Research article

Peter Grasch*, Alexander Felfernig

On the Importance of Subtext in Recommender Systems

Eliciting Nuanced Preferences Using a Speech-based Conversational Interface

Abstract: Conversational recommender systems have been shown capable of allowing users to navigate even complex and unknown application domains effectively. However, optimizing preference elicitation remains a largely unsolved problem. In this paper we introduce SpeechRec, a speech-enabled, knowledge-based recommender system, that engages the user in a natural-language dialog, identifying not only purely factual constraints from the users' input, but also integrating nuanced lexical qualifiers and paralinguistic information into the recommendation strategy. In order to assess the viability of this concept, we present the results of an empirical study where we compare SpeechRec to a traditional knowledge-based recommender system and show how incorporating more granular user preferences in the recommendation strategy can increase recommendation quality, while reducing median session length by 46 %.

Keywords: Knowledge-based Recommender Systems; Speech Interfaces; Applied Speech Recognition; Emotive User Interface; Sentiment Analysis; Paralanguage
CCS: Information System Applications: Types of System – Decision support; Information Interfaces and Presentation: User Interfaces – Evaluation/methodology, Interaction styles, Natural language, Voice I/O

DOI 10.1515/icom-2015-0017

1 Introduction

With the advent of ever larger collections of information, navigating these databases has become a major challenge. Recommender systems help users by eliciting, processing and acting on their preferences and requirements, thus allowing efficient search and navigation in even large and complex domains that may be unknown to the user.

Over the years a multitude of approaches have been proposed, many of which have proven viable and found footing in different domains for varying reasons. In general, we can group most recommender systems in three categories. The most well known and common class of recommender systems, collaborative- or social-filtering systems, base their recommendations on the activity of other, similar users. Content-based systems focus instead on product similarity and use the user's own interaction history to find items similar to previously accepted ones. In contrast, the third major approach and main focus of this paper, so-called knowledge-based recommender systems, employ domain knowledge and an explicit, usually comparatively detailed user model to make informed recommendations based on the user's actual requirements [4, 14, 23].

Compared to content-based and collaborative approaches, knowledge-based recommender systems are less affected by ramp up issues for new products and users, for whom there is no interaction history to draw on. Their more intelligible recommendation process further enables application in high-involvement domains, such as the recommendation of financial services, where trust in the system is crucial [4, 13].

However, in order to be effective, knowledge based systems depend on a product database that often requires laborious knowledge engineering and a fairly extensive model of an individual user's preferences, which commonly requires direct user involvement to build. Depending on the complexity of the domain, this time investment may be enough of a deterrent for casual users, limiting the applicability of knowledge-based systems, especially in high-paced environments with ephemeral user sessions such as e-commerce. As a result, optimizing preference elicitation has become a major concern and attracted substantial research interest [3, 6].

***Corresponding author: Peter Grasch,** Graz University of Technology, Institute for Software Technology, Graz, Austria, e-mail: peter@grasch.net
Alexander Felfernig: Graz University of Technology, Institute for Software Technology, Graz, Austria

Next to asking users to express their preferences by answering questions posed by the system ("navigation by asking", or "search"), example based critiquing has emerged as a primary means of eliciting user feedback [18, 24, 31]. Critiquing recommenders enable navigation by requesting improvements over a currently proposed product. Such introduced constraints are commonly called "critiques". For example, in an e-commerce system the critique "cheaper" may introduce a new constraint (critique) to the user model, representing price < x, where x is the price of the currently shown item [5, 6]. Critiquing has repeatedly shown to be an effective method of navigating even large problem sets while requiring comparatively low cognitive load. However, as critiques tend to introduce relatively little new information to the user model, excessively long user sessions remain an issue [7, 17, 24].

This general tradeoff between minimizing the required effort for giving feedback versus maximizing the utility of the received feedback seems to exist not only in critiquing-based systems, yet has rarely been openly acknowledged. While part of the problem is certainly due to the most effective type of feedback often involving decisive compromises, it seems reasonable to assume that more complex, and therefore more useful types of feedback are also generally harder to synthesize in current recommender systems interfaces'. We argue, that a recommender system that makes it trivial to specify arbitrarily complex preferences and critiques in a natural way would therefore allow users to articulate their true (hidden) preferences more directly and more effectively.

In this paper we introduce SpeechRec, a speech-based, conversational knowledge-based recommender system. We discuss, how natural language input can enable users to better specify their true preferences, and show how this new interaction paradigm can be used to extract significantly richer feedback from users' input without demanding extra effort. To assess the viability of our approach, we present the results of an empirical study. We show that our novel user interface slightly increases recommendation performance, while significantly decreasing session length over both the reduced and the baseline system.

The remainder of this paper is organized as follows. In Section 2 we position our work in relation to the state of the art of the field. Section 3 outlines the algorithms and components of the developed prototype in detail. In Section 4 we present the empirical study, the result of which are reported in Section 5. We conclude the paper with Section 6.

2 Related Work

Natural language recommender systems have been researched before, although with often limited implementations. In [31], Shimazu presents the recommender system "ExpertClerk", that uses written natural language interaction to ask domain questions until the product set has been sufficiently constrained, at which point three sample products are proposed and the user can navigate further by critiquing them. Text-based natural language recommender systems were further researched by Wärnestål in [35], with just as promising results. One of the earliest speech-based conversational recommender systems was introduced by Thompson et al. in [34]. However, their Adaptive Place Advisor only supported navigation by specifying concrete values of product attributes, significantly reducing the usefulness of the natural language input. To the best of the authors knowledge the first attempt at supporting unrestricted spoken natural language input was introduced by Grasch's et al. ReComment in [16]. Their speech-based critiquing recommender system outperformed a traditional user interface by both reducing session length and increasing recommendation quality. However, the selected application domain, digital compact cameras, was arguably "easy" and expected user requirements, and therefore expected user input, was relatively predictable.

To handle the kind of diverse user input we expect for unrestricted natural language interaction for our selected domain of laptops, SpeechRec would need to support more complex forms of user input. A pilot study was used to ascertain what kind of statements users would typically use. The simplest form of feedback, explicitly stated attribute values, are parsed and codified similarly to traditional constraint-based recommender systems [12]. Statements relative to the currently displayed products are treated as unit critiques [4]. General statements about the user's preferences (e. g., "I need a laptop for University.") are treated as compound constraints on the attributes they affect. Some statements, however, referred to data that usually is difficult to collect such as "I am looking for a laptop that stays cool under load" or "I want a notebook that looks good". In an effort to support these kinds of requests, we turned to product reviews and augmented our product database with extracted user sentiment similarly to the process outlined by Moghaddam et al. in [21]. A recommender system utilizing customer sentiment extracted from product reviews has previously been introduced by Dong et al. in [9], but whereas their system uses sentiment differences as their main means of navigation, in SpeechRec user sentiment is only part of the recommendation strategy.

3 System Description

The following section describes the developed prototype for the domain of consumer laptops.

3.1 Case Base

A well designed product database is a core component of any knowledge-based recommender. For our prototype, we collected information on 632 laptops currently on the market.

Each product is tagged with information on 40 attributes, ranging from basic information like screen size to details such as the type of panel used. Every product has at least one photo attached.

To extract user sentiment, we collected a total of 3246 user reviews from a popular online retailer[1].

A manually compiled list of around 100 product aspects was expanded with similar words based on a word clustering of the extracted reviews using the algorithm presented in [8]. The resulting list was pruned to arrive at a total of 304 keywords mapped to 41 distinct aspects, which were hierarchically grouped such that reviews commending, for example, a laptop's viewing angles would not just increase the product's "Viewing Angles" aspect's sentiment score but also the score of its more general "Display" aspect.

To identify the polarity of a statement referring to a given aspect, we follow a lexical approach, assigning words or short phrases either positive or negative sentiment, similar to what Andreevskaia et al. outlined in [1]. 40 positive and 40 negative seed phrases are used with a fixed polarity of 1 and -1, respectively. Starting at these seeds, we recursively navigated their synsets using the German OpenThesaurus web service, reducing the assigned polarity of encountered phrases proportionally to the distance from their initial seeds [20, 22]. Employing this method with a maximum seed-to-leaf distance of 2, we created a polarity dictionary for Austrian German laptop reviews consisting of 4628 polarity infused words and phrases.

Our approach of extracting sentiment information from reviews is roughly comparable to Shakih's et al. approach in [30]. For each review, we parse every sentence using ParZu, the Zurich Dependency Parser for German [28, 29], and identify aspects and polarity laden phrases in the parse tree. Each polarity inducing node is associated with its closest aspect, where scores are summed to calculate the aspect's final polarity. To avoid noise, only polarity of aspects that were discussed in at least two reviews of the product are included in the database.

3.2 Spoken Language Input

Unrestricted, conversational spoken language poses a significant challenge to the input layer.

SpeechRec uses a version of the Simon[2] speech recognition system that was modified to include the arousal score calculated by openEAR as described in Section 3.2.2. Simon in turn uses the PocketSphinx decoder of the CMU SPHINX speech recognition framework[3]. A schematic overview of SpeechRec's speech processing architecture is provided in Figure 1.

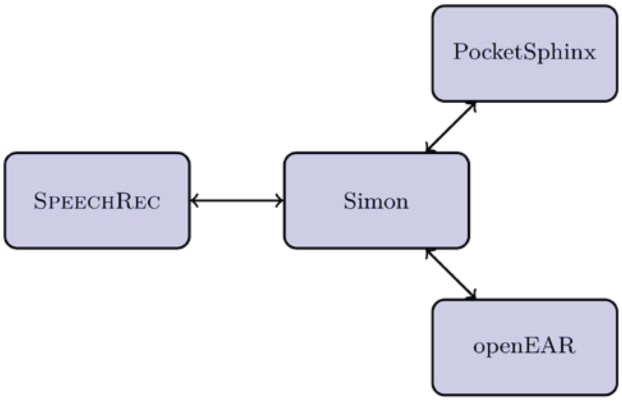


Figure 1: SpeechRec's speech processing architecture.

3.2.1 Speech Model

Spontaneous, conversational speech is among the most challenging speech recognition tasks. In contrast to read speech, users are commonly formulating their statement as they speak, making speech disfluencies such as filler words, false starts (self-interruptions) and repetitions common [32].

These demanding decoding conditions and the pervasiveness and strength of the local dialect necessitated the development of a custom, task dependent speech model of Austrian German for SpeechRec.

A slightly modified and heavily extended version of Schuppler's et al. phonetic dictionary for conversational Austrian German presented in [27] was used as

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

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

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

SpeechRec's dictionary. A task dependent 3-gram language model was estimated 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.

To estimate SpeechRec's continuous HMM acoustic model, the recordings of the SpeechRec pilot and earlier, related studies were augmented with the German version of the open source Voxforge corpus[4], 19 audio books from the LibriVox project[5] as well as the Austrian portion of the ADABA database[6]. 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.

3.2.2 Paralinguistic Analysis

Spoken language input affords more insight into the user's actual, hidden preferences than a lexical analysis of the input could ever detect. In real-life sales dialogs, for example, a user's pronunciation also clearly plays a significant role in helping human sales clerks recommend better products to the customer.

Emotion detection and classification has long been an active field of research. In his seminal paper on emotion categorization, James Russel mapped out human emotions on a two dimensional plane defined by valence, which could be described as the emotions "positivity", and its arousal, an indicator of its intensity [26]. We propose to use this concept of "arousal" as an indicator of the user's personal investment in a given statement, and thus as an indicator of the priority of the ultimately resulting constraints. We believe, that this can help to differentiate casual remarks from a user's core priorities, and thus provide better conflict resolution.

Our developed prototype of SpeechRec analyzes every user input with the open source emotion and affect recognition toolkit openEAR and weighs recognized statements with their associated arousal value, calculated with an SVM classifier trained on the SAL corpus [10, 11, 19].

See Section 3.4 for details on how this information is integrated in SpeechRec's recommendation strategy.

3.2.3 Natural Language Understanding

SpeechRec's natural language understanding process is based around the concept of extracting "statements" from the spoken input. We recognize the following types of statements. Aspect statements, that refer to a product aspect, constraint statements that introduce new critiques or constraints on absolute values to the user model, use case statements, that codify the primary use case of the product envisioned by the user, and finally command statements, which in turn express either "Yes", "No", or "Back".

The parsing process first identifies all possible token matches based on a relatively extensive language profile that consists of attributes (e. g., "weight"), modifiers (e. g., "smaller"), meta modifiers (e. g., "slightly") and commands (e. g., "previous product" but also "for University").

All extracted statements are assigned an "influence" score. A statement's influence is governed by the following three attributes. The statement quality is a number between 0 and 1 that specifies how certain SpeechRec is in the statement capturing the user's intended meaning. For example, a recognition result of simply "main memory" would produce a statement "Good main memory", with a quality of only 0.5, because, while unlikely, the user could have been asking for less, not more main memory. Secondly, the arousal score calculated by the paralinguistic analysis (see Section 3.2.2) is a number between 0 and 2 that specifies how forcefully the user pronounced the relevant segment of the speech input and contributes to the statements "importance". Lastly, the lexical polarity score encodes the meaning of qualifiers ("meta modifiers") such as "a little" and may range from -1 ("not") to 2 ("very"). As shown in Equation 1, all three values are multiplied to arrive at the statement's final influence.

$$statement\ Influence = quality \star arousal \star polarity \quad (1)$$

3.2.4 Dialog Strategy

SpeechRec's dialog strategy is engineered to guide a user through the process of choosing a laptop that fits their requirements.

Initially, SpeechRec will introduce himself and ask the user to describe their ideal product. If the user does not yet have any special preferences, SpeechRec will take initiative and ask one of the following five 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

[4] http://voxforge.org
[5] http://librivox.org
[6] http://www-oedt.kfunigraz.ac.at/ADABA/

that is very portable?”, and “Is a cheap price very important to you?”. Figure 2 shows SpeechRec taking initiative.

Similarly to Shimazu’s ExpertClerk system, SpeechRec automatically selects the question, whose answer it expects to limit the remaining product set the most [31].

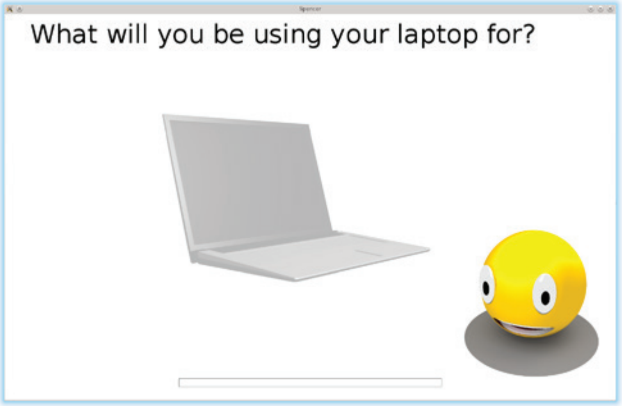

What will you be using your laptop for?

Figure 2: SpeechRec taking initiative after the user expressed no particular preferences. (Text enlarged for readability.)

An overview of SpeechRec’s dialog strategy is outlined in Figure 3.

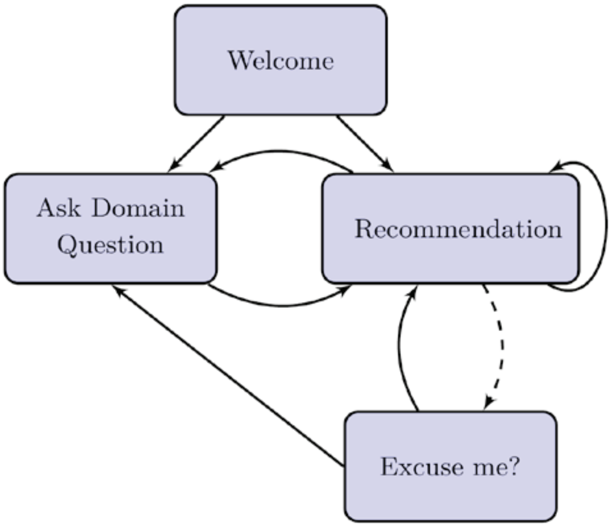


Figure 3: SpeechRec’s dialog strategy: An overview.

When SpeechRec doesn’t understand the user, for example the dialog turn produces no actionable statements, it may ask the user to rephrase what they said. However, in an effort to avoid asking the user to repeat themselves as suggested in [33], SpeechRec never asks that kind of question more than once in succession and never more than twice in a user session. Instead, the system takes initiative and asks a domain question to drive the conversation back to interaction paradigms it understands. We use a simple, semantic approach for end-of-turn detection. After SpeechRec receives actionable input, it waits for 1.5 seconds before considering the turn complete. This timeout is increased to up to 6 seconds if no valid user input was received.

3.3 User Interface

SpeechRec’s user interface was designed to be easy to use. There are no traditional controls, only a small bar on the bottom that shows the current microphone recording level (VU meter). This was included to assure users that SpeechRec is indeed “listening”.

3.3.1 Avatar

Holding a conversation with a computer program can be uncomfortable for some users. This fundamental problem of speech-based interfaces can be partly addressed by providing an avatar for the user to talk to [16, 25]. For SpeechRec, we built an animated character using the 3D animation software Blender[7] and the text to speech framework MARY TTS[8].

3.3.2 Product Display

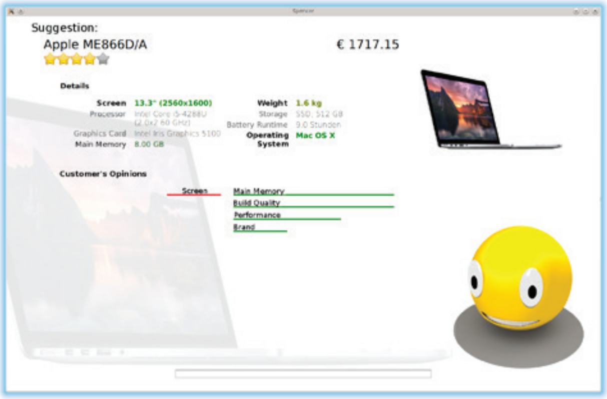
In an effort to keep the user interface simple, SpeechRec limits the displayed attributes of recommendations to a core set of only ten attributes which can be seen in Figure 4. This set is automatically extended with attributes that the user has expressed an opinion on, either explicitly or implicitly (e. g., by mentioning a use case). Attributes that the system considers important for the user are printed in bold, and colored according to how well the current product fulfills the user’s stated requirements of the attribute in question. Below the product’s factual attributes, SpeechRec displays the extracted user sentiment of the current product.

Figure 4 shows the SpeechRec interface after the initial turn of a user from the pilot study. The corresponding user input can be found in Table 1.

[7] http://www.blender.org
[8] http://mary.dfki.de

Table 1: Sample user interaction.

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?



Spencer

Suggestion:

Apple ME866D/A

€ 1717.15

★★★★☆

Details

Screen 13.3" (2560x1600)
Processor Intel Core i5-4288U (2.0x2 60 GHz)
Graphics Card Intel Iris Graphics 5100
Main Memory 8.00 GB
Weight 1.6 kg
Storage SSD; 512 GB
Battery Runtime 9.0 Stunden
Operating System Mac OS X

Customer's Opinions

Figure 4: SpeechRec recommending a product.

3.4 Recommender

SpeechRec uses a knowledge-based recommender employing a rich user preference model.

3.4.1 User Preference Model

The collected user preference model consists of constraints, relative to a given product or on absolute values, and mentioned aspects. We group both of these observations under the term "recommender item"[9].

Every recommender item impacts the recommendation result based on its utility for a given offer and the item's influence determined by its age and the influence of the statement that spawned it. The influence calculation for recommender items is described in Equation 2.

$$influence = \left(1 - \frac{age}{timeToLive}\right)^2 \star statement\ Influence \quad (2)$$

3.4.2 Utility Calculation

All recommender items of the user's preference model contribute to the final utility score of a given item. We differentiate between constraint items and aspect items.

Constraint Items In SpeechRec, constraint utility is expressed as a signed distance from a given or implicit ideal. For example, the constraint screenSize > x calculates the distance from x to the product's screenSize attribute. If the constraint is violated, we multiply the result by -1 to signify a negative "score". Equality constraints evaluate to the (unsigned) distance of the offer's attribute value from the value of the constraint, discounted such that distances of more than 50 % produce negative scores. Absolute constraints, such as screenSize large are interpreted as relating to that attribute space's median value (i. e., larger than the median screen size). Constraints based on the concept of good (bad) are resolved by determining the best (worst) product based on a manually annotated optimality criteria (e. g., minimal for price, maximal for storageCapacity, "SSD" for storageType, etc.).

All distance values are linear and in the range of [-1,1] (negative values represent violated constraints). The final returned value is the logistically scaled distance multiplied by the constraint item's polarity. Refer to Equation 3 for the complete formula. A plot of the utility function with polarity = 1 is shown in Figure 5. Logistic scaling was chosen as it fittingly codifies an intuitive understanding of "satisfying" a given requirement, where it is more important that a given criteria is met, than by how much it is exceeded[10].

$$cUtility = \left(\frac{2}{1 + exp\,(-6 \star distance)} - 1\right) \star polarity \quad (3)$$

[9] Please note that "recommender item" denotes an encoded user preference, not an element of the product space.

[10] For example, an input of "cheaper than 1000 euros" should cause SpeechRec to strongly prefer a product that costs € 995.- over one that costs € 1005.-, whereas the price difference between an offer that costs € 900.- and another one that costs € 910.- is arguably less significant and could be more easily justified by increased value.

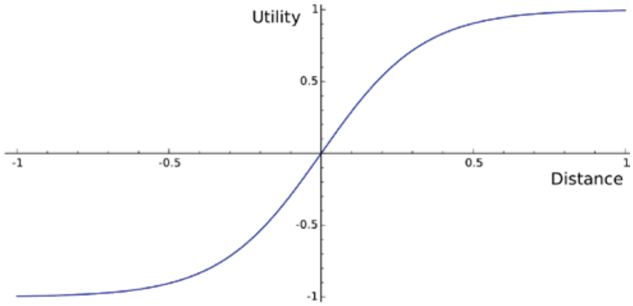


Figure 5: Logistic scale of a constraint's utility based on the offer's attribute's (signed) distance.

Aspect Items Aspect items evaluate to the extracted review sentiment of the referenced aspect of the given product, weighed by their influence. Their utility is further linearly scaled to a tenth of a constraint items' influence, ensuring that constraints are fulfilled before aspects are taken into account. For example, the user input "Screen larger than 15 inch" produces statements that ultimately create one constraint item encoding screenSize > 15 and additionally one aspect item expressing "Screen good". The constraint takes precedence over the review sentiment, but the aspect statement encodes the implicit information that all other things being similar, products with "good" screens should receive preference.

3.4.3 Recommendation Strategy

As Grasch et al. describe in [16], we first limit our product database by products that fulfill at least one of the constraints uttered in the last interaction cycle to avoid earlier collected items of the user model "outvoting" a newly introduced constraint. Out of this reduced set of products, we score each product based on a prior probability calculated from the Amazon sales rank of the product as well as the product rating, the utilities of all recommender items of the user's preference model and discount the final score by a measure of distance from the current recommendation. The full process is outlined in Algorithm 1.

4 Evaluation

In order to evaluate how incorporating nuances captured from spoken natural language input in the recommendation strategy would affect recommendation quality and session length, an empirical study was conducted.

```
**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**
    $utility \leftarrow o.rating() + \frac{o.salesRank()}{MaxSalesRank}$;
    **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 1: Schematic recommendation strategy. Scaling factors omitted for brevity.

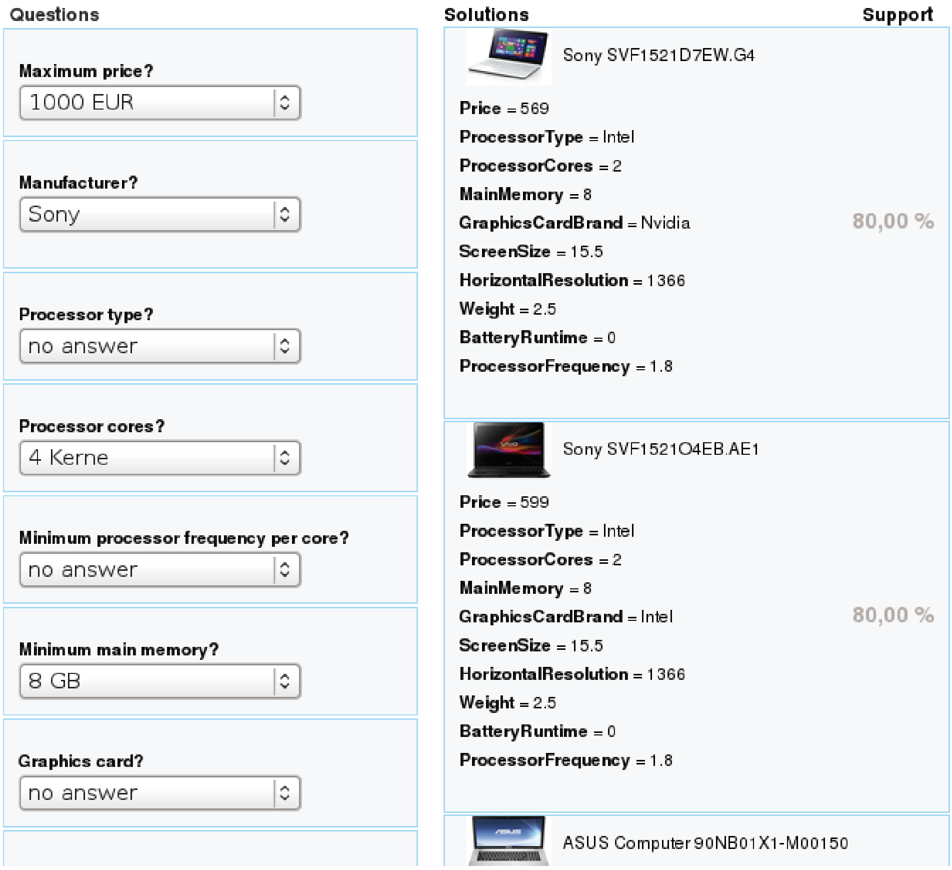
4.1 Compared Systems

We compared SpeechRec against both a "standard" knowledge-based recommender system with traditional interface, as well as a reduced version of itself.

4.1.1 Baseline

The WeeVis framework presented in [15] was used to build a basic knowledge-based recommender system using the same database as the one used for SpeechRec. Out of performance considerations, the product set had to be limited to the 100 most frequently sold laptops, and the mouse-based interface necessitated a reduction of the possible feedback options to what we felt to be the 14 most important "questions"[11]. We will refer to this system, shown in Figure 6, as "WeeVis".

[11] The following constraint options were provided: Maximum prize, manufacturer name, processor brand, processor cores, minimum



Questions

Maximum price?
1000 EUR

Manufacturer?
Sony

Processor type?
no answer

Processor cores?
4 Kerne

Minimum processor frequency per core?
no answer

Minimum main memory?
8 GB

Graphics card?
no answer

Solutions | Support

Sony SVF1521D7EW.G4

Price = 569
ProcessorType = Intel
ProcessorCores = 2
MainMemory = 8
GraphicsCardBrand = Nvidia
ScreenSize = 15.5
HorizontalResolution = 1366
Weight = 2.5
BatteryRuntime = 0
ProcessorFrequency = 1.8

80,00 %

Sony SVF1521O4EB.AE1

Price = 599
ProcessorType = Intel
ProcessorCores = 2
MainMemory = 8
GraphicsCardBrand = Intel
ScreenSize = 15.5
HorizontalResolution = 1366
Weight = 2.5
BatteryRuntime = 0
ProcessorFrequency = 1.8

80,00 %

ASUS Computer 90NB01X1-M00150

Figure 6: Baseline knowledge-based recommender system WeeVis showing best matching products on overconstrained user input. Clicking on a suggested product opens a more detailed description.

4.1.2 Reduced SpeechRec

We evaluated a reduced setup of SpeechRec, which retains the novel user interface but limits any recommender item's utility to -1, 0 or 1. Additionally, all statements' influence, and in turn all recommender items' influence is reduced to purely depend on the statement's age, without taking any lexical or paralinguistic nuances into account. This reduces SpeechRec's recommendation strategy to something comparable to e. g., a WeeVis based recommender system. We refer to this system as "SpeechRec reduced".

4.1.3 SpeechRec

The full version of SpeechRec uses the following extracted nuances over SpeechRec reduced. Lexical polarity ("slightly cheaper"), statement quality (parser confidence), paralinguistic importance (arousal) and the degree of satisfaction of each recommender item (activation of the utility function). We refer to this system simply as "SpeechRec".

4.2 Task

Users received instructions to imagine that their laptop, should they already own one, had been stolen the day before the study. They were asked to then put themselves in the mindset of shopping for a replacement.

Users of the traditional, mouse-based WeeVis interface received a very short description of how to interact with it. Study participants using SpeechRec were instead simply told that they would be using a "virtual shopping assistant", whom they could talk to like they would with a human sales clerk.

Both groups were instructed to notify the person overseeing the trial when they found a product they wanted to accept or it became clear that no such product could be found. At that point they would be immediately

processor frequency per core, minimum main memory, graphics card brand, minimum and maximum display size, anti glare coated screen, maximum weight, minimum warranty duration, minimum battery runtime and operating system.

presented with a questionnaire assessing their opinion on the systems they used, using a slightly modified version of the standard usability survey scale presented in [2], as well as the last shown product, and the used interaction method. Except for questions about the user's personal background (age, occupation, etc.), all answers were verified with inverse control questions. Participants' responses reported in the following section follow the form positiveQuestionResponse – negativeQuestionResponse.

4.3 Test Demography

The study participants, primarily students and post-graduate researchers, were split into 3 groups of 22 each, for a total of 66 participants. A detailed breakdown of the demography of the study participants can be found in Table 2.

Table 2: Demography of the participants of the empirical study.

Characteristics	WeeVis	SpeechRec Reduced	SpeechRec Full
Male	17	15	18
Female	5	7	4
Total	22	22	22
Mean age	25	25	28
Personally own a laptop	91 %	91 %	91 %
Seeked help when buying their last laptop	23 %	36 %	32 %

5 Results

The following section discusses the results of the study outlined in Section 4.

The statistical significance of results ascertained through comparative analysis was verified through Welch's t-tests.

5.1 Input Processing

In an effort to judge the effectiveness of SpeechRec's natural language processing layer, we manually reviewed all collected interaction logs, comparing SpeechRec's input interpretation with that of a human reviewer. We define a fault as any misinterpretation (or missing interpretation) of user input, including failures that are a result of our limited dialog strategy (e. g., "How fast is fast?"). We found that out of 579 turns, 230, or 39.72 %, failed to parse. Another 55 turns, 9.5 %, included errors in their interpretation, leaving 294 turns, or 50.78 %, faultless.

While these numbers may look discouraging initially, it is crucial to keep in mind that most "failed" turns simply resulted in SpeechRec temporarily taking initiative (see Section 3.2.4), and as such very often went unnoticed by the user. This is well reflected in the results of the questionnaire, which asked users to rate the reliability of the speech input, where SpeechRec's system achieved a median rating of 2 on a scale of -3 to 3 (higher is better).

That said, while we believe SpeechRec's natural language processing capability to be sufficient for the purpose of this study, there is obviously room for improvement. In an effort to quantify this, we simulated interaction sessions using manually transcribed reference transcriptions. The number of failed turns reduced to 81 (13.99 %), with 56 (9.67 %) partial faults, increasing the number of correct parses to 442 (86.01 %). Speech recognition errors therefore account for roughly half of SpeechRec's mistakes. Conversational speech recognition remains a hard task, but it seems reasonable to assume that an improved speech recognition component could substantially improve SpeechRec's performance in practice.

5.2 Usability

Participants were asked to evaluate the usability of the system they used (see Section 4.2). The resulting scores are presented in Figure 7. Interestingly, all three reviewed systems score similarly well, with no statistically significant differences between them.

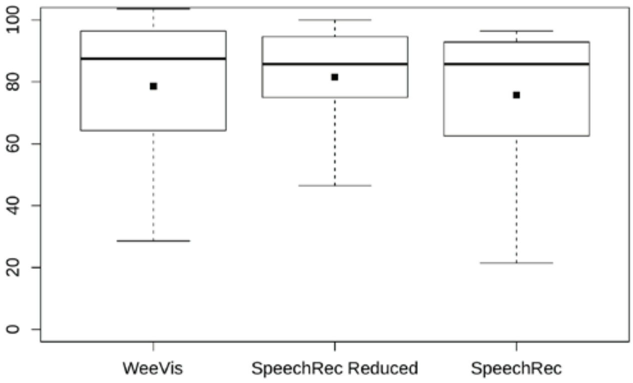


Figure 7: Overall SUS scores. (Higher is better. Black squares indicate the arithmetic mean.)

As described in Section 3.3.2, SpeechRec's user interface only displays a small subset of known product attributes by default. We found that users did not hesitate to also talk about attributes that were not initially shown on screen. We believe this to be a noteworthy advantage of speech-based user interfaces: SpeechRec allowed users to organically work with a wide range of attributes without cluttering the user interface with rarely used controls.

5.3 Recommendation Performance

All study participants were requested to rate the product last presented to them by their respective system. Figure 8 shows the result.

SpeechRec outperforms both the traditional knowledge-based recommender and the reduced version. The improvement over the WeeVis baseline shows a trend, but does not quite reach the commonly accepted threshold for statistical significance ($p \approx 0.07$).

In addition to an improvement in recommendation quality, SpeechRec's speech-based interface led to substantially shorter recommendation sessions ($p < 0.001$). Figure 9 shows an overview of measured session lengths. Figure 10 further highlights the reduction in dialog turns and recommendation iterations between the reduced and the full version of SpeechRec, afforded by the additional nuances captured from the user input.

It is worth noting that while the full version of SpeechRec presents a clear improvement over the WeeVis baseline, the benefit of the novel, speech-based interface alone, as represented by the reduced version of SpeechRec, is much less significant.

We believe the substantial improvement of SpeechRec over SpeechRec reduced to be primarily rooted in the system's conflict resolution. As is typical for recommender systems in non-trivial domains, over-specification of requirements proved to be extremely common[12]. In order to recommend an appropriate product, the system needs to acknowledge and consider the finer nuances of the user's input, thus explaining the superior performance of SpeechRec over SpeechRec reduced.

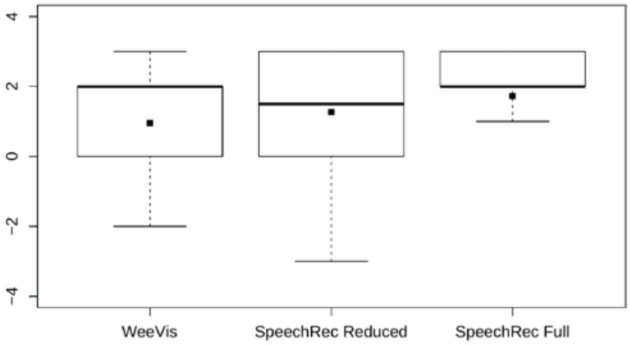


Figure 8: Participant's subjective score of the last shown product of the interaction session ([-3,3], higher is better. Black squares indicate the arithmetic mean.)

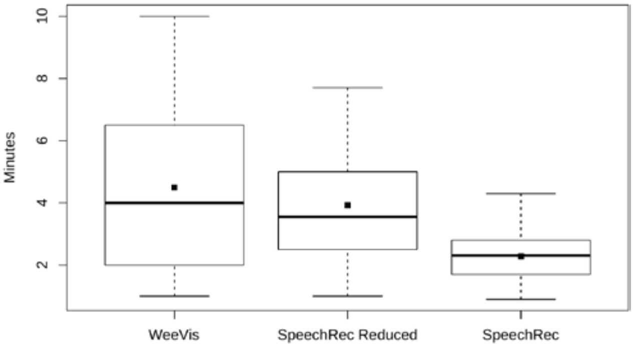


Figure 9: Recommendation session duration. (Lower is better. Black squares indicate the arithmetic mean.)

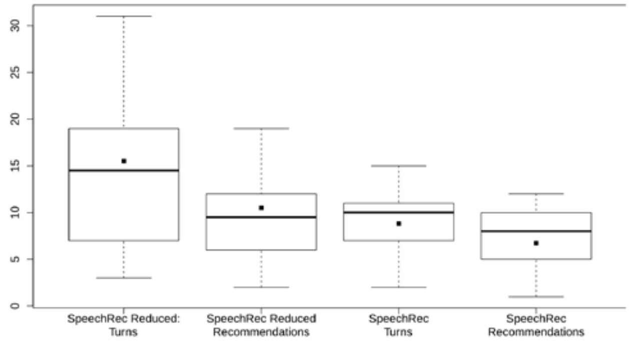


Figure 10: Recommendation session length in completed iterations. (Lower is better. Black squares indicate the arithmetic mean.)

6 Conclusion and Further Work

In this paper we present SpeechRec, a natural language driven, knowledge-based recommender system using speech interaction, that detects lexical and paralinguistic nuances and incorporates them into its recommendation strategy. We reported on an empirical study, comparing SpeechRec against a traditional knowledge-based recommender system, where we found that the richer user preference model employed by SpeechRec allowed it to significantly outperform the baseline system, enabling users

[12] For example, consider the user input "I really want a fast laptop, that is still kind of light". The fastest laptop is not also the lightest.

to find better products, in substantially shorter recommendation sessions. Furthermore, we compared SpeechRec to a restricted version of itself, which uses the same speech-based user interface but does not extract the aforementioned subtextual information. We showed that this reduced version indeed provided a much less pronounced improvement over the baseline, verifying our assumption that the more nuanced information SpeechRec extracts from user input is a major reason for its favorable performance.

In future works on speech-based recommender systems, we hope to revisit paralinguistic analysis and expand our current implementation. While the basic priority (arousal) detection implemented in SpeechRec already proved promising, we think that further analysis could provide helpful information for improving the recommendation strategy and ultimately also recommendation quality. For example, user uncertainty and frustration both have acoustic indicators and could provide valuable feedback to steer user interaction.
Moreover, the conversations between SpeechRec and its users provide insights into what people expect of natural-language driven recommender systems and should be studied further to improve follow-up systems' dialog strategies.

Finally, it is worth stressing that many of the main contributions of this paper not only apply to speech-based recommender systems. Given that speech-enabled applications are rapidly gaining popularity, fueled by the success of mobile devices, it is the authors strong believe that many future user interfaces would eventually benefit from the betterment of understanding of user's intention afforded by a more nuanced, more "human" approach of input processing.

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

References

[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.

[2] John Brooke. SUS - a quick and dirty usability scale. Usability evaluation in industry, 189:194, 1996.

[3] Robin Burke. Integrating knowledge-based and collaborative-filtering recommender systems. In Proc. of the Worksh. on AI and Electr. Commerce, pages 69–72, 1999.

[4] Robin Burke. Knowledge-based recommender systems. Encyclopedia of library and information systems, 69(Supplement 32):175–186, 2000.

[5] 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, AAAI'96, pages 462–468. AAAI Press, 1996.

[6] Li Chen and Pearl Pu. Survey of preference elicitation methods. Technical report, Swiss Federal Inst. of Techn. in Lausanne (EPFL, 2004.

[7] Li Chen and Pearl Pu. Evaluating critiquing-based recommender agents. In Proc. of the 21st Natl. Conf. on AI - Volume 1, AAAI'06, pages 157–162. AAAI Press, 2006.

[8] 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, pages 59–66, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[9] Ruihai Dong, Markus Schaal, MichaelP. O Ḿ ahony, Kevin McCarthy, and Barry Smyth. Opinionated product recommendation. In SarahJane Delany and Santiago Ontañón, editors, Case-Based Reasoning Research and Development, volume 7969 of Lecture Notes in Computer Science, pages 44–58. Springer Berlin Heidelberg, 2013.

[10] F. Eyben, M. Wollmer, and B. Schuller. OpenEAR - introducing the munich open-source emotion and affect recognition toolkit. In Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd Intl. Conf. on, pages 1–6, Sept 2009.

[11] Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller. Recent developments in openSMILE, the munich open-source multimedia feature extractor. In Proc. of the 21st ACM Intl. Conf. on Multimedia, MM '13, pages 835–838, New York, NY, USA, 2013. ACM.

[12] A. Felfernig and R. Burke. Constraint-based recommender systems: Technologies and research issues. In Proc. of the 10th Intl. Conf. on Electronic Commerce, ICEC '08, pages 3:1–3:10, New York, NY, USA, 2008. ACM.

[13] A. Felfernig, K. Isak, K. Szabo, and P. Zachar. The VITA financial services sales support environment. In Proc. of the 19th Natl. Conf. on Innovative Applications of AI - Volume 2, IAAI'07, pages 1692–1699. AAAI Press, 2007.

[14] Alexander Felfernig, Michael Jeran, Gerald Ninaus, Florian Reinfrank, Stefan Reiterer, and Martin Stettinger. Basic approaches in recommendation systems. In Recommendation Systems in Software Engineering, pages 15–37. Springer, 2014.

[15] Alexander Felfernig, Stefan Reiterer, Martin Stettinger, and Michael Jeran. An overview of direct diagnosis and repair techniques in the WeeVis recommendation environment. In Knowledge-based Configuration: From Research to Business Cases, pages 297–307. Elsevier, 2014.

[16] 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, pages 157–164. ACM, 2013.

[17] Lorraine McGinty and James Reilly. On the evolution of critiquing recommenders. In Recommender Systems Handbook, pages 419–453. Springer, 2011.

[18] 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, pages 276–290. Springer, 2003.

[19] Gary McKeown, Michel François Valstar, Roderick Cowie, and Maja Pantic. The SEMAINE corpus of emotionally coloured character interactions. In Multimedia and Expo (ICME), 2010 IEEE Intl. Conf. on, pages 1079–1084. IEEE, 2010.

[20] George A. Miller. WordNet: A lexical database for english. Commun. ACM, 38(11):39–41, November 1995.

[21] 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, pages 1825–1828, New York, NY, USA, 2010. ACM.

[22] Daniel Naber. OpenThesaurus: Building a thesaurus with a web community. Retrieved January, 3:2005, 2004.

[23] Michael J. Pazzani and Daniel Billsus. Content-based recommendation systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, The Adaptive Web, pages 325–341. Springer, 2007.

[24] Pearl Huan Z. Pu and Pratyush Kumar. Evaluating example-based search tools. In Proc. of the 5th ACM Conf. on Electronic Commerce, EC '04, pages 208–217, New York, NY, USA, 2004. ACM.

[25] 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. ACM Trans. Comput.-Hum. Interact., 12(4):329–355, 2005.

[26] James A Russell. A circumplex model of affect. Journal of personality and social psychology, 39(6):1161, 1980.

[27] Barbara Schuppler, Martine Adda-Decker, and Juan A. Morales-Cordovilla. Pronunciation variation in read and conversational Austrian German. In Interspeech 2014, pages 1453–1457, Singapore, 2014.

[28] Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. A new hybrid dependency parser for german. Proc. of the German Society for Comp. Linguistics and Language Techn., pages 115–124, 2009.

[29] Rico Sennrich, Martin Volk, and Gerold Schneider. Exploiting synergies between open resources for german dependency parsing, POS-tagging, and morphological analysis. In RANLP, pages 601–609, 2013.

[30] 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, pages 191–202. Springer, 2007.

[31] 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, pages 1443–1448, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

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

[33] 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.

[34] Cynthia A. Thompson, Mehmet H. Göker, and Pat Langley. A personalized system for conversational recommendations. J. Artif. Int. Res., 21(1):393–428, March 2004.

[35] Pontus Wärnestål. User evaluation of a conversational recommender system. In Proc. of the 4th Workshop on Knowledge and Reasoning in Practical Dialogue Systems, 2005.

Author Biographies

Peter Grasch
Graz University of Technology, Austria
peter@grasch.net

Peter Grasch is a master's student at Graz University of Technology, majoring in Computer Science with a special focus on Algorithms and Software Technology. He received his Bachelor's degree in the fall of 2012 from the same university. Next to his studies, Peter Grasch has been involved in numerous research projects revolving around speech recognition, including the EU project Astromobile. His research interests include human-computer interaction, recommender systems, and speech recognition.

Prof. Alexander Felfernig
Graz Universtiy of Technology, Austria
afelfern@ist.tugraz.at

Alexander Felfernig is a full professor at the Graz University of Technology (Austria) since March 2009 where he directs the Applied Software Engineering (ASE) research group at the Institute for Software Technology (IST). He received his PhD in Computer Science from the University of Klagenfurt. Alexander Felfernig has published numerous papers in renowned international conferences and journals, co-authored the book on "Recommender Systems" and acted as an organizer of international conferences and workshops. He is also a member of the Editorial Board of Applied Intelligence and the Journal of Intelligent Information Systems. His research interests include configuration systems, recommender systems, model-based diagnosis, software requirements engineering, different aspects of human decision making, and knowledge acquisition methods.