Efficient on-demand Opinion Mining*

Adriano Veloso, Wagner Meira Jr.

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brazil

Abstract. *Every day, a multitude of people express their opinions regarding diverse entities, such as services, places and products, in blogs (e.g., The BBC "Have Your Say" Blog), online forums (e.g.,* `slashdot.org`*) and review sites (e.g.,* `www.amazon.com`*). This constantly growing availability of opinionated content has created massive amounts of extremely valuable information. Currently, search engines are unable to explore such information, because (1) it is difficult to distinguish opinionated content from factual content, and (2) opinionated content may present different connotations or polarities (i.e., positive or negative, interesting or boring etc.). Recently, some attention has been devoted to the first problem − opinion retrieval, which consists of distinguishing opinionated content from factual content. However, research on opinion mining, which consists in classifying opinionated content with regards to the opinion it expresses, is still lacking. The main challenge is that the search space is huge due to the sparseness typically associated with textual evidence, and thus, the classification model needs to be very complex in order to achieve accurate results. In this paper we present a novel strategy for opinion mining, based on a lazy, on-demand, associative classification approach which reduces the complexity of the model by adopting a highly specific bias during the inductive process. The proposed approach was evaluated using collections obtained from two actual application scenarios: an online forum and a large product review site. The results demonstrate that the proposed approach can provide gains up to 9%, when compared against the state-of-the-art general purpose classification approach.*

1. Introduction

The Web has dramatically changed the way that people express their views and opinions. One can express opinions on almost anything at review sites, forums, discussion groups and blogs. The immediate consequence is a growing availability of opinionated content, which can be explored for marketing intelligence (e.g., *search for positive opinions about the resolution of Sony SDP digital camera*), helping individual choices (e.g., *search for nice places for vacation*), and public opinion retrieval (e.g., *are people worried about climate change?*). Despite this huge potential of applications, current IR (information retrieval) tools are still unable to search for opinions as conveniently as general Web search. However, users searching for information on the Web may have more complex information needs than simply finding any documents on a certain subject matter.

*This research was sponsored by UOL (www.uol.com.br) through its UOL Bolsa Pesquisa program, process number 20060519184000a.

For instance they may want to find documents containing other people's opinions on a certain topic, as opposed to documents with objective content, such as technical specifications. At least two steps are necessary in order to enable IR tools to search for opinions [Yu and Hatzivassiloglou 2003]:

1. Separating facts[1] from opionions – Facts and opinions are the two main types of textual content in the Web. Both types of content tend to be composed of a mixture of subjective and objective language, and thus, it is hard to automatically differentiate opinions from facts. Differently from opinions, facts are easily represented (and retrieved) with topic keywords, and due to this easy representation, current search engines treat all textual content as facts. However, the ability to detect factual and opinionated content allows distinct advantages in deciding what information to extract, and enabling the use of appropriate strategies for dealing specifically with facts (i.e., more appropriate ranking strategies) or opinions (i.e., opinion mining).
2. Identifying the polarity (or connotation) of opinions – Having distinguished whether a content is a fact or opinion, automatically identifying the polarity of opinions is required for processing more sophisticated queries.

Thus, given a query (e.g., *Play Station II opinion*), the opinion search engine must (1) retrieve all opinions about a particular entity (i.e., distinguish opinions from facts about Play Station II), and then (2) identify those opinions pointing to the desired polarity/connotation (i.e., from the retrieved opinions, identify positive and negative ones, and summarize the result). The first task (opinion retrieval) is receiving substantial attention, as observed in the TREC 2006 Blog Track [Ounis et al. 2006], which was focused on opinion retrieval tasks. The second task (opinion mining) is also important, and is the focus of this paper.

Opinion mining is concerned not with the topic some content is about, but with the opinion it expresses (i.e., identifying the connotation, or polarity, of opinions). This is a particularly hard task, because identifying the connotation associated with opinions may require the comprehension of textual content. Although full comprehension of natural language text remains well beyond the power of machines, the statistical analysis of opinionated text can provide an effective approach for opinion mining, while being computationally attractive. In this paper we considered opinion mining essentially as a (supervised) classification problem, that is, a set of examples (opinions for which the connotation is explicitly informed) is used to build a classification model which relates patterns that are implicit in the given examples, to a connotation (or a rating, a category etc.). This model is then used to classify opinions for which the corresponding connotations are unknown. Put in that way, the difference between the various classification approaches (or classifiers) resides basically in the format of the patterns that compose the model and in the bias[2] that is employed during pattern enumeration. The proposed approach uses *class association rules* [Liu et al. 1998] as basic components of the classification model. These

[1] Anything that can be proven true.

[2] In general, the (training) examples do not determine a unique classification model. Frequently there are an infinite number of models that are consistent with the given examples. Therefore, there must be factors other than just the examples that determine the model selected by the classifier. These other factors are called bias.

rules have the form $\mathcal{X} \rightarrow c$, where $\mathcal{X}$ is a combination of features within the opinions (i.e., words and sentences such as "excellent", "resolution", "not good", or even information about the opinion holder), and c is a connotation (i.e., positive, interesting etc.). These rules are automatically discovered by progressively combining features until $\mathcal{X}$ is sufficiently discriminative (this is an important advantage when compared with typical approaches that are based on the semantic orientation of some predefined adjectives and adverbs). Further, to avoid the enumeration of an excessive amount of patterns (which is a common problem due to the sparseness associated with textual content), the proposed classification approach adopts a highly specific bias, which induces the patterns on a demand-driven basis, as exactly as needed to classify a given opinionated sentence (i.e., positive or negative, interesting or boring etc.). More specifically, instead of generating a single (and extremely complex) classification model that is good on average for classifying all opinions, the proposed lazy approach delays the inductive process until a specific opinion is given for classification. Then this opinion is used as a filter which removes from consideration irrelevant examples, and a specific classification model is generated for this opinion, on a demand-driven basis. Since a much smaller number of examples are considered, the generated models are extremely simple when compared to the model that would be generated from the entire set of examples.

The proposed classification approach is evaluated using opinions obtained from two actual application scenarios: the Slashdot.org forum and the Amazon.com review site. Our results demonstrate that the proposed approach consistently achieves better performance than the baseline, showing gains up to 9% in classification accuracy. Further, the proposed lazy approach is much faster than the baseline.

The remaining of the paper is organized as follows. In the next section we discuss related work. In Section 3 we present the proposed classification approach for opinion mining, which is evaluated in Section 4. Finally, in Section 5 we present our concluding remarks and possibilities for future work.

2. Related Work

Over the past few years, the growing availability of opinionated content on the Web has fueled the research in sentiment analysis [Godbole et al. 2007], summarization of product reviews [Hu and Liu 2004, Turney 2001, Dave et al. 2003], analysis of blogger mood [Balog et al. 2006] and other opinion mining related tasks [Esuli and Sebastiani 2006, Yu and Hatzivassiloglou 2003]. It has also sparked research on information retrieval applications, and question answering system (for example, using information retrieval techniques to classify opinionated comments posted in forums [Veloso et al. 2007], and question answering techniques to answer opinion questions [Somasundaran et al. 2007]).

Approaches for analyzing and comparing customer reviews and product reputation were presented in [Hu and Liu 2004, Morinaga et al. 2002]. A simple unsupervised learning approach for classifying products and services as *recommended* (thumbs up) or *not recommended* (thumbs down) was proposed in [Turney 2001]. Another approach for semantic classification of product reviews was presented in [Dave et al. 2003]. While these approaches may be related to opinion mining, they are specifically developed to perform product review.

Sentiment analysis of natural language texts is a large and growing field, which can be considered an opinion mining task. Previous work on sentiment analysis relates to techniques to automatically generate sentiment lexicons (i.e., the vocabulary of a language related to a specific sentiment). An example of such techniques was presented in [Hatzivassiloglou and McKeown 1997], in which a list of seed words to determine whether a sentence contains positive or negative sentiments was produced (for instance, *honest* and *intrepid* are seeds of positive connotation, while *disturbing* and *superfluous* are seeds of negative connotation). A dictionary of polarity lexicons to extract positive and negative sentiments from a sentence was presented in [Nasukawa and Yi 2003]. This dictionary was constructed under the assumption that terms with similar orientation tend to co-occur in documents.

Other approaches for opinion mining [Esuli and Sebastiani 2006, Yu and Hatzivassiloglou 2003, Hatzivassiloglou and Wiebe 2000] use results from psychological studies [Bradley and Lang 1999], which found measurable association between words and human emotions. These approaches rely mostly on natural language processing techniques, which are used to determine the semantic orientation of words. Then, the polarity of an opinion is identified based on the words within it, and on their respective semantic orientations. One major problem is that, typically, only few words have the semantic orientation found (some predefined adjectives and adverbs). Further, combinations of different words (i.e., *not good*) are rarely employed by these approaches. Our approach, on the other hand, is based solely on supervised machine learning techniques, which use the vast amount of spontaneously annotated (labelled) opinionated content available in the Web (i.e., reviews from large Web sites). Implicit patterns hidden in sentences and reviews are automatically discovered, and the semantic orientation of some words may arise naturally from the association among these words and the known connotation of the opinion. Further, other evidential information, such as authorship, can be explored transparently. Other classification approaches were also used in opinion mining. In [Pang et al. 2002] a SVM-based approach was used to classify the sentiment associated with subjective sentences. The problem with this approach is that performing classification with SVMs may be slow, due to the high complexity of the kernels that are typically generated. In [Liu et al. 2005] a document classifier was used to extract targets of sentiment expressions in a sentence. However, this approach suffers from poor coverage, due to the huge search space associated with textual content. The proposed approach is based on technique called *lazy (on-demand) associative classification* [Li et al. 2004, Veloso et al. 2006b], in which the classification model is composed of *class association rules* [Liu et al. 1998]. These rules are induced on a demand-driven basis, providing a better coverage of the examples. A simple caching mechanism is used to avoid work replication, making classification much faster. Lazy associative classification has already demonstrate to be extremely effective in important classification tasks, such as document categorization [Veloso et al. 2006a] and spam detection [Veloso and Meira 2006].

3. Classification Approaches for Opinion Mining

Classification is defined as follows. We have an input dataset called the *training data* ($\mathcal{D}_k$) which consists of a set of multi-attribute instances along with a special variable called *label*. The training data is used to build a model which relates the feature

variables of an instance in the training data to the correct label. The *test instances* ($\mathcal{D}_u$) for the classification problem consist of a set of instances for which only the feature variables are known while the label is unknown. The model is used to predict the correct labels for such test instances [3]. Several classification techniques have been proposed over the years, which include neural networks [Lippmann 1987], decision trees [Breiman et al. 1984, Quinlan 1993], support vector machines [Joachims 1998], and associative classification [Liu et al. 1998].

In associative classification, the model is composed of *class association rules* (CARs), which are rules of the form $\mathcal{X} \xrightarrow{\sigma,\theta} c$, where the set $\mathcal{X}$ is allowed to contain only features (i.e., $\mathcal{X} \subseteq \mathcal{I}$, where $\mathcal{I}$ is the set of all possible features), and c is one of the n labels (i.e., $c \in \mathcal{C}$, where $\mathcal{C}$ is the set of all possible labels). A valid CAR has support[4] (σ) and confidence[5] (θ) greater than or equal to the corresponding thresholds, σ_{min} and θ_{min}. Valid CARs depict the association between a combination of features and a label.

There are two approaches for associative classification. In the eager approach a single (very complex) model $\mathcal{M}$ (i.e., a single set of CARs) is generated, and this model is then used to classify all test instances. In the lazy approach, several (very simple) models are generated (one model, $\mathcal{M}_i$, for each test instance i). It has been formally shown that, under the same configuration of σ_{min} and θ_{min}, the lazy approach always outperforms the corresponding eager one [Veloso et al. 2006b]. These two approaches are discussed in the following.

3.1. Eager Associative Classification (Classifiers with Broad Bias)

Common approaches for associative classification mine valid CARs directly from the training data (i.e., using a slightly modified algorithm for association rule mining [Agrawal et al. 1993]). When a sufficient number of valid CARs are found, the model (denoted as $\mathcal{M}$) is finally completed, and it is used to predict the label of the test instances. Due to class overlapping, and since labels are mutually exclusive, CARs may perform contradictory predictions (i.e., different CARs may perform different predictions for the same test instance). To address this problem, we use a probabilistic strategy which basically interprets the classification model, $\mathcal{M}$, as a poll, in which CAR $\mathcal{X} \xrightarrow{\sigma,\theta} c \in \mathcal{M}$ is a vote of weight $\sigma \times \theta$ given by $\mathcal{X}$ for label c [6]. Weighted votes for each label are then summed, and the score of label c is given by the real-valued function s showed in Equation 1. In the end, the label associated with the highest score is finally predicted. Figure 1 shows a sketch with the basic steps of the eager opinion classifier, which is refered to as EOC.

$$s(i,c) = \sum_{\mathcal{X} \xrightarrow{\sigma,\theta} c \in \mathcal{M} | \mathcal{X} \subseteq i} \sigma \times \theta \qquad (1)$$

[3] In the context of opinion mining, each instance corresponds to an opinionated sentence (i.e., a product review or a comment about a story), and a label corresponds to the connotation/polarity of the corresponding opinion. The training data is composed of opinionated sentences for which the connotation is explicitly informed.

[4] The joint probability of $\mathcal{X} \cup \{c\}$ in the training data.

[5] The conditional probability of c given that $\mathcal{X}$ occurs.

[6] Other criteria for weighting the votes can be used.

1. $\mathcal{M} \leftarrow$ all valid CARs in $\mathcal{D}_k$
2. for each opinion $i \in \mathcal{D}_u$ do
3. $\quad \mathcal{M}_i \leftarrow$ all CARs $\mathcal{X} \rightarrow c \in \mathcal{M} \mid \mathcal{X} \subseteq i$
4. $\quad$ perform poll using CARs in $\mathcal{M}_i$
5. $\quad$ predict the winner connotation

Figure 1. Eager Opinion Classifier (EOC).

1. for each opinion $i \in \mathcal{D}_u$ do
2. $\quad d_i \leftarrow \mathcal{D}_k$ after projection based on i
3. $\quad \mathcal{M}_i \leftarrow$ all valid CARs in d_i
4. $\quad$ perform poll using CARs in $\mathcal{M}_i$
5. $\quad$ predict the winner connotation

Figure 2. Lazy Opinion Classifier (LOC).

To facilitate the understanding of eager associative classification in the context of opinion mining, please consider the example in Table 1, used as a running example in this paper. In this illustrative example, each instance corresponds to an opinionated sentence (a product review), and to each sentence is assigned a rating (how good, or bad, the product is). In this case, if we set σ_{min} to 0.30 and θ_{min} to 0.66, then the model $\mathcal{M}$ will be composed of the CARs showed in Figure 3.

	Id	Rating	Opinionated Sentence
Training Data	1	$\star\star\star\star\star$	Perfect first timer's camera
	2	$\star\star\star\star\star$	Perfect, lots of technology
	3	$\star\star\star\star\star$	Perfect, excellent choice!
	4	$\star\star\star\star\star$	Perfect for beginners
	5	$\star\star\star$	Excellent!
	6	$\star\star\star$	Excellent, great pictures
	7	$\star\star\star$	Great camera with an excellent design
	8	$\star\star\star$	Great camera, but not that much
	9	$\star$	Completely disappointing
	10	$\star$	Picture quality was disappointing
Test Set	11	? [$\star\star\star\star\star$]	Perfect camera, great features
	12	? [$\star$]	Zoom is disappointing

Table 1. Training and Test Instances.

Suppose we want to classify sentence 11. In this case, only the first and third CARs are applicable to this instance, since feature *excellent* is not present in instance 11. According to Equation 1, s(11, $\star\star\star\star\star$)=0.40 and s(11, $\star\star\star$)=0.30, and thus rating $\star\star\star\star\star$ is correctly predicted. Now, suppose we want to classify instance 12. In this case, there is no valid CAR, since features *great*, *excellent* and *perfect* are not present in instance 12 (note that there is a strong association between feature *disappointing* and rating $\star$, but $\mathcal{M}$ does not provide such information). In order to generate valid CARs that are applicable to instance 12, σ_{min} should be lowered to 0.20, but in this case the number of valid CARs can be drastically increased, and $\mathcal{M}$ will become extremely complex. In such cases, where no valid CARs are found, the most frequent class (i.e., rating or connotation) is predicted. Next we will present an alternative approach, which generates CARs on a demand-driven basis, depending on the instance being classified, without increasing the complexity of the model (the generated model is, in fact, much simpler).

3.2. On-Demand, Lazy, Associative Classification (Classifiers with Specific Bias)

Typically, eager associative classifiers do not perform well on complex search spaces. This is because they generate CARs before the test instance is even known, and the difficulty in this case is in anticipating all the different directions in which it should attempt to generalize its training examples (i.e., which CARs must be generated). The common eager strategy of using a single value of σ_{min} to restrict the search space for CARs can be problematic, since strong and important associations may be lost due to this absolute cut-off value. Therefore, this strategy can reduce the performance in complex spaces, where not so frequent, but very strong associations may be important to classify specific instances. Lazy classifiers, on the other hand, follow a very specific bias, generalizing the examples exactly as needed to cover a specific test instance. Thus, lazy classifiers are most appropriate when the search space is complex, and there are myriad of ways to generalize a case.

	Id	Rating	Opinionated Sentence
Training Data	9	$\star$	$-$ disappointing
	10	$\star$	$- - -$ disappointing

Table 2. Training Data after Projection based on Instance 12.

In lazy associative classification, whenever a test instance is being considered, that instance is used as a filter to remove irrelevant features and examples from the training data. This process generates a projected training data, d_i, which is focused only on the useful examples for a specific test instance, i. Therefore, there is an automatic reduction of the size and dimensionality of the training data, since irrelevant examples are not considered. As a result, for a given value of σ_{min}, important CARs that are not frequent in the original training data ($\mathcal{D}_k$), may become frequent in the filtered/projected training data (d_i)[7], providing a better coverage of the examples. Since a specific model is generated for each test instance, in the end of the process several different models are generated. However, the models that are induced from the projected training data (i.e., $\mathcal{M}_i$) are much simpler than the model that would be induced from the entire training data (i.e., $\mathcal{M}$). The process of computing weighted votes is basically the same (as shown in Equation 2), except from the fact that all CARs in $\mathcal{M}_i$ are applicable to instance i, since only relevant features are considered during lazy enumeration of CARs. Figure 2 shows a sketch with the basic steps of the on-demand, lazy, opinion classifier, which is refered to as LOC.

$$s(i,c) = \sum_{\mathcal{X} \xrightarrow{\sigma,\theta} c \in \mathcal{M}_i} \sigma \times \theta \tag{2}$$

To illustrate how LOC works, suppose again that we want to classify instance 12. The first step is to project the training data based on the features present in instance 12, forming d_{12} which is shown in Table 2. As can be seen, only two examples are relevant to this instance. From the filtered training data, only one CAR is found, as shown in Figure 4. According to Equation 2, $s(12, \star)$=1.00 and therefore rating $\star$ is correctly predicted.

[7]Note that the absolute value of σ_{min} (which is $\sigma_{min} \times \mid d_i \mid$) may change according to the size of d_i. Thus, different test instances may imply in different cut-off values.

1. perfect $\xrightarrow{0.40,1.00} \star\star\star\star\star$
2. great $\xrightarrow{0.30,1.00} \star\star\star$
3. excellent $\xrightarrow{0.30,0.75} \star\star\star$

Figure 3. CARs induced from the Entire Training Data ($\mathcal{M}$).

1. disappointing $\xrightarrow{1.00,1.00} \star$

Figure 4. CAR induced from the Filtered Training Data, showed in Table 2.

3.2.1. Caching Common CARs

Processing a CAR has a significant computational cost, since it involves accessing the training data (which can be very large). Different instances may induce different models (i.e., a set of CARs), but different models may share common CARs. In this case, caching is very effective in reducing work replication.

Our cache is a pool of entries, and each entry has the form $<key, data>$, where key=$\{\mathcal{X}, c\}$ and $data$=$\{\sigma, \theta\}$. Our implementation has a limited storage and stores all cached CARs in main memory. Before generating a CAR $\mathcal{X} \xrightarrow{\sigma,\theta} c$, the classifier first checks whether this CAR is already in the cache. If an entry is found with a key matching $\{\mathcal{X}, c\}$, the CAR in the cache entry is used instead of processing it. If it is not found, the CAR is processed and then it can be inserted into the cache.

The cache size is limited, and when the cache is full, some CARs must be discarded to make room for other ones. The replacement heuristic is based on the support of CARs. More specifically, the least frequent CAR in the cache is the first to be discarded (and it will only be discarded if the CAR to be inserted is more frequent than it). There are two main reasons to adopt this heuristic. First, the more frequent a CAR is, the higher is the chance of using this CAR for classifying other test instances. Second, the computational cost associated with generating more frequent CARs is higher than the cost associated with generating less frequent ones (more frequent CARs necessitates more accesses to the training data). We show empirically that caching CARs is extremely effective in reducing the computation time for lazy opinion mining.

4. Experimental Evaluation

In this section we describe and analyze the experimental results for the evaluation of the proposed opinion mining approaches in terms of both classification effectiveness and computational efficiency. Our evaluation is based on a comparison against the current state-of-the-art SVM-based approach [Pang et al. 2002], which is used as our standard baseline. We first present the application scenarios employed, and then we discuss the effectiveness and the computational efficiency of our approach in these applications.

4.1. Application Scenarios

User reviews and moderated comments where authors and moderators provide quantitative and qualitative opinions about products or comments, are perfect for training and testing a classifier for opinion mining. The evaluation is based on two actual application scenarios, which are described next:

- Slashdot.org: Several stories are published every day in the Slashdot forum. Readers of the forum have the ability to post comments about specific stories. Each comment has an author, a title and a text. More than a forum for publishing stories, Slashdot constitute a large social network, where users may interact with each other. Depending on the comments that were posted by a certain user, she/he may acquire fans, friends or enemies throughout her/his existence as a participant of Slashdot [8]. This interaction among users may result in communities and groups of users that share similar opinions (i.e., friends or fans) or not (i.e, enemies). All comments are manually classified according to the opinion of a moderator, and fall in one of 8 connotations: informative, insightful, interesting, funny, redundant, troll, off topic or flamebait. We collected a set of 8 stories about politics and 9 stories about science. The corpora are composed of moderated comments that were posted to the forum in response to these stories. Features within each comment include the words in the text of the comment, and the author. Further, since the opinion of the moderator can be influenced by the relationships of the author (i.e., the moderator may be a fan of an enemy of the author), we also include friends, fans and enemies of the author in the feature set.
- Amazon.com: Amazon allows users to input a (long) text review, a title and one scalar rating per product (number of stars). The corpora are composed of three components: 7 years of reviews about a movie (Star Wars Trilogy), 4 years of reviews about a book (The Davinci Code), and one year of reviews about a specific camera. Features within each review include the words in the text of the review and in the title.

In all corpora we strip out HTML tags and removed stop words. Also, tokens that only occurs once were discarded. Table 3 shows the number of reviews for each product and the number of comments for each subject topic. Table 4 shows the proportion of comments and reviews associated with each connotation or rating. Since no comments with troll, off topic and flamebait connotations were posted, we do not include these connotations.

Slashdot.org		Amazon.com		
Politics	Science	Movie	Book	Camera
3,432 comments	2,556 comments	2,165 reviews	3,461 reviews	1,084 reviews
8 stories	9 stories	7 years	4 years	one year

Table 3. Number of Comments and Reviews.

4.2. Results

In all experiments with the aforementioned corpora, we used 10-fold cross-validation and the final results of each experiment represent the average of the ten runs. We quantify the classification effectiveness of the various approaches through the conventional precision, recall and accuracy measures. Precision p is defined as the proportion of correctly classified reviews/comments in the set of all reviews/comments. Recall r is defined as the proportion of correctly classified reviews/comments out of all the opinions having the target rating/connotation. Traditional accuracy were applied to quantify single classification

[8] Fans, friends and enemies of a particular user are explicitly informed by Slashdot.

Slashdot.org			Amazon.com			
Connotation	Politics	Science	Rating	Movie	Book	Camera
Interesting	0.18	0.17	$\star$	0.20	0.20	0.02
Insightful	0.50	0.26	$\star\star$	0.07	0.11	0.01
Informative	0.18	0.16	$\star\star\star$	0.10	0.14	0.06
Funny	0.13	0.37	$\star\star\star\star$	0.13	0.16	0.16
Redundant	0.02	0.04	$\star\star\star\star\star$	0.50	0.39	0.75

Table 4. Proportion of Comments and Reviews Associated with each Connotation or Rating.

effectiveness values over all classification tasks. The computational efficiency is evaluated through the total execution time, that is, the processing time spent in training and classifying all comments or reviews. For EOC and LOC we set σ_{min}=0.005, θ_{min}=0.80, and for SVM polynomial kernels of degree 8 were used [9]. The experiments were performed on a Linux-based PC with a Intel Pentium III 1.0 GHz processor and 1.0 GBytes RAM. All the results to be presented were found statistically significant at the 99% confidence level when tested with the two-tailed paired t-test.

Table 5 shows precision and recall numbers obtained from the execution of EOC on the Slashdot corpora (Politics and Science). As expected, better results were obtained in more frequent connotations (*Insightful* for Politics, and *Funny* for Science). On the other hand, results obtained in low frequent connotations (*Redundant*) are very poor. This is because for the value of σ_{min} used, there is almost no CAR predicting connotation *redundant*. In fact, this is also the main explanation for the low recall numbers achieved by EOC. More specifically, applying a single value minimum support cut-off may lead to the loss of important and strong CARs, that are not as frequent as σ_{min}. This problem is worsened due to the skewness distribution of connotations.

	Politics		Science	
	Prec	Rec	Prec	Rec
Interesting	0.69	0.64	0.63	0.60
Insightful	0.76	0.73	0.72	0.77
Informative	0.59	0.52	0.61	0.57
Funny	0.37	0.33	0.72	0.68
Redundant	0.00	0.00	0.50	0.10

Table 5. Precision and Recall Numbers for Slashdot Corpora, using EOC.

Table 6 shows precision and recall numbers obtained from the execution of LOC on the Slashdot corpora (Politics and Science). As we can see, there is a great improvement, specially in terms of recall. Low frequent, but strong, associations are captured by LOC because the absolute value of σ_{min} is automatically adjusted according to the test instance being classified (i.e., the training data is projected according to the feature in the test instance). This result shows that generating CARs on a demand-driven basis is a very effective approach.

[9] These parameters yield the best performance in a validation step.

It is worth noting that, although *Interesting* and *Funny* connotations in the Politics corpus are relatively frequent, their language seems to be often more varied, and thus, achieving good recall on these connotations is more difficult.

	Politics		Science	
	Prec	Rec	Prec	Rec
Interesting	0.75	0.72	0.73	0.69
Insightful	0.80	0.82	0.77	0.78
Informative	0.72	0.66	0.69	0.67
Funny	0.62	0.55	0.79	0.81
Redundant	0.75	0.20	0.50	0.25

Table 6. Precision and Recall Numbers for Slashdot Corpora, using LOC.

Table 7 shows precision and recall numbers obtained from the execution of EOC on the Amazon corpora (Movie, Book and Camera). The results show that EOC is not suitable for problems presenting highly skewed distribution of connotations, such as the one observed in the Camera corpus. For less frequent ratings, there is no valid CAR for the value of σ_{min} that was employed. In these cases, the most frequent rating (i.e., $\star\star\star\star\star$) is predicted by default, and thus extremely low values of precision and recall are achieved.

	Movie		Book		Camera	
	Prec	Rec	Prec	Rec	Prec	Rec
$\star$	0.83	0.78	0.81	0.73	0.00	0.00
$\star\star$	0.72	0.63	0.70	0.65	0.00	0.00
$\star\star\star$	0.76	0.61	0.70	0.68	0.00	0.00
$\star\star\star\star$	0.72	0.64	0.72	0.74	0.66	0.32
$\star\star\star\star\star$	0.84	0.82	0.77	0.74	0.83	0.77

Table 7. Precision and Recall Numbers for Amazon Corpora, using EOC.

Table 8 shows precision and recall numbers obtained from the execution of LOC on the Amazon corpora (Movie, Book and Camera). Again, great improvements were observed (in relation to EOC), specially in the Camera corpus. This shows that LOC is a great alternative when the connotations/rating follow a skewed frequency distribution. It is also important to note that the reviews from Amazon corpora are apparently easier to classify than the comments in the Slashdot corpora. This is at least in part because of the generally longer size of the reviews.

Table 9 shows the comparison between different classification approaches. Lazy approaches (i.e., LOC) learn quickly but classify slowly, while eager approaches (i.e., EOC and SVM) learn slowly but classify quickly. However, the use of caching is extremely useful for speeding up lazy classification. EOC was faster than LOC only in Book and Camera corpora. Its effectiveness, however, was much worse than the effectiveness obtained by LOC. The SVM approach is always more accurate than EOC, but it is also always much slower than EOC. LOC showed the best accuracy numbers in all corpora used, and it is also the fastest approach in Politics, Science and Movie corpora. This is because its eager counterpart, EOC, spent much time generating a large number

	Movie		Book		Camera	
	Prec	Rec	Prec	Rec	Prec	Rec
⋆	0.85	0.96	0.87	0.94	0.80	0.20
⋆⋆	0.76	0.70	0.74	0.68	0.50	0.25
⋆⋆⋆	0.82	0.71	0.79	0.71	0.62	0.41
⋆⋆⋆⋆	0.79	0.78	0.79	0.82	0.78	0.44
⋆⋆⋆⋆⋆	0.90	0.94	0.82	0.84	0.85	0.80

Table 8. Precision and Recall Numbers for Amazon Corpora, using LOC.

of irrelevant CARs (i.e., CARs that were not used to classify any test instance), hurting computational performance. LOC, on the other hand, generates only useful CARs, since they are generated on a demand-driven basis.

	EOC	LOC	SVM	EOC	LOC	SVM
Politics	0.58	0.71	0.65	324 secs	292 secs	4,183 secs
Science	0.61	0.73	0.67	461 secs	367 secs	3,499 secs
Movie	0.72	0.86	0.79	627 secs	492 secs	5,243 secs
Book	0.68	0.81	0.75	458 secs	515 secs	3,757 secs
Camera	0.59	0.72	0.69	212 secs	282 secs	2,394 secs

Table 9. Accuracy Numbers and Execution Times for Amazon and Slashdot Corpora.

The computational performance of LOC was further evaluated. Table 10 depicts the execution times obtained by employing different cache sizes. We allowed the cache to store from 0 to 100,000 CARs (approximately 73 MBytes), and for each storage capacity we obtained the corresponding execution time. As expected, execution time is sensitive to cache size, showing improvements of about 300% for larger cache sizes. Similar trends were observed in all corpora.

Cache Size (#CARs)	Politics	Science	Movie	Book	Camera
0	782 secs	1,067 secs	1,383 secs	1,577 secs	685 secs
1,000	649 secs	828 secs	1,177 secs	1,353 secs	593 secs
10,000	327 secs	418 secs	557 secs	593 secs	341 secs
50,000	298 secs	373 secs	495 secs	519 secs	286 secs
100,000	292 secs	367 secs	492 secs	515 secs	282 secs

Table 10. Execution Times for Different Cache Sizes.

To finish our evaluation, we show some advantages of statistical based approaches, such as LOC and SVM, when compared with semantic based approaches. Table 11 shows some discriminative words, discovered during the execution of LOC in the Amazon corpora. Typically, semantic based approaches make use of the polarity orientation of predefined adjectives and adverbs to classify opinions. However, as we can see in Table 11, not only adjectives and adverbs are useful for sake of classification. Words with apparently no semantic orientation, such as “resolution”, is listed as a top positive feature in the Camera

corpus. This is because this word appears in a large portion of the camera reviews, and most of those are positive. This suggests that semantic based approaches can be improved when combined with statistical based approaches.

	Movie	Book	Camera
⋆	garbage	disappointing	caution
⋆⋆	problem	not too	zoom
⋆⋆⋆	audio	controversial	over-rated
⋆⋆⋆⋆	helpful	not based	beautiful
⋆⋆⋆⋆⋆	wow	perfect	resolution

Table 11. Some Discriminative Words or Sentences Associated with Different Ratings in the Amazon Corpora.

5. Conclusions and Future Work

Opinion mining is an emerging discipline concerned with the opinion a document expresses. Opinion-driven content management has several important applications, such as determining critics' opinions about a given product by classifying online product reviews, or tracking the shifting attitudes of the general public towards a political subject matter by mining online forums.

Fully analyzing and classifying opinions involve tasks that relate to some fairly deep semantic and syntactic analysis of the text. However, in this paper we showed that appropriate statistical analysis of opinionated text can provided an effective approach for opinion mining. We proposed a basic approach (EOC) based on associative classification, which makes use of *class association rules* (CARs) to classify opinionated sentences. This basic approach, however, provides low recall as observed in the experiments. Further investigation revealed that the reason for low recall numbers is the use of a single minimum support cut-off (i.e., σ_{min}), which may lead to the loss of strong and important associations (that are not so frequent). Lowering the value of σ_{min} would discover these strong associations, but in this case the generated model becomes huge. We proposed an alternative approach (LOC), which generates CARs on a demand-driven basis, in which the inductive process of generating CARs is delayed until the test instance is known, so that CARs are generated specifically to this instance. We are able to achieve fairly good results with LOC. It achieves much better results, and in some cases is even faster than EOC (with the use of a simple caching mechanism). Further evaluation showed that LOC is also superior than the state-of-the-art opinion mining approach which is based on SVMs, both in terms of accuracy (9% of improvement) and computational performance (much faster). This is a valuable advance with respect to the state of the art.

There is room for improvement. Combining other language modeling approaches, such as the semantic orientation of terms, might lead to further improvements in accuracy (as suggested in the experimental section), and we intend to investigate this strategy in future work. Furthermore, some opinionated content may present more than one connotation simultaneously. For instance, a camera can be reviewed based on several features (i.e., resolution, battery, zoom, size etc.), and each of these features may receive different ratings. This is an example of multi-labelled classification problem (which are more

complicated than traditional classification problems), and we are considering to address this problem in the context of opinion mining.

References

Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proc. of the Int. Conf. on Management of Data, SIGMOD*, pages 207–216, Washington, USA. ACM Press.

Balog, K., Mishne, G., and de Rijke, M. (2006). Why are they excited? Identifying and explaining spikes in blog mood levels. In *Proc. of the Conf. of the European Chapter of the Association for Computational Linguistics*. The Association for Computer Linguistics.

Bradley, M. and Lang, P. (1999). Affective norms for english words (ANEW): Stimuli, instruction manual and affective ratings. (Technical Report C-1, The Center of Research in Psychophysiology).

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and regression trees. *Wadsworth Intl.*

Dave, K., Lawrence, S., and Pennock, D. (2003). Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proc. of the Intl. Conf. on World Wide Web*, pages 519–528. ACM Press.

Esuli, A. and Sebastiani, F. (2006). Determining term subjectivity and term orientation for opinion mining. In *Proc. of the Conf. of the European Chapter of the Association for Computational Linguistics*. The Association for Computer Linguistics.

Godbole, N., Srinivasaiah, M., and Skiena, S. (2007). Large-scale sentiment analysis for news and blogs. In *Proc. of the Intl. Conf. on WebLogs and Social Media*.

Hatzivassiloglou, V. and McKeown, K. (1997). Predicting the semantic orientation of adjectives. In *Proc. of the Conf. on European Chapter of the Association for Computational Linguistics*, pages 174–181. Association for Computational Linguistics.

Hatzivassiloglou, V. and Wiebe, J. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *Proc. of the Conf. on Computational Linguistics*, pages 299–305. Association for Computational Linguistics.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proc. of the Intl. Conf. on Knowledge Discovery and Data Mining*, pages 168–177. ACM Press.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proc. of the European Conf. on Machine Learning*, pages 137–142.

Li, J., Dong, G., Ramamohanarao, K., and Wong, L. (2004). Deeps: A new instance-based lazy discovery and classification system. *Machine Learning*, 54(2):99–124.

Lippmann, R. (1987). Introduction to computing with neural nets. *IEEE ASSP Magazine*, 4(22).

Liu, B., Hsu, W., and Ma, Y. (1998). Integrating classification and association rule mining. In *Knowledge Discovery and Data Mining*, pages 80–86.

Liu, B., Hu, M., and Cheng, J. (2005). Opinion observer: analyzing and comparing opinions on the web. In *Proc. of the Intl. Conf. on World Wide Web*, pages 342–351. ACM Press.

Morinaga, S., Yamanishi, K., Tateishi, K., and Fukushima, T. (2002). Mining product reputations on the web. In *Proc. of the Intl Conf. on Knowledge Discovery and Data Mining*, pages 341–349. ACM Press.

Nasukawa, T. and Yi, J. (2003). Sentiment analysis: capturing favorability using natural language processing. In *Proc. of the Intl. Conf. on Knowledge Capture*, pages 70–77. ACM Press.

Ounis, I., de Rijke, M., Macdoanld, C., Mishne, G., and Soboroff, I. (2006). Overview of the TREC 2006 blog track. In *Proc. of the Text Retrieval Conference, TREC*. NIST.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pages 79–86. Association for Computational Linguistics.

Quinlan, J. (1993). *C4.5: Programs for Machine Learning*. M. Kaufmann.

Somasundaran, S., Wilson, T., Wiebe, J., and Stoyanov, V. (2007). QA with attitude: Exploting opinion type analysis for improving question answering in on-line discussions and the news. In *Proc. of the Intl. Conf. on WebLogs and Social Media*.

Turney, P. (2001). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proc. of the Annual Meeting on Association for Computational Linguistics*, pages 417–424. Association for Computational Linguistics.

Veloso, A. and Meira, W. (2006). Lazy associative classification for content-based spam detection. In *Proc. of the Latin American Web Congress*, pages 154–161. IEEE Computer Society.

Veloso, A., Meira, W., Cristo, M., Gonçalves, M., and Zaki, M. (2006a). Multi-evidence, multi-criteria, lazy associative document classification. In *Proc. of the Intl. Conf. on Information and Knowledge Management*, pages 218–227. ACM Press.

Veloso, A., Meira, W., Macambira, T., Guedes, D., and Almeida, H. (2007). Automatic moderation of comments in a large on-line journalistic environment. In *Proc. of the Intl. Conf. on WebLogs and Social Media*.

Veloso, A., Meira, W., and Zaki, M. J. (2006b). Lazy associative classification. In *Proc. of the Intl. Conf. on Data Mining*, pages 645–654. IEEE Computer Society.

Yu, H. and Hatzivassiloglou, V. (2003). Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pages 129–136. Association for Computational Linguistics.