Extracting Product Comparisons from Discussion Boards

Ronen Feldman, Moshe Fresko,
Jacob Goldenberg
School of Business Administration
The Hebrew University of Jerusalem
Jerusalem, ISRAEL
{ronen.feldman, freskom, msgolden}
@huji.ac.il

Oded Netzer
Columbia Business School
Columbia University
on2110@columbia.edu

Lyle Ungar
Computer and Information Science
University of Pennsylvania
ungar@cis.upenn.edu

Abstract

In recent years, product discussion forums have become a rich environment in which consumers and potential adopters exchange views and information. Researchers and practitioners are starting to extract user sentiment about products from user product reviews. Users often compare different products, stating which they like better and why. Extracting information about product comparisons offers a number of challenges; recognizing and normalizing entities (products) in the informal language of blogs and discussion groups require different techniques than those used for entity extraction in the more formal text of newspapers and scientific articles. We present a case study in extracting information about comparisons between running shoes and between cars, describe an effective methodology, and show how it produces insight into how consumers view the running shoe and car markets.

1. Introduction

There is increasing recognition that product reviews by consumers can provide important insight into how they view products, and that automated text analysis methods can be fruitfully used to extract such information. For example, the rapidly growing field of sentiment analysis looks to extract how authors feel about different products [3-7]. Such work has tended to look at single products, in spite of the fact that much shopping, and hence much marketing, is based on product comparisons. This paper describes a methodology for automatically analyzing products and comparisons between them. Given a sentence such as "The Inspire is more flexible than the Alchemy," we automatically extract the products (the *Inspire* and *Alchem*y models of shoes) and what attributes they are compared on (*flexibility*). Our goal is to automatically create meaningful profiles for each product. Including how it compares to other products. What products do people think are similar, better or worse, and in what attributes?

The first step in *comparative sentiment analysis* from (answering questions such as "How do consumers feel about this product compared to that product?") is extracting from discussion boards the co-occurring product names and other frequently occurring terms and the phrases they occur in. The extracted phrases provide snippets of text, similar to those in product reviews such as those provided by *Zagat's*. The extracted products and attributes can also be used to construct product comparison tables or graphs, which can be used either by consumers or by marketing managers.

Even noting what pairs of products are mentioned together gives valuable insight into the structure of the market and how consumers view products. Questions can be addressed such as: "Which are the 'reference products' against which others are compared?" We build on work extracting entities from text and visualizing the co-occurrence of these entities in the same sentence or paragraph. Named entity recognition methods have been used to find pairs of companies or genes that are mentioned together, sometimes in the context of a word such a "merger." These relationships are then displayed as graphs. Such work has been directed at corpora such as newswire articles or scientific abstracts. We extend these methods to the more informal styles common in consumer product reviews and blogs, and extract the type and sentiment of the relationship is well as simply co-occurrence. We locate snippets such as

... *Alchemy* has more control than the *Inspire*

identify the brands (shown here in italics), and extract the relations between them.

Comments and reviews by consumers differ in many ways from news and scientific articles. In the more informal language of blogs and newsgroups, "articles" are often quite short, and need to be interpreted in the context of the thread of discussion that they are part of. Questions

1550-4786/07 $25.00 © 2007 IEEE
DOI 10.1109/ICDM.2007.27

are common, and synonyms and abbreviations may be used without being defined. The texts do not necessarily confirm to grammatical rules and conventions. For example, “Like em better than the NB900 or NB901.” Some comparisons include properties such as stability, flexibility or price, others just mention or ask about similarity (e.g., “… NB 831's so maybe those are kind of similar to the Nike Free …”). Capitalization and punctuation are often non-standard, and pictures, references, and quotes, can be included.

This paper describes an implemented system that extracts product mentions and comparisons from user discussion forums. The following sections describe the underlying methodology, and then show results obtained from a set of 19,410 consumer-written reviews of running shoes extracted from RunnersWorld[1] Magazine Discussions (→ Shoes & Gear → Shoes) and from a set of 868,174 reviews of cars, extracted from Edmund's Car[2] Space (→ Forums → Sedans) .

2. Methodology for Analyzing Messages

Extracting structured product comparison data from forum messages requires several processing steps: **(1) Downloading:** The html-pages are downloaded from a given forum site **(2) Cleaning:** Html-like tags and non-textual information like images, commercials, etc. are cleaned from the downloaded files. **(3) Information Extraction:** Products and product attributes are extracted from the messages. **(4) Comparison Analysis**: Two forms of product comparisons are computed. First, we generate graphs of which products are mentioned together; this gives an overview of the overall market structure. Then we drill down and extract snippets in which products are compared, and look at what terms are used in the comparisons, and the direction of the comparison

The first non-trivial step is the extraction of the products and the terms used to describe them.

2.1. Information Extraction

After downloading and cleaning, a slightly longer than average message from "Runner's World” looks like:

```
MessageId: 78
ThreadId: 14
ThreadTitle: Brooks Running/Racing Shoes
ThreadUserName: trp
MessageUserName: Al The Penguin
MessageUserLocation: Lincoln, Nebraska
MessageUserRegDate: August 28, 2006
MessageDate: February 13, 2007 01:53 PM
```

```
I have some experience with the Burn, and I
race in the T4 Racers.  Both of them have a
narrow heel and are slightly wider in the
forefoot. The most noticeable thing about the
T4s (for most people) is the arch.  It has
never bothered me, but some people are really
annoyed by it.  You can cut the arch out of the
insole if it bothers you. The Burn's arch is
not as pronounced.
```

We extract the brand names (the 20 shoe or 28 car companies), the model names, and some common terms (mostly noun-phrases and adjectives) used to describe them. Brand names were found to be relatively easy to extract – one can just string-match on the names – while model names showed significant variation and ambiguity.

Extraction by string matching on names was not suitable for extracting models because of the variations in the writing styles. Instead, a set of regular expressions was developed for recognizing models from the model-names appearing in these informal texts. As an example, here are the terms that appear referring to the model "Mizuno Wave Alchemy" with various model numbers. Note the highly erratic capitalization and the elision of various words or spaces.

- `"Mizuno Wave Alchemy 5", "Alchemy 5", "Mizuno Alch.5", "Mizuno Alchemy V", "Mizuno Alchemy 5", "Alchemy V", "mizuno wave alchemy 5", "Alchemy5"`

Although hand-crafting extraction rules is out of fashion, we believe that there are cases, such as these informal product reviews, when it is substantially faster to build rules by hand than to label data, write feature extractors, and train models. This is particularly true since we want not only to tag terms as products, but also to resolve which product is being referred to.

The key to constructing such entity extraction tools is to note that lists of product brands and models are easy to come by on the web; the trick is to generalize the formal product names to the many variants used. Terms referring to products generally consist of a subset of the brand name, model name, and model number. Any combination of the three components can be kept or dropped. Different delimiters (e.g. space, hyphen, or nothing) can also be used between them. Brand names are mostly clean string matches to a list, while model names are often abbreviated (or misspelled). Numbers can be Roman or Arabic. Thus, the full name "Asics GT 2110" could be represented (among other examples) by shortened versions "Asics.?2110", "GT[\\-]+2110", and "2110".

Thus, the terms listed above for the "Mizuno Wave Alchemy" can mostly be extracted by the following the regular expression (ignoring case):

```
(?:Mizuno +)?(?:Wave +)?Alch(emy|\.)
(?: ?:VIII|VII|VI|V|\d)
```

The actual recognition of brand names is slightly more complex, as it needs to handle spelling errors (e.g., “Achemy”), and to account for a number of variations:

[1] http://forums.runnersworld.com/eve/forums/a/frm/f/687106477/p[1]

[2] http://townhall-talk.edmunds.com/WebX/.ee9e22f/

1. The numerical parts of the models may be written in different formats: with digits ("Alchemy 5"), with roman numbers ("Alchemy V"), or even sometimes words ("alchemy five"). Symbols other then digits can occur inside numbers (Shox 2:45).
2. The names often have spelling errors (Glycerin, Glycerine, Gylcerin), are represented by abbreviations (Alch.=Alchemy, Prov.=Providence), or by their initials (New Balance 1222= NB1222, Trail Attack 2=TA2).
3. Sometimes only the numbers are written but not the actual model name, since it can be understood for a reader from the context:
   ```
   Asics Speedstar (both I and II) ...
   I love the I and II's and can't wait for the
   III's. The Nike Zoom Elites ...
   ```
4. Some models can be referred with their numbers only.

Matching is handled in two stages. First there is a tagging stage that works on basis of regular expressions and a list of search strings. This stage is greedy and tries to match as many terms as possible, giving high recall. After that, a second stage removes or modifies the falsely extracted entities in order to improve the precision. For example " <Model>1100</Model> km" will be dropped from Models, and "…the <Model>trail attack I</Model> had…" will be modified to "…the <Model>trail attack</Model> I had…", because for clarity of presentation in the analyses presented below we collapse models that differ only in the final number. Based on a random sample of 500 messages and manual evaluation of the results we achieved recall of 89.4% and precision of 96.7% leading to F1 of 92.9%.

One could, of course, take the initial models and features that we have extracted, and refine them in an unsupervised or semi-supervised fashion in a way similar to that described in ([3, 9]). In future work, we hope to see how well one can do starting with a list of products, a model structure, and a text corpus containing unlabeled product mentions.

2.2. Co-occurrence Analysis

After information extraction, the records are divided into chunks at three levels: threads, messages, and sentences. Threads often contain hundreds of messages, while messages are short, often with only one or a few sentences or sentence fragments. Each record includes the terms (Brands, Models, and Terms) in that chunk.

The co-occurrence analysis finds all frequent co-occurrences of pairs or triples of Model, Brand, and (other) terms. Co-occurrence graphs (presented below) are then derived by taking all interactions where pairs of items show up more than a specified number of times. Using either brands or models with different thresholds allows one to look at coarser or more fine-grained comparisons. In other work, not shown here, we have also extracted association rules and created graphs based on the support and lift of the extracted rules.

3. Snippet extraction

It is often useful to look at terms other than product mentions, to see what terms show up most often in connection with a product or pair of products and what attributes consumers are using when they compare products. A term recognizer was built by using a CRF model trained on the CoNLL-2000 shared task corpus [10]. It uses two consecutive CRF models - one for part of speech tagging and another for chunking. After chunking, there is filtering step, which retains only noun phrases that contain actual nouns (and not, pronouns, for instance).

The terms extracted for shoes fall into several categories: There are sentiment terms (good, love, great, nice), attributes of the shoe (stability, cushion, feel, trail, support, fit, control, wide, weight, light, road), including parts of the shoe (forefoot, heal, arch) and comparatives (lighter, heavier, softer). One can see what attributes of shoes consumers are discussing by looking at what terms occur most frequently with each shoe type. For example, *Saucony Grid Tangent* is strongly associated with *lightweight*, *Brooks Adrenaline* is tied to *stability* and *Adidas Supernova* to *trail*.

The presence of comparatives among the attributes suggests that one might be able to extract *relative* sentiment between pairs of shoes, rather than just absolute sentiment. A fairly simple pattern match, looking for patterns of the form *.?er than* finds many useful comparisons (“better than”, “firmer than”, “lighter than”, “softer than”, etc.), along with a few false matches ("other than", "rather than", "more rubber than"). We describe below how we do a better extraction using a part of speech tagger to identify comparative adjectives.

3.1. Basic description

The snippet extraction component takes as input a large set of sentences in which the relevant product models are labeled. The output is a set of *snippets* – small sentence fragments, each containing a description of opinion, either *factual* (e.g., “Model *A*’s *X* is *Y*”, or “Model *A* has *X*), *sentiment-relating* (e.g. “Model *A* is good”, or “I like model *A*’s *X*) or *comparative* (e.g., “Model *A* is better than model *B*”, or more generally, “Model *A*’s *X* is better than model *B*”).

Snippet extraction is done using the following five steps:

3.1.1. Preprocessing: The texts are tagged with parts of speech (PoS) and chunked into noun phrases using a CRF-based tagger and chunker, both trained on CoNLL-2000 shared task training set. During preprocessing we also find lists of models (“Models *X*, *Y*, and *Z*”) and convert them into a single term, so the system can more easily extract opinions expressed about multiple models simultaneously.

3.1.2. Pattern extraction: We generate the set of surface patterns by a suitable modification of the Apriori association mining algorithm. Each such pattern is a sequence of tokens including the special slot-mark tokens for product names and (optionally) *skips*, which indicate gaps in the pattern [8]. First we extract all sequences (without gaps) of tokens, PoS tags, and NP chunks that appear in the set of all sentences with frequency greater than a given minimal support value. Then we mine the sentences (as ordered sets of such sequences) for frequent itemsets. The result is the set of all sufficiently frequent surface patterns.

3.1.3. Pattern filtering: In order to reduce the number of irrelevant patterns, we keep only patterns that contain specified words or parts of speech. Also, all of the patterns must start with a model. If a pattern contains two models, then the pattern must also end with a model. Details are given below. We also tried simple parsing, but we found that fairly simple pattern matching works better than an off-the-shelf parser due to the may “ungrammatical” sentences.

3.1.4. Snippet extraction: The patterns are run over the corpus, and all matching snippets are extracted. Matching is conceptually straightforward. We use a fast algorithm that finds all matches of all patterns with complexity logarithmic in the number of patterns and linear in the corpus size.

3.1.5. Snippet filtering: Poor quality or overlapping snippets are removed. First, we filter out snippets that end with poor ending PoS tags, such as *conjunction* or *determiner*. Then we remove all snippets that appear as parts in other extracted (and unfiltered) snippets. Thus, the output contains only the longest snippets.

4. Example Results

We present results below for cars. Table 1 characterizes the data used in the experiments, and shows that there are a couple dozen brands, a couple hundred models, and tens or hundreds of thousands of mentions (for shoes or cars, respectively). Thus, extracted consumer sentiment can be statistically significant, even in the face of some extraction errors. We did a similar analysis on shoes, which we will mention only briefly.

Table 1: The Data Used in the Experiments

	CAR	**SHOE**
Number of Messages:	868,174	19,410
Number of Threads:	557,193	3,025
Number of Sentences:	5,972,695	65,010
Date Range:	1999–2007	2006–2007
Total number of different Named Entities:		
Brands	28	20
Models	180	200
Terms	1,037	188
Total number of Named Entity mentions identified:		
Brands,	503,895	20,768
Models	575,110	32,606
Terms	6,194,507	74,931

Example sentences with pairs of shoes, with italics indicating the automatically recognized models, give the flavor of typical comparisons:

- I am thinking of either the *Asics Foundation 7* or the *New Balance 1010*
- The *adizero* LT2 has a blown rubber outsole and I wonder how it compares to the *speedstar*?
- I would say that the *adiStar Cushion* is more pillowy soft than the *Supernova Cushion* is

Note that brand names are often left off, and that comparisons sometimes describe the attributes of interest, and sometimes do not. In our first set of results, we look at what pairs of products (e.g. shoes or cars) are mentioned together. All three examples above would thus be included in the co-occurrence results. Later, we will present snippets extracted from comparative sentences such as the one containing “more pillowy soft than.”

4.1. Co-occurrence results

The co-occurrence patterns of the extracted entities can be nicely visualized using one of the many graphing tools available; The results presented below use pajek [1].. Figure 1 shows a plot for car models. The sizes of the circles indicate the relative number of mentions of the product. Links between vertices indicate frequently co-occurring models. It can be seen that the most central (the most compared against) cars are popular ones such as the Honda Cord, Toyota Campy, and Honda Civic. The Accord, Camry and Maxima are very similar cars and therefore appear together. Similarly, the Civic, Sentra, Corolla, and Protégé, are similar cars and appear closer to one another. It is interesting to note that the fact that Ford is the only American manufacturer to compete directly with the Japanese

These graphs of product co-occurrence also show which products are the most central to the discussion; i.e., the most compared against. Although the identity of the car brands central to the car reviews are perhaps not surprising, it was not obvious (even to a runner) that the Asics Gel Kayano would be the most central shoe in the market. (Figure not shown due to space limits.) Of more interest that *what* is compared is *how* the items are compared. The next section addresses this issue.

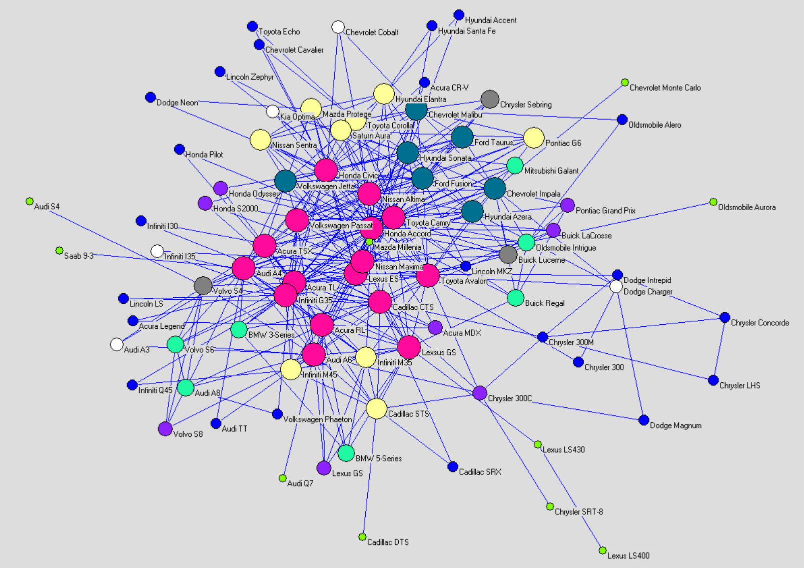

Figure 1: Car brand co-occurrence patterns. Links are shown for all pairs of cars mentioned more than 100 times together.

4.2. Snippet Results

Two very general patterns were used to extract snippets:

```
<Model>  *  /JJ  than  *  <Model>
<Model>  *  /JJ  *  to  *  <Model>
```

where /JJ means "adjective" and "*" means a gap is allowed. This resulted in 443 snippets for shoes, and 3025 snippets for the cars. For example:

- *1024* is very similar to my *1023*
- *1110* which I can get cheaper than the *1120*
- *2051* is noticeably heavier than the *inspire 2*

4.2.1 Relations from Snippets

When the snippets are extracted, we also get the pairs of products and the comparative term used between them. This information can be extracted and put into a table or a graph.

Snippets were classified into three categories, depending on whether the pair of cars was judged to be (a) similar, (b) different or (c) neither. Similarity of products is fairly reliably indicted by the word "similar":

- *Sonata* has soft ride similar to *Camry and Accord*

Differences between products rarely mention the word 'different'; it appears that it is true that each unhappy family is unhappy in its own way:

- *300 C* Touring looks so much better than the Magnum
- *300 C* was even more ridiculous than the $ 1200 quoted to me for the *SRT 8*

Many snippets neither indicate similarity nor difference, but merely mention a pair of products, for example asking how they compare. We tried training a support vector machine to predict the category labels of the snippets. Training on cars and testing on shoes gave a micro-averaged F1 of 0.66; training on shoes and testing on cars gave an F1 of 0.76. We also wrote a very simple rule-based classifier of the following form:

```
 if(/similar/) { $L = "similar"; }
elsif(/comparable/) { $L = "similar"; }
elsif(/differ/) { $L = "different"; }
elsif(/JJR/) { $L = "different"; }
elsif(/more/) { $L = "different"; }
elsif(/less/) { $L = "different"; }
 else { $L = "other"; }
```

where *JJR* is the part of speech tag for a comparative adjective. These gave significantly better micro-averaged F1's of 0.785, and 0.82 on the same shoe and car test sets.

Focusing in on the snippets classified as "different," we can easy extract from those snippets which car or shoe was labeled as better, and the keyword or phrase (if any) indicating on what attribute they were better. The result is a count of occurrences of 4-tuples of the form: 4 *compare(Camry, Accord, noise, better).*
This is interpreted as "there were 4 snippets in which the Camry was judged to be better than the Accord in terms of noise".

A summary of the results for comparisons between the Accord and Camry gives the flavor of what is learned. In the results below, tuples of the form (X,Y,Z, worse) have been converted to tuples of form (Y,X,Z, better). The "vote" is 12 to 8 saying the

Accord is better than the Camry (these are general comparisons, where no attribute was found).

There is almost an even split (7 to 5) on which has a better price. The count is 5 to 0 saying the Accord has worse noise than the Camry. This is an example of a finding that would be extremely interesting to people who, like one of the authors, is highly sensitive to noise. Comments were 3 to 0 saying the Accord has a better interior. The Accord is perceived to be more sportive, and to drive better, while the Camry is bigger and has more leg room and looks better. Note that in each case there is an extracted phrase (e.g. "leg room" and a direction of comparison. Similar results are, of course, extracted for all shoe and car comparisons.

Combining results of the same form for multiple cars gives chains of relative rankings of products. Although these need not, in theory, be transitive, so far all that we have found give partial orderings. For example, in terms of the interior (with ">" indicating better, and ">>" indicating a particularly strong signal)

Fusion > Accord >> Camry

In terms of overall preferences (with "," indicating explicit statement of items being similar), we find:

Camry, Sonata > Accord, Passat >> Jetta
Civic > Jetta, Accord > Altima

5. Related Work

Detecting the sentiment of text fragments has been done at the word level [11] the phrase/sentence level and the document level [11]. Researchers have also proposed systems for aggregating and visualizing collections of opinions about specific products [2, 4, 11] and have developed review mining systems [6, 7] by extracting opinions about specific product features [5, 8]. None of these researchers focused on the relationships among or comparisons between different products.

6. Conclusions

Marketing campaigns are often designed in part based on product comparison tables: what is my product compared against, and on what dimensions? Extracting co-mentioned pairs of products and the terms that show up frequently with them provides precisely this kind of table. Web sites like those of Altrec or Sports Authority provide either predefined or user-selectable sets of shoes for comparison, along with select features of these shoes. They do not, however, answer the important question: What models of shoes are most frequently (or best) compared against a particular model of shoe? And on what dimensions should they be compared? Sports Authority at last count offered comparisons of 65 shoe models, with no mechanism for selecting a subset other than picking a brand. Other review sites like competitive runner (http://competitiverunner.com) are no better, offering similarly long lists. With evidence accumulating that offering more choice can lead to lower consumer happiness and lower purchase rates, it is becoming ever more clear that picking the right set of products for comparison is of great value both for consumers and for vendors. Mining consumer product reviews provides a relatively inexpensive mechanism not only for selecting potential products to compare against a give product, but also the dimensions on which to compare it.

7. References

[1] Batagelj, V. and Mrvar, A. *Pajek - Analysis and Visualization of Large Networks*. in *Graph Drawing Software*. 2003. Berlin: Springer.

[2] Chklovski, T. *Deriving Quantitative Overviews of Free Text Assessments on the Web*. in *Proceedings of 2006 International Conference on Intelligent User Interfaces (IUI06)*. 2006. Sydney, Australia.

[3] Etzioni, O., et al., *Unsupervised named-entity extraction from the Web: An experimental study*. Artificial Intelligence, 2005. **165**(1): p. 91-134.

[4] Gregory, M., et al. *User-Directed Sentiment Analysis: Visualizing the Affective Content of Documents*. in *Sentiment and Subjectivity in Text Workshop at the Annual Meeting of the Association of Computational Linguistics (ACL 2006)*. 2006.

[5] Hu, M. and Liu, B. *Mining and summarizing customer reviews*. in *KDD-2004*. 2004.

[6] Kim, S.-M. and Hovy, E. *Automatic Identification of Pro and Con Reasons in Online Reviews*. in *In Proceedings of the Conference on Computational Linguistics/Association for Computational Linguistics (COLING/ACL-2006)*. 2006. Sydney, Australia.

[7] Kim, S.-M. and Hovy, E. *Identifying and Analyzing Judgment Opinions*. in *Proceedings of HLT/NAACL-2006*. 2006. New York City, NY.

[8] Popescu, A.-M. and Etzioni, O. *Extracting Product Features and Opinions from Reviews*. in *Proceedings of HLT-EMNLP*. 2005.

[9] Rosenfeld, B. and Feldman, R. *Using Corpus Statistics on Entities to Improve Semi-supervised Relation Extraction from the Web*. in *Proceeding of ACL-07* 2007.

[10] Tjong, E.F., Sang, K., and Buchholz., S. *Introduction to the CoNLL-2000 Shared Task: Chunking*. in *in Proceedings of CoNLL-2000*. 2000. Lisbon, Portugal.

[11] Turney, P.D. *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*. in *In Proceedings of ACL 2002*. 2002.