Neural Computing and Applications
https://doi.org/10.1007/s00521-019-04105-z

ORIGINAL ARTICLE

Aspect-based sentiment analysis using deep networks and stochastic optimization

Ravindra Kumar[1] · Husanbir Singh Pannu[1] · Avleen Kaur Malhi[1]

Received: 27 November 2017 / Accepted: 16 February 2019

Abstract
Sentiment analysis, also known as opinion mining, is a computational study of unstructured textual information which is used to analyze a persons attitude from a piece of text. This paper proposes an efficient method for sentiment analysis by effectively combining three procedures: (a) creating the ontologies for extraction of semantic features (b) Word2vec for conversion of processed corpus (c) convolutional neural network (CNN) for opinion mining. For CNN parameter tuning, a multi-objective function is solved for nondominant Pareto front optimal values using particle swarm optimization. Experiments show that the proposed technique outperforms other state-of-the-art techniques while yielding 88.52%, 94.30%, 85.63% and 86.03% in accuracy, precision, recall and *F*-measure, respectively.

Keywords Sentiment analysis · Convolutional neural networks · PSO · Multi-objective function

1 Introduction

Sentiment analysis is always a challenging task in the field of natural language processing. It is the process of finding opinion attached with a piece of text by evaluating whether it is positive, negative or neutral. It also finds the degree of polarity (high, mild or moderate) well known as opinion mining. It analyzes the feeling, thought, attitude from collecting opinion as review on various Web sites. Opinion mining can be done mainly by two approaches, first one is machine learning approach and the second one is a lexicon-based approach. Further machine learning-based approach is divided into two parts, supervised and unsupervised. There are three ways to perform sentiment analysis. First one is document level. The least demanding of the three specified methodologies is unquestionably the first, and this concept is followed by most of current opinion mining approaches right now (essentially grouping documents or posts like tweets as positive or negative). It does not consider the diverse points of the archive; however, it can have moderately great outcomes particularly in cases, where we are intrigued just in discovering which records a particular client needs to peruse [1]. At sentence level, the sentence-level grouping considers each sentence as a different unit and expect that sentence ought to contain just a single supposition. Sentence-level examination has two undertakings as subjectivity characterization and estimation grouping. In phrase level, also known as aspect level aspect. The objective of performing classification at feature level is to create a component-based perspective outline of different surveys. It has basically three undertakings. The primary assignment is to recognize and extricate question includes that have been remarked on by a conclusion holder (e.g., picture, battery life). The second job is to decide the extremity of viewpoints on different classes of features: positive, negative and impartial. The third job is identified by gathering words equivalent to features. The contents present on the Internet in the form of literature, review, blogs are increasing day by day. These data are available as structured data and unstructured data. Structured data comprise of numeric data, transactional data and information that is gathered and stored in well mannered form by enterprises. Further, such data are used to access,

✉ Avleen Kaur Malhi
avleen@thapar.edu

Ravindra Kumar
ravi.sliet@gmail.com

Husanbir Singh Pannu
hspannu@thapar.edu

[1] Computer Science and Engineering, Thapar University, Patiala 147004, India

perform query so as to help in taking important decisions by organizations. Unstructured data include data that are present in the form of text documents, PDF files, SMS, e-mail, customer comments, review about an entity, social post, audio and video. Aspect-based sentiment analysis takes the reviews and learns the aspects present inside it. For every aspect, score is provided and on the basis of these scores, the polarities for complete sentence are calculated. After this, reviews are classified as training and testing and then the model is trained. Training of the model makes it capable to find out the polarity of new review. The working model of the proposed technique is shown in Fig. 1 where it depicts the training and testing modules of text which are given to machine learning algorithm based on which sentiment analysis is done.

Main difficulty lies in the extraction of emotions, structure of text, from unstructured data, i.e., image or text, the language used on Internet for communication varies with every individual or status to status. So a systematic approach must be framed for opinion mining involving data preprocessing and text classification. The available tools used for sentiment analysis are shown in Table 1.

1.1 Motivation

This paper proposes an efficient method for sentiment analysis by:

1. A systematic approach has been proposed for extracting features from unstructured data of hotel reviews collected from booking.com.
2. The ontologies have been created for extraction of semantic features.
3. The convolutional neural network (CNN) has been used for opinion mining.
4. CNN parameter tuning has been done using a multi-objective function using particle swarm optimization (PSO).

Rest of the paper is organized as follows: Sect. 2 gives the literature review of the subject and Sect. 3 details the background related to the proposed technique. Section 4 gives the proposed technique with result analysis done in Sect. 5. Finally, Sect. 6 concludes the paper by giving its future scope as well.

2 Literature review

A summarized literature survey of the research work done in the domain of sentiment analysis is represented. Efficiency for a system is achieved by using data structure optimization, query technique optimization and parallel processing optimization. Since a designed model perform well on capacity concurrent users, response rate and expandability. Al-molsmi et al. [2] presented a review on sentiment analysis in cross-domain since sentiment analysis has received a lot of attention in recent years. Thus, a detailed overview of the techniques, methods and approaches that can be used for sentiment analysis in cross-domain is presented by giving the comprehensive literature. A comparative study was also conducted by Jianqiang et al. [3] for the analysis of preprocessing methods used for twitter analysis. The accuracy and F1-measure of twitter classification classifier are improved by using the appropriate preprocessing method.

Wei et al. [4] designed a system based on ontology to analyze the product review. A sentiment ontology tree is formulated to represent the knowledge in hierarchical relation of product features and sentiment attached to it. The analysis of human labeled data assures to provide better accuracy. The future work of this model is to automate the feature or attribute extraction. It can reduce the manual effort, but there is probability that a bit of accuracy may be compromised. Kontopoulos et al. [5] proposed the implementation of real ontology-based deployment

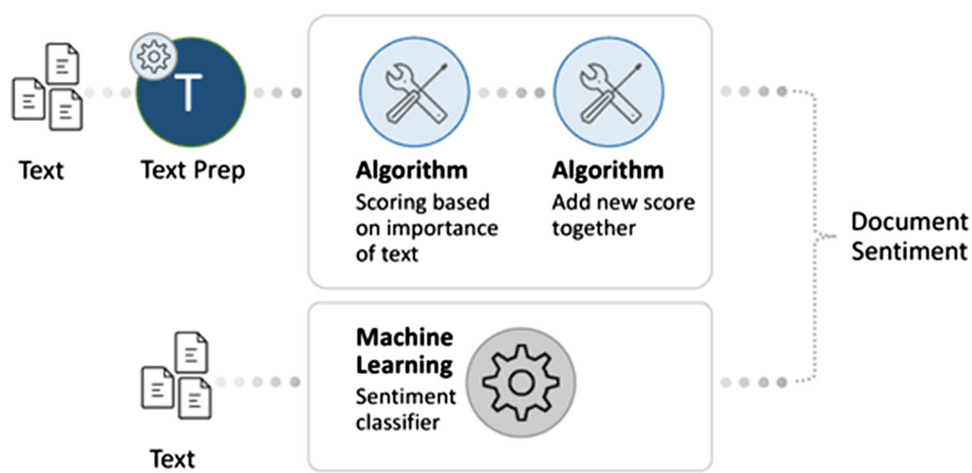


Fig. 1 An aspect-based sentiment analysis model

Table 1 List of available tools to perform sentiment analysis

Tools	Uses
STANFORD CORENLP	Sentiment analysis, bootstrapped pattern learning, POS tagging, named entity recognizer, parsing, coreference resolution system
WEKA	Classification, regression, clustering, association rules, visualization, data preprocessing, machine learning, algorithm for data mining
NLTK	Tokenization provides lexical resources such as WordNet classification, parsing, semantic reasoning, stemming, tagging
APACHE OPENNLP	Parsing, coreference resolution, tokenization, part-of-speech tagging, chunking, sentence segmentation
LING PIPE	Classification entity extraction, POS tagging, clustering

technique for more effective sentiment analysis of twitter tweets. The proposed model uses the ontology technique in more efficient manner to analyze the twitter not only by giving the score for complete tweets but analyzes every hidden aspects of tweets then provides score for a particular tweet that makes feature engineering process more accurate for achieving higher level of accuracy. Ciric et al. [6] proposed a framework for sentiment analysis of twitter tweets. In the proposed work, multiple machine learning models are used to perform opinion mining and their assembling is done to provide better results. Obtained results are compared with different approaches for providing detailed analysis. Behrainian et al. [7] proposed a hybrid model for target-based sentiment analysis of twitter tweets. It is shown that hybrid approach outperforms other approaches and also shows higher performance with various features and functionaries.

Freitas et al. [8] proposed a model in which feature extraction is done by using ontology technique and data sources are *moviereview* and *hotelreview* datasets. Sentiment analysis by using ontology is highly effective in their experiments and achieved high performance and accuracy. Zhao et al. [9] proposed an approach to work on Chinese language and achieved a better performance than earlier models. It achieved higher accuracy. Bakliwal et al. [10] implemented a model in which 3-class sentiment classification was done on Iris General electronics tweets. Using supervised learning approach and subjectivity lexicon-based score, it achieved accuracy of 61.6%. Sam et al. [11] work concerned about generalized model that analyzes the customer reviews about the electronics products available on social networking Web sites. The extraction of keyword and the ontologies designing of electronics product deals in understanding the behavior of online customers. A survey of the techniques used for opinion mining have been discussed by Ravi et al. [12] and Yadollahi et al. [13]. Sentiment analysis has also been used in human–agent interaction by Clavel et al. [14] since opinion mining has been rarely used before for human–agent interaction. A new commonsense ontology is being created for sentiment analysis by Dragoni et al. [15].

Kim [16] reported a series of experiments with convolutional neural network (CNN) trained on top of pretrained word vector for sentence-level classification tasks. In this, a simple CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks. It is implemented with single convolutional layer architecture. Stojanvaski et al. [17] proposed a model by using the deep convolutional neural network. In this work, pretrained word vector embedding is obtained by implementing unsupervised learning on large corpora. The dataset used belongs to SemEval 2015 and result supports the Twitter SemEval 2015 benchmark and no handcrafted features were utilized in this framework. The measuring F1 score was 64.85%. Future Scope for this framework is this whole framework can be reconstructed by using twitter-based corpora that can enhance the performance of discussed model.

Ouyang et al. [18] proposed a framework consisting of Word2Vec and CNN. In this architecture, 3 pairs of Convolutional layer and max-pooling layer were used. This was the first time that seven layer framework model was applied using word2vec and CNN to find out the sentiment of the sentence. Parametric rectifier linear unit with normalization and dropout technology. Publicly available *moviereview* data were used with five different labels negative, somewhat negative, neutral, somewhat positive and positive. The network had acquired test accuracy of 45.4%. Jindal et al. [19] designed a system on image, sentiment identification framework was built with a deep convolutional neural network. This framework was retrained on large data for object recognition and transfer learning. The dataset used was Flicker labeled images. Yang et al. [20] presented a different neural network model that using a convolutional neural network with tree bank information for performing sentiment analysis task. This model uses a number of aspects like syntax information, structure information that works better than the other single aspect model. Static word embedding is used which can be improvised by defining what kinds of sentence architecture can express the opinion in a better manner. It is also mentioned that the prefix and suffix knowledge for the

Table 2 Sentiment analysis comparison based on ontology and CNN

Article name	Authors	Approach	Accuracy
Sentiment learning on product reviews via sentiment ontology tree [5]	Wei Wei, Jon Atle Gulla	HL (Hierarchical Learning)-SOT (sentiment ontology tree) approach	NA
Sentiment analysis and classification based on textual reviews [28]	Ms.K. Mouthami, Ms.K. Nirmala Devi, Dr.V.Murali Bhaskaran	Sentiment fuzzy classification algorithm with parts of speech tags	NA
Sentiment analysis using convolutional neural network [18]	Xi Ouyang, Pan Zhou, Cheng Hua Li, Lijun Liu	A 7-layers architecture CNN model is applied using word2vec to analyze sentences	45.5% accuracy
Image sentiment analysis using deep convolutional neural networks with domain-specific fine tuning [19]	Stuti Jindal and Sanjay Singh	An image sentiment prediction framework is built with convolutional neural networks (CNN) pretrained on a large scaling data and CNN for object recognition pretrained on a large scaling data and CNN for object recognition	53.5% accuracy
Tb-CNN: Joint tree bank information for sentiment analysis using CNN [20]	Tao Yang, Yang Li, Quan Pan, Lantian Guo	Convolution neural network with the tree bank information	94.7% (Binary classification accuracy), 49.9% (Fine grained accuracy)
Twitter sentiment analysis using deep convolutional neural network [17]	Dario Stojanovski, Gjorgji Strezoski, Gjorgji Madjarov and Ivica Dimitrovski	CNN with multiple filters with varying window sizes, 2 fully connected layers with dropout and a softmax layer	F1 score 64.85%

sentiment extraction can also be given. Bouazizi et al. [21] designed a framework to do sentiment analysis of twitter tweets with specific capability of detecting sarcastic statements that enhance the performance of opinion mining. Its capability to detect sarcastic tweets results as better accuracy level.

Abbasi et al. [22] proposed a sentiment analysis on multiple languages for exchange of information on Internet. The sentiment analysis of English and Arabic language was done by extracting the specific feature components by utilizing stylistic and syntactic features. The performance results depicted higher accuracy of over 91%. Another article is proposed by Valdiva et al. [23] for performing sentiment analysis on tripadvisor. The match between the user sentiments and automatic sentiment detection algorithms is studied for analysis by also discussing the challenges. Che et al. [24] proposed a technique for aspect-based sentiment analysis by using sentence compression. Discriminative conditional random field model is applied for automatically compressing the sentiment sentences by which the performance of the aspect-based sentiment analysis is significantly improved. Bui et al. [25] given a study on cancer survivor network for temporal causality analysis of change in sentiments. It is examined by introducing a novel framework for cancer survivor network. Machine learning is used to train sentiment classifier over posts which are tagged manually with sentiment labels for classification of sentiment post as positive or negative. Clavel et al. [14] employed the sentiment detection methods on human–agent interaction in a similar manner as they have been applied in opinion mining. Then, the various possibilities for mutual benefit are proposed and the proposed general guidelines have also been applied in some specific perspective which can be used for sentiment analysis of human–agent interaction more effectively. Wu et al. [26] adduced a sentiment analysis approach for decision making in online stock forum facilitating decision making of investor and perception of stock companies. The result analysis showed the stronger effect of investor sentiment on value stocks. Another sentiment analysis system has been designed to facilitate the teaching and learning system by Rani et al. [27]. It works by analyzing the comments of students on various courses and online sources for identifying sentiments and emotions. The comparative study with direct assessment tools depicts the reliability of the system. The comparison of the state-of-the-art techniques has been given in Table 2.

3 Preliminaries

The proposed work is performed in the phases of data collection, preprocessing, ontology creation, Word2vec representation and CNN implementation. These phases are shown in Fig. 2.

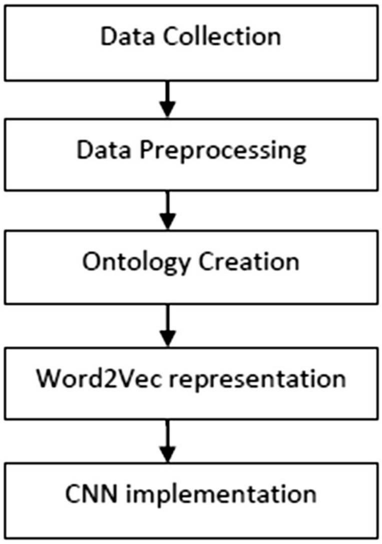


Fig. 2 Phases of aspect-based sentiment analysis

3.1 Data collection

In this phase, reviews are collected by using web scrapping technique. With the help of scrapy, large numbers of hotel reviews were fetched from booking.com. These reviews were already divided into two parts, positive and negative contents. It makes the task simple and helps to easily discern different aspects with relative positive and negative views.

3.2 Preprocessing

In this phase, reviews collected as raw data format are converted into processed informational data. Raw reviews as shown in Fig. 3 are collected using scrapy contains empty rows and empty cells. The preprocessing steps are shown in Fig. 4. By using Panda library, these data are cleaned and only the useful data instances are retained. The hotel reviews are used after they were analyzed manually for labels in our experiment during the development of ontology. Ontology is designed by parsing the reviews and final dataset is prepared by combining positive and negative part of reviews with the help of SPARQL query language which is executed on the prepared hotel ontology.

The goal of ontology is to facilitate knowledge about a particular domain in such a format that can be easily understood by developers as well as machines as shown in Fig. 5. Ontology is created by using positive and negative content of a particular review and score is provided for different aspects of hotel reviews [5]. For the requirement satisfaction, a strategy has been developed for software applications while utilizing a web technology. Hotel ontology is created as shown in Fig. 6, and the web-based system can also avail for some audit outcomes.

negative_content	positive_content	score	tags	title
The rooms showed there	Upon arrival we checked into our two-	10		Large rooms, perfect for family trips!
The problem is that the front desk is aloof and can be rude.They		6.3		The problem is that the front desk is aloof and can be rude.
Additional charges. Taxes	Bathroom, and the bed is extremely cc	5		Poor value for money.
very dirty wallpapers in th	nice for large group gathering (in the s	7.9		Good stay for Times square/Central park area
The restaurant was extrer	The location was really the only positi	5		A bit disappointing...
The carpets everywhere a	The large and clear room, the double s	7.1		Fine but will not return.
The entrance hall needs a	Comfortable quite clean	7.5		Comfortable quite clean
I do wish that they offere	I have been staying at the London for t	9.6		The London is spectacular! like a vacation even though I'm working
The staff was not up to pa	24 hour room service.	6.3		24 hour room service.
Shabby room. Booked the	View amazing. Comfortable beds and I	5		Great location - great square footage for NYC - needs some tlc
I had many issues at the r	The hotel location is amazing and very	7.5		Comfortable Rooms ..
Nothing	Large, beautiful and modern apartmen	10		Large, beautiful and modern apartments, excellent service...
No view whatsoever.	Overall, very good. Only reason why c	7.5		Great location!, but will try other hotels next time.

Fig. 3 Raw reviews

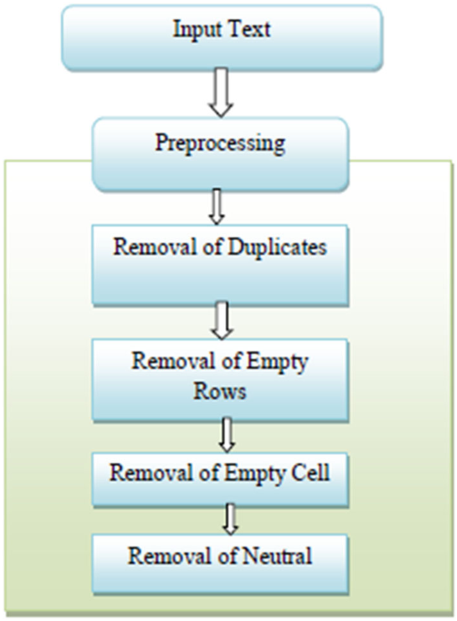


Fig. 4 The preprocessing steps

3.3 Dataset

Data are collected by using web scrapping with Python followed by preprocessing to extract valuable data from raw dataset. Afterward, this ontology model is prepared, data are fed into the classifier model and semantic features are extracted from the given domain. Overall score is calculated on the basis of initial features extracted from the ontology model. Word2vec is created for given processed corpus by using unsupervised neural network. Finally, a vector form of the used corpus is trained through the CNN along with tenfold cross-validation.

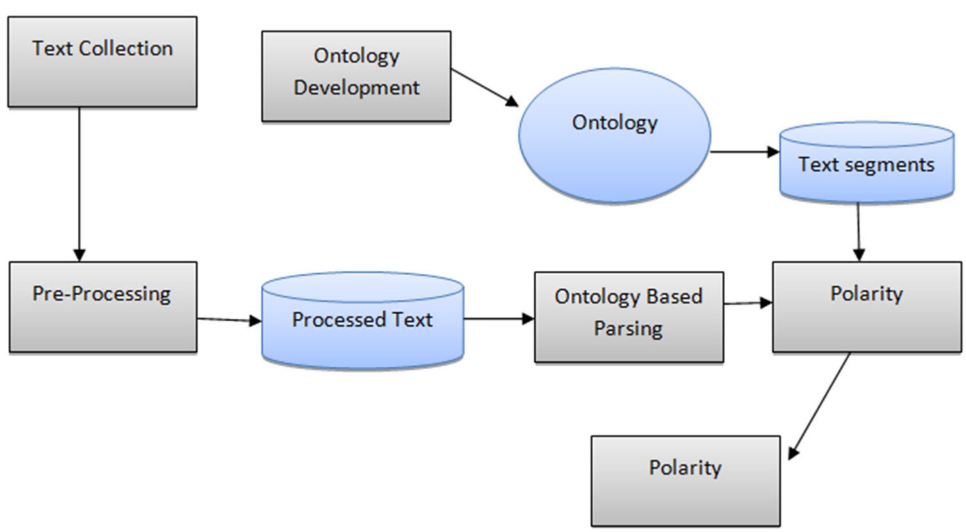
3.4 Convolutional neural network (CNN)

Architecture of CNN is shown in Fig. 7. It is formed by the distinct layer stack which transforms the input volume to the target output (class score holding) through a differential function. Constituent layers of CNN are:

1. Convolution layer
2. Max-pooling layer
3. ReLU layer
4. Back propagation layer

The idea is to use enough filters (128 in this case) so that they can capture a lot of features in the given sentence. As in image classification, various filters collect various attributes such as edges, density of the colors at various areas, turning certain areas into black and white, etc. Text classification problem extends the similar concept through various filters to capture features such as *like* means positive instead of similarity, *very much* expresses the degree of the expression using filter with size 2. The basic idea is to set up enough number of features to capture all possible descriptors of the text. Maxpool is to get the maximum output vector value upon the application of the filter. It selects a strongest expression element from the output in the extracted feature and nothing to do with the length of the word. Each sample is expressed as $n \times 1$ where n is some length expressing the dimension. The filter is applied as a drifting window, for example 3×1 filter on the sentence *I like the movie very much !* would yield {I like this, like this movie, this movie very, movie very much, very much !}. Sentences get padded to equal length before the embeddings take place. Thus all filters does not result in the identical dimensional outcomes. The region sizes (2, 3, 4) is similar to 2, 3, 4-g word and the first filter in this trigram will give various weight values to various words in trigrams. It means one filter may assign

Fig. 5 Ontology architecture



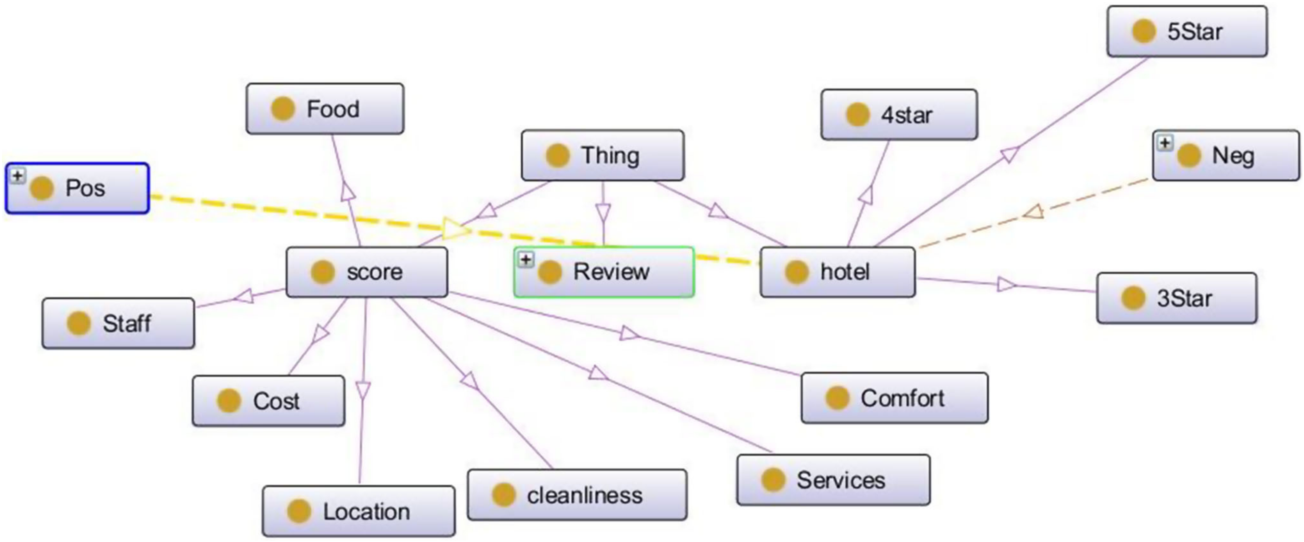
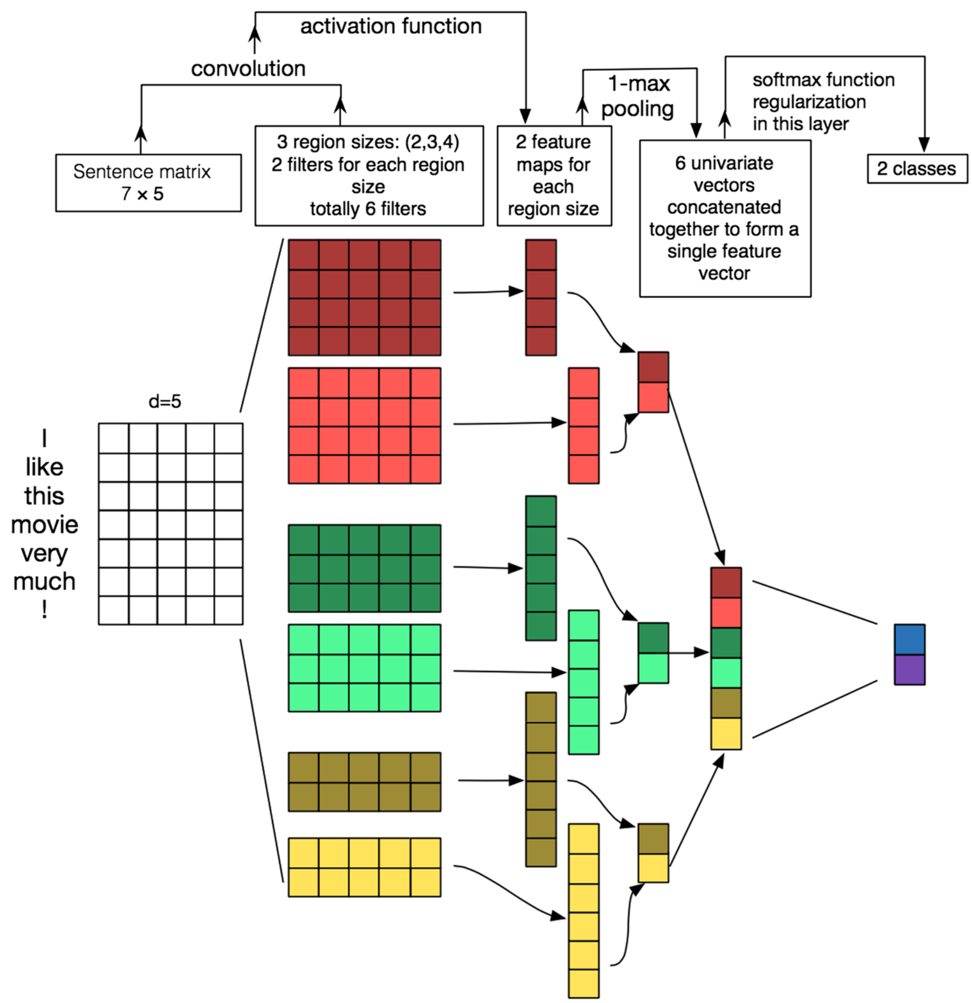


Fig. 6 Ontology for positive and negative contents respective to subjective aspects

Fig. 7 Convolutional neural network [29]



relatively higher weight to first index (0-index) and lower to the second. So 128 filters will assign different weights which would be learned to optimal weight values over time for correct prediction.

For a given sentence, a sentiment label is created by analyzing its score. For scoring a sentence, the words sequence in the sentence is considered as input in a network and it passes through layers sequence which extract its features with increased level of complexity. The feature extraction of network can be performed on the sentence level and character level. The network architecture novelty is to include two convolutional layers that allows to handle any size sentence and words.

4 Proposed technique

Data are collected by using web scrapping using Python followed by its preprocessing. Afterward, an ontology model is prepared and data are fed into the classifier model. The semantic features are extracted from the given domain, and overall score is calculated on the basis of initial features extracted from the ontology model. Word2vec is then created for the given processed corpus by using unsupervised neural network. Finally, a vector form of used corpus is trained through the CNN classifier. Then classification is performed and tenfold cross-validation is employed for better model generalization. Development stages of sentiment analysis model are shown in Fig. 8. CNN tuning is performed using PSO as discussed in the subsection below.

4.1 Feature extraction

First layer of the network converts words into feature vectors of real-value (embedded) which captures syntactic, semantic and morphological information about words. The word vocabulary is of fixed size v^{word}, and words comprise of characters from character vocabulary of fixed-size vchar. Given sentence comprises of n words $\{W_1, W_2, \ldots W_n\}$, and converting the W_n (each word) into vector $V_n = [r^{word}, r^{wchar}]$, that consist of two sub-vectors: the embedding word level $r^{word} \in R^{d^{word}}$ and the embedding character level $r^{wchar} \in R^{d_n^0}$. The embedding of word level are meant for capturing semantic and syntactic information and embedding of character level are meant for capturing shape and morphological information.

4.2 Word level embedding

Column vectors encode the word level embedding in an embedding matrix $w^{word} \in R^{d^{word}} \times v^{word}$. The embedding of word level corresponds to each column $w_i^{word} \in R^{d^{word}}$ from which the ith vocabulary word is generated. A word W transforms into embedding of word level by utilizing product of matrix-vector:

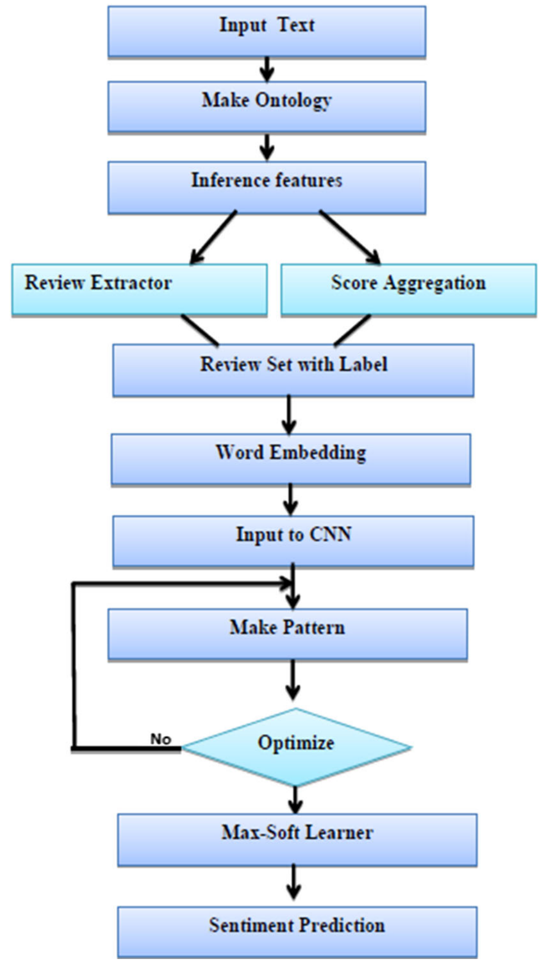


Fig. 8 Proposed model

$$r^{word} = v^w w^{word} \tag{1}$$

where $v^w \leftarrow$ vector size of $|vword|$ have 1 value at index w and 0 at other positions. The w^{word} matrix is a parameter for learning, and the embedding word level size *dword* is the user choosing the hyperparameter. In this paper, Word2vec is used to perform the word level embedding.

4.3 Word2Vec

Word2vec is a tool provided as open source by Google in 2013 under the license of Appache License 2.0. It extracts features from a given text corpus without any intervention or help needs from a human expert. Most importantly, it will perform quite well even if the text size is too small or just an individual word. By giving a large corpus context and uses Word2vec makes an appropriate words meaning and it also works fast with big datasets. In deep learning, the meaning of words is one of the most important aspects that is completely fulfilled with use Word2vec for classifying larger entities [18]. In the proposed technique,

Fig. 9 Similarity representation [18]

bad	terrible	horrible	lousy	crummy	horrid
	0.682861	0.670260	0.664764	0.567782	0.565169
good	great	decent	nice	excellent	fantastic
	0.729151	0.683735	0.683609	0.644293	0.640778
awful	horrible	terrible	dreadful	horrendous	horrid
	0.759767	0.747892	0.721818	0.697003	0.672018
beautiful	gorgeous	lovely	wonderful	fabulous	loveliest
	0.835300	0.810064	0.685409	0.670007	0.661258
terrible	horrible	horrendous	dreadful	awful	horrid
	0.924392	0.846727	0.802277	0.747891	0.717903
fantastic	wonderful	great	amazing	marvelous	fabulous
	0.804792	0.793521	0.778987	0.768760	0.760597

vectors are trained by using Google News dataset (about 100 billion words). This vector is available on the URL https://code.google.com/p/word2vec/. The model contains 3 million combinations of words and phrases with 300 dimensional representations. It is possible to achieve a precise relation by using such a huge corpus. Figure 9 explains the similarity relationship among words by using Word2vec conversion.

In Fig. 10, diffusion network is being represented to visualize the relationship among different words in the Google news dataset. The edge width represents similarity and node represents the word. Now, it easily describes the similarity such that words with same sentiment labels have the same vector.

4.4 Character-level embedding

The robust methods for extracting the shape and morphological information from words are taken into all characters

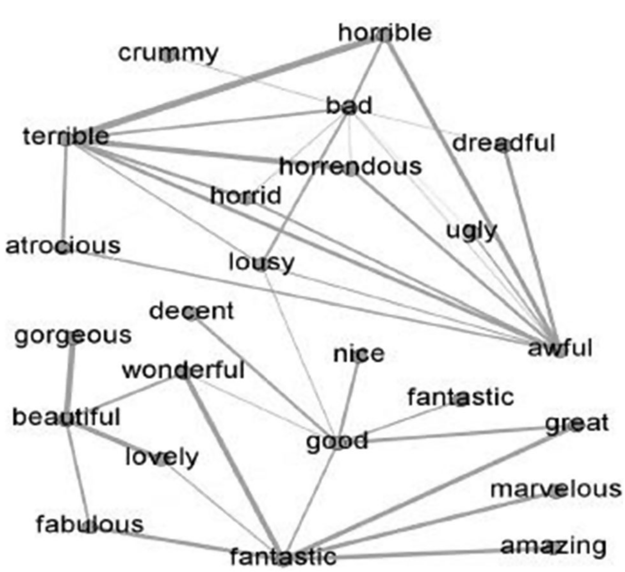


Fig. 10 Diffusion network of Word2Vec [18]

consideration of the words which select the important features for the classification phase. For instance, in the sentimental analysis task of twitter data, and appearing important information in different hash tag parts (e.g., *#ilikeit*) and various adverb information endings having suffix *ly* (e.g., *badly*). The local features are produced by convolutional approach around word's each character and further max operation are utilized for combining them for creating embedding character-level of fixed-size word. Given words are composed of m character $\{k_1, k_2, \ldots, k_m\}$, each character k_m is transformed into an embedding character r_m^{char}. Column vectors encode the embedding character in the matrix embedding $w^{char} \in |v^{char}|$. Given k characters, the matrix-vector product obtains its embedding r^{char}:

$$r^{char} = v^k w^{char} \tag{2}$$

where v^k vector size of $|v^{char}|$ having 1 value at k index and 0 in other positions. The convolution layer input is the embedding character sequence $\{r_1^{char}, r_2^{char}, \ldots, r_n^{char}\}$. The matrix vector operation is applied by convolution layer for each window having size $k^c har$ of the sequence of successive windows. Consider the defined vector as $x^m \in r^{char} R^{d^{char}}$ which is the embedding character concatenation n, having its left neighbors $(r^{char} - 1)/2$, and its right neighbors $(r^{char} - 1)/2$:

$$x^n = \left(k^{char}_{n-\frac{r^{char}-1}{2}}, \ldots k^{char}_{n+\frac{r^{char}-1}{2}}\right)^T \tag{3}$$

The vector $k^{char} \in R^{c_u^o}$ having jth element computed by convolutional layer that is the w embedding character level as follows:

$$[k^{wchar}]_j = max_{1<n<N}[w^o x_n + a^0]_j \tag{4}$$

$w^o \in R^{c_u^o \times d^{char} r^{char}} \leftarrow$ [Convolution layer weight matrix]. Then, local features are extracted using same matrix around each window character in a given word.

4.5 Sentence-level analysis

A sentence x is given with m words $\{w^1, w^2, \ldots w^m\}$ which is then converted to the word level joints and embedding level character $\{u^1, u^2, .., u^n\}$ SCNNchar. Next step comprises of representation of sentence-level extraction $r_x st$. Sentence-wide extraction of feature set method deals with two major issues:

1. Different sizes of sentences.
2. Any position in the sentence may contain the important information.

These issues are tackled with the use of convolutional layer for computing the feature vector of wide sentence r^{st}. In the CNN architecture, second convolutional layer operates in a similar way as the layer used for character-level feature extraction. Local features are produced in this layer around each word, and they are combined after utilizing max operation for creating feature vectors of fixed size for the sentence. A matrix-vector operation is applied by second convolution layer for each window size of k^{word}in successive window sequence $\{u^1, u^2, \ldots, u^n\}$. The vector $x_m \in R^{(d^{word}+c_u^o)k^{word}}$ does the concatenation of embedding k^{word} in nth word centralization as follows:

$$x_m = (u_a, \ldots, u_b) \tag{5}$$

where $a = m - \frac{u^{word}-1}{2}, b = m + \frac{u^{word}-1}{2}$. The jth element is computed by convolutional layer having vector $r^{st} \in R^{c^{1u}}$ as follows:

$$[r^{st}]_j = 1 < m < M max[w^1_{x_m} + a^1]_j \tag{6}$$

where $w^1 \in R^{c_u^1} \times (d^{word} + c_u^0)k^{word} \leftarrow$ [Convolutional layer weight matrix]. Finally, r_x^{st} vector having the feature vector global relative to x sentence is processed with two neural network layers. They extract one more representation level and each sentiment label $\tau \in T$ score is then computed:

$$S(x) = w^3 h(w^2 r_x^{st} + a^2) + a^3 \tag{7}$$

where learning parameter matrix and vectors are $w^2 \in R^{h_u \times c_u^1}$, and $w^3 \in R^{|T| \times h_u}$, h(.)$\leftarrow$ Hyperbolic tangent.

4.6 Negative training

A negative probability is minimized in the network training over training set A. A sentence x is given having parameter set θ in the network is computing each sentiment label $\tau \in T$ Score $S_\theta(x)_T$ These scores are transformed into given labels with conditional probability distribution of the sentence and network parameter set θ and applied a softmax operation over $\tau \in T$ score:

$$P(\tau|x, \theta) = \frac{e^{S_\theta(x)_i}}{\sum_{\forall i \in T} e^{S_\theta(x)_i}} \tag{8}$$

The stochastic Gradient descent (SDG) is considered for minimizing the negative log gradient:

$$\theta \longrightarrow \sum_{(u,v)\in A} - log P(v|x, \theta) \tag{9}$$

where (u, v) is the sentence corresponding in the corpus training A and v represents a respective label.

4.7 CNN tuning using PSO

A multi-objective cost function is created to maximize accuracy (A), precision (P), recall (R) and F-measure (F) simultaneously as follows:

$$\begin{aligned} \texttt{maximize}\, \mathcal{C} &= w_1 \times A + w_2 \times P + w_3 \times R + w_4 \times F \\ &\texttt{subject to} \sum_i w_i = 1, w_i \in [0, 1]. \end{aligned} \tag{10}$$

This multi-objective function $\mathcal{C}$ is solved for nondominant Pareto optimal value using particle swarm optimization (PSO). Nondominant Pareto front means none of the objective functions can be improved without sacrificing the values of other objective function values.

Algorithm 1: Algorithm for testing of the reviews

Result: Labelled review upon testing
Input - Hotel reviews dataset;
while *While more test data* **do**
  Positive and Negative content of each review is parsed by using Ontology;
  Rating for each predefined Features (Cleaniliness, Services, Location, Cost etc) are extracted from review;
  Summation of ratings of each feature is calculated $\rightarrow \sum$;
  Summation is compared with predefined threshold value $\rightarrow \Delta$;
  if $\sum > \Delta$ **then**
    Review will be labeled with positive tag;
  else
    Review will be labeled with negative tag and disregard neutral labels ;
  end
end

For PSO, the input position vector consists of weight and bias variables of CNN to yield these testing metrics as objective functions. CNN consists of two major layers (a) convolution (b) max-pooling to yield the final full connected layer. The intermediate connections of these layers consists of weights. Now, CNN parameters are encapsulated into the PSO particles in the form of $X = \{W, B\}$ where W, B refer to the weight and bias vectors.

$$W = \{w_l^j\}, B = \{b_l^j\} \tag{11}$$

where l is the layer index for the particle n. PSO working details can be found in [30]. In each PSO iteration a solution moves to a new value by adjusting the velocity variable. The velocity vector gets modified relative to global best and personal best positions (Gbest and Pbest) for each solution particle while solving for the cost function $\mathcal{C}$. The v_i and x_i are velocity and position at current iteration j of the particle i:

$$v_i^{k+1} = w \times v_i^k + C_1 \times r_1(Pbest_i^k - p^j) + C_2 \times r_2(Gbest^k - x_i^j) \tag{12}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{13}$$

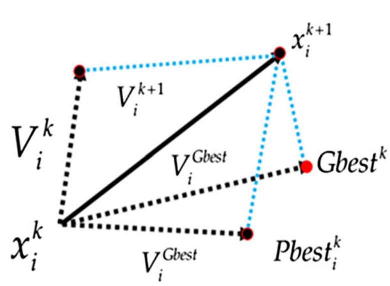


Fig. 11 Position and velocity vector relationship in PSO

C_1, C_2 are acceleration constants for social and cognitive terms, w is inertia weight, $r_1, r_2 \in (0, 1)$ are random numbers. The visual description of PSO is shown in Fig. 11.

5 Results

We have implemented sentiment analysis model by using deep learning. Apart from this, different models has been implemented to show the strength of proposed model relative to conventional methods. Intel Core i5-2450M processor CPU @ 2.50 GHz (3M Cache) and 4 GB RAM. MATLAB 2017a has been used for implementation. The different metrics for model validation are Accuracy, Precision, Recall, F1-Score as defined in Table 3. In the classification model of CNN, 30 epoches have been considered for model validation and generalization. Validation with highest accuracy is eventually used as the model that can be used for testing. In each epoch, the whole data are scanned two times and tenfold validation is implemented.

1. True positive (TP): correctly classified as the class of interest.

Table 3 Performance metrics

Metric	Definition
Accuracy	$(TP + TN)/(TP + TN + FP + FN)$
Precision	$TP/(TP + FP)$
Recall	$TP/(TP + FN)$
F1-score	$(2 \times precision \times recall)/(recall + precision)$

Table 4 Comparative performance results in percentages

Model	Accuracy	Precision	Recall	F-measure
Proposed	8.52	94.30	85.63	86.03
Dictionary based	82.08	92.70	70.81	80.22
SVM	85.56	86.21	85.05	85.11
Random Forest	83.62	86.12	80.61	83.24
Decision Tree	66.69	65.66	71.83	68.54
Max Entropy	81.25	82.24	80.37	81.22
Generalized Linear	82.31	83.5	81.85	84.16
Stabilized LDA	70.36	72.64	66.23	69.22

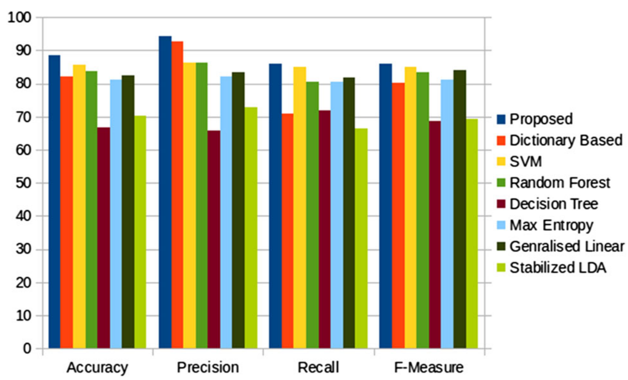


Fig. 12 Comparative analysis of performance metrics

2. True negative (TN): correctly class as not the class of interest.
3. False positive (FP): incorrectly classified as the class of interest.
4. False negative (FN): incorrectly classified as not the class of interest.

In experimental observation, it is concluded that ontology implementation provides highly refined data mining. Information collected by parsing the reviews of specific domain reduces the false-positive and false-negative rate and provides higher accuracy. The comparison of precision, recall and accuracy of different classifier such as SVM, Random Forest, Decision Tree, Maximum Entropy, Generalized Linear Model, Stabilized Discriminant Analysis and CNN has been performed and shown in Table 4 and Fig. 12. It has been found that CNN shows significant improvement of 88.52%, 94.30%, 85.63% and 86.03% in accuracy, precision, recall and F-measure, respectively. The proposed technique has been compared with the dictionary-based approach resulting in higher accuracy rate and f score value for both positive and negative labels in proposed technique as shown in Table 5. The accuracy % and loss % for the proposed method is shown in Fig. 13. The figure depicts that the accuracy % increases with number of iterations for training as well as testing and similarly, loss % decreases with number of iterations. The detailed results of the accuracy %, loss %, time elapsed and base learning rates are depicted in Table 6 for various iteration levels with different epoches. The confusion matrices for SVM, Random Forest, Decision Tree, Maximum Entropy, Generalized Linear Model and Stabilized LDA are shown in Tables 7, 8, 9, 10, 11 and 12, respectively, which depicts the no. of true positives, true negatives, false positives and true negatives for each of the model. The results are compared for all the six classifiers with the proposed classifier on various parameters as shown in Table 13 depicting that the proposed technique outperforms all other classifiers in all testing parameters. Thus, it can be concluded that the proposed method has

Table 5 Comparison of proposed model with dictionary model results

Parameters	Proposed approach	Dictionary model
Data points	1600	1600
Labels	2	2
Accuracy	88.50%	82.69%
Label	NEG	NEG
F-score	89.21%	84.52%
	TP = 761, FP = 145, TN = 655, FN = 39	TP = 756, FP = 233, TN = 567, FN = 44
Label	POS	POS
F-score	87.68%	80.37%
	TP = 655, FP = 39, TN = 761, FN = 145	TP = 567, FP = 44, TN = 756, FN = 233

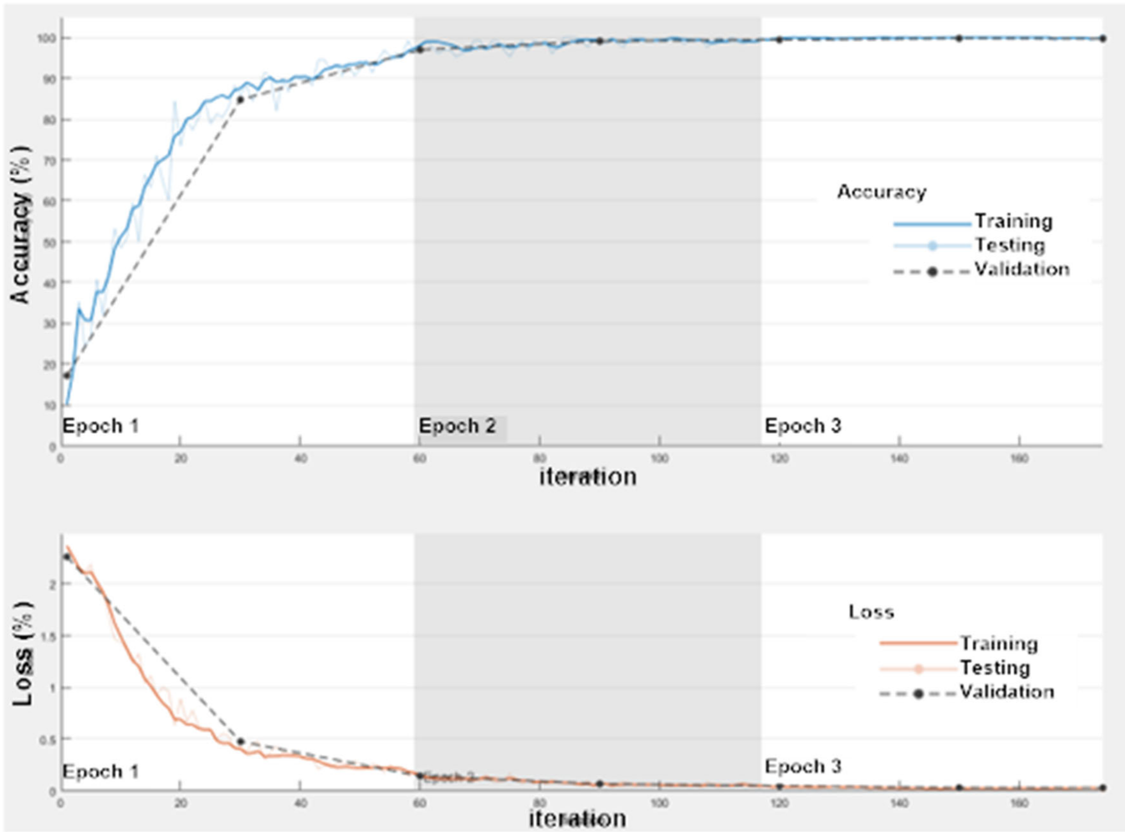


Fig. 13 Accuracy and loss in percentages versus epoches as the graph reaches plateau

Table 6 Detailed results on datasets

Epoch	Iteration	Time elapsed (s)	Loss%	Accuracy %	Base learning rate
1	1	0.53	3.0845	13.28	1.00E−04
1	50	17.73	1.0945	65.63	1.00E−04
2	100	35.09	0.7281	74.22	1.00E−04
3	150	51.84	0.4736	83.59	1.00E−04
4	200	68.18	0.3083	91.41	1.00E−04
5	250	84.73	0.2325	92.97	1.00E−04
6	300	101.14	0.1541	97.66	1.00E−04
7	350	117.58	0.1314	97.66	1.00E−04
7	400	133.86	0.0942	96.09	1.00E−04
8	450	150.22	0.0665	98.44	1.00E−04
9	500	166.66	0.0462	99.22	1.00E−04
10	550	183.83	0.0544	100	1.00E−04
11	600	200.44	0.0658	99.22	1.00E−04
12	650	217.01	0.0338	100	1.00E−04
13	700	233.59	0.034	100	1.00E−04
13	750	250.19	0.037	99.22	1.00E−04
14	800	267.06	0.0264	100	1.00E−04
15	850	284.03	0.0182	100	1.00E−04
15	870	290.62	0.0236	100	1.00E−04

Table 7 SVM confusion matrix results

	1	2
1	690	121
2	110	679

Table 8 Random Forest confusion matrix results

	1	2
1	654	157
2	105	684

Table 9 Decision Tree confusion matrix results

	1	2
1	582	228
2	305	485

Table 10 Maximum Entropy confusion matrix results

	1	2
1	652	159
2	141	648

Table 11 Generalized linear model confusion matrix results

	1	2
1	695	115
2	149	641

Table 12 Stabilized LDA confusion matrix results

	1	2
1	537	273
2	202	588

obtained better accuracy of 88.52% compared to all other six classifiers used for comparison.

Based on experimental setup, ten cross-validation and convolution neural network show high accuracy for particular dataset grouped into positive and negative reviews.

6 Conclusion

This paper has proposed an effective classification method for sentiment analysis using aspects base ontology and CNN. Semantic feature is extracted, and then, various models are trained including the proposed CNN based, SVM, Random Forest, Decision Tree, Maximum Entropy, Generalized Linear Model, Stabilized Discriminant Analysis. Using the ontology-based CNN implementation and information collected by parsing, the reviews of specific domain reduce the false-positive and false-negative rate to yield higher accuracy. For CNN parameter tuning, a multi-objective function is solved for nondominant Pareto front optimal values using particle swarm optimization (PSO). Experiments show that the proposed technique outperforms other state-of-the-art techniques while yielding 88.52%, 94.30%, 85.63% and 86.03% in accuracy, precision, recall and *F*-measure, respectively.

Future plan is to incorporate parallel computing to accelerate the calculations and to explore meta-heuristic relative feature extraction. An ideal work would be to have ontology automation on web-based application to implement sentiment analysis for social Web site.

Table 13 Result details for various methods

Parameters	SVM	RandFor	DecTree	MaxEntr	LinModel	LDA	Proposed
Accuracy	0.8556	0.8362	0.6669	0.8125	0.8350	70.31	88.50
95% CI	(0.84, 0.87)	(0.82, 0.85)	(0.64, 0.69)	(0.79, 0.83)	(0.82, 0.85)	(0.68, 0.72)	(0.87, 0.90)
Information rate	0.5	0.5256	0.5544	0.5044	0.5275	0.5381	.558
*P*0-value	2e−16	2.2e−16	2.2e−16	2.2e−16	2.2e−16	2.2e−16	2.2e−16
Kappa	0.71125	0.6727	0.3328	0.625	0.6698	0.4068	0.7321
Mcnemar's test *P*-value	0.5106	0.001628	0.000995	0.6263	0.0422	0.0013	0.5203
Sensitivity	0.8625	0.8617	0.6561	0.8222	0.8235	0.7267	95.12
Specificity	0.8488	0.8133	0.6802	0.8030	0.8479	0.6829	81.87
Pos Pred value	0.8508	0.8064	0.7185	0.8039	0.8580	0.6630	83.99
Neg Pred value	0.8606	0.8669	0.6139	0.8213	0.8114	0.7443	94.38
Prevalence	0.5000	0.4744	0.5544	0.4956	0.5275	0.4619	0.5000
Detection rate	0.4313	0.4088	0.3638	0.4075	0.4344	0.3356	0.4756
Detection prevalence	0.5069	0.5069	0.5062	0.5069	0.5062	0.5062	0.5069
Balance accuracy	0.8556	0.8375	0.6682	0.8126	0.8357	0.7048	0.8850

Compliance with ethical standards

Conflict of interest There is no conflict of interest involved in this research.

References

1. Antoniou C, Doukas JA, Subrahmanyam A (2012) Sentiment and the CAPM. Available at SSRN 2101724
2. Al-Moslmi T, Omar N, Abdullah S, Albared M (2017) Approaches to cross-domain sentiment analysis: a systematic literature review. IEEE Access 5:16173–16192
3. Jianqiang Z, Xiaolin G (2017) Comparison research on text preprocessing methods on twitter sentiment analysis. IEEE Access 5:2870–2879
4. Wei W, Gulla JA (2010) Sentiment learning on product reviews via sentiment ontology tree. In: Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, pp 404–413
5. Kontopoulos E, Berberidis C, Dergiades T, Bassiliades N (2013) Ontology-based sentiment analysis of twitter posts. Expert Syst Appl 40(10):4065–4074
6. Ciric M, Stanimirovic A, Petrovic N, Stoimenov L (2013) Comparison of different algorithms for sentiment classification. In: 2013 11th international conference on telecommunication in modern satellite, cable and broadcasting services (TELSIKS), vol 2. IEEE, pp 567–570
7. Bahrainian S-A, Dengel A (2013) Sentiment analysis and summarization of twitter data. In: 2013 IEEE 16th international conference on computational science and engineering (CSE). IEEE, pp 227–234
8. Freitas LA, Vieira R (2013) Ontology based feature level opinion mining for Portuguese reviews. In: Proceedings of the 22nd international conference on World Wide Web. ACM, pp 367–370
9. Zhao Y, Niu K, He Z, Lin J, Wang X (2013) Text sentiment analysis algorithm optimization and platform development in social network. In: 2013 sixth international symposium on computational intelligence and design (ISCID), vol 1. IEEE, pp 410–413
10. Bakliwal A, Foster J, van der Puil J, O'Brien R, Tounsi L, Hughes M (2013) Sentiment analysis of political tweets: towards an accurate classifier. Association for Computational Linguistics, pp 49–58
11. Sam KM, Chatwin C (2013) Ontology-based sentiment analysis model of customer reviews for electronic products. Int J e-Educ e-Bus e-Manag e-Learn 3(6):477
12. Ravi K, Ravi V (2015) A survey on opinion mining and sentiment analysis: tasks, approaches and applications. Knowl Based Syst 89:14–46
13. Yadollahi A, Shahraki AG, Zaiane OR (2017) Current state of text sentiment analysis from opinion to emotion mining. ACM Comput Surv: CSUR 50(2):25
14. Clavel C, Callejas Z (2016) Sentiment analysis: from opinion mining to human-agent interaction. IEEE Trans Affect Comput 7(1):74–93
15. Dragoni M, Poria S, Cambria E (2018) Ontosenticnet: a commonsense ontology for sentiment analysis. IEEE Intell Syst 33(3):77–85
16. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882
17. Stojanovski D, Strezoski G, Madjarov G, Dimitrovski I (2015) Twitter sentiment analysis using deep convolutional neural network. In: International conference on hybrid artificial intelligence systems. Springer, pp 726–737
18. Ouyang X, Zhou P, Li CH, Liu L (2015) Sentiment analysis using convolutional neural network. In: 2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing (CIT/IUCC/DASC/PICOM). IEEE, pp 2359–2364
19. Jindal S, Singh S (2015) Image sentiment analysis using deep convolutional neural networks with domain specific fine tuning. In: 2015 international conference on information processing (ICIP). IEEE, pp 447–451
20. Yang T, Li Y, Pan Q, Guo L (2016) Tb-CNN: joint tree-bank information for sentiment analysis using CNN. In: 2016 35th Chinese control conference (CCC). IEEE, pp 7042–7044
21. Bouazizi M, Ohtsuki T (2016) Sentiment analysis: from binary to multi-class classification: a pattern-based approach for multi-class sentiment analysis in twitter. In: 2016 IEEE international conference on communications (ICC). IEEE, pp 1–6
22. Abbasi A, Chen H, Salem A (2008) Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. ACM Trans Inf Syst: TOIS 26(3):12
23. Valdivia A, Luzón MV, Herrera F (2017) Sentiment analysis in tripadvisor. IEEE Intell Syst 32(4):72–77
24. Che W, Zhao Y, Guo H, Su Z, Liu T (2015) Sentence compression for aspect-based sentiment analysis. IEEE/ACM Trans Audio Speech Lang Process 23(12):2111–2124
25. Bui N, Yen J, Honavar V (2016) Temporal causality analysis of sentiment change in a cancer survivor network. IEEE Trans Comput Soc Syst 3(2):75–87
26. Wu DD, Zheng L, Olson DL (2014) A decision support approach for online stock forum sentiment analysis. IEEE Trans Syst Man Cybern Syst 44(8):1077–1087
27. Rani S, Kumar P (2017) A sentiment analysis system to improve teaching and learning. Computer 50(5):36–43
28. Mouthami K, Devi K.N, Bhaskaran VM (2013) Sentiment analysis and classification based on textual reviews. In: 2013 international conference on information communication and embedded systems (ICICES). IEEE, pp 271–276
29. Zhang Y, Wallace B (2015) A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820
30. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73