: SpeechRec questionnaire: Demographic questions.

Bitte markieren Sie das zutreffende Feld neben der jeweiligen Frage mit einem ×. Können Sie eine Frage nicht beantworten, so kreuzen Sie stattdessen bitte 'Keine Angabe' an.

	Stimme gar nicht zu			Stimme voll zu	Keine Angabe
Ich bin technisch versiert.	☐	☐	☐	☐	☐
Ich kenne die Laptops, die derzeit auf dem Markt sind.	☐	☐	☐	☐	☐
Ich fühle mich unsicher beim Kauf eines Laptops.	☐	☐	☐	☐	☐
Beim Kauf meines letzten Laptops habe ich mich von einem Fachhändler beraten lassen.	☐	☐	☐	☐	☐
Ich benötige viel Zeit, bevor ich einen Laptop kaufe.	☐	☐	☐	☐	☐
Das zuletzt vom System vorgeschlagene Produkt entspricht meinen Anforderungen an einen Laptop.	☐	☐	☐	☐	☐
Ich kann mir vorstellen, Spencer vor dem nächsten Kauf eines Laptops zu nutzen.	☐	☐	☐	☐	☐
Das Produkt, dass ich schlussendlich ausgewählt habe, gefällt mir nicht.	☐	☐	☐	☐	☐
Ich empfinde Spencer als unnötig komplex.	☐	☐	☐	☐	☐
Ich denke, dass ich technischen Support brauchen würde, um Spencer zu nutzen.	☐	☐	☐	☐	☐
Ich empfinde Spencer als einfach zu nutzen.	☐	☐	☐	☐	☐
Ich empfinde die Bedienung als sehr umständlich.	☐	☐	☐	☐	☐

continued on next page

Bitte markieren Sie das zutreffende Feld neben der jeweiligen Frage mit einem ×. Können Sie eine Frage nicht beantworten, so kreuzen Sie stattdessen bitte 'Keine Angabe' an.

	Stimme gar nicht zu			Stimme voll zu	Keine Angabe
Ich habe mich bei der Nutzung von Spencer sehr sicher gefühlt.	☐	☐	☐	☐	☐
Ich kann mir vorstellen, dass die meisten Leute Spencer schnell beherrschen werden.	☐	☐	☐	☐	☐
Die Spracheingabe vereinfacht die Nutzung von Spencer.	☐	☐	☐	☐	☐
Mit der Spracheingabe ist es einfach, meine tatsächlichen Präferenzen anzugeben.	☐	☐	☐	☐	☐
Spencer versteht meine Spracheingaben.	☐	☐	☐	☐	☐
Durch die Spracheingabe wird die Nutzung von Spencer erschwert.	☐	☐	☐	☐	☐
Vorrausgesetzt die Spracheingabe in Spencer funktioniert zuverlässig, kann ich mir vorstellen dass ich sie einem traditionelleren, Maus-basiertem Interface vorziehen würde.	☐	☐	☐	☐	☐
Mit Spencers derzeitigen Interaktionsmöglichkeiten ist es nicht möglich, meine Präferenzen ausreichend gut abzubilden.	☐	☐	☐	☐	☐
Die Spracheingabe in Spencer funktioniert nicht zuverlässig.	☐	☐	☐	☐	☐
Selbst wenn Spencer mich perfekt versteht, würde ich lieber nicht mit einem Computer reden.	☐	☐	☐	☐	☐

continued on next page

Bitte markieren Sie das zutreffende Feld neben der jeweiligen Frage mit einem ×. Können Sie eine Frage nicht beantworten, so kreuzen Sie stattdessen bitte 'Keine Angabe' an.	Stimme gar nicht zu			Stimme voll zu	Keine Angabe
Ich habe bereits mit Systemen mit Spracheingabe gearbeitet.	☐	☐	☐	☐	☐
Am Spencer Interface würde ich folgendes ändern:					

Table B.3.: SpeechRec questionnaire: General questions.

List of Figures

2.1. SpeechRec: Speech processing architecture. 8
2.2. ReComment: Mouse-based user interface. 12
2.3. ReComment: Speech-based user interface. 13
2.4. ReComment: Guiding user input. 14
2.5. ReComment's utility function of the critique $x > 50$. 17
2.6. ReComment's utility functions of the subsequent critiques $x > 50$ and $x < 100$, as well as $x > 50$ and $x <_{slightly} 100$. . . . 18
2.7. Comparison of the ReComment's speech recognition system with a state of the art off-the-shelf online service. 24
2.8. ReComment: Participants' perception of speech recognition accuracy. 25
2.9. ReComment usability evaluation. 26
2.10. ReComment evaluation: "Did the speech recognition make ReComment easier to use?" 27
2.11. ReComment: User score of last recommended item 28
2.12. ReComment: Session length. 28

3.1. SpeechRec: Speech processing architecture. 33
3.2. SpeechRec's dialog strategy: An overview. 39
3.3. SpeechRec taking initiative after the user expressed no particular preferences. (Text enlarged for readability and translated from German.) 40
3.4. SpeechRec recommending a product. 42
3.5. SpeechRec: Participant's subjective score of the last shown product of the interaction session. 51
3.6. SpeechRec: Recommendation session duration. 52
3.7. SpeechRec: Recommendation session length in completed iterations. 53

B.1. SpeechRec's recognized product aspects. 84

List of Tables

2.1. ReComment: Product attributes. 8
2.2. ReComment: Selected language model mixture weights. . . . 10
2.3. ReComment: Sample adjectives that affect the modifier factor. 17
2.4. ReComment: Demography of the user groups of the empirical study. 22
2.5. ReComment: Sample user interaction. 25

3.1. SpeechRec: Selected language model mixtures. 34
3.2. SpeechRec: Sample user interaction. 43
3.3. SpeechRec: Demography of the participants of the empirical study. 49

A.1. ReComment: Product attributes. 68
A.2. ReComment questionnaire: Demographic questions. 69
A.3. ReComment questionnaire: General questions. 71
A.4. ReComment questionnaire: Questions for participants reviewing the speech-based interface. 72
A.5. ReComment questionnaire: Questions for participants reviewing the mouse-based interface. 73

B.1. SpeechRec: Product attributes. 83
B.2. SpeechRec questionnaire: Demographic questions. 85
B.3. SpeechRec questionnaire: General questions. 88

Bibliography

[1] Alina Andreevskaia, Sabine Bergler, and Monica Urseanu. "All Blogs Are Not Made Equal: Exploring Genre Differences in Sentiment Tagging of Blogs." In: *Intl. Conf. on Weblogs and Social Media*. Boulder, CO, USA, 2007 (cit. on p. 32).

[2] Derek Bridge. "Towards conversational recommender systems: A dialogue grammar approach." In: *Proceedings of the Workshop in Mixed-Initiative Case-Based Reasoning, Workshop Prog. at the 6th Europ. Conf. in CBR*. 2002, pp. 9–22 (cit. on p. 5).

[3] John Brooke. "SUS - A quick and dirty usability scale." In: *Usability evaluation in industry* 189 (1996), p. 194 (cit. on p. 23).

[4] Robin Burke. "Integrating knowledge-based and collaborative-filtering recommender systems." In: *Proc. of the Worksh. on AI and Electr. Commerce*. 1999, pp. 69–72 (cit. on p. 2).

[5] Robin Burke. "Knowledge-based recommender systems." In: *Encyclopedia of Library and Information Systems*. Vol. 69. Supplement 32. Marcel Dekker, 2000, pp. 175–186 (cit. on pp. 1–3, 30).

[6] Robin D Burke, Kristian J Hammond, and BC Yound. "The FindMe approach to assisted browsing." In: *IEEE Expert* 12.4 (1997), pp. 32–40 (cit. on p. 11).

[7] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. "Knowledge-based Navigation of Complex Information Spaces." In: *Proc. of the 13th Natl. Conf. on AI - Volume 1*. Vol. 462. AAAI'96. Portland, Oregon: AAAI Press, 1996, pp. 462–468. ISBN: 0-262-51091-X. URL: http://dl.acm.org/citation.cfm?id=1892875.1892944 (cit. on pp. 3–5, 21).

[8] Li Chen and Pearl Pu. "Evaluating Critiquing-based Recommender Agents." In: *Proc. of the 21st Natl. Conf. on AI - Volume 1*. Vol. 21. AAAI'06 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999. Boston, Massachusetts: AAAI Press, 2006, pp. 157–162. ISBN: 978-1-57735-281-5. URL: http://dl.acm.org/citation.cfm?id=1597538.1597564 (cit. on p. 3).

[9] Li Chen and Pearl Pu. *Survey of preference elicitation methods*. Tech. rep. Swiss Federal Inst. of Techn. in Lausanne (EPFL, 2004 (cit. on pp. 2, 3).

[10] Alexander Clark. "Combining Distributional and Morphological Information for Part of Speech Induction." In: *Proc. of the Tenth Conf. on European Chapter of the Association for Computational Linguistics - Volume 1*. EACL '03. Budapest, Hungary: Association for Computational Linguistics, 2003, pp. 59–66. ISBN: 1-333-56789-0. DOI: 10.3115/1067807.1067817. URL: http://dx.doi.org/10.3115/1067807.1067817 (cit. on p. 31).

[11] Sam-Joo Doh. "Enhancements to transformation-based speaker adaptation: principal component and inter-class maximum likelihood linear regression." PhD thesis. Carnegie Mellon University, 2000 (cit. on p. 10).

[12] Ruihai Dong et al. "Opinionated Product Recommendation." English. In: *Case-Based Reasoning Research and Development*. Ed. by SarahJane Delany and Santiago Ontañón. Vol. 7969. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 44–58. ISBN: 978-3-642-39055-5. DOI: 10.1007/978-3-642-39056-2_4. URL: http://dx.doi.org/10.1007/978-3-642-39056-2_4 (cit. on p. 6).

[13] Alexander Felfernig and Robin Burke. "Constraint-based Recommender Systems: Technologies and Research Issues." In: *Proc. of the 10th Intl. Conf. on Electronic Commerce*. ICEC '08. ACM. Innsbruck, Austria: ACM, 2008, 3:1–3:10. ISBN: 978-1-60558-075-3. DOI: 10.1145/1409540.1409544. URL: http://doi.acm.org/10.1145/1409540.1409544 (cit. on p. 30).

[14] Alexander Felfernig et al. "An Overview of Direct Diagnosis and Repair Techniques in the WeeVis Recommendation Environment." In: *Knowledge-based Configuration: From Research to Business Cases*. Elsevier, 2014, pp. 297–307 (cit. on p. 47).

[15] Alexander Felfernig et al. "Basic approaches in recommendation systems." In: *Recommendation Systems in Software Engineering*. Springer, 2014, pp. 15–37 (cit. on p. 1).

[16] A. Felfernig et al. "The VITA Financial Services Sales Support Environment." In: *Proc. of the 19th Natl. Conf. on Innovative Applications of AI - Volume 2*. IAAI'07. Vancouver, British Columbia, Canada: AAAI Press, 2007, pp. 1692–1699. ISBN: 978-1-57735-323-2. URL: http://dl.acm.org/citation.cfm?id=1620113.1620117 (cit. on p. 2).

[17] Peter Grasch and Alexander Felfernig. "On the Importance of Subtext in Recommender Systems." In: *i-com Special Issue on Recommender Systems* 1.1 (2015), to appear (cit. on pp. 45, 47, 52).

[18] Peter Grasch, Alexander Felfernig, and Florian Reinfrank. "ReComment: Towards critiquing-based recommendation with speech interaction." In: *Proc. of the 7th ACM conf. on Recommender systems*. ACM. 2013, pp. 157–164 (cit. on pp. 10, 13, 25).

[19] John B Hey. *System and method for recommending items*. US Patent 4,996,642. Feb. 1991 (cit. on p. 1).

[20] Christopher John Leggetter and PC Woodland. *Speaker adaptation of HMMs using linear regression*. University of Cambridge, Department of Engineering, 1994 (cit. on p. 10).

[21] Kevin McCarthy, Lorraine McGinty, and Barry Smyth. "Dynamic Critiquing: An Analysis of Cognitive Load." In: *Proc. of the 16th Irish Conf. on AI and Cognitive Science*. 2005, pp. 19–28 (cit. on p. 3).

[22] Kevin McCarthy, Yasser Salem, and Barry Smyth. "Experience-Based critiquing: reusing critiquing experiences to improve conversational recommendation." In: *Case-Based Reasoning. Research and Development*. Springer, 2010, pp. 480–494 (cit. on pp. 3, 5).

[23] Kevin McCarthy et al. "On the dynamic generation of compound critiques in conversational recommender systems." In: *Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer. 2004, pp. 176–184 (cit. on pp. 3, 19).

[24] Lorraine McGinty and James Reilly. "On the evolution of critiquing recommenders." In: *Recommender Systems Handbook*. Springer, 2011, pp. 419–453 (cit. on pp. 3, 5, 15).

[25] Lorraine McGinty and Barry Smyth. "On the Role of Diversity in Conversational Recommender Systems." In: *Proc. of the 5th Intl. Conf. on Case-based Reasoning: Research and Development*. ICCBR'03. Trondheim, Norway: Springer, 2003, pp. 276–290. ISBN: 3-540-40433-3. URL: http://dl.acm.org/citation.cfm?id=1760422.1760447 (cit. on p. 2).

[26] David McSherry and David W. Aha. "The Ins and Outs of Critiquing." In: *Proc. of the 20th Intl. Joint Conf. on Artifical Intelligence*. IJCAI'07. Hyderabad, India: Morgan Kaufmann Publishers Inc., 2007, pp. 962–967. URL: http://dl.acm.org/citation.cfm?id=1625275.1625431 (cit. on pp. 5, 15).

[27] Samaneh Moghaddam and Martin Ester. "Opinion Digger: An Unsupervised Opinion Miner from Unstructured Product Reviews." In: *Proc. of the 19th ACM Intl. Conf. on Information and Knowledge Management*. CIKM '10. Toronto, ON, Canada: ACM, 2010, pp. 1825–1828. ISBN: 978-1-4503-0099-5. DOI: 10.1145/1871437.1871739. URL: http://doi.acm.org/10.1145/1871437.1871739 (cit. on p. 6).

[28] Michael J. Pazzani and Daniel Billsus. "Content-based Recommendation Systems." In: *The Adaptive Web*. Ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Springer, 2007, pp. 325–341. ISBN: 978-3-540-72078-2. URL: http://dl.acm.org/citation.cfm?id=1768197.1768209 (cit. on p. 1).

[29] Pearl Huan Z. Pu and Pratyush Kumar. "Evaluating Example-based Search Tools." In: *Proc. of the 5th ACM Conf. on Electronic Commerce*. EC '04. ACM. New York, NY, USA: ACM, 2004, pp. 208–217. ISBN: 1-58113-771-0. DOI: 10.1145/988772.988804. URL: http://doi.acm.org/10.1145/988772.988804 (cit. on pp. 2, 3).

[30] Lingyun Qiu and Izak Benbasat. "An investigation into the effects of Text-To-Speech voice and 3D avatars on the perception of presence and flow of live help in electronic commerce." In: *ACM Trans. Comput.-Hum. Interact.* 12.4 (2005), pp. 329–355. ISSN: 1073-0516. DOI: 10.1145/1121112.1121113. URL: http://doi.acm.org/10.1145/1121112.1121113 (cit. on p. 40).

[31] James Reilly et al. "Evaluating compound critiquing recommenders: a real-user study." In: *Proceedings of the 8th ACM conference on Electronic commerce*. ACM. 2007, pp. 114–123 (cit. on p. 2).

[32] James Reilly et al. "Explaining compound critiques." In: *Artificial Intelligence Review* 24.2 (2005), pp. 199–220 (cit. on p. 19).

[33] James Reilly et al. "Incremental critiquing." In: *Knowledge-Based Systems* 18.4 (2005), pp. 143–151 (cit. on pp. 3, 5, 11, 15).

[34] James A Russell. "A circumplex model of affect." In: *Journal of personality and social psychology* 39.6 (1980), p. 1161 (cit. on p. 36).

[35] Barbara Schuppler, Martine Adda-Decker, and Juan A. Morales-Cordovilla. "Pronunciation variation in read and conversational Austrian German." In: *Interspeech 2014*. Singapore, 2014, pp. 1453–1457 (cit. on p. 35).

[36] Rico Sennrich, Martin Volk, and Gerold Schneider. "Exploiting Synergies Between Open Resources for German Dependency Parsing, POS-tagging, and Morphological Analysis." In: *RANLP*. 2013, pp. 601–609 (cit. on p. 32).

[37] Rico Sennrich et al. "A new hybrid dependency parser for German." In: *Proc. of the German Society for Comp. Linguistics and Language Techn.* (2009), pp. 115–124 (cit. on p. 32).

[38] Mostafa Al Shaikh, Helmut Prendinger, and Ishizuka Mitsuru. "Assessing Sentiment of Text by Semantic Dependency and Contextual Valence Analysis." In: *Proc. of the 2Nd Intl. Conf. on Affective Computing and Intelligent Interaction*. ACII '07. Lisbon, Portugal: Springer, 2007, pp. 191–202. ISBN: 978-3-540-74888-5. DOI: 10.1007/978-3-540-74889-2_18. URL: http://dx.doi.org/10.1007/978-3-540-74889-2_18 (cit. on p. 32).

[39] Hideo Shimazu. "ExpertClerk: Navigating Shoppers' Buying Process with the Combination of Asking and Proposing." In: *Proc. of the 17th Intl. Joint Conf. on AI - Volume 2*. IJCAI'01. Seattle, WA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 1443–1448. ISBN: 1-55860-812-5, 978-1-558-60812-2. URL: http://dl.acm.org/citation.cfm?id=1642194.1642287 (cit. on pp. 2, 5, 39).

[40] Elizabeth Shriberg. "Toerrrr'is human: ecology and acoustics of speech disfluencies." In: *Journal of the Intl. Phonetic Association* 31.1 (2001), pp. 153–169 (cit. on p. 8).

[41] Gabriel Skantze. "Exploring human error handling strategies: Implications for spoken dialogue systems." In: *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*. 2003 (cit. on p. 39).

[42] Cynthia A. Thompson, Mehmet H. Göker, and Pat Langley. "A Personalized System for Conversational Recommendations." In: *J. Artif. Intell. Res. (JAIR)* 21.1 (Mar. 2004), pp. 393–428. ISSN: 1076-9757. URL: http://dl.acm.org/citation.cfm?id=1622467.1622479 (cit. on p. 5).

[43] Peter D. Turney and Michael L. Littman. "Measuring Praise and Criticism: Inference of Semantic Orientation from Association." In: *ACM Trans. Inf. Syst.* 21.4 (Oct. 2003), pp. 315–346. ISSN: 1046-8188. DOI: 10.1145/944012.944013. URL: http://doi.acm.org/10.1145/944012.944013 (cit. on p. 32).

[44] Pontus Wärnestål. "User evaluation of a conversational recommender system." In: *Proc. of the 4th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*. 2005 (cit. on p. 5).

[45] Jiyong Zhang and Pearl Pu. "A comparative study of compound critique generation in conversational recommender systems." In: *Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer. 2006, pp. 234–243 (cit. on p. 3).