DiG: A Task-based Approach to Product Search

Scott Carter[1] **Francine Chen**[1] **Aditi S. Muralidharan**[2] **Jeremy Pickens**[1*]
[1]FX Palo Alto Laboratory, Inc.
3400 Hillview Ave, Bldg 4
Palo Alto, CA 94304
[2]Department of Computer Science
UC Berkeley
Berkeley, CA 94720
carter@fxpal.com, chen@fxpal.com, aditi@cs.berkeley.edu, jeremy@fxpal.com

ABSTRACT
While there are many commercial systems to help people browse and compare products, these interfaces are typically *product* centric. To help users identify products that match their needs more efficiently, we instead focus on building a *task* centric interface and system. Based on answers to initial questions about the situations in which they expect to use the product, the interface identifies products that match their needs, and exposes high-level product features related to their tasks, as well as low-level information including customer reviews and product specifications. We developed semi-automatic methods to extract the high-level features used by the system from online product data. These methods identify and group product features, mine and summarize opinions about those features, and identify product uses. User studies verified our focus on high-level features for browsing products and low-level features and specifications for comparing products.

Author Keywords
Interactive interfaces, text mining, products

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: Miscellaneous

General Terms
Human Factors, Algorithms

INTRODUCTION
Websites like Amazon and CNET collect reviews of products and display them in search results and product description pages. Amazon also recommends products based on analysis of user behavior and buying patterns. To help buyers evaluate products, Amazon shows ratings, reviews for each camera, and lists of attributes that other customers rated.

While useful, this information can be overwhelming to sort through. Most interfaces provide only the most basic means of browsing a product collection, usually by selecting from well-known technical specifications. In addition, standard interfaces fail to take into account the things users will be doing while they use a product, which may affect the importance of different technical specifications. They also ignore the experience of other users, which may be at odds with specifications (e.g., a camera's close-up mode is useless if it is impossible to find). It would be useful instead if product search interfaces focused on what users want to do with a product rather than low-level specifications.

To that end, we built and tested a task-based product search kiosk that could appear at the front of a store. This situation requires an interface that allows people to quickly browse through a collection of products to find the few they are most interested in, compare them, and be able to take product information with them as they move from the kiosk into the store. Sample screenshots from our interface, DiG, are shown in Figure 1.

A core contribution of this work is the semi-automatic extraction of product uses to support task-based search. Other types of data are mined to provide supporting information for the interface, as well as to provide greater coverage of the terminology and interests of the bulk of users.

A user searching for products with the interface first answers a series of questions that help her select activities she anticipates for a product. In the background, the system accesses the product use data to set feature weights that match the selected activities. The user can optionally manipulate feature weights directly. The interface also includes a product detail view that links uses and features back to source material from the review. A comparison interface allows users to contrast selected products. Finally, 2D barcodes allow users to download product data to their mobile device when they are ready to see the product in the store.

RELATED WORK
As outlined below, we adapt many existing methods for our scenario, performing most of the processing automatically, with a small amount of manual review to refine the final results. We have not found related work on identifying uses.

Product Feature Extraction: There are two primary extraction methods: those that use pattern-based information extraction, and those that use term-frequency statistics. In our pipeline, we use a high-precision, pattern-based technique developed by Popescu and Etzioni [11].

Term Similarity and Grouping Product Features: A number of methods have been proposed for grouping words, var-

*Jeremy is now at Catalyst and can be contacted at jeremy@catalystsecure.com.

IUI'11, February 13–16, 2011, Palo Alto, California, USA.

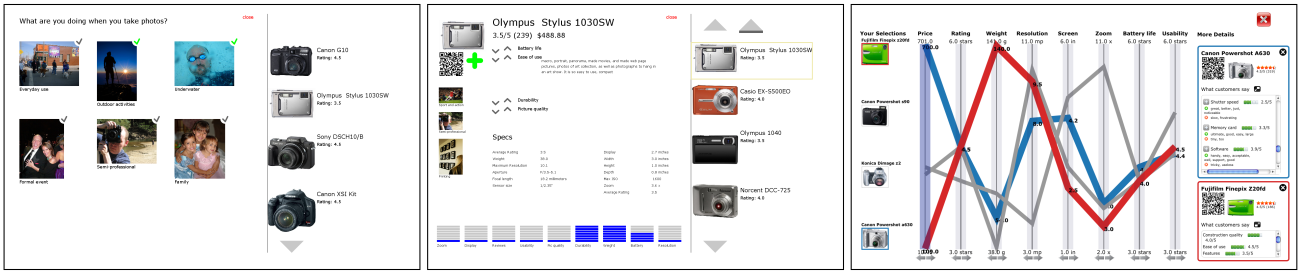

Figure 1. The DiG interface. (left) Users specify activities for which they plan to use the product. (middle) Clicking on a product icon shows details, including the activities with which it is associated, a 2D barcode, and a button to add it to the user's current collection. Users browse products by manipulating feature or specification weights (horizonal bars) directly. (right) Users can launch a parallel coordinates interface to compare products.

iously based on WordNet by Liu et al. [7], mutual information of n-grams by Brown et al. [1], and syntactic relations by Lin [6]. For computing noun-sequence similarity, we use a simplified version of Lin's approach [6].

Opinion Mining: Opinion mining can refer to tasks of various levels of granularity. We operate on the within-sentence scale, and take the approach of identifying all the "opinion words" that apply to a feature, and then aggregating their individual polarities to give a score. In our work, we follow Popescu and Etzioni [11] who use dependency relations to extract adjectives associated with noun phrases. We use Turney's web-PMI method [12] to classify adjectives as positive or negative.

Opinion Visualization: Jodange (http://jodange.com) is a commercial site that extracts opinions from articles and media and presents the opinions for mining. Oelke [10] visualizes the opinions of products in a table comparing products vs. product features. In contrast to these visualizations, we embed the opinions with other product information. Carenini and Rizoli [2] describe a multimedia interface for comparing digital cameras. Their interface uses multiple histograms to show how opinions on each camera feature vary, and can only compare one pair of cameras at a time with no support for selecting a subset of attributes, filtering out value ranges, or seeing more detailed information about a camera on demand. Another similar interface is Opinion Observer by Liu, Hu, and Cheng [7], which uses a bar-graph-like display to show how opinions on camera attributes vary and is not interactive. On the other hand, our approach focuses on interactive browsing and rich product comparisons.

PRODUCT DATA

We designed our data gathering and analysis pipeline to support a variety of products. However, in this paper we focus on digital cameras. The interface uses several types of data:

Standard Product Specifications: e.g., weight and resolution.
Product Features: Extracted from reviews. These go beyond technical specifications, e.g., whether the face-detection works, or whether a camera is durable.
Product Attributes: Users explicitly select a rating for each attribute e.g., battery life, or ease of use.
Product Uses: Derived from reviews, and include: (1) the types of tasks people are doing when using the product (2) how they apply the product in that task (3) what tasks the product is used for. A set of uses is displayed in Table 2.
Activities: A meta-type linking uses to other data types. An activity is simply a use and the sets of specifications, features, and attributes associated with it. A set of activities is displayed on the left in Figure 1.

DATA EXTRACTION AND ANALYSIS

Technical specifications are available via open APIs and reviews (for cameras, we use Amazon's Product Advertising API). In this section we describe our pipeline for analyzing product reviews to extract product data.

Preprocessing

We crawled Amazon's camera review web pages to download the review data and user ratings (i.e., attributes), creating a corpus containing 168,638 reviews for 3,700 cameras. Next, the text in each review is split into sentences using NLTK [8], resulting in 1,155,531 sentences. Each sentence is then parsed using the Stanford Parser [5] for syntactic information. As a by-product, each word is also tagged with part-of-speech to simplify identification of noun sequences (i.e., one or more adjacent nouns).

Reliable Product Feature Extraction

Product features are parts and properties of a product that customers explicitly mention in reviews. After following Popescu and Etzioni [11] to identify candidate features, we parse the sentences in which the candidates occur and keep only noun sequences. Then an SVM [3] is used to select reliable product features. The SVM features are based on web-PMI [12] between a candidate feature, *<candidate>*, (e.g., "face detection") and discriminator phrases. Our discriminator phrases include *<product> features <candidate>* and *<product> has <candidate>*, where *<product>* is the type of the product, i.e. "camera". Counts for computing PMI are obtained using an API to query Yahoo!. This gives a set of *reliable product features*.

Grouping Product Features

We cluster reliable product features to capture various ways that reviewers refer to the same concept. Although we could directly cluster the reliable features, clustering frequent noun sequences and using the reliable features to "filter" the noun

Table 1. The top 10 automatically-produced camera feature clusters.

camera, body
photos, pics, pictures, shots
battery life, photo quality, quality, picture quality, image quality
zooms, zoom
screen, lcd, view screen, lcd screen, lcd display, display
lens, lenses
image, shot, picture
bang, deal, value, job
settings, setting
aa batteries, batteries

Table 2. The top 10 "camera uses" and the two most frequent phrases.

light	in low light	in bright light
people	of people	with people
conditions	in low light conditions	in all conditions
time	at a time	at one time
kids	of the kids	of kids
family	of family	of the family
friends	of friends	of family and friends
computer	on the computer	on computer
price	for the price	at a great price
flowers	of flowers	of the flowers

sequences in the clusters gave us better results. We use a modification of Lin's method [6] to compute similarity, considering noun sequences instead of words, and using only adjective modifier relations rather than all dependency relations. The similarity between all pairs of the most frequent noun sequences (we ignored sequences occurring fewer than 50 times) is used for clustering the phrases using complete-linkage agglomerative clustering to keep the phrases compact. To split the hierarchical tree into clusters, a threshold is manually set after examining a visualization of the cluster tree. The top automatically-produced camera feature clusters are shown in Table 1. The resulting list of clusters is manually filtered to remove extraneous clusters such as 'bang, deal, value, job'.

Opinion Mining

To extract opinions about product features, we first classify review sentences as either objective or subjective, then identify and classify opinion words, and finally aggregate the opinion-word polarities to get an opinion score. Subjective sentences are identified using a classifier trained on Pang and Lee's Movie Database labeled for Subjectivity[1]. We take opinion words to be the adjectives (and adverbs) that refer to product features in subjective sentences through *amod*, *nsubj* or *advmod* dependency relations. If negations modify the adjectives, we mark them as such. To classify the adjectives as positive or negative, we use Turney's web-PMI method [12]. We train an SVM [3] with web-PMI features for the labeled data from OpinionFinder's subjectivity lexicon [9], and use the trained model to classify adjective polarity.

We aggregate the positive and negative opinions about each feature to provide an opinion polarity score that is used when ranking products. To handle features that are mentioned very few times, a beta-binomial model is used to smooth the opinion score, where the score for observing a positive-polarity adjective is generated by a beta distribution and the correct classification is generated by a binomial distribution.

A few sentences are automatically selected to represent opinions about a selected sample of camera features. For a given camera, the set of feature clusters are scored by frequency and PMI. Then for each feature cluster, ordered by score, *sentences* in the camera reviews containing the feature are scored and ordered, until a preset maximum number of sentences are identified. The sentence score favors frequently mentioned product features, adjective/noun-sequence pairs in a camera's reviews, and sentences where the PMI between the adjective(s) and noun-sequence is high.

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data/

Extracting Product Uses

Camera uses are found by searching for patterns representing common expressions that indicate a use. For this, we use noun sequences associated with picture, which include {picture, pictures, photo, photos, pic, pics} in the pattern *<picture term> <prepositional phrase> <noun phrase>*.

The matching phrases are filtered to remove product features, numerical values, and and stopwords, such as 'anything'. The remaining phrases are then grouped by combining all phrases with the same last word. For example, 'zoo', 'Washington Zoo', and 'San Diego Zoo' are all grouped under 'zoo'. The groups are then sorted by frequency for presentation. The resulting list is short enough for manual inspection and cleanup. The top 10 automatically-identified uses, along with the two most frequent phrases associated with each use, are shown in Table 2.

Activity Creation

The final step is to create an activity, which involves linking specifications, features, and attributes to each use. In our current system we construct these links manually.

RANKING

We currently use a ranking algorithm that orders products according to user-specified weights. A simple scale selector (Figure 1 (middle)) shows the current importance of a specification, feature, or attribute. Each weight is then applied to a normalized value for the specification, feature, or attribute for each camera.

INTERFACE

Our ranking system depends on user-specified weights of camera specifications and features, and the interface allows users to adjust weights both indirectly and directly. Users can specify weights indirectly by selecting activities they want to do with the product (see Figure 1 (left)). Activities are organized manually into groups that address a specific question. Currently we have organized them into three groups: what the user is doing when the photo is taken (e.g., hiking), what the user is taking pictures of (e.g., mountain scenery), and what the user intends to do with the photos (e.g., put them in a scrapbook). Since activities map uses to specifications, features, and attributes, selecting activities implicitly adjusts weights. Users can also manipulate weights directly by selecting different levels for each facet (see Figure 1 (middle)). Note that the approach of manipulating weights of features is relatively unusual – most search interfaces involve selecting facets, or set ranges of target values. We focus on weights rather than facets because they do not require detailed technical knowledge (e.g. users can say how much they care about camera resolution, rather than specifying resolution exactly, which would require users to know the state of the art for that particular feature).

Detail views and pivoting

Deciding among products is never as simple as choosing from a list. A user will normally wish to explore the product and its features in more detail. The interface includes a detailed view of each camera showing not only all of its specifications but also highlights from reviews about specific product features (Figure 1 (middle)). These highlights were automatically extracted to summarize important issues from reviews. These highlights are linked to actual reviews so that users can see the context of the reviewer's comments. Users can also click on widgets next to review highlights to manipulate the features to which that highlight is linked. In this way, the interface creates connections between abstracted product features and review details.

Comparison

It is important that users be able to collect products along the way and compare them. To support this, we built a parallel coordinates interface [4] (Figure 1 (right)). Unlike a classic parallel coordinates display, there are only a few items, so we allow users to click on a camera's plot line to see more details. A display box appears on the right, showing the rating, 2D barcode, and opinion scores for product features (extracted from review sentences by our processing pipeline).

STUDIES

Our early prototypes had widgets that gave users more control over selecting ranges and weights, but pilot users made it clear that they were far too difficult to use. After redesigning based on this feedback, we ran a user evaluation with six users (two female). During the evaluation we first described that the main goal of the interface is to support product browsing on a public display in a storefront. We then showed the users all of the core components of the UI and had them become comfortable clicking on interactive content. Finally, we began the study, which consisted of three tasks: 1) Buying a gift for a friend with a variety of parameters (such as price and sharing features) 2) An extension of the first task that added another constraint (the product should also be easy to use) 3) Finding cameras that best match an activity of the users' choice (watching a sporting event or attending a formal concert), comparing the cameras using the comparison visualization, reading camera reviews, and downloading the best ones to a mobile device provided to the user. When the users completed the study, we asked them a series of Likert-scale and open-response questions.

While users noted some usability issues, overall they found the interface easy to use. They all rated selecting activities and downloading information to the mobile device as less than average difficulty, and all but one found comparing cameras and adjusting features less than average difficulty. As expected, users thought that the initial activity selection was useful for a storefront. One commented that if she were "making a purchase decision in the store this would be useful ...". Additionally, users thought that the interface might be more useful *before* going to the store. One mentioned that the interface "could help me select candidate for online or in-store purchase ahead of time" and another said that "it would be very useful [ahead of time] if it was associated with the store ...'. This suggests that the next iteration of the interface may need to focus on features that support web browsing, such as "sav[ing] results to come back to."

CONCLUSIONS AND FUTURE WORK

We have described a system that helps people identify products that meet their needs with a minimum of technical knowledge. A variety of data is mined semi-automatically to identify different product features and popular product uses, and the mined results are used to inform the user interface.

One focus of future work is to infer the camera weights used in the ranking algorithm from user activity and interests. We are also exploring methods for further automating the selection of uses and the creation of activities. In the interface, we would like to reintroduce redesigned versions of some advanced features that tested poorly in pilots, such as selectable feature hierarchies. Finally, we plan to demonstrate our system with other products.

REFERENCES

1. P. Brown, P. deSouza, R. Mercer, V. Della Pietra, and J. Lai. 1992. Class-based N-Gram Models of Natural Language. Computational Linguistics. 18:4, pp. 467–479.
2. G. Carenini and L. Rizoli. 2009. A Multimedia Interface for Facilitating Comparisons of Opinions. Proc. IUI. pp. 325–334.
3. C. Chang and C. Lin, 2001. LIBSVM: A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/c̃jlin/libsvm.
4. A. Inselberg and B. Dimsdale. 1990. Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry. Proc. Conf. on Visualization. pp. 361–378.
5. D. Klein and C. D. Manning. 2003. Accurate Unlexicalized Parsing. Proc. ACL. pp. 423-430. http://nlp.stanford.edu/software/lex-parser.shtml.
6. D. Lin. 1998. Automatic Retrieval and Clustering of Similar Words. Computational Linguistics. pp. 768–774.
7. B. Liu, M. Hu, and J. Cheng. 2005. Opinion Observer: Analyzing and Comparing Opinions on the Web. Proc. WWW. pp. 342–351.
8. NLTK. Available from http://www.nltk.org/. August 2010.
9. OpinionFinder Subjectivity Lexicon. Available from http://www.cs.pitt.edu/mpqa/. August 2010.
10. D. Oelke, M. Hao, C. Rohrdantz, D. Keim, U. Dayal, L.-E. Haug, and H. Janetzko. 2009 Visual Opinion Analysis of Customer Feedback Data. Proc. IEEE Sympos. VAST. pp. 187–194.
11. A.M. Popescu and O. Etzioni. 2005. Extracting Product Features and Opinions from Reviews. Proc. HLT-EMNLP. pp. 339–346.
12. P.D. Turney. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Proc. ACL. pp. 417–424.