Applying Supervised Opinion Mining Techniques on Online User Reviews

Ion SMEUREANU[1], Cristian BUCUR[2]
[1]Academy of Economic Studies, Bucharest, Romania
[2]ISUD Academy of Economic Studies, Bucharest, UPG Ploiesti, Romania
ion.smeureanu@csie.ase.ro, cr.bucur@gmail.com

In recent years, the spectacular development of web technologies, lead to an enormous quantity of user generated information in online systems. This large amount of information on web platforms make them viable for use as data sources, in applications based on opinion mining and sentiment analysis. The paper proposes an algorithm for detecting sentiments on movie user reviews, based on naive Bayes classifier. We make an analysis of the opinion mining domain, techniques used in sentiment analysis and its applicability. We implemented the proposed algorithm and we tested its performance, and suggested directions of development.

Keywords: *Opinion Mining, Web Content, Mining, Sentiment Analysis, Naïve Bayes*

1 Introduction

The development of internet and web 2.0 technologies, enabled by cost reduction of technological infrastructure, has been an exponential increase in the amount of information in online systems. These very large volumes of information are very difficult to process by individuals, leading to information overload and affecting decision-making processes in organizations. Therefore, providing new techniques for creation of knowledge is important in organizational strategy [1]. Knowledge discovery using automated techniques is relevant for companies' success, promoting research in the development of methodologies, techniques and systems for extracting knowledge from data warehouses and data mining [2].

A large volume of information in current online systems is stored in text form. This is the way information is transmitted on the internet, being the most natural representation form and easier to read by the people [3]. In this context, applying of data mining techniques to the content of web pages, (*text mining* on web pages) or *content mining* become important [4].

Between web mining and data mining are important differences in terms of data collection. In data mining is assumed that the data is already collected and stored in databases, while in web mining, are used special mechanisms, taken from areas such as information retrieval - *IR (Information Retrieval)* and information extraction - *IE (Information Extraction)*, to obtain data and to pre-process them to apply data mining techniques. From the terms of proposed objectives, Web mining is divided into three categories:

- *Web structure mining* - knowledge discovery from hyperlinks to maximize information about the relations between web pages;
- *Web usage mining* - extracting models and patterns of users, from web logs (logs), that stores data access and activities of each visitor to a website, detecting website users requirements;
- *Web content mining* - extracting knowledge from web page content [5].

Text mining techniques use methods of knowledge discovery in unstructured text data. These techniques performing the process of extracting information from the text unstructured format, are used in research areas such as natural language processing (NLP), artificial intelligence and machine learning. The techniques are applied in document classification, content spam detection and trend analysis.

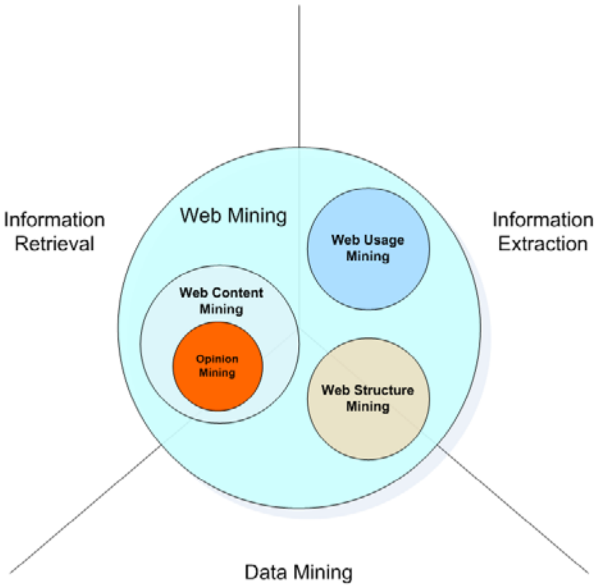


Fig. 1. Taxonomy of Web Mining [6]

An increased importance in knowledge discovery in web page content domain is detection and extraction of opinions or sentiments from textual information. Determination of customer sentiment on a launched new product, based on feedback from web pages is important for assessment of impact and making decision on directions of development. *Opinion mining* is a research domain dealing with automatic methods of detection and extraction of opinions and sentiments presented in a text. Opinion mining applications methods can result in: creation of effective referral systems, financial analysis, market research and product development.

2 Techniques of opinions mining

Currently, has become a practice for websites, to facilitate the expression of opinions by guests and visitors on products marketed or on presented topics. Also, the expansion of social networking, facilitated users posting opinions online. Thus, the content of reviews has increased rapidly, making the big e-commerce sites, or recommendations of products and services sites, to contain hundreds to tens of thousands of reviews per item. The large number of reviews promotes access to useful and relevant information to visitors. They can be used, for example to compare offers from different competitors on the market and make an informed decision about buying a certain offer. It is very difficult for a visitor to read all of them and to form an opinion on the subject or product because:

- in some cases these reviews can be very long and only a few sentences may express opinions or may not contain opinions at all. Navigating only part of the may create a false impression about the topic;
- the user is not familiar with the various metrics used in comparing offers in a certain specialized field.

Also, the large number of reviews makes it difficult for producers to follow reactions of potential customers. They face additional difficulties in pursuing wide range of products, traded on a variety of web sites [7]. So, it is useful to make a system to detect indicators of performance of a product, and domain specific metrics, to summarize the opinions obtained from the large amount of reviews, in several positive and negative aspects.

One major concern in the analysis of user-generated content in online applications is, to determine the polarity of opinions, by extracting the subject whom opinions are addressed and the arguments are based on.

Analyzing existing techniques in opinion mining, we can summarize the following:

1. User opinions and sentiments are easier to extract when they are applied to an entity (eg product, film).
2. The detection process is done most easily at the sentence level, then by aggregating individual results, applying a certain algorithm, we get to document level.
3. There are many problems faced by systems of Opinion Mining, from which we mention language issues [8] such as:
   - modeling syntactic properties and negation is an important aspect in improving systems;
   - the process of linguistic resources adaptation - dictionary / lexicon to various fields and possibility of their reusability [9].
4. For some analyzed entity, we determine

the words or phrase that expresses feelings. The process is conducted in three stages [10]:

a. *Entity determination* – identification of texts that contain information about the entity under review;

b. *Determination of sentiments* – for the text of the previous stage is considered the content of opinions and sentiments, by searching a set of words carrying sentiments, or by prior training a classifier;

c. *Determination of entity - sentiment relationship* - at this stage is analyzed whether opinions extracted are addressed to the entity under review. Usually this is done through a predefined list of patterns.

5. Opinion mining process is centered on a domain, so, a solution determined for a given area (i.e. movie reviews analysis), will not work on another (e.g. foto-camera). The way of expressing feelings varies from one domain to another, the developed model requiring adaptation.

3 Applications of sentiment analysis

There are many areas where sentiment analysis can be used as the following:

- sentiment analysis on financial markets. For investors is important information the analysts and other investors opinions about the stocks of a company, to identify price trends.
- sentiment analysis on products. A company is interested in customers' perceptions about its products. Information may be used to improve products and identifying new marketing strategies [11].
- sentiment analysis on a location. Tourists want to know the best places to visit or famous restaurants. Applying sentiment analysis can be obtained relevant information for planning a trip.
- sentiment analysis on elections. Using sentiment analysis we can identify the voter opinions about a certain candidate.
- analysis on movies or software programs. We can detect users' sentiments from posted reviews on specialized sites.

Social networks like Twitter and Facebook have become a large scale public information source that cannot be ignored. People use them to express their sentiments on various topics. Applying sentiment analysis on these reviews and their automated classification on positive, negative and neutral categories can provide valuable information to companies as market reports.

4 Stages of sentiment analysis

To carry out sentiment analysis are necessary several steps, in which are applied various techniques and methodologies:

a) Data collection and pre-processing

In this stage it is acquired the text that will be analyzed for detection of opinions. It is important, according to the methodology used, to eliminate all matters that not express opinions. In this phase, pre-processing is done to eliminate unnecessary words or irrelevant opinions. It is necessary to extract keywords from the text which can provide an accurate classification. These keywords are usually stored as an array of features A = (A1, A2, ..., An). Each element of array is a word from the original text, called aspect (feature). For every feature, can exist a binary value, indicating the presence or absence of the feature, or a value that expresses the frequency of appearance in the text. Selecting common aspects is necessary so that they express those relevant opinions for sentiment analysis.

b) Classification

In this phase, content polarity is identified. Usually three classes are used for classification: positive, negative and neutral. Classification algorithms used for sentiment analysis, depending on the method used (supervised or unsupervised), require a training set with pre marked examples. It is important to train the model used for classification with domain-specific data. Marking is done by expressing subjectivity and polarity of training sets.

c) Aggregation and presentation of results

At this stage, opinions classification process result obtained at the previous stage is subjected to a process of aggregation after some algorithms to determine the general opinion of the analyzed text. Presentation can be done directly expressing sentiment textual or using graphics.

5 Proposed opinions mining method

The paper aims to analyze user comments on movies, in order to determine the positive or negative opinion. Sites specialized in this field, like RottenTomatos.com and Imdb.com, contain from several tens to several thousands of reviews for each movie that we can extract with a crawler.

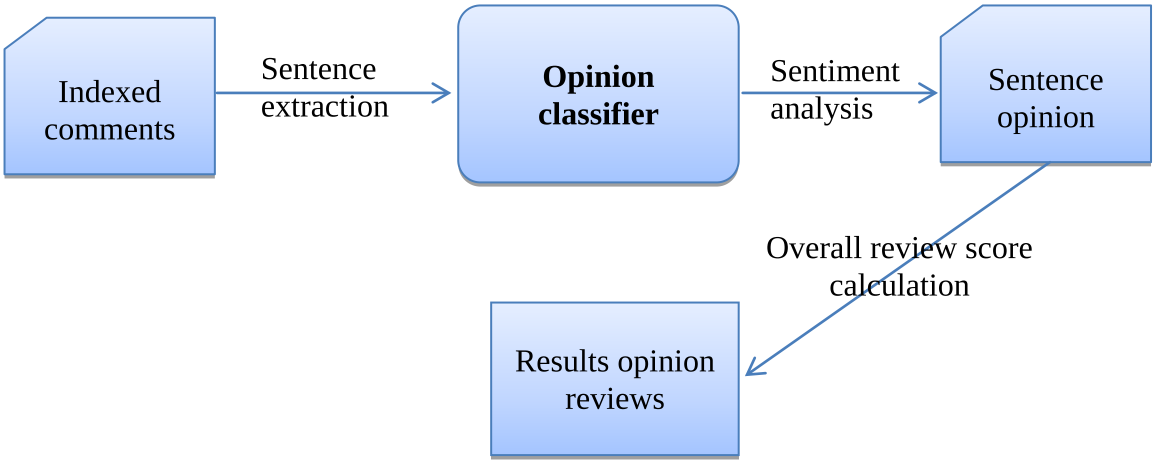


Fig. 2. The overall opinion mining process

In this case we will use, for training, a collection of comments (sentence polarity dataset v1.0) already extracted from these sites and available online at http://www.cs.cornell.edu/people/pabo/movie-review-data/ [12]. This collection contains 5331 sentences already classified as positive and 5331 negative opinions from 2000 comments processed and classified in two categories. Comments usually contain several sentences, but opinion will be determined at sentence level, then later determining overall comment opinion. Obtained collection consists of two files, one for each set of positive and negative opinions, containing one sentence per line, making it easy to process. To extract opinions we will use a Naive Bayesian classifier. This type of classifier has the advantage that it is easy to implement, quickly and generate good results.

6 Using Naive Bayes algorithm

The Naive Bayes classifier is a probability classifier, based on Bayes' theorem. Bayes' theorem specifies mathematically the relation between probability of two events A and B, P (A) and P (B) and conditional probability of event A conditioned by B and event B conditioned by A, P (A | B) and P (B | A). Thus Bayes’ formula is [13]:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

This theorem enables us to determine a conditional probability having the probability of contrary event and independent probabilities of events. Thus, we can estimate the probability of an event based on the examples of its occurrence. Thus, we can estimate the probability of an event based on the examples of its occurrence. In this case, we estimate probability that a document is positive or negative, in a certain context, or the likelihood that an event to take place if it was predetermined to be positive or negative. This is facilitated by the collection of positive and negative examples chosen. The process is naive Bayesian because of how we calculate the probability of occurrence of an event - is the product of probability of occurrence of each word in the document. This presumes that there is no connection between the words. This assumption of independence is introduced to facilitate the construction of classifier, it is not entirely true, and there are words that appear together more frequently than individual.

We estimate the probability of a word with positive or negative meaning by analyzing a series of positive and negative examples and calculating the frequency of each of the classes. This learning process is supervised, requiring the existence of pre-classification examples for training.

Starting from:

$$P(sentiment|sentence) = \frac{P(sentiment)P(sentence|sentiment)}{P(sentence)}$$

we assume that *P(sentence|sentiment)* is the product of *P(word|sentiment)* for all words in a sentence. We estimate P(word|sentiment) as:

$$P(word|sentiment) = \frac{number\ of\ word\ occurences\ in\ class + 1}{number\ of\ words\ beloging\ to\ a\ class + total\ number\ of\ words}$$

The steps in the classification method proposed in the paper are presented in Fig. 3, below:

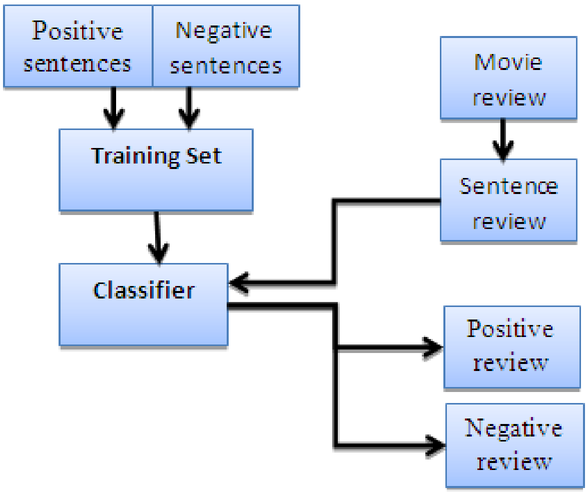
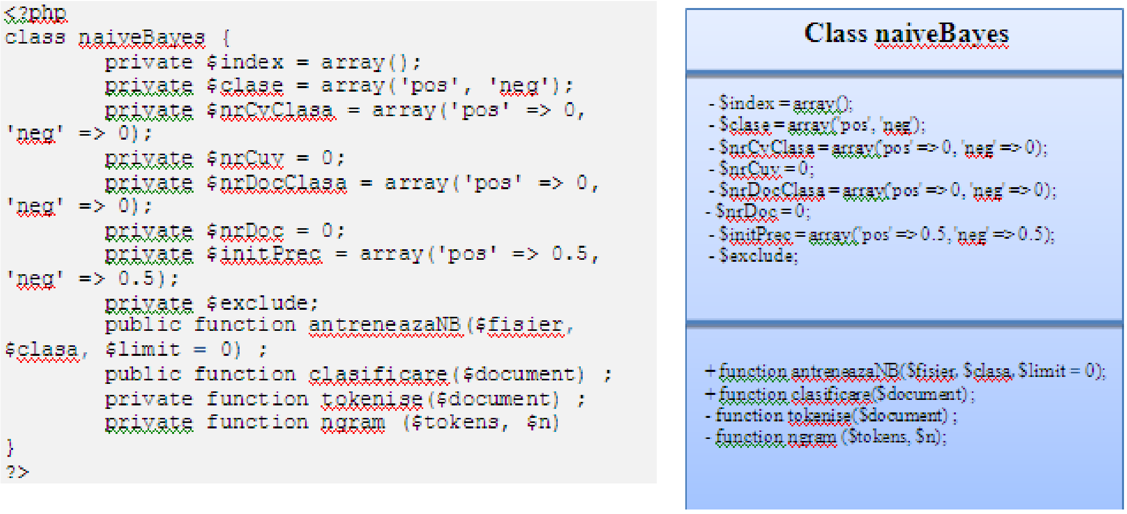


Fig. 3. Stages of classification process

The proposed algorithm has following steps:

```
Initialize P (pos) <- nr_popozitii (pos) / nr_total_propozitii
Initialize P (neg) <- nr_popozitii (neg) / nr_total_propozitii
Tokenize sentence in words
For each class of {pos, neg}:
       For each word in {phrase}
               P (word | class) <- nr_apartii (word | class) 1 / nr_cuv
(class) + nr_total_cuvinte
       P (class) <-P (class) * P (word | class)
Returns max {P(pos), P(neg)}
```

We will implement the algorithm described above in PHP using naiveBayes class: Method *clasificare(),* that effectively implements the classification algorithm using the training collection, starts by calculating the prior probability, before word analysis, based on the number of positive and negative examples. In this case, the collections being equal, the probability is of 0.5. Every sentence is tokenized into words, and for each of the two classes, we compute a score by successive multiplying probability of each word belongs to one of two classes. Finally we return the class with the highest score, the one in which it will classify the sentence.



```
<?php
class naiveBayes {
        private $index = array();
        private $clase = array('pos', 'neg');
        private $nrCvClasa = array('pos' => 0,
'neg' => 0);
        private $nrCuv = 0;
        private $nrDocClasa = array('pos' => 0,
'neg' => 0);
        private $nrDoc = 0;
        private $initPrec = array('pos' => 0.5,
'neg' => 0.5);
        private $exclude;
        public function antreneazaNB($fisier,
$clasa, $limit = 0) ;
        public function clasificare($document) ;
        private function tokenise($document) ;
        private function ngram ($tokens, $n)
}
?>
```

```
Class naiveBayes

- $index = array();
- $clase = array('pos', 'neg');
- $nrCvClasa = array('pos' => 0, 'neg' => 0);
- $nrCuv = 0;
- $nrDocClasa = array('pos' => 0, 'neg' => 0);
- $nrDoc = 0;
- $initPrec = array('pos' => 0.5, 'neg' => 0.5);
- $exclude;

+ function antreneazaNB($fisier, $clasa, $limit = 0);
+ function clasificare($document);
- function tokenise($document) ;
- function ngram ($tokens, $n);
```

Fig. 4. naiveBayes class

```
public function clasificare($document) {
      $this->initPrec['pos'] = $this->nrDocClasa['pos'] / $this->nrDoc;
      $this->initPrec['neg'] = $this->nrDocClasa['neg'] / $this->nrDoc;
      $tokens = $this->tokenise($document);
      $scorClasa = array();
      $tokens  = $this->ngram($tokens,2);
      foreach($this->clase as $clasa) {
          $scorClasa[$clasa] = 1;
          foreach($tokens as $toeken) {
              $st = new Stemmer();
              $token=$st->stem($token);
              if (!$this->toExclude($token)) {
                  $count = isset($this->index[$token][$clasa]) ?
                                   $this->index[$token][$clasa] : 0;
                  $scorClasa[$clasa] *= ($count + 1) /
                                   ($this->nrCvClasa[$clasa] + $this->nrCuv);
              }
           }
           $scorClasa[$clasa] = $this->initPrec[$clasa] * $scorClasa[$clasa];
       }
       arsort($scorClasa);
       return key($scorClasa);
 }
```

Method *antreneazaNB()*, train algorithm using input classified data collections. It reads preclasificate collections of files, loops through the test examples, tokenize sentences into words and stores the frequency of each word in both positive and negative classes for classification:

```
public function antreneazaNB ($file, $clasa, $limit = 0) {
                $fh = fopen($file, 'r');
                $i = 0;
                if(!in_array($clasa, $this->clase)) {
                        echo "Invalid class specified\n";
                        return;
                }
                while($line = fgets($fh)) {
                        if($limit > 0 && $i > $limit) {
                                break;
                        }
                        $i++;
                        $this->nrDoc++;
                        $this->nrDocClasa[$clasa]++;
                        $tokens = $this->tokenise($line);
                        $tokens  = $this->ngram($tokens,2);
                        foreach($tokens as $token) {
                            $st = new Stemmer();
                            $token=$st->stem($token);
                            if (!$this->toExclude($token)){
                                if(!isset($this->index[$token][$clasa])) {
                                        $this->index[$token][$clasa] = 0;
                                }
                                $this->index[$token][$clasa]++;
                                $this->nrCvClasa[$clasa]++;
                                $this->nrCuv++;
                            }
                        }
                }
           fclose($fh);
  }
```

It will be constructed a procedure which will take comment and will split into sentences. Each sentence will be evaluated to establish the determinant sentiment by the vector *$scor* with indexes corresponding to the two classes, and ultimately, it will determine the overall score of comment and its polarity by returning the index of the array with the highest value:

```
$propozitii = explode(".", $doc);
$scor = array('pos' => 0, 'neg' => 0);
foreach($propozitii as $propozitie) {
        if(strlen(trim($propozitie))) {
            clasa = $op->classify($propozitie);
            echo "Clasificare: \"" . trim($propozitie) . "\" - " . $clasa .
"<br/>\n";
            $scor[$clasa]++;
        }
}
var_dump($scor);
arsort($scor);
```

We present the result returned by the program for the next comment:

```
Classification: "It's content to be a solid, well-crafted genre product that knows
what audiences expect from a musical and delivers in spades" - pos
Classification: "Chicago is not quite the masterpiece some of the early reviews
have suggested" - pos
Classification: "The lack of a more experienced director keeps it from being more
than a top-notch screen transfer of a venerated stage work" - neg
Classification: "Nevertheless, the film is funny and exciting, with plenty of
memorable numbers, and it proves for sure that the success of Moulin Rouge wasn't a
fluke" - pos

array
  'pos' => int 3
  'neg' => int 1
Comment Classification:- pos
```

7 Evaluate the performance of algorithm

We use two specific measures for information retrieval IR systems to evaluate the results of algorithm used: accuracy (precision) and the recall, both comparing the results with relevance. To express these concepts will be used contingency table [14] as follows:

Table 1. Contingency table of correctly classified reviews

	Relevant	Irrelevant
Detected opinions	true positive (tp)	False positive (fp)
Undetected opinions	False negative (fn)	True negative (tn)

Precision is the ratio of the correctly classified extracted opinions and all extracted opinions, the percentage of correctly classified opinions from classified ones:

$$Precision = \frac{tp}{\mathrm{tp} + \mathrm{fp}}$$

Recall expresses the ratio of correctly classified extracted opinions and classified opinions in data source, the percent of correctly classified opinions from all opinions in a class:

$$Recall = \frac{tp}{\mathrm{tp} + \mathrm{fn}}$$

Another evaluation measure for algorithm may be accuracy, expressing the percentage of correct made classifications, and F-measure, a weighted harmonic mean of precision and recall:

$$Accuracy = \frac{tp + tn}{\mathrm{tp} + \mathrm{tn} + \mathrm{fp} + \mathrm{fn}} \quad , \quad F = \frac{2 * Precizion * Recall}{\mathrm{Precision} + \mathrm{Recall}}$$

$$Accuracy = \frac{tp + tn}{\mathrm{tp} + \mathrm{tn} + \mathrm{fp} + \mathrm{fn}} \quad , \quad F = \frac{2 * Precizion * Recall}{\mathrm{Precision} + \mathrm{Recall}}$$

We calculate accuracy of classifier, the recall and precision for the two classes, training the algorithm on 5000 sentences for each class of pre-classification test examples and applying it on the rest of the remaining examples:

```
<?php
$op = new naiveBayes();
$op-> antreneazaNB ('opinion/rt-polaritydata/rt-polarity.neg', 'neg', 5000);
$op->antreneazaNB ('opinion/rt-polaritydata/rt-polarity.pos', 'pos', 5000);
$i = 0; $tn = 0; $fn = 0;  $tp = 0; $fp = 0;
$fh = fopen('opinion/rt-polarity.neg', 'r');
while($line = fgets($fh)) {
        if($i++ > 5001) {
                if($op->clasifica($line) == 'neg') {
                        $tn++;
                } else {
                        $fp++;
                }
        }
}
$fh = fopen('opinion/rt-polarity.pos', 'r');
while($line = fgets($fh)) {
        if($i++ > 5001) {
                if($op->classifica($line) == 'pos') {
                        $tp++;
                } else {
                        $fn++;
                }
        }
}
echo "<br />Precizie: " . ($tp / ($tp+$fn));
echo "<br />Recall : " . ($tp / ($tp+$fn));
echo "<br />Accuracy: " . (($tp+$tn) / ($tp+$tn+$fp+$fn));
?>
```

Analyzing the algorithm efficiency by the above parameters we achieved a **0.814332247557** value of correct classification of opinions. This efficiency is close to those obtained by Opinion mining researches conducted in recent years. We aim to find ways to improve efficiency. A solution to improve the quality of the algorithm is to eliminate insignificant words for classification. Algorithm originally classified words without lexical content, so that besides nouns, verbs, adverbs and adjectives, are considered articles, prepositions and pronouns without semantic value.

We will eliminate these words (called stop words in English) that can induce noise in the classification. For this we have built an array of four vectors *$exclude* corresponding to those prepositions, conjunctions, articles and pronouns (e.g. articles - the, the, year, conjunctions - and, now, so, still, only, pronouns - who, whom, which, that, this, me, you, ours, prepositions - about, above, across, after, at, around, with, up). After this step we observe that the algorithm efficiency has improved a little, giving a value of: **0.81699346405229**.

Algorithm considers that there is no relationship between words in a sentence, but in reality they are interrelated. So there are certain words that occur frequently together in one of two classes, and also the juxtaposition of very uncommon words to express an opinion. To take into account this aspect we introduced in the algorithm the method *ngram()* that tokens a sentence into groups of n words (N-grams).

```
private function ngram ($tokens, $n){
   $ngramtokens=$tokens;
   $len= count($tokens);
   for ($l=2;$l<=$n;$l++){
     $i=0;
     $j=$i+$l;
     while ($j<=$len){
       $token='';
       for ($k=0;$k<$l;$k++) {
          $token.=$tokens[$i+$k]." ";
       }
       $ngramtokens[]=trim($token);
       $i++;
       $j++;
     }
   }
   return $ngramtokens;
}
```

It takes the words of a sentence and returns an array containing the initial words and groups of one to n words which may be obtained from the original sentence. The value of n is given as input parameter. Usually, we observe that the algorithm efficiency increases for values up to a maximum of three groups of words. In this application we tested the introduction in classification of groups of words for the n = 2 and n = 3. Results are presented in the table below:

Table 2. Indicators of algorithm efficiency

	Initial Algorithm, groups of n=1 words	**Initial Algorithm, groups of n=1 words, eliminate stop words**	**Algorithm for groups of n=2 words, eliminate stop words**	**Algorithm for groups of n=3 words, eliminate stop words**
Precision	0.814332247557	0.81699346405229	**0.82084690553746**	0.79723502304147
Recall	0.75987841945289	0.75987841945289	0.76595744680851	0.5258358662614
Acurracy	0.79331306990881	0.79483282674772	**0.79939209726444**	0.69604863221884
F-measure	0.78616352201258	0.78740157480315	0.79245283018868	0.63369963369963
Execution time (s)	**1.153508902**	3.249027014	8.068459988	14.778712988

The following chart presents the influence of data training volume on the accuracy of classifications in the method used. We detect a critical number of training data, from which point the increase number of the initial training set, will produce very little influence on precision of the algorithm.

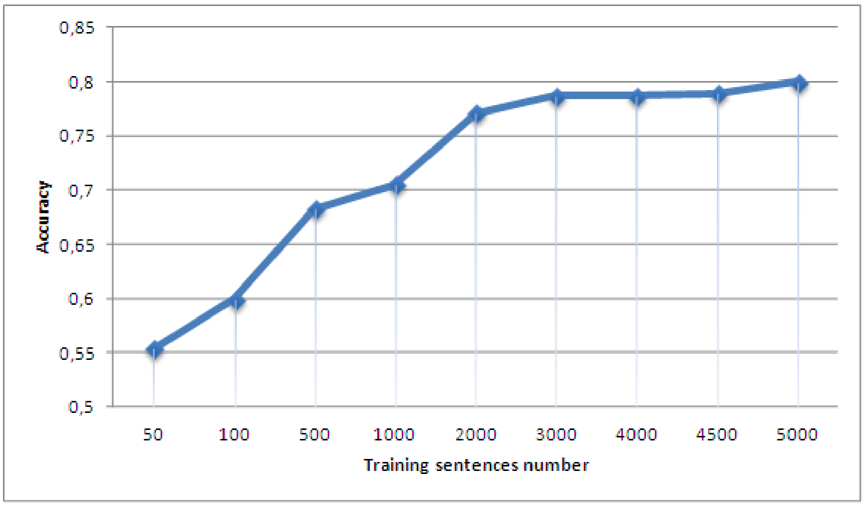


Fig. 5. Accuracy influenced by volume of training set

8 Conclusions

The expression of opinions of users in specialized sites for evaluation of products and services, and also on social networking platforms, has become one of the main ways of communication, due to spectacular development of web environment in recent years. The large amount of information on these platforms make them viable for use as data sources, in applications based on opinion mining and sentiment analysis. This paper presents a method of sentiment analysis, on the review made by users to movies. Classification of reviews in both positive and negative classes is done based on a naive Bayes algorithm. As training data we used a collection (pre-classified in positive and negative) of sentences taken from the movie reviews. To improve classification we removed insignificant words and introduced in classification groups of words (n-grams). For n = 2 groups we achieved a substantial improvement in classification.

As an extension of the research presented in this paper we want to improve the algorithm, enriching the training set of examples, on the way, with examples classified as strong positive or negative, by an established score of classification. We try to determine, in a review, those sentences which do not express opinions, or determine opinions about the film or the film actors and identify opinions addressed strictly on these items. We try to highlight the main aspects on which opinions are expressed and to extract opinions based on aspects identification.

Acknowledgements

This article is a result of the project POSDRU/88/1.5./S/55287 „Doctoral Programme in Economics at European Knowledge Standards (DOESEC)". This project is co-funded by the European Social Fund through The Sectorial Operational Programme for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies in partnership with West University of Timisoara

References

[1] S. Wang and H. Wang, "A Knowledge Management Approach to Data Mining," Industrial Management and Data Systems, vol. Vol. 108, No. 5, pp. 622-634, 2008.

[2] I. Smeureanu, A. Diosteanu, C. Delcea, L.A. Cotfas. “Busines Ontology for Evaluating Corporate Social Responsibility,” (Ontologii de afaceri pentru evaluarea responsabilității, corporațiilor), *Amfiteatru Economic*, vol. 29, 2011, pp. 28-42

[3] I. Smeureanu, M. Zurini, “Spam Filtering for Optimization in Internet Promotions using Bayesian Analysis,” *Journal of Applied Quantitative*

Methods, Vol. 5, Issue.2, pp. 198-211, 2010.

[4] C. Bucur, T. Bogdan “Solutions for Working with Large Data Volumes in Web Applications“, *Proceedings of the 10th International Conference on Informatics in Economy - IE 2011 „Education, Research & Business Technologies”*, **5-7** Mai 2011, Printing House ASE, Bucharest, 2011.

[5] B. Liu, *Web Data Mining - Exploring Hyperlinks, Contents and Usage Data*, Secound ed.: Springer, 2011.

[6] F.J. A. P. Mattosinho, *Mining Product Opinions and Reviews on the Web*, Technische Universitat Dresden, Ed.: Department of Computer Science, 2010.

[7] B. L. Minqing Hu, "Mining and Summarizing Customer Reviews," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (KDD-2004, full paper), Seattle, USA, 2004.

[8] M. Leenhardt. (2012, Mar.) http://blog.leenhardt.name/post/2011/05/04/Opinion-Mining-Sentiment-Analysis%2C-or-what-sets-up-a-hot-topic. [Online]. http://blog.leenhardt.name/post/2011/05/04/Opinion-Mining-Sentiment-Analysis%2C-or-what-sets-up-a-hot-topic

[9] L. L. B. Pang, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no 1-2, pp. 1-135, 2008.

[10] A. Moran. (2012, Mar.) Sentiment Analysis: How does sentiment analysis work? [Online]. http://www.quora.com/Sentiment-Analysis/How-does-sentiment-analysis-work#ans606524

[11] M Caraciolo. (2012, Mar.) Working on sentiment analysis on Twitter with Portuguese language. [Online]. http://aimotion.blogspot.com/2010/07/working-on-sentiment-analysis-on.html

[12] L. L. B. Pang, "*Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales.*," in Proceedings of ACL, 2005, pp. 115–124.

[13] (2012, Mar.) Wikipedia. [Online]. http://en.wikipedia.org

[14] P. Raghavan, H. S. Christopher and D. Manning, *Introduction to Information Retrieval*. USA, New York: Cambridge University Press, 2008.

Ion SMEUREANU has graduated the Faculty of Planning and Economic Cybernetics in 1980, as promotion leader. He holds a PhD diploma in "Economic Cybernetics" from 1992 and has a remarkable didactic activity since 1984 when he joined the staff of Bucharest Academy of Economic Studies. Currently, he is a full Professor of Economic Informatics within the Department of Economic Informatics and the dean of the Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies. He is the author of more than 16 books and an impressive number of articles. He was also project director or member in many important research projects. He was awarded the Nicolae Georgescu-Roegen diploma, the award for the entire research activity offered by the Romanian Statistics Society in 2007 and many others.

Cristian BUCUR is a doctoral student at the Institute for Doctoral Studies in ASE Bucharest, in the field of Economic Informatics, from 2009. He has a MSc. diploma in Management of Economic Systems since 2009, a MSc in Computer Science since 2007 at UPG Ploiesti and BSc. in Mathematics - Computer Science at Letters and Sciences Faculty in UPG Ploiesti. He is currently assistant professor in the Faculty of Economics at UPG Ploieşti. Domains of interest: Web technologies, webminig, semantic web, business intelligence.