Chapter 11
Automating opinion analysis in film reviews: the case of statistic versus linguistic approach

Damien Poirier, Cécile Bothorel, Émilie Guimier De Neef, and Marc Boullé

Abstract Websites dedicated to collecting and disseminating opinions about goods, services, and ideas,attract a diversity of opinions comprising attitudes and emotions. *www.flixster.com* is an example of a participative web site, where enthusiastic reviewers share their feelings/views on movies - usually expressing polar opinions. The participative web-sites usually contain substantial amount of data which is continually been updated.The contents of such websites is regarded as a key source of information by academic and commercial researchers keen to gauge this sample of public opinion. The key challenge is to automatically extract the reviewers opinion. Our goal is to use the reviews for building a model which can then be used to predict the user's verdict on a movie. We explore two different methods for extracting opinion. The first, machine learning method that uses a naive Bayesian classifier. The second method builds upon existing NLP techniques to process opinions and build dictionaries: those dictionaries are then used to determine the polarity of a comment comprising a review. We compare and contrast the relative merits of the two methods with special reference to movie review data bases.

Damien Poirier
France Telecom RD, TECH / EASY, 2 avenue Pierre Marzin, 22300 Lannion, FRANCE. Now at Le Laboratoire d'Informatique Fondamentale d'Orléans, Université d'Orléans, Rue Léonard de Vinci, B.P. 6759, F-45067 ORLEANS Cedex 2. e-mail: damien.poirier@gmail.com

Cécile Bothorel
France Telecom RD, TECH / EASY, 2 avenue Pierre Marzin, 22300 Lannion, FRANCE, e-mail: cecile.bothorel@orange-ftgroup.com

Émilie Guimier De Neef
France Telecom RD, TECH / EASY, 2 avenue Pierre Marzin, 22300 Lannion, FRANCE, e-mail: emilie.guimier@orange-ftgroup.com

Marc Boullé
France Telecom RD, TECH / EASY, 2 avenue Pierre Marzin, 22300 Lannion, FRANCE, e-mail: marc.boulle@orange-ftgroup.com

11.1 Introduction

With the spread of high speed access to the Internet and new technologies, there has been tremendous growth in online music and video markets. As more players appear on this field, competition increases and content provider can no longer wait for the customer. Instead, they try to trigger purchases by pushing contents: suggesting different choices of movies or songs has become the big thing when it comes to selling content on-line. Actually recommendation is not a new concept, it has already been used on commercial sites (*Amazon*, *Fnac*, *Virgin* ...) as well as on musical platforms (*Lastfm*, *Radioblog*, *Pandora* ...). But looking at the recommendation techniques used on such web sites suggests that there is still room for innovation.

[5] presents an overview of recommendation techniques. These techniques are either based on Internet users' notations or content descriptions (*user*- and *item-based* techniques using collaborative filtering), or based on matching Internet user profiles and content descriptions (content filtering), or based on hybrid techniques combining both approaches. Although these techniques are different, they have the same limitation: the hollow nature of a matrix describing users and content profiles. Indeed, the sites proposing recommendations to their customers often have a large catalogue while users only give their opinion on a small number of products. This phenomenon makes the comparisons between profiles risky. In the recommendation field, the difficulty in collecting descriptions about users taste (ratings, interests ...) and content (meta data) is a recurrent problem.

In order to address these problems, a new research area has opened up: mining the resources of the *open* Internet to boost *closed* sites performance. Instead of focusing solely on the data that can be retrieved from a single web site, recommendation techniques may include the vast amount of data that is now available from the Internet. In the era of Web 2.0 and community sites, it is now common for users to share pictures, tags, news, opinions ... Such data could be gathered to support automatic information extraction.

Motivated by this potential shift in providing recommendation, we have focused on methods for extracting opinions from movie reviews published on community sites[1]. Our main objective is to establish a user profile based on what he or she declares to like or dislike in movies through his or her published writings (blogs, forums, personal page on the flixster website, etc...).

We focus on two different opinion extracting methods. The first machine-learning method was developed to classify textual reviews into either a positive or negative class. The second NLP-based is used to build an opinion dictionary and to detect words carrying opinion in the corpus and then to predict an opinion.

[1] This work enters in the frame of European project IST Pharos (PHAROS is an Integrated Project co-financed by the European Union under the Information Society Technologies Programme (6th Framework Programme), Strategic Objective "Search Engines for Audiovisual Content" (2.6.3))

We did apply those two approaches to data from the movie-review web-site, www.flixter.com. We discuss the results to compare the two approaches and we provide insights as to which approach should be used for a given corpus of opinions.

11.2 Related work

Opinion extraction in (trademark) product reviews is important. For instance, [7] present a method for automatically classifying reviews according to the polarity of the expressed opinions, i.e. the tool labels reviews positively or negatively. They index opinion words and establish a scale of rates according to intensity of words. They determine words intensity by using machine learning techniques. Finally, to classify a new review, they build an index reflecting the polarity of each sentence by counting identified words.

[25] explain how they verify reputation of targeted products by analyzing customers' opinions. They start by finding Web pages *talking* about a product, for example a television, then they look for sentences which express opinions in these websites, and finally they determine if the opinions are negative or positive. They determine this by locating in reviews opinion words which were indexed previously in an *opinion dictionary*. Related work of import here includes [31] who have classified reviews in two categories: recommended and not recommended;[35] which categorizes sentences according to polarity and strength of opinion; and, [26] which seeks opinions on precise subjects in documents.

11.2.1 Machine learning for opinion analysis

Systems using learning machine techniques generally classify textual comments into two classes, positive and negative; extensions of these methods incorporate a third class, neutral, and sometimes the classification comprises five classes, very positive, positive, neutral, negative and very negative. These supervised classification methods assume a comment describes only one product and try to predict the rate given by the author.

Many methods use NLP techniques to annotate the corpus. [34] describe a scheme for annotating expressions of opinions, beliefs, emotions, sentiment and speculation [...] in the news and other discourse; [35] test three different learning methods, frequently used by linguists: *BoosTexter* [29], *Ripper* [6] and *SVM*^{light}, an implementation of Vapnik's [33] Support Vector Machine by [16]. The use of *SVM*^{light} gives the best results on [35] annotated corpus. [27] use a naive Bayes classifier and a classifier maximizing the entropy. Similarly, in order to characterize what is appreciated or not appreciated in a sentence, [**?**] combine a *parsing* technique with a Bayes classifier to associate polarity with sets of themes.

Furthermore, [27] and [7] show that corpus preparation with a lemmatizer or a negation detection for example, does not lead to better annotation. In order to predict reviewers' opinion, these two papers explore learning methods and show that these methods are more powerful than parsing methods followed by a calculation as shown in the next section. If reviewers' comments are treated as bags of words and a relevant learning technique is used, this leads to 83% correct predictions. We will show that our own experiments confirm these conclusions.

11.2.2 Linguistic methods of opinion analysis

[23] describe their *Opinion Observer* system which compares competitive products by using product reviews left by the Internet users. The system finds features such as pictures, battery, zoom size, etc. in order to explain the sentiment about digital cameras. *Opinion Observer* is a supervised pattern discovery method for automatically identifying product features described in the reviews. The system uses a five step algorithm to analyse reveiwers' comments:

STEP 1. PERFORM Part-Of-Speech (POS) tagging and remove digits: For example, the comment 'Battery usage; included software could be improved; included 16MB is stingy' will be transformed as *<N> Battery <N> usage <V> included <N> MB <V>is <Adj> stingy*

STEP 2. REPLACE actual feature words in a sentence with *[feature]*: *[feature] <N> usage <V> included <N> [feature] <V> is <Adj> stingy*

STEP 3. USE n-gram to produce shorter phrase embedded in a long clause: *<V> included <N> [feature] <V> is <Adj> stingy*, will be parsed into two smaller segments: *"<Adj> included <N> [feature] <V> is* and *<N> [feature] <V> is <Adj> stingy*. (Only 3-grams are used in *Opinion Observer*)

STEP 4. DISTINGUISH duplicate tags.

STEP 5. PERFORM word stemming:

This five step algorithm generates 3-gram segments which are saved in a transaction file. *Opinion Observer* then uses association rule learning algorithm to extract rules like

(a) <N1>, <N2> –> [feature]
(b) <V>, easy, to –> [feature]
(c) <N1> –> [feature], <N2>
(d) <N1>, [feature]–> <N2>

Not all the generated may help in extracting product features, so *Opinion Observer* selects only the 'relevant' rules:

(i) Rules that have [feature] on the right-hand-side

(ii) Rules that have the appropriate sequence of items in the conditional part of each rule.
(iii) Rule Transformation: Rule, as such, still cannot be used to extract features and have to be transformed into patterns to be used to match test reviews.

Moreover, there are other steps that help to group synonyms and to deal with the weighting that has to be attached to opinion-bearing terms. The system then decides the orientation of the extracted feature according to the words extracted near the features. Then the system classifies sentences as negative or positive by determining the dominant orientation of the opinion words of the sentence. The result of the comparison between two products is given in the form of diagram with features on X-coordinate and opinions polarity on Y-coordinate.

Opinion Observer is an example of a system based on the analysis of sentences that facilitates in computing the frequency of sentiment-bearing text excerpts (words, expressions, and patterns). Like many similar systems [25, 31, 35, 26], the *Opinion Observer* needs an *Opinion Dictionary* with as many words or expressions as possible that are used for expressing opinions. To build such a dictionary, different techniques can be used but in almost all the cases there is one proviso: the hand-crafting of a set of words and expressions that are used in expressing an opinion, especially polar or neutral opinion. This set is usually referred to as a *seed* and the aim is to find other words and expressions yielding opinions and classify them according to their semantic orientation (positive, negative, but seldom neutral).

Such an opinion-annotated lexicon can be built by using machine learning techniques. For example, [12] and [32] use an unsupervised learning algorithm to associate new words with the seed words. [28] and [21] describe methods of discovering synonyms by analyzing words collocation.

Linguistic methods use syntactic and grammatical analyses in order to extend the lexicon. [12] use conjunctions between a word for which semantic orientation is known and an unclassified word. For example, if there is the conjunction *and* between two adjectives, we can consider that the two have the same polarity if any. Contrariwise, if there is the conjunction *but* between two adjectives, then there is a good chance that the two adjectives have a different semantic orientation.

[31] uses more complex patterns and count the frequency of the words, or expressions beside a word, or expression already classified, and define the semantic orientation of those new words or expressions according to the orientation of the neighbours. Each time an adverb or an adjective is encountered, a pair of consecutive words is extracted:

- Adjective with noun
- Adverb with adjective when they are not followed by a noun
- Adjective with adjective when they are not followed by a noun
- Noun with adjective when they are not followed by a noun
- Adverb with verb

The second extracted word allows the system to confirm polarity of the adjective or adverb by giving an outline of the sentence's context.

The above mentioned method, of counting co-occurrences with words semantically oriented and manually selected, is also used in [36] to determine words are semantically oriented, in terms of the direction and the strength of the orientation. To measure more precisely the strength of opinion expressed in a sentence, adverbs which are associated to adjectives are extracted. Indeed, [1] propose a classification of adverbs into five categories: adverbs of affirmation, adverbs of doubt, adverbs of weak intensity, adverbs of strong intensity and adverbs which have a role of minimizer. The strengths of adverb-adjective combinations are computed according to the weights assigned to these five different adverb categories.

Google's work [10] attempts to find semantic orientation of new words from *WordNet* databases [24]. In similar vein, [14] use sets of synonyms and antonyms present in WordNet to predict semantic orientation of adjectives. In WordNet, words are organised in tree (see Figure 11.1). To determine polarity of a word, the system traverses the trees of synonyms and antonyms of this word and if it finds a seed word in the synonyms, it allocates the same class, but if it finds a seed word in the antonyms, it allocates the opposite class. If it doesn't find a seed word, it rebuilds the analysis with synonyms and antonyms, and loops until a seed is found.

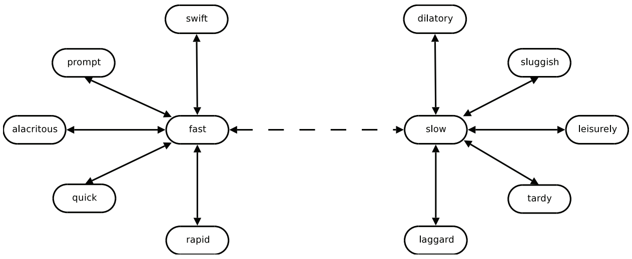

Fig. 11.1 Tree of synonyms and antonyms in WordNet (full arrow = synonyms, dotted arrow = antonyms)

In our opinion, method outlined in [10] is not well-grounded as words can have different meaning according to the context and thus can have synonyms not signaling the same thing. For example, the word *like* has for synonym *love*, but in the sentence *It is like that*, one can use the synonym *love* instead of *like* for instance.

To associate a polarity, negative or positive, to a sentence, we can count the number of terms with positive semantic orientation and the number of terms with negative orientation. If there are more positive terms, the sentence is declared positive, if there are more negative terms, the sentence is declared negative, and if there are as many positive as negative terms, either sentence is declared neutral [36]; with another strategy, the last term carrying opinion determines the sentence polarity [14]. Otherwise, we can extract opinion one by one associated with the feature it refers to

[35, 14].

11.3 Linguistic and Machine Learning methods: A comparative study

In this section we compare two opinion analysis methods using reviews rated by authors and their results. These reviews are available as a corpus of texts. The initial corpus comprises 60,000 films divided equally in positive and negative reviews. We have use 50,000 reviews for training and 10,000 for testing.

The main difficulty with this corpus is the small size of reviews (12 words on average). This makes opinion extraction difficult, even for humans. Moreover, the corpus comprises textual messages very similar to forum messages and include punctuation marks ”!!!”, emoticons ”$:-)$”, expressions from SMS texts ”*ur*”,”*gr*8” and word equivalent for emphasis (*veryyyyy cooooool* instead of *very, very cool*).

Each review in our corpus has a rating given by the author, on a scale of 0 (zero) to 5, and our aim is to predict this rating from our automatic analysis. We have decided to classify reviews in two classes. Reviews with a rating lower than three are considered a negative review and otherwise positive. Here follow examples of reviews with their rating (Table 11.1).

Rate	Review
POS	Great movie!
NEG	this wasn’t really scary at all i liked it but just wasn’t scary...
POS	I loved it it was awesome!
NEG	I didn’t like how they cursed in it......and this is suppose to be for little kids....
NEG	Sad ending really gay
POS	sooo awesome!! (he’s soo hot)
POS	This is my future husband lol (orlando bloom)
NEG	Will Smith punches an alien in the face, wtf!!??
NEG	i think this is one of those movies you either love or hate, i hated it! :o)

Table 11.1 Polarity (NEG/POS) and Exemplar Reviews

11.3.1 Linguistic Approach

11.3.1.1 Method

First we constructed a dictionary of opinion words from a reviewer's corpus.

Next, we separated all reviews according to their rating. For each review category (e.g., set of reviews rated 1 star, set of reviews rated 2 stars . . .), we applied a shallow parser [8] to lemmatize and tag the text. We filtered the words according to their Part of Speech tag and frequency. Verbs and adjectives have then been manually classified according to the opinion they convey.

This list has been expanded using a synonym dictionary (`www.wordreference.com`). Only verbs and adjectives that are not ambiguous have been classified. For example, the word *terrible* is not classified because it can expresses both opinion polarities.

A total of 183 opinion words have been classified in two classes, positive words (115) and negative words (68), in this manner. An excerpt from the dictionary is show in Table 11.2. This dictionary was not made using the corpus used to evaluate this method.

Positive words	good, great, funny, awesome, cool, brilliant, hilarious, favourite, well, hot, excellent, beautiful, fantastic, cute, sweet ...
Negative words	bad, stupid, fake, wrong, poor, ugly, silly, suck, atrocious, abominable, awful, lamentable, crappy, incompetent ...

Table 11.2 Part of hand-crafted lexicon

The last step of the analysis consists of counting opinion words in each review to determine the polarity. For that we first lemmatized all reviews (the same pretreatment as the lexicon) and only adjectives and verbs were kept. Then, a polarity was assigned to reviews according to the majority number of positive words or negative words.

No sophisticated NLP techniques were performed, such as a grammatical structural analysis. But keeping only verbs and adjectives avoids misinterpretations of words such as "*like*" which can have different roles in a sentence.

11.3.1.2 Results

This method enabled us to rate 74% of film reviews on the 20,000 present in the test corpus. All the following results are calculated according to the rated reviews. To compare results with other techniques, we calculate three values: precision, recall and F_{score}.

- $precision = \frac{number\ of\ positive\ examples\ cover}{number\ of\ examples\ cover}$
- $recall = \frac{number\ of\ positive\ examples\ cover}{number\ of\ positive\ examples}$
- $F_{score} = \frac{2 * precision * recall}{precision + recall}$

The confusion matrix of results is presented in Table 11.3.

	Pos. reviews	Neg. reviews
Predicted pos. reviews	8,089	3,682
Predicted neg. reviews	218	2,823

Table 11.3 Confusion matrix obtained with the hand-crafted lexicon

With our technique we obtained 0.81 for precision, 0.70 for recall and 0.75 for F_{score}.

Our principal problems was in determining the polarity of negative reviews. Indeed, the recall of negative reviews is 0.43, whereas it is 0.97 for positive reviews. Contrarily, precision of positive reviews (0.69) is worse than precision of negative reviews (0.93).

This problem can be related to our dictionary: the positive category contains almost twice as many words than negative category. However, the problem is not the detection of negative reviews but their interpretation. These results lead us to think that sometimes people use negation of a positive expression to express their negative feelings without using an adjective or verb carrying negative opinion. This intuition will be confirmed by using a statistical method.

In order to evaluate the quality of our dictionary, an experiment was performed using a set of English words already classified by [30] and [17]. The new lexicon contains 4,210 opinions words (2,293 negative words and 1,914 positive words). With this new opinion dictionary, the technique classifies more reviews (a gain of 4% essentially on negative ones) but prediction results are worse than previous: 0.67 for precision, 0.65 for recall and 0.66 for F_{score}. See Table 11.4.

The explanation for these results is certainly a lexicon less adapted to this corpus. It is a more general lexicon whereas our lexicon was built with words appearing regularly in a similar corpus.

	Pos. reviews	Neg. reviews
Predicted pos. reviews	7,027	3,743
Predicted neg. reviews	1,165	3,716

Table 11.4 Confusion matrix obtained with *General Inquirer* lexicon

These new results show the same problem with negative reviews, though this second lexicon contains more negative words. This confirms our first idea that negation is an important part of better interpretation of negative reviews.

11.3.1.3 Analyzing Errors

There are errors that we wish to discuss in particular - one associated with review that our method could not rate and the other error was associated with poorly rated reviews.

Unrated reviews

There are several explanations why reviews are not rated:

- Gaps in the hand crafted lexicon. Examples: "wooohooo film", "watched it all the time when i was younger", "no please no", "I can not remember story".
- Presence of adjectives expressing sentiments or beliefs that can be associated with different opinion. Examples: "so romantic", "weird movie", "I was afraid", "it is very sad".
- Presence of as many positive words as negative words. In this case, the classifier considers the review neutral. Examples: "bad dish good opinion", "not bad - not great either", "really bad film, I thought it would be a lot better".
- Some of the reviews are emptied by the NLP pretreatment. They don't contain any verbs or adjectives.

Poorly rated reviews

The majority of errors are due to negation words which are not considered in this approach. The solution could be to change opinion polarity when a negation is present in the review. Indeed, reviews are very short, statistically, that is, these reviews are composed of only one sentence, thus the negation modifies the polarity of all the verbs or adjectives present. Perhaps what is needed is dependency parsing in order to find which word the negation is related to, and thus reversing the polarity only on the involved words.

We can find numbers of ironic or sarcastic sentences as "fun 4 little boys like action heroes and stuff u can get into it :p" which was rated negatively by the author whereas we rate it positively.

11.3.2 Machine learning approach

Let us first present the method we used and then comment on the results. We will analyze the prediction quality of our classifier, and we will show how a deeper exploration gives information on the Internet users' writing style.

11.3.2.1 Compression-Based Averaging of Selective Naive Bayes Classifiers

In this section, we summarize the principles of the method used in the experiments. This method, introduced in [3], extends the naive Bayes classifier owing to optimal preprocessing of the input data, to an efficient selection of the variables and to averaging the models.

Optimal discretization

The naive Bayes classifier has proved to be very effective on many real data applications [19, 11]. It is based on the assumption that the variables are independent in each output label, and relies on an estimation of univariate conditional probabilities.

The evaluation of the probabilities for numeric variables has already been discussed in the literature [9, 22]. Experiments demonstrate that even a simple equal width discretization brings superior performance compared to using a Gaussian distribution.

In the MODL approach [2], the discretization is turned into a model selection problem. First, a space of discretization models is defined. The parameters of a specific discretization are the number of intervals, the bounds of the intervals and the output frequencies in each interval. Then, a prior distribution is proposed on this model space. This exploits the hierarchy of the parameters: the number of intervals is first chosen, then the bounds of the intervals and finally the output frequencies. The choice is uniform at each stage of the hierarchy.

Finally, the multinomial distributions of the output values in each interval are assumed to be independent. A Bayesian analysis is applied to select the best discretization model, which is found by maximizing the probability $p(Model|Data)$ of the model given the data.

Owing to the definition of the model space and its prior distribution, the Bayes formula is applicable to derive an exact analytical criterion to evaluate the posterior probability of a discretization model.

Efficient search heuristics allow us to build the most probable discretization given the data sample. Extensive comparative experiments report high performance.

Bayesian Approach for Variable Selection

The naive independence assumption can lead to misleading inference when the constraints are not respected. In order to better deal with highly correlated variables, the selective naive Bayes approach [20] uses a wrapper approach [18] to select the subset of variables which optimizes the classification accuracy.

Although the selective naive Bayes approach performs quite well on datasets with a reasonable number of variables, it does not scale on very large datasets with hundreds of thousands of instances and thousands of variables, such as in marketing applications or, in our case, text mining. The problem comes from the search algorithm, whose complexity is quadratic in the number of the variables, and from the selection process which is prone to over fitting.

In [3], the overfitting problem is solved by relying on a Bayesian approach, where the best model is found by maximizing the probability of the model given the data.

The parameters of a variable selection model are the number of selected variables and the subset of variables. A hierarchic prior is considered, by first choosing the number of selected variables and then choosing the subset of selected variables. The conditional likelihood of the models exploits the naive Bayes assumption, which directly provides the conditional probability of each label. This allows an exact calculation of the posterior probability of the models.

Efficient search heuristics with super-linear computation time are proposed, on the basis of greedy forward addition and backward elimination of variables.

Compression-Based Model averaging

Model averaging has been successfully used in Bagging [4] with multiple classifiers trained from re-sampled datasets. In this approach, the averaged classifier uses a voting rule to classify new instances. Unlike this approach, where each classifier has the same weight, the Bayesian Model Averaging (BMA) approach [13] weights the classifiers according to their posterior probability.

In the case of the selective naive Bayes classifier, an inspection of the optimized models reveals that their posterior distribution is so sharply peaked that averaging them according to the BMA approach almost reduces to the MAP model. In this situation, averaging is useless.

In order to find a trade-off between equal weights as in bagging and extremely unbalanced weights as in the BMA approach, a logarithmic smoothing of the posterior distribution called compression-based model averaging (CMA) is introduced in [3].

Extensive experiments have demonstrated that the resulting compression-based model averaging scheme clearly outperforms the Bayesian model averaging scheme.

11.3.2.2 Results

Recall that in a machine learning approach there is no *a priori* data: We used the original reviews by the authors; all tokens were rendered into lower case letters and all punctuation marks were removed. The reviews were processed as a bag of words. We trained our system on a corpus containing 20,000 positive reviews and 20,000 negative. We then used a test corpus to evaluate the training regimen.

The training corpus comprised 24,825 tokens: Our system found only 305 tokens to be informative. Very few of the tokens are very informative (Figure 11.2) and are classified according an associated *level* value. The *level* value is directly related to the posterior probability of a discretization model, with a 0-1 normalization. The zero level indicates that token has no information, and the unity value (the value only approaches unity asymptotically) suggests that the token has maximum information.

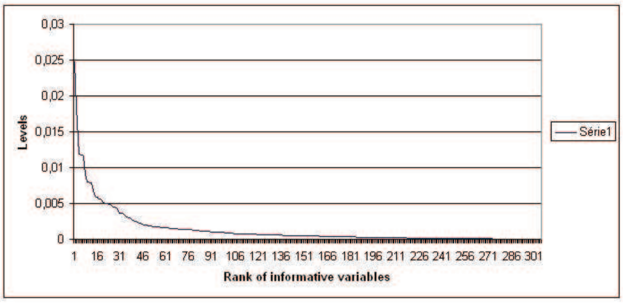

Fig. 11.2 Evolution of levels of informative variables

The majority of words having a positive level express opinion, but other words also appear in this list.

We found that some of the grammatical words that have a very high frequency in general language texts are comparatively rare in our movie review corpus: over 70% of the reviews do not have determiners *a*, *the* together with the conjunction *and*. It turns out that the presence of relatively ’rare’ words in the reviews that do contain these grammatical words have a positive polarity.(See Table 11.5) .

The occurrence of the domain-specific words, for example *film*, *movie* and qualifiers *action*,*thriller*, are comparatively rare in our corpus: The more formal *film* occurs only in 8% of the reviews and the informal *movie* in around 23% of the reviews. However, it appears that the reviewers who write positive reviews use these

Token	Token *f*	Tot. Reviews	% Neg Rev	% Pos Rev
a	0	31,177	52.31	47.69
	1	7,960	43.05	56.95
	2	863	30.48	69.52
the	0	29,179	52.29	47.71
	1	8,923	45.77	54.23
	2	1,898	34.61	65.39
and	0	33,725	52.54	47.46
	1	4,439	40.17	59.83
	2,3,or 4	1,619	29.96	70.04
	5	217	5.99	94.01

Table 11.5 Information 'content' of grammatical tokens *a,and* and *the*

terms in around 2 in 3 of all the those reviews comprising these domain words. The even rare qualifiers,*action* and *thriller* appearing in no more than 1% and 2% respectively of all the 40,000 reviews, are used in 75% (and 90% respectively) of the positive reviews that have the qualifiers. (See Table 11.6).

Token	Token *f*	Tot. Reviews	% Neg Rev	% Pos Rev
movie	0	30,849	53.55	46.45
	1	9,151	38.05	61.95
film	0	37,013	51.33	48.67
	1	2,987	33.58	66.42
action	0	39,262	50.51	49.49
	1	738	23.04	76.96
thriller	0	39,725	50.29	49.71
	1	275	8.00	92.00

Table 11.6 Information 'content' of domain specific words

Intuitively, negation words appear mainly in negative polarity reviews despite the fact that two of these types of words *did'nt* and *not* appear only in 3% and 10% of all our reviews (Table 11.7). This explains the weak score of precision for positive reviews and recall for negative reviews in the previously approach. Indeed, we can note that negation terms appear much more in negative reviews than in positive reviews .

Token	Token *f*	Tot. Reviews	% Neg Rev	% Pos Rev
not	0	36,189	48.51	51.49
	1	3,811	64.13	35.87
did'nt	0	38,896	49.42	50.58
	1	1,104	70.47	29.53

Table 11.7 Information content of *not* and *did'nt*

Concerning the opinion prediction, the confusion matrix of results in Table 11.8 shows that this time, all the reviews are classified. Scores obtained are 0.77 for precision, 0.76 for recall and F_{score}. They are better than those obtained with the classic naive Bayes classifier (approximately 0.70 for the three indicators). Results are equivalent to our linguistic results regarding to the F_{score}, but, recall is significantly better for negative reviews (0.82 instead of 0.43), as is the precision on positive reviews (0.80 instead of 0.69). On the contrary, recall is worse for positive reviews (0.70 instead of 0.97) and so is the precision on negative reviews (0.74 instead of 0.93). The ML technique provides balanced results for each class, but overall it does not outperform the NLP approach.

	Pos. reviews	Neg. reviews
Pos. reviews predict	7,060	1,793
Neg. reviews predict	2,940	8,207

Table 11.8 Confusion matrix obtained with Machine Learning

11.4 Conclusion and Prospects

We have tested and evaluated two approaches for opinion extraction. The first one consists of building a lexicon containing opinion words using *low-level* NLP techniques. This lexicon facilitates the classification of reviews as either positive or negative. The second method consists of using a machine learning technique to predict the polarity of each review.

We used data from the flixster website as a benchmark to evaluate those two recommendation methods, using part of the opinion corpus as a learning testbed and the rest of it to evaluate classification performance: we were are able to discriminate the qualities of the two techniques according to various criteria. In the rest of this conclusion, we synthesize our results, trying to provide the reader with an understanding of each technique's specificity and limitation.

The results obtained with the machine learning (ML) technique appear to provide an inherently deeper understanding of how the authors express themselves according to what they thought about a movie. Indeed, they show that people generally write more when they appreciated the movie for example, giving more detailed reviews of movies features. It turns out that opinion words are not the only opinion indicator, at least for this kind of corpus.

Independently of the analysis technique, an important issue with automating opinion extraction is that we cannot expect a machine to predict good polarity for each review. Consider for instance the sentence "*Di Caprio is my future husband*": it does not indicate whether the author appreciated the film or not. Thus our aim is not to know the polarity of each review but to have the best possible classification. Improvement of prediction results with ML can be obtained by using an indecision threshold. i.e. when the probability to for a good prediction is too weak, we can decide not to classify the review.

With the NLP technique, this problem does not exist because reviews which do not contain opinion words are not classified. However, results from this technique can be improved. For instance, detecting negations would be an important step forward. Indeed, ML results show that negative opinions are often expressed by using words carrying positive opinion associated with a negation. Since our linguistic approach ignores every negation, most of the negative reviews are labeled as positive ones. The best solution is probably to proceed to a dependency parsing. But the kind of prose we are faced with (SMS writing, spelling errors, strange sentence construction . . .) will complicate this step.

The main advantage of the ML technique is that new datasets can be analysed without *a priori* knowledge (i.e. lexicon) and then be deployed with a confidence for both positive and negative reviews. However, the corpus has to be large enough to offer a consistent training dataset and must contain ratings to supervise the training.

This approach may also be used to detect pertinent words and help build dictionary, particularly in the context of Web Opinion Mining, where it is necessary to adapt the lexicon to the *inventive* vocabulary of Internet users' writings.

Contrarily, NLP techniques do not require a learning step, except regular updates to the lexicon. So it can be deployed immediately on a small corpus without rated examples. With a dependency parsing step in order to detect negations, the results could be competitive with ML techniques.

By way of a conclusion, we propose to using a *low-level* NLP approach when the corpus is too small to facilitate good training: the cost of building a lexicon (small ones bring satisfying quality) and designing a negation detection remains reasonable. If the corpus is large enough, ML approaches will be easier to deploy.

To go further, we may explore whether linguistic pretreatments on the corpus for ML techniques can reduce the number of variables (by reducing the vocabulary describing the reviews) without losing information and damaging the quality. We may also focus on a higher level NLP approach and try to explain why people (dis)like movies.

References

1. Farah Benamara, Carmine Cesarano, Antonio Picariello, Diego Reforgiato, and VS Subrahmanian. Sentiment analysis: Adjectives and adverbs are better then adjectives alone. *Proc. Int. Conf. on Weblogs and Social Media (ICWSM)*, 2007. (available at http://www.icwsm.org/papers/paper31.html, seen 24/02/2011)
2. M. Boullé. MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165, 2006.
3. M. Boullé. Compression-based averaging of selective naive Bayes classifiers. *Journal of Machine Learning Research*, 8:1659–1685, 2007.
4. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123Ű-140, 1996.

5. Laurent Candillier, Frank Meyer, and M. Boullé. Comparing state-of-the-art collaborative filtering systems. *International Conference on Machine Learning and Data Mining MLDM 2007*, Leipzig/Germany, 2007.
6. William W. Cohen. Learning trees and rules with set-valued features. *Proc. AAAI*, pages 709–716, 1996.
7. Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *WWW '03 Proceedings 12th Int. Conference on World Wide Web*. (available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.2424&rep=rep1&type=pdf) 2003.
8. E. Guimier de Neef, M. Boualem, C. Chardenon, P. Filoche, and J. Vinesse. Natural language processing software tools and linguistic data developed by France Telecom R&D. Indo European Conference on Multilingual Technologies, Pune, India, 2002.
9. J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th Int. Conf. on Machine Learning*, pages 194–202. Morgan Kaufmann, San Francisco, CA, 1995.
10. Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. Large-scale sentiment analysis for news and blogs. *Proc. Int. Conf. on Weblogs and Social Media (ICWSM)*, 2007. (available at http://www.icwsm.org/papers/paper26.html, seen 24/02/2011)
11. D.J. Hand and K. Yu. Idiot Bayes ? not so stupid after all? *International Statistical Review*, 69(3):385–399, 2001.
12. Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives, 1997. In *ACL '98 Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL) and 8th Conference of the European Chapter of the ACL.* pages 174–181 (DOI:10.3115/979617.979640)
13. J.A. Hoeting, D. Madigan, A.E. Raftery, and C.T. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417, 1999.
14. Minqing Hu and Bing Liu. Mining and summarizing customer reviews, 2004. *Proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '04).* Seattle, WA, USA - August 22 - 25, 2004. pages 168–177. *DOI* : 10.1145/1014052.1014073
15. Minqing Hu and Bing Liu. Mining opinion features in customer reviews, In *(Ed.) A. G. Cohn AAAI'04 Proceedings of the 19th Nat. Conf. on Artificial Intelligence.* pages 755–760, 2004.(Available at https://www.aaai.org/Papers/AAAI/2004/AAAI04-119.pdf).
16. Thorsten Joachims. Making large-scale support vector machine learning practical Thorsten Joachims, 1999. In (Eds.)Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola (Eds.).. MIT Press, Cambridge, MA, USA. (Eds.) . *Advances in Kernel Methods: Support Vector Learning.* MIT Press Cambridge, MA, USA, pages 169–184
17. Edward Kelly and Philip Stone. *Computer recognition of English word senses*, North Holland Publishers., 1975.
18. R. Kohavi and G. John. Wrappers for feature selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.
19. P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *AAAI–10, 10th National Conf. on Artificial Intelligence*, pages 223–228. San Jose: AAAI Press, 1992.
20. P. Langley and S. Sage. Induction of selective Bayesian classifiers. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 399–406. Morgan Kaufmann, 1994.
21. Dekang Lin. Automatic retrieval and clustering of similar words, 1998. In *COLING '98: Proceedings of the 17th Int. Conf. on Computational linguistics - Volume 2* Association for Computational Linguistics Stroudsburg, PA, USA. pages 768-774 (DOI>10.3115/980691.980696)
22. H. Liu, F. Hussain, C.L. Tan, and M. Dash. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, 4(6):393–423, 2002.
23. Bing Liu, Minqing Hu and Junsheng Cheng. Opinion Observer: Analyzing and Comparing Opinions on the Web. 2005 In *WWW 2005, May 10-14, 2005, Chiba, Japan.*, (Available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.81.7520&rep=rep1&type=pdf)

24. George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Introduction to Wordnet: An on-line lexical database. *Int J Lexicography*, 3(4):235–244, 1990.
25. Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web, 2002. *In Proc. of the 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 341–349. New York:ACM
26. Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In *K-CAP '03 Proc. of the 2nd Int. Conf. on Knowledge Capture*. pages 70–77, 2003.
27. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Techniques Thumbs up? Sentiment Classification using Machine Learning Techniques, 2002. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, July 2002*, pages. 79–86.
28. Fernando Pereira, Naftali Tishby, and Lillian Lee. Distributional clustering of English words, 1994. In *ACL '93 Proceedings of the 31st Annual Meeting of Association for Computational Linguistics*, pages 183–190 (DOI>10.3115/981574.981598)
29. Robert E. Shapire and Yoram Singer. Boostexter: A boosting-based system for text categorization, 2000.
30. Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, Daniel M. Ogilvie, and associates. *The General Inquirer: A computer approach to content analysis*, 1996. MIT Press Cambridge, MA, USA
31. Peter D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, 2002. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, Pennsylvania, USA, July 8-10, 2002. pages 417-424. (Available as *National Research Council of Canada Report No. 44946* at http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=8914166.
32. Peter D. Turney and Michael L. Littman. *Unsupervised learning of semantic orientation from a hundred billion-word corpus*, 2004. (Available at http://arxiv.org/ftp/cs/papers/0212/0212012.pdf)
33. Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995
34. Theresa Wilson and Janyce Wiebe, 2003. Annotating Opinions in the World Press. In *4th SIGdial Workshop on Discourse and Dialogue (SIGdial-03). ACL SIGdial.* (Available at http://www.cs.pitt.edu/ wiebe/pubs/papers/sigdial03FixedLater.pdf, site visited 7 Feb 2011).
35. Theresa Wilson, Janyce Wiebe, and Rebecca Hwa, 2004. Just how mad are you? finding strong and weak opinion clauses. In *(Ed.) A. G. Cohn AAAI'04 Proceedings of the 19th Nat. Conf. on Artificial Intelligence.*, pages 761–767. (Available – http://www.aaai.org/Papers/AAAI/2004/AAAI04-120.pdf, site visited 7 Feb 2011).
36. Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences, 2003.
In *Proceeding EMNLP '03 Proceedings of the 2003 conference on Empirical methods in natural language processing Association for Computational Linguistics.*, pages 129-136.