BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 67, No. 2, 2019
DOI: 10.24425/bpas.2019.128608

Exploring the use of syntactic dependency features for document-level sentiment classification

K.S. KALAIVANI* and S. KUPPUSWAMI

Kongu Engineering College, Perundurai – 638060, Erode, India

Abstract. An automatic analysis of product reviews requires deep understanding of the natural language text by machine. The limitation of bag-of-words (BoW) model is that a large amount of word relation information from the original sentence is lost and the word order is ignored. Higher-order-N-grams also fail to capture the long-range dependency relations and word order information. To address these issues, syntactic features extracted from the dependency relations can be used for machine learning based document-level sentiment classification. Generalization of syntactic dependency features and negation handling is used to achieve more accurate classification. Further to reduce the huge dimensionality of the feature space, feature selection methods based on information gain (IG) and weighted frequency and odds (WFO) are used. A supervised feature weighting scheme called delta term frequency-inverse document frequency (TF-IDF) is also employed to boost the importance of discriminative features using the observed uneven distribution of features between the two classes. Experimental results show the effectiveness of generalized syntactic dependency features over standard features for sentiment classification using Boolean multinomial naive Bayes (BMNB) classifier.

Key words: document-level sentiment classification, syntactic dependency features, generalized dependency features, information gain, weighted frequency and odds.

1. Introduction

With the ongoing advancements of the web, an increasing number of users prefer to buy products online and to post reviews for a wide range of products they buy. Online reviews posted by people who have tested the products have become an important source of subjective information. However, a customer is confused by the large number of reviews given online for a product. Sentiment analysis is a field of study that aims to automatically extract such product reviews to identify opinions and to further classify them as positive and negative [1].

The task of sentiment classification can be performed using two approaches, namely semantic orientation [2–4] and machine learning approaches [5, 6]. The performance of machine learning approach is greatly dependent on the representation of documents and the choice of algorithms used for classification. Although BoW model is the most dominating document representation method, the problem with this model is that a large amount of word relation information from the original sentence is lost and the word order is ignored [7]. Hence word relation features with a deeper understanding of the text are essential to improve the accuracy of sentiment classification.

Traditional higher-order-N-grams are used to capture the word relation information from the text. N-grams are proximity-based sequence of elements like characters, words or part-of-speech (POS) tags that appear one after the other in a text where N corresponds to the number of elements that appear in the sequence. They are constructed by taking into account the words as they appear in the surface structure of the text or by sliding a window of size N over the text [8–10]. Since higher-order-n-grams fail to capture the long-range dependency relations and word order information, features based on syntactic dependency relations are employed.

Syntactic N-grams are sequences of words constructed from the elements that appear one after the other in the path of a syntactic tree of a sentence [11]. They are used to introduce syntactic information into the statistical machine learning methods, thereby eliminating the arbitrariness presented by the surface structure of the text. Sometimes syntactic dependency features fail to identify the correct sentiment polarity because they ignore the influence of negations in the sentence. So, all negated words in a sentence are presented as composite features in their negated status. In order to reduce the sparse data problem and make the feature space more effective, generalization of dependency features is done by backing-off the head word or the modifier word to their POS cluster.

In order to reduce the huge dimensionality of the feature space, feature selection methods based on information gain (IG) and weighted frequency and odds (WFO) are used. The weight of each feature used in feature vector is also the key component in the representation of a document. In addition to the traditional feature weighting schemes like term presence (TP), term frequency (TF) and term frequency-inverse document frequency (TF-IDF), a supervised feature weighting scheme called delta TF-IDF is used to weigh the features.

*e-mail: kalaiprani123@gmail.com

Manuscript submitted 2018-06-30, revised 2019-01-10 and 2019-03-03, initially accepted for publication 2019-03-04, published in April 2019.

The main contribution of this paper are as follows:

i) To explore the use of sentiment-oriented generalizations of dependency features with detected negations for document-level sentiment classification.

ii) To examine the use of IG and WFO as feature selection metrics to reduce the high dimensionality of the syntactic dependency feature space.

iii) To investigate the use of Delta TF-IDF as feature weighting metric for sentiment classification.

The remaining part of this paper is organized as follows: Section 2 deals with the other researches related to the current study. Document-level sentiment classification using syntactic dependency features is proposed in Section 3. Section 4 focuses on the dataset and evaluation metrics used and presents experimental results. Finally, Section 5 provides the conclusions of the present study.

2. Related works

Due to the availability of a large amount of opinionated information in the web, sentiment classification has become increasingly important. A number of feature extraction methods like unigrams, bigrams, trigrams, POS based features like adjectives, adverbs, verbs, nouns, sentiwordnet features and so on have been used by the researchers for sentiment classification [12–16]. Machine learning sentiment analysis has been introduced at first by Pang, Lee and Vaithyanathan [6]. These researchers have used unigrams, bigrams and adjectives as features and Naive Bayes, Support Vector Machine (SVM) and Maximum Entropy for sentiment classification on movie review dataset. The experimental results show that unigrams along with TP give better performance than TF and SVM gives the best accuracy among the classifiers used. Various POS-tagged features like adjectives, adverbs and nouns are used as features to analyze the performance of supervised sentiment analysis by Mejova and Srinivasan [17]. It is concluded that adjectives perform better than the other features as individual POS-tagged features.

Furthermore, the use of syntactic dependency features has yielded mixed results in the field of natural language processing. The subgraphs extracted from the dependency tree of a parsed sentence are used to construct the feature vector by Pak and Paroubek [5]. It is shown that the subgraph-based features along with SVM classifier outperform the other bag-of-words and N-gram features on movie review dataset. Gamon has used syntactic dependency features extracted from the phrase structure trees to yield improvements in the prediction of customer satisfaction rating [18]. Matsumoto et al. have used frequently occurring subtrees obtained from dependency relation parse tree as features for machine learning based sentiment classification and shown better performance in classifying the movie reviews as positive and negative [19].

An improved performance in identifying the opinions in deeply-nested clauses and classifying their strengths is observed by Wilson et al. by using several features extracted from dependency parse trees [20]. Dave et al. have proposed that adjective-noun dependency relationships used as features for the task of polarity prediction do not perform well in comparison to simple BoW features [21]. In addition to adjective-noun dependency relationships, the subject-verb and verb-object relationships are also considered by Ng et al. for polarity prediction [22]. Wiebe and Riloff have noted that syntactic patterns are very effective for subjective detection which is a preliminary step for sentiment analysis [23]. Sidorov has extracted N-grams as features based on the order in which the elements are present in the syntactic trees [8]. Syntactic relations represented using syntactic bigrams and trigrams are able to outperform other features for the task of authorship attribution.

Many authors have attempted to find more generalized dependency features to solve the sparsity problem. Gamon used the back off technique in N grams and dependency relations to their respective POS tags [18]. Joshi and Penstein-Rose proved that backing off only the head word in the dependency pairs to their POS tag yield better results for polarity classification [24].

Feature weighting schemes play a vital role in improving the classification results by assigning weights to the features according to their sentiment importance [6, 26]. In addition to the traditional feature weighting schemes like TP, TF and TF-IDF, a supervised feature weighting scheme called Delta TF-IDF is utilized to assign weights to the features.

Most of the earlier researches on sentiment analysis have used all the dependency relations extracted from the syntactic tree as features. Moreover, no feature selection is done to improve the classification accuracy. All the syntactic dependency relations are not sentiment bearing. So, only ten sentiment bearing relations are used as features in this study. The present study also differs from the earlier ones based on the construction of generalized syntactic dependency features and the use of negation handling.

3. Proposed system

The process of extraction of generalized syntactic dependency features involves the following steps:

1. Parse the sentence with the Stanford parser.
2. From the dependency relations obtained from the parse tree, use only the relations given in Table 1 to form syntactic bigrams. Syntactic trigrams are constructed from syntactic bigrams.
3. Generalized dependency features are constructed by backing off the head word or the modifier word to their respective POS cluster as shown in Table 2.

3.1. Feature extraction. The dependency parse for a given sentence is a set of triplets. The first component of the triplet is a grammatical relation that holds between the pair of words represented as second and third component. Let $\{rel_i, w_j, w_k\}$ be a triplet, where rel_i is the dependency relation between the words w_j and w_k. There are approximately 50 grammatical relations that exist between the words in a sentence, but not all of them are useful for sentiment analysis. So, only sentiment bearing features are extracted using the dependency relations as presented in Table 1. For example, let the sentence "This product is very good" be considered. The dependency relations extracted from Stanford parser are det (product_this), nsubj

Table 1
Selected dependency relations

S.No.	Dependency relation	Expansion	Backing-off the head/modifier word
1	Acomp	Adjectival complement	head
2	Advmod	Adverbial modifier	head
3	Amod	Adjectival modifier	head
4	Ccomp	Clausal complement	head
5	Cop	Copula	modifier
6	Dobj	Direct object	modifier
7	Neg	Negation modifier	modifier
8	Nsubj	Nominal subject	modifier
9	Rcmod	Relative clause modifier	head
10	Xcomp	Open clause complement	head

Table 2
POS Clustering

POS cluster	POS tags
J	JJ, JJR, JJS
R	RB, RBR, RBS
V	VB, VBZ, VBD, VBN, VBG, VBP
N	NN, NNP, NNS, NNPS, PRP
O	Other POS tags

(good_product), cop (good_is) and advmod (good_very). According to the dependency relations given in Table 1, "good_product, good_is and good_very" are the syntactic bigram features selected for further analysis.

To understand the importance of syntactic bigrams over traditional bigrams, let the sentence "This product is very good and cheap" be considered. Traditional bigrams for this sentence are "This_product, product_is, is_very, very_good, good_and and and_cheap". Likewise, syntactic bigrams extracted for the same sentence are "good_product, cheap_product, good_is, good_very and cheap_very". But, traditional bigrams are not able to extract good_product and cheap_product as features.

Once the syntactic bigrams are extracted, syntactic trigrams are formed by concatenating two syntactic bigrams. Let sb1 and sb2 be two syntactic bigrams. If the second element of sb1 is the same as the first element of sb2, then they can be combined to form a syntactic trigram. For example, the syntactic bigrams "very_good" and "good_product" can be combined to form a syntactic trigram "very_good_product".

3.2. Negation handling. Consider the following two sentences.

i) Sphere by Michael Crichton is an interesting novel.
ii) This is not an interesting novel.

The first sentence is a positive sentence and the second one is a negative sentence. When the sentences are parsed by the dependency parser, the resulting dependency relations contain the relation amod (novel, interesting) for expressing both positive and negative sentiments. The sentiment classifier cannot benefit from it as novel_interesting becomes a common feature for both positive and negative training examples. This study handles negation by presenting the feature in their negated status as not_novel_interesting for the negative sentence.

3.3. Construction of generalized dependency features. Consider the following examples.

i) I will definitely recommend this ipod.
ii) I will recommend this cd to anyone.

Both the above sentences have a direct object relationship as recommend_ipod and recommend_cd repectively. Both these features are good indicators of positive sentiment. If these features are treated independently, a sentiment classifier may not be able to generalize their relationship to the target class. Consider a test sentence with a different noun say "pendrive" (other than "ipod" or "cd") that participates in a similar relationship "dobj". It may not be able to get any importance in favor of the positive class because the classifier may not have seen it even once in the training data.

If the modifier word in each of the above features are backed-off to their POS cluster, it leads to a single feature (recommend_N). Now, the sentiment classifier may learn the weight for a more general feature which has a strong evidence of the target class. Also, a new test sentence with an unseen noun in a similar relationship with the verb "recommend" will receive some weight in support of the target class. Either the head word or the modifier word in backed off based on the dependency relation as shown in Table 1.

3.4. Feature selection. Feature selection is done to optimize the classifier's performance in terms of accuracy and computational speed by reducing the size of feature vector.

A) IG

IG recognizes the presence or absence of a feature in a document in order to determine the number of bits of information acquired for category prediction [26–28]. For a given feature f_i, IG is calculated as follows.

$$IG(f_i) = -\frac{A_i + B_i}{N}\log\frac{A_i + B_i}{N} + \frac{A_i}{N}\,log\,\frac{A_i}{A_i + C_i} + \\ + \frac{B_i}{N}\,log\,\frac{B_i}{B_i + D_i}, \tag{1}$$

where A_i is the number of the documents that contain the feature f_i and also belong to category c_i;
B_i is the number of the documents that do not contain the feature f_i, but belong to category c_i;
C_i is the number of the documents that contain the feature f_i but do not belong to category c_i;

D_i is the number of the documents that neither contain the feature f_i nor belong to category c_i;

N is the total number of documents in the training collection;

N_i is the number of documents that belong to category c_i.

B) WFO

Good features should possess high document frequency (A_i or C_i) and high category ratio $\left(^{A_i}/_{B_i}\text{ or }^{C_i}/_{D_i}\right)$. In real time applications, the document frequency and category information measures have to be varied appropriately to select the optimal features. A feature selection method called WFO proposed by [25] tunes the importance of features accordingly. It is calculated as follows.

$$WFO(f_i) = \left(\frac{A_i}{N_i}\right)^{\lambda}\left(log\,\frac{A_i(N-N_i)}{C_i N_i}\right)^{1-\lambda}, \qquad (2)$$

where λ is the parameter used to tune the weight between document frequency and category ratio and its value varies from 0 to 1. When the value of λ is equal to 0, the formula becomes equal to Mutual Information (category ratio) and when the value of λ is equal to 1, the formula becomes equal to document frequency. The value of λ is varied from 0 to 1 in steps of 0.1 during each run of 10-fold cross validation to achieve the best performance.

3.5. Feature weighting. The relevant features selected are then assigned with weights using unsupervised feature weighting schemes like TP, TF, and TF-IDF. In TP, a feature will be given a weight '1' if it is present in a document and '0' otherwise. TF measures the number of times a particular feature is present in a document. TF-IDF is calculated as the product of TF and IDF where

$$\text{IDF} = \frac{\text{Total number of documents}}{\text{Number of documents in which feature occurs}}, \qquad (3)$$

To select more informative features, a supervised feature weighting scheme, Delta TF-IDF is used in this study. Features that occur evenly in both positive and negative reviews are not good at discriminating the classes. Delta TF-IDF is the difference between the TF-IDF scores in the positive and negative classes and calculated as

$$\text{Delta TF} - \text{IDF}(\text{f}_\text{i}, \text{d}) = \text{TF}(\text{f}_\text{i}, \text{d}) * \log\left(\frac{C_i}{A_i}\right), \qquad (4)$$

where f_i is the feature present in document d.

Delta TF-IDF assigns weights to features based on their distribution in the reviews. It boosts the weight of those features that are unevenly distributed in the positive and the negative reviews and discounts the value of those features that are evenly distributed. If the feature occurs evenly in both positive and negative reviews, then it will be assigned zero. If the feature occurs prominently in negative reviews than in positive reviews, then it will have a positive weight and those occurring prominently in positive reviews than in negative reviews will have a negative weight. So, Delta TF-IDF weighting scheme better represents the importance of features and hence performs better than the unsupervised weighting schemes.

3.6. Classification. Even though a number of machine learning algorithms like Naive Bayes, SVM and Maximum Entropy have been used for sentiment analysis, SVM proves to be the best machine learning algorithm for sentiment analysis. Agarwal and Mittal have shown that the BMNB classifier combined with mRMR feature selection method outperforms SVM classifier by including more informative and less redundant features [29]. This study uses SVM and BMNB machine learning algorithms for sentiment classification. Since the dataset does not have a separate testing set, 10-fold cross validation technique is used to evaluate the results of the proposed system.

4. Dataset, evaluation metrics and results

4.1. Dataset. One of the most popular product review datasets that consists of Amazon product reviews is used to evaluate the performance of the proposed study. This dataset contains reviews of various domains like books, DVD, electronics and kitchen [30]. Each domain has 1000 positive and 1000 negative labelled reviews. Books and DVD domain consist of longer reviews in comparison to electronics and kitchen domain. For all the four domains, 1800 reviews (900 positive and 900 negative reviews) are used for training the classification model and the remaining reviews are used for testing.

4.2. Evaluation metrics. Precision, recall, F-measure and accuracy are used to evaluate the performance of a sentiment classifier. For a given category c_i, the values of precision, recall, F-measure and accuracy are computed as given in (5) to (8).

$$\text{Precision} = \frac{\text{Documents correctly classified to category ci}}{\text{Total documents classified to category ci}}, \qquad (5)$$

$$\text{Recall} = \frac{\text{Documents correctly classified to category ci}}{\text{Total documents in category ci}}, \qquad (6)$$

$$\text{F} - \text{measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}, \qquad (7)$$

$$\text{Accuracy} = \frac{\text{Total number of correctly classified documents}}{\text{Total number of documents}}. \qquad (8)$$

4.3. Results and discussion. Experiments are conducted to compare the performance of traditional N-grams like unigrams, bigrams and trigrams, syntactic dependency features like syntactic bigrams and syntactic trigrams and their generalized features with negation handling for document-level sentiment classification.

A) Comparison of feature extraction methods

Tables 3 and 4 present the accuracy of various feature extraction methods for all the four selected datasets. Among traditional

Table 3
Accuracy (in %) for various feature sets for Books and DVD dataset

Feature Extraction methods	**Books**								**DVD**							
	SVM				***BMNB***				***SVM***				***BMNB***			
	TP	***TF***	***TF-IDF***	***Delta TF-IDF***	***TP***	***TF***	***TF-IDF***	***Delta TF-IDF***	***TP***	***TF***	***TF-IDF***	***Delta TF-IDF***	***TP***	***TF***	***TF-IDF***	***Delta TF-IDF***
Traditional unigrams	80.8	76.1	74.4	**81.2**	76.7	75.8	74.1	**79.2**	79.1	78.2	78.5	**79.8**	76.2	75.1	75.3	**78.2**
Traditional bigrams	66.5	67.2	65.9	69.3	69.3	70.2	69.2	71.3	66.2	67.8	67.2	69.2	67.4	68.2	68.9	70.9
Traditional trigrams	52.1	53.4	53.1	55	52.4	54.6	54.3	56.2	51.3	52.7	52.5	54.6	51.8	54.1	54	55.3
Syntactic bigrams	80.5	81.2	79.5	81.9	81.4	81.3	80.2	82.6	80.2	80.7	79.3	81.1	81.7	81.6	80.3	82.4
Syntactic trigrams	66.5	68.4	67.3	69.5	67.4	69.2	68.1	71.4	66.4	68.1	67.4	69.3	67.1	69.8	68.6	71.5
Syntactic bigrams-IG	82.3	83.2	82.8	84.5	83.6	84.2	83.4	85.1	82.1	83.6	82.7	84.6	83.2	84.5	83.2	85.3
Syntactic trigrams-IG	67.8	71.2	71	72.5	73.6	76.2	75.5	78.1	67.9	71.4	71.3	72.4	73.2	76.4	75.7	78.4
Syntactic bigrams-WFO	84	84.8	84.1	87.3	84	86.4	85.8	88.2	83.8	84.7	84.4	86.3	84.3	86.5	85.6	88.2
Syntactic trigrams-WFO	70.6	71.5	70.7	74.9	75.6	77.1	76.3	78.5	70.5	71.7	70.3	74.2	75.7	77.6	76.1	78.7
Gen. dependency features with neg. handling	83.6	84.4	84.2	85	84.4	85.6	85.2	88.4	83.4	84.4	84.5	86.2	84.3	85.6	85.3	88.6
Gen. dependency features with neg. handling-IG	84.8	85.2	85	87.3	85.2	87.2	86.5	89	85.1	85.5	85.4	87.3	85.2	87.4	86.7	89.4
Gen. dependency features with neg. handling-WFO	86.4	86.5	86.2	**88.3**	87.6	88.3	87.2	**89.2**	86.2	87.3	86.7	**88.4**	87.2	88.5	87.4	**89.9**

Table 4
Accuracy (in %) for various feature sets for electronics and kitchen dataset

Feature Extraction methods	Electronics								Kitchen							
	SVM				*BMNB*				*SVM*				*BMNB*			
	TP	*TF*	*TF-IDF*	*Delta TF-IDF*	*TP*	*TF*	*TF-IDF*	*Delta TF-IDF*	*TP*	*TF*	*TF-IDF*	*Delta TF-IDF*	*TP*	*TF*	*TF-IDF*	*Delta TF-IDF*
Traditional unigrams	82.3	81.4	81.1	**82.8**	79.4	78.2	78.1	**81.6**	83.2	82.1	82.3	**83.7**	80.6	79.2	79.4	**82.7**
Traditional bigrams	69.3	70.1	70.4	72.3	70.5	71.1	71.7	73.8	70.4	71.2	71.6	73.4	71.5	72.3	72.8	74.6
Traditional trigrams	54.2	55.3	55.8	57.8	54.2	57.6	57.3	58.7	55.1	56.4	56.6	58.9	55.3	58.5	58.4	59.8
Syntactic bigrams	84.3	84.6	83.5	85.2	85.2	85.4	84.3	86.1	85.4	85.9	84.8	86.4	86.6	86.2	85.6	87.8
Syntactic trigrams	70.3	72.4	71.6	73.2	71	73.2	72.8	75.5	71.4	73.2	72.5	74.3	72.2	74.4	73.5	76.6
Syntactic bigrams-IG	86.3	87.2	86.3	88.2	87.3	88.6	87.4	89.1	87.1	88.3	87.4	89.1	88.3	89.3	88.8	90
Syntactic trigrams-IG	71.4	75.2	75.5	76.3	77.5	80.1	79.6	82.8	72.2	76.3	76.5	77.2	78.5	81	80.2	83.5
Syntactic bigrams-WFO	85.2	86.1	86.2	88.7	86.5	87.7	87.1	89.5	86.1	87.2	87.4	89.5	87.6	88.8	88.2	90.2
Syntactic trigrams-WFO	72.4	73.8	72.1	76.4	77.8	79.3	78.4	80.8	73.5	74.9	73.2	77.5	78.9	80.4	79.6	81.8
Gen. dependency features with neg. handling	86.1	87.2	87.1	87.5	87.3	87.6	88.3	88.2	87.3	88.1	88.2	88.4	88.3	88.5	89.2	89.4
Gen. dependency features with neg. handling-IG	87.2	87.3	87.5	88.6	87.6	88.4	88.3	89.3	88.1	88.3	88.5	89.6	88.5	89.4	89.3	90
Gen. dependency features with neg. handling-WFO	88.4	89.1	88.6	**89.3**	89.6	89.3	88.4	**90.2**	88.7	90.2	89.1	**90.1**	89.3	90.5	89.6	**91.2**

N-grams, it is found that unigrams perform better than bigrams and trigrams for all the four datasets. The first reason is that bigrams and trigrams are sparser than unigrams and so the performance degrades. The other reason may be that the bigrams and trigrams contain more noisy features which deteriorate the classification accuracy.

Syntactic dependency features are also utilized to introduce syntactic information into the statistical machine learning methods and to capture the long-range dependencies present in the text. Among the syntactic features, syntactic bigrams perform better than syntactic trigrams because the trigrams suffer from the data sparseness problem. In addition, they introduce more noisy features which reduce the performance of machine learning methods. Backing-off either head or modifier word to their respective POS category captures more generalizable and informative patterns in the training data. Generalized dependency features give the highest classification accuracy among all the feature extraction methods used in the study.

B) Comparison of feature selection methods

Feature selection methods are used to identify important and relevant features that represent the class attribute in a low dimensional feature space. They are also used to improve the classification accuracy and to reduce the computational time of the machine learning algorithms. Further around 10% to 20% of the features are sufficient to classify the reviews efficiently. Figure 1 presents the comparison of generalized dependency features with and without feature selection. WFO performs better than IG because IG selects only the most important features for further processing, whereas WFO feature selection technique selects those features that are both important and less correlated.

C) Comparison of feature weighting schemes

TF-IDF weighting scheme boosts the value of features that are more frequent in a document but only for those that occur in a very small number of other documents in the collection. So, sentiment bearing words like good, bad, love, hate, great, worse, recommend and not_recommend that occur in large number of documents are assigned less weights. Moreover, these words have low TF (occur lesser number of times in any document). The reason for this is that the reviewers try to use synonymous words to write the reviews in order to avoid boring the readers. So, features in a document should be assigned greater weight if they occur more often in a category and comparatively rare in another category. Delta TF-IDF weighting scheme does this by assigning weights to features based on their distribution in the reviews. From Fig. 2, it is clear that Delta TF-IDF performs better than the other unsupervised feature weighting schemes.

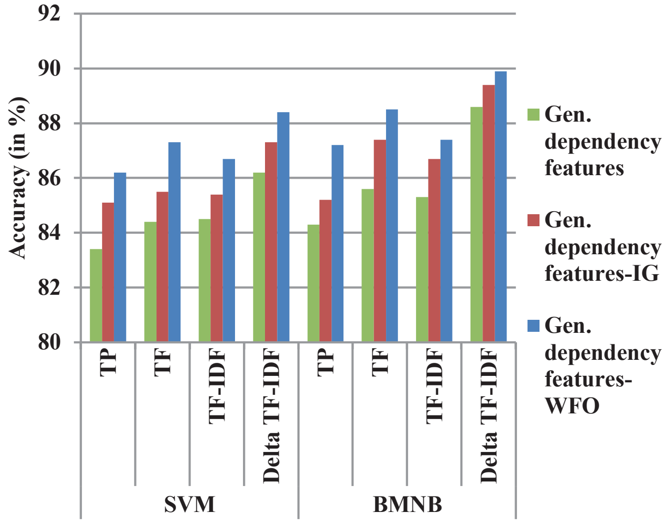

Fig. 1. Comparison of generalized dependency features with and without feature selection on DVD dataset

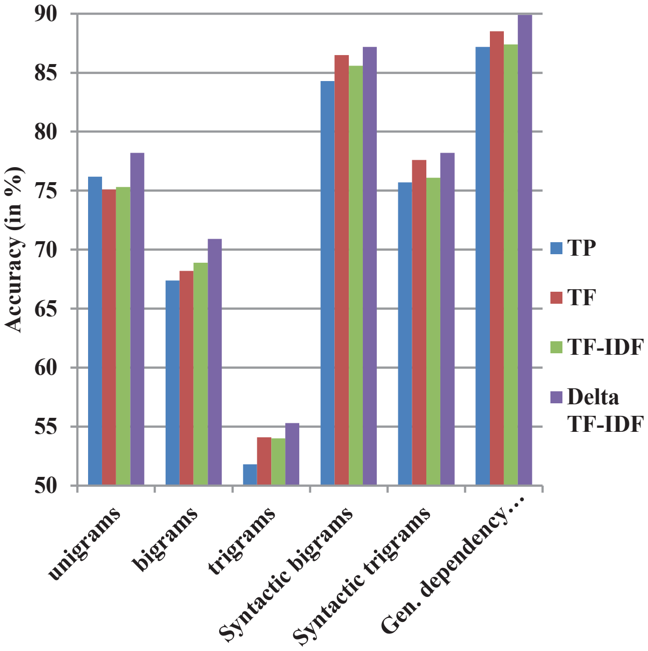

Fig. 2. Comparison of various feature weighting schemes on DVD dataset using BMNB classifier

D) Comparison of classifiers

From Tables 3 and 4, it is found that BMNB performs better than SVM for all the feature extraction methods, except unigrams. The reason is that bigrams and trigrams are less relevant and more independent than unigrams. Since BMNB classifier works based on conditional independence assumption, it performs well on bigrams and trigrams. Like traditional bigrams and trigrams, syntactic dependency features are also more independent and less relevant. So, BMNB classifier outperforms SVM on syntactic dependency features also.

Figure 3 shows the total time taken for classification process by SVM and BMNB classifiers for different feature set sizes. For all the datasets, it is found that BMNB classifier performs better than SVM in terms of classification time.

From Fig. 3, it is also clear that feature selection plays an important role in improving the performance of sentiment

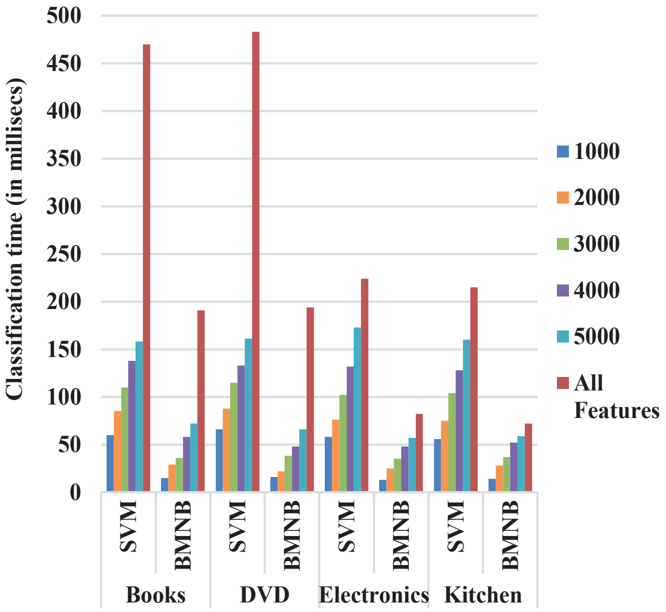

Fig. 3. Comparison of classification time of SVM and BMNB classifier on all the four datasets

classification. With no feature selection (considering all features), books and DVD datasets take longer time than electronics and kitchen dataset. The reason is that the number of features is more in books and DVD datasets as they have longer reviews.

Figure 4 shows the results of the comparison of evaluation metrics like precision, recall, F-measure and accuracy for different feature sets on books dataset using BMNB classifier. From the figure, it is clear that unigrams perform better than bigrams and trigrams among the traditional N-grams and generalized dependency features perform better than syntactic bigrams and syntactic trigrams. As BMNB classifier is based on conditional independence assumption, it performs well on bigrams and trigrams.

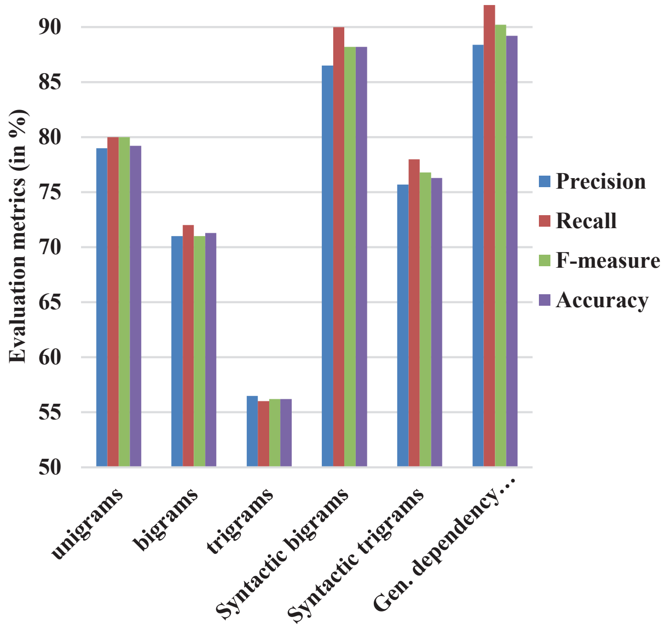

Fig. 4. Evaluation metrics of different feature sets for BMNB classifier on books dataset

From the experiments conducted, it can be concluded that the performance of document-level sentiment classification can be improved by using generalized dependency features along with negation handling. Moreover, the use of WFO for feature selection along with Delta TF-IDF as feature weighting technique helps to further enhance the performance of machine learning methods. Among the classifiers, BMNB performs better than SVM in terms of execution time and accuracy.

5. Conclusions and future work

In this paper, the performance of traditional N-gram and syntactic dependency features for document-level sentiment classification was investigated on four different standard datasets containing Amazon product reviews. Generalized dependency features with detected negations gave better performance compared to traditional N-gram features and syntactic N-grams. IG and WFO feature selection methods were used for extracting relevant features. Comparative performance of IG and WFO was investigated and it was observed that WFO performs better than IG on both types of N-gram features. The reason is that WFO feature selection method selects relevant features based on optimal document frequency and category ratio unlike IG which can only compute the importance of the feature. BMNB gives better performance in terms of execution time and accuracy for sentiment classification. As future work, new feature extraction techniques may be explored as the machine learning methods require them for effective sentiment classification.

References

[1] C.C. Aggarwal, "Opinion Mining and Sentiment Analysis." *Machine Learning for Text*, pp. 413–434. Springer, Cham, 2018.

[2] Y. Dang, Y. Zhang, and H. Chen, "A lexicon enhanced method for sentiment classification: an experiment on online product reviews", *IEEE Intell Syst* 25(4), 46–53 (2010).

[3] M. Dragoni, S. Poria, and E. Cambria., "OntoSenticNet: A commonsense ontology for sentiment analysis", *IEEE Intelligent Systems*, 33 (3), pp. 77–85, 2018.

[4] P. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews", *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, 417–424 (2002).

[5] A. Pak and P. Paroubek, "Text representation using dependency tree sub-graphs for sentiment analysis", *Proceedings of the 16th international conference DASFAA workshop*, vol. 6637, pp. 323–332, Hong Kong, 2011.

[6] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques", *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, Prague, pp. 79–86 (2002).

[7] R. Xia and C. Zong, "Exploring the use of word relation features for sentiment classification", *Proceedings of the 23rd International Conference on Computational Linguistics,* pp. 1336–1344 (2010).

[8] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, "Syntactic dependency-based n-grams as classification features", *Mexican International Conference on Artificial Intelligence*, Springer, Berlin, Heidelberg, pp. 1–11 (2012).

[9] G. Sidorov, "Syntactic dependency based n-grams in rule based automatic English as second language grammar correction", *International Journal of Computational Linguistics and Applications*, 4 (2),169–88 (2013).

[10] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, "Syntactic n-grams as machine learning features for natural language processing", *Expert Systems with Applications*. 41 (3), 853–60 (2014).

[11] A. Segura-Olivares, A. García, and H. Calvo, "Feature Analysis for paraphrase recognition and textual entailment", *Research in Computing Science*, 119–44 (2013).

[12] A. Esuli and F. Sebastiani, "SentiWordNet: a publicly available lexical resource for opinion mining", *Proceedings of Language Resources and Evaluation* (2006).

[13] L.P. Hung and R. Alfred, "A performance comparison of feature extraction methods for sentiment analysis", *Advanced Topics in Intelligent Information and Database Systems*, Springer International Publishing, 2017.

[14] S.D. Sarkar and S. Goswami, "Empirical study on filter based feature selection methods for text classification", *Int. J. Comput. Appl*., 81 (6), 0975–8887 (2013).

[15] A. Sharma and S. Dey, "Performance investigation of feature selection methods and sentiment lexicons for sentiment analysis", IJCA Special Issue on Advanced Computing and Comm Technologies for HPC Applications, vol. 3, pp. 15–20 (2012).

[16] A. Novikov, M. Trofimov and I. Oseledets. "Exponential machines", *Bull. Pol. Ac.: Tech.* 66, no. 6 (2018).

[17] Y. Mejova and P. Srinivasan, "Exploring feature definition and selection for sentiment classifiers", *ICWSM* (2011).

[18] M. Gamon, "Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis", Proceedings of the 20th international conference on Computational Linguistics, (2004).

[19] S. Matsumoto, H. Takamura, and M. Okumura, "Sentiment classification using word sub-sequences and dependency sub-trees", *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, Berlin, Heidelberg, pp. 301–311 (2005).

[20] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis", *Proceedings of the conference on human language technology and empirical methods in natural language processing* (2005).

[21] K. Dave, S. Lawrence, and D.M. Pennock, "Mining the peanut gallery: opinion extraction and semantic classification of product reviews", *Proceedings of the 12th international conference on World Wide Web (WWW)*, Budapest, 519–528 (2003).

[22] V. Ng, S. Dasgupta, and S.M. Arifin, "Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews", *Proceedings of the COLING/ACL on Main conference poster sessions*, pp.611–618 (2006).

[23] J. Wiebe and E. Riloff, "Creating subjective and objective sentence classifiers from unannotated texts", *International Conference on Intelligent Text Processing and Computational Linguistics,* Springer, Berlin, Heidelberg, pp. 486–497 (2005).

[24] M. Joshi and C. Penstein-Rosé, "Generalizing dependency features for opinion mining", *Proceedings of the ACL-IJCNLP 2009 conference short papers*, pp. 313–316 (2009).

[25] C.D. Manning, P. Raghvan, and H. Schutze, "Introduction to information retrieval", *Cambridge University Press*, Cambridge (2008).

[26] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization", *Proceedings of International Conference of Machine Learning,* pp. 412–420 (1997).

[27] T. Parlar, S.A. Özel, and F. Song, "QER: a new feature selection method for sentiment analysis", Human-centric Computing and Information Sciences, 8 (1), p. 10 (2018).

[28] Z.H. Deng, K.H. Luo, and H.L. Yu, "A study of supervised term Weighting Scheme for sentiment analysis", *Expert Systems with Applications,* pp.3506–3513 (2014).

[29] B. Agarwal and N. Mittal, "Optimal feature selection for sentiment analysis", *CICLing*. 7817 (1), pp. 13–24 (2013).

[30] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification", ACL (2007).