

Have2eat: a Restaurant Finder with Review Summarization for Mobile Phones

Giuseppe Di Fabbrizio and Narendra Gupta Sveva Besana and Premkumar Mani
AT&T Labs - Research, Inc. AT&T Interactive - Applied Research
{pino,ngupta}@research.att.com {sbesana,pmani}@attinteractive.com

Abstract

Have2eat is a popular mobile application available for iPhone and Android-based devices that helps users to find and assess nearby restaurants. It lists restaurants located around the device and provides a quick highlight about the opinions expressed by online reviewers. Have2eat summarizes textual reviews by extracting relevant sentences and by automatically generating detailed ratings about specific aspects of the restaurant. A compact one-screen digest allows users to quickly access the information they need, expand to full review pages, and report their experience online by entering ratings and comments.

1 Introduction

Bloggers, professional reviewers, and consumers continuously create opinion-rich web reviews about products and services, with the result that textual reviews are now abundant on the web and often convey a useful overall rating. However, an overall rating cannot express the multiple or conflicting opinions that might be contained in the text and screening the content of a large number of reviews could be a daunting task. For example, a restaurant might receive a great evaluation overall, while the service might be rated below-average due to slow and discourteous wait staff. Pinpointing opinions in documents, and the entities being referenced, would provide a finer-grained sentiment analysis and better summarize users' opinions. In addition, selecting salient sentences from the reviews to textually summarize opinions would add useful details to consumers that are not expressed by numeric ratings. This is especially true for so-called *road warriors* and mobile users "on the run" who are often dealing with limited time and display real estate in searching for a restaurant to make a decision.

Have2eat¹ is a popular² mobile application available for iPhone and Android-based devices that addresses these challenges. Have2eat uses the geo-location information either from the GPS device or explicitly entered by the user to produce a list of restaurants sorted by distance and located within a specific radius from the originating location. In addition, when restaurant reviews are available, a compact one-screen digest displays a summary of the reviews posted on the web by other customers. Customers can expand to read a full review page and also enter their own ratings, comments and feedback. The review summaries are visualized on the mobile screen:

- **graphically** by *thumbs-up* (positive reviews) and *thumbs-down* (negative reviews) for different aspects of the restaurant;
- **textually** by a few sentences selected from review texts that best summarize the opinions about various aspects of the restaurant expressed in the reviews;

Extracting opinions from text presents many natural language processing challenges. Prior work on sentiment analysis has been focusing on binary classification of positive and negative opinions (Turney, 2002; Pang et al., 2002; Yu and Hatzivassiloglou, 2003), while aspect rating inference (e.g., the task of determining the opinion polarity in a multi-point scale) has been previously analyzed in Pang and Lee (2005); Goldberg and Zhu (2006); Leung et al. (2006). More recently, Snyder and Barzilay (2007); Shimada and Endo (2008) extended the inference process to multi-aspect ratings where reviews include numerical ratings from mutually dependent aspects. Snyder and Barzilay (2007) shows that modeling the dependencies between aspect ratings in the same reviews helps to reduce the rank-loss (Crammer and Singer, 2001).

¹www.have2eat.com

²More than 400,000 downloads to-date for the iPhone version alone

There are similar mobile applications obtainable either on the Apple iPhone App Store or as web-based mobile application, such as Zagat³, UrbanSpoon⁴, YP Mobile⁵, and Yelp⁶, but, to the extent of our knowledge, most of them are only focused on finding the restaurant location based on proximity and some restaurant filtering criterion. When available, restaurant reviews are simply visualized as contiguous list of text snippets with the overall experience rating. None of the listed applications include extended rating predictions and reviews summarization.

2 System Description

The have2eat system architecture is composed of two parts: 1) predictive model training – illustrated in Figure 1 and described in section 2.1, and 2) graphical and textual summarization – shown in Figure 2 and described in section 2.2.

2.1 Graphical summarization by thumbs up/down

The majority of textual reviews available online are accompanied by a single overall rating of the restaurant. To predict consistent ratings for different aspects, namely *food*, *service*, *atmosphere*, *value*, and *overall experience*, we use machine learning techniques to train predictive models, one for each aspect; see Figure 1. More specifically, we used approximately 6,000 restaurant reviews scraped from a restaurant review website⁷. On this website, besides textual reviews, users have also provided numerical ratings for the five aspects mentioned above. Ratings are given on a scale of 1 to 5, 1 being poor and 5 excellent. We experimented with different regression and classification models using a host of syntactic and semantic features. We evaluated these models using rank-loss metrics which measure the average difference between predicted and actual ratings. We found that a maximum entropy (Nigam et al., 1999) model combined with a re-ranking method that keeps in consideration the interdependence among aspect ratings, provided the best predictive model with an average rank-loss of 0.617 (Gupta et al., 2010). This result is better than previous work on the same task as described in Snyder and Barzilay (2007).

To cope with the limited real estate on mobile phones for displaying and allowing users to input their opinions, the predicted ratings were mapped onto thumbs-up and thumbs-down. For each restau-

rant the proportion of reviews with rating of 1 and 2 was considered thumbs down and ratings of 4 and 5 were mapped to thumbs up. Table 1 shows an example of this mapping.

	Reviews			Thumbs	
	a	b	c	Up	Down
Atmosphere	3	2	4	50%	50%
Food	4	4	5	100%	0
Value	3	2	4	50%	50%
Service	5	5	5	100%	0
Overall	4	4	5	100%	0

Table 1: Mapping example between ratings and thumbs up/down. Ratings of 3 are considered neutral and ignored in this mapping

2.2 Textual summaries by sentence selection

Figure 2 shows how summary sentences are selected from textual reviews. As described in the previous section, we trained predictive models for each aspect of the restaurant. To select summary sentences we split the review text into sentences⁸. Using the predictive models and iterating over the restaurant listings, sentences in the reviews are classified by aspect ratings and confidence score. As a result, for each sentence we get 5 ratings and confidence scores for those ratings. We then select a few sentences that have extreme ratings and high confidence and present them as summary text.

We evaluated these summaries using the following metrics.

1. Aspect Accuracy: How well selected sentences represent the aspect they are supposed to.
2. Coverage: How many of the aspects present in the textual reviews are represented in the selected sentences.

⁸For this purpose we used a sentence splitter based on statistical models which besides n-grams also uses word part-of-speech as features. This sentence splitter was trained on email data and is 97% accurate.

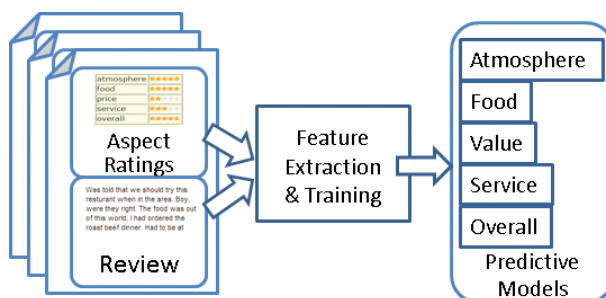


Figure 1: Predictive model training

³mobile.zagat.com

⁴www.urbanspoon.com

⁵m.yip.com

⁶m.yelp.com

⁷www.we8there.com

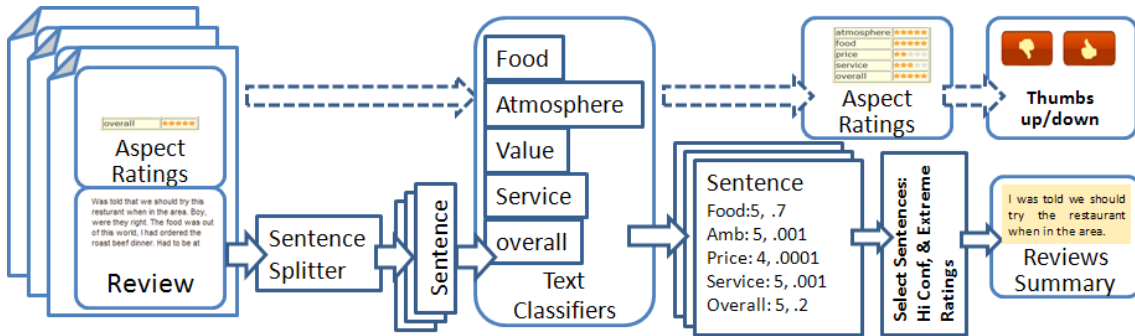


Figure 2: Graphical and textual summarization

3. Rating Consistency: How consistent the selected sentences with the summarizing aspect ratings are.
4. Summary quality: Subjective human judgments as to how good the summaries are and automatic multi-document summarization to how good the summaries are compared to a manually created GOLD standard using ROUGE-based (Lin, 2004) metrics.

A detailed description of the summarization task evaluation will be published elsewhere.

3 Demonstration

When launching the application, users are presented with a list of twenty nearby restaurants. The user can browse more restaurants by tapping on a link at the bottom of the page. For each listing we show the distance from the current location and, if available, we provide a thumbs-up or thumbs-down, price information and the summary sentence with the highest confidence score across aspects. Figure 3 shows an example of the *List* page. If users want a list of restaurants for a different location they can tap the *Change* button at the top of the page. This action will bring up the *Location* page where the user can enter city and state and/or a street address.

Users can select a restaurant in the list to view the details, see Figure 4. Details include address, phone number and thumbs up/down for the overall, food, service, value and atmosphere aspects. The user can provide feedback by tapping on the thumbs-up or thumbs-down buttons, as well as by leaving a comment at the bottom of the screen. This page also includes a few summary sentences with extreme ratings and high confidence scores. An example of selected sentences with their polarity is shown in Table 2. By tapping on any of the sentences the users can view the full text of the review from which the sentence was selected. Users can also add a new restaurant by tapping the *Add* icon in the tab bar.

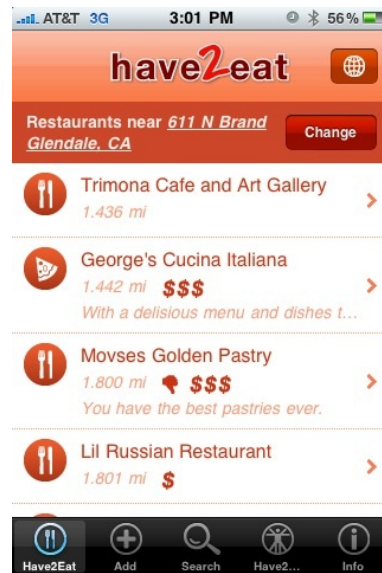


Figure 3: Have2eat listings screen shot on iPhone

Figure 5 displays the review selected in the *Details* page along with any other reviews which exist for the restaurant. Users can give feedback on whether they found the review helpful or not by using a thumbs-up or a thumbs-down respectively. Users can also add a review by tapping on a link at the bottom of the page.

4 Conclusion

This demonstration has shown a restaurant finder application for mobile phones, which makes use of summarization techniques to predict aspect ratings from review text and select salient phrases expressing users' opinions about specific restaurant aspects. Users can directly contribute with their feedback by tapping on the aspect thumbs buttons or by directly typing comments.

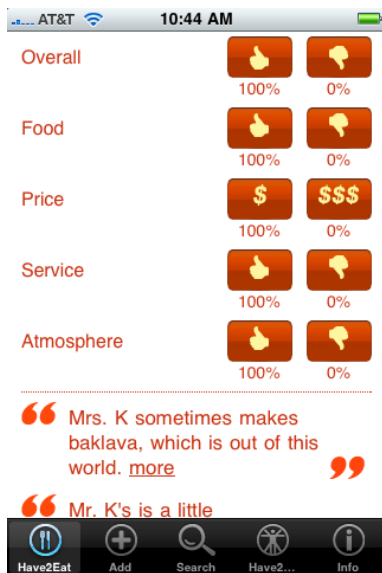


Figure 4: Have2eat automatically predicted aspect ratings and summary

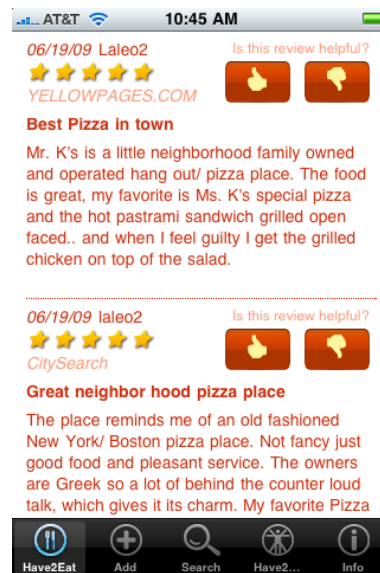


Figure 5: Have2eat reviews

Restaurant 1 (3 reviews)	
+	The soups are GREAT! Everything that we have ever ordered has exceeded the ex...
+	Delivery is prompt and credit cards are welcome
+	Their chicken fried rice is the second best in Southern California.
Restaurant 2 (8 reviews)	
+	Great tasting burgers, friendly fast service!
+	The inside is warm and even though the chairs looked uncomfortable, they were not at all.
-	Too many other places to try to worry about getting mediocre food as a high price.
Restaurant 3 (4 reviews)	
+	The salads are tasty, the breadsticks are to die for.
-	We waited approximate 10 more minutes and then asked how much longer.
+	A fun place to go with family or a date.
+	If you like salt then this is the place to go, almost everything is full of s...

Table 2: Example of extracted summaries

Acknowledgments

We thank Jay Lieske, Kirk Boydston, Amy Li, Gwen Christian, and Remi Zajac for their contributions and great enthusiasm.

References

Crammer, Koby and Yoram Singer. 2001. Pranking with ranking. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Neural Information Processing Systems: Natural and Synthetic (NIPS)*. MIT Press, Vancouver, British Columbia, Canada, pages 641–647.

Goldberg, Andrew B. and Jerry Zhu. 2006. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *TextGraphs: HLT/NAACL Workshop on Graph-based Algorithms for Natural Language Processing*.

Gupta, Narendra, Giuseppe Di Fabrizio, and Patrick Haffner. 2010. Capturing the stars: Predicting ratings for service and product reviews. In *Proceedings of the HLT-NAACL Workshop on Semantic Search (Semantic Search 2010)*. Los Angeles, CA, USA.

Leung, Cane Wing-ki, Stephen Chi-fai Chan, and Fu-lai Chung. 2006. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *Proceedings of The ECAI 2006 Workshop on Recommender Systems*. Riva del Garda, I, pages 62–66.

Lin, Chin-Yew. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10.

Nigam, Kamal, John Lafferty, and Andrew McCallum. 1999. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67.

Pang, Bo and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 115–124.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86.

Shimada, Kazutaka and Tsutomu Endo. 2008. Seeing several stars: A rating inference task for a document containing several evaluation criteria. In *Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference, PAKDD 2008*. Springer, Osaka, Japan, volume 5012 of *Lecture Notes in Computer Science*, pages 1006–1014.

Snyder, Benjamin and Regina Barzilay. 2007. Multiple aspect ranking using the Good Grief algorithm. In *Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL)*, pages 300–307.

Turney, Peter. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 417–424.

Yu, Hong and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.