
Journal of Big Data
Editors-In-Chief:
Borko Furht · Taghi M. Khoshgoftaar

Submit manuscript

Contact
Editorial Board

Editorial Board

Editorial Board

Sign up for article alerts and news from this journal

Follow

Annual Journal Metrics

Speed
40.7 days to first decision for reviewed manuscripts only
40.3 days to first decision for all manuscripts
108 days from submission to acceptance
18 days from acceptance to publication

Citation Impact
4.117 - Source Normalized Impact per Paper (SNIP)
1.124 - SCImago Journal Rank (SJR)
8.79 - CiteScore

Usage
647,436 downloads
549 Altmetric mentions

More about our metrics

Journal of Big Data Accepted into Scopus!

We are pleased to announce that the *Journal of Big Data* has been accepted into Scopus, the world's largest abstract and citation database of peer-reviewed literature. Read more about the journal's abstract and indexing on the 'About' page.

Latest Tweets

ISSN: 2196-1115 (electronic)

Submit manuscript

Featured article: DIMPL: a distributed in-memory drone flight path builder system

Drones are increasingly being used to perform risky and labor intensive aerial tasks cheaply and safely. To ensure operating costs are low and flights autonomous, their flight plans must be pre-built. In existing techniques drone flight paths are not automatically pre-calculated based on drone capabilities and terrain information. This paper presents details of an automated flight plan builder DIMPL that pre-builds flight plans for drones tasked with surveying a large area to take photographs of electric poles to identify ones with hazardous vegetation overgrowth.

27. **Development of a predictive model for on-time arrival flight of airliner by discovering correlation between flight and weather data**

    An important business of airlines is to get customer satisfaction. Due to bad weather, a mechanical reason, and the late arrival of the aircraft to the point of departure, flights delay and lead to customer di...

    Authors: Noriko Etani

    Citation: *Journal of Big Data* 2019 6:85

    Content type: Research

    Published on: 20 September 2019

    - View Full Text
    - View PDF

28. **Evaluating the performance of sentence level features and domain sensitive features of product reviews on supervised sentiment analysis tasks**

    With the popularity of e-commerce, posting online product reviews expressing customer's sentiment or opinion towards products has grown exponentially. Sentiment analysis is a computational method that plays an...

    Authors: Bagus Setya Rintyarna, Riyanarto Sarno and Chastine Fatichah

    Citation: *Journal of Big Data* 2019 6:84

    Content type: Research

    Published on: 12 September 2019

    - View Full Text
    - View PDF

29. **Advancing community detection using Keyword Attribute Search**

    As social network structures evolve constantly, it is necessary to design an efficient mechanism to track the influential nodes and accurate communities in the networks. The attributed graph represents the inf...

    Authors: Sanket Chobe and Justin Zhan

    Citation: *Journal of Big Data* 2019 6:83

    Content type: Research

    Published on: 7 September 2019

    - View Full Text
    - View PDF

Journal of Big Data

Research | Open Access | Published: 12 September 2019

Evaluating the performance of sentence level features and domain sensitive features of product reviews on supervised sentiment analysis tasks

Bagus Setya Rintyarna, Riyanarto Sarno & Chastine Fatichah

Journal of Big Data 6, Article number: 84 (2019) | Cite this article

Download PDF

Download ePub

Sections   Figures   References

Abstract

Introduction

Rintyarna *et al. J Big Data* (2019) 6:84
https://doi.org/10.1186/s40537-019-0246-8

Journal of Big Data

RESEARCH Open Access

Evaluating the performance of sentence level features and domain sensitive features of product reviews on supervised sentiment analysis tasks

Bagus Setya Rintyarna[1,2*], Riyanarto Sarno[1] and Chastine Fatichah[1]

*Correspondence:
bagus.setya@unmuhjember.ac.id
[2] Department of Electrical Engineering, Universitas Muhammadiyah Jember, Jember, Indonesia
Full list of author information is available at the end of the article

Abstract
With the popularity of e-commerce, posting online product reviews expressing customer's sentiment or opinion towards products has grown exponentially. Sentiment analysis is a computational method that plays an essential role in automating the extraction of subjective information i.e. customer's sentiment or opinion from online product reviews. Two approaches commonly used in Sentiment analysis tasks are supervised approaches and lexicon-based approaches. In supervised approaches, Sentiment analysis is seen as a text classification task. The result depends not only on the robustness of the machine learning algorithm but also on the utilized features. Bag-of-word is a common utilized features. As a statistical feature, bag-of-word does not take into account semantic of words. Previous research has indicated the potential of semantic in supervised SA task. To augment the result of sentiment analysis, this paper proposes a method to extract text features named sentence level features (SLF) and domain sensitive features (DSF) which take into account semantic of words in both sentence level and domain level of product reviews. A word sense disambiguation based method was adapted to extract SLF. For every similarity employed in generating SLF, the SentiCircle-based method was enhanced to generate DSF. Results of the experiments indicated that our proposed semantic features i.e. SLF and SLF + DSF favorably increase the performance of supervised sentiment analysis on product reviews.

Keywords: Sentiment analysis, Online product reviews, Supervised approach, Machine learning

Introduction

The exponential growth of e-commerce has triggered it to become a rich source of information nowadays. On e-commerce, customers provide a qualitative evaluation in the form of an online review that describes their opinions on a specific product [1]. With a huge number of OPRs, manual processing is not an efficient task. Sentiment analysis (SA) technique emerges in response to the requirement of processing OPRs in speed [2]. In terms of product review analysis, SA which is also named Opinion Mining can be defined as a task of recognizing customer's opinion or sentiment toward the products or the product features [3] that can be categorized into positive, negative, or neutral responses [4]. SA plays an important role to automate the extraction of subjective information i.e. sentiment embodied in OPRs. The success of SA application on product reviews will in turn help customers in suggesting about buying a certain product [5] based on the analysis of OPRs. Meanwhile, for companies and online marketers, they can make use SA technique to foresee customer satisfaction toward a certain product [6]. Two major approaches commonly employed for SA tasks on product reviews are lexicon-based approaches and ML-based approaches [7]. In extracting opinions or sentiments from the text data, lexicon-based methods rely on a sentiment lexicon e.g. SentiwordNet [8], SO-CAL [9], MPQA subjectivity lexicon [10], Harvard general inquirer, Bing Liu's opinion lexicon [11], SenticNet [12], and NRC emotion lexicon [13]. Sentiment lexicon is a dictionary of precompiled sentiment terms [14]. Sentiment term is term, commonly verb and adjective, representing the sentiment of the text document. In brief, lexicon-based method extract all sentiment terms for any given text and assign their sentiment value using sentiment lexicon. Meanwhile, ML-based techniques rely on ML algorithms and see SA as a regular text classification task. Text classification task assigns a piece of text data into several predefined classes involving ML algorithms [15]. In terms of SA task, ML-based techniques classify text document into one out of three classes namely positive class, neutral class, and negative class. For a given set of training text data, ML algorithms build a model based on the extracted features of a labeled text. The model is then utilized to classify unlabeled text. The result of supervised SA task is therefore influenced by the robustness of both extracted text features and ML algorithms. Mostly, recent works [16–19] dealing with supervised SA concerned more on the extension of the employed ML algorithms instead of the development of robust text features. We briefly overview those works on "Related work" section. Concerning on the extraction of text features is therefore still challenging task in the area of supervised SA.

Referring to the previously research gap, the motivation for this study comprises:

1. Enhancing the result of supervised SA by proposing a method to extract robust text features for supervised SA task.
2. Evaluating the performance of the proposed text features using several ML-algorithms and feature selection methods.

In proposing the method to extract text features for supervised SA, we consider the finding reported by [3]. Rintyarna [3] highlighted the importance of semantics for SA task. Taking into account semantics of words is important for SA since the same term appears in different text data may reveals different meaning i.e. different sentiment value. In turn, capturing sementics is potential to augment the result of Sentiment Analysis task. In this study, we present a method to extract text features capturing semantic in sentence level and domain level of product reviews. We introduce two feature sets namely sentence level feature (SLF) and domain sensitive feature (DSF). For extracting SLF, a WSD based technique was adapted [20]. And for extracting DSF, a Senti-Circle based method was enhanced. We arrange several scenarios of experiment using several ML algorithms and feature selection methods to evaluate our proposed features compared with common features employed for SA task i.e. BOW. We utilized Waikato Environment for Knowledge Analysis (WEKA) for the implementation of ML-algorithms and feature selection methods. The result of experiment indicated that our proposed features outperformed BOW.

The rest of the manuscript is arranged in the following sections. "Related work" section reviews state of the art study related with this work. "Proposed method" section describes the proposed method for extracting SLF and DLF. We explore the result of experiment and the discussion in "Experimental results and discussion" section. Finally we summarize the result of this work in "Conclusion" section.

Related work

Using BOW, [16] performed an SA task on an Amazon product review dataset. RFSVM, a hybrid method that combines Random Forest (RF) and Support Vector Machine (SVM), was employed to make use of the capabilities of both classifiers. Precision, recall, F-Measure, and accuracy were used as the performance metrics to evaluate the proposed method compared with the baseline methods i.e. RF and SVM. Using instances of 500 positive datasets and 500 negative datasets, the result of the experiment showed that RFSVM outperformed the baseline methods in terms of all three performance metrics.

A word embedding-based sentiment classification is proposed [17]. Using google toolkit word2vec, a continuous bag-of-words (CBOW) model and a Skip-gram model were generated in order to produce meaningful features. For representing the document, the sum of weighted word embeddings was used. Combined with SVM, this work proposed an extension of the SVM classifier, called SVM-WE. The method was evaluated using four datasets i.e. RT-s, CR, RT-2k, and IDBM. The result of the experiment indicated that the proposed method performed slightly better compared with the baseline method.

Another work [18] proposed a set of 13 sentiment features for supervised SA in Twitter dataset classification. Features F1 to F8 were generated based on three sentiment lexicons, i.e. SenticNet, SentiWordNet, and NRC Emotion Lexicon. Features F9 to F13 were generated using a seed word list i.e. Subjective Words. Two datasets, namely TaskA Twitter and TaskB Twitter, were employed to validate feature performance in classification. The Naïve Bayes classifier was used as performance metric to calculate its accuracy. The best accuracy achieved by the proposed features was 75.60%.

In order to analyze social media content, Yoo [19] proposed a system to predict user sentiment. For representing the text data, the work adopted a two-dimensional representation of word2vec. The model for the sentiment analysis task was built using Convolutional Neural Network for Sentence Classification (CNN) by making use of TensorFlow, an open-source library for various dataflow programming tasks. Validated using the Sentiment140 dataset, containing 800,000 positive documents and 800,000 negative documents, the proposed model outperformed th baseline method i.e. Naïve Bayes, SVM, and Random Forest.

As an utmost advanced topic in the field of Natural Language Processing (NLP), many approaches have been developed for SA application [21]. Among the approaches is called Aspect Based Sentiment Analysis (ABSA). The main task of ABSA is inferring the sentiment polarity toward a specific target called aspect within a given piece of text data. In terms of product review analysis, it is useful for determining the product features that require improvement [22]. In the following paragraphs we briefly review several works discussing ABSA.

A method called joint aspect-based sentiment topic (JABST) has been introduced [23]. It proposed a unified framework to perform common ABSA task including aspect extraction and sentiment polarity identification. The study made use graphical model to describe relationship among aspects, opinion, sentiment polarity and granularity. A maximum entropy based model called MaxEnt-JABST has also been proposed to improve the word distribution description. In the evaluation step, two real world datasets from [24] were employed. The evaluation step focused on two points i.e.: (1) comparing the quality of the extracted topics and (2) calculating the precision of aspects and opinions. The results of experiment confirmed that the JABST significantly outperformed baseline model.

To perform ABSA tasks on customer reviews, a novel system called W2VLDA was presented by [25] based on the combination of a topic modeling approach and a Maximum Entropy classifier. The system performed the main tasks of ABSA simultaneously. Employing Brown cluster to train the model of Maximum Entropy classifier, W2VLDA was able to separate aspect-terms and opinion-words into word classes without any language dictionary. The work conducted experiment to evaluate the performance of different subtasks using different datasets. Restaurant review dataset [26] containing domain-related aspects was used to evaluate aspect category classification. Dataset on the domain of Laptops and Digital-SLR [24] containing English reviews was employed to evaluate sentiment classification subtask. Meanwhile, SemEval-2016 task 5 from [27] was used to perform multilingual experiments. Compared with the other LDA-based approaches as baseline methods, the system achieved slightly better results.

Another work [28] focused on three subtasks of ABSA i.e.: sentiment extraction, aspect assignment, and aspect category determination. The work contributed to improving the functionality of the current state-of-the-art topic model approach by adding product description as another dimension of the model. Two extended topic model-based ABSA methods were presented: Seller-aided Aspect-based Sentiment Model (SA-ASM) and Seller-aided Product-based Sentiment Model (SA-PSM). SA-ASM outperformed two baseline methods on sentiment classification and aspect assignment. Meanwhile, SA-PSM performed better compared with the baseline methods on subtask aspect categorization.

Aspect extraction which aims at identifying the object of user's opinion from online reviews holds an important role in ABSA approach. Motivated by the vulnerability of syntactic patterns-based approach due to its dependency to dependency parser, a study [29] proposed two-fold rule-based model (TF-RBM) to perform ABSA tasks. Sequential pattern-based rules (SPR) [30] was firstly employed to extract all aspects and opinions. Since many extracted aspects were not related to the product, the study performed a pruning method based on normalized Google distance calculation to improve aspect extraction accuracy. The last step of the proposed method was called concept extraction i.e. domain specific opinions that reveal user's sentiment.

Proposed method

The steps of the proposed method are: (1) capturing semantic values in product review texts at the sentence level and extracting the sentence level features (SLF), (2) capturing semantic values in product reviews influenced by different product domain extracting the domain sensitive features (DSF). Since there are many notations employed in this section, we present details of the notations in Table 1.

Extracting sentence level feature (SLF)

Capturing sentence-level semantic is important since the same words that appear in different piece of text may share different meaning i.e. different sentiment value as described in Table 1. In Table 2, we describe that the word "enjoy" has different sense i.e. different sentiment value when it appears in different sentence. This characteristic is known as polysemy. The task aims at assigning correct sentiment value to a word with respect to its local context i.e. sentence. We describe the step of extracting SLF in Fig. 1.

Table 1 Details of notations

Notations	Details
D	Product review document
S_k	Review sentence within D with index-k
w_i	Word with index-i
ws_i^j	Sense of word i with index-j
$spos_i^j$	Raw positive sentiment value of ws_i^j picked from SentiwordNet
$sneg_i^j$	Raw negative sentiment value of ws_i^j picked from SentiwordNet
$sneu_i^j$	Raw neutral sentiment value of ws_i^j picked from SentiwordNet
sim_{ab}^{cd}	Similarity value between ws_a^c and ws_b^d calculated using one of Wordnet similarity algorithms
$deg\left(ws_i^j\right)$	Indegree score of ws_i^j
$cspos_i$	Contextual positive sentiment value of w_i
$csneg_i$	Contextual negative sentiment value of w_i
$csneu_i$	Contextual neutral sentiment value of w_i
$fposS_k$	Positive value of feature of S_k
$fnegS_k$	Negative value of feature of S_k
$fneuS_k$	Neutral value of feature of S_k
$fposD$	Positive value of feature of D
$fnegD$	Negative value of feature of D
$fneuD$	Neutral value of feature of D
wd	Domain word of product review dataset
pw_k	Pivot word of review sentence S_k
θ_i	Angle representing semantic orientation adjustment of w_i
r_i	Degree of correlation between pw_k and w_i
cts_i	Prior sentiment value of w_i determined using Rule (11)
x_i	Senticircle representation in Cartesian coordinate
y_i	Senticircle representation in Cartesian coordinate
fxS_k	Feature value of S_k calculated using x of Senticircle
fyS_k	Feature value of S_k calculated using y of Seinticircle
fxD	Feature value of D calculated using x of Senticircle
fyD	Feature value of D calculated using y of Senticircle

Table 2 Example of different sentiment of the word "enjoy"

Word	Sentence	Sense	Sentiment
enjoy	I *enjoy* using the camera of this smartphone	Get pleasure from	Positive
	The vendor *enjoys* new regulation issued by the authority	Possess and benefit from	Neutral

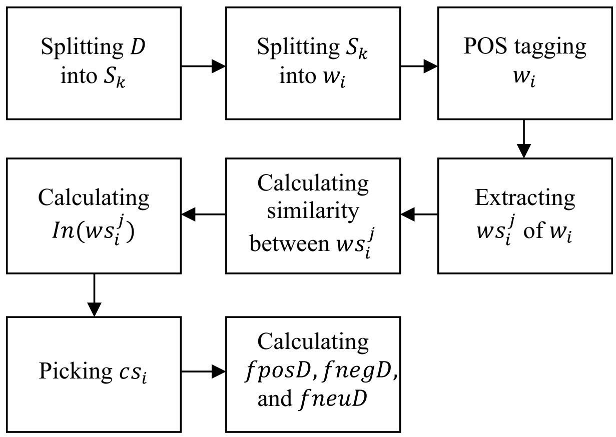

Fig. 1 The calculation of *fposD*, *fnegD*, and *fneuD*

To capture semantic value in product reviews at sentence level i.e. extracting SLF, product review document D is split into review sentence S_k. The process is done at sentence level. Suppose S_k consists of n words, $w_1, w_2, \ldots w_n$. The aim of this stage is to find contextual sentiment value cs_i of word w_i associated with sentiment score s_i picked from SentiwordNet [8]. In the next step, part of speech (POS) tagging is done, which is part of common text processing, including filtering. It is a process of assigning a part of speech value to a word in a piece of text [31]. Since we employ SentiwordNet [8], which is based on WordNet [32], POS tagging is important for selecting the correct sense of w_i in accordance with its POS tag [33]. WordNet [32] itself employs 4 POS tags, i.e. *noun*, *verb*, *adjective*, and *adverb*. POS tagging is important for the next step, i.e. extracting ws_i^j from w_i. For every extracted ws_i^j its associated sentiment value is picked from SentiwordNet [8]. Every ws_i^j has three different sentiment scores, namely $spos_i^j$, $sneg_i^j$, and $sneu_i^j$.

The similarity between ws_i^j is calculated using WordNet similarity algorithms, i.e. from Lin, Jiang and Conrath, Resnik, Leacock and Chodorow, and Wu and Palmer. Adapted Lesk [34] is also employed. Similarity between word senses, denoted as sim_{ab}^{cd}, means similarity value of ws_a^c and ws_b^d. They are calculated for all possible combinations, as can be seen in Table 3. The calculation adopts the WSD technique firstly introduced by [20]. For simple, the task illustrated in Table 4 can be assumed as building undirected weighted graph of every review sentence with ws_i^j as the vertex and sim_{ab}^{cd} as the weight of the edges of the graph.

The results of the previous step are the three different sentiment scores from SentiWordNet [8]. For example, the result of processing the review sentence '*The screen is*

Table 3 Similarity between word senses

		w_1			w_2	
		ws_1^1	ws_1^2	ws_1^3	ws_2^1	ws_2^2
w_1	ws_1^1				sim_{12}^{11}	sim_{12}^{12}
	ws_1^2				sim_{12}^{21}	sim_{12}^{22}
	ws_1^3				sim_{12}^{31}	sim_{12}^{32}
w_2	ws_2^1					
	ws_2^2					

Table 4 Word senses along with their sentiment score

Word	Senses		Sentiment score
w_1			
Screen	w_1^1	A white or silvered surface where pictures can be projected for viewing	$spos_1^1$ $sneg_1^1$ $sneu_1^1$
	w_1^2	A protective covering that keeps things out or hinders sight	$spos_1^2$ $sneg_1^2$ $sneu_1^2$
	w_1^3	The personnel of the film industry	$spos_1^3$ $sneg_1^3$ $sneu_1^3$
w_2			
Great	w_2^1	Relatively large in size or number or extent	$spos_2^1$ $sneg_2^1$ $sneu_2^1$
	w_2^2	Of major significance or importance	$spos_2^2$ $sneg_2^2$ $sneu_2^2$

great', can be seen in Table 4. After the POS tagging step, including filtering, there are two words, i.e. 'screen' with POS tag *noun* and 'great' with POS tag *adjective*.

To assign cs_i of w_i, the indegree score of ws_i^j, denoted by $In\left(ws_i^j\right)$, is calculated. Indegree score is important to assign contextual sense of w_i. Among the senses of w_i i.e. ws_i^j, a sense with the highest Indegree score is assigned as contextual sense of w_i. Contextual sense is a sense where $cspos_i$, $csneg_i$, and $csneu_i$ are picked from the collection of SentiwordNet and assigned as contextual sentiment value of w_i. For the above case there are three indegree scores for w_1, i.e. $deg\left(ws_1^1\right)$, $deg\left(ws_1^2\right)$ and $deg\left(ws_1^3\right)$ while there are two indegree scores for w_2, i.e. $deg\left(ws_2^1\right)$ and $deg\left(ws_2^2\right)$. They are calculated as follows:

$$deg(ws_1^1) = sim_{12}^{11} + sim_{12}^{12}$$

$$deg\left(ws_1^2\right) = sim_{12}^{21} + sim_{12}^{22}$$

$$deg\left(ws_1^3\right) = sim_{12}^{31} + sim_{12}^{32}.$$

The next task is determining the selected sense of w_i by calculating $\max\{deg(ws_1^1), deg(ws_1^2), deg(ws_1^3)\}$. The sense that has the highest indegree score is selected as the contextual sense of w_i and its sentiment score is labeled with $cspos_i$, $csneg_i$, or $csneu_i$. Once these values have been assigned for every w_i, the last procedure in this step is calculating the numeric feature value at the sentence level, $fposS_k$, $fnegS_k$, and $fneuS_k$, using Eqs. (1), (2) and (3).

$$fposS_k = \sum_{i=1}^{n} cspos_i \tag{1}$$

$$fnegS_k = \sum_{i=1}^{n} cspos_i \tag{2}$$

$$fneuS_k = \sum_{i=1}^{n} cspos_i \tag{3}$$

where n is the number of words in S_k. To calculate the numeric feature value at review document level, Eqs. (4), (5), and (6) are employed. For o is the number of sentences in review document D, $fposD$, $fnegD$, and $fneuD$ are calculated as follows:

$$fposD = \frac{\sum_{k=1}^{o} fposS_k}{k} \tag{4}$$

$$fnegD = \frac{\sum_{k=1}^{o} fnegS_k}{k} \tag{5}$$

$$fneuD = \frac{\sum_{k=1}^{o} fneuS_k}{k} \tag{6}$$

Capturing domain sensitive features (DSF)

In this step, we adopt Senticircle approach [35]. The main principle of Senticircle suggest that terms exist in the same context tend to share the same semantics. In terms of product review, we define the context as product domain. In consequence, the same terms that appears in different product domains tend to share different meaning. In terms of SA, sharing different meanings means carrying different sentiment. For example, 'long battery life' in Electronics domain express positive sentiment, while 'long stopping time' in the Automobile domain share negative sentiment.

$$maxSim = argmax_{Sim_i} Sim_i(wd, w_i) \tag{7}$$

$$Sim_i = \frac{2 * Depth(LCS(wd, w_i))}{Depth(wd) + Depth(w_i)}. \tag{8}$$

To generate the DSF, several formulas are provided. Figure 2 describes the steps that need to be carried out. The first three steps, including POS tagging, are the same as in the first step of the method. The next step is determining pivot word pw_k of sentence S_k.

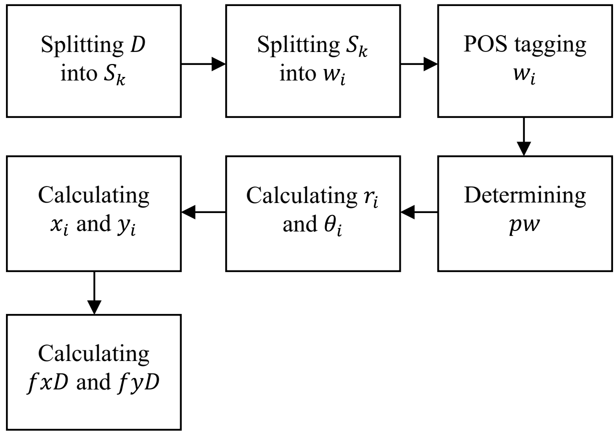

Fig. 2 The calculation of *fxD* and *fyD*

A pivot word is a representative of the domain word at sentence level [3]. In this work, pw_k is defined as the noun with the closest similarity to the domain word. For measuring similarity, Wu and Palmer's algorithm is employed [36]. For *wd* as the domain word (e.g. Smartphone, Book, Beauty, or Computers), the similarity between *wd* and w_i is computed using (7) and (8). The pivot word from w_i that has the highest value, *maxSim*, is selected.

In Eq. (8), *LCS* means the Least Common Subsumer between the first sense of *wd* and the contextual sense of w_i in the WordNet [32] taxonomy. Since the method from [37] was adopted in this stage, r_i is computed to represent the distance between w_i and pw_k using Eq. (9). In (9), N is the total number of words in the corpus of product reviews and Nw_i is the total number of w_i.

$$r_i = f(pw_k, w_i) \log \frac{N}{Nw_i} \tag{9}$$

To generate the SentiCircle representation of w_i, we need to assign θ_i using Eq. (10).

$$\theta_i = cts_i * \pi\, rad \tag{10}$$

In Eq. (10), cts_i is determined using rule (11).

$$cts_i = \begin{cases} cspos_i \text{ if } |cspos_i| > |csneg_i| \\ csneg_i \text{ if } |csneg_i| > |cspos_i| \end{cases} \tag{11}$$

The last step is to generate the SentiCircle representation by using (12) and (13). The sentiment value of a word is represented using the values of x and y in a Cartesian coordinate system as seen in Fig. 3. To calculate the numeric value of the features in sentence S_k, Eqs. (14) and (15) are introduced, where NwS_k is the number of words in S_k.

$$x_i = r_i \cos \theta_i \tag{12}$$

$$y_i = r_i \sin \theta_i \tag{13}$$

$$fxS_k = \frac{\sum_{i=1}^{NwS_k} x_i}{NwS_k} \tag{14}$$

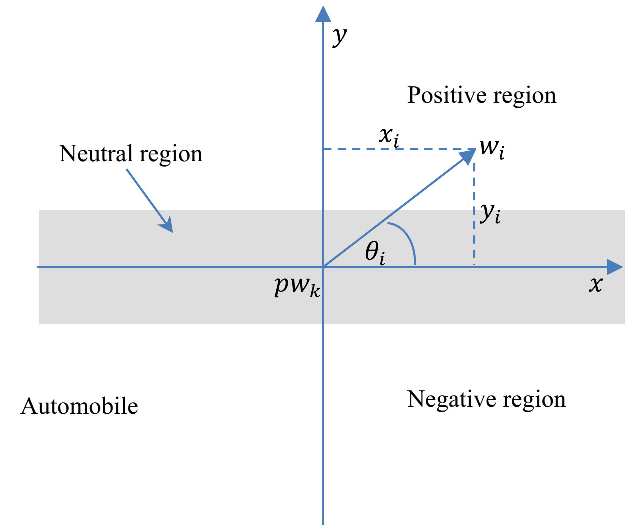

Fig. 3 Representation of Senticircle in Cartesian coordinate system

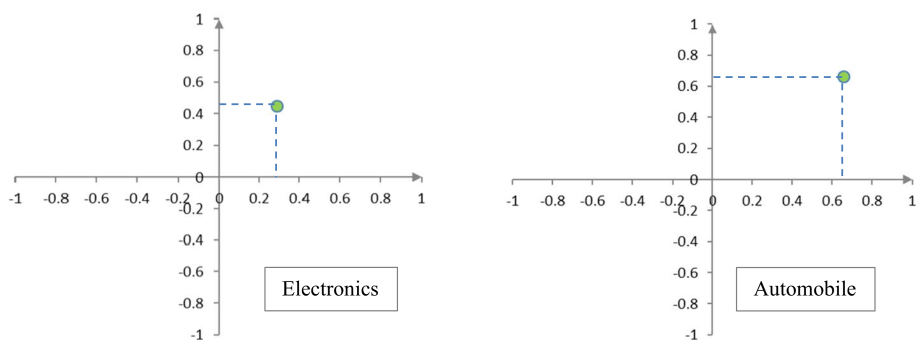

Fig. 4 Example of how Senticircle adjust Sentiment value of the word "long" of two different domains

$$fyS_k = \frac{\sum_{i=1}^{NwS_k} y_i}{NwS_k} \tag{15}$$

In Fig. 4, we provide an example of how Senticircle adjust a sentiment value of the same word "long" but from different domain e.g. Electronics and Automobile. The word "long" is picked from review document of the dataset as presented in Table 5. In Table 5, we also provide the variable value of the Senticircle of the word "long". In the first domain e.g. Electronics, the word "long" has relatively neutral value while in the second domain e.g. Automobile, this word has highly positive value. The value of x_i and y_i presented in the table is the value after normalization.

To represent a document with its semantic features, the numeric value of the features in the review document is calculated using Eqs. (16) and (17). In both equations, o is the number of sentences in D. For every similarity algorithm, a set of features is generated,

Table 5 Variable value of the word "long" calculated for both domains

Variable value	Domain	
	Electronics	**Automobile**
Review sentence	I mounted a shelf above the TV to get the cable box out of the way and avoid having to run a *long* HDMI cable through the wall	…but they are built solid, nice tough big hard clamps and love having a *long* cable so I never have to move cars around or anything if needed
pw_k	230	495
$f(pw_k, w_i)$	85	145
N	130,765	170,873
Nw_i	186	208
r	241	422
cts_i	0.32	0.25
θ_i	*57.6°*	*45°*
x_i	*0.29*	*0.66*
y_i	*0.45*	*0.66*

Italic values indicate Senticircle parameters calculated in both domains

i.e.: *fposD*, *fnegD*, *fneuD*, *fxD*, and *fyD*. Since 5 similarity algorithms are employed (Wu and Palmer, Jiang and Conrath, Leacock and Chodorow, Resnik, and Li), the complete set of review document features consists of 25 features, as listed in Table 6. In the table, we describe the notation of the features, the details and the type of the features. F1–F15 is local features. Meanwhile, F16–F25 is domain sensitive features.

$$fxD = \frac{\sum_{k=1}^{o} S_k}{o} \tag{16}$$

$$fyD = \frac{\sum_{k=1}^{o} S_k}{o} \tag{17}$$

Experimental results and discussion

Experimental setup

An experiment was conducted to evaluate the features extracted by the proposed method employing several machine learning algorithms available in WEKA [38], i.e. Bayesian Network, Naïve Bayes, Naïve Bayes Multinomial, Logistic, Multilayer Perceptron, J48, Random forest, and Random tree. Another experiment was conducted using feature selection method. In the implementation, WEKA feature selection methods were employed, i.e.: ClassifierAttributeEval (CA), GainRatioAttributeEval (GR), InfoGainAttributeEval, OneRAttributeEval (OneR) and PrincipalComponent (PCA). Precision, recall and F-measure were calculated as performance metrics. Although important, extending Machine learning algorithms is not part of our contribution. A key point of this work is to demonstrate as well as to evaluate the performance of our proposed semantic features. For that reason, in all experiment we employ default setting of the ML parameters provided by WEKA to avoid bias in the result of experiment. The experiments were performed on IBM System X3400 M3 Tower Server.

Table 6 Details of the features

Feature	Details	Type
F1 *fposD*(*wup*)	Average positive value of review document where Wu and Palmer is employed as similarity algorithm	Sentence level features (SLF)
F2 *fnegD*(*wup*)	Average negative value of review document where Wu and Palmer is employed as similarity algorithm	
F3 *fneuD*(*wup*)	Average neutral value of review document where Wu and Palmer is employed as similarity algorithm	
F4 *fposD*(*jcn*)	Average positive value of review document where Jiang and Conrath is employed as similarity algorithm	
F5 *fnegD*(*jcn*)	Average negative value of review document where Jiang and Conrath is employed as similarity algorithm	
F6 *fneuD*(*jcn*)	Average neutral value of review document where Jiang and Conrath is employed as similarity algorithm	
F7 *fposD*(*lch*)	Average positive value of review document where Leacock and Chodorow is employed as similarity algorithm	
F8 *fnegD*(*lch*)	Average negative value of review document where Leacock and Chodorow is employed as similarity algorithm	
F9 *fneuD*(*lch*)	Average neutral value of review document where Leacock and Chodorow is employed as similarity algorithm	
F10 *fposD*(*res*)	Average positive value of review document where Resnik is employed as similarity algorithm	
F11 *fnegD*(*res*)	Average negative value of review document where Resnik is employed as similarity algorithm	
F12 *fneuD*(*res*)	Average neutral value of review document where Resnik is employed as similarity algorithm	
F13 *fposD*(*lin*)	Average positive value of review document where Lin is employed as similarity algorithm	
F14 *fnegD*(*lin*)	Average negative value of review document where Lin is employed as similarity algorithm	
F15 *fneuD*(*lin*)	Average neutral value of review document where Lin is employed as similarity algorithm	
F16		

Table 6 (continued)

Feature	Details	Type
fxD(*wup*)	Average *x* value of review document where Wu and Palmer is employed as similarity algorithm	Domain sensitive features (DSF)
F17		
fyD(*wup*)	Average *y* value of review document where Wu and Palmer is employed as similarity algorithm	
F18		
fxD(*jcn*)	Average *x* value of review document where Jiang and Conrath is employed as similarity algorithm	
F19		
fyD(*jcn*)	Average *y* value of review document where Jiang and Conrath is employed as similarity algorithm	
F20		
fxD(*lch*)	Average *x* value of review document where Leacock and Chodorow is employed as similarity algorithm	
F21		
fyD(*lch*)	Average *y* value of review document where Leacock and Chodorow is employed as similarity algorithm	
F22		
fxD(*res*)	Average *x* value of review document where Resnik is employed as similarity algorithm	
F23		
fyD(*res*)	Average *y* value of review document where Resnik is employed as similarity algorithm	
F24		
fxD(*lin*)	Average *x* value of review document where Lin is employed as similarity algorithm	
F25		
fyD(*lin*)	Average *y* value of review document where Lin is employed as similarity algorithm	

Dataset description

The experiment was conducted using Amazon product data [39] downloaded from http://jmcauley.ucsd.edu/data/amazon/. The collection contains product review dataset grabbed from Amazon including 142.8 millions reviews. The experiment was conducted on a small subset of this collection, i.e. the electronics and automobile datasets. The number of sample for building model and running evaluation follow the rule of tenfold cross-validation. The dataset contains reviewerID, asin, reviewerName, helpfulness, reviewText, overall, summary, unixReviewTime, and reviewTime as described in Table 7. We pick the review text for experiment from reviewText. To build the ground truth, we established a label out of three sentiment categories i.e. positive, negative, and neutral for every reviewText based on its overall score. Datasets with overall score 1–2 were assigned as negative reviews. Meanwhile, reviewTexts with overall score 4–5 were labeled positive. And the rest was assigned as neutral review.

Table 7 Dataset details

Data	Details
reviewerID	ID of the reviewer
asin	ID of the product
revewerName	Name of the reviewer
Helpfulness	Helpfulness rating of the review
reviewText	Text of the review
Overall	Rating of the product
Summary	Summary of the review
unixReviewTime	Time of the review (unix time)
reviewTime	Time of the review (raw time)

Results and discussion

Three scenarios were arranged for the experiment, i.e. (1) using a baseline features i.e. BoW (BF) that is commonle employed for recent supervised sentiment analysis task, (2) using sentence level feature only (SLF), and (3) using sentence level features combined with domain sensitive features (SLF + DSF). For each scenario, we calculate precision, recall and F-measures as the performance metrics in tenfold cross validation. We present the result of the experiment in Tables 8 and 9.

We reveal the result of experiment using Electronic dataset on Table 8. We indicate the best performance of both SLF and SLF + DSF for precision, recall and F-measure using asterisk symbol. The best performance of SLF for precision, recall, and f measure is 0.792, 0.817, and 0.758 respectively. Meanwhile, SLF + DSF achieve the best performance by 0.823, 0.800, and 0.760 for precision, recall and F-measure respectively.

In Table 9, we describe the result of experiment using Automobile dataset. We also indicate the best performance of SLF and SLF + DSF using asterisk symbol. The top performance of SLF for Automobile dataset is achieved for precision, recall, and F-measure by 0.796, 0.847, and 0.811 respectively. Meanwhile, SLF + DSF works best for precision, recall, and F-measure by 0.825, 0.854, and 0.831 respectively.

In Fig. 5, we calculate the average performance of our proposed features over all ML algorithms and feature selection methods compared with the baseline features. We present the result in the bar charts. Both bar charts indicate that our proposed features outperformed the baseline features measured in all performance metrics. In average SLF favorably increase the performance by 6.2%, 6.1%, and 6.0% for precision, recall, and F-measure respectively. Meanwhile, SLF + DSF successfully augment the performance by 7.1%, 7.2%, and 7.4% for precision, recall and F-measure. Overall trend, SLF + DSF is better than SLF by 0.8%, 1%, and 1.2% for precision, recall and F-measure. Yet, in Electronic dataset, SLF + DSF experienced slight decrease by 0.3% for recall (as indicated by the arrow mark in Fig. 5a).

Table 8 Result of the experiment using electronics dataset

Feature selection method	ML algorithm	BF			Proposed feature					
					SLF			SLF + DSF		
		Prec	Rec	F-meas	Prec	Rec	F-meas	Prec	Rec	F-meas
None	Bayes Net	0.632	0.481	0.517	0.674	0.519	0.570	0.733	0.752	0.741
	Naïve Bayes	0.595	0.595	0.595	0.674	0.519	0.570	0.678	0.495	0.549
	Logistic	0.544	0.544	0.544	0.641	0.740	0.687	0.628	0.657	0.642
	MLP	0.701	0.722	0.710	0.707	0.750	0.724	0.712	0.733	0.722
	J48	0.607	0.658	0.629	0.651	0.798	0.717	0.743	0.733	0.738
	Random Forest	0.660	0.747	0.670	*0.792	*0.817	*0.758	0.823	0.646	0.752
	Random Tree	0.689	0.684	0.686	0.730	0.750	0.739	0.757	0.762	*0.760
CA	Bayes Net	0.632	0.481	0.517	0.674	0.519	0.570	0.733	0.752	0.741
	Naïve Bayes	0.595	0.595	0.595	0.674	0.519	0.570	0.678	0.495	0.549
	Logistic	0.544	0.544	0.544	0.641	0.740	0.687	0.628	0.657	0.642
	MLP	0.701	0.722	0.710	0.707	0.750	0.724	0.712	0.733	0.722
	J48	0.607	0.658	0.629	0.651	0.798	0.717	0.743	0.733	0.738
	Random Forest	0.660	0.747	0.670	*0.792	*0.817	*0.758	0.646	0.752	0.695
	Random Tree	0.689	0.684	0.686	0.730	0.750	0.739	0.757	0.762	*0.760
GR	Bayes Net	0.632	0.481	0.517	0.674	0.519	0.570	0.733	0.752	0.741
	Naïve Bayes	0.595	0.595	0.595	0.674	0.519	0.570	0.678	0.495	0.549
	Logistic	0.544	0.544	0.544	0.641	0.740	0.687	0.628	0.657	0.642
	MLP	0.701	0.722	0.710	0.707	0.750	0.724	0.712	0.733	0.722
	J48	0.607	0.658	0.629	0.651	0.798	0.717	0.743	0.733	0.738
	Random Forest	0.660	0.747	0.670	0.792	0.817	0.758	0.646	0.752	0.695
	Random Tree	0.689	0.684	0.686	0.730	0.750	0.739	0.757	0.762	*0.760
IG	Bayes Net	0.632	0.481	0.517	0.674	0.519	0.570	0.733	0.752	0.741
	Naïve Bayes	0.595	0.595	0.595	0.674	0.519	0.570	0.678	0.495	0.549
	Logistic	0.544	0.544	0.544	0.641	0.740	0.687	0.628	0.657	0.642
	MLP	0.701	0.722	0.710	0.707	0.750	0.724	0.712	0.733	0.722
	J48	0.607	0.658	0.629	0.651	0.798	0.717	0.743	0.733	0.738
	Random Forest	0.660	0.747	0.670	*0.792	*0.817	*0.758	0.646	0.752	0.695
	Random Tree	0.689	0.684	0.686	0.730	0.750	0.739	0.757	0.762	*0.760
OneR	Bayes Net	0.632	0.481	0.517	0.674	0.519	0.570	0.707	0.771	0.730
	Naïve Bayes	0.595	0.595	0.595	0.674	0.519	0.570	0.652	0.790	0.715
	Logistic	0.544	0.544	0.544	0.641	0.740	0.687	0.654	*0.800	0.720
	MLP	0.701	0.722	0.710	0.707	0.750	0.724	0.712	0.733	0.722
	J48	0.607	0.658	0.629	0.651	0.798	0.717	0.743	0.733	0.738
	Random Forest	0.660	0.747	0.670	0.792	0.817	0.758	0.714	0.714	0.714
	Random Tree	0.689	0.684	0.686	0.730	0.750	0.739	0.692	0.686	0.689
PCA	Bayes Net	0.560	0.544	0.552	0.648	0.683	0.665	0.733	0.752	0.741
	Naïve Bayes	0.590	0.620	0.604	0.648	0.683	0.665	0.679	0.619	0.645
	Logistic	0.550	0.633	0.589	0.648	0.779	0.707	0.644	0.743	0.690
	MLP	0.569	0.532	0.549	0.648	0.779	0.707	0.621	0.629	0.625
	J48	0.633	0.620	0.627	0.651	0.798	0.717	0.652	0.790	0.715
	Random Forest	0.564	0.696	0.623	0.649	0.788	0.712	0.648	0.762	0.700
	Random Tree	0.653	0.684	0.666	0.720	0.731	0.725	0.683	0.629	0.652

Table 9 Result of the experiment using automobile dataset

Feature selection method	ML algorithm	BF			Proposed feature					
					SLF			SLF + DSF		
		Prec	Rec	F-meas	Prec	Rec	F-meas	Prec	Rec	F-meas
None	Bayes Net	0.664	0.664	0.664	0.764	0.759	0.762	0.786	0.818	0.800
	Naïve Bayes	0.701	0.770	0.735	0.764	0.759	0.762	0.786	0.818	0.800
	Logistic	0.700	0.752	0.724	0.751	0.796	0.772	0.779	0.847	0.801
	MLP	0.739	0.796	0.761	0.782	0.810	0.795	0.770	0.810	0.788
	J48	0.681	0.761	0.719	*0.796	*0.847	*0.811	0.779	0.847	0.801
	Random Forest	0.689	0.814	0.747	0.740	0.847	0.790	0.740	0.847	0.790
	Random Tree	0.736	0.708	0.721	0.773	0.788	0.781	0.776	0.766	0.771
CA	Bayes Net	0.707	0.770	0.735	0.764	0.759	0.762	0.786	0.818	0.800
	Naïve Bayes	0.664	0.664	0.664	0.764	0.759	0.762	0.786	0.818	0.800
	Logistic	0.700	0.752	0.724	0.751	0.796	0.772	0.779	0.847	0.801
	MLP	0.739	0.796	0.761	0.782	0.810	0.795	0.770	0.810	0.788
	J48	0.681	0.761	0.719	*0.796	*0.847	0.811	0.779	0.847	0.801
	Random Forest	0.689	0.814	0.747	0.740	*0.847	0.790	0.740	0.847	0.790
	Random Tree	0.736	0.708	0.721	0.773	0.788	0.781	0.776	0.766	0.771
GR	Bayes Net	0.707	0.770	0.735	0.764	0.759	0.762	0.786	0.818	0.800
	Naïve Bayes	0.664	0.664	0.664	0.764	0.759	0.762	0.786	0.818	0.800
	Logistic	0.700	0.752	0.724	0.751	0.796	0.772	0.779	0.847	0.801
	MLP	0.739	0.796	0.761	0.782	0.810	0.795	0.770	0.81	0.788
	J48	0.681	0.761	0.719	*0.796	*0.847	*0.811	0.779	0.847	0.801
	Random Forest	0.689	0.814	0.747	0.740	*0.847	0.790	0.740	0.847	0.790
	Random Tree	0.736	0.708	0.721	0.773	0.788	0.781	0.776	0.766	0.771
IG	Bayes Net	0.707	0.770	0.735	0.764	0.759	0.762	0.786	0.818	0.800
	Naïve Bayes	0.664	0.664	0.664	0.764	0.759	0.762	0.786	0.818	0.800
	Logistic	0.700	0.752	0.724	0.751	0.796	0.772	0.779	0.847	0.801
	MLP	0.739	0.796	0.761	0.782	0.810	0.795	0.770	0.810	0.788
	J48	0.681	0.761	0.719	*0.796	*0.847	*0.811	0.779	0.847	0.801
	Random Forest	0.689	0.814	0.747	0.740	*0.847	0.790	0.740	0.847	0.790
	Random Tree	0.736	0.708	0.721	0.773	0.788	0.781	0.776	0.766	0.771
OneR	Bayes Net	0.664	0.664	0.664	0.764	0.759	0.762	0.786	0.818	0.800
	Naïve Bayes	0.728	0.717	0.722	0.764	0.759	0.762	0.786	0.818	0.800
	Logistic	0.700	0.752	0.724	0.751	0.796	0.772	0.779	0.847	0.801
	MLP	0.739	0.796	0.761	0.782	0.810	0.795	0.77	0.81	0.788
	J48	0.681	0.761	0.719	*0.796	*0.847	*0.811	0.779	0.847	0.801
	Random Forest	0.689	0.814	0.747	0.740	*0.847	0.790	0.74	0.847	0.790
	Random Tree	0.736	0.708	0.721	0.773	0.788	0.781	0.776	0.766	0.771
PCA	Bayes Net	0.681	0.761	0.719	0.770	0.810	0.788	0.806	*0.854	0.816
	Naïve Bayes	0.749	0.743	0.746	0.770	0.810	0.788	0.806	*0.854	0.816
	Logistic	0.688	0.805	0.742	0.740	0.847	0.790	0.740	0.847	0.790
	MLP	0.700	0.752	0.724	0.742	0.759	0.750	0.782	0.832	0.802
	J48	0.691	0.823	0.751	0.738	0.832	0.782	0.740	0.847	0.790
	Random Forest	0.688	0.805	0.742	0.740	*0.847	0.790	0.740	0.847	0.790
	Random Tree	0.741	0.752	0.747	0.766	0.766	0.766	*0.825	0.839	*0.831

Limitation of the study and the future work

SLF extraction is based on a word sense disambiguation technique that relies on WordNet similarity algorithms. Therefore, the result depends on the effectiveness of

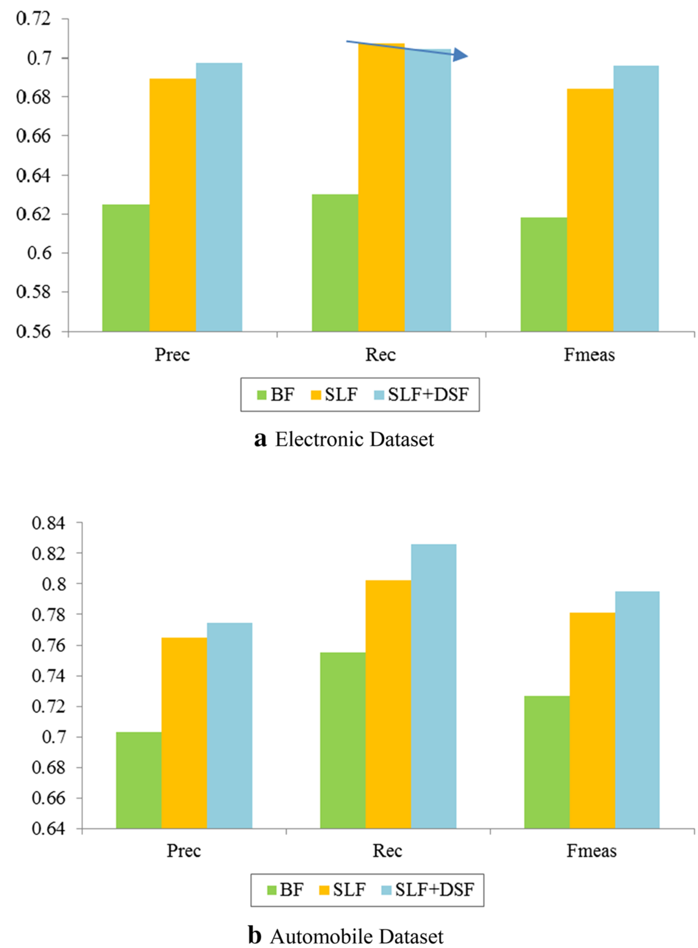

Fig. 5 Average performance of our proposed features compared with baseline features

Table 10 Technique for determining pivot word

Study	Rule for determining pivot word
Senticircle [37]	Simply pick word that has POS tags NN in tweet
This study	NN + similarity algorithm

the algorithms. Meanwhile, for SLF + DSF, the implementation is based on a Senticircle technique [37]. In this study, senticircle has an important role to adjust sentiment value of an opinion word based on its product domain. The value of cts_i that is the result of SLF has a role in determining sentiment orientation of an opinion word by assigning the value of θ_i. More importantly, pivot word pw_k is responsible for assigning the rate of the adjustment. Compare to Saif's technique in determining pivot word [37], this study has actually provided extension as seen in Table 10.

The extension and the adopted technique of SLF + DSF yields slight increase in performance metrics compared with SLF. In Electronic dataset, on the contrary, recall experienced slight decrease (see Fig. 3a). We hypothesize that pivot word is responsible for this result. Therefore in our future work we will develop technique to determine pivot word. We hypothesize that pivot word is product feature called aspect. We will develop rule to extract product aspect and carry a more fine grain SA task based on pair of aspect and opinion word to provide better increase in performance metrics. In the future work, we also plan to extent the implementation using Python and R language and big data platforms e.g. Hadoop, Sparkle.

Conclusion

We have implemented the proposed semantic features extraction namely SLF and DSF, which have achieved better performance on supervised SA task. The performance of the proposed features was evaluated using several machine learning algorithms and feature selection methods of WEKA compared with a baseline features. SLF favorably escalate the performance of SA task by 6.2%, 6.1%, and 6.0% for precision, recall, and F-Measure respectively. Meanwhile, SLF + DSF successfully enhance the performance of supervised SA by 7.1%, 7.2%, and 7.4% for precision, recall and F-Measure.

Abbreviations

OPRs: online product reviews; SA: sentiment analysis; ML: machine learning; BOW: bag of words; CBOW: continuous bag of words; WSD: word sense disambiguation; SLF: sentence level features; DSF: domain sensitive features; MPQA: multi perspective question answering; RF: Random Forest; SVM: Support Vector Machine; CNN: Convolutional Neural Network; POS: part of speech; LCS: Least Common Subsumer; MLP: multi layer perceptron; BF: baseline feature; CA: classifier attribute evaluator; GR: gain ratio attribute evaluator; IG: information gain attribute evaluator; OneR: one rule attribute evaluator; PCA: principal component analysis.

Acknowledgements

We would like to thank both Institut Teknologi Sepuluh Nopember and Universitas Muhammadiyah Jember for supporting this work by providing laboratory for running the experiment.

Authors' contributions

BSR developed the methodology and designed the experiment. BSR also analysed the result and wrote the manuscript under the supervision of RS and CF as academic supervisors. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The raw dataset used in this study is publicly available and the source is included in the manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. [2] Department of Electrical Engineering, Universitas Muhammadiyah Jember, Jember, Indonesia.

Received: 2 June 2019 Accepted: 27 August 2019
Published online: 12 September 2019

References

1. Sridhar S, Srinivasan R. Social influence effects in online product ratings. J Mark. 2012;76(5):70–88.
2. Zheng L, Wang H, Gao S. Sentimental feature selection for sentiment analysis of Chinese online reviews. Int J Mach Learn Cybern. 2018;9:75–84.
3. Rintyarna BS, Sarno R, Fatichah C. Enhancing the performance of sentiment analysis task on product reviews by handling both local and global context. Int J Inf Decis Sci; 2018 **(in press)**.
4. Budiharto W, Meiliana M. Prediction and analysis of Indonesia presidential election from Twitter using sentiment analysis. J Big Data. 2018;5:1–10.
5. Hsieh WTM. eWOM persuasiveness: do eWOM platforms and product type matter ? Electron Commer Res. 2015;57:509–41.

6. Tsao H, Chen M. The asymmetric effect of review valence on numerical rating: a viewpoint from a sentiment analysis of users of TripAdvisor. 2019;43(2):283–300.
7. Saad S, Saberi B. Sentiment analysis or opinion mining: a review. Int J Adv Sci Eng Inf Technol. 2017;7(5):1660.
8. Baccianella FSS, Esuli A. SentiwordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the 9th conference on language resources and evaluation; 2010. p. 2200–4.
9. Taboada M, Brooke J, Tofiloski M. Lexicon-based methods for sentiment analysis. Comput Linguist. 2011;37(September 2010):267–307.
10. Wilson PHT, Wiebe J. Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of human language technology conference and conference on empirical methods in natural language processing. Vancouver, Br. Columbia, Canada; 2005.
11. Qiu G, Liu B, Bu J, Chen C. Opinion word expansion and target extraction through double propagation. Comput Linguist. 2011;37:9–27.
12. Cambria E, Havasi C, Hussain A. SenticNet 2: a semantic and affective resource for opinion mining and sentiment analysis. In: Twenty-fifth international FLAIRS conference; 2012. p. 202–7.
13. Mohammad SM, Turney PD. NRC emotion lexicon. Ottawa: National Research Council; 2013. p. 1–234.
14. Medhat W, Hassan A, Korashy H. Sentiment analysis algorithms and applications: a survey. Ain Shams Eng J. 2014;5(4):1093–113.
15. Staš J, Juhár J, Hládek D. Classification of heterogeneous text data for robust domain-specific language modeling. EURASIP J Audio Speech Music Process. 2014. https://doi.org/10.1186/1687-4722-2014-14.
16. Al Amrani Y, Lazaar M, El Kadiri KE. Random forest and support vector machine based hybrid approach to sentiment analysis. Procedia Comput Sci. 2018;127:511–20.
17. Yin Y, Jin Z. Document sentiment classification based on the word embedding. In: 4th international conference on mechatronics, materials, chemistry and computer engineering; 2015. p. 456–61.
18. Gezici G, Dehkharghani R, Yanikoglu B, Tapucu D, Saygin Y. SU-Sentilab : a classification system for sentiment analysis in Twitter. In: Seventh international workshop on semantic evaluation, vol. 2, no. SemEval; 2013. p. 471–7.
19. Yoo SY, Song JI, Jeong OR. Social media contents based sentiment analysis and prediction system. Expert Syst Appl. 2018;105:102–11.
20. Sinha R, Mihalcea R. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In: International conference on semantic computing (ICSC 2007); 2007. p. 363–9.
21. Pandey H, Mishra AK, Kumar N. Various aspects of sentiment analysis. In: International conference on advanced computing and software engineering; 2019.
22. Vyas V, Uma V. Approaches to sentiment analysis on product reviews. In: Sentiment analysis and knowledge discovery in contemporary business, IGI Global; 2019. p. 15–30.
23. Tang F, Fu L, Yao B, Xu W. Aspect based fine-grained sentiment analysis for online reviews. Inf Sci. 2019;488:190–204.
24. Jo Y, Oh A. Aspect and sentiment unification model for online review analysis. In: Proceedings of the fourth ACM international conference on Web search and data mining; 2011. p. 815–24.
25. García-Pablos A, Cuadros M, Rigau G. W2VLDA: almost unsupervised system for aspect based sentiment analysis. Expert Syst Appl. 2018;91:127–37.
26. Ganu G, Elhadad N, Marian A. Beyond the stars : improving rating predictions using review text content. In: Proceeding of WebDB, no. 9; 2009. p. 1–6.
27. Pontiki M, et al. "SemEval-2016 task 5 : aspect based sentiment analysis. In: Proceedings of the tenth international workshop on semantic evaluation (Se-meval-2016); 2016. p. 19–30.
28. Amplayo RK, Lee S, Song M. Incorporating product description to sentiment topic models for improved aspect-based sentiment analysis. Inf Sci. 2018;454:200–15.
29. Rana TA, Cheah Y. A two-fold rule-based model for aspect extraction. Expert Syst Appl. 2017;89:273–85.
30. Rana TA, Cheah YN. Exploiting sequential patterns to detect objective aspects from online reviews. In: International conference on advanced informatics: concepts, theory and application; 2016.
31. Rintyarna BS, Sarno R, Yuananda AL. Automatic ranking system of university based on technology readiness level using LDA-Adaboost.MH. In: 2018 international conference on information and communications technology (ICOIACT), vol. 2018; 2018. p. 495–9.
32. Miller GA. WordNet: a lexical database for english. Commun ACM. 1995;38(11):39–41.
33. Aliyanto D, Sarno R, Rintyarna BS. Supervised probabilistic latent semantic analysis (sPLSA) for estimating technology readiness level. In: International conference on information & communication technology and system; 2017. p. 79–84.
34. Banerjee S, Pedersen T. An adapted lesk algorithm for word sense disambiguation using WordNet. Comput Linguist Intell Text Process. 2002;2276:136–45.
35. Saif H, He Y, Fernandez M, Alani H. Contextual semantics for sentiment analysis of Twitter. Inf Process Manag. 2016;52(1):5–19.
36. Wu Z, Palmer M. Verb semantics and lexical Zhibiao W u. In: Proceedings of the 32nd annual meeting of the association for computational linguistics; 1994. p. 133–8.
37. Saif H, He Y, Fernandez M, Alani H. Contextual semantics for sentiment analysis of Twitter. Inf Process Manag. 2014;52(1):5–19.
38. Hall M, et al. The WEKA data mining software: an update, vol. 11, no. 1, p. 10–8.
39. McAuley J, Pandey R, Leskovec J. Inferring networks of substitutable and complementary products. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. 2015; p. 785–94.

Publisher's Note