Cross Domain Sentiment Analysis Using Different Machine Learning Techniques

S. Mahalakshmi and E. Sivasankar

Abstract Sentiment analysis is the field of study that focuses on finding effectively the conduct of subjective text by analyzing people's opinions, sentiments, evaluations, attitudes and emotions towards entities. The analysis of data and extracting the opinion word from the data is a challenging task especially when it involves reviews from completely different domains. We perform cross domain sentiment analysis on Amazon product reviews (books, dvd, kitchen appliances, electronics) and TripAdvisor hotel reviews, effectively classify the reviews to positive and negative polarities by applying various preprocessing techniques like Tokenization, POS Tagging, Lemmatization and Stemming which can enhance the performance of sentiment analysis in terms of accuracy and time to train the classifier. Various methods proposed for document-level sentiment classification like Naive Bayes, k-Nearest Neighbor, Support Vector Machines and Decision Tree are analysed in this work. Cross domain sentiment classification is useful because many times we might not have training corpus of specific domains for which we need to classify the data and also cross domain is favoured by lower computation cost and time. Despite poor performance in accuracy, the time consumed for sentiment classification when multiple testing datasets of different domains are present is far less in case of cross domain as compared to single domain. This work aims to define methods to overcome the problem of lower accuracy in cross-domain sentiment classification using different techniques and taking the benefit of being a faster method.

Keywords Natural language processing · Machine learning · Cross domain

S. Mahalakshmi (✉) · E. Sivasankar
Department of Computer Science and Engineering, National Institute of Technology,
Tiruchirappalli, India
e-mail: maha.81193@gmail.com

E. Sivasankar
e-mail: sivasankar@nitt.edu

V. Ravi et al. (eds.), *Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO - 2015)*, Advances in Intelligent Systems and Computing 415, DOI 10.1007/978-3-319-27212-2_7

1 Introduction

With the evolution of web, people express or exchange their perspectives, feedbacks and opinions through online social media sites, such as user forums, blogs, discussion boards and review sites, etc. [1]. User views are considered to be valuable sources for improving the caliber of the service provided [2]. The user reviews are usually in unstructured format. Analyzing and gaining information from them can be a difficult task. Sentiment Classification (SC) is the technique which provides a solution to this problem. It focuses on identifying sentiment words and their polarity from user's review, which are then classified as positive and negative. This will help the user to arrive at a correct decision. Cross Domain Sentiment Classification (CDSC) is a ranging field of research in Machine Learning (ML). A fraction of the dataset is used for training the classifier and remaining fraction is used for testing in single domain whereas in cross domain, two completely different domains are considered and the classifier is trained with one domain dataset and tested using the other. The challenge in SC of user reviews is to improve the computational efficiency and also to ensure real time response for processing huge amounts of data with the feasibility of the technology to work across heterogeneous commercial domain [3]. Thus the computational complexity of the proposed technology is crucial.

Major research in sentiment analysis focuses on identifying people's emotion or attitude from text. In heterogeneous domain, the group is split into clusters and various techniques can be adopted for improving the results like spotting the keyword, lexical affinity, statistical methods and concept based approach [3]. Each method help in certain way to the problem formulation. Main reason for considering heterogeneous domains is the crucial problem faced by people in finding the sentiment of the texts or reviews extracted from the web accurately in an automated way irrespective of the domain. This paper is structured as follows: Sect. 2 describes the related work on CDSC. Different ML methods are elaborated in Sect. 3. Proposed frameworks are detailed in Sect. 4. Section 5 gives the experimental result obtained for the proposed models and performance of the CDSC for all classifiers are analyzed. Finally Sect. 6 concludes the work.

2 Literature Review

Bisio et al. [3] have done research on the classification across heterogeneous domains. In their research, they have defined a feature space for reviews using an integrated strategy which combines semantic network and contextual valence shifters. They have used empirical learning to map the reviews and its sentiment labels. Augmented k-Nearest Neighbor (kNN) [4] aided their model by improving the computational efficiency. Li et al. [5] proposed a sentiment transfer mechanism based on constrained non-negative matrix tri-factorizations of term-document matrices in the source and target domains.

Bollegala et al. [6] proposed a technique for CDSC that uses an automatically extracted Sentiment Sensitive Thesaurus (SST). They have used labeled data from multiple source domains and unlabeled data from source and target domains to construct SST. This constructed thesaurus is used for training and testing the binary classifier. A relevant subset of the features is selected using L1 regularization. Jambhulkar and Nirkhi [7] have compared and analyzed three techniques of cross domain sentiment analysis viz SST, spectral feature alignment, structural correspondence learning and discussed the challenges. This gave us an idea about the existing techniques and scope. We propose our model which uses techniques from the above literatures and the effectiveness is talked about in the following sections.

The performance of our proposed approach is evaluated experimentally by considering two large heterogeneous datasets of reviews : the first one is product reviews from AmazonTM [6] and the other one is hotel reviews from TripAdvisorTM [8]. The chosen datasets are available online to reproduce the showed results. The experimental results show that the proposed framework attains satisfactory performance in terms of both accuracy and computational efficiency.

The experimental datasets used for the analysis are two heterogeneous domains: Amazon Dataset (books, dvd, kitchen appliances, electronics) [6] and hotel reviews from TripAdvisor [8] which are quite distant commercial areas. The complications in SC of user reviews are majorly due to Linguistic nuances such as the quality of lexicon and incongruity in syntax that make it hard to apply any classical text mining approaches [6]. Drawing separating boundaries can be extremely tricky for classifiers especially in sparse scenarios. Thus the efficiency of classification is most dependent on the review patterns [9]. Apart from these impediments, cognitive issues also occurs for text to sentiment mapping of an established model. To facilitate these issues a holistic approach is required that integrates lexical and semantic information with ease [13].

3 Methodology

This part describes various ML techniques used to categorize the review as positive or negative. The reviews are preprocessed to improve the efficiency. Classifiers are trained using one dataset to predict the sentiment of the reviews from another dataset in cross domain. We measure the exactness of the result using confusion matrix.

For analysis we have used the following classifiers, K-Nearest Neighbour, Naive Bayes-Gaussian, Multinomial, Bernoulli, Support Vector Machine using Linear kernel and Radial Basis Function (RBF) kernel, Decision Tree.

K-Nearest Neighbour The kNN algorithm is a non-parametric method [4] which takes training set with class label as input for classification. Output is the class membership. This algorithm classifies the sample based on majority vote of its neighbors, the output is the most common class among its kNN [3]. K is a fixed constant which is generally small and an odd number is often chosen to avoid the neutrality case while doing the majority voting.

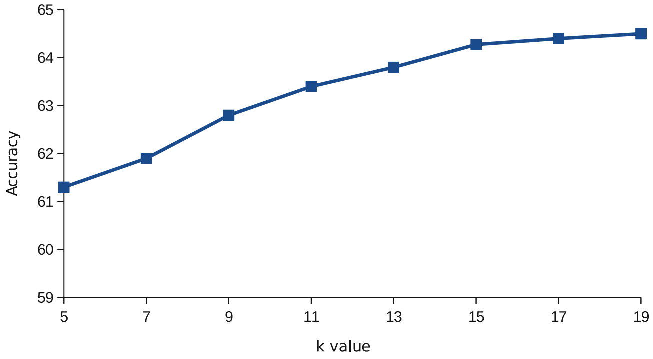

Fig. 1 Accuracy of KNN for different K value

The training set given as input to the classifier consist of vectors in multi-dimensional feature space, each with its class label. While training the classifier, the algorithm stores only the feature vectors and corresponding class labels for each sample. A user defined constant is chosen for k. An unlabeled feature vector is given as input for the classifier to identify the class. While classifying, k nearest feature vectors to the test data are identified by opting a distance metric. Most trivial and efficient distance metric is Euclidean Distance. Majority voting is done for the k identified neighbors and the class is assigned to the test vector.

An experiment is carried out on finding the accuracy of the k-NN classifier by training the Amazon dataset (pair 1) and testing with TripAdvisor dataset for Model 1. The Fig. 1 illustrates the experiment and it shows the increase in accuracy for increasing k and after k = 15, the increase in accuracy is not considerable and so we fixed the value of k as 15.

Naive Bayes NB classifiers [9] are simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. In Gaussian NB, the assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution.

$$p(x = v \mid c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{(v-\mu_c)^2}{2\mu_c^2}} \tag{1}$$

where x is a continuous attribute, c is a class, σ_c^2 is the variance and μ is the mean of the values in x associated with class c. With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events are generated by a multinomial. Most document classification uses this model, with events representing the occurrence of a word in a single document. The likelihood of observing a histogram X is given by,

$$p(x \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_{k_i}^{x_i} \tag{2}$$

where p_i is the probability that event i occurs. A feature vector $X = (x_1, ...x_n)$ is then a histogram, with X_i counting the number of times event i was observed in a particular instance.

In the multivariate Bernoulli event model, features are independent booleans (binary variables) describing inputs. If X_i is a boolean denoting the occurrence or absence of the $i'th$ term from the vocabulary, then the likelihood of a document given a class C_k is given by,

$$p(x \mid C_k) = \prod_{i=1}^{n} p_{k_i}^{x_i}(1 - p_{ki})^{(1-x_i)} \tag{3}$$

where p_{ki} is the probability of class C_k generating the term w_i. Same like Multinomial NB, this is also most suited for document classification [10]. Small amount of training data is sufficient to train the NB model to efficiently classify the test data. It is a simple technique for constructing classifiers.

Support Vector Machine SVMs or Support vector networks are supervised learning models. The data are analyzed and a pattern is recognized using learning algorithms that are part of the model. These patterns are used for analysis [9]. A set of training sample, each belonging to one of the two categories, is given as input to the SVM training algorithm. The algorithm constructs a model that will classify the new samples into one of the categories. Thus SVM is a non-probabilistic binary linear classifier. SVM model represents the training sample as points in space. Analysis of the points for pattern recognition results in identifying a clear gap which separates the two categories. When a test data is given to the model, it maps the data to the space and categorizes it based on which side of the gap it falls on [10]. Linear svm are generally applied on linearly separable data. If the data is not linearly separable, you'll need to map your samples into another dimensional space, using kernels like RBF.

Decision Tree DT is used in this learning technique. It as a predictive model which maps the details of the data to its class. This approach is used in statistics, data mining and ML [9]. Classification trees are models that accept the class to take finite set of values. In these models, class label take the leaf's position and the intermediate branches represent conjunctions of features that lead to the class labels. After preprocessing, review versus common words matrix is generated for Amazon and TripAdvisor dataset. These matrices are used for training the classifiers. The classifiers are trained using one dataset and tested using the other in each dataset pair.

Table 1 Confusion matrix for SC

		Predicted sentiment	
		+ ve sentiment	− ve sentiment
Actual sentiment	+ ve sentiment	True positive (TP)	False negative (FN)
	− ve sentiment	False positive (FP)	True negative (TN)

3.1 Performance Analysis

Accuracy measure is used to evaluate the performance of SC. Accuracy is determined as the proportion of correct number of predictions given total number of text reviews. Let us assume 2 × 2 confusion matrix as mentioned in Table 1.

The accuracy of SC is,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

4 Proposed Framework

The experimental evaluation of the cross-domain approach needs two entirely different training and testing datasets [3]. The first dataset reviews are from Amazon, [6] related to four product categories: books, DVDs, electronics and kitchen appliances which has 8000 reviews (1000 positive and 1000 negative reviews in each category). The second dataset is hotel reviews obtained from TripAdvisor, [8] which has millions of review. First N sampling technique is used to sample the data and consider only 4000 reviews in which 2000 positive and negative reviews each. This sampling method is opted because the database reviews are in random order. All the reviews are labelled.

We analyse the reviews in three stages—Preprocessing, Matrix Generation, Classification. Figure 2 shows the steps followed in the proposed method for the different feature models and classifiers considered.

4.1 Preprocessing

We preprocess the reviews using ML techniques such as Tokenization, removing stop words, Lemmatization, Stemming and Part-of-Speech (POS) Tagging.

Tokenization occurs at the word level, it is the process of breaking up the stream of text to words, phrases or symbols which are meaningful. These elements are termed as tokens [6, 11]. Stop words are words which don't express the sentiment of the sentences and so they are removed [6, 12]. In our work, we perform removal of stop

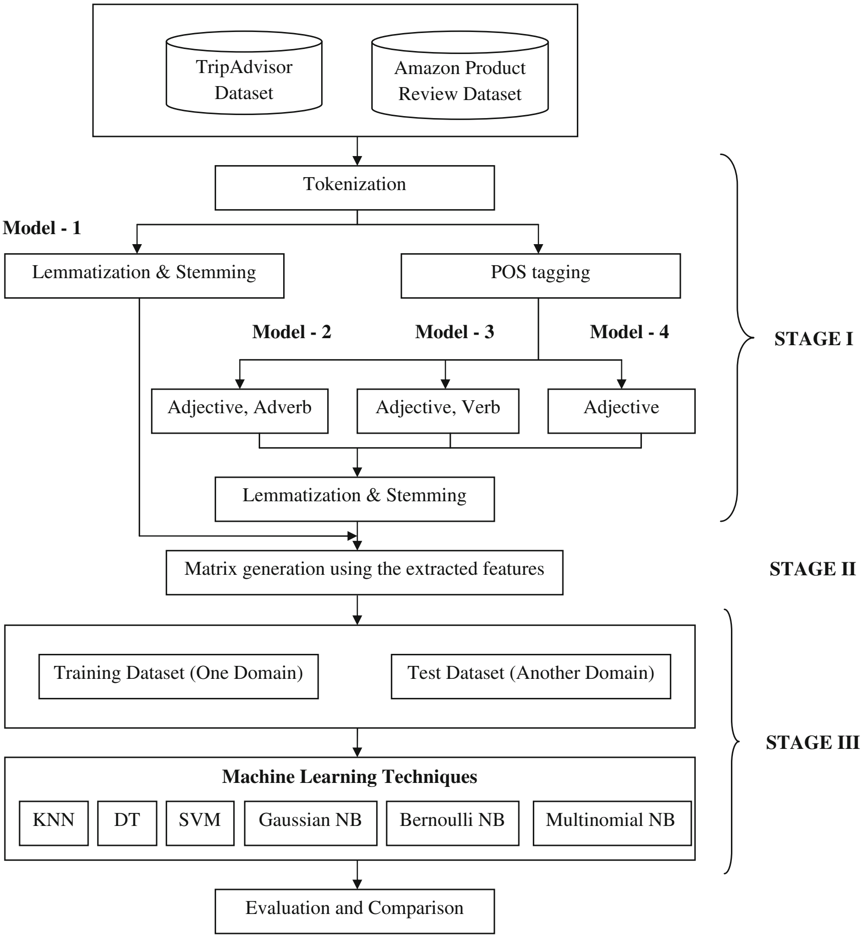

Fig. 2 Experimental setup

words after tokinizing the reviews. There is no single universal list of stop words. For a given purpose, any group of words can be chosen as the stop words. All words less than 3 characters, articles, prepositions and most common words can be part of the stop words [1, 12]. In Lemmatization, different inflected forms of a word are grouped together, so that they can be analyzed as a single item. In computational linguistics, it is the process of determining the lemma of a given word [6, 9]. Stemming is an informational retrieval process which reduces the words to their word stem or base form [6, 9]. Lemmatization and Stemming appear to be similar processes. The difference is that a stemmer operates on a single word without knowing the context and so it cannot discriminate between the words having different meanings depending upon the context unlike lemmatizer. The advantage of stemmer is that it can be easily implemented and is faster which overshadow the reduced accuracy in some applications. Part-of-Speech tagging which is also called grammatical tagging or word-category disambiguation, is the process of identifying the part of speech for each word in the review based on its definition and context. It considers the relationship with the adjacent and related words in a phrase, sentence or paragraph.

4.2 Model Generation

Following models are constructed for the analysis as mentioned in Fig. 2.

Model 1 Both the datasets are tokenized. Stop words are removed from the tokenized words and it is lemmatized. Lemmatized words are stemmed.

Model 2 Both the datasets are tokenized, pos tagging is applied and adjectives, adverbs are extracted and it is lemmatized and stemmed.

Model 3 Both the datasets are tokenized, pos tagging is applied and adjectives,verbs are extracted and it is lemmatized and stemmed.

Model 4 Both the datasets are tokenized, pos tagging is applied and adjectives are extracted and it is lemmatized and stemmed.

4.3 Matrix Generation

Common words between the two datasets are extracted from the preprocessing stage. In matrix generation, these common words are considered and term frequency matrix or binary matrix between reviews and the words is constructed for each of the datasets [3]. In our work, we consider frequency matrix for analysis.

In the classification stage, classifiers are trained and tested using this matrix.

5 Results and Analysis

The consistency of the result is checked by experimenting twice. First, the experiment is done using books and DVD reviews of Amazon (2000 pos and 2000 neg) and the hotel reviews from TripAdvisor (2000 pos and 2000 neg). Next, the experiment is done using electronics and kitchen appliances reviews from Amazon (2000 pos and 2000 neg) and the hotel reviews from TripAdvisor (2000 pos and 2000 neg). Each dataset pair is trained and tested. Amazon dataset is in xml format. Using regular expressions, reviews and labels (pos or neg) are extracted. Similarly, reviews are extracted from the TripAdvisor dataset, here the label is identified as positive if the rating (on a scale of 5) is 4 or 5 and negative if it is 1 or 2. The extracted reviews are preprocessed and common words are identified between the datasets for all the considered four models. Table 2 shows the number of common words after preprocessing in each model.

Table 2 Number of common words after preprocessing in each model

Dataset pair	Model 1	Model 2	Model 3	Model 4
Books, Dvd from Amazon, hotel from TripAdvisor	6889	3190	1818	1395
Electronics, kitchen appliances from Amazon, hotel from TripAdvisor	5254	2486	1444	1093

Table 3 Accuracy of classifiers—training Amazon and testing TripAdvisor

Classifiers	Model 1		Model 2		Model 3		Model 4	
	Dataset pair 1	Dataset pair 2	Dataset pair 1	Dataset pair 2	Dataset pair 1	Dataset pair 2	Dataset pair 1	Dataset pair 2
KNN (k = 15)	64.275	58.1	60.25	62.9	59.6	61.775	58.875	61.875
NB Gaussian	51.55	55.875	52.175	55.95	54.5	57.525	53.025	57.9
NB Multinomial	72.05	70.775	71.925	73.4	70.075	70.125	66.55	67.65
NB Bernoulli	69.35	73.25	68.45	74.275	69.8	68.575	64.25	65.125
DT	65.275	58.85	64.7	65.75	61.75	60.725	61.775	60.575
SVM rbf	66.85	66.275	65.8	65.375	62.225	60.725	59.275	58.325
SVM linear	70.65	68.15	68.675	70.7	66.15	70.125	61.65	65.85

Table 4 Accuracy of classifiers—training TripAdvisor and testing Amazon

Classifiers	Model 1		Model 2		Model 3		Model 4	
	Dataset pair 1	Dataset pair 2	Dataset pair 1	Dataset pair 2	Dataset pair 1	Dataset pair 2	Dataset pair 1	Dataset pair 2
KNN (k = 15)	60.625	63.225	59.875	61.85	57.25	59	56.875	56.625
NB Gaussian	53.55	54.15	53.675	55.35	51.975	53.225	51.75	53.275
NB Multinomial	58.925	66.8	65.725	72	64.65	69.075	63.175	67.3
NB Bernoulli	57.375	60.8	58.3	61.975	60.725	62.325	60.725	62.425
DT	64.275	68.95	61.75	64.325	58.65	59.45	57.45	56.9
SVM rbf	53.95	53.375	59.6	62.075	60.825	63.925	58.075	63.575
SVM linear	65.425	70.15	65.825	69.05	65.1	70.125	62.475	65.925

Review vs. Common word frequency matrix for both the datasets are constructed. Classification is done using the matrixes. One dataset is used for training and the other is used for testing. i.e. Classifier is trained with Amazon dataset and tested using TripAdvisor dataset and vice versa.

Table 3 shows the accuracy of classifiers when trained using Amazon dataset and tested with TripAdvisor dataset for each model. Table 4 shows the accuracy of classifiers when trained using TripAdvisor dataset and tested with Amazon dataset.

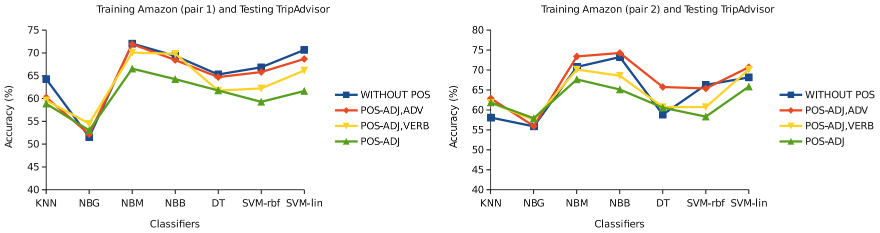

Fig. 3 Training with Amazon dataset and testing TripAdvisor dataset

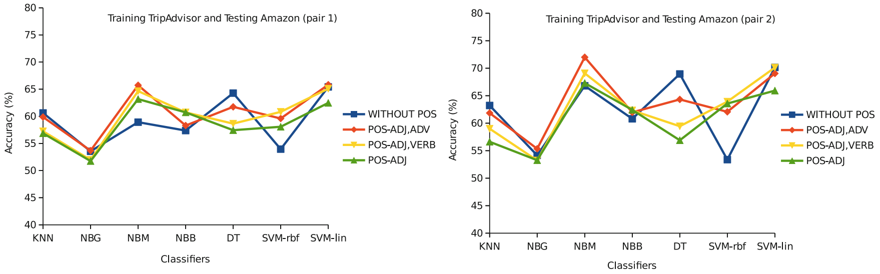

Fig. 4 Training with TripAdvisor dataset and testing Amazon dataset

From Figs. 3, 4 we observe that, for any model, NB Multinomial, SVM with linear kernel outperform the other classifiers with better accuracy. Performance of NB Bernoulli and DT are average, KNN, SVM with rbf kernel and NB Gaussian are poor with respect to accuracy.

Based on time,

1. With respect to models,
   Model 1 takes the highest time for training and testing, as it does not employ POS Tagging technique and so more common words between datasets, hence huge matrix and thus the impact in time. Model 2, 3, 4 take very less time compared to model 1.
2. With respect to classifiers,
   KNN takes the highest time to train and also to test the data irrespective of any model. All forms of NB, DT, SVM with linear kernel take almost the same time and they are the least compared to KNN and SVM with rbf kernel. SVM with rbf kernel takes more time compared to the above classifiers but considerably lesser time compared to KNN.

Considering both performance analysis and time analysis, we conclude that NB Multinomial, SVM with linear kernel are better than other classifiers for cross domain.

6 Conclusion

The overriding purpose of this analysis was to create an effective and efficient model for cross domain which can perform better with respect to computation and also space. To determine the best method, lot of models are considered and many preprocessing techniques are applied. Doing which determined, the model which incorporated POS Tagging performs better and Multinomial NB and SVM linear kernel classify the model very well compared to other classifiers. This work can be extended to big data with relatively large datasets. The results are promising with labelled dataset. The effectiveness of the same model on unstructured data can be worked upon.

References

1. Feldman, R.: Techniques and applications for sentiment analysis. Commun. ACM **56**(4), 82–89 (2013)
2. Dang, Y., Zhang, Y., Chen, H.: A lexicon-enhanced method for sentiment classification: an experiment on online product reviews. IEEE Intell. Syst. **25**(4), 46–53 (2010)
3. Bisio, F., Gastaldo, P., Peretti, C., Zunino, R., Cambria, E.: Data intensive review mining for sentiment classification across heterogeneous domains. In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 1061–1067, IEEE (2013)
4. Devroye, L., Wagner, T.J.: Nearest neighbor methods in discrimination. Handb. Stat. **2**, 193–197 (1982)
5. Li, T., Sindhwani, V., Ding, C., Zhang, Y.: Knowledge transformation for cross domain sentiment classification. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 716–717, ACM (2009)
6. Bollegala, D., Weir, D., Carroll, J.: Cross-domain sentiment classification using a sentiment sensitive thesaurus. IEEE Trans. Knowl. Data Eng. **25**(8), 1719–1731 (2013)
7. Jambhulkar, P., Nirkhi, S.: A survey paper on cross-domain sentiment analysis. Int. J. Adv. Res. Comput. Commun. Eng. **3**(1), 5241–5245 (2014)
8. Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: a rating regression approach. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 783–792, ACM (2010)
9. Cambria, E., Hussain, A.: Sentic computing: techniques, tools, and applications, vol. 2. Springer, Heidelberg (2012)
10. Cambria, E., Schuller, B., Xia, Y., Havasi, C.: New avenues in opinion mining and sentiment analysis. IEEE Intell. Syst. **2**, 15–21 (2013)
11. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retrieval **2**(1–2), 1–135 (2008)
12. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
13. Leoncini, A., Sangiacomo, F., Decherchi, S., Gastaldo, P., Zunino, R.: Semantic oriented clustering of documents. In: Advances in Neural Networks ISNN2011, pp. 523–529. Springer (2011)