Building a Sentiment Summarizer for Local Service Reviews

Sasha Blair-Goldensohn
Google Inc.
76 Ninth Avenue
New York, NY 10011
sasha@google.com

Kerry Hannan
Google Inc.
76 Ninth Avenue
New York, NY 10011
khannan@google.com

Ryan McDonald
Google Inc.
76 Ninth Avenue
New York, NY 10011
ryanmcd@google.com

Tyler Neylon
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
tylern@google.com

George A. Reis[*]
Dept. of Electrical Engineering
Princeton University
Princeton, NJ 08544
gareis@princeton.edu

Jeff Reynar
Google Inc.
76 Ninth Avenue
New York, NY 10011
jreynar@google.com

ABSTRACT

Online user reviews are increasingly becoming the de-facto standard for measuring the quality of electronics, restaurants, merchants, etc. The sheer volume of online reviews makes it difficult for a human to process and extract all meaningful information in order to make an educated purchase. As a result, there has been a trend toward systems that can automatically summarize opinions from a set of reviews and display them in an easy to process manner [1, 9]. In this paper, we present a system that summarizes the sentiment of reviews for a local service such as a restaurant or hotel. In particular we focus on aspect-based summarization models [8], where a summary is built by extracting relevant aspects of a service, such as *service* or *value*, aggregating the sentiment per aspect, and selecting aspect-relevant text. We describe the details of both the aspect extraction and sentiment detection modules of our system. A novel aspect of these models is that they exploit user provided labels and domain specific characteristics of service reviews to increase quality.

1. INTRODUCTION

Online reviews for a wide variety of products and services are being created every day by customers who have either purchased these products or used these services. The volume of reviews for a given entity can often be prohibitive for a potential customer who wishes to read all relevant information, compare alternatives, and make an informed decision. Thus, the ability to analyze a set of online reviews and produce an easy to digest summary is a major challenge for online merchants, review aggregators[1] and local search services[2]. In this study, we look at the problem of *aspect-based sentiment summarization*. An aspect-based summarization system takes as input a set of user reviews for a specific product or service and produces a set of relevant aspects, an aggregate score for each aspect, and supporting textual evidence. For example, figure 1 summarizes a restaurant using aspects *food*, *decor*, *service*, and *value*.

Aspect-based sentiment summarization has been studied in the past [8, 17, 7, 3, 23]. However, these studies typically make the highly limiting assumptions that no a priori knowledge of the domain being summarized is available, and that every review consists solely of the text of the review. In reality, most online reviews come with at least some labeling – usually the overall sentiment of the review is indicated – and we can often say something about the domain.

In this study we specifically look at the problem of summarizing opinions of *local services*. This designation includes restaurants and hotels, but increasingly users are reviewing a wide variety of entities such as hair salons, schools, museums, retailers, auto shops, golf courses, etc. Our goal is to create a general system that can handle all services with sufficient accuracy to be of utility to users. The architecture we employ is standard for aspect-based summarization. For every queried service S, it consists of three steps,

1. Identify all sentiment laden text fragments in the reviews

2. Identify relevant aspects for S that are mentioned in these fragments

3. Aggregate sentiment over each aspect based on sentiment of mentions

Central to our system is the ability to exploit different sources of information when available. In particular, we show how user provided document level sentiment can aid in the prediction of sentiment on the phrase/sentence level through a variety of models. Furthermore, we argue that the service domain has specific characteristics that can be exploited in order to improve both quality and coverage of generated summaries. This includes the observation that nearly all services share basic aspects with one another and that a large number of queries for online reviews pertain only to a small number of service types.

We begin with a quick overview of our system's architecture followed by a detailed description and analysis of each

[*] This work was undertaken while at Google.

[1] e.g., `tripadvisor.com` or `yelp.com`

[2] e.g., `maps.google.com`, `local.yahoo.com` or `maps.live.com/localsearch`

NLPIX2008, April 22, 2008, Beijing, China.
.

Nikos' Fine Dining

Food	4/5	"Nikos' has the Best fish in the city."
Decor	3/5	"It's cozy with an old world feel.
Service	1/5	"Our waitress was really rude!"
Value	5/5	"Good Greek food for the $ here ..."

Figure 1: An example aspect-based summary.

of its components. We discuss related work in section 5 and conclude in section 6.

1.1 System Overview

A general overview of the system is given in figure 2. The input to the system is a set of reviews corresponding to a local service entity. The text extractor breaks these review texts into a set of text fragments that might be of use in a summary. This can include sentences, clauses and phrases. These text fragments will be used to aggregate ratings for any aspect mentioned within them, but also as candidates for the final summary where evidence for each aspect rating will be included. Our system uses both sentence and phrase level text fragments when generating a summary. However, to simplify presentation, we will generally discuss our processing at the sentence level in this paper.

The second stage is to classify all extracted sentences as being positive, negative or neutral in opinion. This component of the system is described in section 2. The model we employ for sentiment classification is a hybrid that uses both lexicon-based and machine learning algorithms. We show that by modeling the context of a sentence as well as the global information provided by the user, e.g., an overall star rating, we can improve the sentiment classification at the sentence level.

The next step in our system is aspect extraction, which is discussed in section 3. Again we employ a hybrid, but this time we combine a dynamic aspect extractor, where aspects are determined from the text of the review alone, and a static extractor, where aspects are pre-defined and extraction classifiers trained on a set of labeled data. Static extractors leverage the fact that restaurants and hotels constitute a bulk of online searches for local reviews. Thus, by building specialized extractors for these domains we can improve the overall accuracy of the system.

The output of the sentiment classifier and aspect extractor will be a set of sentences that have been labeled with sentiment and the corresponding aspects that they discuss. These sentences are then input into the final summarizer that averages sentiment over each aspect and selects appropriate textual evidence for inclusion in the summary. This final component is described in section 4.

2. SENTIMENT CLASSIFICATION

After the system has extracted all sentences for a service of interest, the next stage is to classify each sentence as being positive, negative or neutral on some numeric scale. Note that sentiment classification at the sentence level is not a contrived task since users have typically only given a numeric sentiment rating for the entire review. Even highly positive reviews can include negative opinions and vice-versa. Thus, we will still have to classify sentences automatically, but our models should take into account any user provided numeric ratings when present.

Positive	Negative	Neutral
Good	Bad	And
Great	Terrible	Where
Excellect	Stupid	Too
Attractive	Expensive	Should
Wonderful	Frustrating	She

Table 1: Partial seed sets for lexicon induction.

Automatic sentiment analysis has been well studied with a variety of lexicon-based [21, 20, 8] and machine learning based systems [16, 5, 12, 6, 13, 18]. In our system we employed a hybrid as we desired the domain independence of a general lexicon sentiment classifier, but with the power of a machine learning classifier that can optimize system parameters on a large data set. A potential alternative to domain portability can come from machine learning techniques like those presented in [6], but currently these models are far more computationally intensive than lexicons.

2.1 Lexicon Construction

The first step in our hybrid model is to construct a general sentiment lexicon. This is done by defining a small initial seed lexicon of known positive and negative sentiment terms that is then expanded through synonym and antonym links in WordNet [14]. Our method is similar to that of Hu and Liu [8], where WordNet is used to grow sets of positive and negative terms. However, in our work we wish not only to create these sets, but also to weigh each member of the set with a confidence measure that represents how likely it is that the given word has the designated positive or negative sentiment. Thus, we use a modified version of the standard label propagation algorithms over graphs [22], adapting it to the sentiment lexicon task as described below.

Examples of positive, negative, and neutral sentiment words are given in Table 1. Note that we append simplified part-of-speech tags (adjective, adverb, noun or verb) to our seed set in order to help distinguish between multiple word senses.

The inputs to the algorithm are the three manually constructed seed sets that we denote as P (positive), N (negative), and M (neutral). Also provided as input are the synonym and antonym sets extracted from WordNet for arbitrary word w and denoted by $\mathsf{syn}(w)$ and $\mathsf{ant}(w)$ respectively.

The algorithm begins by defining a score vector $\mathbf{s}^m$ that will encode sentiment word scores for every word in WordNet. This vector will be iteratively updated (each update indicated by the superscript). We initialize $\mathbf{s}^0$ as:

$$\mathbf{s}_i^0 = \begin{cases} +1 & \text{if } w_i \in P \\ -1 & \text{if } w_i \in N \\ 0 & \forall w_i \in \text{WordNet} - P \cup N \end{cases}$$

That is, $\mathbf{s}^0$ is initialized so that all positive seed words get a value of $+1$, all negative seed words get a value of -1, and all other words a value of 0. Next, we choose a scaling factor $\lambda < 1$ to help define an adjacency matrix for the set of all words w_i in the WordNet lexicon $\mathbf{A} = (a_{ij})$ as:

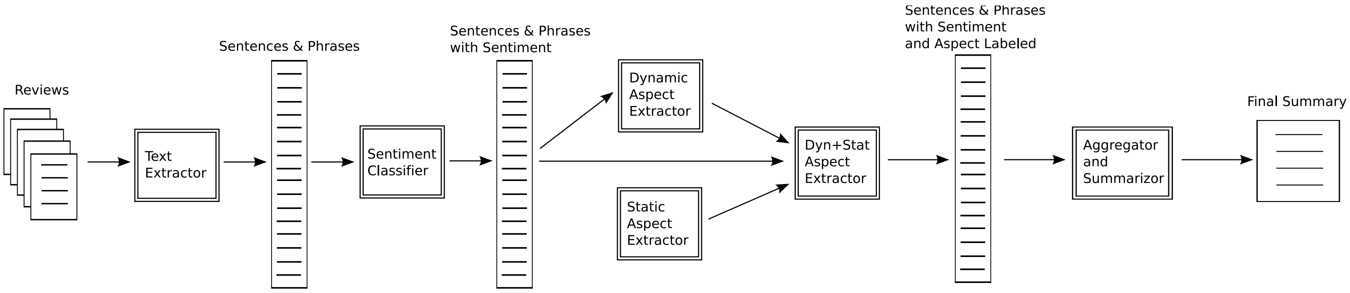
Reviews
Text Extractor
Sentences & Phrases
Sentiment Classifier
Sentences & Phrases with Sentiment
Dynamic Aspect Extractor
Static Aspect Extractor
Dyn+Stat Aspect Extractor
Sentences & Phrases with Sentiment and Aspect Labeled
Aggregator and Summarizor
Final Summary

Figure 2: System Overview. Double boxed items are system components and single boxed items are text files (possibly marked-up with sentiment/aspect information).

$$a_{ij} = \begin{cases} 1+\lambda & \text{if } i == j \\ +\lambda & \text{if } w_i \in \mathsf{syn}(w_j) \ \&\ w_i \notin M \\ -\lambda & \text{if } w_i \in \mathsf{ant}(w_j) \ \&\ w_i \notin M \\ 0 & \text{otherwise.} \end{cases}$$

$\mathbf{A}$ is simply a matrix that represents a directed, edge-weighted semantic graph where neighbouring nodes are synonyms or antonyms *and* are not part of the predefined neutral set — the latter being necessary to stop the propagation of sentiment through neutral words. For example, the neutral word "condition" may be a synonym of both "quality," a generally positive word, and "disease" (as in "a medical condition"), a generally negative word.

We then propagate the sentiment scores over the graph via repeated multiplication of $\mathbf{A}$ against score vectors $\mathbf{s}^m$, augmented with a sign-correction function for the seed words to compensate for relations which are less meaningful in the context of reviews. For example, the word "fast" – usually good in a review – may look negative as a synonym of "immoral" (an antonym of "good"), but instead of artificially labeling any of these as neutral, we could choose "fast" as a positive seed word, and maintain its sign at each of the m iterations:

```
for $m := 1$ to $M$
  $\mathbf{s}^m := \mathsf{sign\text{-}correct}(\mathbf{A}\,\mathbf{s}^{m-1})$
```

Here, the function $\mathbf{t} = \mathsf{sign\text{-}correct}(\mathbf{s})$ maintains $|\mathbf{t}_i| = |\mathbf{s}_i| \forall i$, ensures that $\mathsf{sign}(\mathbf{t}_i) = \mathbf{s}_i^0$ for all seed words w_i, and preserves the sign of all other words.

On every iteration of the algorithm, words in the graph that are positively adjacent to a large number of neighbours with similar sentiment will get a boost in score. Thus, a word that is not a seed word, but is a neighbour to at least one seed word, will obtain a sentiment score similar to that of its adjacent seed words. This will then propagate out to other words, and so on. Note that we take advantage of the disambiguation offered by part-of-speech labels in WordNet when traversing its heirarchy (recall that our seed set is also POS-labeled). For example, *model_a* (i.e., "model" as an adjective) is a synonym of *worthy_a*, whereas the noun *model_n* is not. Thus *model_a* and *worthy_a* can effect each other's scores, but not have an (incorrect) effect on *model_n*.

We use the decaying parameter λ to limit the magnitude of scores that are far away from seeds in the graph. In our experiments we used $\lambda = 0.2$ and ran for $M = 5$ iterations. Larger lambda led to too skewed a distribution of scores (the high word scores far outweighed all the other scores); while too small of a lambda gave the seed words too much importance. Large values of M did not seem to improve performance.

The final score vector $\mathbf{s}$ is derived by logarithmically scaling $\mathbf{s}^M$

$$\mathbf{s}_i := \begin{cases} \log(|\mathbf{s}_i^M|) * \mathsf{sign}(\mathbf{s}_i^M) & \text{if } |\mathbf{s}_i^M| > 1 \\ 0 & \text{otherwise} \end{cases}$$

We scaled scores to limit the impact that high scoring terms have on final classification decisions, since these scores can frequently be quite high.

In our experiments, the original seed set contained 20 negative and 47 positive words that were selected by hand to maximize domain coverage, as well as 293 neutral words that largely consist of stop words. Note that these neutral words serve as a kind of sanity check, in that we do not allow propagation of signed (positive/negative) scores through a neutral seed word. Running the algorithm resulted in an expanded sentiment lexicon of 5,705 positive and 6,605 negative words, some of which are shown in Table 2 with their final scores. Adjectives form nearly 90 percent of the induced vocabulary, followed by verbs, nouns and finally adverbs.

Most of the score polarities agree with human intuition, although not in all cases. Frequently, our overall score is correct, even if some contributing weights have a polarity that is incorect or based in a rare word sense.For instance, "dull" receives mild positive weight as an antonym of "cutting," yet its overall score is correctly negative because of antonymy with many strong positives like "keen" and "smart."

2.2 Classification

Using this bootstrapped lexicon, we can classify the sentiment of sentences or other text fragments. Given a tokenized string $x = (w_1, w_2, \ldots, w_n)$ of words, we classify its sentiment using the following function,

$$\mathsf{raw\text{-}score}(x) := \sum_{i=1}^{n} \mathbf{s}_i.$$

The score s_i for any term is given by the induced lexicon described above; we use a simple lexical negation detector to reverse the sign of s_i in cases where it is preceeded like a negation term like "not."

When $|\mathsf{raw\text{-}score}(x)|$ is below a threshold we classify x as neutral; otherwise positive or negative, according to its sign. Furthermore, we can rank sentences based on magnitude. An additional measure of interest is the *purity* of a fragment,

$$\mathsf{purity}(x) := \frac{\mathsf{raw\text{-}score}(x)}{\sum_{i=1}^{n} |\mathbf{s}_i|}.$$

Positive	Negative
Good_a (7.73)	Ugly_a (-5.88)
Swell_a (5.55)	Dull_a (-4.98)
Naughty_a (-5.48)	Tasteless_a (-4.38)
Intellectual_a (5.07)	Displace_v (-3.65)
Gorgeous_a (3.52)	Beelzebub_n:(-2.29)
Irreverent_a (3.26)	Bland_a (-1.95)
Angel_n (3.06)	Regrettably_r (-1.63)
Luckily_r (1.68)	Tardily_r (-1.06)

Table 2: Example terms from our induced sentiment lexicon, along with their scores and part-of-speech tags (adjective = a, adverb = r, noun = n, verb = v). The range of scores found by our algorithm is [-7.42,7.73].

This score is always in the range $[-1, 1]$, and correlates to the weighted fraction of words in x which match the overall sentiment of the raw score; it gives an added measure of the bias strength of x. For example, if two fragments, x_i and x_j, both have raw scores of 2, but x_i obtained it through two words with score 1, whereas x_j obtained through 2 words of scores 3 and -1, then x_i would be considered more pure or biased in the positive sense due to the lack of any negative evidence.

Though lexicon-based classifiers can be powerful predictors, they do not exploit any local or global context, which has been shown to improve performance [12, 13]. Furthermore, the scores are set using ad-hoc decaying functions instead of through an optimization on real world data. In order to overcome both shortcomings , we collected a set of 3916 sentences that have been manually labeled as being positive, negative or neutral [13]. We then trained a maximum entropy classifier (with a gaussian prior over the weights) [2, 11] to predict these ratings based on a small number of local and global contextual features for a sentence x_i occurring in the review $r = (x_1, x_2, \ldots, x_m)$, namely,

1. raw-score(x_i) and purity(x_i)
2. raw-score(x_{i-1}) and purity(x_{i-1})
3. raw-score(x_{i+1}) and purity(x_{i+1})
4. raw-score(r) and purity(r)

A common theme in our system is to use as much a priori information as possible. Consequently, we take advantage of user provided star ratings in our review data that essentially describe the overall sentiment of the service.[3] Note that this sentiment does not prescribe the sentiment of individual sentences, but only the sentiment conveyed overall by the review. It is frequently the case that a review may have a good/bad overall sentiment but have some sentences with opposite polarity. This is especially frequent for reviews with sentiment in the middle range of the scale. Thus, this information should be used only as an additional signal during classification, and not as a rigid rule when determining the sentiment of sentences or other fragments of text. In our maximum entropy model, we can simply add an additional feature (when present) whose weight will be optimized on a training set:

5. user-generated-rating(r)

The resulting maximum entropy classifiers will make sentiment predictions based not only on the scores of the sentence itself, but on the predicted neighbouring context scores and the predicted/gold overall scores of the document. Additionally, we could have trained the model using the words of the sentence as features, but in order to maintain domain independence we opted not to.

In order to train our classifieres, we randomly split our hand-labeled data into two equally sized sets, one to train our maximum entropy models and the other for evaluation. Each sentence was automatically annotated with its raw and purity scores, the raw and purity scores of its neighbouring sentences, the raw and purity scores of the document, and the user provided rating of the review from which the sentence was extracted (1.0 for positive, 0.0 for neutral, and -1.0 for negative.)

We then compared 4 systems:

- review-label: This system simply assigns a score of 1 to all sentences in a positive document, a score of -1 to all sentences in a negative document, and a score of 0 to all sentences in a neutral document, where the documents sentiment has been provided by the user who left the review. This is a simple baseline for when users have provided numeric ratings for a review and serves to show that even in these circumstances sentence level sentiment classification is non-trivial.
- raw-score: The system uses the raw-score to score sentences and then rank them in increasing or decreasing order for negative or positive classification respectively.
- max-ent: This system trains a model using the features defined above *excluding* the user provided review rating.
- max-ent-review-label: This system trains a model using the features defined above *including* the user provided review rating.

We compared the systems by measuring precision, recall, F1, and average precision for both the positive and negative classes since these are the classifications that will be used to aggregate and summarize the sentiment. For average precision we used a threshold of 0.0 for the raw-score and review-label and a probability of 0.5 for the max-ent classifiers. We chose to include average precision since our scoring functions (either raw score or conditional probability with maximum entropy) primarily serve to rank sentences for inclusion in the summary.

Results are given in table 3. Systems above the line *do not* use any user provided information, whereas the two systems below the line do. There are three important points to make here,

1. raw-score has relatively poor performance. However, adding context through a meta maximum entropy classifier leads to substantial improvements in accuracy.

[3] Though not the case in our data, it is further conceivable that a user will also have even identified some aspects and rated them explicitly, e.g., tripadvisor.com.

	Positive				**Negative**			
	Precision	Recall	F1	Avg. Prec.	Precision	Recall	F1	Avg. Prec.
raw-score	54.4	74.4	62.9	69.0	61.9	49.0	54.7	70.2
max-ent	62.3	76.3	68.6	80.3	61.9	76.7	68.5	71.3
review-label	63.9	89.6	74.6	66.2	77.0	86.1	81.3	76.6
max-ent-review-label	**68.0**	**90.7**	**77.7**	**83.1**	**77.2**	**86.3**	**81.4**	**84.4**

Table 3: Sentiment Classification Precision, Recall, F1, and Average Precision. Systems above the line do not use any user provided information. Bolded numbers represent the best result.

2. When we include features for the user provided review rating, performance again increases substantially – upwards of > 10-15% absolute.

3. The system that assigns all sentences the same polarity as the user provided review rating does quite well in terms of precision and recall, but very poor in terms of average precision and thus cannot be relied upon to rank sentences. Interestingly, this system does much better for negative sentences, indicating that sentences in a negative review are much more likely to be negative than sentences in a positive review being positive.

Considering these results, we decided to use the max-ent classifier for sentences in reviews that are not rated by users and max-ent-review-label for those reviews where users left a rating. We use the conditional probabilities of both these models to rank sentences as being either positive or negative.

3. ASPECT EXTRACTION

In this section we describe the component of our system that identifies the aspects of a service that users typically rate. This includes finding corresponding sentences that mention these aspects. Again we employ a hybrid. The first component is a string-based dynamic extractor that looks for frequent nouns or noun compounds in sentiment laden text, which is similar to the models in [8]. The second component leverages the fact that we observe a Zipfian, or at least head-heavy, distribution of service categories, where restaurants and hotels account for a large number of online searches for local services. Further supporting this observation is the existence of specialized websites which offer online reviews in the hotel or restaurant domains, e.g., `tripadvisor.com` or `zagats.com`.

To account for this limited number of high-importance categories, we build specialized models that have been trained on hand labeled data. Crucially, this hand labeled data can be used for other services besides restaurants and hotels since much of it deals with generic aspects that apply to many other services, e.g., *service* and *value*. We combine both components to provide a dynamic-static aspect extractor that is highly precise for a specific set of frequently queried services but is general enough to summarize reviews for all types of services.

3.1 Dynamic Aspect Extraction

Our first aspect extractor is dynamic in that it relies solely on the text of a set of reviews to determine the ratable aspects for a service. The techniques we use here are especially useful for identifying unique aspects of entities where either the aspect, entity type, or both are too sparse to include in our static models. For instance, dynamic analysis might find that for a given restaurant, many reviewers rave about the "fish tacos," and a good analysis of the reviews should promote this as a key aspect. Yet it is clearly not scaleable to create a *fish taco* classification model or an ontology of foods which would be so detailed as to include this as a food type. Similarly, for entity types which are infrequently queried, it may not be cost-effective to create any static aspects; yet we can still use dynamic aspect extraction to find, e.g., that a given janitorial service is known for its "steam carpet cleaning." Thus, dynamic extraction is critical to identifying key aspects both for frequent and rare service types.

We implement dynamic aspect extraction in a similar manner to [8]. We identify aspects as short strings which appear with high frequency in opinion statements, using a series of filters which employ syntactic patterns, relative word frequency, and the sentiment lexicon discussed in Section 2.

Briefly, we find candidate aspect strings which are nouns or noun compounds of up to three words, and which appear either in sentiment-bearing sentences and/or in certain syntactic patterns which indicate a possible opinion statment. While the presence of a term in a sentiment-bearing sentence improves its status as a possible aspect, we find that using syntactic patterns is more precise. For instance, the most productive pattern looks for noun sequences which follow an adjective, e.g. if a review contains "... great *fish tacos* ...", we extract *fish tacos* as a candidate aspect.

We then apply several filters to this list, which include removing candidates composed of stopwords, or candidates which occur with low relative frequency within the set of input reviews. Next, using our learned sentiment lexicon, we sum the overall weight of sentiment-bearing terms that appear in the syntactic patterns with the candidate aspects, and drop aspects which do not have sufficient mentions alongside known sentiment-bearing words. Finally, we collapse aspects at the word stem level, and rank the aspects by a manually tuned weighted sum of their frequency in sentiment-bearing sentences and the type of sentiment phrases mentioned above, with appearances in phrases carrying a greater weight. Table 4 shows the ranked list of dynamic aspects produced for several sample local services.

The dynamic aspects, and corresponding sentences, are then fed to the sentiment aggregation and summarization process discussed in Section 4, so that these unique, dynamically discovered properties may be included in the review summary.

3.2 Static Aspect Extraction

Dynamic aspect extraction is advantageous since it assumes nothing more than a set of relevant reviews for an entity. However, it suffers from fundamental problems that stem from the fact that aspects are fine-grained. Since an

Local Service	Dynamic Aspects Found
Casino	casino, buffet, pool, resort, beds
Children's Barber	haircut, job, experience, kids
Greek Restaurant	food, wine, service, appetizer, lamb
Department Store	selection, department, sales, shop, clothing

Table 4: Dynamic aspects identified for varoius local services types. The aspects are listed in descending ranking order as described in Section 3.1.

Restaurant

Food	883
Decor	284
Service	287
Value	169
Other	201

Hotel

Rooms	505
Location	310
Dining	165
Service	310
Value	165
Other	213

Table 5: Annotated static aspect data. Number of sentences annotated with corresponding aspect.

Restaurant

	Precision	Recall	F1
Food	84.2	82.2	83.2
Decor	70.5	47.1	56.5
Service	86.9	66.9	75.6
Value	90.3	55.6	68.9

Hotel

	Precision	Recall	F1
Rooms	86.1	78.2	82.0
Location	94.6	78.7	85.9
Dining	87.1	65.5	74.7
Service	83.9	65.5	73.6
Value	83.3	54.5	65.9

Table 6: Static aspect classification results.

aspect is essentially a string, then there is no way of knowing that "clam chowder" and "lobster bisque" are both an instance of the coarser aspects *soup*, *appetizer*, or *food*. Providing the user a summary consisting of a laundry list of aspects will most likely not be beneficial. Furthermore, the more fine-grained our aspects are, the less instances we have to aggregate sentiment, e.g., if we have three sentences each positively mentioning either "steak", "chicken", and "fries", then this should be sufficient to rate *food* high, but not any of the specific strings themselves. One possibility is to induce some kind of aspect clusters [7, 19], but this will rely on co-occurrence counts being sufficiently accurate.

A less general but more precise approach would be to learn to map string mentions, such as sentences, to a set of coarse-grained aspects using hand labeled training examples. This would require a labeled set of data for all possible aspects for all possible services, which we are unlikely to find. However, we can make the following observation. In any sentiment summarization system, users will tend to search for certain services with a much higher frequency than others. For example, a user searching for restaurant reviews is much more common than a user searching for hair salon reviews. Thus, if we can create highly accurate aspect extractors for the services that are queried most frequently, then we should be able to improve the system as a whole substantially. Towards this end, we will build accurate coarse-grained aspect extractors for *restaurants* and *hotels*, two of the most queried services for online opinions. Additional possibilities we plan to explore include retailers and entertainment vendors.

Our method is simple, we first identify all coarse-grained aspects of interest for these two domains. We chose *food*, *decor*, *service*, and *value* for restaurants and *rooms*, *location*, *dining*, *service*, and *value* for hotels. We then randomly selected a large number of sentences from each domain and labeled them with the corresponding aspects that were mentioned (we had a default *other* category for sentences that did not discuss any of the pre-defined coarse-grained aspects). A sentence could potentially be annotated with multiple aspects, e.g., "I loved my meal and the waiter was very attentive." Specifically, we annotated 1500 randomly selected sentences for both restaurants and hotels. Information on this data can be found in Table 5.

With this labeled data we trained a binary maximum entropy classifier (with a gaussian prior over the weights) for every aspect that predicted simply whether a sentence mentions that aspect or not. To evaluate the classifiers we used 10-fold cross-validation on the training set with results given in table 6. Note that for the most part the classifiers obtain a reasonably high precision with a modest recall. Precision can easily be increased further (at the expense of recall) by adjusting the classification threshold, which is by default set to a probability of 0.5 from the conditional distribution of positive classification. On the top end are categories like *location*, which has a limited vocabulary, as well as *rooms* and *food* that have diverse vocabularies, but account for a bulk of labeled data. On the bottom end is *decor*, *dining*, *service*, and *value*, all of which had the least amount of labeled data.

It is important to point out that these training sets were annotated in less than 2 person days. Thus, with a minimal amount of annotation we can build relatively precise classifiers for aspect extraction. Further techniques such as active learning and semi-supervised learning would almost certainly improve the time to performance trade-off.

Another point of interest is that restaurants and hotels both share the *service* and *value* categories. Though the vocabulary is not identical, e.g., "waiter" versus "front-desk", there is a sufficient overlap that we might expect additional gains through merging the training sets for these aspects. To test this, we built two additional classifiers for these categories using the labeled data from both domains as training data. Results are given in table 7. This table shows both the original measures plus the improved measures when we combine training sets. It is clear that combining training sets almost always leads to noticeable improvements, with the exception of the service category for restaurants. This fact suggests that our combined service and value classifiers can work for all other services, and not just restaurants and hotels.

3.3 Combining Static and Dynamic Aspects

Given an entity and set of associated reviews, we can use the methods in the previous two subsections to find relevant static and dynamic aspects, where the static aspects may

Restaurant

	Precision	Recall	F1
Service	**86.9** / 82.3	**66.9** / 66.6	**75.6** / 73.6
Value	90.3 / **94.1**	55.6 / **65.6**	68.9 / **77.4**

Hotel

	Precision	Recall	F1
Service	**83.9** / 82.1	65.5 / **69.7**	73.6 / **75.4**
Value	83.3 / **85.3**	54.5 / **66.6**	65.9 / **74.8**

Table 7: Combined service and value results (original/combined). Bolded numbers represent the best result.

include entity-type-specific aspects if the entity is a restaurant or hotel. However, in the summarizer presented in the next section, we wish to combine the two lists of aspects in order to present an aspect-based summary.

Currently, we apply the following algorithm to combine the list of relevant static aspects S and dynamic aspects D which are extracted for a set of reviews, and produce an output list C of ranked aspects to be shown in the summary.

1. Get as input a list of static aspects S and dynamic aspects D, as well as a count A such that A_a for some aspect a equals the weighted sum of phrases and sentences (as discussed in Section 3.1) from all reviews which has been classifed under aspect a.

2. Remove from D any aspects a where $a \in S$, i.e., duplicates of static aspects which were found by the dynamic aspect extraction process.

3. Filter both lists such that we drop aspects a for which A_a is less than a manually tuned threshold.

4. Add all aspects $s \in S$ to C, ranking by descending A_s.

5. Add dynamic aspects $d \in D$ to the end of C, in order of descending A_d, until either: (a) D is exhausted (b) A_d falls below a manually tuned threshold or (c) $|C|$ exceeds a manually tuned maxiumum number of aspects.

6. Add catch-all aspect *general comments* to end of C that will contain all opinion mentions about any aspect (including those not specifically included in C).

Section 4 shows how this aspect list is presented to the user in our summarization module.

There is clearly interesting work in the ranking and selection of this hybrid list of aspects, particularly when a large number of dynamic aspect candidates is available. While we do simple string-overlap-based redundancy pruning as in Hu and Liu [8] when combining the dynamic and static aspects, one can imagine further analysis not only to reduce redundancy but to improve organization or selection of aspects. For instance, if we were to notice that the majority of sentences mentioning the dynamic feature *fish taco* are categorized by our static aspect module as *food*, this might be used to rerank with respect to other *food*-related dynamic aspects to improve diversity, or at the interface level to nest one category beneath another.

4. SUMMARIZER

Using input prepared from the modules for sentiment classification and aspect extraction, the last stage in our system is the summarizer module. The summarizer extracts a concise sampling of the original input review texts, providing high-level overviews organized by aspect and sentiment.

Our current system provides both quantitative and qualitative information at the aspect level. Quantitatively, we aggregate all of the sentences which have been classified under a given aspect, and translate this into a “star rating” which represents the simple proportion of comments for that aspect which were classified as positive by a linear mapping to a ranking between one and five stars. Qualitatively, we represent the aspect by presenting a set of sentences to represent the key sentiments being expressed about each aspect.

After running our sentiment classifiers, aspect extractors, and aspect classifiers over all input review text we produce a summary via this high-level algorithm:

Take as input $\{S, P, A, C, L\}$:

S the set of non-neutral, sentiment-laden sentences extracted across all input reviews using the best applicable sentiment classifier as discussed in Section 2.

P the set of sentiment polarity scores P such that P_i is the sentiment score of sentence i. The scores in P have a sign and magnitude, indicating whether a sentence is classified as positive or negative and the probability of that prediction.

A the relevant aspects to summarize in the rank order defined in Section 3.3.

C the scores for each sentence under each aspect. For aspects that have a static classifier, C_{ia} is one if the classifier for aspect a classified sentence i as belonging to the aspect with probability above a manually tuned threshold, otherwise zero. For dynamic aspects, C_{ia} is one if the sentence contains a token which is a stem-level match for the aspect, else zero. For the *general comments* aspect, C_{ia} is always one, i.e. all sentences “belong” to general comments.

L the desired length of the summary in terms of the maximum number of sentences to display for each aspect.

For 1 up to L, For each aspect $a \in A$:

- If all sentences $s \in S$ which have $C_{sa} > 0$ have been included in the summary, continue to the next iteration of l.

- Set a flag variable d to either 1 or -1 depending on whether we wish to select a positive or negative sentence, respectively, in this iteration. We alternate d between positive and negative in such a way that positive and negative sentences will be inserted roughly in proportion to the total observed for aspect a. For example, if an aspect’s mentions are evenly divided between positive and negative, we alternate d’s value at each iteration.

- Of the remaining sentences $s \in S$ not yet inserted in the summary which have $C_{sa} > 0$, choose the sentence s which maximuzes $d * P_i$, i.e. the most negative/positive sentence belonging to aspect a. Add s to the summary.

The output is a summary which includes L sentences (if available) for each aspect in A.

4.1 Example Summaries

Figure 3 shows actual system outputs for several types of local services. Each output was created using the above-described summarization algorithm, using as input the set of service reviews returned with a local service result[4] on Google Maps (`maps.google.com`). We include examples of output for varied service types, two of which use domain-specific static aspect classifiers (restaurants and hotels), and two of which do not. We also chose services for which there are at least 16 available reviews. The amount of review input data varies, but each example has a minimum of 16 input reviews (56 total sentences) of input text.

One example summarizes the reviews for a barber shop specializing in childrens' haircuts. Note that the dynamic aspect extractor identifies the crucial "haircut" aspect, which is needed in this case since barber shops (unlike hotels and restaurants) are not a sufficiently popular category to have a specialized static extractor. The next-highest-ranking dynamic aspects are "job", "experience", and "kids", but these are not included because none of them occur with sufficent frequency in sentiment bearing phrases (as described in Step 4 of the aspect ranking algorithm in Section 3.3). The fact that our thresholding truncates these attributes is probably good, since they are of lower quality ("experience" is arguably okay).

In this case, therefore, a summary using dynamic aspects alone will include only "haircut" and "general comments." Because of variation in lexical choice, a dynamic extractor does not pick up on the fact that the concepts of service and value are in fact discussed frequently in the input reviews. However, our trained "service" classifier correctly identifies sentences including terms like *staff, treated, service, decency, people* and groups them under the "service" aspect.

If we did not have these static aspect classifiers, a simple alternative would be to show more sentences under "general comments." However, having the "service" and "value" aspect classifiers improves the summary in two specific ways. First, we are able to aggregate within each aspect and give it a quantitative score (i.e. the star rating). Secondly, the textual content is improved. If we simply toss in a bunch of uncategorized "general comments" information, we risk having an imbalanced summary which is overly focused on a single aspect. Instead, by including summary text for each of these aspects, we increase the diversity of information included and ensure that the summary does not fail to mention a core topic like "value." These are key advantages of our hybrid method over a dynamic-only aspect extraction.

We include three other examples, discussed more briefly due to space constraints. The first summarizes reviews of a department store, another category for which we have no domain-specific aspect classifiers. In this case, we find three dynamic aspects, including the clearly useful "sales" and "selection." The "department" dynamic aspect is selected at the wrong degree of granularity; our algorithm considers bigrams like "children's department, "women's department," and wrongly attempts to back off to the more general "department" since it covers more mentions.

The other examples include a hotel and a restaurant. Note that in these cases, we are able to take advantage of our static domain-specific aspect classifiers, e.g., for "rooms" in the hotel domain and "food" in the restaurant domain. Yet in both cases we see contribution from the dynamic classifier. For the restaurant, it calls out the fact that the establishment is known for its wine list; for the hotel, the dynamic aspects find several features which relate to the fact that this particular hotel is also a casino and resort. In both cases, the dynamic features complement the static models, by identifying particulars of the instance which would not necessarily be applicable to all services in the domain, e.g., since not all hotels have a "pool."

We emphasize that while the examples are chosen to highlight our system's capabilities, we present the output "as-is" and without editing, in order to give an accurate idea of the type of mistakes our system can make. For instance, our classifiers make several errors, e.g., the first and third "value" sentences in the restaurant example do not get the correct aspect. In the first case the presence of the word "decent" trips up the classifier as it is typically only mentioned sentences discussing the price. The second sentence is most likely mislabeled due to the presence of numbers, which again are highly correlated with price. Even though these suggest more sophisticated techniques are required, we have confidence that classifier-level mistakes are relatively rare, based on the formal evaluations described in the previous sections.

At this stage in our project we have no formal evaluation of the final output summaries produced by the system. As with most summarization studies, this is very difficult since users will often disagree on what constitutes the best content and delivery mechanism for the summary. We also do not have access to any reference summaries, which would enable use to at least attempt some kind of quantitative analysis using Rouge [10] or Pyramid scoring [15]. Even so, it is not at all clear that reference-based evaluation would be useful in this situation. A more precise measure would be to track user behaviour online to determine if time savings are achieved when searching for a local service.

5. RELATED WORK

Summarizing sentiment [1] and in particular summarizing sentiment by extracting and aggregating sentiment over ratable aspects has been a very active area of research recently. The work of Hu and Liu [8, 9] was one of the first studies into aspect-based summarization. Hu and Liu primarily examined association mining to extract product aspects, where each aspect was fine-grained a unique string. Sentiment over aspects was aggregated using a WordNet-based lexicon. Popescu and Etzioni [17], in their OPINE system, also extract string-based aspect mentions and find relevant opinions about them.

Gamon et al. [7] present an unsupervised aspect identification algorithm that employs clustering over sentences with each cluster representing an aspect. Sentence clusters are labeled with the most frequent non-stop word stem in the cluster. Carenini et al. [4] use the algorithms of Hu and Liu [8, 9] to extract explicit aspect mentions from reviews and extend this through a user supplied aspect hierarchy of a product class. Extracted aspects are clustered by placing them into the hierarchy using string and semantic similarity metrics. An interesting aspect of this work is a comparison between extractive and abstractive summarizations for sentiment [3]. The use of an ontology relates that work to the

[4] Business names have been anonymized.

Department Store (43 Reviews)
value (*) (5/5 stars, 9 comments)
(+) Good prices if you catch end-of-season sales.
(+) Worth looking at for a very few black, stretchy basic.
(+) It certainly made me aware of the popularity of this store.
service (*) (3/5 stars, 6 comments)
(+) They also have frequent sales so check in every now and then and you ...
(-) Not only did they lose good business, but everyone will positively know ...
(+) Pro: huge department storeCon: service not always exceptional Although ...
selection (5/5 stars, 14 comments)
(+) great quality, great selection very expensive, I expected them to carry ...
(+) Nice selection of baby clothes and accessories.
(+) I love the women's department as well as their selection of accessories.
department (5/5 stars, 7 comments)
(+) Best children's department as far as department stores go.
(+) This is the department store of choice for many of the city's visitors ...
sales (5/5 stars, 6 comments)
(+) Although chaotic during sales, there's a good reason why – they ...
(+) A great place for baby clothes when there are sales!
(+) Sometimes they have some great sales and you can get some really nice.
general comments (4.5/5 stars, 131 comments)
(+) This store has it all - except for good help!
(+) This Eastside art-deco landmark has been serving sophisticated New York...
(-) I had a bad experience while there

Greek Restaurant (85 Reviews)
food (4.5/5 stars, 130 comments)
(+) Food is very good and the ambiance is really nice too... butthe staff ...
(+) They do well with whole fish and lambshanks were very good.
(-) Desserts were 2/5 – i.e. uninspired and bad.
service (4/5 stars, 38 comments)
(+) Good food, good athmosphere, good service... no more to say ...
(-) Don't be put off by sometimes rude reservations staff or difficult to ...
(+) The hostess was not overly friendly, but the service was very good.
ambiance (*) (5/5 stars, 23 comments)
(+) I loved the atmosphere and the food was really well done.
(+) The atmosphere is subtle and not over-done and the service is excellent.
(+) Still, nice ambience and the great for carnivores.
value (*) (4/5 stars, 10 comments)
(+) Went here last night – great decor, decent but not excellent service.
(+) The food and value is definately worth it.
(-) Greeks found this restaurant right away when it opened 3-4 years ago and ...
wine (4.5/5 stars, 21 comments)
(+) Great wine selection and their dips are some of the best I've had ...
(+) The all Greek wine list is a nice touch as well.
(-) The wine list is all Greek so difficult to navigate unless you are ...
general comments (4.5/5 stars, 295 comments)
(+) My boyfriend and I call this place "Fancy Greek" ...
(+) The best, most authentic gourmet Greek in NY - no contest!
(-) The restaurant was able to accommodate my party of 15 in full comfort.

Children's Barber Shop (16 Reviews)
service (*) (3.5/5 stars, 7 comments)
(+) The staff does a nice job with cranky toddlers.
(+) We asked them not to cut the front of our sons hair, but they did.
(-) Better try another salon if you want to be treated with common decency.
value (*) (2.5/5 stars, 2 comments)
(+) This place is well worth the travel and the money.
(-) Quite pricey for a young childs haircut.
haircut (3/5 stars, 5 comments)
(+) This is a great place for your first haircut, but beware it's a toy ...
(+) Car seats are cute and the first haircut certificate is something i will ...
(-) Why can't kids just get a haircut and get out like the used to.
general comments (3.5/5 stars, 55 comments)
(+) We have always had the best experience at the shop.
(+) The whole scene gets on my nerves.
(+) And the haircutters range from excellent to fair - so get a ...

Hotel/Casino (46 Reviews)
rooms (*) (3/5 stars, 41 comments)
(+) The room was clean and everything worked fine – even the water pressure ...
(+) We went because of the free room and was pleasantly pleased ...
(-) The Treasure Bay Hotel was the worst hotel I had ever stayed at ...
service (*) (3/5 stars, 31 comments)
(+) Upon checking out another couple was checking early due to a problem ...
(+) Every single hotel staff member treated us great and answered every ...
(-) The food is cold and the service gives new meaning to Jamacian SLOW.
dining (*) (3/5 stars, 18 comments)
(+) our favorite place to stay in biloxi.the food is great also the service ...
(+) Offer of free buffet for joining the Players Club so we took them up on it.
(-) The buffet at the casino was terrible.
location (*) (3.5/5 stars, 31 comments)
(+) The casino across the street is a great theme casino and shuttles go ...
(+) The decor here is Spanish colonial in style, infused with lots of rich, ...
(-) Take it from me, pass it by.
value (*) (4/5 stars, 5 total comments)
(+) Excellent prices for rooms.
(-) just spent 5 hours wasting my money here.
casino (3/5 stars, 18 comments)
(+) The entertainment at the Casino is the best in town.
(+) Casino buffet was good also.
(-) The Casino was ok, but the actual hotel was gross.
buffet (3.5/5 stars, 8 comments)
(+) "Infinity" was simply EXCELLENT and the best buffet around with juicy ...
(+) Their buffet is typical and maybe a little better than average ...
(-) The selection at this buffet is somewhat limited compared ...
pool (4/5 stars, 5 comments)
(+) Our balcony overlooked the pool which was very nice in the evening.
(+) They have a great swimming pool and recreation area.
(-) The pool view rooms are noisy (that is the non smoking rooms) IF you ...
general comments (3/5 stars, 209 comments)
(+) Just a very nice place to stay and visit for a while.
(+) Initially we stayed at the Holiday Inn (which was very very nice) a ...
(-) I've gone several times with my parents over the years.

Figure 3: Summarizer outputs for a barber shop and deparment store. Sentences are marked with a (+)/(-) to indicate positive/negative polarity. Aspects marked with a (*) are not found by dynamic aspect extraction but are included using the static aspect classifiers

present study. However, instead of relying on an ontology and string matching, we specifically learn a classifier to map sentences to predefined coarse aspects.

Fully supervised methods for sentiment summarization include the Zhuang et al. [23] work on analyzing movie reviews. In that work all relevant information is annotated in a labeled training set allowing one to train highly accurate classifiers. The shortcoming of such an approach is that it requires a labeled corpus for every domain of interest, which is overcome in our system by employing a hybrid model.

Recently, Microsoft Live Labs (`live.com`) launched an aspect-based summarizer for products. Though it is unclear what technology is being used, their system clearly benefits from using user provided pros-cons list, which are common for many types of products – most notably electronics. This relates their system to ours in that both rely heavily on user provided signals to improve performance.

6. CONCLUSIONS

In this paper we presented an architecture for summarizing sentiment. The resulting system is highly precise for frequently queried services, yet also sufficiently general to produce quality summaries for all service types. The main technical contributions include new sentiment models that leverage context and user-provided labels to improve sentence level classification as well as a hybrid aspect extractor and summarizer that combines supervised and unsupervised methods to improve accuracy.

In the future we plan to adapt the system to products, which is a domain that has been well studied in the past. Just as in services, we believe that hybrid models can improve system performance since there again exists a pattern that a few products account for most review queries (e.g., electronics). Additionally, there is a set of aspects that is common across most products, such as *customer service*, *warranty*, and *value*, which can be utilized to improve the performance for less queried products.

We also plan to run user interface studies to determine the best mode of delivery of aspect-based sentiment summarization for both desktop and mobile computing platforms. By varying the number of aspects shown and the granularity of the associated text, a summary can change substantially.

Finally, more investigation of semi-supervised and active learning methods for aspect classification may provide a mechanism for further reducing the amount of labeled data required to produce highly accurate coarse-grained aspects.

Acknowledgements

We thank the anonymous reviewers for helpful comments. This work has benefited from discussions with Raj Krishnan,

Ivan Titov, Mike Wells, Chris Wang and the Google Maps team, Corrina Cortes, and Fernando Pereira.

7. REFERENCES

[1] P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan. An exploration of sentiment summarization. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, 2003.

[2] A. Berger, V. Della Pietra, and S. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[3] G. Carenini, R. Ng, and A. Pauls. Multi-document summarization of evaluative text. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2006.

[4] G. Carenini, R. Ng, and E. Zwart. Extracting knowledge from evaluative text. In *Proceedings of the International Conference on Knowledge Capture*, 2005.

[5] Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings the Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, 2005.

[6] M. Dredze, J. Blitzer, and F. Pereira. Biographies, bollywood, boom-boxes, and blenders: Domain adaptation for sentiment classification. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*, 2007.

[7] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. Pulse: Mining customer opinions from free text. In *Proceedings of the 6th International Symposium on Intelligent Data Analysis (IDA)*, 2005.

[8] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, 2004.

[9] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, 2004.

[10] C. Lin and E. Hovy. Automatic evaluation of summaries using n-gram cooccurrence statistics. In *Proceedings of the Conference on Human Language Technologies and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2003.

[11] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *International Conference on Computational Linguistics (COLING)*, 2002.

[12] Y. Mao and G. Lebanon. Isotonic conditional random fields and local sentiment flow. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[13] R. McDonald, K. Hannan, T. Neylon, M. Wells, and J. Reynar. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*, 2007.

[14] G. A. Miller. Wordnet: A lexical database for English. *Communications of the ACM*, (11):39–41, 1995. http://wordnet.princeton.edu/.

[15] A. Nenkova and R. Passonneau. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Conference on Human Language Technologies and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2004.

[16] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

[17] A. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2005.

[18] B. Snyder and R. Barzilay. Multiple aspect ranking using the Good Grief algorithm. In *Proceedings of the Joint Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies (NAACL-HLT)*, 2007.

[19] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the Annual World Wide Web Conference (WWW)*, 2008.

[20] P. Turney. Thumbs up or thumbs down? Sentiment orientation applied to unsupervised classification of reviews. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*, 2002.

[21] J. Wiebe. Learning subjective adjectives from corpora. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2000.

[22] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, CMU CALD tech report CMU-CALD-02, 2002.

[23] L. Zhuang, F. Jing, and X. Zhu. Movie review mining and summarization. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, 2006.