

 Open access • Journal Article • DOI:10.1109/MIS.2007.55

Informed Recommender: Basing Recommendations on Consumer Product Reviews

— [Source link](#) 

Silvana Aciar, Debbie Zhang, Simeon J. Simoff, John Debenham

Institutions: University of Technology, Sydney

Published on: 01 May 2007 - IEEE Intelligent Systems (IEEE Computer Society)

Topics: Recommender system and Collaborative filtering

Related papers:

- [Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions](#)
- [Mining and summarizing customer reviews](#)
- [Matrix Factorization Techniques for Recommender Systems](#)
- [Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations](#)
- [Opinion Mining and Sentiment Analysis](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/informed-recommender-basing-recommendations-on-consumer-1man2b75lo>

Informed Recommender: Basing Recommendations on Consumer Product Reviews

Silvana Aciar, *University of Girona, Spain*

Debbie Zhang, Simeon Simoff, and John Debenham, *University of Technology, Sydney*

Informed Recommender uses prioritized consumer product reviews to make recommendations. Using text-mining techniques, it maps each piece of each review comment automatically into an ontology.

Recommender systems attempt to predict items in which a user might be interested, given some information about the user's and items' profiles. Most existing recommender systems use content-based or collaborative filtering methods or hybrid methods that combine both techniques (see the sidebar for more details). We created Informed

Recommender to address the problem of using consumer opinion about products, expressed online in free-form text, to generate product recommendations.

Our process builds a collection of relevant consumer product reviews. Technically, the procedure for collecting reviews follows the algorithms for automated news extraction from news sites.¹ Once the product opinions mining base is populated, we employ text-mining techniques to extract useful information from review comments. Here, we discuss the overall framework for automating the use of consumer reviews and the framework's individual components. Where possible, we've used existing algorithms (for example, in the text-mining process) because our goal is to demonstrate our approach's strengths.

System overview

Our recommender process involves several steps. For review information to be useful for the recommendation process, we must translate it into a structured form and communicate it to the process in a form suitable for generating recommendations. We've developed and employed an ontology to translate opinions' quality and content into a form the recom-

mender process can use. The text-mining process automatically maps the review comments into the ontology's information structure.

A ranking mechanism computes a product's rating using the information from the consumer reviews stored in the ontology. It prioritizes that information with respect to the consumer's level of expertise in using the product under consideration. The recommender system makes a recommendation based on the ontology data. Therefore, the recommendation quality depends on accurately mapping the proper knowledge from the semantic features in the review comments into the ontology structure. Figure 1 shows our proposed system's overall process structure, and the following sections outline the steps involved.

Representing consumer reviews

Our first step was to find a suitable tool for extracting the information in the text and converting it into structured data. Identifying an appropriate representation of consumer opinions that the system can use is a key problem. One way to convert these opinions to a structured form is to use a translation ontology, which is typically used as a form of knowledge representation and sharing. In this application, the

Content-Based Filtering, Collaborative Filtering, and Hybrid Methods

Content-based filtering methods use information about the item itself to make suggestions, rather than information about other consumers' preferences. Such systems emulate the behavior of a consumer recommending a product to a friend because he or she has used the product and knows the friend's preferences in terms of product features.

Content-based recommender systems uniquely characterize each consumer without having to match his or her interests to other consumers. They can provide a list of content features that explain why an item has been recommended. Such a list can strengthen consumer confidence in the recommendation and reflect the consumer's own preferences. In the content-based approach, consumers can provide some initial information about the product to assist the system.

Collaborative filtering makes recommendations about the preferences of a user (filtering) on the basis of the other users' collective taste information (collaborative). The underlying assumption of such methods is that those who agreed in the past also tend to agree again in the future.¹ In other words, these systems emulate the behavior of a consumer recommending a product to a friend because other consumers that she knows, and believes have tastes similar to her friend's, like the product.

Technically, such a system operates similarly to a case-based-reasoning system, without the adaptation step. It maintains a case base of individual consumers' preferences, finds other consumers whose known preferences correlate significantly with the user's, and recommends to the user other items that matched patrons enjoyed. The system can give the user a list of some of these patrons and their previous purchases to provide an explanation and make the user more confident about the recommendation.

This approach requires a sufficient number of consumer ratings. Collaborative-filtering systems use a collection of historical rating data of m users on n products as input, which they collect by asking users to rate products on a scale.² Collecting such ratings requires the consumer to spend time responding, and the actual values might not necessarily provide reliable estimates of consumer preferences. Another issue such systems face is how to recommend products that haven't been rated by enough consumers.

The traditional collaborative-filtering approach doesn't provide effective recommendation strategies. Collaborative filtering and content-based filtering perform unsatisfactorily without large amounts of usage data, which discourages users from using the system, and the system's performance can't be improved without sufficient user participation.³ To solve this problem, we use reviews from experienced consumers that are already available on the Internet to determine the most popular product according to the given criteria.

Rather than completing forms with rating values, many consumers prefer to use natural language and express their opinions about the product in a free-text form, similar to a conversation with a friend. In the online world, consumers can exchange their experiences with a product in several ways:^{4,5} product review forums, virtual-community logs, product discussion boards, and e-commerce Web sites. There's growing evidence that such forums inform and influence consumers' purchase decisions.^{6,7} Decision makers use expert advice either to make their decisions more accurate or to reduce their effort expenditure.⁸ Despite the importance and value of such information, no comprehensive mechanism exists that formalizes selecting, retrieving, and using opinions.

Part of the problem resides in the complexity of extracting information from text data and converting it into product recommendation. Gediminas Adomavicius provided an overview of recent developments in recommender systems.³ According to this review, recommender systems that use review comments using text mining techniques are yet to be developed. Francesco Ricci and René Wietsma proposed using review comments to give some explanation about recommendations made.^{9,10} They believed the review comments could be widely used in recommender systems and result in better recommendations. So far, Ricci and Wietsma seem to have created the only recommender system that integrates reviews in the recommendation process.^{9,10} They use product reviews in the product selection decision process for a mobile recommender system, employing social-filtering algorithms to extract knowledge from the reviews. Their system aims to improve the explanation of the recommendation by providing relevant reviews from users with similar tastes. The reviews are used to explain the recommendations but not to actually make recommendations.

References

1. D. Billsus and M. Pazzani, "Learning Collaborative Information Filters," *Proc. 15th Int'l Conf. Machine Learning*, Morgan Kaufmann, 1998, pp. 46–54.
2. W. Yang, Z. Wang, and M. You, "An Improved Collaborative Filtering Method for Recommendations' Generation," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics*, IEEE Press, 2004, pp. 4135–4139.
3. G. Adomavicius, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 6, 2005, pp. 734–749.
4. C. Dellarocas, "The Digitization of Word-of-Mouth: Promise and Challenges of Online Feedback Mechanisms," *Management Science*, vol. 49, no. 10, 2003, pp. 1407–1424.
5. N. Curien et al., "Online Consumer Communities: Escaping the Tragedy of the Digital Commons," *Internet and Digital Economics*, E. Brousseau and N. Curien, eds., Cambridge Univ. Press, 2006, pp. 201–219.
6. S. Senecal and J. Nantel, "The Influence of Online Product Recommendations on Consumers' Online Choices," *J. Retailing*, vol. 80, no. 2, 2004, pp. 159–169.
7. J. Chevalier and D. Mayzlin, "The Effect of Word of Mouth on Sales: Online Book Reviews," *NBER Working Paper Series*, US Nat'l Bureau of Economic Research, 2003.
8. G.E. Schrah, R.S. Dalal, and J.A. Sniezek, "No Decision-Maker Is an Island: Integrating Expert Advice with Information Acquisition," *J. Behavioral Decision Making*, vol. 19, no. 1, 2006, pp. 43–60.
9. R. Wietsma and F. Ricci, "Product Reviews in Mobile Decision Aid Systems," *Proc. Workshop Pervasive Mobile Interaction Devices*, in conjunction with Pervasive 2005, LMU Munich, 2005, pp. 15–18.
10. F. Ricci and R. Wietsma, "Product Reviews in Travel Decision Making," *Information and Communication Technologies in Tourism: Proc. Int'l Conf.*, Springer, 2006, pp. 296–307.

ontology contains two main parts: *opinion quality* and *product quality*, which summarize the consumer's skill level and experience with the product under review, respectively. Figure 2 shows the ontology's general structure. Opinion quality includes several variables to measure the opinion provider's expertise with the product. Product quality represents the opinion provider's valuation of the product features, which is highly domain specific.

Mapping comments to the ontology

An ontology provides a controlled vocabulary and relationships among words to describe the consumer's skill level and experience with the product in the review comment in the system. We need to define the classes and relationships in the ontology only once and can use them until the products have new features. Each review comment is represented as an ontology instance. Manually mapping the ontology instances is tedious and time consuming, so we've automated the process using text-mining techniques. As the ontology has been defined, the mapping process includes the identification of both the classes involved in the instance and their attributes. The mapping process involves two steps.

The first step, sentence selection and classification, identifies the class attributes. In the valuation from the text data, the user assigns each feature from the comment either a "good" or "bad" value. Therefore, the system selects and classifies the sentences in the review into three categories: "good," "bad," and "quality." The "good" category groups sentences containing information about features that the consumer has valued as product strengths. The "bad" category groups sentences containing information about features that the consumer considers product weaknesses. The "quality" category contains the sentences that indicate the opinion quality as determined by the consumer's skill level.

Once the system has selected the relevant sentences, the second step, *concept identification*, identifies the classes to which the sentences belong. The concepts implicated in the sentences determine the classes in the ontology, which are identified by related words.

Sentence selection and classification

Under the text-mining paradigm, this application treats each sentence as a document. We first considered using a shallow parser as an analyzer tool to group review sentences. However, most such parsers give complicated, incorrect results. Furthermore, each document is very short. Classification algorithms based on term frequencies don't provide satisfactory results either. So, our system employs rule-based classification techniques.

At this stage, our work has focused on using text mining for automatically mapping review comments onto ontology instances. Hence, we employed off-the-shelf text-mining kits. Sholom M. Weiss and

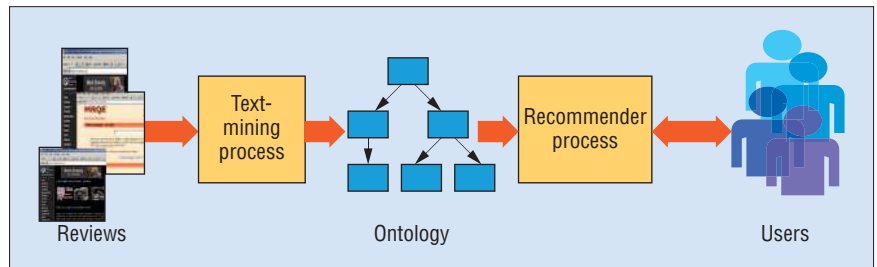


Figure 1. The process structure of a recommender system that uses online consumer reviews as its input information source.

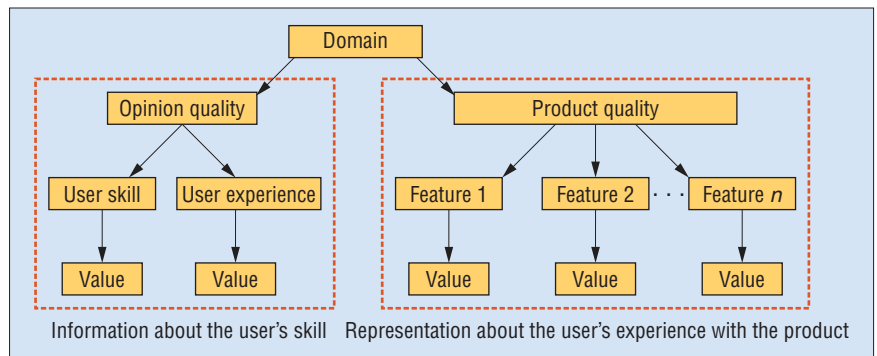


Figure 2. The structure of the ontology used in the recommendation from consumer opinion applications.

his colleagues used the Text-Miner Software Kit and the Rule Induction Kit for Text to obtain classification rule sets.² TMSK generates a dictionary from a set of documents and converts a set of sentences into sparse vectors based on the dictionary. RIKTEXT uses the dictionary and the vectors representing each category to learn how to classify sentences into categories. RIKTEXT is a complete software package for learning decision rules from document collections. It induces rules automatically from training data and outputs a rule set of classifications of "good," "bad," and "quality" categories.

Test case

We used opinions from 68 reviews of the Canon PowerShot SD500 (Digital IXUS 700) digital camera from the Digital Photography Review Web site (www.dpreview.com) to create the training data set. We obtained 195 sentences for the "good" category, 127 sentences for "bad," and 47 sentences for "quality." We used two-thirds of the data for training and the rest for testing. We specified how many cases should be used for testing, and RIKTEXT selected them randomly. Table 1 presents the results. As you can see, it displays a number of rule sets. This test used 123 test cases.

In the table, "*" indicates the best rule set according to the error rate and simplicity. "Variables" indicates the total number of conjuncts in the left-hand side of the rules. The column "Training error" gives the error rate of the rule sets on the training data. "Test error" is an error-rate estimate, and "Test standard deviation" is the standard deviation of the estimate. "Error/variable" gives an indication of the solution's quality. The chosen rules are those that have the minimum error rate or are very close to the minimum but may be simpler than those that have the minimum. The mean variable (the average

Table 1. Pruned rule sets to classify sentences into the “good” category.

| Rule set | Number of rules | Variables | Training error | Test error | Test standard deviation | Error/variable |
|----------|-----------------|-----------|----------------|------------|-------------------------|----------------|
| 1 | 32 | 34 | 0.1696 | 0.3171 | 0.0420 | 0.00 |
| 2* | 30 | 31 | 0.1736 | 0.3171 | 0.0420 | 0.67 |
| 3 | 29 | 29 | 0.1795 | 0.3333 | 0.0425 | 2.00 |
| 4 | 24 | 24 | 0.2091 | 0.3659 | 0.0434 | 3.00 |
| 5 | 22 | 22 | 0.2249 | 0.3496 | 0.0430 | 4.00 |
| 6 | 1 | 1 | 0.2308 | 0.4715 | 0.0450 | 0.14 |

*best rule set according to error rate and simplicity.

Table 2. Pruned rule sets to classify sentences into the “bad” category.

| Rule set | Number of rules | Variables | Training error | Test error | Test standard deviation | Error/variable |
|----------|-----------------|-----------|----------------|------------|-------------------------|----------------|
| 1 | 25 | 28 | 0.1731 | 0.3710 | 0.0434 | 0.00 |
| 2 | 25 | 27 | 0.1755 | 0.3710 | 0.0434 | 1.00 |
| 3 | 24 | 25 | 0.1779 | 0.3468 | 0.0427 | 1.50 |
| 4 | 20 | 20 | 0.2043 | 0.3065 | 0.0414 | 2.20 |
| 5 | 14 | 14 | 0.2500 | 0.2742 | 0.0401 | 3.17 |
| 6 | 12 | 12 | 0.2740 | 0.2661 | 0.0397 | 5.00 |
| 7 | 10 | 10 | 0.3029 | 0.2661 | 0.0397 | 6.00 |
| 8* | 9 | 9 | 0.3221 | 0.2581 | 0.0393 | 8.00 |
| 9 | 8 | 8 | 0.3438 | 0.2742 | 0.0401 | 9.00 |
| 10 | 7 | 7 | 0.3678 | 0.2823 | 0.0404 | 10.00 |
| 11 | 1 | 1 | 0.3870 | 0.6532 | 0.0427 | 1.33 |

*best rule set according to error rate and simplicity.

1. fast>=1 → gd
2. results → gd
3. good>=1 → gd
4. nice → gd
5. overall → gd
6. pocketable → gd
7. great → gd
8. use>=1 → gd
9. underwater>=2 → gd
10. problems & no → gd
11. impressed → gd
11. function → gd
12. like>=1 → gd
13. better>=1 → gd
14. speed → gd
15. compact → gd
16. pocket → gd
17. ps>=1 → gd
18. love → gd
19. user → gd
20. sd500 → gd
21. quality>=1 → gd
22. small → gd
23. able → gd 24. sd → gd
25. far → gd
26. photos → gd
27. shots → gd
28. really>=1 → gd
29. mode>=1 → gd
30. [TRUE] → ~gd

Figure 3. The selected rule set to classify sentences into the “good” category. The rules are the output of RIKTEXT.

number of variables of the resampled rule set that approximates in size the rule set for the full data) for each test case was zero.

Tables 2 and 3 show the rule sets obtained from classification of the “bad” and “quality” categories. Each of these tests used 124 test cases.

Table 4 shows the precision, recall, and f-measure obtained from training and test cases for all three categories. Figures 3–5 show the selected rule sets to classify sentences into each category.

Concept identification

Once each sentence has been classified into a category, the concept (class) in the ontology implicated in the sentence must be identified. Each concept in the ontology contains a label name and a related word list. The lists contain vocabulary (sets of keywords) that the system can use to match the concept with a sentence in the comments. For example, related words for the concept “comparison” include “compare,” “compared,” “equal,” and “same.”

Notes on implementation

We manually created the ontology to ensure that it was complete and well defined. However, mapping the ontology instances from review comments is fully automatic after training. Similar to other classification applications, collecting and labeling training exam-

ples for sentence classification are manual processes. Once the system has been trained, it automatically classifies a sentence as “good” or “bad.” In the concept identification step, we created the synonym database manually. The system identifies the concept automatically if it identifies in the sentence a keyword from the database.

Ranking

The review comments are first mapped onto the ontology to make the ranking calculations possible. As we mentioned earlier, the ontology contains two main parts: opinion quality and product quality. The system computes a set of measures—opinion quality (OQ), feature quality (FQ), overall feature quality (OFQ), and overall assessment (OA)—on the basis of the data in the ontology. OQ evaluates opinions’ weighting value according to the opinion provider’s expertise. OFQ is the global valuation of the feature from all reviews, which is calculated from the FQ value of individual comment. OA provides a final score of the product based on the valuation of each feature. The system makes a recommendation in response to a user request on the basis of these measurements. The recommendation is based on the review comments as summarized by an OFQ value for each feature. In this section, we detail the calculation of these measures.

Table 3. Pruned rule sets to classify sentences into the “quality” category.

| Rule set | Number of rules | Variables | Training error | Test error | Test standard deviation | Error/variable |
|----------|-----------------|-----------|----------------|------------|-------------------------|----------------|
| 1 | 20 | 34 | 0.0422 | 0.1371 | 0.0309 | 0.00 |
| 2 | 17 | 28 | 0.0552 | 0.1290 | 0.0301 | 0.67 |
| 3 | 13 | 18 | 0.0779 | 0.1774 | 0.0343 | 0.70 |
| 4 | 9 | 9 | 0.1104 | 0.1371 | 0.0309 | 1.22 |
| 5* | 7 | 7 | 0.1299 | 0.1129 | 0.0284 | 3.00 |
| 6 | 6 | 6 | 0.1396 | 0.1210 | 0.0293 | 3.00 |
| 7 | 5 | 5 | 0.1558 | 0.1371 | 0.0309 | 5.00 |
| 8 | 4 | 4 | 0.1818 | 0.1371 | 0.0309 | 8.00 |
| 9 | 3 | 3 | 0.2143 | 0.1210 | 0.0293 | 10.00 |
| 10 | 2 | 2 | 0.2500 | 0.1290 | 0.0301 | 11.00 |
| 11 | 1 | 1 | 0.3019 | 0.1290 | 0.0301 | 16.00 |

*best rule set according to error rate and simplicity.

Table 4. Additional statistics obtained from the training and test cases for all three categories.

| | “Good” category | | “Bad” category | | “Quality” category | |
|-----------|-----------------|------------|----------------|------------|--------------------|------------|
| | Training cases | Test cases | Training cases | Test cases | Training cases | Test cases |
| Precision | 71.6049 | 67.5676 | 73.0159 | 70.3704 | 74.0741 | 60.0000 |
| Recall | 89.2308 | 76.9231 | 54.1176 | 44.1860 | 64.5161 | 37.5000 |
| F-measure | 79.4521 | 71.9424 | 62.1622 | 54.2857 | 68.9655 | 46.1538 |

Rating the consumer skill level

People with diverse experience and skill levels made review comments. In general, people who have been using a product longer can provide more reliable opinions. Therefore, rather than treating all opinions equally, we should give more experienced people’s opinions more weight than those of people with little knowledge of the product.

We define *opinion quality* as the sum of the weight w_j given for each variable j representing the skills and experiences of consumer i divided by the number of variables representing the information about the consumer’s skill and expertise provided in the ontology:

$$OQ_i = \frac{\sum_j^n w_j}{n} \quad (1)$$

We calculate OQ from the values stored in the corresponding part of the ontology. We calculate an OQ value for each piece of a comment.

Product quality ranking

Informed Recommender ranks a product according to consumer comments for each feature. Owing to the difficulties of quantifying user valuation from texture data, each feature from the comment can only be assigned either “good” or “bad,” calculated as 1 or -1 respectively. The system calculates an FQ score for each feature, which is a function of consumer valuation and OQ.

We define FQ as the quality value for each feature of the product

| |
|--------------------|
| 1. purple>=1 → bd |
| 2. iso>=1 → bd |
| 3. manual → bd |
| 4. problem>=1 → bd |
| 5. battery>=1 → bd |
| 6. not → bd |
| 7. lcd → bd |
| 8. no → bd |
| 9. [TRUE] → ~bd |

Figure 4. The selected rule set to classify sentences into the “bad” category.

| |
|--------------------|
| 1. own → ql |
| 2. bought → ql |
| 3. digital → ql |
| 4. powershot → ql |
| 5. cameras>=1 → ql |
| 6. sony>=1 → ql |
| 7. [TRUE] → ~ql |

Figure 5. Selected rule set to classify sentences into the “quality” category.

in a review and calculate it by multiplying the rating by the customer’s OQ value:

$$FQ_f = r * OQ_i \quad (2)$$

Selecting the relevant opinion and making recommendations

When a user requests an evaluation of a particular product based on certain features, the overall feature quality is calculated from reviews containing the valuation of this feature.

We define OFQ as the global valuation of the feature from all reviews, which is calculated by the average FQ value:

$$OFQ_f = \frac{\sum (Scalingfactor * FQ)}{NumberOfOpinions} \quad (3)$$

Here, we used a scaling factor to make a minor adjustment of a user valuation, which can be set to

$$Scalingfactor = \frac{1}{n} \quad (4)$$

where n is the number of all the features the consumer rated. Each review rated a different number of features so that n could be different. To provide the user with a comprehensive valuation of the product quality in relation to the requested features, we define an *overall assessment* (OA) score, which provides a final score of the product based on each feature’s valuation. It’s the sum of all OFQs (from equation 3) multiplied by the importance index:

$$OA = \sum OFQ * ImportanceIndex \tag{5}$$

The importance index measures the features' different influences on a consumer's decision making. It can be assigned in two ways: according to the importance of the feature in the user request or

Canon PowerShot SD630 (nzjss, 16 Apr 06):
 You cant take a photo unless you are carrying a camera and this is a good one to carry all the time! I have owned most IXUS cameras since the first—Just traded my Ixus 700 on this one (an Ixus 65 but still have an Ixus 40). The huge screen is excellent for framing shots and showing them off. I like the new low shutter speed showing on the screen and the grid lines. The screen is quite scratch resistant (unlike the 40 or 700 which scratch easily). Images are "good" (not stunning) better handheld in low light than the 700 but still blurry for distant features in landscapes. Gives excellent 6x4 prints on Canon CP330. The smaller 5-way selector button is touch sensitive and shows fancy icons on the screen that get bigger when you touch (not push) that side of the button—nice! The Flash is undersized but works well with closer subjects and has good balance for backlit shots. The bigger strap eye on this camera allows you to fit the strap of your choice. I use a black early Ixus strap that works better with my Jenova black leather Ixus belt case. Interesting to see other camera brands copying the Canon zoom button around the shutter release—works well on the Ixus 65/SD630 with the new low profile buttons. The slim mode switch on top is now needs the left hand. This is my "main" camera—and while there are times when I want a longer or wider lens I would not be dragging along the big camera that they are attached to. Recommended—if there was a better camera this size I would have bought it! The Images are not that sharp but better than the 700. No manual ASA select and still no Battery power meter. Not so much room for the right thumb to grip the back but practicing a different finger arrangement holding the camera in the forefinger and thumb with the other fingers clamped on the strap. The Auto-Rotate is such crap—who needs to see a smaller image rather than turn the camera? Mode switch in this position changes easilly when pulling the camera out of it's case.

Figure 6. The review from Digital Photography Review (www.dpreview.com) that we used in our example.

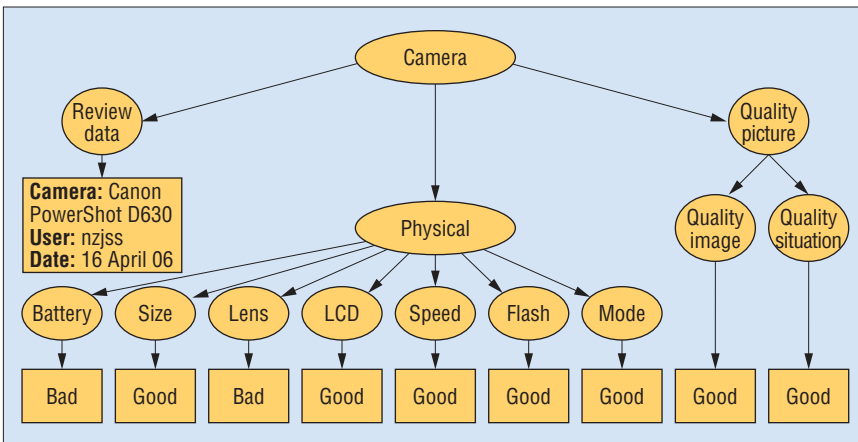


Figure 7. Mapped ontology from a consumer review comment.

according to the frequency with which consumers have rated the feature in their reviews.

Illustrated example

Here we present the steps that Informed Recommender follows to offer a recommendation about a digital camera in response to a user request. Again, we conducted this example using data from the Digital Photography Review. First, we explain how we defined the ontology and how the reviews are mapped onto the ontology.

Representing consumer reviews: The ontology

First of all, we define an ontology—in this case, for the digital-camera domain. We obtained each concept in the ontology by analyzing consumer reviews. Consumers can rate any digital camera on a scale of half a star to four stars and write free-form text reviews.

To construct the ontology, we first listed all possible objects necessary to cover given camera reviews. We determined that such a list should include different digital camera brands such as Canon, Sony, and so on. Furthermore, different cameras can be qualified by features such as size, zoom, lens, and picture quality. The concept "features" represents this information. The reviews can also be qualified by reviewer, depending on whether the opinion comes from a beginner or professional and on the reviewer's level of expertise using digital cameras.

Mapping a review comment into an ontology

Once the ontology has been defined, we must match the information in the review with the ontology. We now show a new mapping to map the information into the ontology. We conducted this example using the review in figure 6. The next sections describe the classification process applied to the new review.

Classifying each sentence into a category. Informed Recommender applies the set of rules we obtained in the previous section to each sentence of the new review to classify it into one category. For example, the system classified the first sentence into the "good" category on the basis of rule 26 in figure 3 and because no "bad" or "quality" rules applied.

Twenty sentences have been classified—eight into the "good" category, three into the "bad" category, and one into the "quality" category. Five are irrelevant (none of the rules has been applied).

Finding the concept represented in the sentence.

For each sentence that's classified as "good" or "bad," the system finds the mapping feature by searching the keywords in the related-words list. For example, in the first sentence, that has been classified as a good opinion. Also, the system has found a word ("carry") that's related to the concept "size." We suppose that the size of the camera is good, so the system assigns the value "good" for the feature "size" in the ontology.

Figure 7 shows the mapping of the new review onto the predefined ontology.

Table 5. Ontology instances mapped from consumer reviews.

| Consumer information | | | | | |
|-------------------------------------|---|---|---|------------------------------|-----------------------------------|
| Consumer | John | Karem | James | Laura | Andy |
| Camera | Sony W70 | Sony W70 | Sony W70 | Olympus Z25 | Canon A630 |
| Opinion quality | | | | | |
| Consumer skill | Beginner | Professional | Beginner | Beginner | Professional |
| Experience using this camera | 2 months | 1 year | 2 weeks | 3 months | 2 months |
| Experience using any digital camera | 4 months | 1 year | 3 weeks | 5 months | 2 years |
| Number of different cameras used | 2 | 1 | 1 | 2 | 3 |
| Product quality | | | | | |
| Features | Size: good Interface: bad Documentation: good Zoom: good | Zoom: good Interface: bad Price: bad Battery: good | Weight: good Documentation: bad Material: bad Sof.: bad Wi-Fi: bad Start-up: bad | Size: good Interface: bad | Size: good Documentation: good |

Computing the recommendation

In this section, we provide detailed calculations of the recommendation in response to a user request. Table 5 shows five ontology instances mapped from review comments, which are used for the recommendation calculations in the example.

Obtaining OQ. OQ is calculated using equation 1. Table 6 presents the weighting value of each variable defined in the equation.

The OQ values for each consumer in table 5 are

$$OQ_{John} = \frac{0.5 + 0.7 + 0.7 + 0.5}{4} = 0.6$$

$$OQ_{Karem} = \frac{0.9 + 0.9 + 0.9 + 0.3}{4} = 0.75$$

$$OQ_{James} = \frac{0.5 + 0.5 + 0.3}{3} = 0.43$$

$$OQ_{Laura} = \frac{0.5 + 0.7 + 0.7 + 0.5}{4} = 0.6$$

$$OQ_{Andy} = \frac{0.9 + 0.7 + 0.9 + 0.7}{4} = 0.8$$

With the calculated values, the best opinion came from Andy: he’s a professional photographer, he has used digital cameras for a longer period of time than the other consumers in the sample, and he has used three different cameras.

As equation 1 explained, the system calculates OQ as the average of the four variables. In the case of missing information on one or more variables, it calculates OQ as the average of the remaining variables. In case no information is available, it sets the OQ as the average of the lowest possible values of all variables, which is 0.35 (= (0.3 + 0.3 + 0.3 + 0.4)/4) according to table 6.

Obtaining the FQ. The system also calculates the FQ value for each feature rated by the consumers. For example, as table 5 shows, John gave the value “good” or “bad” for each feature of the digital cam-

Table 6. Variables representing the consumer’s level of expertise in using a digital camera.

| | Value | Weight | |
|----------------------------|-------------------------------------|-----------------|-----|
| Consumer skill | Beginner | 0.5 | |
| | Advanced | 0.7 | |
| | Professional | 0.9 | |
| Consumer experience | Experience using this camera | Day | 0.3 |
| | | Week | 0.5 |
| | | Month | 0.7 |
| | Experience using any digital camera | Year | 0.9 |
| | | Day | 0.3 |
| | | Week | 0.5 |
| | Number of different cameras used | Month | 0.7 |
| | | Year | 0.9 |
| | | One | 0.3 |
| | | Two | 0.5 |
| | | Three | 0.7 |
| | | More than three | 0.9 |

era Sony W70. As calculated earlier, his OQ value is 0.6. As we described in the “Product quality ranking” section, we calculate FQ for each feature in John’s opinion by assigning the value 1 for “good” and -1 for “bad.” So, for example, his “good” categorization for the Documentation and Zoom features equals 0.6, and his “bad” categorization for the Interface feature is -0.6.

The system applied the same process to all consumers. It calculated OQ and FQ for each review comment offline to quickly respond to the user requests. The system requires users to input the camera

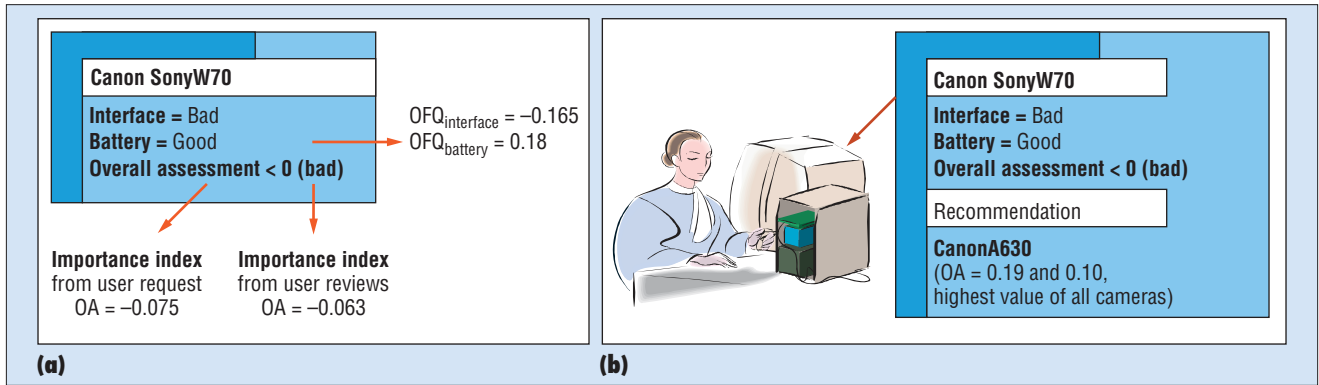


Figure 8. Recommendations: (a) a partial result of a recommendation based on a user request and (b) the final recommendation including both a response to the user request and a recommendation of the best camera according to all user ratings.

model they're interested in and to select the features that they're most concerned with. The features in the selection panel are the same set of features that the ontology covers.

Making a recommendation. Consider this user request: "I would like to know if Sony W70 is a good camera, specifically its interface and battery consumption." The system can identify three keywords ("Sony W70," "interface," and "battery"). First, only the opinions for Sony W70 are selected. Table 5 gives three opinions about the Sony W70: John's, Karem's, and James's. Each feature's OFQ is calculated using equation 3:

$$OFQ_{interface} = \frac{1}{2} \left(\frac{1}{4} * (-0.6) + \frac{1}{4} * (-0.75) \right) = -0.165$$

$$OFQ_{battery} = \frac{1}{4} * 0.75 = 0.18$$

The system obtains the OA on the basis of the two features requested using equation 5. The importance index was calculated in two ways. If the user has expressed that the interface is more important than the battery, the value of 1 is assigned for interface and 0.5 for battery. Using these values, the OA for Sony W70 camera is

$$OA = -0.165 * 1 + 0.18 * 0.5 = -0.075$$

If the user doesn't give a preference, the importance index is calculated on the basis of how frequently the feature has been reviewed:

$$ImportanceIndex = \frac{n}{N} \tag{6}$$

where n is the number of times that the feature appears in the reviews and N is the total number of reviews. Using equation 6, the OA for the Sony W70 camera is

$$ImportanceIndex\ Interface = \frac{6}{10} = 0.6$$

$$ImportanceIndex\ Battery = \frac{2}{10} = 0.2$$

$$OA = -0.165 * 0.6 + -0.18 * 0.2 = -0.063$$

Assigned the value "good" for OA and $OFQ > 0$ and "bad" for OA and $OFQ < 0$, the Sony W70 camera is bad, according to consumers' opinions. Figure 8 shows the response to the user request.

Informed Recommender also recommends the best camera according to the features the user is concerned with. It applies the same process to all other camera reviews and recommends the Canon A630.

Research using reviews in a recommender system is still in its infancy. To the best of our knowledge, this is the first attempt to build a recommender system based on review comments in free-form text. Other researchers have used reviews to give some explanation about a product recommendation.^{3,4} We have proposed a potentially novel approach for retrieving review information.

We drew the following conclusions about the mapping process:

- The comments we used in the example are all for one model of camera (Canon PowerShot SD500). The recall and precision measures could be improved in the classification process by using multiple models.
- The "good" category contains more training data than other categories, so it achieved the best results.
- We couldn't classify some long, complicated sentences into any category. Such sentences should be broken into several short sentences before classification.

Despite these issues, we consider the results we obtained to be good because we can accurately map a large portion of a review into the predefined ontology. Implementing this method lets our system use valuable textual information to make recommendations. In addition, using this textual information lets us obtain ratings of products that haven't been rated by a sufficient number of consumers. This helps our system overcome the cold-start problem, which challenges collaborative-filtering techniques. In future work, we intend to evaluate our approach with the intended consumer groups in a real-world application. ■

Acknowledgments

This research was partially funded by Agaur grant # 2005 BE 00333 (Agèn-

The Authors



Silvana Aciar is a PhD student in information technology in the University of Girona's Department of Electronics, Computer Science, and Automatic Control. She's also a member of the Agents Research Lab in the Institute of Informatics and Applications at the UdG. Her research interests include multiagent systems, recommender systems, and ontologies. She received her MSc in information technology from the UdG. Contact

her at the Dept. of Electronics, Computer Eng., and Automatics, Univ. of Girona, Campus Montilivi, Bldg. P4, 17071 Girona, Spain; saciar@eia.udg.es.



Debbie Zhang is a research fellow in the E-Markets Research Group, Faculty of IT at the University of Technology, Sydney. Her work includes developing smart text-mining agents to extract unstructured text and numerical data from the Internet and apply machine-learning techniques to verify, combine, filter, and condense the extracted information for e-negotiators. Her research interests include text mining, modeling of financial market behaviors, and discrete-element modeling. She received her PhD from the University of Queensland, Australia. Contact her at the Faculty of Information Technology, Univ. of Technology, Sydney, PO Box 123 Broadway, NSW 2007, Australia; debbiez@it.uts.edu.au.

cia at the Dept. of Electronics, Computer Eng., and Automatics, Univ. of Girona, Campus Montilivi, Bldg. P4, 17071 Girona, Spain; saciar@eia.udg.es.



Simeon Simoff is a professor of information technology in the Department of Computer Systems, Faculty of IT at the University of Technology, Sydney and is the leader of the university's E-Markets Research Group. His research interests include information-rich trading environments and technologies that facilitate extraction, synthesis, and delivery of condensed information from them, including network data mining, text

analytics, multimedia, and visual data mining. He's also exploring the use of context networks that are established by and surround the electronic trading processes. He received his PhD in computer science from the Moscow Power Engineering Institute. Contact him at the Faculty of Information Technology, Univ. of Technology, Sydney, PO Box 123 Broadway, NSW 2007, Australia; simeon@it.uts.edu.au.



John Debenham is a professor emeritus in the Department of Software Engineering, Faculty of IT at the University of Technology, Sydney and a member of the university's E-Markets Research Group. His recent work has focused on multiagent systems, e-negotiation, and argumentation agents. He's particularly interested in the design of agents that can exploit the information-rich environment of electronic markets. He received his PhD in

computer science from the University of Sydney. Contact him at the Faculty of Information Technology, Univ. of Technology, Sydney, PO Box 123 Broadway, NSW 2007, Australia; debenham@it.uts.edu.au.

cia de Gestió d' Ajuts Universitaris i de Recerca, Catalunya) and the European Union project 34744 ONE: Open Negotiation Environment, FP6-2005-IST-5, ICT-for Networked Businesses. Silvana Aciar especially acknowledges the support and valuable comments of her PhD supervisors, Josep Lluís de la Rosa i Esteva and Josefina López Herrera.

References

1. D. Zhang and S. Simoff, "Informing the Curious Negotiator: Automatic News Extraction from the Internet," *Data Mining: Theory, Methodology, Techniques, and Applications*, G. Williams and S. Simoff, eds., Springer, 2006, pp. 176–191.
2. S.M. Weiss et al., *Text Mining: Predictive Methods for Analyzing Unstructured Information*, Springer, 2004.
3. G. Adomavicius, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. Knowledge and Data Eng.*, vol 17, no. 6, 2005, pp. 734–749.
4. R. Wietsma and F. Ricci, "Product Reviews in Mobile Decision Aid Systems," *Proc. Workshop Pervasive Mobile Interaction Devices*, in conjunction with Pervasive 2005, LMU Munich, 2005, pp. 15–18.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.

IEEE computer society

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field. Visit our Web site at www.computer.org.

EXECUTIVE COMMITTEE

President: Michael R. Williams*
President-Elect: Rangachar Kasturi,* **Past President:** Deborah M. Cooper,* **VP, Conferences and Tutorials:** Susan K. (Kathy) Land (1ST VP);* **VP, Electronic Products and Services:** Sorel Reisman (2ND VP);* **VP, Chapters Activities:** Antonio Doria;* **VP, Educational Activities:** Stephen B. Seidman;† **VP, Publications:** Jon G. Rokne;† **VP, Standards Activities:** John Walz;† **VP, Technical Activities:** Stephanie M. White;* **Secretary:** Christina M. Schober,* **Treasurer:** Michel Israel;†
2006–2007 IEEE Division V Director: Oscar N. Garcia;†
2007–2008 IEEE Division VIII Director: Thomas W. Williams;† **2007 IEEE Division V Director-Elect:** Deborah M. Cooper;* **Computer Editor in Chief:** Carl K. Chang†

* voting member of the Board of Governors
 † nonvoting member of the Board of Governors

BOARD OF GOVERNORS

Term Expiring 2007: Jean M. Bacon, George V. Cybenko, Antonio Doria, Richard A. Kemmerer, Itaru Mimura, Brian M. O'Connell, Christina M. Schober
Term Expiring 2008: Richard H. Eckhouse, James D. Isaak, James W. Moore, Gary McGraw, Robert H. Sloan, Makoto Takizawa, Stephanie M. White
Term Expiring 2009: Van L. Eden, Robert Dupuis, Frank E. Ferrante, Roger U. Fujii, Anne Quiroz Gates, Juan E. Gilbert, Don F. Shafer

Next Board Meeting: 9 Nov. 2007, Cancún, Mexico



revised 1 May 2007

EXECUTIVE STAFF

Associate Executive Director: Anne Marie Kelly; **Publisher:** Angela R. Burgess; **Associate Publisher:** Dick J. Price; **Director, Administration:** Violet S. Doan; **Director, Finance and Accounting:** John Miller

COMPUTER SOCIETY OFFICES

Washington Office: 1730 Massachusetts Ave. NW, Washington, DC 20036-1992 • Phone: +1 202 371 0101 • Fax: +1 202 728 9614 • Email: hq.ofc@computer.org
Los Alamitos Office: 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314 • Phone: +1 714 821 8380 • Email: help@computer.org • Membership and Publication Orders: • Phone: +1 800 272 6657 • Fax: +1 714 821 4641 • Email: help@computer.org
Asia/Pacific Office: Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan
 Phone: +81 3 3408 3118 • Fax: +81 3 3408 3553
 Email: tokyo.ofc@computer.org

IEEE OFFICERS

President: Leah H. Jamieson; **President-Elect:** Lewis Terman; **Past President:** Michael R. Lightner; **Executive Director & COO:** Jeffrey W. Raines; **Secretary:** Celia Desmond; **Treasurer:** David Green; **VP, Educational Activities:** Moshe Kam; **VP, Publication Services and Products:** John Baillieul; **VP, Regional Activities:** Pedro Ray; **President, Standards Association:** George W. Arnold; **VP, Technical Activities:** Peter Staecker; **IEEE Division V Director:** Oscar N. Garcia; **IEEE Division VIII Director:** Thomas W. Williams; **President, IEEE-USA:** John W. Meredith, P.E.