# A Study of Opinion Mining and Visualization of Hotel Reviews

Eivind Bjørkelund
Dept. of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway

Thomas H. Burnett
Dept. of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway

Kjetil Nørvåg*
Dept. of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway

## ABSTRACT

Travel websites like TripAdvisor are nowadays important tools for travelers when deciding which hotels to stay in, and what restaurants and tourist attractions to visit. In this paper, we study opinion mining applied on data from travel review sites. We also describe how the results of sentiment analysis of textual reviews can be visualized using Google Maps, providing possibilities for users to easily detect good hotels and good areas to stay in. More advanced features also provides for faceted and filtered visualization. An evaluation of the techniques presented, shows high accuracy in opinion mining, and that the prototype can help detect hotel features and possible reasons for changes in opinion as well as show "good" and "bad" geographical areas based on hotel reviews.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

## General Terms

Experimentation

## Keywords

Semantic analysis, opinion mining, feature extraction, temporal opinion mining

## 1. INTRODUCTION

Travel websites like TripAdvisor are nowadays important tools for travelers when deciding which hotels to stay in, and what restaurants and tourist attractions to visit. The contents on such travel websites is user-generated, thus giving access to the opinions of many individuals. When contributing opinions to the travel websites, users typically select grades for a a number of facets (cleanliness, location, etc), and additionally add a textual review. During subsequent search, giving a particular location, users get a ranked list of hotels, where ranking is based on the grades given by previous travelers. It is also possible to get the hotels and other sites marked on a map.

When studying existing travel websites and previous research in the area, two observations can be made: 1) the visualization on map is quite primitive just showing the location of the hotels, and 2) the only use of the textual descriptions is for browsing, they are not part of the ranking process or visualized. To our knowledge, these issues have not been studied before. In this paper we also describe how to use opinion mining techniques to analyze changes in opinions about hotels. Hotels as a customer-based service is an area where multiple factors may impact customer sentiment. For instance noise, nearby construction, weather, even customer expectations. Such events may influence the overall sentiment at any given time, creating a dynamically changing sentiment. Managing to identify why changes occur in such a setting, may provide both customers and hotel owners valuable information regarding the interpretation of large amounts of opinionated data. Typical scenarios might include 1) how to detect up and coming areas, 2) study of sudden changes in hotel sentiment, and 3) finding hotels with the best breakfast.

Opinion mining tools are used to identify and extract subjective information from user reviews, and then to determine the sentiment of the text. Three different techniques are studied: feature extraction, burst detection and visualization. Feature extraction is a technique to identify and extract product features, burst detection is used to detect and analyze abnormal changes and visualization means to present the information graphically. Evaluation is performed by comparing the actual review scores with our sentiment scores. The initial sources chosen as a basis for the data set were Booking.com[1] and TripAdvisor[2]. They have large databases of hotels along with corresponding reviews.

In order to perform visualization experiments, a web prototype was created, and can be seen in Figure 1. This provides a way to detect "good" and "bad" areas based on hotel reviews in a user-friendly map view. Additionally, it includes a feature search, where users can find hotels based on features from user reviews. These scores are calculated based on user opinions, and is an effective way for users to filter sentiment data. For commercial use, the prototype can help analyze the massive amount of hotel information published each day by customers, and can help hotel managers analyze their products. It can also be used as a more advanced hotel search engine where users can find extra information in a map user interface.

The main contributions of this paper are as follows: 1) a study

*Email of contact author: Kjetil.Norvag@idi.ntnu.no


*iiWAS2012,* 3-5 December, 2012, Bali, Indonesia.


[1]http://www.booking.com/
[2]http://www.tripadvisor.com/

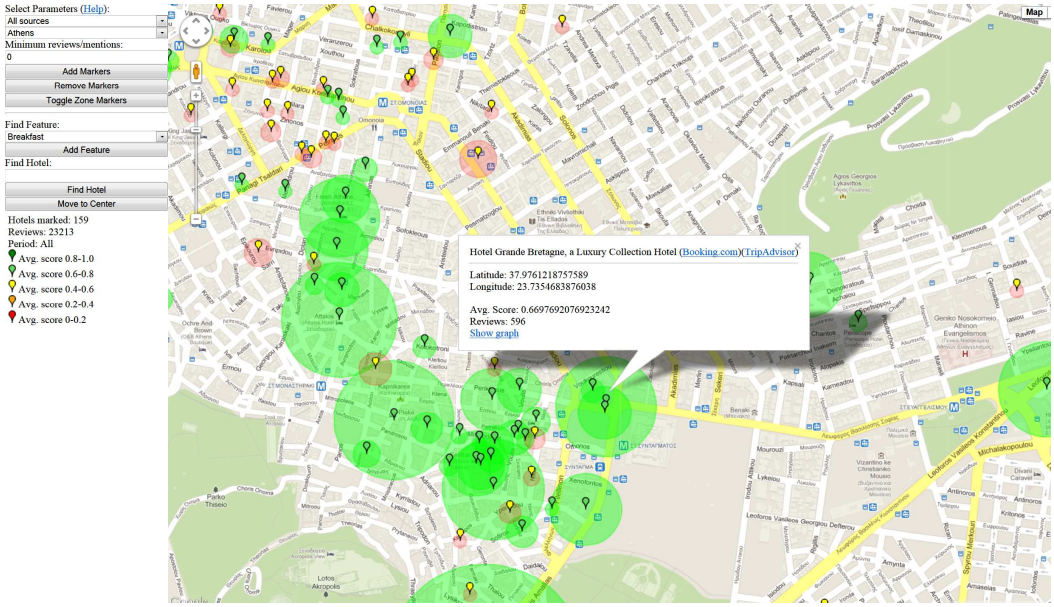


**Figure 1: Screenshot of our prototype.**[3]

of techniques for opinion mining of hotel reviews, 2) combining opinion mining with spatial aspects, 3) visualization and a proof-of-concept prototype, 4) development of a hotel-review dataset, and 5) an extensive experimental study of accuracy of the techniques in a hotel review context.

The organization of the rest of the paper is as follows. In Section 2, we give an overview of related work. In Section 3, we outline aspects of mining opinions in hotel reviews. In Section 4 we describe how to visualize such opinions. In Section 5, we evaluate our proposed techniques. Finally, in Section 6, we conclude and outline future work.

## 2. RELATED WORK

In this section we focus in particular on the main topics of this paper, i.e., temporal opinion mining and visualization. For a more general overview of opinion mining we refer to the excellent survey by Pang and Lee [17].

The basic process of temporal opinion mining involves determining the average opinion on a given topic at two or more unique moments in time. Ideally this results in a complete time series, which is then mapped out with changes over time. The timeline can be presented by the predominant polarity [8] or as a graph based on sentiment value [6]. Any changes in opinion can then be identified, and may be used to find patterns [2]. This is one of main advantages compared to basic opinion mining. However, changes in opinions are by themselves not necessarily useful without something to compare them to. Therefore the usefulness of change detection becomes more apparent when combined with understanding why said change occurred.

To determine why an opinion has changed, one needs to find correlations between opinions and the textual content of a data set. This can be done by determining a set of association rules from each collected data set. These rules match textual tokens and map them to a certain opinion. This gives a set of tokens which are determined to belong to a given opinion. With enough such rules, a system can be taught to determine subjectivity of textual data sets. When utilizing these rules and opinions when comparing data sets, it is theoretically possible to determine patterns in any changes of opinion. Different change patterns can be defined in several ways, but they mostly relate to whether or not changes match the rules for each data set.

Change is defined as either gradual or sudden, and based on any determined rules, expected or unexpected [2]. Gradual or sudden describe the degree in which an opinion has changed. An expected change is a change which corresponds to the rules for each data set. An unexpected change does not correspond to the specified rules. For instance, if two data sets have very similar rules, but completely different opinions attached to them, the change might be unexpected.

Fukuhara et al. [6] considered news and blog articles and produced two sets of graphs: a topic graph and an emotion graph. The topic version graphed out topics associated with a certain sentiment. With a specific sentiment, it was possible to see when certain events were highly associated with that sentiment. As such the graph illustrated which events had the greatest impact on the given sentiment at specific moments in time. The emotional graphs showed a range of sentiments over time regarding a given event. This approach is the more common way to illustrate temporal opinion mining results; having a specific topic or event, and seeing how sentiment toward it changes and fluctuates as time goes by. Combining the analysis of these two types of graphs, the result is a solid visual representation of the association between sentiment and event. However, this approach requires both sentiment and event to be known prior to analysis. It does not attempt to discover and identify previously unknown events which might have had an effect on sentiment.

Das et al. developed a prototype system to create visualizations of opinions over time and track changes, focuses on temporal relations between events associated with sentiments [4]. They employed a machine learning approach based on Conditional Random Field for solving the identification problem of event-event relations. Like the approach by Fukuhara et al., this differs from our approach

[3]The prototype is available from:
`http://research.idi.ntnu.no/wislab/vito/`

since it does not attempt to find unknown events based on changes in sentiment.

Prior to the approach by Das et al., a system called MoodViews was presented by Mishne et al. [12]. The system tracks the mood of blogs hosted by LiveJournal. Conceptually similar to [4], the system continuously downloads updates from thousands of blogs. It tracks moods, predicts what they might become and analyses them in an attempt to understand why specific changes in mood occur. The mood tracker follows moods in close to real time and creates graphs based on these time series. The mood predictor combines natural language processing with machine learning to estimate moods based purely on textual content. This is then compared to the actual mood data gathered from tagging and an accuracy statistic can be determined. Using language statistics it finally identifies terms which occur more often or less often than usual during certain peaks in mood.

Another approach to visualizing hotel customer feedback is presented in [21], based on opinion wheel and tag cloud diagram. However, they don't include the spatial aspect in the visualization like we do.

# 3. MINING OPINIONS IN HOTEL REVIEWS

This section describes the approaches utilized for analyzing opinion. The goal is to find efficient methods for extracting the semantic context of documents. The section starts with describing some of the challenges with opinion analysis, then presents the two main approaches: knowledge-based and supervised. Other operations and techniques related to sentiment analysis like unsupervised learning, language models and feature extraction will also be presented.

Additionally, temporal aspects will be studied. This part will describe ways to detect changes in opinion over time, describing two aspects: burst detection and opinion visualization.

## 3.1 Known Challenges

Most of the challenges in opinion mining are related to the authenticity of extracted data and the methods used [17]. Often a document contains sentences with mixed views. For example, assume a news article about two different companies, say Microsoft and Apple, contains positive news about Microsoft, but negative news about Apple. Should this text be classified as positive or negative? Or maybe neutral? Another issue is that a word could be considered positive in one situation and negative in another situation. For example the word rise, can be considered both positive and negative depending on the context. If the costs rise, that is definitely negative for the company, but if the value of the company or the revenue rises then that is positive. This example is maybe more of a general text mining problem than pure opinion mining, but the same will apply to subjective opinions.

To only look at opinions as simply negative or positive is one form of evaluation, but it overlooks the comparing factor. Comparisons may be objective or subjective. For example, an objective sentence is "this cellphone is 10 grams heavier than iPhone 4S". This is a stated fact, and does not necessarily have an impact on which cellphone is the better one. A subjective sentence can for example be the sentence "this cellphone is better than Sony Ericsson Xperia Play, but worse than iPhone 4S". Is this a positive sentence? Or a negative sentence? Clearly, the cellphone is regarded, according to the author, better than Sony Ericsson Xperia Play, so that is a positive evaluation, but worse than iPhone 4S, which is a negative statement.

Additionally, opinions may change over time. If a lot of customers complain about a product, the company will eventually take notice and try to fix it. In other words opinions can be outdated after some time. Other challenges include misleading opinions like sarcasm and irony and that people have different writing styles. A person can, for example, use a word that can be regarded negative for some, but neutral or even positive for others.

Determining subjectivity and sentiment of a document is one thing. Finding the general sentiment in huge collections of data sets is quite another. What one person thinks of a product is often not interesting. What 10,000 people think of the same product almost always is. Due to this, along with the sheer amount of information available at times, the need to summarizing opinions [1] regarding given topics arises.

Another problem is domain dependence, that is that the general notion of positive and negative opinions can be different depending on the domain. For example the sentence "go read the book" can be a positive sentence in book reviews, but negative in movie reviews. One way to deal with this was provided by Yang et al., who saw if the features were good domain-independent indicators by checking that the features were good in two different domains, in this case movie reviews and product reviews [22].

It is also worth mentioning that human classification has around 70% correctness because human raters typically agree about 70% of the time [11]. Thus, a system that has around 70% accuracy is as good as human raters, even though it may not sound too impressive. If a program were "right" 100% of the time, the average human would still disagree with it around 30% of the time.

## 3.2 Knowledge-based Approaches

A number of approaches to opinion mining take the effort of first creating an opinion lexicon. This can be done in many ways. The simplest method is to manually decide the degree of positivity and negativity of words, and then have some way of calculating the sentiment for each sentence or whole text based on the values of the words. This can of course be tricky because it is difficult to set a sentiment value on a simple word, since many words can be both positive and negative depending on the context. Adding word phrases and have some form of word disambiguation can help improve the results, but it will still be extremely time consuming doing this manually.

In the other extreme, one can create an opinion lexicon in an unsupervised manner. An example of that can be found in [20], where adjectives and adverbs are first extracted. Then the semantic orientation is decided by an algorithm that uses mutual information as a measure of the strength of semantic association between two words [3]. The last step is to calculate the average semantic orientation of phrases in the given text based on the created lexicon, and classify the text as positive or negative.

An approach that combines both manually and automatically labeling is SentiWordNet [5]. SentiWordNet is a publicly available lexical resource for opinion mining and is freely available for research purposes. SentiWordNet is based on the lexical database of WordNet, and automatically annotates each synset of WordNet according to three sentiment scores: positivity, negativity, objectivity, describing how positive, negative and objective the terms contained in the synset are. In SentiWordNet 3.0 only the positive and negative scores are included in the database, so the objectivity score is calculated as: $ObjScore = 1 - (PosScore + NegScore)$.

Terms may also have different senses, and thus possibly different sentiment properties. For example the synset "estimable" with the sense "may be computed or estimated" has 1.0 in objectivity and 0.0 in negativity and positivity. Another synset "estimable" with the sense "deserving of respect or high regard" has a positivity score of 1.0, and negativity and objectivity of 0.0. Each of the three scores range in the interval from 0.0 to 1.0, and their sum is 1.0 for

each synset. Additionally, a synset may have a non zero score for each of the categories, which means that a synset may have some degree of each of the three opinion-related properties. For instance, the synset "adventurous" with the meaning "willing to undertake or seeking out new and daring enterprises" has a 0.625 positive score, 0.25 negative score and 0.125 in objectivity.

Interestingly, sentiment keywords are best determined automatically by machine learning, which SentiWordNet for the most part does. Manual selection has proven to be less effective. In a study by Pang [16], they found that keyword lists based on statistical information of the selected data set provided a better result than manual selection. This was found to be true even though the length of the keyword lists were the same. The manually created lists resulted in an average accuracy of roughly 60%, while the statistics based result averaged to around 70%.

A few studies exist which utilize SentiWordNet in relation to opinion mining techniques, among them [15] for sentiment classification of film reviews. It got an overall accuracy between 65.85% with regular term counting and 69.35% with a linear support vector machine classifier with scores used as features.

We use SentiWordNet 3.0 in our application which is based on WordNet 3.0. Each adjective, adverb, subjective and noun in a review are looked up in the SentiWordNet lexicon, and their scores are subsequently added. If the total positive score is larger than the negative score, the review is marked as positive and vice versa if the negative score is larger than the positive score. If the two scores are equal, it is deemed objective or neutral.

Since SentiWordNet outputs sentiment value, it is easy to classify the degree of sentiment. In other words, it is easy to classify the reviews in more than two or three categories. For example we tested with five categories: strong positive, weak positive, neutral, weak negative and strong negative.

## 3.3 Feature Extraction and Summary

Evaluating a text at the document level and evaluate that document as a whole has some disadvantages. For example a negative evaluation as the document as a whole does not necessarily mean that everything that is mentioned in that document is negative. There can be some specific aspect about that particular object that is positive. Likewise, a positive evaluation does not mean that the author dislikes everything about the object. For instance, in hotel reviews an author usually writes both positive and negative aspects about that hotel, even though the overall sentiment of the review can be either positive or negative. To obtain such detailed aspects, we will have to go to the sentence level and extract the interesting features.

There are basically two different formats that we will have to deal with:

1. Format 1 - Pros and cons: The reviewer is asked to describe pros and cons separately. This is the review structure in Booking.com.

2. Format 2 - Free format: The reviewer can write freely. In other words there is no specific separation of pros and cons. This is the review structure in TripAdvisor.

For format 1, only the features have to be identified since the semantic orientation is already known, but in format 2 both the features and their opinion orientations have to be found. One of the simplest way to identify features is to rely on the simple notion that features are usually expressed as nouns or noun phrases. This is obviously not always true, for instance, the verb "use" in the sentence "difficult to use". However, in most cases the features are nouns [10]. More advanced methods used to identify features includes association mining and rule mining, but also different heuristic methods [7] [14] [23]. After a feature has been found, it together with its opinion have to be extracted. This can be done by a simple heuristic to extract adjectives that appear in the same sentence as the features [7] or with more advanced procedures like manually or semi-automatically developed language rules [18]. Also sentiment lexicons can be used [13].

In our case we use six predefined features: "breakfast", "staff", "service", "clean", "location" and "internet". These six features are a combination of features from Booking.com and TripAdvisor which a user can grade separately. These features are used together with part of speech tag patterns and a synonym check. We also include cases where the feature is a verb, adverb, adjective and of course a noun. That way most combinations of the features are covered. This is a simplified identification process of features, but it does cover the most important ones, since we also cross-checked them with the two review sites. For extraction we keep the sentence or clause where the feature was included, and then uses SentiWordNet as a sentiment lexicon to determine the sentiment of the feature.

So far, we have talked about analyzing and extracting semantic information from documents. The next step will be to present the opinion information in an orderly fashion. For instance, it might be desirable to summarize the main point in a single review or summarize the main points from a lot of reviews of the same product or service. This is called summarization. There are many different kinds of approaches to summarization. Everything from recapitulation of a single document to more advanced summarization where multiple documents about the same topic shall be summarized. This includes techniques for detecting redundancy, identifying important differences among documents and ensuring summary coherence.

In our case, since there is clearly a strong connection between feature extraction and summarization, the extracted features together with parts of the sentence or whole sentences can serve as a summary [19], and is what we do in our approach.

An interesting aspect about our approach is its domain independence. The predefined list of features may contain any desired feature, and can be run on almost any type of data set containing a sufficient amount of raw text. One drawback is it is somewhat difficult to take into account typographical errors. The most common mistakes may in theory be added as synonyms to existing features, but considering all alternatives may be problematic and time-consuming. However, assuming no spelling mistakes should still achieve an adequate amount of feature mentions.

## 3.4 Temporal Aspects

Within the domain of temporal opinion mining, opinion lexicon and statistical modeling are the most popular techniques to predict and estimate changes in opinion. These changes can be things such as time or recent events. For instance, as time goes by, consumer interest towards a certain product might naturally diminish. When a competitor releases a clearly superior product, consumer opinion towards older products might plummet. These techniques build on the algorithms presented earlier in this section, but includes a new time element.

One of the major problems in temporal opinion mining is to find meaningful data from documents that arrive continuously over time. Reviews with different opinions about the same object, growing and fading in intensity for a period of time can be an example of such a problem. More specifically, the main problem will be to find the time periods where certain features have risen in intensity. This provides a framework for analyzing why there has been an increase of activity.

We used burst detection to identify changes in grade scores and opinions over time, employing a version with frequencies and thresholds. The first thing our burst detection does is to check if there is a change in the average monthly sentiment score. If that is the case, there could possibly be a big change in sentiment, but this could also just be a one-time event. This has to be factored in, so we also check if the reviews in the next, future month is somewhat stable to eliminate possible fluctuation. Of course, this also eliminates the chance of grades moving up or down over the threshold two month in a row. However, this rarely happens. The last thing we do is to make sure that we have substantial data. In this case enough reviews to minimize the importance of just a few reviews.

# 4. OPINION VISUALIZATION

In order to be able to evaluate our ideas in a controlled fashion, we developed a prototype for visualizing sentiment changes over time. With the data sets we gathered, we had all the data we needed to look at sentiment changes over time.

## 4.1 Opinion Mining Framework

This framework was written in Java and consists of tools for data set crawling, burst detection, POS tagging and different methods for calculating the sentiment of the reviews.

The prototype was created to visualize certain aspects of the data set. The intention was to have a simple portal to evaluate any and all parts of the data set deemed interesting. The resulting prototype is a web site matching several interesting features. Among them are the abilities to view a map with averaged hotel sentiments and quickly identify hotels with a high rating. For each hotel, it is also possible to view a graph visualizing the fluctuations in customer opinion toward the hotel. From the instance the first review is written, until the last, users can easily see if a hotel is improving or falling apart. Additionally, it is possible to filter hotels based on specific features, not only overall sentiment. For instance, rank hotels based on the quality of their service or friendliness of their staff. Finding areas with clusters of highly thought of hotels is also available, by evaluating the area sentiment. With this feature, one can easily find out in which areas one may find the best hotels, and which areas one perhaps should avoid.

The visual representation is set up as a website, created with JavaScript, jQuery[4] and Google Maps[5] javascript API version 3. The site reads in processed review scores in a predefined XML format using a DOM parser and outputs correct markers on the map based on location and score.

## 4.2 Visualization on Map

In an attempt to more easily identify areas with a high degree of good hotels, functionality for displaying the score of a hotel as a coloration of a small radius was added. The radius of the colored circle is dependent on the amount of reviews a hotel has, although with a minimum and maximum radius. Figure 1 illustrates how this looks like.

The basic calculation for determining the radius is performed as follows, where r is the radius:

$$r = \max(20, \min((reviewCount) * 0.25, 300)) \quad (1)$$

The resulting radius lies in the range 20-300 meters. If filtering on amount of reviews is used, that is minimum reviews for a marker to be displayed is greater than 0, this formula has some flaws. If all the hotels displayed have more than 1200 reviews, all circles have a radius of 300, and separating the hotels becomes difficult. To remedy this, a modified version of the formula is used in such cases. The radius is scaled to still lie within the same range and avoid all circles having radiuses close to the maximum limit. The modified version is as follows:

$$r = \max(20, \min((reviewCount - minReviews) * 0.25, 300)) \quad (2)$$

where $r$ is the radius and *reviewCount* is always greater than or equal to *minReviews*. This ensures the whole range of radiuses is still used, and the larger circles have notably more reviews than the smaller ones.

In addition to scaling the radius of each circle based on review counts, the color of each circle is based on the calculated sentiment score. Scores of 0.6 and higher are colored green, while lower are colored red. Also, colors are further divided into varying degrees of opacity. For the green circles, higher scores means a more solid circle, while for the red circles the lower the score the more solid the circle. The colors and opacities make sure that multiple overlapping circles of the same color result in less opacity on the overlapping areas. This gives the illusion that certain areas containing all green or all red circles are very good or very bad areas, respectively (see Figure 2 for an example).

To plot sentiment changes on graphs, we used the jQuery plugin Flot.[6] Figure 3 shows what a graph may look like when showing changes on a month by month basis. The x-axis shows dates and is dynamically determined based on the dates the reviews were written. It starts at the earliest review and ends at the last. The y-axis shows score values between -1 and 1 and is static. The graphs show:

- Sentiment Score: The calculated sentiment scores averaged over all reviews within each month. Always between 0 and 1.

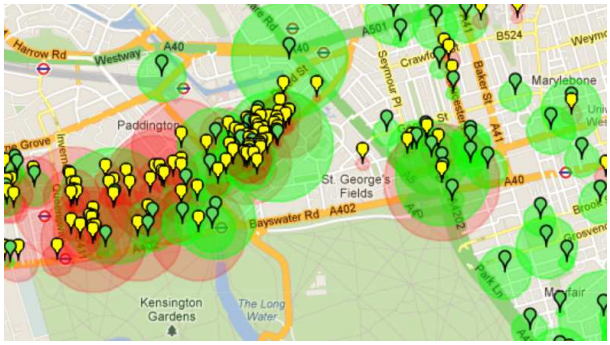


**Figure 2: Example of detection of good and bad areas in a city.**


**Figure 3: Month by month score changes.**

[4]http://www.jquery.com/
[5]http://maps.google.com/
[6]http://code.google.com/p/flot/

- Actual Score: The actual score review authors gave the hotel. Always between 0 and 1.
- Sentiment Score Change: Change to the sentiment score from month to month. Always between -1 and 1.
- Actual Score Change: Change to the actual score from month to month. Always between -1 and 1.

Most of the graphs have large fluctuations early on, but they stabilize more and more for the more recent periods. This is mainly due to the distribution of reviews. Earlier periods have significantly fewer reviews to base their scores on, and as such a single review has a bigger impact when averaging over equally sized periods.

## 4.3 Feature Search

One of the issues we wanted to investigate with the framework was ranking and grading hotels based on certain features. For instance emphasize hotels offering a good breakfast, or hotels with a friendly and helpful staff. This is a helpful way for users to find hotels which have the facilities that each individual user finds important. If one does not need a breakfast, but require good accessibility to nearby public transportation, it should be possible to filter out what previous users think about specific services and filter results accordingly.

These features are defined ahead of time and are generally meant to be domain specific. In our experiments, we defined and identified opinions about the following features: breakfast, location, staff, service, clean, internet.

# 5. EXPERIMENTS AND RESULTS

This section presents results from our experimental evaluation. First we describe the data set created for the evaluations. Then we present an evaluation of the accuracy of SentiWordNet and machine learning. Finally, we evaluate burst detection and temporal change results.

## 5.1 Data Set

Two of the largest travel websites, TripAdvisor and Booking.com, with reviews on hotels, were chosen to as source for the data collection, and a focused crawler was developed in order to retrieve the data stored there. There are some significant differences between these sites, reflecting the fact that Booking.com is a general hotel reservation website where only users which have booked a hotel from that site can give a review. On TripAdvisor on the other hand, anybody can make a review, independently of where they have booked the hotel (with the obvious possibilities for cheating this creates). On the positive side, on TripAdvisor reviewers are registered as users, which makes it possible to also look at previous reviews from a particular user.

An observation is that reviews on TripAdvisor in general contain more text than Booking.com. Compared to other sites, where the review often contains a block of text of varying size, each review on Booking.com has clear and separate areas for pros and cons. Generally, this makes it easier to determine positive and negative sentences, as most text is already divided into positive and negative. Still, by manual examinations, it can be seen that text is not always tagged correctly.

The formula for calculating the sentiment of each review is fairly simple. It takes the calculated positive score for the text and divides it by the sum of the positive score and the negative score. In other words, $\frac{score_p}{score_n + score_p} = \textit{Sentiment}$. The resulting score is a value between 0 and 1, where 1 is perfectly positive, 0 perfectly negative and 0.5 has an equal positive and negative score.

When crawling TripAdvisor and Booking.com, two separate data sets were created. In order to be able to have a larger number of reviews in one collection for the experiments, we have merged these two sets. As can be expected, this has to be done carefully. One reason is that each source has a different amount of hotels and some hotels exist in one source but not the other. However, even when a hotel is in both sets, the matching is not trivial.

The logical first step of comparing hotel names to each other and finding duplicates, but this only resulted in a match of about 20% of the hotels from each source. A fair starting point, but not very accurate. Another approach was to use geo-coordinates to find overlapping points. However, the coordinates found on Booking.com and TripAdvisor were occasionally very far off. They seemed to be set either manually or using an address, which results in coordinates varying quite a lot depending on sources used and size of the hotel. For instance, Bab Al Shams Desert Resort & Spa in Dubai spans a huge area. Using the coordinates from Booking.com (24.8163127229457, 55.2300953865051) and TripAdvisor (24.90699, 55.440754) puts the hotel at two different locations roughly 23.5 kilometers apart. This is a rather extreme example, but it is useful in showing that distances may vary from just a few meters to several kilometers, making the accuracy highly debatable.

As such a combination of hotel name and coordinate matching was the natural solution. Different sources may have slightly different spellings for the same hotel. The Jaro-Winkler distance was chosen because it is specifically designed to be efficient and accurate on shorter strings, which is the case of hotels from the different data sets in an effort to determine whether or not they were the same hotel. The resulting score is a value between 0 and 1, where 0 is no similarity and 1 means identical. The Jaro-Winkler distance for each of the hotels in the original data sets was calculated, and a closest match for each was determined. If the Jaro-Winkler distance of the closest match was greater than 0.9 it was assumed to be a match. If the distance was between 0.8 and 0.9, the Jaro-Winkler distance of a subset of both hotels addresses were compared. Additionally, the distance between each hotel's geo coordinates was calculated using the Haversine formula. If the geo-coordinate distance is below 500 meters, and the Jaro-Winkler distance of the addresses were a match greater than 0.9, they were assumed to be a match. If the closest match did not qualify for these terms, it was assumed no match had been found.

The final result from determining hotel matches using a combination Jaro-Winkler and Haversine was a combined data set containing around 70% of the hotels. Hotels without a match in both original data sets are discarded from the merged data set.

The following results are based on a subset of data, selected for testing based on location. The locations used are Athens, Dubai, London, New York City and Paris. The locations were selected due to being spread out across the world, and having varying amounts of reviews. London, for instance, has a high amount of reviews from both Booking.com and TripAdvisor. New York has many reviews from TripAdvisor, but not so many from Booking.com. Dubai and Paris all have an average amount of reviews, while Athens is on the low side.

### *5.1.1 Statistics*

Following are statistics detailing certain aspects of the data sets. Figure 4 shows the distribution of reviews according to their allotted scores. It seems that Booking.com does not have any reviews with scores below 2.5, and that for both sources scores are somewhat inflated toward the higher end of the scale.

Figure 5 is a simple graph showing how many hotels have specific amounts of reviews. As seen, most hotels have less than 50,

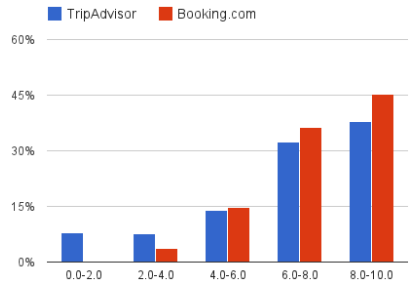


**Figure 4: Distribution of review scores.**

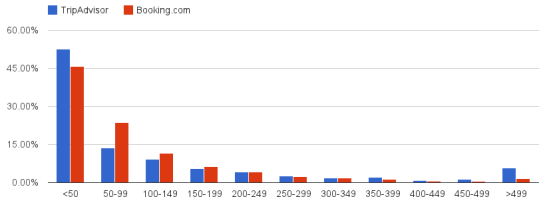


**Figure 5: Percentage of hotels for each amount of reviews interval.**

with declining numbers all the up to 500. The reason for the upward trend at above 500 is due to that range being a much larger interval than the lower ones. Figure 6 shows the average scores of hotels and groups them within specific intervals. It is related to Figure 4, in that average hotel scores are based on the review scores. Therefore the graphs are somewhat similar, with scores leaning toward the higher end of the scale, Booking.com even more than TripAdvisor.

Overall the data from these two sources are quite similar. They both consist of user-generated content with a short amount of text and an arbitrarily determined overall score. TripAdvisor reviews contain slightly more full-sentence text, while Booking.com consists mostly of short summaries and keywords. TripAdvisor contains more reviews for most locations. Scores from both sources are shifted toward the higher end of the scale, but Booking.com scores are noticeably higher, with more than 80% of review scores being higher than 6.0 on a scale from 0-10.

It should also be noted that from a control set of 662,991 reviews from Booking.com, the lowest score from this set was 2.5. No reviews at all were rated less than 2.5. Part of this likely due to the way scores are set on Booking.com. Scores are an average of a range of subscores. Each reviewer must rate each of six different criteria individually, including comfort, location and value for money. This would likely increase the average scores somewhat, due to most people finding something they like about a part of their stay, and the extremely negative scores therefore are not so common. However, one would still assume that some very angry customers would rate all the subcriteria with the bottom score, at least when checking hundreds of thousands of reviews. It is therefore possible that Booking.com employ some sort of filtering mechanism, and ratings with all negative scores are regarded as spam.

Table 1 shows the distribution of reviews on a year by year basis from 2004 up until march 2012. From 2004 up until roughly 2008, there are quite few reviews. From 2009 up until march 2012 the number of reviews increases drastically. Note that from Booking.com there are no reviews before 2010. The merged data set contains roughly the average of these two for each year, although

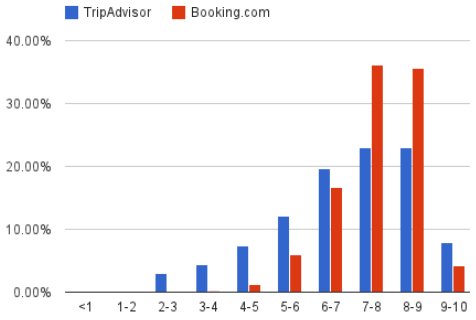


**Figure 6: Score distribution for TripAdvisor and Booking.com hotels.**

**Table 1: Percentage of reviews year by year.**

| | TripAdvisor | | Booking.com | |
|---|---|---|---|---|
| **Year** | London | New York | London | New York |
| Earlier | 0.94% | 1.47% | 0.0% | 0.0% |
| 2004 | 2.91% | 3.90% | 0.0% | 0.0% |
| 2005 | 3.91% | 5.83% | 0.0% | 0.0% |
| 2006 | 5.88% | 6.39% | 0.0% | 0.0% |
| 2007 | 7.71% | 8.45% | 0.0% | 0.0% |
| 2008 | 8.52% | 9.29% | 0.0% | 0.0% |
| 2009 | 13.52% | 11.99% | 0.0% | 0.0% |
| 2010 | 18.39% | 17.48% | 1.16% | 0.63% |
| 2011 | 32.24% | 28.85% | 85.79% | 86.16% |
| 2012 (≈march) | 5.97% | 6.34% | 13.05% | 13.2% |

there is a slight skew in advantage for TripAdvisor, which has a somewhat higher average number of reviews per hotel.

## 5.2 Opinion Mining Accuracy

This section details the accuracy of SentiWordNet when tested on our data set. The reason for this testing is to determine if SentiWordNet works well enough for our purpose.

### *5.2.1 SentiWordNet Accuracy*

Previous studies which have incorporated SentiWordNet have generally achieved accuracies of around 65-70%. In a 2009 study by Ohana and Tierney, they used SentiWordNet to classify a data set of movie reviews [15]. The data set consisted of 1000 reviews. They concluded that SentiWordNet was quite accurate, with an average accuracy of around 65%. However, their data set was arguably not very large.

**Standard SentiWordNet.** Table 2 shows the resulting accuracy from running SentiWordNet on our range of data sets. Generally, reviews from TripAdvisor achieved an accuracy of around 80%, which is a very good result. It performed slightly worse with Booking.com, being correct slightly below 70% of the time on average. This is more in line with previous studies, and was closer to the expected result. Although compared to the results from TripAdvisor, it is slightly lower than desirable. However, it was not entirely unexpected that TripAdvisor reviews were easier to determine than Booking.com reviews. Reviews from TripAdvisor generally consist of full sentence texts with multiple paragraphs, and therefore contain a lot of text for SentiWordNet to work with. Booking.com reviews are more often than not key word based in comparison. Along with the innate split between pros and cons, this often results in short summaries of the things that went well, but longer, more detailed texts with the negative things, or vice versa. Even

Table 2: SentiWordNet accuracy results.

| Source | Location | #correct | #reviews | %correct |
|---|---|---|---|---|
| TripAdvisor | London | 133,492 | 167,655 | 79.62 |
| | New York | 130,049 | 158,950 | 81.82 |
| | Athens | 14,479 | 17,579 | 82.37 |
| | Paris | 99,776 | 122,883 | 81.20 |
| | Dubai | 29,327 | 34,016 | 86.22 |
| Booking.com | London | 100,082 | 147,076 | 68.05 |
| | New York | 27,059 | 39,485 | 68.53 |
| | Athens | 8,476 | 11,327 | 74.83 |
| | Paris | 40,812 | 58,363 | 69.93 |
| | Dubai | 26,683 | 37,628 | 68.26 |

Table 3: SentiWordNet accuracy results with a simplified Lesk Algorithm.

| Source | Location | #correct | #reviews | %correct |
|---|---|---|---|---|
| TripAdvisor | London | 129,822 | 167,655 | 77.43 |
| | New York | 125,889 | 158,950 | 79.20 |
| | Athens | 14,145 | 17,579 | 80.47 |
| | Paris | 98,026 | 122,883 | 79.77 |
| | Dubai | 28,666 | 34,016 | 84.27 |
| Booking.com | London | 99,294 | 147,076 | 67.51 |
| | New York | 26,712 | 39,485 | 67.65 |
| | Athens | 8,285 | 11,327 | 73.14 |
| | Paris | 40,343 | 58,363 | 69.12 |
| | Dubai | 25,095 | 37,628 | 66.69 |

comments such as "Nothing in particular" may exist for one of the sentiments. This can skew the ratio of positive to negative text, and may for instance make the text as a whole seem negative, even though the opposite might be true.

**Simplified Lesk.** The standard SentiWordNet implementation simply selects the most commonly used definition of a word. Most of the time this is fairly accurate. However, some words have several very different definitions. In an attempt to achieve a higher degree of accuracy, a simplified Lesk algorithm was implemented [9]. The Lesk algorithm attempts to find the definition which is most likely correct for the given context. It does this by matching other words in the original setting with each word in the descriptive definition. The definition with the highest amount of overlapping words is assumed to be the most likely choice. Table 3 lists the results of SentiWordNet using a simplified Lesk algorithm. Figures 7 and 8 compare the accuracy from the Lesk algorithm versus the standard SentiWordNet algorithm.

As one can see from Figure 7 and 8, the Lesk algorithm performs worse than one that simply selects the most commonly used

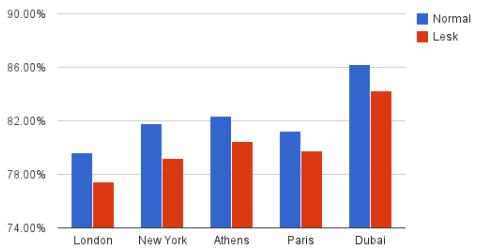


**Figure 7: Accuracy of the Lesk algorithm vs. Standard SentiWordNet for TripAdvisor.**

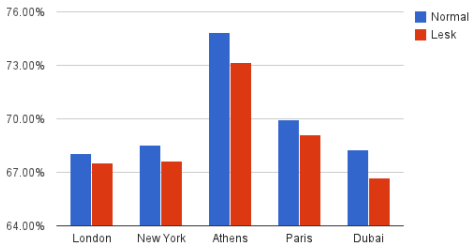


**Figure 8: Accuracy of the Lesk algorithm vs. Standard SentiWordNet for Booking.com.**

Table 4: SentiWordNet accuracy results with five categories.

| Source | Location | #correct | #reviews | %correct |
|---|---|---|---|---|
| TripAdvisor | London | 103,326 | 167,655 | 61.63 |
| | New York | 104,640 | 158,950 | 65.83 |
| | Athens | 11,544 | 17,579 | 65.67 |
| | Paris | 78,750 | 122,883 | 64.09 |
| | Dubai | 24,724 | 34,016 | 72.68 |
| Booking.com | London | 63,396 | 147,076 | 43.10 |
| | New York | 18,185 | 39,485 | 46.06 |
| | Athens | 6,503 | 11,327 | 53.44 |
| | Paris | 26,850 | 58,363 | 46.01 |
| | Dubai | 16,783 | 37,628 | 44.60 |

definition of a word. This was a bit surprising, but could possibly be explained by the simplicity of the Lesk algorithm. There are many different fine-tuning tools to adapt an improved Lesk algorithm for better results. We did, however, decide not to focus on this, since selection of the most commonly used definition of a word performed very well.

**Five Categories.** We also tested the SentiWordNet results on five different categories: strong positive, weak positive, neutral, weak negative, strong negative. The results can be seen in Table 4. More categories generally means increased difficulty in achieving higher amounts of accuracy, especially when categorizing based on numeric values. This is due to the way categorization occurs. Different methods and techniques have different ways of estimating category values. Placing them within the same range of values provides a way to compare them, but with smaller ranges for each category (due to more categories), getting matches naturally becomes harder [17].

### 5.2.2 *Machine Learning Accuracy*

We also tried machine learning techniques to see how they performed versus SentiWordNet. We used 20 000 reviews as training data and tested with 10 000 reviews. The results can be seen in Table 5. Despite good results, we do not get the same polarity degree as in SentiWordNet. SentiWordNet is unique since it has an unlimited amount of polarity degree because of its score-based categorization. However, we did test machine learning with five categories, to get some degree of polarity. The results from five categories are seen in Table 6.

Machine learning does perform very well on few categories, usually better than SentiWordNet. Nevertheless, we decided to go forward with SentiWordNet because of its unlimited polarity degrees and the polarity degree results being better than machine learning.

## 5.3 Burst Detection

The main point of burst detection is to try to detect abnormal

**Table 5: Result table for machine learning algorithms.**

| Source | Algorithm | #correct | % |
|---|---|---|---|
| TripAdvisor | Naive Bayes | 8,924 | 89.24 |
| | Dyn. LM Classifier | 9,003 | 90.03 |
| Booking.com | Naive Bayes | 6,411 | 64.11 |
| | Dyn. LM CLassifier | 6,592 | 65.92 |

**Table 6: Result table for machine learning algorithms with five categories.**

| Source | Algorithm | #correct | % |
|---|---|---|---|
| TripAdvisor | Naive Bayes | 5,742 | 57,42 |
| | Dyn. LM Classifier | 5,712 | 57,12 |
| Booking.com | Naive Bayes | 4,408 | 44.08 |
| | Dyn. LM Classifier | 4,618 | 46.18 |

changes in hotel reviews, and then try to analyze why those changes have occurred. We define a burst as a sudden change in review score exceeding a given threshold. The results can be seen in Table 7. The algorithm has a 51.75% correctness with a burst defined threshold of 1.0. However, if one increases the threshold, it has higher degree of correctness. This is because we set the definition of what a burst is by a threshold, and then compare our sentiment grade score with the actual one. So, if there is greater changes in our sentiment score, there is also a higher possibility that that is the case in the actual score. With a 2.0 threshold it reaches a correctness of 64.17%, but finds substantial fewer bursts.

When the bursts were found, we had to find out why those changes had occurred. This was done by implementing a simple feature extraction where we grouped two and two words together, namely adverbs and verbs together in pairs and adjectives and nouns in pairs. Some examples can be seen in Table 8 and 9. Table 8 contains some features from user reviews about the hotel Grand Midwest Express Hotel Apartments in Dubai. The period the reviews are taken from is July of 2011, and the average score in that month decreased by around 1.15 points from June 2011. Some of the features from June is shown in Table 9. As can be seen in the tables there is a more mixed response in the month of June. The actual score for that month was 6.70 based on 9 reviews. In July 2011 the average score was 5.55 based on 13 reviews. The reason for the decline in its average monthly grading score seems to be that some of the customers have experienced insects or rats in the hotel.

The goal of the burst detection algorithm was to detect sudden and lasting changes in sentiment. Several bursts were found in the data sets. However, the value of these results was rather limited. Most of the bursts seemed to stem from normal variations in the SentiWordNet calculations. The main reason for this is likely due the limited amount of reviews. Although some hotels had thousands, roughly 95% had less than 500, and 45% had less than 50 (Figure 5). Spread out over 2-8 years, this does not result in a high amount of data points for a proper temporal evaluation. Finding useful bursts containing enough mentions on a specific topic, with enough reviews before, during and after the burst, was therefore quite difficult. The bursts we did find did not provide much useful data about why the sentiment changed either. The accompanying review texts mostly contained generic terms and consequently the same terms and features as before the burst took place. However, some interesting points were still identified. A small subset of the detected bursts had some features which indicated a cause behind the sudden change in sentiment. With a larger data set and further tweaking of the algorithm, positive results are not unlikely.

**Table 7: Burst detection results with different thresholds.**

| Threshold | Location | # correct | # bursts | % correct |
|---|---|---|---|---|
| 1.0 | London | 380 | 712 | 53.37 |
| 1.0 | New York | 239 | 389 | 61.44 |
| 1.0 | Athens | 39 | 90 | 43.33 |
| 1.0 | Paris | 232 | 478 | 49.79 |
| 1.0 | Dubai | 125 | 246 | 50.81 |
| 1.0 | Average | | | 51.75 |
| 1.5 | London | 74 | 137 | 54.01 |
| 1.5 | New York | 44 | 66 | 66.67 |
| 1.5 | Athens | 12 | 23 | 52.17 |
| 1.5 | Paris | 68 | 124 | 54.84 |
| 1.5 | Dubai | 34 | 61 | 55.74 |
| 1.5 | Average | | | 56.69 |
| 2.0 | London | 18 | 28 | 64.29 |
| 2.0 | New York | 12 | 19 | 63.16 |
| 2.0 | Athens | 4 | 7 | 57.14 |
| 2.0 | Paris | 18 | 32 | 56.25 |
| 2.0 | Dubai | 8 | 10 | 80.00 |
| 2.0 | Average | | | 64.17 |

## 5.4 Feature Selection

The initial idea for the feature extraction was to identify sentences containing each feature and determine scores based on these. Mostly this approach worked fine. However, some sentences contain multiple clauses, and not all parts are necessarily relevant. Sometimes they branch of into completely different topics. This results in quite a bit of noise when it comes to calculating the sentiment scores. Consider for instance the following sentence from a Booking.com/TripAdvisor review, and the selected feature "staff":

> "[..](1) location is good, (2) staff is very helpful and friendly, (3) rooms are modern and functional, (4) roof bar is nice to have some drinks and breakfast with super views of acropolis... (5) a perfect stay[..]"

This sentence contains multiple clauses containing sentiments about several features (numbered 1-5 to simplify referring). In this example, only the second part, "staff is very helpful and friendly", relates to the staff sentiment. The rest of the sentence is about location, decor and available facilities. The sentiment scores for these additional clauses are not relevant with regards to the intended score. As such, one can not be certain of the accuracy of the rankings.

An alternative to full sentence analysis was to split sentences further by seeing each clause as a separate entity. This increased the likelihood that the part containing the feature was indeed about the desired feature. Most of the noise found by the previous approach disappeared. In the mentioned example, only the part "staff is very helpful and friendly" would be used in staff sentiment calculation. The drawback of this new technique was of course that the calculations lost some of clauses which were indeed relevant to the feature. Looking at the same sentence, but from a "location" point of view, one would find the first clause "location is good". Initially this seems fine. However, the last part of clause 4, "[roof bar]with super views of acropolis", can arguably also be considered quite relevant regarding the location. With this new approach, that information would be lost.

In our evaluation, we found that splitting on clauses performs better. The difference is not huge though, and the correctness for both is very high, between 80%-89%. However, these results are

**Table 8: Features from Grand Midwest Express Hotel Apartments in Dubai from a positive burst month.**

| Feature | # mentioned |
|---|---|
| dangerous toilet | 2 |
| poor insect | 1 |
| mouldy bed | 2 |
| infested hotel | 2 |
| unprofessional staff | 1 |

**Table 9: Features from Grand Midwest Express Hotel Apartments in Dubai month before burst.**

| Feature | # mentioned |
|---|---|
| shabby bed | 1 |
| clean rats | 1 |
| friendly staff | 2 |
| limited parking | 1 |
| good room | 2 |

based on a very small data sample, and it will have a high potential error rate and therefore should be interpreted with care. The reason for the small amount is simply because we had to do it manually to be able to judge the scores correctly, and this does take a substantial amount of time.

# 6. CONCLUSIONS AND FURTHER WORK

In this paper we have studied opinion mining applied on data from travel review sites and how the results of sentiment analysis of textual reviews can be visualized using Google Maps. An evaluation of the techniques presented in the paper showed high accuracy in opinion mining, and that the prototype can help detect hotel features and possible reasons for changes in opinion as well as show "good" and "bad" geographical areas based on hotel reviews.

The techniques and prototype we have created is designed for evaluating customer sentiment regarding hotels. The underlying techniques used are fairly general, and could be applied to any relevant and sufficiently large data set containing sentiment data. To fully utilize the prototype, the only requirement is that the data set contains both a geographical and a temporal aspect. However, all techniques and features can also be used separately. Feature search and extraction may be used on any opinionated data set, although presentation of results in the prototype would need altering to some sort of ranked list, for instance. Burst detection may be used on any temporal data set, and might even function better on other types of data sets, for instance movie or video game reviews.

# 7. REFERENCES


[1] P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan. An Exploration of Sentiment Summarization. In *Proceedings of AAAI'2003*, 2003.

[2] L.-C. Cheng, Z.-H. Ke, and B.-M. Shiue. Detecting changes of opinion from customer reviews. In *Proceedings of FSKD'2011*, volume 3, 2011.

[3] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, 1990.

[4] D. Das, A. K. Kolya, A. Ekbal, and S. Bandyopadhyay. Temporal analysis of sentiment events: a visual realization and tracking. In *Proceedings of CICLing'2011*, 2011.

[5] A. Esuli and F. Sebastiani. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC'06*, 2006.

[6] T. Fukuhara, H. Nakagawa, and T. Nishida. Understanding sentiment of people from news articles: Temporal sentiment analysis of social events. In *Proceedings of ICWSM'2007*, 2007.

[7] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of AAAI'04*, 2004.

[8] L. Ku, Y. Liang, and H. Chen. Opinion extraction, summarization and tracking in news and blog corpora. In *Proceedings of AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs*, 2006.

[9] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC'86*, 1986.

[10] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., 2006.

[11] A. McCallum. Text classification by bootstrapping with keywords, EM and shrinkage. In *Proceedings of ACL99 - Workshop for Unsupervised Learning in Natural Language Processing*, 1999.

[12] G. Mishne and M. de Rijke. MoodViews: tools for blog mood analysis. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, 2006.

[13] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima. Mining product reputations on the web. In *Proceedings of SIGKDD'2002*, 2002.

[14] K. Nørvåg and O. K. Fivelstad. Semantic-based temporal text-rule mining. In *Proceedings of CICLing'2009*, 2009.

[15] B. Ohana and B. Tierney. Sentiment classification of reviews using SentiWordNet. In *Proceedings of the 9th. IT & T Conference*, 2009.

[16] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL'2004*, 2004.

[17] B. Pang and L. Lee. Opinion Mining and Sentiment Analysis. *Foundation and Trends in Information Retrieval 2*, pages 1–135, 2008.

[18] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of HLT'05*, 2005.

[19] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. *Comput. Linguist.*, 28(4):399–408, 2002.

[20] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL'2002*, 2002.

[21] Y. Wu, F. Wei, S. Liu, N. Au, W. Cui, H. Zhou, and H. Qu. OpinionSeer: interactive visualization of hotel customer feedback. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1109–1118, 2010.

[22] H. Yang, L. Si, and J. Callan. Knowledge transfer and opinion detection in the TREC2006 blog track. In *Proceedings of TREC*, 2006.

[23] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of ICDM'2003*, 2003.