Taylor & Francis
Taylor & Francis Group

Journal of Organizational Computing and Electronic Commerce

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/hoce20

A hybrid approach for generating reputation based on opinions fusion and sentiment analysis

Abdessamad Benlahbib & El Habib Nfaoui

To cite this article: Abdessamad Benlahbib & El Habib Nfaoui (2020) A hybrid approach for generating reputation based on opinions fusion and sentiment analysis, Journal of Organizational Computing and Electronic Commerce, 30:1, 9-27, DOI: 10.1080/10919392.2019.1654350

To link to this article: https://doi.org/10.1080/10919392.2019.1654350

Published online: 29 Aug 2019.

JOURNAL OF ORGANIZATIONAL COMPUTING AND ELECTRONIC COMMERCE
2020, VOL. 30, NO. 1, 9–27
https://doi.org/10.1080/10919392.2019.1654350

A hybrid approach for generating reputation based on opinions fusion and sentiment analysis

Abdessamad Benlahbib and El Habib Nfaoui

LIIAN Laboratory, Faculty of Sciences Dhar EL Mehraz (F.S.D.M), Sidi Mohamed Ben Abdellah University Fez, Fes-Atlas, Morocco

ABSTRACT
Amazon, eBay, IMDb as well as several websites provide a convenient platform where users share their opinions on any entities without hindrance. Though those opinions are too many to be examined one by one, this is why a general reputation value will help people make a decision toward a target entity (purchase, download, rent …). This fact makes reputation generation task very challenging because an inaccurate reputation system will directly damage the credibility and popularity of the target entity. This paper aims to improve a recent work that handles the task of generating reputation based on fuzing and mining opinions expressed in natural languages and user feedback ratings. Therefore, we have proposed a hybrid approach that, (i) separates reviews into positive and negative based on their sentiment polarity by applying the two classifiers Naïve Bayes and Linear Support Vector Machine (LSVM), (ii) groups positive and negative reviews into principal opinion sets based on their semantic relations, (iii) calculates a custom reputation value ***separately*** for positive and negative groups by considering some statistics of principal opinion sets and finally (iv) computes the final reputation value using Weighted Arithmetic Mean. Experimental results show a significant improvement with respect to recent work.

KEYWORDS
Opinion fusion; opinion mining; reputation generation; sentiment analysis and machine learning

1 Introduction

Over the twenty-first century, the web has been expanding at an incredible rate. Thanks to social media and web communities, people express freely their opinions on any entities they desire, with no chains or restrictions. Those opinions represent a valuable source of information which can contribute to track public perspectives and generate reputation about a target entity for the reason that they carry the preferences and attitudes of humans. We can find many research papers and books for reputation systems, Farmer and Glass (2010), Yan (2013) and Jøsang, Ismail, and Boyd (2007). However, there is particularly a lack of studies that handle the task of generating reputation based on mining opinions expressed in natural languages. Hence, the importance of Yan, Jing, and Pedrycz (2017) pioneering work that presents an approach to generate reputation by fuzing and mining opinions expressed in natural languages.

In this paper, we have an interest in improving Yan, Jing, and Pedrycz (2017) approach by generating an accurate reputation value toward a target entity. The fact that the majority of opinions expressed in natural languages holds a positive or negative sentiment (point of view) toward a target entity gives us the idea of using a classification step in order to separate positive and negative opinions before grouping them into different sets based on semantic relations, then computing

CONTACT Abdessamad Benlahbib abdessamad.benlahbib@usmba.ac.ma LIIAN Laboratory, Faculty of Sciences Dhar EL Mehraz (F.S.D.M), Sidi Mohamed Ben Abdellah University Fez, B.P. 1796, Fes-Atlas 30003, Morocco
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/hoce.

reputation toward the target entity using Weighted Arithmetic Mean. Indeed, the main steps of this contribution can be summarized as follows:

(1) We apply a classification step based on the two classifiers Naïve Bayes and Linear Support Vector Machine (LSVM) in order to separate positive and negative opinions.
(2) We group **separately** positive and negative opinions into different sets based on semantic relations.
(3) We calculate a custom reputation value **separately** for positive and negative groups by considering some statistics of principal opinion sets.
(4) We compute the final reputation value toward the target entity using Weighted Arithmetic Mean.

The paper is organized as follows. Section 2 gives a brief review of related work for reputation generation and opinion mining tasks. In Section 3, we develop our method. Section 4, shows experimental results, comparison, and discussion. Finally, we conclude this work by summarizing the main contribution and suggesting some interesting research directions.

2. Related work

In this section, we will present the literature pertaining to the reputation generation and sentiment analysis since our proposed approach generates reputation using Natural Language Processing (NLP) techniques and opinion mining methods.

2.1. *Reputation generation based on NLP*

Yan (2013) defines reputation as: *"a measure that is derived from direct or indirect knowledge on earlier interactions of entities and is used to assess the level of trust an entity puts into another entity."* . The simplest form of computing reputation scores is simply to sum the number of positive ratings and negative ratings separately and to keep a total score as the positive score minus the negative score. This is the principle used in eBay's reputation forum which is described in Paul and Richard (2002). Schneider et al. (2000) have proposed a more advanced scheme to compute the reputation score as the average of all ratings, and this principle is used in the reputation systems of numerous commercial web sites, such as Epinions and Amazon. Advanced models in this category compute a weighted average of all the ratings, where the rating weight can be determined by factors such as rater trustworthiness/reputation, age of the rating, distance between rating and current score, etc. However, none of these works generate reputation based on mining opinions expressed in natural languages.

Ya-Han, Chen, and Chou (2017) illustrated a novel multi-text summarization technique for identifying the top-k most informative sentences of hotel reviews. Poria, Cambria, and Gelbukh (2016) presented the first deep learning approach to aspect extraction in opinion mining. They used a seven-layer deep convolutional neural network to tag each word in opinionated sentences as either aspect or non-aspect word. In Barnaghi, Ghaffari, and Breslin (2016), the authors provided a positive or negative sentiment on Twitter posts using a well-known machine learning method for text categorization. In addition, they used manually labeled (positive/negative) tweets to build a trained method to accomplish the task of looking for a correlation between twitter sentiment and events that have occurred. He, Zhang, and Akula (2018) conducted a case study to analyze and compare the written online consumer reviews of three large US retailers including Sears, Home Depot, and Best Buy for a same home appliance product. Afterward, they combined online consumer reviews from these large retailers and conduct overall text analytics and sentiment analysis. The overall results are further compared with the results from individual retailers. The authors concluded that relying on a single data source to make a purchase decision is not a wise idea because the sentiment of the online consumer reviews could vary substantially. Amblee, Ullah, and Kim (2017) conducted an experiment using an eye-tracking machine to measure the impact of online editorial and customer reviews on decision confidence. They found that when present, both editorial reviews and customer reviews separately increase decision confidence considerably. Yan, Jing, and Pedrycz (2017) have proposed a novel reputation generation approach based on fuzing and mining opinions expressed in natural languages in which they claimed that *"no work has explored the opinions expressed in natural languages, opinion voting, opinion citation and user feedback ratings in a comprehensive way for reputation generation"*. In their approach, Yan, Jing, and Pedrycz (2017) have filtered opinions to eliminate unrelated ones and then grouped into a number of fuzed principal opinion sets that contain opinions with a similar or the same attitude or preference. By aggregating the ratings attached to the fuzed opinions, they normalize the reputation of an entity. In order to gain reputation, they have applied data fusion and mining techniques along with semantic analysis to process the opinions described in natural language. Latent Semantic Analysis (LSA) model and cosine similarity have been used to compute the similarity between opinions before grouping them into different sets.

In their paper, Yan, Jing, and Pedrycz (2017) said: *"The opinions in each set hold a similar or same perspective"*. However, relying only on Latent Semantic Analysis model is not enough for extracting opinions with a similar perspective.

In order to justify our claim, we study the three reviews in Table 1.

Table 1 represents the results after applying the LSA model, then computing document similarity using LSA components and cosine similarity metric. The similarity value between the first and the second review is very high, on the contrary, we notice that the similarity between the first and the third review is low, so based on the opinion fusion and grouping algorithm proposed by Yan, Jing, and Pedrycz (2017), and by choosing an opinion fusion threshold $t_0 \leq 0.23$, we will get two opinion sets, the first set contains the first and the second reviews and the second set contains the third review. But it is obvious that the first and the third reviews stand for a positive point of view and the second review stands for a negative point of view, so, we believe that adding a classification step is an effective idea in order to classify positive and negative opinions before grouping them. As a matter of fact, applying a classification step before the grouping phase will overcome the weakness of LSA model in extracting opinions with the same perspective and will guarantee that positive opinions will be grouped together and so on.

2.2. *Sentiment analysis*

Opinion mining or sentiment analysis is the field of study that aims to extract sentiment (polarity) from opinions expressed in natural languages.

In Liu (2010), Bing Liu represents an opinion as 5-tuple $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ where e_i: i-th entity target, a_{ij}: j-th aspect of e_i, h_k: k-th opinion holder, t_l: time when the opinion was posted and s_{ijkl}: polarity of the opinion (positive or negative) toward the attribute a_{ij} of entity e_i from opinion holder h_k at time t_l. For example, in the review *"The battery capacity of my laptop is short."*, battery is an aspect of the entity laptop and has a short capacity which stands for negative sentiment polarity. For aspect level sentiment analysis, the first three components must be determined. However, only the third component matter in the document level.

Balazs and Velásquez (2016) published a survey on opinion mining and information fusion where they defined opinion mining as *"a sub-field of text mining in which the main task is to extract opinion from content generated by Web users"*. Furthermore, Cambria (2016) discussed merits and limitations of various sentiment analysis methods such as knowledge-based, statistical, and hybrid.

Plenty of works used machine learning or lexicon-based techniques to handle sentiment analysis task. Turney (2002) presented a simple unsupervised learning algorithm to classify reviews as recommended or not recommended. It determines the polarities by averaging Semantic Orientation (SO) of the phrases in the

Table 1. Latent semantic analysis model weakness in extracting opinions with a similar or same perspective.

	The movie was good	The movie was not good at all	Amazing acting
The movie was good	1.000000	0.996730	0.221941
The movie was not good at all	0.996730	1.000000	0.142422
Amazing acting	0.221941	0.142422	1.000000

review that contain adjectives or adverbs. The SO of a phrase is calculated as the mutual information (PMI-IR) (Turney (2001)) between the given phrase and the word "excellent" minus the mutual information between the given phrase and the word "poor". A review is classified as recommended if the average SO of its phrases is positive. In Hu and Liu (2004), Hu and Liu created a list of opinion words using WordNet (Miller (1995)) to predict the orientation of opinion sentences by determining the prevalent word orientation.

Shawe-Taylor and Sun (2011) presented Naïve Bayes classifier, Support Vector Machine (SVM) and maximum entropy classifier as the most frequently used models to determine text polarity. Pang, Lee, and Vaithyanathan (2002) were the first to implement these three methods in order to classify movie reviews as positive or negative and found that the use of unigrams as features in classification gave good results. Jing, Yu, and Lin (2015) applied Naïve Bayes classifier to 3046 review comments related to 58 business-to-team (B2T) websites to investigate the critical factors to the survival of online B2T businesses in an overcrowded buyer-side market. Nguyen et al. (2012) present a strategy of building statistical models from the social media dynamics to predict collective sentiment dynamics by using two different machine learning approaches, *"Dynamic Language Model"* (DynamicLM) (Pang and Lee (2004)) and a *"Constrained Symmetric Nonnegative Matrix Factorization"* (CSNMF) (Peng and Park (2011)). Kennedy and Inkpen (2006) present two methods based on SVMs and lexicons to determine the polarity of a movie review. The authors incorporate three types of valence shifters: *"negations, intensifiers, and diminishers"*. Mohan, Janani, and Karthiga (2017) have recently published a survey on sentiment analysis on social network data, they used different algorithms like Naïve Bayes, Support Vector Machine and others in order to compare them on different domains (business, Education, Entertainment, health, Law, Lifestyle, nature, places, politics, Sport, technology). Liu and Cocea (2017) adopted a fuzzy rule-based system for sentiment analysis, they compared it with commonly used sentiment classifiers (e.g., Decision Trees, Naïve Bayes) and found that their approach outperforms the other algorithms. Sun, Luo, and Chen (2017) presented an excellent review of natural language processing techniques for opinion mining systems. They introduced general NLP techniques which are required for text preprocessing. After, they investigated the approaches of opinion mining for different levels and situations. Then, they introduced comparative opinion mining, opinion summarization and deep learning approaches for opinion mining. Tang, Qin, and Liu (2015) presented an overview of some successful deep learning algorithms that have improved the state of the art in many sentiment analysis tasks involving word embedding learning, sentiment classification, opinion extraction, and sentiment lexicon learning. Wang et al. (2016) proposed an Attention-based Long Short-Term Memory Network for aspect-level sentiment classification. Baktha and Tripathy (2017) analyzed the performance of three RNNs, namely, vanilla RNNs, Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). They evaluated the performance of these networks on the Amazon health product reviews dataset and sentiment analysis benchmark datasets SST-1 and SST-2. Recently, Sun, Huang, and Qiu (2019) utilized BERT (Bidirectional Encoder Representations from Transformers) for aspect-based sentiment analysis via constructing an auxiliary sentence and they achieved new state-of-the-art results on SentiHood and SemEval-2014 Task 4 datasets.

3. Proposed approach

3.1. *System overview*

Figure 1 describes the pipeline of our work.

(1) **Classification phase**. We separate comments into positive and negative sets based on their sentiment polarity and their attached ratings by combining two classifiers: Naïve Bayes and Linear Support Vector Machine (LSVM). They both are trained with movie reviews dataset that contains 1000 positive and 1000 negative processed reviews.

(2) **Fusion and grouping phase**. We apply the opinion fusion and grouping algorithm proposed by Yan, Jing, and Pedrycz (2017) ***separately*** for positive and negative reviews in order

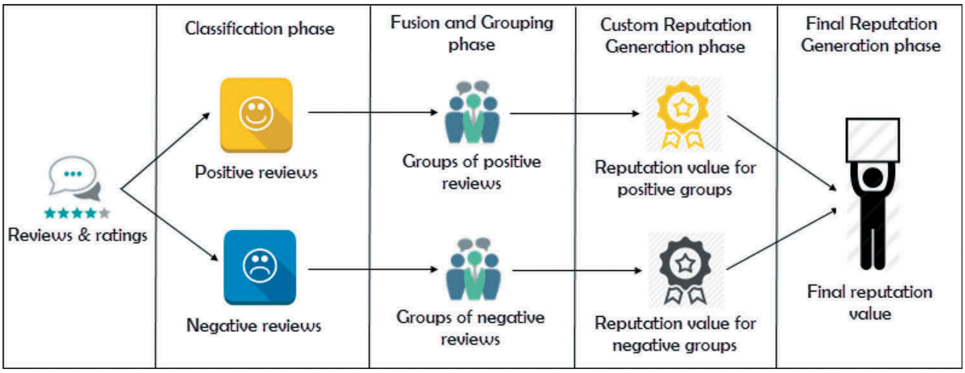

Figure 1. Pipeline of our work.

to classify opinions into principal opinion sets based on their semantic relations (LSA model and cosine similarity metric).

(3) **Custom reputation generation phase**. We calculate a custom reputation value ***separately*** for positive and negative groups by considering some statistics of principal opinion sets.

(4) **Final reputation generation phase**. We compute the final reputation value toward the target entity based on the custom reputation value for positive and negative groups using the Weighted Arithmetic Mean.

3.2. *Classification phase*

We propose to combine Naïve Bayes and LSVM classifiers in order to separate positive and negative reviews. The classification step is organized as follows:

(1) First, we apply a Naïve Bayes classifier to our reviews and we check the polarity of each one of them and their attached ratings (numeric scale from 1 to 10 as an example):
   (a) If the review polarity is predicted as positive and its attached rating is greater than 5, the review will be added to the *"positiveReviews"* set.
   (b) If the review polarity is predicted as negative and its attached rating is less or equal to 5, the review will be added to the *"negativeReviews"* set.
   (c) If the review polarity is predicted as positive and its attached rating is less or equal to 5, the review will be added to the *"remainingReviews"* set.
   (d) If the review polarity is predicted as negative and its attached rating is greater than 5, the review will be added to the *"remainingReviews"* set.

(2) Second, we apply an SVM classifier with a linear kernel to *"remainingReviews"* set and we compare SVM polarity prediction with Naïve Bayes polarity prediction:
   (a) If the review polarity is predicted as positive by SVM and Naïve Bayes classifier, the review will be added to the *"positiveReviews"* set.
   (b) If the review polarity is predicted as negative by SVM and Naïve Bayes classifier, the review will be added to the *"negativeReviews"* set.
   (c) If the review polarity is predicted as positive (respectively, negative) by SVM and negative (respectively, positive) by Naïve Bayes classifier, the review will be added to the *"remaining1Reviews"* set.

(3) Third and finally, reviews contained in *"remaining1Reviews"* set will be separated based on their attached ratings:
   (a) If the review attached rating is greater than 5, the review will be added to the *"positiveReviews"* set.

(b) If the review attached rating is less or equal to 5, the review will be added to the *"negativeReviews"* set.

The motivation behind choosing Naïve Bayes and SVM classifiers to predict reviews polarity is based on the works done by Shawe-Taylor and Sun (2011) and Pang, Lee, and Vaithyanathan (2002). Shawe-Taylor and Sun (2011) presented Naïve Bayes classifier, Support Vector Machine (SVM) and maximum entropy classifier as the most frequently used models to determine text polarity. Furthermore, Pang, Lee, and Vaithyanathan (2002) were the first to implement these three methods in order to classify **movie reviews** as positive or negative and found that the use of unigrams as features in classification gave good results.

We apply Naïve Bayes before SVM because we have found that Naïve Bayes achieves 82% accuracy in predicting sentiment polarity of our collected reviews compared to 77% accuracy for SVM. Table 4 depicts precision, recall, f-score, and accuracy of Naïve Bayes and SVM applied to all 1000 reviews of datasets. Furthermore, Pang, Lee, and Vaithyanathan (2002) have applied Naïve Bayes and SVM to a movie reviews dataset (polarity dataset v2.0)[1] and they have found that when they use unigrams as features, Naïve Bayes achieves higher accuracy compared to SVM in predicting movie reviews polarity. So, we use the accurate classifier: Naïve Bayes as our primary classifier, then, we apply the SVM classifier when there is a mismatch between the Naïve Bayes polarity prediction toward the review and its attached rating.

Algorithm 1 below provides all details about the classification step.

Algorithm 1 Classification phase

Input: V **and** R
Output: positiveReviews **and** negativeReviews
Define
{
V = {v{1}, v{2}, …, v{n}}: the set of ratings attached to each review;
R = {r{1}, r{2}, …, r{n}}: the set of reviews expressed in natural language;
NBPrediction = {nbp{1}, nbp{2}, …, nbp{n}}: the set of polarity of reviews R predicted by Naïve Bayes classifier;
SVMPrediction = {svmp{1}, svmp{2}, …, svmp{p}}: the set of polarity of remainingReviews predicted by SVM classifier;
remainingReviews: the set of indexes for remaining reviews after the first step;
remaining1Reviews: the set of indexes for remaining reviews after the second step;
}
Begin
{ Apply a Naïve Bayes classifier to predict polarity of reviews R and store the result in NBPrediction}
for $i = 0; i \leq n; i++$ **do**
  if nbp{i} is 'positive' **and** v{i} $>$ 5 **then**
    add i to positiveReviews
  else if nbp{i} is 'negative' **and** v{i} $\leq$ 5 **then**
    add i to negativeReviews
  else
    add i to remainingReviews
  end if
end if

[1]www.cs.cornell.edu/people/pabo/movie-review-data/review_polarity.tar.gz.

{Apply an SVM classifier to predict polarity of remainingReviews and store the result in SVMPrediction}
lengthRemainingReviews $\leftarrow$ length of remainingReviews
for $i = 0; i \leq lengthRemainingReviews; i++$ **do**
  if nbp{remainingReviews[i]} is 'positive' **and** svmp{i} is 'positive' **then**
    add remainingReviews[i] to positiveReviews
  else if nbp{remainingReviews[i]} is 'negative' **and** svmp{i} is 'negative' **then**
    add remainingReviews[i] to negativeReviews
  else
    add remainingReviews[i] to remaining1Reviews
  end if
end for
lengthRemaining1Reviews $\leftarrow$ length of remaining1Reviews
for $i = 0; i \leq lengthRemaining1Reviews; i++$ **do**
  if v{remaining1Reviews[i]} $>$ 5 **then**
    add remaining1Reviews[i] to positiveReviews
  else
    add remaining1Reviews[i] to negativeReviews
  end if
end for

3.3. *Fusion and grouping phase*

In order to overcome the weakness of LSA model in extracting opinions with the same perspective, we apply the opinion fusion and grouping algorithm (Yan, Jing, and Pedrycz (2017)) ***separately*** for positive and negative reviews after the classification step, which guarantees that positive opinions will be grouped together and so on. Yan, Jing, and Pedrycz (2017) describe the opinion fusion and grouping algorithm as follows: *"By applying Algorithm 1, we fuse and group opinions into several principal opinion sets. The opinions in each set hold a similar or same perspective"* and *"Once the processing based on Algorithm 1 has been completed, the opinions are grouped into a number of K fuzed principal opinion sets. Meanwhile, we also get the statistics of the principal opinions, i.e., the number of similar opinions in each set, the sum of their ratings and the sum of their similarity".*

3.4. *Reputation generation*

Based on the result of the grouping phase, we propose to compute a custom reputation value ***separately*** for positive and negative groups. We tailor the average similarity $S_k^{polarity}/N_k^{polarity}$ with the average rating value $V_k^{polarity}/N_k^{polarity}$ of principal opinion set k for a specific polarity. We compute reputation separately for positive and negative groups by applying this formula:

$$Rep(A^{polarity}) = \frac{1}{K^{polarity}} \cdot \sum_{k=1}^{K^{polarity}} \frac{V_k^{polarity}.S_k^{polarity}}{N_k^{polarity}.N_k^{polarity}} \tag{1}$$

$K^{polarity}$: The number of opinion sets for a specific polarity (positive or negative opinions).
$N_k^{polarity}$: The number of similar opinions in principal opinion set k for a specific polarity.
$S_k^{polarity}$: The sum of the similarity in principal opinion set k for a specific polarity.
$V_k^{polarity}$: The sum of ratings in principal opinion set k for a specific polarity.

In general, the number of positive opinions and the number of negative opinions are not equal, which implies that $O^{positive}$ and $O^{negative}$ don't contribute equally to the final reputation value. For that, we propose to use the Weighted Arithmetic Mean for computing the final reputation value instead of using the ordinary arithmetic mean:

$$finalRep(A) = \frac{\sum_{polarity}^{[positive,negative]} Rep(A^{polarity}).O^{polarity}}{\sum_{polarity}^{[positive,negative]} O^{polarity}} \tag{2}$$

Which leads us to:

$$finalRep(A) = \frac{Rep(A^{positive}).O^{positive} + Rep(A^{negative}).O^{negative}}{O^{positive} + O^{negative}} \tag{3}$$

$O^{positive}$: The number of positive opinions.

$O^{negative}$: The number of negative opinions.

$A^{positive}$: Groups of positive opinions.

$A^{negative}$: Groups of negative opinions.

$Rep(A^{positive})$: Custom reputation value for groups of positive opinions.

$Rep(A^{negative})$: Custom reputation value for groups of negative opinions.

4. Experimental results and discussion

4.1. *Training phase*

As we had mentioned before, we used Naïve Bayes and Linear Support Vector Machine classifiers, both are trained with a movie reviews dataset (polarity dataset v2.0)[2] that contains 1000 positive and 1000 negative processed reviews. The sentences in the corpus are processed and downcased.

The chosen features were unigrams for the Naïve Bayes classifier and TF-IDF (Term frequency – Inverse document frequency) weighted word frequency features for SVM.

4.2. *Datasets collection and preprocess*

In order to evaluate the proposed algorithm, we need a dataset that contains both reviews and ratings. There is no standard dataset suitable for our evaluation, so, we have used the IMDb[3] (Internet Movie Database) website which represents the world's most popular and authoritative source for movie, TV and celebrity content. Figure 2 represents a sample of dataset without review polarity.

We have created manually 10 datasets for 10 different movies, each one contains 100 reviews [comment + rating + sentiment polarity (manually annotated)] randomly extracted and we made sure that the datasets are representatives based on IMDb users weighted average vote. The statistical information of datasets is shown in Table 2.

You may wonder why collecting 100 reviews for each entity instead of using all the reviews attached to it? Because not every single user who rates the entity (movie) shares its review toward it in the platform (IMDb, Amazon, etc…). And as we know, the approach is based on both reviews and ratings, if we use all the reviews attached to an entity (movie), we can encounter cases where a movie has a high IMDb weighted average vote, but only users holding a negative point of view toward it decide to share their opinions (review + rating) and vice versa, in this case, those opinions are not significant, and, based on them, the reputation value will not be fair toward the target entity. For that reason, at the moment of gathering manually the datasets, we make sure that the rating is representative. Table 3 contains more details about datasets.

[2]www.cs.cornell.edu/people/pabo/movie-review-data/review_polarity.tar.gz.
[3]www.imdb.com.

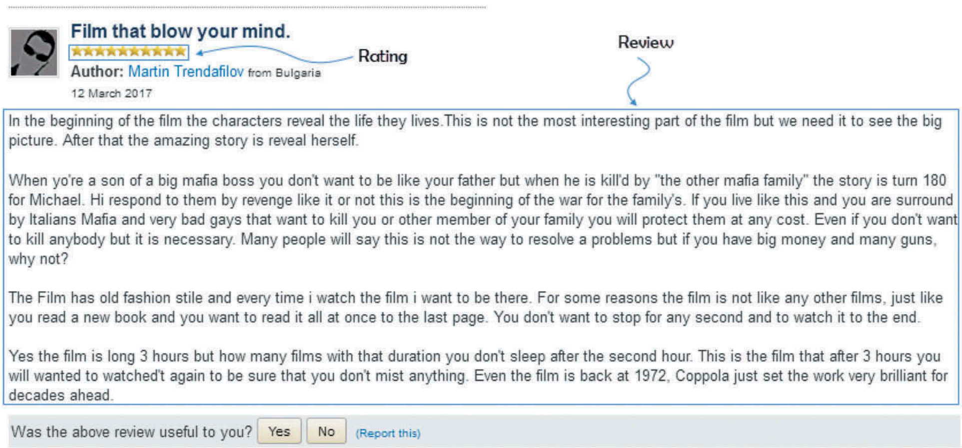

Figure 2. Dataset sample.

Table 2. Statistical information of datasets.

"The total number of movie descriptions"	10
"The total number of reviews and ratings (i.e., opinions)"	1000
"The number of reviews per movie"	100

Table 3. Datasets details.

Movie	IMDb weighted average vote	Dataset average rating
2012 (Dataset 1)	5.8 (301440 users)	5.8
A Beautiful Mind (Dataset 2)	8.2 (672962 users)	8.2
Amadeus (Dataset 3)	8.3 (295560 users)	8.3
Avatar (Dataset 4)	7.8 (946090 users)	7.8
Clash of the Titans (Dataset 5)	5.8 (240456 users)	5.8
Les Miserables (Dataset 6)	7.6 (260252 users)	7.6
Star Wars Episode I (Dataset 7)	6.5 (579916 users)	6.5
The Expendables (Dataset 8)	6.5 (286373 users)	6.5
The Godfather (Dataset 9)	9.2 (1257206 users)	9.2
The Matrix Revolutions (Dataset 10)	6.7 (384761 users)	6.7

According to Table 3, all datasets have an average rating equal to IMDb weighted average vote for each target movie.

After collecting all reviews, we developed a preprocessing algorithm to remove word segmentation and stop words in the reviews and we applied it to our datasets.

In their work, Yan, Jing, and Pedrycz (2017) developed a web spider to collect recent product descriptions, customer reviews and review ratings from Amazon China (https://www.amazon.cn) and Amazon English (https://www.amazon.com). Each rating is associated with a review and subjectively given by customers. The number of reviews per product is between 80 and 100.

4.3. *Sentiment classification*

After training our classifiers, we separate reviews into positive and negative based on their sentiment polarity by applying Algorithm 1. We need to evaluate the effectiveness of Algorithm 1 in classifying reviews. For that, we compared it with Naïve Bayes and SVM classifiers. Figure 3 shows the accuracy of the proposed classification step compared with Naïve Bayes and SVM for each dataset.

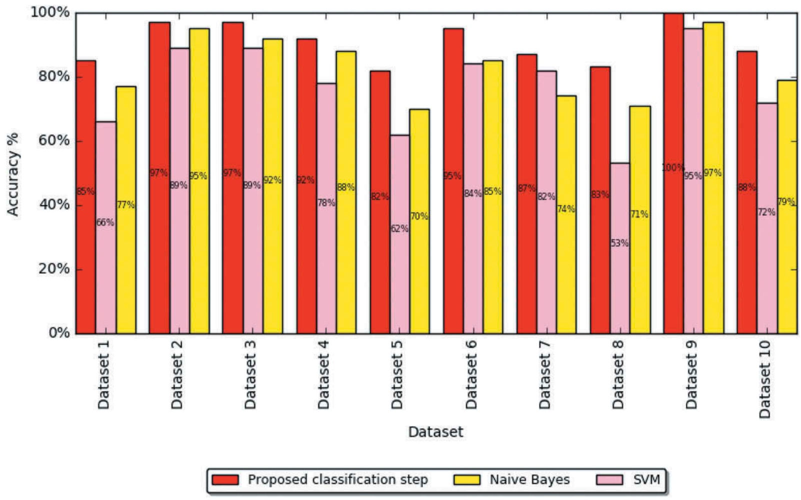

Figure 3. Classification step accuracy of Algorithm 1, Naïve Bayes and SVM.

We can see from Figure 3 that the combination of Naïve Bayes and SVM classifiers (Algorithm 1) gives the highest sentiment classification accuracy for all datasets compared with the separate use of each classifier.

Table 4 depicts precision, recall, f-score, and accuracy of Naïve Bayes, SVM and Algorithm 1 applied to all 1000 reviews of datasets.

Table 4. Precision, recall, F-score, and accuracy of Naïve Bayes, SVM, and Algorithm 1 for all reviews.

Approach	Polarity	Precision	Recall	F-score	Accuracy
Naïve Bayes	Positive	0.88	0.91	0.89	0.82
	Negative	0.54	0.46	0.5	
SVM	Positive	0.94	0.76	0.84	0.77
	Negative	0.43	0.79	0.56	
Algorithm 1	Positive	**0.95**	**0.92**	**0.94**	**0.90**
	Negative	**0.71**	**0.81**	**0.76**	

We can see from Table 4 that our proposed classification method (Algorithm 1) outperforms Naïve Bayes and Linear Support Vector Machine (LSVM) in terms of precision, recall, f-score, and accuracy for both positive and negative reviews.

4.4. *Opinion fusion and grouping*

During this phase, the reviews can be grouped into a number of opinion sets after classifying them into positive and negative based on their sentiment polarity. At the end of this process, we acquire some statistics such as the sum of similarity $S_k^{polarity}$, the sum of ratings $V_k^{polarity}$ and the number of similar opinions $N_k^{polarity}$ for each set. Table 5 provides example results of the grouping phase based on the 100 reviews of a movie in datasets.

We denote:

Set: the principal review set by fuzing and grouping reviews.

Sim: the sum of the similarity in a principal opinion set.

Rat: the sum of ratings in a principal opinion set.

Table 5. Opinion fusion and grouping example results.

"Positive Reviews"				"Negative Reviews"			
"Set"	"Sim"	"Rat"	"Num"	"Set"	"Sim"	"Rat"	"Num"
R_1	66.86	569	67	R_1	3.99	16	4
R_2	3.99	33	4	R_2	1	3	1
R_3	13.98	113	14				
R_4	7.99	69	8				
R_5	1	8	1				
R_6	1	9	1				

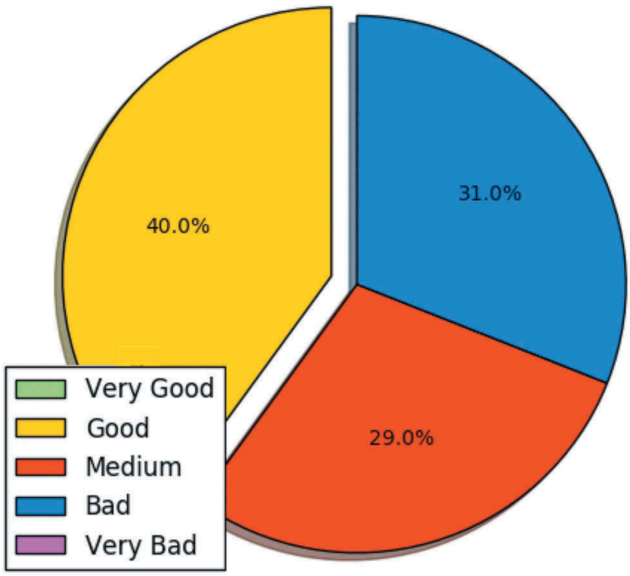

Figure 4. Principal opinions visualization.

Num: the number of similar reviews in a principal opinion set.

From Table 5, we can see that the movie reviews are grouped into several sets R_i after the fuzing and grouping phase. Also, each set contains one review at least.

In order to show the details of the grouping phase results, we provide an opinion visualization to show the top principal opinions and their popularity.

As illustrated in Figure 4, 40% of the total of 100 opinions show that the users are satisfied with the movie, 29% opinions indicate that the movie is *"OK"* and the remaining 31% opinions dissatisfy with the target movie.

4.5. Reputation generation

In their work, Yan, Jing, and Pedrycz (2017) have recruited 10 volunteers to manually rate 1000 products after reading its description and reviews. They averaged their ratings on each product and then computed deviations between the manually generated reputation values and reputation values generated by their approach. However, we believe that even if they have recruited 1000000 volunteers, there is no guarantee that their judgment is right toward the target products, since they may not have enough knowledge or experience to give proper judgment toward it. For that, we have chosen the IMDb weighted average vote as ground truth for the reason that IMDb is a platform where both regular movies fans and expert reviewers rate and share their opinions. We can see in Table 3 that more than 1257206 people gave a rating toward the Godfather 1 movie which is an enormous amount. Besides, the majority of those people share their ratings after watching the target movie, which mean that they have knowledge and opinion toward it, so, in order to evaluate the effectiveness of our method in generating reputation, we compare the final reputation value generated by our method and Yan, Jing, and Pedrycz (2017) with IMDb users weighted average vote applied by IMDb to represent a rating toward the target movie, which is a number ranging from 1 to 10 as shown in Figure 5.

We compare our approach with Yan, Jing, and Pedrycz (2017) by applying them to the datasets that we presented before. We varied the value of the opinion fusion threshold t_0 from 0.05 to 0.95 with a step of 0.05, and we have used the evaluation measure "*Absolute Error*" where:

$$AER_{t_0}^{m_i} = |IMDbWAV^{m_i} - Rep_{t_0}^{m_i}| \tag{4}$$

We denote:

$AER_{t_0}^{m_i}$: Reputation Absolute Error for a specific t_0 value toward the target movie m_i.

$IMDbWAV^{m_i}$: IMDb weighted average vote toward the target movie m_i.

$Rep_{t_0}^{m_i}$: Reputation value computed by our method or Yan, Jing, and Pedrycz (2017) for a specific t_0 value toward the target movie m_i.

Figures 6–10 represent the comparison results between the two methods.

In Figure 6: dataset 1, Figure 7: dataset 3 and 4, Figure 8: dataset 5 and 6 and Figure 10: dataset 9, our method outperforms Yan, Jing, and Pedrycz (2017) method for all values of t_0 varied from 0.05 to 0.95

Figure 5. IMDb users weighted average vote for The Godfather I movie.

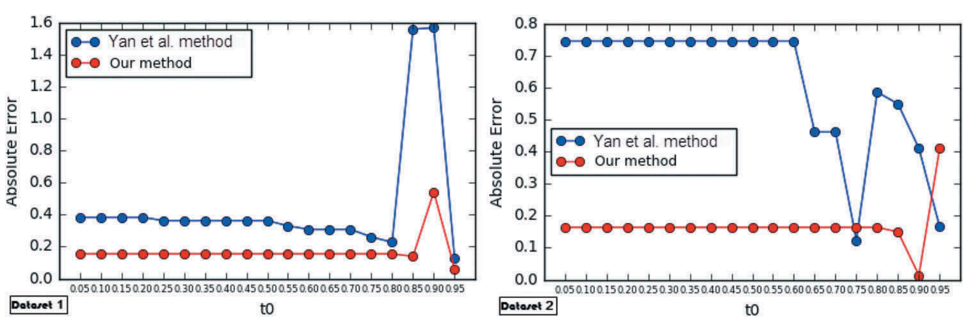

Figure 6. Comparison between Yan, Jing, and Pedrycz (2017) method and our contribution for dataset 1 and dataset 2.

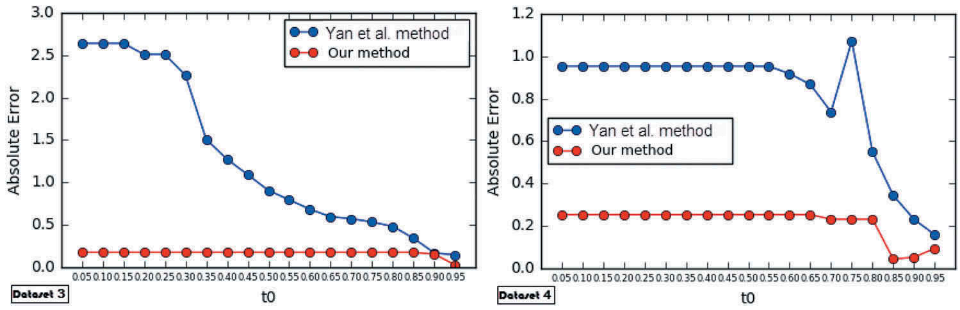

Figure 7. Comparison between Yan, Jing, and Pedrycz (2017) method and our contribution for dataset 3 and dataset 4.

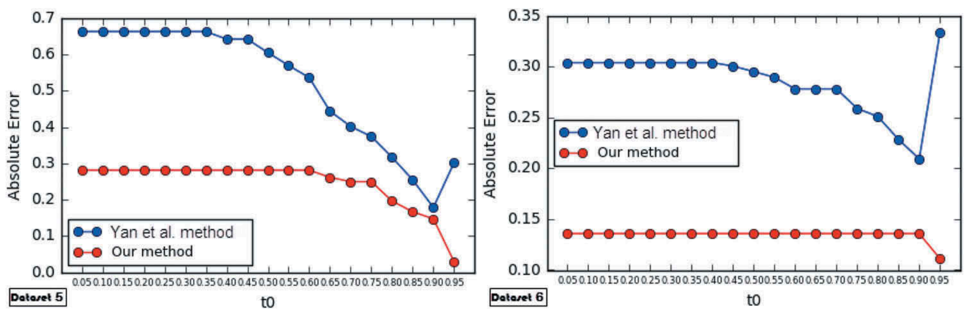

Figure 8. Comparison between Yan, Jing, and Pedrycz (2017) method and our contribution for dataset 5 and dataset 6.

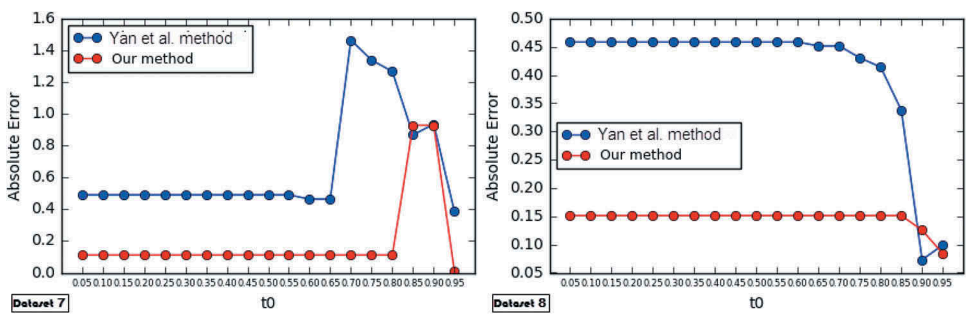

Figure 9. Comparison between Yan, Jing, and Pedrycz (2017) method and our contribution for dataset 7 and dataset 8.

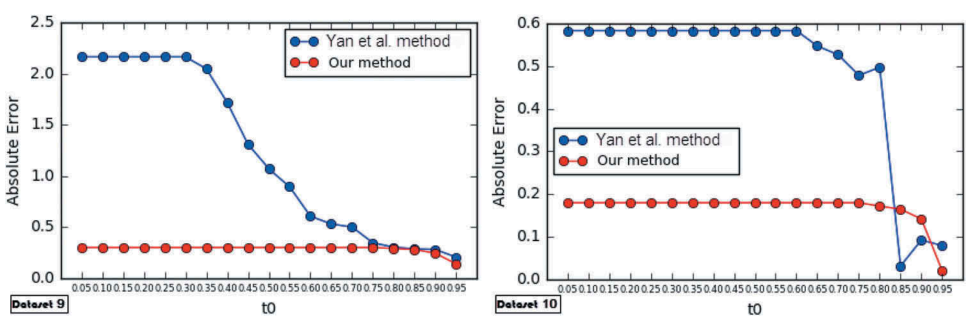

Figure 10. Comparison between Yan, Jing, and Pedrycz (2017) method and our contribution for dataset 9 and dataset 10.

with a step of 0.05. More than that, we can see that for all datasets, our method works better than Yan, Jing, and Pedrycz (2017) when $0.05 \leq t_0 \leq 0.70$, even in the worst cases of our method, Figure 6: dataset 1 when $t_0 = 0.90$, dataset 2 when $t_0 = 0.95$ and Figure 9: dataset 7 when $t_0 = \{0.85, 0.90\}$, the Absolute Error (4) doesn't exceed 0.97.

Table 6 summarizes the results given by the two methods.

From Table 6, we can see that our method provides the nearest reputation value to the ground truth (IMDb weighted average vote) for all datasets, also, the $MAER^{m_i}$ (5) of our method doesn't exceed 0.29, but in Yan, Jing, and Pedrycz (2017) method, the $MAER^{m_i}$ reaches 1.27 in the third dataset and 1.21 in the ninth dataset. Moreover, the ninth dataset contains reviews of The Godfather I movie which is considered as one of the best movies of all times. This high error value could influence the popularity of the movie, which makes reputation generation a very delicate task.

We compute the $MAER^{m_i}$ by applying this formula:

$$MAER^{m_i} = \frac{1}{T} \cdot \sum_{\substack{min \leq t_0 \leq max \\ t_0 \text{ multiple of step}}} AER_{t_0}^{m_i} \tag{5}$$

We denote:

$MAER^{m_i}$: Reputation Mean Absolute Error toward the target movie m_i for all t_0 values.

$AER_{t_0}^{m_i}$: Reputation Absolute Error for a specific t_0 value toward the target movie m_i.

T: Total number of t_0 values, in our case, it varies from 0.05 to 0.95 with a step of 0.05, which means $t_0 = \{0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$ and $T = 19$.

4.6. *Opinion fusion threshold* t_0

After comparing the two methods and showing the effectiveness of ours, we can see that different values of the opinion fusion threshold t_0 could lead to different grouping results which cause different custom reputation values and different final reputation values. Therefore, choosing a suitable t_0 value is very important. For that, we conducted experiments to determine the value of t_0 that provides the best results. We set the value of the opinion fusion threshold t_0 from 0.05 to 0.95 with a step of 0.05. Figure 11 shows the Absolute Error (AE) (4) of the final reputation value computed by our method.

From Figure 11, we observed that for all datasets, the Absolute Error (4) value is stable when $0.05 \leq t_0 \leq 0.6$, and then, it begins to vary either by increasing or decreasing. But, it is still hard to determine the suitable t_0 value for generating an accurate reputation toward the target movie. That's why we conducted more experiments, we computed the $MAER_{t_0}$ (6) between IMDb weighted

Table 6. Experiment results in summarization.

	IMDb weighted average vote	Average reputation (Yan, Jing, and Pedrycz (2017) method)	$MAER^{m_i}$ (Yan, Jing, and Pedrycz (2017) method)	Average reputation (our method)	$MAER^{m_i}$ (our method)
Dataset 1	5.80	5.68401533142	0.458584523451	5.69411656318	**0.169344478854**
Dataset 2	8.20	7.58329159051	0.616708409489	8.033252139	**0.168193964067**
Dataset 3	8.30	7.0236673294	1.2763326706	8.14088245291	**0.161469513573**
Dataset 4	7.80	6.99168965196	0.808310348038	7.60117164592	**0.218177388535**
Dataset 5	5.80	5.27820594837	0.521794051633	5.55387542084	**0.246124579156**
Dataset 6	7.60	7.31385618805	0.286143811945	7.46540816442	**0.134591835581**
Dataset 7	6.50	6.26579054984	0.662862169432	6.30898667249	**0.191809871998**
Dataset 8	6.50	6.09915992375	0.408525260606	6.35358792992	**0.146412070077**
Dataset 9	9.20	7.98707747356	1.21292252644	8.91509031525	**0.284909684753**
Dataset 10	6.70	6.26880591164	0.486749395551	6.53429997883	**0.167701915098**

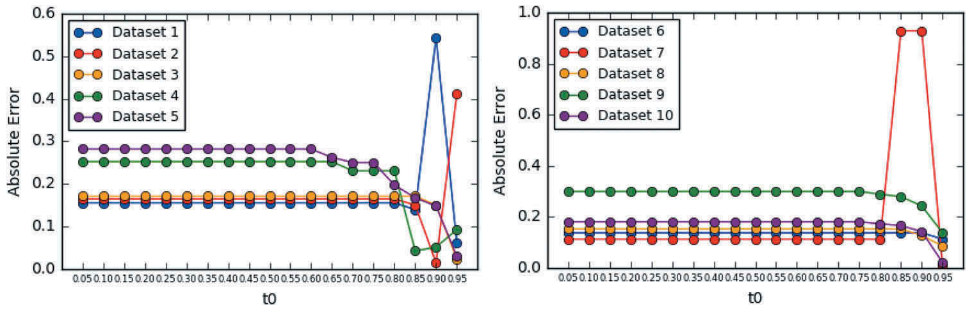

Figure 11. Absolute error (AE) of reputation value computed by our method.

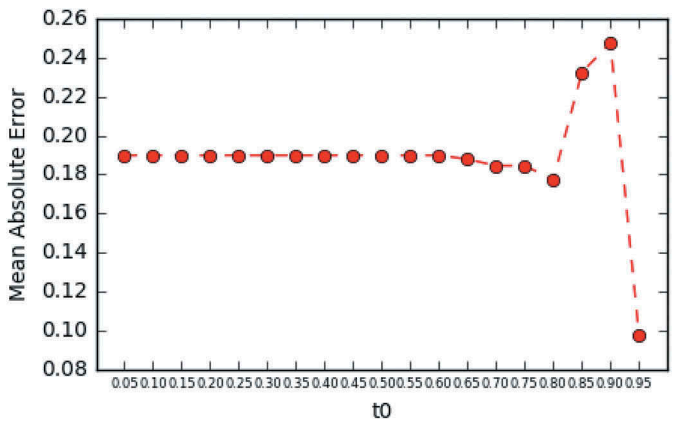

Figure 12. $MAER_{t_0}$ between IMDb weighted average votes and the reputation values computed by our method using all the reviews of 10 movies for t_0 varied from 0.05 to 0.95 with a step of 0.05.

Table 7. $MAER_{t_0}$ between IMDb weighted average votes and the reputation values computed by our method using all the reviews of 10 movies for t_0 value varied from 0.05 to 0.95 with a step of 0.05.

Opinion fusion threshold t_0 value	$MAER_{t_0}$
$t_0 = 0.05$	0.18978855
$t_0 = 0.10$	0.18978855
$t_0 = 0.15$	0.18978855
$t_0 = 0.20$	0.18978855
$t_0 = 0.25$	0.18978855
$t_0 = 0.30$	0.18978855
$t_0 = 0.35$	0.18978855
$t_0 = 0.40$	0.18978855
$t_0 = 0.45$	0.18978855
$t_0 = 0.50$	0.18978855
$t_0 = 0.55$	0.18978855
$t_0 = 0.60$	0.18978855
$t_0 = 0.65$	0.18780241
$t_0 = 0.70$	0.18445943
$t_0 = 0.75$	0.18445943
$t_0 = 0.80$	0.17724106
$t_0 = 0.85$	0.23243659
$t_0 = 0.90$	0.24759925
$t_0 = 0.95$	**0.09713621**

average votes (ground truth) and the reputation values generated by our method using all movies of datasets for each t_0 value between 0.05 and 0.95 with a step of 0.05.

Both Figure 12 and Table 7 show that the $MAER_{t_0}$ (6) is stable when $t_0 \leq 0.60$. We also found that our reputation generation method performs better when $t_0 = 0.95$, since the Mean Absolute Error between IMDb weighted average votes and the values computed by formula (3) using all the reviews of 10 movies reaches its minimum.

We compute the $MAER_{t_0}$ by applying this formula:

$$MAER_{t_0} = \frac{1}{E} \cdot \sum_{1 \leq i \leq E} AER_{t_0}^{m_i} \qquad (6)$$

We denote:

$MAER_{t_0}$: Mean Absolute Error between IMDb weighted average votes (ground truth) and the reputation values generated by our method for all movies of datasets for a specific t_0 value.

E: number of target entities (In this case $E = 10$).

4.7. *Classification phase impact on reputation generation*

As shown in Figure 3, Our proposed classification step gives the highest sentiment classification accuracy for all datasets compared with Naïve Bayes and SVM classifiers. It's important to analyze the impact of the classification step on reputation generation. For this purpose, we conducted more experiments in which we compared our method with two other approaches. The first approach uses Naïve Bayes for the classification phase, and the second one uses SVM. For each approach, we computed the $MAER^{m_i}$ between IMDb weighted average vote and the final reputation value computed by each approach for all datasets using formula (5).

We can see from Figure 13 that our proposed classification step leads to the lowest reputation Mean Absolute Error value (5) for datasets 1, 2, 3, 4, 5, 6 and 10, more than that, there is a very

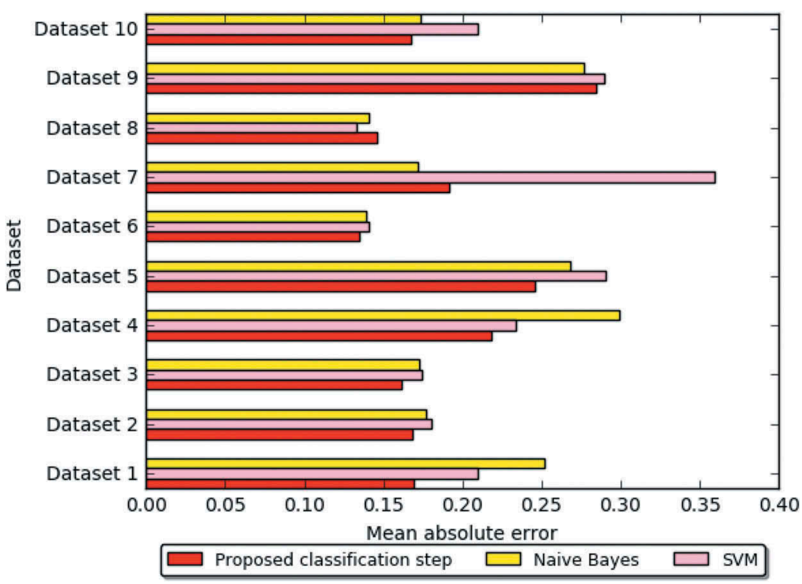

Figure 13. Reputation mean absolute error for the three approaches.

small difference between the $MAER^{m_i}$ of our approach and the $MAER^{m_i}$ computed during the use of Naïve Bayes for the classification phase in datasets 7, 8 and 9. We can conclude that the use of a high accuracy sentiment classification step leads to an accurate reputation value toward the target entity.

5. Summary and limitations

In this work, we have proposed an approach for reputation generation based on two classifiers: Naïve Bayes and Linear Support Vector Machine (LSVM) that aim to separate positive and negative opinions before grouping them into different opinion sets based on semantic relations, we also propose to compute reputation value using the Weighted Arithmetic Mean. As has been noted, our approach improves the work proposed in Yan, Jing, and Pedrycz (2017) by generating an accurate reputation value toward different movies. However, the main issues of our work are:

- Classification step: the classifier depends on the domain of use (politics, economic, business, sport …) and the language (English, French, Chinese, Arabic …). Consequently, it is not practical to use different datasets for each domain and each language. Fortunately, there are many methods to handle those issues such as Machine Translation, Transfer Learning, etc… Sun, Luo, and Chen (2017).
- Useless reviews: a lot of reviews are irrelevant to the target entity (ads, spam). Therefore, adding a filtering phase will reduce the processing time and increase the accuracy of the reputation system (the system will only deal with relevant and useful reviews). Many approaches that were proposed to handle this task are mentioned in Sun, Luo, and Chen (2017). In addition, Wang et al. (2015) proposed an approach to rank reviews by fuzing and mining opinions based on review pertinence.

Notes on contributors

Abdessamad Benlahbib has received his Master degree in computer science from the Faculty of Science Dhar El Mahraz of Sidi Mohamed Ben Abdellah University, Fez, Morocco. Currently, he is pursuing his Ph.D. studies at the Faculty of Science Dhar El Mahraz, Fez, Morocco. His research interests are in machine learning and its application to reputation generation and natural language processing (NLP).

El Habib Nfaoui is currently an associate Professor of Computer Science at the University of Sidi Mohammed Ben Abdellah. He obtained his PhD in Computer Science from University of Sidi Mohamed Ben Abdellah in Morocco and University of Lyon (LIESP Laboratory) in France under a COTUTELLE (co-advising) agreement. His current research interests are Information Retrieval, Semantic Web, Social networks, Machine learning, Web services, Multi-Agent Systems, Decision-making and modeling. He is a Guest Editor at the International Journal of Intelligent Engineering Informatics (ACM, DBLP…). He co-founded the International Conference on Intelligent Systems and Computer Vision (ISCV2015) and has served as Program Committee of various conferences. He has published several papers in reputed journals and international conferences.

References

Amblee, N., R. Ullah, and W. Kim. 2017. Do product reviews really reduce search costs? *Journal of Organizational Computing and Electronic Commerce* 27 (3):199–217. doi:10.1080/10919392.2017.1332142.

Baktha, K., and B. K. Tripathy. 2017. Investigation of recurrent neural networks in the field of sentiment analysis. 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2047–50, April 6-8.

Balazs, J. A., and J. D. Velásquez. 2016. Opinion mining and information fusion. *Information Fusion* 27 (C, January):95–110. ISSN 1566-2535. doi:10.1016/j.inffus.2015.06.002.

Barnaghi, P., P. Ghaffari, and J. G. Breslin. 2016. Opinion mining and sentiment polarity on twitter and correlation between events and sentiment. 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, UK, 52–57, March 29–April 1.

Cambria, E. 2016. Affective computing and sentiment analysis. *IEEE Intelligent Systems* 31:102–07. doi:10.1109/MIS.2016.31.

Farmer, R., and B. Glass. 2010. *Building web reputation systems*. Sebastopol, CA: O'Reilly.

He, W., Z. Zhang, and V. Akula. 2018. Comparing consumer-produced product reviews across multiple websites with sentiment classification. *Journal of Organizational Computing and Electronic Commerce* 28 (2):142–56. doi:10.1080/10919392.2018.1444350.

Hu, M., and B. Liu. 2004. Mining and summarizing customer reviews. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, New York, NY, 168–77. ACM. ISBN 1-58113-888-1. doi: 10.1145/1014052.1014073.

Jing, R., Y. Yu, and Z. Lin. 2015. How service-related factors affect the survival of b2t providers: A sentiment analysis approach. *Journal of Organizational Computing and Electronic Commerce* 25 (3):316–36. doi:10.1080/10919392.2015.1058125.

Jøsang, A., R. Ismail, and C. Boyd. 2007. A survey of trust and reputation systems for online service provision. *Decision Support Systems* 43 (2, March):618–44. ISSN 0167-9236. doi:10.1016/j.dss.2005.05.019.

Kennedy, A., and D. Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence* 22:110–25. doi:10.1111/coin.2006.22.issue-2.

Liu, B. 2010. Sentiment analysis and subjectivity. In *Handbook of natural language processing*,2nd ed. Boca: Taylor and Francis Group.

Liu, H., and M. Cocea. 2017. Fuzzy rule based systems for interpretable sentiment analysis. 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), Doha, Qatar, 129–36, February 4–6.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Communications of the ACM* 38 (11, November):39–41. ISSN 0001-0782. doi:10.1145/219717.219748.

Mohan, I., K. Janani, and M. Karthiga. 2017. A survey on sentiment analysis on social network data. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 2:0:562–68.

Nguyen, L. T., P. Wu, W. Chan, W. Peng, and Y. Zhang. 2012. Predicting collective sentiment dynamics from time-series social media. Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining, WISDOM '12, New York, NY, Vol. 6, 1–6. ACM. ISBN 978-1-4503-1543-2. doi:10.1145/2346676.2346682.

Pang, B., and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics, ACL '04, Stroudsburg, PA. Association for Computational Linguistics. doi:10.3115/1218955.1218990.

Pang, B., L. Lee, and S. Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP '02, Stroudsburg, PA, 79–86. Association for Computational Linguistics. doi:10.3115/1118693.1118704.

Peng, W., and D. H. Park. 2011. Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization. The Fifth International AAAI Conference on Weblogs and Social Media (ICWSM-11), Barcelona, Spain, July 17–21.

Poria, S., E. Cambria, and A. F. Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems* 108:42–49. doi:10.1016/j.knosys.2016.06.009.

Resnick, P. and R. Zeckhauser. 2002. Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. *The Economics of the Internet and ECommerce*. ed. M. R. Baye, Volume 11 of Advancesin Applied Microeconomics. Amsterdam, Elsevier Science. 127–157

Schneider, J., G. Kortuem, J. Jager, S. Fickas, and Z. Segall. 2000. Disseminating trust information in wearable communities. *Personal and Ubiquitous Computing* 40 (4, January):245–48. ISSN 1617-4909. doi:10.1007/PL00000012.

Shawe-Taylor, J., and S. Sun. 2011. A review of optimization methodologies in support vector machines. *Neurocomputing* 74:3609–18. doi:10.1016/j.neucom.2011.06.026.

Sun, C., L. Huang, and X. Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, 380–85. Association for Computational Linguistics. doi:10.18653/v1/N19-1035.

Sun, S., C. Luo, and J. Chen. 2017. A review of natural language processing techniques for opinion mining systems. *Information Fusion* 36 (C, July):10–25. ISSN 1566-2535. doi:10.1016/j.inffus.2016.10.004.

Tang, D., B. Qin, and T. Liu. 2015. Deep learning for sentiment analysis: Successful approaches and future challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5 (6, November):292–303. ISSN 1942-4787. doi:10.1002/widm.1171.

Turney, P. D. 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl. Proceedings of the 12th European Conference on Machine Learning, EMCL '01, London, UK, 491–502. Springer-Verlag. ISBN 3-540-42536-5. http://dl.acm.org/citation.cfm?id=645328.650004.

Turney, P. D. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, Stroudsburg, PA, 417–24. Association for Computational Linguistics. doi:10.3115/1073083.1073153.

Wang, J.-Z., Z. Yan, L. T. Yang, and B.-X. Huang. 2015. An approach to rank reviews by fusing and mining opinions based on review pertinence. *Information Fusion* 23 (C, May):3–15. ISSN 1566-2535. doi:10.1016/j.inffus.2014.04.002.

Wang, Y., M. Huang, X. Zhu, and L. Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, 606–15. Association for Computational Linguistics. doi:10.18653/v1/D16-1058.

Ya-Han, H., Y.-L. Chen, and H.-L. Chou. 2017. Opinion mining from online hotel reviews a text summarization approach. *Information Processing and Management* 530 (2, March):436–49. ISSN 0306-4573. doi:10.1016/j.ipm.2016.12.002.

Yan, Z. 2013. *Trust management in mobile environments: Autonomic and usable models*. 1st ed. Hershey, PA: IGI Global. ISBN 1466647655, 9781466647657.

Yan, Z., X. Jing, and W. Pedrycz. 2017. Fusing and mining opinions for reputation generation. *Information Fusion* 36:172–84. doi:10.1016/j.inffus.2016.11.011.