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

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

France Telecom RD, TECH / EASY
2 avenue Pierre Marzin, 22300 Lannion, FRANCE
firstname.name@orange-ftgroup.com

Abstract

Community sites are by nature dedicated places to express and publish opinions. *www.flixster.com* is an example of participative web site, with dozens of millions of enthusiasts sharing their feelings/views on movies, providing positive feedback as well as vivid critics. For anyone interested in understanding net user expectations, such web sites are of major importance because they offer the opportunity to probe huge volume of user generated contents. But to actually benefit from those large amount of data, one has to be able to automatically extract users opinions. This is the challenge we tackle in this paper. Our goal is to exploit the various reviews written by a user in order to compute a model which can then be used to predict the user's verdict on a movie. We explore two different methods to extract opinions. The first one relies on a machine learning technique based on a naive bayesian classifier. The second method consists in applying NLP techniques to process opinions and build dictionaries : those dictionaries are then used to determine the polarity of a comment given the words it may contain. We did apply those two approaches to contents from flixster.com : the results we provide enable us to discern the most appropriate approach for a given set of data.

1. Introduction

With the spread of high speed access to the internet and new technologies, there is a tremendous growth in online music and video market. 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 sell content on-line. Actually recommendation is not a new concept, it is already used on internet 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 shows there is room for innovation.

Candillier et al. (2007) 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 problems: the hollow nature of 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 very low number of products. This phenomenon makes the comparisons between profiles risky. In the recommendation field, the difficulty to collect descriptions about users taste (rates, interests ...) and content (metadata) is a recurrent problem.

In order to compensate for these problems, a new research lane is open : 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 should shift to 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. Considering Internet like a wide open catalogue opens the way to learn the tastes of a large number of people : in the future, it could be possible to describe fan profiles, film typology or to discover new models to describe films and provide decisive pieces of advice on which films to recommend.

Motivated by this potential shift in recommendation, the purpose of this study is to extract opinions from movie reviews published on community sites[1]. Our main objective is to establish a user profile based on what he/she declares to like or dislike in movies through his/her published writings (blogs, forums, personal page on the flixster website ...).

We focus on two different approach to do so. The first method consists in applying a machine learning technique to classify textual reviews into either a positive or negative class. The second method consists in using a NLP approach to build an opinion dictionary and to detect words carrying opinion in the corpus and then predict an opinion.

We did apply those two approaches to data from the flixster web site. 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.

[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))

2. Related work

Opinion extraction in trademark product reviews is a stake so important that a lot of researches have been done in the field. Dave et al. (2003) present a method for automatically classify 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.
In an article by Morinaga et al. (2002), the authors explain how they verify reputation of targeted products by analyzing customers' opinions. They start by seeking 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 it by locating in reviews opinion words which were indexed previously in an *opinion dictionary*.
Other articles present works which are closely related to the previous one like Turney (2002), which classifies reviews in two categories: recommended and not recommended, or Wilson et al. (2004) which categorizes sentences according to polarity and strength of opinion, or Nasukawa and Yi (2003) which seeks opinions on precise subjects in documents.

We find two distinct types of methods: methods based on Natural Langage Processing (NLP) techniques and methods based on machine learning techniques. These two methods types can also be combined.

2.1. Linguistic methods of opinion analysis

Liu et al. (2005) describe a system which compares competitive products by using product reviews left by the Internet users. The system, named *Opinion Observer*, finds features such as pictures, battery, zoom size, etc. in order to explain the sentiment about digital cameras. They designed a supervised pattern discovery method to automatically identify the product features described in the reviews. A language pattern constrains a sequence of words and can be instanciated in many ways: *included/VERB [feature]/NOUN */VERB stingy/ADJECTIVE*. From the multiple instanciations, they extract association rules to find out what describes each feature: $noun1 noun2 \Rightarrow [feature]$.
They only keep the statistical relevant rules, and then generate language patterns: *noun1 [feature] noun2*. They analyse the reviews with those patterns and compare the opinion on each of these characteristics. A component decides the orientation of the extracted feature according to the words extracted near the features. Then they classify 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 complete system based on the fine analysis of sentences and a process counting the Sentiment signs (words, expressions, patterns). Like many others (Morinaga et al., 2002; Turney, 2002; Wilson et al., 2004; Nasukawa and Yi, 2003), they need an *Opinion Dictionary* with as more words or expressions as possible expressing opinions. To build such a dictionary, different techniques are possible but they have all the same first steps : creating, manually, a set of words and expressions carrying opinion; this set is called *seed*; from the seed, the aim is to find other words and expressions yielding opinions and classify them according to their semantic orientation (positive, negative, but seldom neutral).

Lexicon can be built by using machine learning techniques. For example, Hatzivassiloglou and McKeown (1997) or Turney and Littman (2004) use an unsupervised learning algorithm to associate new words with words already registered. Pereira et al. (1994) and Lin (1998) describe methods to discover synonyms by analyzing words collocation.
Linguistic methods exploit syntactic and grammatical analyzis in order to extend the lexicon. Hatzivassiloglou and McKeown (1997) use conjonctions between a word which semantic orientation is known and a not classified word. For example, if there is the conjunction *and* between two adjectives, we can consider that the terms have a close signification. On the contrary, if there is the conjunction *but* between two adjectives, we can suppose that the two words have a different semantic orientation.

Turney (2002) uses a little more complex patterns. They 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 their neighbours. Each time they meet an adverb or an adjective, they extract a pair of consecutive words:

- Adjective with noun
- Adverb with adjective when they are not followed by a noun
- Adjectif 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 to confirm polarity of the adjective or adverb by giving an outline of the context of the sentence.
This method, counting co-occurences with words semantically oriented and manually selected, is also used in the research by Yu and Hatzivassiloglou (2003) in order to determine which words are semantically oriented, in which direction and the strength of their orientation. To measure more precisely the strength of opinion expressed in a sentence, a mean is to extract adverbs which are associated to adjectives. Indeed, Benamara et al. (2007) 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. A system of attribution of points according to the category of the adverb allows to calculate strengths to adverb-adjective combinations.

Google's work (Godbole et al., 2007) find semantic orientation of new words from WordNet databases (Miller et al., 1993). In a close manner, Hu and Liu (2004a) use sets of synonyms and antonyms present in WordNet to predict semantic orientation of adjectives. In WordNet, words are organised in tree (see figure 1). To determine polarity of a word, they traverse the trees of synonyms and antonyms of this word and if they find a seed word in the synonyms, they allocate the same class, but if they find seed word in the antonyms, they allocate the opposite class. If they do not find any seed word, they remake the analysis with synonyms and antonyms, and so on until finding a seed word.

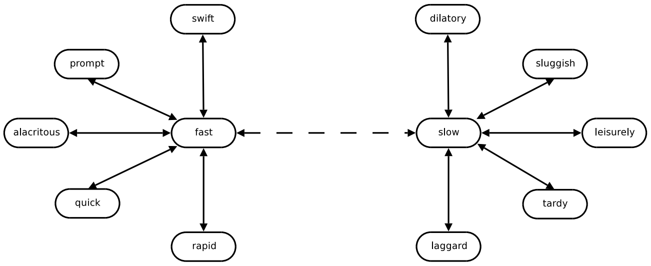

Figure 1: Tree of synonyms and antonyms in WordNet (full arrow = synonyms, dotted arrow = antonyms)

We think this method is a little too random because words can have different meaning according to the context and thus they can have synonyms not significating the same thing. For example, the word *like* has for synonym *love* but in the sentence *It is like that*, it has not the same meaning. This method finds a positive opinion in this sentence whereas there is not. But using the same method after having linguistically processed the corpus before, i.e. grammatical analysis, could be more effective. For the precedent example, if the seed word is *like/VERB*, we would not find opinion in the sentence *It is like that*.

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 semantic 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 (Yu and Hatzivassiloglou, 2003); with another strategy, the last term carrying opinion determines the sentence polarity (Hu and Liu, 2004a). Otherwise, we can extract opinion one by one associated with the feature it refers to (Wilson et al., 2004; Hu and Liu, 2004b).

2.2. Machine learning for opinion analysis

Systems using learning machine techniques generally classify textual comments in two classes (positive and negative), but sometimes seek to predict five rates or more. These supervised classification methods consider that a comment describes only one product and try to predict the rate given by the author.

Many methods use NLP techniques to prepare the corpus. Wilson and Wiebe (2003) expose how to label opinion words with an intensity; Wilson et al. (2004) test three different learning methods, frequently used by the linguists: *BoosTexter* (Shapire and Singer, 2000), *Ripper* (Cohen, 1996) and *SVMlight*, the light version of Support Vector Machine by Joachims (1998). The last one obtains the best results on their annotated corpus. Pang et al. (2002) use a naive bayes classifier and a classifier maximizing the entropy. In the same way, in order to characterize what is appreciated or not in a sentence, Nigam and Hurst (2004) combine a *parsing* technique with a bayes classifier to associate polarity to sets of themes.

In addition, Pang et al. (2002) and Dave et al. (2003) show that corpus preparation with a lemmatizer or a negation detection for example, is useless. In order to predict reviews opinion, these two papers explore some learning methods and show that they are more powerful than *parsing* methods followed by a calculation as presented in the previous section. Considering comments as bags of words and using the relevant learning technique lead to 83% of good predictions. We will see in the following part of this paper that our own experiments confirm those conclusions.

3. Our two approaches

We compare in this section two opinion analysis approaches with their results. The initial corpus is composed of 60,000 films reviews rated by authors. Half of them express positive opinion and the other half, negative opinion. We keep a set of 10,000 positive and 10,000 negative for the tests. Both approaches are tested on the same test corpus.

The main difficulty of this corpus is the small size of reviews (twelve words on average). This makes opinion extraction difficult even for human sometimes. Moreover, the corpus is composed of textual messages very similar to forum messages. They present common characteristics such as accumulation of the ponctuation (" !!! "), smileys (" :-) "), SMS language (" ur ", " gr8 ") or words stretching (" veryyyyy cooooool ").

Each review in our corpus has a rate given by the author (0 to 5 stars) and our final aim is to predict this rate. We have decided to classify reviews in two classes. Reviews with a rate lower than three stars are considered as negative reviews, other as positive reviews. Here follow examples of reviews with their rate (table 1).

3.1. Linguistic approach

3.1.1. Technique

First step for this method is, as it was seen in the state of the art, the building of a dictionary of opinion words. We have used linguistic techniques to do that.

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 awsome!
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 awsome!! (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 1: Examples of reviews

In first, we have separated all reviews according to their rate. For each review category (set of reviews rated 1 star, set of reviews rated 2 stars ...), we have applied a shallow parser (de Neef et al., 2002) to lemmatize and tag the text. We have 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 increased 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 expressed both opinion polarities.

183 opinion words have been classified in two classes, positive words (115) and negative words (68), in this manner. The table 2 presents a part of the lexicon. Let us note this dictionary was not made on 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 2: Part of hand crafted lexicon

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

We have not performed any sophisticated NLP techniques such as a grammatical structural analysis. But keeping only verbs and adjectives avoid misinterpretations of words such as ”*like*” which can hold different roles in a sentence. Regarding the review style, we can suppose that NLP tools would not face the bad English writing, and indeed, apply more complicated NLP treatments would probably became rapidly costly in adaptation to this specific corpus.

3.1.2. Results

This method allowed to rate 74% of films 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}.

Here follows the functions used to calculate these values:

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

The confusion matrix of results is presented in table 3.

	Pos. reviews	Neg. reviews
Predicted pos. reviews	8089	3682
Predicted neg. reviews	218	2823

Table 3: Confusion matrix obtained with the hand crafted lexicon

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

The largest difficulty is to determine polarity of negative reviews. Indeed, the recall of negative reviews is 0.43, whereas it is 0.97 for positive reviews. Contrary, precision of positive reviews (0.69) is worse than precision of negative reviews (0.93).

This phenomon can be due to the dictionary we used: the positive category contains almost twice more words than negative category. But the problem is not the detection of negative reviews but their bad interpretation. These results lead us to think that people use negation to express their bad feelings sometimes without using any adjective nor verb carrying negative opinion. This intuition will be confirmed with the results of statistic approach.

To check quality of our dictionary, we have remade this experiment by using a English words set already classified by Stone et al. (1966) and Kelly and Stone (1975). 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 results of prediction are worse than previously: 0.67 for precision, 0.65 for recall and 0.66 for F_{score}. See the confusion matrix of results in table 4.

The explanation for these worse results is certainly a lexicon less adapted for this corpus. It is a lexicon more general whereas our homemade lexicon was build with

	Pos. reviews	Neg. reviews
Predicted pos. reviews	7027	3743
Predicted neg. reviews	1165	3716

Table 4: Confusion matrix obtained with General Inquirer lexicon

words appearing regularly in a similar corpus.

These new results show the same problem with negative reviews, although this second lexicon contains more negative words. This confirms our first idea, negation is an important point to well interpret negative reviews.

3.1.3. Observation of the errors

Not rated reviews There are several explanations why reviews are not been 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 according to people. 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 classifior considers the review as neutral. Examples: ”bad dish good opinion”, ”not bad - not great either”, ”really bad film, I thought it would be alot better”.
- Some of the reviews get empty after NLP pretreatment. They are not containing any verbs nor adjectives.

Bad rated reviews Majority of errors are due to negation words which are not considered. The solution could be to change opinion polarity when a negation is present in the review. Indeed, reviews are very short so we can think that, statistically, they are composed of only one sentence, thus the negation modify on all the verbs or adjectives present. If this method is not satisfying, the idea could be to do a 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 heros and stuff u can get into it :p” which was rated negatively by the author whereas we rate it positively.

3.2. Machine learning approach

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

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 Boullé (2007), extends the naive Bayes classifier owing to optimal preprocessing of the input data, to an efficient selection of the variables and to an averaging of the models.

Optimal discretization The naive Bayes classifier has proved to be very effective on many real data applications (Langley et al., 1992; Hand and Yu, 2001). It is based on the assumption that the variables are independent within each output label, and simply relies on the estimation of univariate conditional probabilities.

The evaluation of the probabilities for numeric variables has already been discussed in the literature (Dougherty et al., 1995; Liu et al., 2002). Experiments demonstrate that even a simple equal width discretization brings superior performance compared to the assumption using a Gaussian distribution.

In the MODL approach (Boullé, 2006), 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 prior 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 from each other. A Bayesian approach 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 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 harm the performance when violated. In order to better deal with highly correlated variables, the selective naive Bayes approach (Langley and Sage, 1994) exploits a wrapper approach (Kohavi and John, 1997) 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 both from the search algorithm, whose complexity is quadratic in the number of the variables, and from the selection process which is prone to overfitting.

In Boullé (2007), the overfitting problem is tackled 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 second 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 heuristic 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 exploited in Bagging Breiman (1996) using 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 (Hoeting et al., 1999) 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 Boullé (2007).

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

3.2.2. Results

With this approach we have no *a priori* on the data. Indeed we hang on all reviews as the authors wrote them and process them as bags of words. We do not treat the data with NLP tool. We apply to the text only two treatments; we put in lowercase all letters and we delete the punctuation.

We learned on a corpus containing 20,000 positive reviews and 20,000 negative. We tested this training on the same test corpus than in the precedent method.

Let us start by commenting training results. The tool found 305 informative variables out of the 24,825 words present in the learning corpus. Little of them are very informative as shown in the figure 2. They are classified according their *level* value. The *level* is directly related to the posterior probability of a discretization model, with a 0-1 normalization. Its value is 0 in case a no informative input variable and is asymptotically equal to 1 in case of perfectly informative input variable.

Majority of words having a positive level express opinion. But other words appear in this list.

These results allow to learn opinion vocabulary but also information on the style of the reviews. Informations supplied by ”*and*” (table 5) indicate that authors write longer texts with more details when they talk about a movie they appreciated.

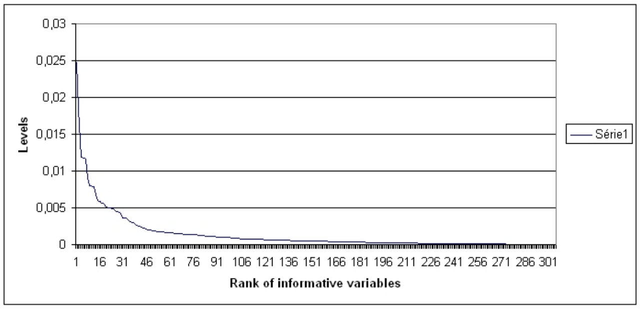

Figure 2: Evolution of levels of informative variables

Value	Neg. review	Pos. review	Frequency
]-inf; 0.5[	0,525397	0,474603	33725
[0.5; 1.5[	0,401667	0,598333	4439
[1.5; 4.5[	0,299568	0,700432	1619
[4.5; inf[	0,0599078	0,940092	217

Table 5: Informations of ”*and*”

This phenomenon is specified with other terms present in the list. We find too ”*movie*” (table 6) and ”*film*” (table 7) and link-words as ”*a*” (table 8), ”*the*” (table 9), ”*of*”, ”*in*” . . .

One can think that users have tendency to be more prolix and detail their point of view on film features when they appreciate the movie.

The presence of words as ”*action*” (table 10) and ”*thriller*” (table 11) can confirm this explanation. Authors explain why they appreciated the film and what they appreciated in the film.

Value	Neg. review	Pos. review	Frequency
]-inf; 0.5[	0,535447	0,464553	30849
[0.5; 1.5[	0,380505	0,619495	9151

Table 6: Informations of ”*movie*”

Another observation is the presence of negation words in informative variables. That explain certainly 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 (table 12 and 13).

Concerning the opinion prediction, the confusion matrix of results in table 14 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), also 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). ML technique provides balanced results for each class, but overall it does not outperforms the NLP approach.

Value	Neg. review	Pos. review	Frequency
]-inf; 0.5[	0,513252	0,486748	37013
[0.5; 1.5[	0,335788	0,664212	2987

Table 7: Informations of ”*film*”

Value	Neg. review	Pos. review	Frequency
]-inf; 0.5[	0,523142	0,476858	31177
[0.5; 1.5[	0,430528	0,569472	7960
[1.5; 2.5[	0,304751	0,695249	863

Table 8: Informations of ”*a*”

Value	Neg. review	Pos. review	Frequency
]-inf; 0.5[	0,522945	0,477055	29179
[0.5; 1.5[	0,457694	0,542306	8923
[1.5; 2.5[	0,346154	0,653846	1898

Table 9: Informations of ”*the*”

Value	Neg. review	Pos. review	Frequency
]-inf; 0.5[	0,505069	0,494931	39262
[0.5; 1.5[	0,230352	0,769648	738

Table 10: Informations of ”*action*”

Value	Neg. review	Pos. review	Frequency
]-inf; 0.5[	0,502907	0,497093	39725
[0.5; 1.5[	0,08	0,92	275

Table 11: Informations of ”*thriller*”

Value	Neg. review	Pos. review	Frequency
]-inf; 0.5[	0,48512	0,51488	36189
[0.5; 1.5[	0,641301	0,358699	3811

Table 12: Informations of ”*not*”

Value	Neg. review	Pos. review	Frequency
]-inf; 0.5[	0,49419	0,50581	38896
[0.5; 1.5[	0,70471	0,29529	1104

Table 13: Informations of ”*didn't*”

	Pos. reviews	Neg. reviews
Pos. reviews predict	7060	1793
Neg. reviews predict	2940	8207

Table 14: Confusion matrix obtained with Machine Learning

4. Conclusion, prospects

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

We used data from flixster 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. Thanks to those experiments, we 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 specificity and limitation.

While digging into the results obtained with the machine learning (ML) technique, it seems that it inherently provides a deeper understanding of how the authors express themselve according to what they thought about a movie. Indeed, results show that people generally write more when they appreciated the movie for example, giving more detailled 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 (including indetermination). Improvement of prediction results with ML can be obtained by using an indecision threshold. i.e. when the probability to have a well prediction is too weak, we can decide not to classify the review.

With NLP technique, this problem does not exists because reviews which do not contain opinion words are not classified. However, results of this technique can be improved. For instance detectinng negations would be an important progress. 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 negations, most of the negative reviews are labelled as positive ones. Best solution is probably to proceed to a dependency parsing. But the kind of prose which we are faced with (SMS writing, spelling errors, weird sentences construction . . .) certainly will complicate this step.

The main point characterising ML techniques is that new datasets can be analysed without any *a priori* knowledge (i.e. lexicon) and then quickly deployed with a confortable reliability on both positive and negative reviews. But the corpus has to be large enough to offer a consistent training dataset and has to contain rates to supervise the training, which is not always the case.

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

Contrary, NLP technique does not require learning step, except regular updates of 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 if not more often.

As a conclusion, we propose to use a *low-level* NLP approach when the corpus is too small to have a 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 approach will be easier to deploy.

To go further, we may explore if linguistic pretreatments on the corpus for ML technique 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.

5. References

Farah Benamara, Carmine Cesarano, Antonio Picariello, Diego Reforgiato, and VS Subrahmanian. 2007. Sentiment analysis: Adjectives and adverbs are better then adjectives alone.

M. Boullé. 2006. MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165.

M. Boullé. 2007. Compression-based averaging of selective naive Bayes classifiers. *Journal of Machine Learning Research*, 8:1659–1685.

L. Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

Laurent Candillier, Frank Meyer, and Marc Boullé. 2007. Comparing state-of-the-art collaborative filtering systems. International Conference on Machine Learning and Data Mining MLDM 2007, Leipzig/Germany.

William W. Cohen. 1996. Learning trees and rules with set-valued features.

Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews.

E. Guimier de Neef, M. Boualem, C. Chardenon, P. Filoche, and J. Vinesse. 2002. Natural language processing software tools and linguistic data developped by france télécom rd. Indo European Conference on Multilingual Technologies, Pune, India.

J. Dougherty, R. Kohavi, and M. Sahami. 1995. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*, pages 194–202. Morgan Kaufmann, San Francisco, CA.

Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. 2007. Large-scale sentiment analysis for news and blogs. ICWSM'2007 Boulder, Colorado, USA.

D.J. Hand and K. Yu. 2001. Idiot bayes ? not so stupid after all? *International Statistical Review*, 69(3):385–399.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives.

J.A. Hoeting, D. Madigan, A.E. Raftery, and C.T. Volinsky. 1999. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417.

Minqing Hu and Bing Liu. 2004a. Mining and summarizing customer reviews.

Minqing Hu and Bing Liu. 2004b. Mining opinion features in customer reviews.

Thorsten Joachims. 1998. Making large-scale support vector machine learning practical.

Edward Kelly and Philip Stone. 1975. Computer recognition of english word senses.

R. Kohavi and G. John. 1997. Wrappers for feature selection. *Artificial Intelligence*, 97(1-2):273–324.

P. Langley and S. Sage. 1994. Induction of selective Bayesian classifiers. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 399–406. Morgan Kaufmann.

P. Langley, W. Iba, and K. Thompson. 1992. An analysis of Bayesian classifiers. In *10th national conference on Artificial Intelligence*, pages 223–228. AAAI Press.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words.

H. Liu, F. Hussain, C.L. Tan, and M. Dash. 2002. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, 4(6):393–423.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: Analyzing and comparing opinions on the web.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1993. Introduction to wordnet: An on-line lexical database.

Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. 2002. Mining product reputations on the web.

Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing.

Kamal Nigam and Matthew Hurst. 2004. Towards a robust metric of opinion.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1994. Distributional clustering of english words.

Robert E. Shapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, Daniel M. Ogilvie, and associates. 1966. The general inquirer: A computer approach to content analysis.

Peter D. Turney and Michael L. Littman. 2004. Unsupervised learning of semantic orientation from a hundred-billion-word corpus.

Peter D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews.

Theresa Wilson and Janyce Wiebe. 2003. Annotating opinions in the world press.

Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. 2004. Just how mad are you? finding strong and weak opinion clauses.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences.