ReComment: Towards Critiquing-based Recommendation with Speech Interaction

Peter Grasch
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b/II
8010 Graz, Austria
peter.grasch@ist.tugraz.at

Alexander Felfernig
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b/II
8010 Graz, Austria
afelfern@ist.tugraz.at

Florian Reinfrank
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b/II
8010 Graz, Austria
freinfra@ist.tugraz.at

ABSTRACT

In contrast to search-based approaches, critiquing-based recommender systems provide a navigation-based interface where users are enabled to critique displayed recommendations as a means of preference elicitation. In this paper we present ReComment, our approach to natural language based unit critiquing. We discuss the developed prototype and present the corresponding user interface. In order to show the applicability of our concepts, we present the results of a user study. This study shows that speech interfaces have the potential to improve the perceived ease of use as well as the overall quality of recommendations.

Categories and Subject Descriptors

H.4.2 [**Information System Applications**]: Types of System – Decision support; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – Evaluation/methodology, Interaction styles, Natural language, Voice I/O

Keywords

Knowledge-based recommender systems; critiquing-based recommendation; speech interfaces; applied speech recognition

1. INTRODUCTION

Critiquing-based recommender systems are a form of conversational recommender systems that employ user-articulated critiques on the currently recommended item as a means of preference elicitation. Over the last decade these systems have attracted a significant amount of academic interest and have repeatedly shown to provide high quality recommendations while requiring relatively little effort from the user [7, 16, 18].

A multitude of different approaches to critiquing-based recommendation have been proposed over the years. Unit critiquing based recommenders use a single, directional critique of the recommended item to improve their recommendations [4]. For example, the unit critique 'less' on the attribute 'price' would cause the recommender system to search for a cheaper product [6, 13]. So-called compound critiques may be used to critique multiple attributes in the course of a single feedback cycle [14]. Different approaches of finding the optimal set of compound critiques to present to the user have been explored with the apriori method outlined in [14], and especially the application of multi-attribute utility theory described in [25] being particularly notable. Orthogonally to the used critiquing method, valuable information may also be extracted from the users evolving preferences over the course of a recommendation session by maintaining a user preference model. For example, incremental critiquing systems keep track of recent critiques in a recommendation session and try to find products that not only fit the current- but also previously articulated critiques [21]. More recently, an experience-based critiquing approach has been introduced that uses the final, accepted product of similar critiquing sessions (of different users) as the next recommendation [15]. This approach has since been improved upon with the introduction of nearest neighbor compatibility critiquing that selects the product of a comparable critiquing session that best fits the current users preference model as the next recommendation [12].

Clearly, a substantial body of research has been dedicated to help improve critiquing-based recommenders by employing intelligent prediction algorithms. The major motivation of this work is to reduce the number of interaction (critiquing) cycles needed to identify an item of relevance for the user. With the same goal but a different approach we introduce a natural language interface for unit critiquing based recommenders.

Natural language interfaces have remained largely unexplored in existing recommender systems that instead support user interactions through mouse, keyboard, and touch devices. While natural language processing remains a hard task in general, its complexity in severely limited domains is manageable [2]. Some fundamentals of applied dialog systems for recommender systems are discussed in [1] and a prototype of a natural language speech-based recommender system, the Adaptive Place Advisor, was presented in [24]. The latter proved to be promising, but user feedback was limited to providing specific attribute values which may be unsuitable for more complex domains where users are less aware of the problem space and preferences are subject to incompleteness or rapid change [22]. Recently, a speech-based natural language recommender system was mentioned as a promising future research direction in [8].

The major contributions of this paper are the following. We present ReComment[1], a speech-based unit critiquing-based recommender system that demonstrates how speech recognition can

[1] Portmanteau of "Recommend" and "Comment".

RecSys'13, October 12–16, 2013, Hong Kong, China.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2409-0/13/09 ...$15.00.
http://dx.doi.org/10.1145/2507157.2507161.

enable natural language communication with conversational recommender systems. We discuss the implementation of a prototype for the domain of digital compact cameras and report the results of an empirical study that compares the performance of ReComment with standard unit critiquing approaches. We show that our novel user interface significantly reduces the number of required critiquing cycles and demonstrate how natural language understanding can help to improve future recommender systems.

The remainder of this paper is organized as follows. In Section 2 we discuss related research to position our work in relation to the state of the art in critiquing-based recommendation. Section 3 presents the algorithms and components of the developed prototype in detail. The conducted empirical study to analyze the impact of ReComment is discussed in Section 4. The results of this study are reported in Section 5. We conclude the paper with Section 6.

2. RELATED WORK

The ReComment implementation of unit critiquing is largely based on the original ideas of the FindMe systems developed by Burke et. al. [6]. An initial product is presented and unit critiques can be articulated by the user. After each critique, a new best-matching product is suggested, completing the feedback cycle. An example critique for the digital camera domain would be "More optical zoom", introducing a 'smaller than the value of the currently shown product'-constraint on the optical zoom attribute [6, 21].

As mentioned earlier, compound critiques are often used in critiquing recommender systems to reduce the number of required interaction cycles [14]. Compound critiques impose multiple constraints in a single feedback cycle (e.g., "More optical zoom but less weight"). The inclusion of such critiques in ReComment was considered but they were ultimately omitted because their selection criteria reveals implicit information of the problem domain that is hard to represent equivalently in both interfaces, thus possibly skewing the results of the study. For example, showing the compound critique "Larger sensor size but more expensive" as an option in the traditional interface (control group) not only provides the user with another critiquing option but also exposes a correlation between sensor size and price that might otherwise be unknown to the user [20]. However, mentioning the same proposed critique (as text) in the speech-based interface would certainly guide the user's choice of words not just in this critiquing cycle but from then on, because the sentence also contains the information that "larger sensor size" and "less weight" are understood commands. In an effort to make the different interfaces more comparable, support for compound critiques was therefore not included[2].

A user preference model consisting of recently added critiques is tracked across several feedback cycles in a manner very similar to the incremental critiquing approach presented in [21]. However, ReComment does not permanently remove recommended products from the search space, but instead temporarily biases the recommender against them to avoid the problem of diminishing choices [17].

ReComment uses the current sales rank of a popular online retailer to add a prior probability to the products in its database. In a sense this could be compared to experience-based critiquing as described in [15] in that it also harnesses information from other users' buying decisions. However, instead of detailed critiquing information, only the accepted products influence current recommendations.

ReComment starts directly with the most popular product in the domain instead of first requiring an initial query (see, e.g., [6]). To ensure quick adaption to a user's preferences, the traditional unit critiquing recommendation algorithm was relaxed to not enforce similarity with the current recommendation. This property is however naturally re-introduced through the maintained user preference model as returned products will become more and more similar once the user's requirements have been sufficiently determined to codify underlying intentions. Additionally, a special utility function discourages large jumps through the search space for critiqued attributes (see Section 3.1.3).

Finally, it is interesting to note that Burke et. al. already presented natural language queries in their seminal paper on the FindMe systems [6] as examples of how a critique might be formulated. On the subject of renting a movie, the authors refer to a possible user thinking "That would be good, but it's too violent for my kids." leading to a new critique on the "level of violence" attribute. Had the user in their example actually articulated this thought, a system like ReComment could have extracted the contained critiquing information automatically.

3. SYSTEM DESCRIPTION

This section outlines the developed ReComment prototype.

3.1 Recommender

ReComment is based on a unit critiquing algorithm that exploits iteratively added, user-defined critiques. The final recommendation in each step is ensured to fulfill at least the last given critique. From this subspace, the item with the overall highest utility rating in regards to the user preference model is selected (see Subsection 3.1.1 for more information about the maintained user preference model and Subsection 3.1.3 for details about the calculation of the utility value of a single critique). Unfulfillable critiques are rejected. For example, if the user requests a less expensive product than the cheapest one, a warning message will be shown briefly, telling the user that there are no such products. This basic algorithm is outlined in Algorithm 1.

```
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 1: Selecting the next recommendation.

3.1.1 User preference model

Over the course of a recommendation session, a user preference model is maintained. It contains the 15 most recent critiques, linearly weighted based on their age.

[2] The speech-based input accepts more than one condition during a feedback cycle but it treats them as a collection of unit critiques.

Each new critique is checked against the preference model. Critiques that contradict or refine existing constraints replace their older equivalents [21]. If a feedback cycle is undone by the user (by selecting the "Back" button or giving an equivalent voice command, depending on the interface; see Section 3.3), any newly added constraints from this cycle are removed and any replaced critiques are restored, essentially restoring the system state before the last given critique.

In order to curb the problem of unreachable products that are "overshadowed" by numerically better matching but potentially less appealing ones, each feedback cycle also adds a $product \neq current$ critique as suggested in [16]. Because this constraint is subject to the same aging process and eventual removal, ReComment does not suffer from diminishing choices [17].

3.1.2 *Prior probability*

The 100 most frequently sold compact digital cameras were retrieved from the popular online retailer Amazon[3] and their sales rank was added to the domain database to act as a prior probability for the selection process. Because of this additional information, the first suggested product presented to a new user is the most popular product of the domain based on recent sales.

3.1.3 *Rate of change*

In order to be able to represent critiques such as "much cheaper" or "a little bit cheaper", the base utility expressed by each critiqued numerical attribute is not just a binary measure ("cheaper") but rather represents the *distance* of an implicitly expressed goal. The measure of distance between two numerical values is shown in Formula 1 (for non-numerical attributes the distance is either 0, for equal values, or 1 otherwise). The algorithm to calculate the utility value from this calculated distance is outlined in Algorithm 2.

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

The utility function has been designed such that it tries to avoid large jumps in the product space based on a single critique and instead tries to stay within a certain acceptance region from the last shown product. A piecewise function was selected that has the following properties. If the critique is violated, the utility value will be negative, otherwise positive. Attribute values that are very different or too close to the current recommendation are assigned lower utility values than products closer to the deduced implicit goal. Given symmetric relations and both lower and upper bounds specified for one attribute, traditional, linear (triangular) acceptance functions would theoretically create an optimal region between those bounds, therefore having the combined utility function only depend on the age (weight) of the individual critiques. In contrast, the non-linear approach outlined in Formula 1 provides what we believe to be a better representation of the user's goal (see Figure 2 for an example). The implicit goal can be influenced with what we call a "modifier factor". The modifier factor is extracted from adjectives to the critique (e.g., "much cheaper") and defaults to 1.0, representing a desired 50 % change of the attribute value[4]. A selection of sample adjectives influencing the modifier factor can be found in Table 1.

For example, a critique of "larger than 50" for a given attribute would result in ReComment trying to find a product with an attribute value of around 100 (modifier factor: 1.0; refer to Figure 1 for a plot of the utility function). A subsequent critique of "smaller" (from the new position of 100) would result in the recommender trying to find a product with a value of around 75[5] for the given attribute (refer to Figure 2 for a plot of the utility function of a combination of both critiques). Had the second critique instead been "a little bit smaller", the expressed implicit goal would have instead been a value of around 95 (see Figure 2).

Sample adjectives	Modifier factor	Deduced desired change
"slightly"	0.2	10 %
"very"	2.0	100 %
"not"	-1.0	-50 %

Table 1: Sample adjectives that affect the modifier factor.

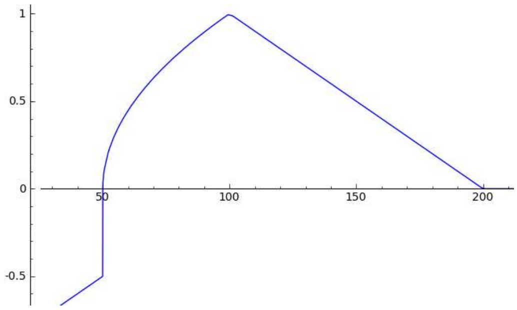


Figure 1: Utility function of the critique $x > 50$.

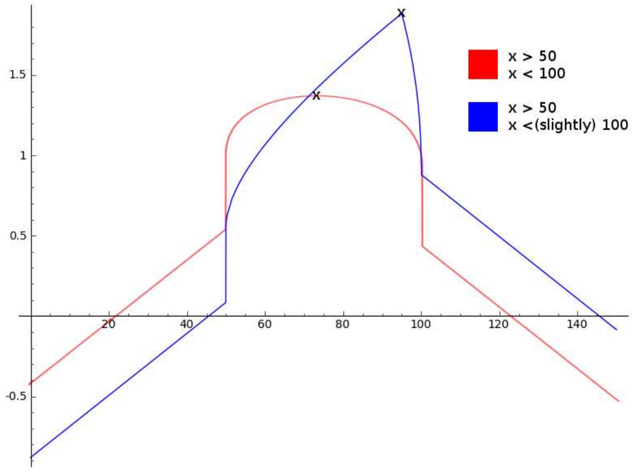


Figure 2: Utility function of the subsequent critiques $x > 50$ and $x < 100$ as well as $x > 50$ and $x <_{slightly} 100$.

Traditional interfaces, including the mouse-based interface developed for ReComment, have not been expressive enough to capture such nuances in a user's critiquing input. To the best of the authors' knowledge, this feature of the speech-based interface is therefore a novel approach for critiquing recommender systems and presents one of the main advantages of natural language interfaces for recommender systems.

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

[4] In order to make "smaller than" and "larger than" relationships symmetric, the percentage is always calculated from the larger value to the smaller value.

[5] In practice, this value would be around 73 as the older critique would have already aged once and thus would have reduced influence on the overall utility of the product.

```
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 2: Schematic utility calculation.

3.2 Speech recognition

Any speech-based natural language interface needs to address a number of challenges that arise when processing spontaneous speech: Speech disfluencies like filler words, repetitions, and false starts (self-interruption) are common [23]. Additionally, dialectal speech, emotive speech patterns and even occasional laughs or angry snorts exacerbate the situation for speech models that are usually trained on manually cleaned utterances articulated in a neutral, sometimes almost clinical tone.

A sufficiently well performing speech recognizer is an essential component of any successful speech-based user interface. Therefore, a custom, domain specific solution for Austrian German[6] and the domain of digital cameras was developed using the Simon[7] system and the CMU SPHINX speech recognition framework[8].

3.2.1 Speech model

The speech model represents the target language and can be broken into the following parts. (1) the dictionary, containing the set of recognized words as well as their phonetic transcriptions; (2) a (context-infused) word probability model (N-Gram) that forms the language model (LM) and (3) a learned representation of how the individual phones that make up the spoken words are pronounced, stored in the acoustic model (AM).

λ	**Base corpus**
0.5	Critiquing sentence fragments
0.4	Sentence fragments for explicit value elicitation
0.1	Standard, written German

Table 2: Selected LM mixtures.

For the purpose of our digital camera project, a custom 3-gram language model was developed based on the accepted sentence fragments of the parser (refer to Section 3.2.2). The training corpus itself was split into three parts. (1) the basic sentence fragments expected during critiquing (“cheaper”, “a bit smaller”, etc.), (2) a training corpus for explicit value elicitation (“more than 5 megapixel”, etc.) and (3) the base corpus modeling standard German[9]. Separate 3-grams of these three components were built and mixed into a final language model heavily focusing on expected, domain specific sentence fragments. Based on experimentation on recordings from pilot testers of early prototypes, the mixture weights shown in Table 2 were selected. These mixture weights (lambdas) govern the influence of the individual N-Grams on the resulting language model. A higher value of λ represents more influence ($\sum \lambda_i = 1$).

An existing acoustic model built from various data sources such as Voxforge[10], LibriVox[11] and supplemental data from professional speakers from the spoken language corpus of the ADABA database[12] was adapted with a small speech corpus recorded from the interactions of pilot testers with early prototypes using the maximum a posteriori probability speaker adaption [10].

3.2.2 Parser

To interpret the output of the speech recognition system, a custom parser was developed. It identifies attributes (mapping to properties of the products of the problem space), pre- and post-binding relationships (which may be conditioned on certain attribute types), as well as modifier factors (“a bit”, “not”) and commands (“cheaper”).

The system supports regular expressions with optionally provided arguments. For example, the post-binding relationship “more than” and the attribute “(\d+) Euro”[13] matches the recognition result “more than 100 Euro” and creates a corresponding critique. Commands without explicit values (“more expensive”) are interpreted to relate to the currently shown product.

Default relationship targets enable ReComment to understand “larger” as referring to object size while also interpreting “larger sensor size” correctly.

Compound critiques (e.g., “Can you show me a smaller Canon?”) are broken into their individual relationships and added as separate, equally prioritized unit critiques. ReComment guarantees that at least one of these constraints will be fulfilled by the next recommended product but will of course prefer products that are closer to all the constraints in the current user model with the just added constraints naturally receiving the highest priority (see Section 3.1.1).

The language description itself is stored in an easily extendable XML file. The final NLP rules used for the empirical study recognize more than 300 different command fragments which can be combined to form millions of different sentences.

3.3 User interface

Simple, unit critiquing-based user interfaces were developed for both the mouse- and the speech-based interaction methods.

Instead of an initial query, the user starts immediately with the “best matching” product. In lieu of critiquing information, the prior probability dominates the selection process resulting in the most sold product being shown initially (see Subsection 3.1.2).

Both interfaces use auditory cues to inform the user of any of the following three events: ‘new recommendation generated’, ‘failed to find a matching product’ and ‘stopped recording’ (after the user releases the PTT switch; only triggered in the speech-based interface). The first two mark both possible outcomes of the end of a critiquing cycle.

[6] Example sentences used in this paper are translated from German.

[7] http://simon.kde.org

[8] http://cmusphinx.sourceforge.net/

[9] Created from a recent dump of the German Wikipedia.

[10] http://voxforge.org

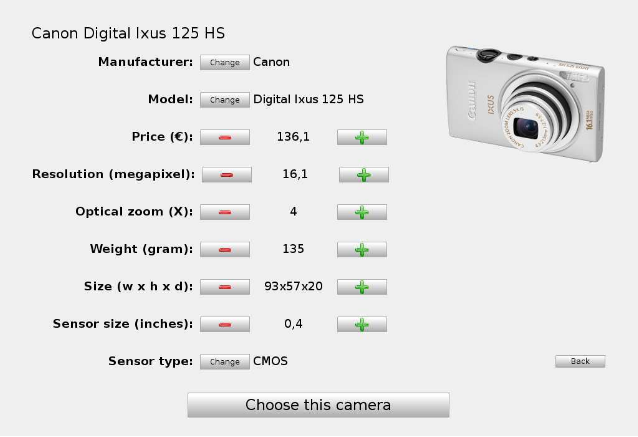
[11] http://librivox.org

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

[13] In Perl Regular expressions, “\d+” match one or more digits whereas the surrounding parenthesis mark the subexpression that ReComment will treat as the extracted attribute value.

3.4 Mouse-based user interface

The traditional, mouse-based user interface shown in Figure 3[14] is very similar to other unit critiquing interfaces such as the FindMe systems and, except for the lack of compound critiques, especially the Qwikshop system [5, 21].



Canon Digital Ixus 125 HS

Manufacturer: Canon

Model: Digital Ixus 125 HS

Price (€): 136,1

Resolution (megapixel): 16,1

Optical zoom (X): 4

Weight (gram): 135

Size (w x h x d): 93x57x20

Sensor size (inches): 0,4

Sensor type: CMOS

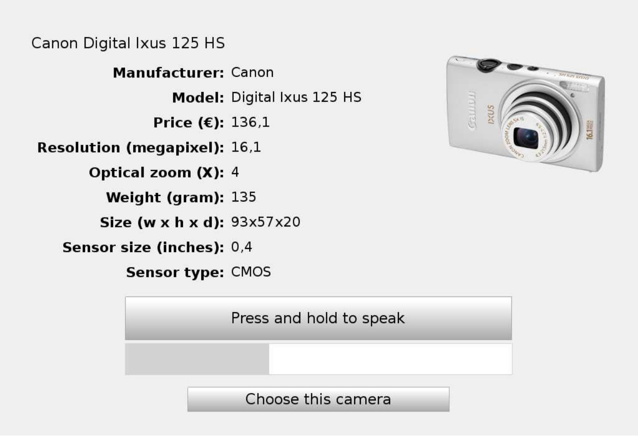
Figure 3: RECOMMENT: Mouse-based user interface.

Next to the unit critiquing controls themselves, the interface also provides a "Back" button to remove critiques from the user preference model. This was added to encourage users to browse for even better matching products after finding an acceptable one without fearing to be unable to return to the current recommendation.

3.5 Speech-based user interface

The speech-based interface is depicted in Figure 4[14].

A simple, energy-based automatic voice activity detection was implemented, but was ultimately deactivated in an effort to make users' speech more focused on the task at hand and to avoid possible accidental recognition results brought on by, e.g., talking with the person conducting the study. Instead, RECOMMENT uses a "push to talk" (PTT) methodology similar to a walkie-talkie, requiring users to press and hold a button while giving voice commands.



Canon Digital Ixus 125 HS

Manufacturer: Canon

Model: Digital Ixus 125 HS

Price (€): 136,1

Resolution (megapixel): 16,1

Optical zoom (X): 4

Weight (gram): 135

Size (w x h x d): 93x57x20

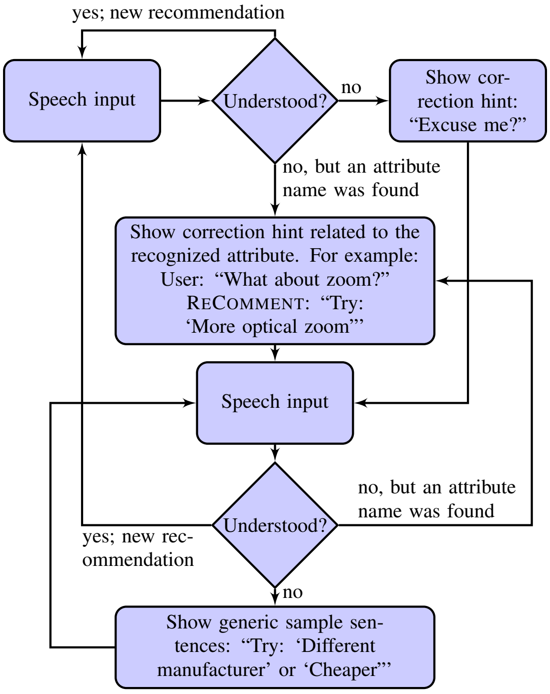
Sensor size (inches): 0,4

Sensor type: CMOS

Figure 4: RECOMMENT: Speech-based user interface.

The measured noise level is shown in the user interface at all times as a moving bar under the PTT button (VU meter). This was included to instill a sense of an actively listening system, especially for users who had not used speech-based interfaces before.

Because study participants were not informed about what kind of sentences would be interpreted correctly (see Section 4), the system instead showed correction hints as needed. These sparse instructions were displayed only in case of multiple successive recognition errors or when a sentence was already partially recognized (in which case the instructions referred specifically to this partial match). A state diagram of this process, including some examples, is shown in Figure 5. Correction hints, just like correct recognition results, are shown inside the VU meter control element discussed above and automatically hidden after 3.5 seconds. Again, these measures were taken to encourage free articulation by minimizing guidance as much as possible without compromising learnability.



yes; new recommendation
Speech input
Understood?
no
Show correction hint: "Excuse me?"
no, but an attribute name was found
Show correction hint related to the recognized attribute. For example: User: "What about zoom?" RECOMMENT: "Try: 'More optical zoom'"
Speech input
no, but an attribute name was found
Understood?
yes; new recommendation
no
Show generic sample sentences: "Try: 'Different manufacturer' or 'Cheaper'"

Figure 5: Guiding user input.

4. STUDY DESIGN

To examine the hypothesis that speech-based recommender systems can outperform traditional interfaces, we conducted an empirical study.

4.1 Test demography

The test group, consisting primarily of undergraduate and postgraduate students, was split into two groups, each consisting of 40 users for a total of 80 participants. One group, henceforth referred to as group A, used the speech-controlled interface whereas the other group, group B, used the mouse-based interface.

A more detailed breakdown of both groups can be found in Table 3.

4.2 Task definition

Each participant was instructed to imagine buying a new digital compact camera. To this end, users were told to try to think about

[14] The original interface is in German, the screenshots show a translated version.

Characteristics	Group A	Group B
Male	35	33
Female	5	7
Total	40	40
Median age	24	22
Personally own camera	67.5 %	60 %
Seeked help when buying this camera	19.4 %	26.7 %

Table 3: Demography of the test groups of the empirical study.

a camera that best fits their personal needs and use RECOMMENT to find such a product. Participants were asked to ignore product dimensions that were not included in the recommendation interface (e.g. display size, shutter speed, etc.).

Both groups received purposefully minimal instructions regarding the actual user interface. The group using the mouse-based interface was simply told to 'use the buttons' to find products that matched their requirements whereas the instructions for the speech-based interface were limited to an equivalent sentence and a small note explaining the PTT system (see Section 3.3).
Specifically, the group testing the developed natural language interface did *not* receive any kind of tutorial or list of supported voice commands but was instead told to 'try it out' in an effort to encourage users to use their own words when interacting with the system.

4.3 Evaluation

Direct feedback was collected with a questionnaire that was filled out by each participant immediately after completion of the assigned task. This questionnaire included questions about the participant's knowledge of the domain, the quality of the last given recommendation, questions about the general usability as well as questions designed to gauge the user's personal impression of the used interaction method.

The standard usability survey scale presented in [3] was used but had to be slightly adapted to better fit the tested system. Specifically, questions related to the "various functions" or the "inconsistency" potentially introduced by a rich feature set were not applicable to RECOMMENT and were thus removed. Additionally, the question about the user "frequently using" the presented system was reworded to instead ask if the user could see themselves using the system when purchasing his or her next digital compact camera. In order to maintain the same scale of 0 to 100 as the original SUS, the final multiplier of the calculation formula was increased, giving the remaining questions more, equally distributed, weight.

4.4 Case Base

To evaluate RECOMMENT, a database of currently available, compact digital cameras was assembled. Product data, images, and recent pricing information for a total of more than 600 distinct models was collected from various online retailers and compiled into a single database.

The following feature set was collected: Model*, Manufacturer*, Price (€)*, Resolution (Megapixel)*, Sensor size (inches)*, Sensor type*, Size (w×h×d)*, Weight (gram)*, Internal memory (megabyte), Digital zoom (times), Optical zoom (times)*, External storage. Features marked with a * were selected for inclusion in the final interface after receiving feedback from early pilot testers. The 100 most popular cameras were additionally tagged with their current product sales rank (see 3.1.2).

5. RESULTS

In this section we will present the results obtained from the conducted empirical study.

5.1 Recognition performance

An adequately performing speech recognition component is a necessary prerequisite for the potential success of any speech-based user interface. To determine the recognizers' performance we will look at both the perceived and the actual recognition accuracy. To judge participants' impressions, they were asked to rate the performance of the speech recognition as part of the questionnaire. The generally very positive result can be seen in Figure 6[15].

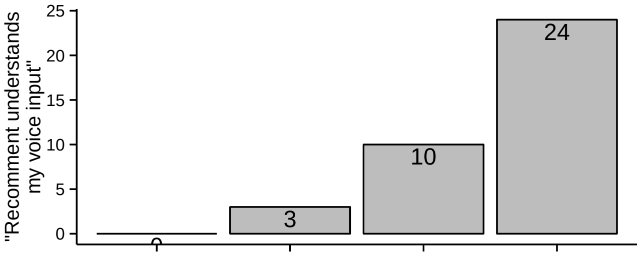


Figure 6: Participants' perception of the speech-recognition accuracy ($[1, 4]$, higher is better).

For the purposes of our evaluation, all recorded dialogs were manually transcribed after the completion of the study. To accurately evaluate the acoustic model and the recognizer itself, we compared the resulting (parsed) critiques (from the recognition results) with the critiques extracted from the manually transcribed reference set. In this context, sentences such as "cheaper, please" and, e.g., "cheaper, pepper" are therefore treated as equal because their difference does not impact RECOMMENT's performance. Sentences consisting of more than one command or including modifying adjectives (see Section 3.1.3) where the system only captured part of the intention, are treated as partial errors. To establish a frame of reference, Google's speech recognition web service[16] was used to transcribe the samples recorded during the course of the study. The results of the comparison of both recognition systems can be seen in Figure 7. As can be seen, our custom recognition component based on Simon and PocketSPHINX manages to significantly outperform the off-the-shelf online service for our limited domain and proved to be sufficiently accurate for the purposes of this study.

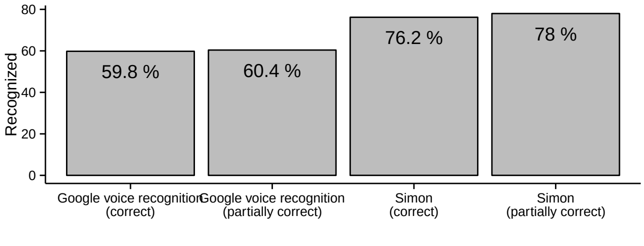


Figure 7: Comparison of the developed recognizer in comparison with a state of the art, off-the-shelf online service.

[15] The results were verified with a counter question. Questionnaires where either both questions were answered positively or both negatively have been removed before interpreting the results

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

The natural language parser was evaluated by manually reviewing spoken commands of utterances and comparing them with the system's interpretation. Out of 384 sentences collected as part of the study, only 21 sentences (5.5 %) failed to parse correctly (in part or completely). Most of these errors were due to unexpected constructs like "7 optical zoom" with notable exceptions being semantically complex commands like "credit card size" or "highest resolution for 120 Euro" or commands referencing earlier critiques like "even more". The majority of sentences referred to a single attribute, i.e., unit critiques were the predominant interaction mode (used in 85.7% of the cases) whereas compound critiques were rarely used. 74 sentences included references to explicit values and 12 used modifier factors (see Section 3.1.3). Table 4 shows an overview of the recognized sentences and Table 5 shows the beginning of a sample recommendation session.

Category	**Count**
Discarded	49 (12.8%)
Unit critique	329 (85.7%)
Compound critique (2 attributes)	3 (0.8%)
Compound critique (3 attributes)	2 (0.5%)
Compound critique (5 attributes)	1 (0.3%)

Table 4: Types of used commands.

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 5: Sample user interactions with RECOMMENT.

5.2 Usability

To determine the usability of our created interfaces, we included a slightly modified version of the standard usability scale in the questionnaire (see Section 4.3). We compared the resulting SUS scores of the participants using the mouse-based interface (group B) with those using the speech-based interface (group A). Additionally, for comparison purposes, we also present the reported usability of those 24 users that reported that the recognizer understood them well as a separate group. All three results are shown in Figure 8.

While the increase in perceived usability between group A and B is not entirely conclusive (p = 0.13), the users that reported to be well understood by RECOMMENT clearly preferred the speech-based interface over the traditional one (p < 0.02). Moreover, participants of group A answered more favorably 'yes' on the question asking if they would use RECOMMENT before next purchasing a digital camera than the control group, suggesting increased user satisfaction (mean of group A: 3.1, group B: 2.7; p < 0.5).

5.3 Recommendation quality

Despite using the same basic recommender algorithm, the reported recommendation quality and efficiency differed substantially. Participants using the speech-based user interface reported to like the last presented product better than those using the traditional user interface (p < 0.05). A more detailed breakdown of their answer to the related question on the questionnaire can be found in Figure 9.

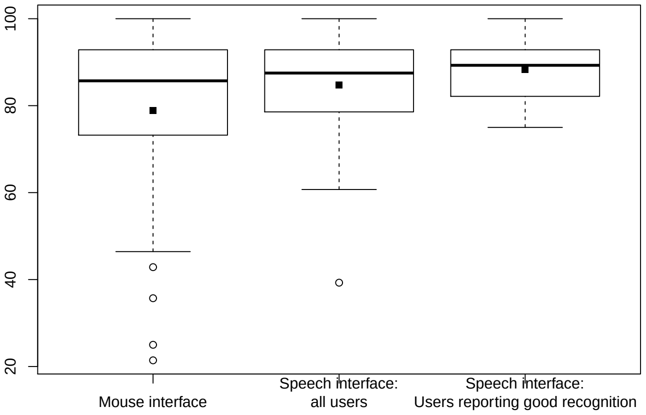


Figure 8: Usability evaluation (adapted SUS scores).

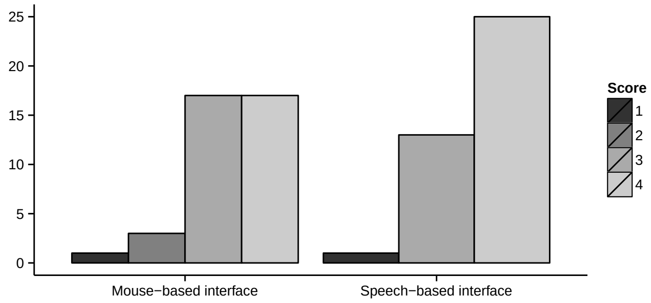


Figure 9: User score of last recommended item ($[1, 4]$, higher is better).

Furthermore, users of the speech-based interface used considerably fewer interaction cycles before arriving at the desired product (see Figure 10). However, although the sessions were not timed, it is worth mentioning that this may not directly translate into significantly shorter overall session length (seconds) because articulating preferences by voice is obviously more time consuming than clicking a button.

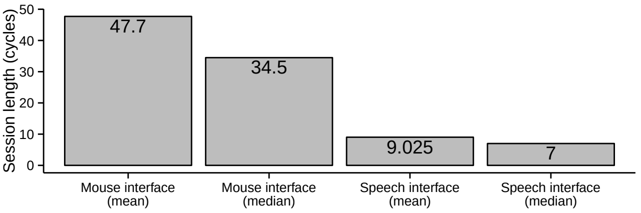


Figure 10: Session length (lower is better).

6. CONCLUSION AND FURTHER WORK

In this paper we presented RECOMMENT, a new approach to critiquing-based recommender systems using a speech-controlled natural language interface. We compared the speech-based interface with a traditional, mouse-based user interface in an empirical study. The results show that users of the speech-based interface found overall more satisfactory products, required less interaction cycles and were more likely to want to use the system again.

Without further research, it is hard to draw definitive conclusions as to why users reliably reached better fitting products with the speech-based interface, but it may be argued that users spent less time exploring the product space and instead concentrated on their actual needs, likely because of the increased effort required to formulate a voice command. This and other theories should be further explored in future work.

The speech recognition and natural language parsing components used in ReComment, while sufficient for basic functionality, certainly leave much room for improvement, including support for other languages than German. The surprisingly low amount of compound critiques in natural interaction should also be investigated further. Monitored user sessions with a voice controlled, compound critiquing system could also yield valuable insight into hidden correlation between attributes by detecting patterns of frequently co-occurring attributes in user input.

Moreover, future works should also compare and contrast the performance of speech-based interfaces for alternate recommender approaches (e.g., constraint-based recommenders [11]).

Some users also had trouble to adjust to talking to what essentially still looked like a traditional computer program. Integration of speech output and an avatar to talk to has been shown to lower this entry barrier [19]. Moreover, human conversational speech also includes other, subtle emotional cues that can be extracted by an appropriate system [9]. Future systems could integrate this additional information to influence (e.g., the weight of) the deduced constraints to potentially improve recommendation quality even further.

7. ACKNOWLEDGMENTS

The work presented in this paper has been partially funded by the Austrian Research Promotion Agency (Casa Vecchia, 825889)

8. REFERENCES

[1] D. 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*, pages 9–22, 2002.

[2] E. Brill and R. J. Mooney. An overview of empirical natural language processing. *AI magazine*, 18(4):13, 1997.

[3] J. Brooke. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189:194, 1996.

[4] R. Burke. Knowledge-based recommender systems. In *Encyclopedia of Library and Information Systems*. Marcel Dekker, 2000.

[5] R. D. Burke, K. J. Hammond, and B. Yound. The findme approach to assisted browsing. *IEEE Expert*, 12(4):32–40, 1997.

[6] R. D. Burke, K. J. Hammond, and B. C. Young. Knowledge-based navigation of complex information spaces. In *Proceedings of the national conference on artificial intelligence*, volume 462, page 468, 1996.

[7] L. Chen and P. Pu. Evaluating critiquing-based recommender agents. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 157. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[8] L. Chen and P. Pu. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction*, 22(1-2):125–150, 2012.

[9] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor. Emotion recognition in human-computer interaction. *Signal Processing Magazine, IEEE*, 18(1):32–80, 2001.

[10] S.-J. Doh. *Enhancements to transformation-based speaker adaptation: principal component and inter-class maximum likelihood linear regression*. PhD thesis, Carnegie Mellon University, 2000.

[11] A. Felfernig and R. Burke. Constraint-based recommender systems: technologies and research issues. In *Proceedings of the 10th international conference on Electronic commerce*, page 3. ACM, 2008.

[12] M. Mandl and A. Felfernig. Improving the performance of unit critiquing. In *User Modeling, Adaptation, and Personalization*, pages 176–187. Springer, 2012.

[13] K. McCarthy, L. McGinty, and B. Smyth. Dynamic critiquing: An analysis of cognitive load. In *Proceedings of the 16th Irish Conference on Artificial Intelligence and Cognitive Science*, pages 19–28, 2005.

[14] K. McCarthy, J. Reilly, L. McGinty, and B. Smyth. On the dynamic generation of compound critiques in conversational recommender systems. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 176–184. Springer, 2004.

[15] K. McCarthy, Y. Salem, and B. Smyth. Experience-based critiquing: reusing critiquing experiences to improve conversational recommendation. In *Case-Based Reasoning. Research and Development*, pages 480–494. Springer, 2010.

[16] L. McGinty and J. Reilly. On the evolution of critiquing recommenders. In *Recommender Systems Handbook*, pages 419–453. Springer, 2011.

[17] D. McSherry and D. W. Aha. The ins and outs of critiquing. In *IJCAI*, pages 962–967, 2007.

[18] P. H. Z. Pu and P. Kumar. Evaluating example-based search tools. In *Proceedings of the 5th ACM conference on Electronic commerce*, pages 208–217. ACM, 2004.

[19] L. Qiu and I. 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. *ACM Trans. Comput.-Hum. Interact.*, 12(4):329–355, 2005.

[20] J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Explaining compound critiques. *Artificial Intelligence Review*, 24(2):199–220, 2005.

[21] J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Incremental critiquing. *Knowledge-Based Systems*, 18(4):143–151, 2005.

[22] J. Reilly, J. Zhang, L. McGinty, P. Pu, and B. Smyth. Evaluating compound critiquing recommenders: a real-user study. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 114–123. ACM, 2007.

[23] E. Shriberg. Toerrrr'is human: ecology and acoustics of speech disfluencies. *Journal of the International Phonetic Association*, 31(1):153–169, 2001.

[24] C. A. Thompson, M. H. Goeker, and P. Langley. A personalized system for conversational recommendations. *J. Artif. Intell. Res. (JAIR)*, 21:393–428, 2004.

[25] J. Zhang and P. Pu. A comparative study of compound critique generation in conversational recommender systems. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 234–243. Springer, 2006.