Generating Recommendation Dialogs by Extracting Information from User Reviews

Kevin Reschke, Adam Vogel, and **Dan Jurafsky**
Stanford University
Stanford, CA, USA
{kreschke,acvogel,jurafsky}@stanford.edu

Abstract

Recommendation dialog systems help users navigate e-commerce listings by asking questions about users' preferences toward relevant domain attributes. We present a framework for generating and ranking fine-grained, highly relevant questions from user-generated reviews. We demonstrate our approach on a new dataset just released by Yelp, and release a new sentiment lexicon with 1329 adjectives for the restaurant domain.

1 Introduction

Recommendation dialog systems have been developed for a number of tasks ranging from product search to restaurant recommendation (Chai et al., 2002; Thompson et al., 2004; Bridge et al., 2005; Young et al., 2010). These systems learn user requirements through spoken or text-based dialog, asking questions about particular attributes to filter the space of relevant documents.

Traditionally, these systems draw questions from a small, fixed set of attributes, such as cuisine or price in the restaurant domain. However, these systems overlook an important element in users' interactions with online product listings: user-generated reviews. Huang et al. (2012) show that information extracted from user reviews greatly improves user experience in visual search interfaces. In this paper, we present a dialog-based interface that takes advantage of review texts. We demonstrate our system on a new challenge corpus of 11,537 businesses and 229,907 user reviews released by the popular review website Yelp[1], focusing on the dataset's 4724 restaurants and bars (164,106 reviews).

This paper makes two main contributions. First, we describe and qualitatively evaluate a framework for generating new, highly-relevant questions from user review texts. The framework makes use of techniques from topic modeling and sentiment-based aspect extraction to identify fine-grained attributes for each business. These attributes form the basis of a new set of questions that the system can ask the user.

Second, we use a method based on information-gain for dynamically ranking candidate questions during dialog production. This allows our system to select the most informative question at each dialog step. An evaluation based on simulated dialogs shows that both the ranking method and the automatically generated questions improve recall.

2 Generating Questions from Reviews

2.1 Subcategory Questions

Yelp provides each business with category labels for top-level cuisine types like *Japanese*, *Coffee & Tea*, and *Vegetarian*. Many of these top-level categories have natural subcategories (e.g., *ramen* vs. *sushi*). By identifying these subcategories, we enable questions which probe one step deeper than the top-level category label.

To identify these subcategories, we run Latent Dirichlet Analysis (LDA) (Blei et al., 2003) on the reviews of each set of businesses in the twenty most common top-level categories, using 10 topics and concatenating all of a business's reviews into one document.[2] Several researchers have used sentence-level documents to model topics in reviews, but these tend to generate topics about fine-grained aspects of the sort we discuss in Section 2.2 (Jo and Oh, 2011; Brody and Elhadad, 2010). We then manually labeled the topics, discarding junk topics and merging similar topics. Table 1 displays sample extracted subcategories.

Using these topic models, we assign a business

[1] https://www.yelp.com/dataset_challenge/

[2] We use the Topic Modeling Toolkit implementation: http://nlp.stanford.edu/software/tmt

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 499–504,
Sofia, Bulgaria, August 4-9 2013.

Category	Topic Label	Top Words
Italian	pizza	crust sauce pizza garlic sausage slice salad
	traditional	pasta sauce delicious ravioli veal dishes gnocchi
	bistro	bruschetta patio salad valet delicious brie panini
	deli	sandwich deli salad pasta delicious grocery meatball
American (New)	brew pub	beers peaks ale brewery patio ipa brew
	grill	steak salad delicious sliders ribs tots drinks
	bar	drinks vig bartender patio uptown dive karaoke
	bistro	drinks pretzel salad fondue patio sanwich windsor
	brunch	sandwich brunch salad delicious pancakes patio
	burger	burger fries sauce beef potato sandwich delicious
	mediterranean	pita hummus jungle salad delicious mediterranean wrap
Delis	italian	deli sandwich meats cannoli cheeses authentic sausage
	new york	deli beef sandwich pastrami corned fries waitress
	bagels	bagel sandwiches toasted lox delicious donuts yummy
	mediterranean	pita lemonade falafel hummus delicious salad bakery
	sandwiches	sandwich subs sauce beef tasty meats delicious
Japanese	sushi	sushi kyoto zen rolls tuna sashimi spicy
	teppanyaki	sapporo chef teppanyaki sushi drinks shrimp fried
	teriyaki	teriyaki sauce beef bowls veggies spicy grill
	ramen	noodles udon dishes blossom delicious soup ramen

Table 1: A sample of subcategory topics with hand-labels and top words.

to a subcategory based on the topic with highest probability in that business's topic distribution. Finally, we use these subcategory topics to generate questions for our recommender dialog system. Each top-level category corresponds to a single question whose potential answers are the set of subcategories: e.g., "What type of Japanese cuisine do you want?"

2.2 Questions from Fine-Grained Aspects

Our second source for questions is based on aspect extraction in sentiment summarization (Blair-Goldensohn et al., 2008; Brody and Elhadad, 2010). We define an aspect as any noun-phrase which is targeted by a sentiment predicate. For example, from the sentence "The place had **great atmosphere**, but the **service** was **slow**." we extract two aspects: +*atmosphere* and –*service*.

Our aspect extraction system has two steps. First we develop a domain specific sentiment lexicon. Second, we apply syntactic patterns to identify NPs targeted by these sentiment predicates.

2.2.1 Sentiment Lexicon

Coordination Graph We generate a list of domain-specific sentiment adjectives using graph propagation. We begin with a seed set combining PARADIGM+ (Jo and Oh, 2011) with 'strongly subjective' adjectives from the OpinionFinder lexicon (Wilson et al., 2005), yielding 1342 seeds. Like Brody and Elhadad (2010), we then construct a coordination graph that links adjectives modifying the same noun, but to increase precision we require that the adjectives also be conjoined by *and* (Hatzivassiloglou and McKeown, 1997). This reduces problems like propagating positive sentiment to *orange* in *good orange chicken*. We marked adjectives that follow *too* or lie in the scope of negation with special prefixes and treated them as distinct lexical entries.

Sentiment Propagation Negative and positive seeds are assigned values of 0 and 1 respectively. All other adjectives begin at 0.5. Then a standard propagation update is computed iteratively (see Eq. 3 of Brody and Elhadad (2010)).

In Brody and Elhadad's implementation of this propagation method, seed sentiment values are fixed, and the update step is repeated until the non-seed values converge. We found that three modifications significantly improved precision. First, we omit candidate nodes that don't link to at least two positive or two negative seeds. This eliminated spurious propagation caused by one-off parsing errors. Second, we run the propagation algorithm for fewer iterations (two iterations for negative terms and one for positive terms). We found that additional iterations led to significant error propagation when neutral (*italian*) or ambiguous (*thick*) terms were assigned sentiment.[3] Third, we update both non-seed and seed adjectives. This allows us to learn, for example, that the negative seed *decadent* is positive in the restaurant domain.

Table 2 shows a sample of sentiment adjectives

[3] Our results are consistent with the recent finding of Whitney and Sarkar (2012) that cautious systems are better when bootstrapping from seeds.

Negative Sentiment
institutional, underwhelming, not_nice, burntish, unidentifiable, inefficient, not_attentive, grotesque, confused, trashy, insufferable, grandiose, not_pleasant, timid, degrading, laughable, under-seasoned, dismayed, torn
Positive Sentiment
decadent, satisfied, lovely, stupendous, sizable, nutritious, intense, peaceful, not_expensive, elegant, rustic, fast, affordable, efficient, congenial, rich, not_too_heavy, wholesome, bustling, lush

Table 2: Sample of Learned Sentiment Adjectives

derived by this graph propagation method. The final lexicon has 1329 adjectives[4], including 853 terms not in the original seed set. The lexicon is available for download.[5]

Evaluative Verbs In addition to this adjective lexicon, we take 56 evaluative verbs such as *love* and *hate* from *admire*-class VerbNet predicates (Kipper-Schuler, 2005).

2.2.2 Extraction Patterns

To identify noun-phrases which are targeted by predicates in our sentiment lexicon, we develop hand-crafted extraction patterns defined over syntactic dependency parses (Blair-Goldensohn et al., 2008; Somasundaran and Wiebe, 2009) generated by the Stanford parser (Klein and Manning, 2003). Table 3 shows a sample of the aspects generated by these methods.

Adj + NP It is common practice to extract any NP modified by a sentiment adjective. However, this simple extraction rule suffers from precision problems. First, reviews often contain sentiment toward irrelevant, non-business targets (*Wayne* is the target of *excellent job* in (1)). Second, hypothetical contexts lead to spurious extractions. In (2), the extraction *+service* is clearly wrong–in fact, the opposite sentiment is being expressed.

(1) Wayne did an **excellent job** addressing our needs and giving us our options.

(2) Nice and airy atmosphere, but **service** could be more **attentive** at times.

[4]We manually removed 26 spurious terms which were caused by parsing errors or propagation to a neutral term.

[5]`http://nlp.stanford.edu/projects/yelp.shtml`

We address these problems by filtering out sentences in hypothetical contexts cued by *if*, *should*, *could*, or a question mark, and by adopting the following, more conservative extractions rules:

i) [BIZ + *have* + adj. + NP] Sentiment adjective modifies NP, main verb is *have*, subject is business name, *it*, *they*, *place*, or absent. (E.g., *This place has some really great yogurt and toppings*).

ii) [NP + *be* + adj.] Sentiment adjective linked to NP by *be*—e.g., *Our pizza was much too jalapeno-y*.

"Good For" + NP Next, we extract aspects using the pattern BIZ + positive adj. + *for* + NP, as in *It's perfect for a date night*. Examples of extracted aspects include *+lunch*, *+large groups*, *+drinks*, and *+quick lunch*.

Verb + NP Finally, we extract NPs that appear as direct object to one of our evaluative verbs (e.g., *We loved the* ***fried chicken***).

2.2.3 Aspects as Questions

We generate questions from these extracted aspects using simple templates. For example, the aspect *+burritos* yields the question: *Do you want a place with good burritos?*

3 Question Selection for Dialog

To utilize the questions generated from reviews in recommendation dialogs, we first formalize the dialog optimization task and then offer a solution.

3.1 Problem Statement

We consider a version of the Information Retrieval Dialog task introduced by Kopeček (1999). Businesses $b \in B$ have associated *attributes*, coming from a set Att. These attributes are a combination of Yelp categories and our automatically extracted aspects described in Section 2. Attributes $\text{att} \in \text{Att}$ take values in a finite domain $\text{dom}(\text{att})$. We denote the subset of businesses with an attribute att taking value $\text{val} \in \text{dom}(\text{att})$, as $B|_{\text{att=val}}$. Attributes are functions from businesses to subsets of values: $\text{att} : B \rightarrow \mathcal{P}(\text{dom}(\text{att}))$. We model a user *information need* I as a set of attribute/value pairs: $I = \{(\text{att}_1, \text{val}_1), \ldots, (\text{att}_{|I|}, \text{val}_{|I|})\}$.

Given a set of businesses and attributes, a *recommendation agent* π selects an attribute to ask

Chinese:	**Mexican:**
+beef +egg roll +sour soup +orange chicken	+salsa bar +burritos +fish tacos +guacamole
+noodles +crab puff +egg drop soup	+enchiladas +hot sauce +carne asade +breakfast burritos
+dim sum +fried rice +honey chicken	+horchata +green salsa +tortillas +quesadillas
Japanese:	**American (New)**
+rolls +sushi rolls +wasabi +sushi bar +salmon	+environment +drink menu +bar area +cocktails +brunch
+chicken katsu +crunch +green tea +sake selection	+hummus +mac and cheese +outdoor patio +seating area
+oysters +drink menu +sushi selection +quality	+lighting +brews +sangria +cheese plates

Table 3: Sample of the most frequent positive aspects extracted from review texts.

Input: Information need I
  Set of businesses B
  Set of attributes Att
  Recommendation agent π
  Dialog length K
Output: Dialog history H
  Recommended businesses B
Initialize dialog history $H = \emptyset$
for *step* $= 0$*; step* $< K$*; step*$++$ **do**
  Select an attribute: $\text{att} = \pi(B, H)$
  Query user for the answer: $\text{val} = I(\text{att})$
  Restrict set of businesses: $B = B|_{\text{att=val}}$
  Append answer: $H = H \cup \{(\text{att}, \text{val})\}$
end
Return (H, B)

Algorithm 1: Procedure for evaluating a recommendation agent

the user about, then uses the answer value to narrow the set of businesses to those with the desired attribute value, and selects another query. Algorithm 1 presents this process more formally. The recommendation agent can use both the set of businesses B and the history of question and answers H from the user to select the next query. Thus, formally a recommendation agent is a function $\pi : B \times H \rightarrow \text{Att}$. The dialog ends after a fixed number of queries K.

3.2 Information Gain Agent

The *information gain recommendation agent* chooses questions to ask the user by selecting question attributes that maximize the entropy of the resulting document set, in a manner similar to decision tree learning (Mitchell, 1997). Formally, we define a function *infogain* : $\text{Att} \times \mathcal{P}(B) \rightarrow \mathbb{R}$:

$$\textit{infogain}(\text{att}, B) = -\sum_{\text{vals} \in \mathcal{P}(\text{dom}(\text{att}))} \frac{|B_{\text{att=vals}}|}{|B|} \log \frac{|B_{\text{att=vals}}|}{|B|}$$

The agent then selects questions $\text{att} \in \text{Att}$ that maximize the information gain with respect to the set of businesses satisfying the dialog history H:

$$\pi(B, H) = \arg\max_{\text{att} \in \text{Att}} \textit{infogain}(\text{att}, B|_H)$$

4 Evaluation

4.1 Experimental Setup

We follow the standard approach of using the attributes of an individual business as a simulation of a user's preferences (Chung, 2004; Young et al., 2010). For every business $b \in B$ we form an information need composed of all of b's attributes:

$$I_b = \bigcup_{\{\text{att} \in \text{Att} | \text{att}(b) \neq \emptyset\}} (\text{att}, \text{att}(b))$$

To evaluate a recommendation agent, we use the *recall* metric, which measures how well an information need is satisfied. For each information need I, let B_I be the set of businesses that satisfy the questions of an agent. We define the recall of the set of businesses with respect to the information need as

$$\text{recall}(B_I, I) = \frac{\sum_{b \in B_I} \sum_{(\text{att},\text{val}) \in I} \mathbb{1}[\text{val} \in \text{att}(b)]}{|B_I||I|}$$

We average recall across all information needs, yielding *average recall*.

We compare against a *random agent* baseline that selects attributes $\text{att} \in \text{Att}$ uniformly at random at each time step. Other recommendation dialog systems such as Young et al. (2010) select questions from a small fixed hierarchy, which is not applicable to our large set of attributes.

4.2 Results

Figure 1 shows the average recall for the random agent versus the information gain agent with varying sets of attributes. 'Top-level' repeatedly queries the user's top-level category preferences, 'Subtopic' additionally uses our topic modeling subcategories, and 'All' uses these plus the aspects extracted from reviews. We see that for sufficiently long dialogs, 'All' outperforms the other systems. The 'Subtopic' and 'Top-level' systems plateau after a few dialog steps once they've asked

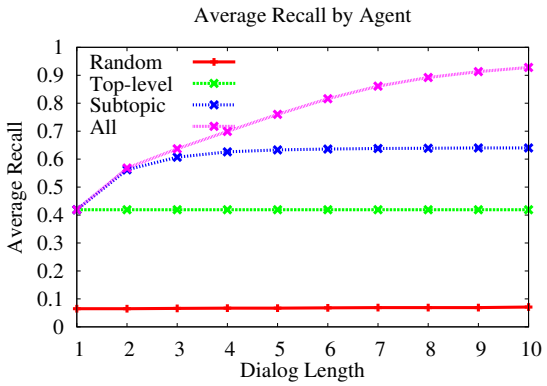

Figure 1: Average recall for each agent.

all useful questions. For instance, most businesses only have one or two top-level categories, so after the system has identified the top-level category that the user is interested in, it has no more good questions to ask. Note that the information gain agent starts dialogs with the top-level and appropriate subcategory questions, so it is only for longer dialogs that the fine-grained aspects boost performance.

Below we show a few sample output dialogs from our 'All' information gain agent.

Q: What kind of place do you want?
A: American (New)
Q: What kind of American (New) do you want: bar, bistro, standard, burgers, brew pub, or brunch?
A: bistro
Q: Do you want a place with a good patio?
A: Yes

Q: What kind of place do you want?
A: Chinese
Q: What kind of Chinese place do you want: buffet, dim sum, noodles, pan Asian, Panda Express, sit down, or veggie?
A: sit down
Q: Do you want a place with a good lunch special?
A: Yes

Q: What kind of place do you want?
A: Mexican
Q: What kind of Mexican place do you want: dinner, taqueria, margarita bar, or tortas?
A: Margarita bar
Q: Do you want a place with a good patio?
A: Yes

5 Conclusion

We presented a system for extracting large sets of attributes from user reviews and selecting relevant attributes to ask questions about. Using topic models to discover subtypes of businesses, a domain-specific sentiment lexicon, and a number of new techniques for increasing precision in sentiment aspect extraction yields attributes that give a rich representation of the restaurant domain. We have made this 1329-term sentiment lexicon for the restaurant domain available as useful resource to the community. Our information gain recommendation agent gives a principled way to dynamically combine these diverse attributes to ask relevant questions in a coherent dialog. Our approach thus offers a new way to integrate the advantages of the curated hand-build attributes used in statistical slot and filler dialog systems, and the distributionally induced, highly relevant categories built by sentiment aspect extraction systems.

6 Acknowledgments

Thanks to the anonymous reviewers and the Stanford NLP group for helpful suggestions. The authors also gratefully acknowledge the support of the Nuance Foundation, the Defense Advanced Research Projects Agency (DARPA) Deep Exploration and Filtering of Text (DEFT) Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-13-2-0040, ONR grants N00014-10-1-0109 and N00014-13-1-0287 and ARO grant W911NF-07-1-0216, and the Center for Advanced Study in the Behavioral Sciences.

References

Sasha Blair-Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A Reis, and Jeff Reynar. 2008. Building a sentiment summarizer for local service reviews. In *WWW Workshop on NLP in the Information Explosion Era*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Derek Bridge, Mehmet H. Göker, Lorraine McGinty, and Barry Smyth. 2005. Case-based recommender systems. *Knowledge Engineering Review*, 20(3):315–320.

Samuel Brody and Noemie Elhadad. 2010. An unsupervised aspect-sentiment model for online reviews.

In *Proceedings of HLT NAACL 2010*, pages 804–812.

Joyce Chai, Veronika Horvath, Nicolas Nicolov, Margo Stys, A Kambhatla, Wlodek Zadrozny, and Prem Melville. 2002. Natural language assistant - a dialog system for online product recommendation. *AI Magazine*, 23:63–75.

Grace Chung. 2004. Developing a flexible spoken dialog system using simulation. In *Proceedings of ACL 2004*, pages 63–70.

Vasileios Hatzivassiloglou and Kathleen R McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of EACL 1997*, pages 174–181.

Jeff Huang, Oren Etzioni, Luke Zettlemoyer, Kevin Clark, and Christian Lee. 2012. Revminer: An extractive interface for navigating reviews on a smartphone. In *Proceedings of UIST 2012*.

Yohan Jo and Alice H Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 815–824.

Karin Kipper-Schuler. 2005. Verbnet: A broad-coverage, comprehensive verb lexicon.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings ACL 2003*, pages 423–430.

I. Kopeček. 1999. Modeling of the information retrieval dialogue systems. In *Proceedings of the Workshop on Text, Speech and Dialogue-TSD 99, Lectures Notes in Artificial Intelligence 1692*, pages 302–307. Springer-Verlag.

Tom M. Mitchell. 1997. *Machine Learning*. McGraw-Hill, New York.

Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of ACL 2009*, pages 226–234.

Cynthia A. Thompson, Mehmet H. Goeker, and Pat Langley. 2004. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research (JAIR)*, 21:393–428.

Max Whitney and Anoop Sarkar. 2012. Bootstrapping via graph propagation. In *Proceedings of the ACL 2012*, pages 620–628, Jeju Island, Korea.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Opinionfinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005 on Interactive Demonstrations*, pages 34–35.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174, April.