Cogn Comput
DOI 10.1007/s12559-014-9316-6

Concept-Level Sentiment Analysis with Dependency-Based Semantic Parsing: A Novel Approach

Basant Agarwal · Soujanya Poria · Namita Mittal · Alexander Gelbukh · Amir Hussain

Received: 21 March 2014 / Accepted: 26 December 2014

Abstract Sentiment analysis from unstructured natural language text has recently received considerable attention from the research community. In the frame of biologically inspired machine learning approaches, finding good feature sets is particularly challenging yet very important. In this paper, we focus on this fundamental issue of the sentiment analysis task. Specifically, we employ concepts as features and present a concept extraction algorithm based on a novel concept parser scheme to extract semantic features that exploit semantic relationships between words in natural language text. Additional conceptual information of a concept is obtained using the ConceptNet ontology: Concepts extracted from text are sent as queries to ConceptNet to extract their semantics. We select important concepts and eliminate redundant concepts using the Minimum Redundancy and Maximum Relevance feature selection technique. All selected concepts are then used to build a machine learning model that classifies a given document as positive or negative. We evaluate our concept extraction approach using a benchmark movie review dataset provided by Cornell University and product review datasets on books, DVDs, and electronics. Comparative experimental results show that our proposed approach to sentiment analysis outperforms existing state-of-the-art methods.

Keywords Sentiment analysis · Semantic parser · Dependency rules · Minimum Redundancy and Maximum Relevance feature selection · ConceptNet

B. Agarwal (✉) · N. Mittal
Department of Computer Science and Engineering, MNIT, Jaipur, India
e-mail: thebasant@gmail.com

N. Mittal
e-mail: mittalnamita@gmail.com

S. Poria · A. Hussain
Department of Computing Science and Mathematics, University of Stirling, Stirling FK9 4LA, UK
e-mail: soujanya.poria@cs.stir.ac.uk

A. Hussain
e-mail: ahu@cs.stir.ac.uk

A. Gelbukh
Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico City, Mexico
e-mail: gelbukh@cic.ipn.mx

Introduction

The textual information available on Web is of two types: facts and opinion statements. Facts are objective sentences about entities and do not show any sentiment. Conversely, opinions are subjective in nature and generally describe people's sentiments toward entities and events.

The interest in sentiment analysis research, both text-based and multimodal [1, 2], has been increasing tremendously in recent years, due to a wide range of business and social applications in human-centric environments [3] such as, e-learning and e-health [4, 5]. Making appropriate sense of what other people think has always been considered important for decision making. Whenever people want to purchase a product, e.g., a mobile phone, camera, laptop, etc., they often ask their friends or peers who have already used the product. Nowadays, due to growing trends in the Web, people express their opinions, feelings, and experiences about products or services on forums, blogs, social networks, and content-sharing services.

Consequently, the necessity of analyzing and understanding these online user-generated data and reviews has arisen. A potential user can know the merits and

Published online: 20 January 2015

disadvantages of a product from the experiences shared by existing users on the Web, which can be useful for making purchasing decisions [6]. E-commerce companies learn current trends and improve their product or services by analyzing opinions of users. Examples of sentiment analysis applications include identifying popularity of movies from online reviews and finding which camera model or which music piece most users like. Sentiment classification is used to assign a document, such as a product review, to a polarity category—positive, negative, or neutral—by its subjective information.

Common-sense knowledge represents basic understanding that people acquire through experience [7]. Automatic common-sense reasoning is often performed using common-sense ontologies and reasoning algorithms such as predicate logic and machine learning. Common sense, in particular, is necessary to properly identify sentiments in natural language text: for example, the concept ‘small room’ should be understood as negative in a hotel review, while ‘small queue’ is positive when referring to a post office; the concept ‘go read the book’ is positive in a book review but negative in a movie review [8].

Common-sense knowledge has not been explored in the domain of sentiment analysis with machine learning methods. Our proposed common-sense knowledge-based approach is able to detect sentiment expressed implicitly through the analysis of concepts that do not explicitly convey any sentiment, but are implicitly linked to other concepts that do convey sentiment. The concept-level sentiment analysis outperforms other existing models because it preserves the semantics associated with multi-word expressions. Our approach relies on the implicit information associated with natural language concepts. A concept parser breaks text down into clauses and then deconstructs such clauses into concepts, which are mapped to a vector space of common-sense knowledge.

In machine learning, a feature is relevant for classification if, after eliminating this feature, the classification performance degrades. Irrelevant features are not necessary for classification [9, 10]. Such redundant features degrade the performance of classification. Redundancy of features can be detected by the measure of correlation between features [11]. With thousands of features, it is very common that a large number of features are not informative, being irrelevant or redundant with respect to the class. Performance of machine learning methods deteriorates due to high dimensionality of feature vectors and inclusion of irrelevant and noisy features [12, 13]. Therefore, removal of these irrelevant and redundant features improves the performance of classification [9, 13].

A characteristic of a salient feature for sentiment analysis is that it should be discriminating enough so that the classifier can use information on the presence of this feature to predict the class for a new sample [13]. The main purpose of feature selection methods is to select features that are relevant and discriminating for the classification, by eliminating noise and irrelevant features. Feature selection techniques are used to identify important and relevant features, and hence select the minimum amount of prominent features that represent the class attribute in a reduced feature space. Feature selection has two main benefits: It significantly improves the classification accuracy and provides better insight into prominent class features, resulting in a better understanding of sentiment arguments and their features [9]. A reduced feature vector consisting of relevant and prominent features improves the computation speed and increases the accuracy of machine learning methods [10, 13]. By eliminating noise and irrelevant features, feature selection results in more efficient data representation in lower-dimensional feature space.

The main advantage of the feature selection technique called Minimum Redundancy and Maximum Relevance (mRMR) is that it eliminates redundant features and selects only relevant features, unlike other well-known feature selection techniques, such as information gain (IG) and mutual information (MI). Other existing feature selection techniques used for sentiment analysis only focus on selecting relevant features without considering redundant information.

In this paper, we propose a novel concept parser based on semantic relationship between words in natural language text, and on the semantic information present in the ConceptNet ontology. After filtering using the mRMR feature selection technique, we employ these extracted concepts to train a machine learning model. With this model, we learn concept patterns in the text, which are then used to classify documents into positive and negative categories.

The paper is organized as follows. Section 2 describes prior works related to sentiment classification. Section 3 discusses various feature extraction methods. Section 4 presents our proposed approach for sentiment classification. Section 5 gives a comparative illustration of various features. Section 6 discusses experimental results and compares our work with existing approaches. Finally, Section 7 concludes the paper and proposes some directions for future work.

Related Work

Various feature extraction methods have been proposed in the literature to extract patterns from unstructured text and train a machine learning model. However, most previous pattern extraction methods mainly rely on simple keywords spotting, POS tags, and syntactic information.

Machine learning methods require good, representative features for delivering good performance. Consequently, the performance of machine learning-based approaches depends greatly on the effectiveness of the feature extraction process. Various types of features have been explored by researchers in an attempt to produce better classification results.

For example, Pang et al. [14] used unigrams, bigrams, and adjectives to generate feature vectors for various machine learning algorithms on a movie review dataset. Matsumoto et al. [15] used syntactic relations between words and word subsequences as features for sentiment classification. They applied text mining techniques to extract frequent word subsequences and dependency sub-trees from sentences to train a machine learning algorithm for sentiment classification on the movie review dataset. Pak and Paroubek [16] extracted dependency tree subgraphs of a parsed sentence as features for sentiment classification. Their experimental results on the movie review dataset show that features based on extracted subgraphs with a SVM classifier outperform bag-of-words and n-gram features. Nakagawa et al. [17] used syntactic dependency trees for sentiment analysis and obtained better performance than using bag-of-word features. Xia et al. [18] investigated word relation features that incorporate the information of relation between words and showed that these are effective features for sentiment classification. Riloff et al. [19] employed a subsumption hierarchy to formally define various types of lexical features. Joshi and Penstein-Rose [20] experimented with syntactic features for sentiment classification and proposed generalized syntactic dependency features, which improved opinion mining results.

In many supervised sentiment classifications, adjectives are also treated as features for machine learning methods. Mejova and Srinivasan [21] experimented with the effectiveness of various POS-tag features for supervised sentiment classification. They used adjectives, adverbs, and nouns in combination and as independent features. Their experimental results showed that adjectives, adverbs, and nouns perform better in combination, than as independent features. They also proved that adjectives outperform other words as individual POS-tag features.

Mullen and Collier [22] proposed a method to expand the feature set based on Osgood's theory of semantic orientation [23] and Turney's [24] semantic orientation method [24] for supervised sentiment classification. Dang et al. [25] proposed a lexicon enhanced method by combining machine learning and semantic orientation-based approaches, which significantly improved the performance of sentiment classification. They proposed sentiment features based on the SentiWordNet lexicon in addition to content-free, content-specific features such as unigrams and bigrams. Their experimental results showed that rarely used sentiment features enhance the performance of sentiment classification.

Other relevant works in concept mining focus on concept extraction from documents. Gelfand et al. [26] have developed a method based on the semantic relation graph to extract concepts from a whole document. They used the relationships between words, extracted from a lexical database, to form concepts.

Feature Extraction

We construct various feature sets to build the machine learning model for sentiment classification. Initially, we construct four basic feature sets that are popular in the literature, namely unigrams (F1 feature set), bigrams (F2 feature set), bi-tagged (F3 feature set), and dependency parse tree-based features (F4 feature set). These feature sets are presented below.

Unigrams

Unigram features are simply a bag of words extracted by separating text by spaces and noise characters. For example, in the sentence 'This is an awesome movie,' the words 'this,' 'is,' 'an,' 'awesome,' 'movie' are all distinct unigram features.

Bigrams

Bigrams are features consisting of two consecutive words in the text. For example, in the sentence 'This is not a good book,' 'this is,' 'is not,' 'not a,' 'a good,' 'good book' are distinct bigram features. These features are capable of incorporating some contextual information.

Bi-tagged

Bi-tagged features are selectively extracted using part-of-speech (POS)-based fixed patterns. Bigrams containing mostly adjectives and adverbs are considered more sentiment bearing. We use POS-based information to extract sentiment-rich features, as it has been reported in literature that adjective and adverbs are subjective in nature [24, 27]. Turney [24] proposed a method to extract two-word sentiment-rich features such that one of the two words is either an adjective or an adverb. We adopted POS-based patterns to extract sentiment-rich features from Turney's study; these features are shown in Table 1. However, we observed that verbs can also contain sentiment information that is useful for sentiment analysis. Thus, we extended the rule set to extract more sentiment-bearing features, as shown in Table 2.

For example, such sentiment-rich bi-tagged features as '*like_VB movie_NN*,' '*feel_VBP good_JJ*,' or '*impressive_JJ looking_VBG*' were not extracted with Turney's rule set.

Table 1 Rules to extract Turney's features

N	First word	Second word	Example
1	JJ	NN/NNS	*technical_JJ directing_NN*
2	RB/RBR/RBS	JJ	*wonderfully_RB underplayed_JJ*
3	JJ	JJ	*strong_JJ supporting_JJ*
4	NN/NNS	JJ	*story_NN interesting_JJ*
5	RB/RBR/RBS	VB/VBD/VBG	*definitely_RB recommend_VB*

Table 2 Rules to extract verb-based features

N	First word	Second word	Example
1	VBN	NN/NNS	*born_VBN killers_NNS*
2	VB/VBG/VBP	JJ/JJR/JJS	*feel_VB good_JJ*
3	JJ	VBN/VBG	*good_JJ looking_JJ*
4	RB/RBR/RBS	RB/RBR/RBS	*most_RBS likely_RB*

Here, the tags JJ, JJR, and JJS indicate adjectives, NN and NNS indicate nouns, RB, RBR, and RBS indicate adverbs, and VB, VBG, and VBP indicate verbs. We used the Stanford POS tagger for tagging words in the sentence.

Dependency Features

An in-depth linguistic analysis in form of syntactic relations between words in a sentence can be useful for a sentiment analysis model. It has been shown in literature that syntactic patterns are effective for subjective detection [15–17]. We used dependency parse tree, constructed using the Stanford Parser, to capture long-distance information in the text.

For example, consider the sentence '*This movie is very nice*.' POS tags for this sentence are as follows: '*this_DT movie_NN is_VBZ very_RB nice_JJ*,' and dependency relations are as follows: det (movie-2, this-1), nsubj(nice-5, movie-2), cop(nice-5, is-3), advmod (nice-5, very-4), root(ROOT-0, nice-5). The basic features for this sentence are shown in Table 3.

Syntactic N-grams (sn-grams)

Syntactic N-grams (sn-grams) were used as features for a machine learning algorithm to learn patterns in the text for sentiment classification [28]. By dependency sn-gram, we understand a sub-tree of the dependency tree of a sentence that contains *n* nodes [29]. Syntactic N-grams can be used as features to represent sentences in the same scenarios as conventional n-grams; more specifically, sn-grams represent dependency trees as vectors in the same way as conventional n-grams represent strings of words. However, unlike conventional n-grams, sn-grams represent linguistic entities and are thus much more informative and less noisy. While sn-grams go a long way toward linguistically meaningful representations, numerous phenomena, ranging from the presence of functional words to synonymous expressions to insignificant details, still introduce noise in this representation and prevent semantically similar constructions to be mapped to identical feature vectors.

Syntactic N-grams convey information on syntactic relations between words in a sentence. In this feature extraction technique, words are grouped by following syntactic relations in the sentence, not by the linear order in which words appear in the text [29]. These features are extracted by following all the possible paths in the syntactic tree of the sentence to its leaf nodes. For example sentence, in the sentence 'The movie sounds interesting,' syntactic N-gram features are extracted as follows: sounds movie, sounds interesting, sounds movie the, sounds, movie, interesting. The syntactic tree for this example sentence is shown in Fig. 1. These features bring syntactic information in the machine learning models.

Main Algorithm

Our processing pipeline is as follows. First, we extract dependency relations between the words of the sentence. Then, we use these dependency relations to construct complex concepts that include semantics. After these concepts have been extracted, we obtain related common-sense knowledge from the ConceptNet lexical resource. Further,

Table 3 Example of various features

N	Feature type	Features
1	Unigram (F1)	this, movie, is, very, nice.
2	Bigram (F2)	this_movie, movie_is, is_very, very_nice,
3	Bi-tagged (F3)	very_nice
4	Dependency features (F4)	movie_this, nice_movie, nice_is, nice_very

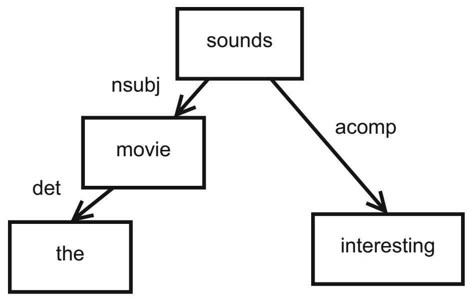


Fig. 1 Example: syntactic tree

only important concepts are selected and redundant concepts are eliminated using the mRMR feature selection technique.

We use the Stanford dependency parser to construct dependency relations between words [30]. Below, we first describe the use of the dependency relations to form concepts and then discuss how related common-sense knowledge is retrieved from ConceptNet. Then, we explain how the mRMR feature selection technique is used to filter irrelevant and redundant information. Figure 2 shows the various steps we followed to evaluate the various feature sets.

Formation of Concepts Using Dependency Relations

We form concepts on the basis of syntactic structures of sentences. A dependency relation can be described as a binary relation that consists of three types of elements: the relation type, the head word, and the dependent word. The type of the relation specifies what syntactic relation holds between the two words in the sentence. The head word of the relation is the pivot of the relation: The main syntactic and semantic properties of the pair are inherited from the head. The dependent word of the relation is the element that depends on the head word [31, 32].

We used the following rules.

Subject Noun Rule

Trigger: when the active token is found to be the syntactic subject of a verb.
Behavior: if a word h is in a subject noun relationship with a word t, then the concept (t-h) is extracted.
Example: *The movie is boring.*

Here, ‘movie’ is in a subject relation with ‘boring.’ The concept (boring movie) is extracted.

Joint Subject Noun and Adjective Complement Rule

Trigger: When the active token is found to be the syntactic subject of a verb and the verb is in an adjective complement relation with an adverb.
Behavior: If a word h is in a subject noun relationship with a word t, and the word t is in an adjective complement relationship with a word w, then the concept (w-h) is extracted.
Example: *The movie sounds interesting.*

Here, ‘movie’ is in a subject relation with ‘sounds’ and ‘sounds’ is in adjective complement relationship with ‘interesting.’ The concept (interesting movie) is extracted.

Direct Nominal Objects

This complex rule deals with direct nominal objects of a verb.

Trigger: When the active token is head verb of a direct object dependency relation.
Behavior: If a word h is in a direct nominal object relationship with a word t, then the concept (h-t) is extracted.
Example: *He saw the movie in 3D.*

Here, the system extracts the concept (see movie), while (see-in-3D) is not treated at this stage since it will be treated later by the standard rule for prepositional attachment.

Adjective and Clausal Complements Rules

These rules deal with verbs having as complements, either an adjective or a closed clause (i.e., a clause, usually finite, with its own subject).

Trigger: When the active token is head verb of one of the complement relations.
Behavior: If a word h is in a direct nominal object relationship with a word t, then the concept (h-t) is extracted.
Example: *The movie sounds boring.*

Here, ‘sounds’ is the head of a clausal complement dependency relation with ‘boring’ as the dependent. The concept (sound boring) is extracted.

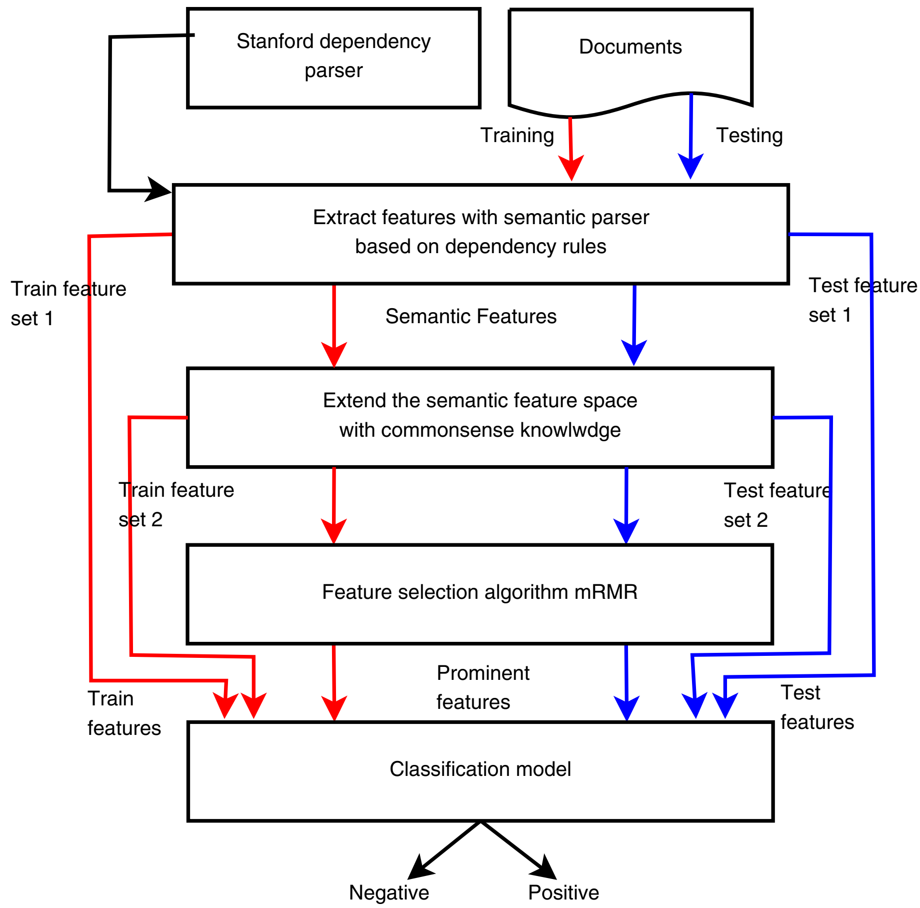
Negation

Negation is also a crucial component of natural language text, which usually flips the meaning of the text. This rule is used to identify whether a word is negated in the text.

Trigger: When in a text a word is negated.
Behavior: If a word h is negation by a negation marker t, then the concept (t-h) is extracted.
Example: *He does not like the movie.*

Here, ‘like’ is the head of the negation dependency relation with ‘not’ as the dependent; ‘like’ is negated by the negation marker ’not.’ The concept (not like) is extracted.

Fig. 2 Flow diagram of our approach to sentiment analysis



Open Clausal Complements

Open clausal complements are clausal complements of a verb that do not have their own subject, meaning that they (usually) share their subjects with that of the matrix clause. The corresponding rule is complex in the same way as the one for direct objects.

> **Trigger**: When the active token is the head of the relation.
> **Behavior**: As for the case of direct objects, the algorithm tries to determine the structure of the dependent of the head verb. Here, the dependent is itself a verb; therefore, the system tries to establish whether the dependent verb has a direct object or a clausal complement of its own. In a nutshell, the system is dealing with three elements: the head verb (h), the dependent verb(d), and the (optional) complement of the dependent verb (t). Once these elements have all been identified, the concept (h-d-t) is extracted.
> **Example**: *He likes to praise excellent movies.*

Here, 'like' is the head of the open clausal complements dependency relation with praise as the dependent and the complement of the dependent verb praise is movie. The concept (like praise movie) is extracted.

Modifiers Adjectival, Adverbial, and Participial Modification

The rules for items modified by adjectives, adverbs, or participles all share the same format.

> **Trigger**: These rules are activated when the active token is modified by an adjective, an adverb or a participle.
> **Behavior**: If a word w is modified by a word t, then the concept (t-w) is extracted.
> **Example**: *Paul is a bad loser.*

Here, the concept (bad loser) is extracted.

Prepositional Phrases

Although prepositional phrases (PP) do not always act as modifiers, we introduce them in this section as the distinction does not really matter for their treatment.

> **Trigger**: The rule is activated when the active token is recognized as typing a prepositional dependency relation. In this case, the head of the relation is the element to which PP attaches, and the dependent is the head of the phrase embedded in the PP.

Behavior: Instead of looking for the complex concept formed by the head and dependent of the relation, the system uses the preposition to build a ternary concept.
Example: *Bob hit Marie with a hammer.*

Here, the parser yields a dependency relation typed *prep_with* between the verb 'hit' and the noun 'hammer' (the head of the phrase embedded in the PP). The complex concept (hit with hammer) is extracted.

Adverbial Clause Modifier

This kind of dependency concerns full clauses that act as modifiers of a verb. Standard examples involve temporal clauses and conditional structures.

Trigger: The rule is activated when the active token is a verb modified by an adverbial clause. The dependent is the head of the modifying clause.
Behavior: If a word *t* is an adverbial clause modifier of a word *w*, then the concept (*t*-*w*) is extracted.
Example: *The machine slows down when the best games are playing.*

Here, the complex concept (play slow) is extracted.

Noun Compound Modifier

Trigger: The rule is activated when it finds a noun composed of several nouns. A noun compound modifier of an NP is any noun that serves to modify the head noun.
Behavior: If a noun word *w* is modified by another noun word *t*, then the complex concept (*t*-*w*) is extracted.
Example: *Battery life of this phone is not good.*

Here, the complex concept (battery life) is extracted.

Single-Word Concepts

Words having POS VERB, NOUN, ADJECTIVE, or ADVERB are also extracted from the text. Single-word concepts that exist in the multiword concepts are discarded as they carry redundant information. For example, since the concept 'party' already appears in the concept 'birthday party,' we discard the concept 'party.'

Obtaining Common-Sense Knowledge from ConceptNet

ConceptNet is a very large semantic network that includes a large number of common-sense concepts [33]. Common-sense knowledge in ConceptNet is contributed by Internet users. It is the largest publicly available common-sense knowledge base, which can be used to conduct various types of inference over text. It consists of nodes (concepts) connected by edges (relations between concepts). Examples of relationships between concepts in ConceptNet are IsA, EffectOf, CapableOf, MadeOf, and DesireOf [33]. After obtaining concepts from the text, we send them as queries to ConceptNet. From ConceptNet, we find common-sense knowledge related to the query concepts. For example, when we send the concept 'birthday party' as a query to ConceptNet, we obtain related concepts such as 'cake' or 'buy present.' From ConceptNet, we find the following relations:

1. Cake— AtLocation— birthday party,
2. Buy present —UsedFor— birthday party.

These common-sense concepts are used to gather more knowledge about the concepts as they have direct connections with 'birthday party.' From ConceptNet, we find that a cake is used at a birthday party and people buy a present for the birthday party. So, this process helps us acquire more knowledge about the concepts, which we extract using the methodology described in the previous section. Hence, joint exploitation of the extracted concepts and ConceptNet brings to the machine, a better understanding of natural language text [34]. Our approach enables the computer to understand the topic of the text, as well as the meaning conveyed by the text, to a better degree.

Optimal Feature Set Construction

Feature sets constructed with semantic parsing scheme and common-sense knowledge contain much noise and redundant information. We use Minimum Redundancy and Maximum Relevance (mRMR) feature selection technique to filter out irrelevant and noisy features to obtain optimal feature set.

The Minimum Redundancy and Maximum Relevance [11] technique is a filter-based feature selection technique that is used to select prominent features of a class. Feature selection is one of the main problems in machine learning; it identifies subsets of features that are highly correlated and strong enough to identify the class; this is termed maximum relevance. These subsets of features generally contain relevant features, but also include redundant features. The mRMR feature selection technique attempts to eliminate these redundant features; this is called minimum redundancy. When two relevant features together are redundant, the less important feature is dropped without compromising the performance of the classifier [11].

The mRMR feature selection technique selects the prominent features as follows:

– It selects features that are highly correlated with the class attribute (maximum relevance).

– Features are selected in such a way that they are less redundant, yet still highly correlated with the class attribute (minimum redundancy).

The mRMR feature selection technique uses MI to measure the correlation between features and class attributes. MI measures the nonlinear correlation between two attributes [35]. Consider a feature set $F = \{f_1, f_2, \ldots f_n\}$ containing n total features, and a class attribute C. The correlation, or relevance, of a feature f_m with a class attribute C can be expressed in terms of MI, i.e., the joint probability distribution $P(f_m, C)$ and the marginal probability distribution $P(f_m), P(C)$:

$$A = MI(f_m, C) = \sum_{m,C} P(f_m, C) \log \frac{P(f_m, C)}{P(f_m)P(C)} \tag{1}$$

Further, redundant features are eliminated by using the correlation between features:

$$B = MI(f_i, f_j) = \sum_{i,j} P(f_i, f_j) \log \frac{P(f_i, f_j)}{P(f_i)P(f_j)} \tag{2}$$

The mRMR feature selection technique selects features with the help of following two schemes, mutual information difference (MID) and mutual information quotient (MIQ):

$$MID = \max(A - B), \tag{3}$$

$$MIQ = \max\left(\frac{A}{B}\right). \tag{4}$$

In our experiments, we used the MID scheme, and prominent features are selected by maximizing A and minimizing B.

Example

The difference between our feature extraction method and other state-of-art methods can be illustrated with the following example. Consider the sentence: 'The movie sounds interesting.' Here, unigrams are all the unique words. The unigram features are simple bag-of-words features, which are unable to incorporate any semantic information. Bigrams features are able to extract some important information such as 'sounds interesting,' but they contain noise features such as 'the movie.' Finally, bi-tagged features contain useful information, but still lack much of the important information. Dependency and syntactic features represent syntactic information by considering the relations between words in the sentence, albeit with much noise. These features are also not able to adequately represent semantic information.

In contrast, our semantic parser scheme is able to keep more important and semantic features, such as 'interesting movie,' which is not extracted by any of the existing methods. Features extracted with the various feature extraction methods for our example sentence, 'The movie sounds interesting,' are presented in Table 4.

Table 4 Example: comparison of various feature extraction methods

Method	Features
Unigram	The, movie, sounds, interesting
Bigram	The movie, movie sounds, sounds interesting
Bi-tagged	Sounds interesting
Dependency features	Sounds movie, sounds interesting, movie the
Syntactic features	Sounds movie, sounds interesting, sounds movie the, sounds, movie, interesting
Our semantic parsing scheme	Movie, movie sound, interesting movie, sound interesting

Consider another sentence: 'I can even now remember the hour from which I dedicated myself to this great enterprise.' In this example, unigrams features are *I, can, even, now, remember, the, hour, from, which, dedicated, myself, to, this, great, enterprise*. Bigram features are extracted as pairs of adjacent words: *I can, can even, even now, now remember, remember the, the hour, hour from, from which, which I, I dedicated, dedicated myself, myself to, to this, this great, great enterprise*.

Next, dependency features extracted are as follows: *remember I, remember can, now even, remember now, hour the, remember hour, dedicated which, dedicated I, remember dedicated, dedicated myself, enterprise this, enterprise great, dedicated enterprise. Further, extracted syntactic N-grams are as follows: remember now, now even, remember hour, remember dedicated, dedicated enterprise, enterprise great, remember now even, remember hour dedicated, hour dedicated enterprise, dedicated enterprise great*.

However, concepts extracted by our concept parser are as follows: *even now, even now remember, remember hour, hour, remember from dedicate, dedicate which to enterprise, dedicate myself to enterprise, dedicate to enterprise, great enterprise*.

After sending the extracted concepts as queries to ConceptNet in order to acquire more common-sense knowledge, we obtain the following concept list: *even now, even now remember, remember hour, hour, remember from dedicate, dedicate which to enterprise, dedicate myself to enterprise, dedicate to enterprise, great enterprise, still, sixty minute*. Here, we find common-sense knowledge from ConceptNet: 'still,' 'sixty minute' are related to the concepts 'even now' and 'hour,' respectively.

From the above examples, we see that our concept parser is able to extract more semantic concepts: 'even now' and 'even now remember' extracted by our concept parser express more semantics in comparison with 'now even' and 'remember now even' extracted by syntactic n-grams and other feature extraction techniques.

Dataset, Experimental Setup, and Results

Dataset Used

To evaluate the performance of the proposed methods, we used one of the most popular publicly available movie review dataset. This standard dataset, known as the Cornell Movie Review Dataset, consists of 2,000 reviews, which contain 1,000 reviews labeled as positive and 1,000 reviews labeled as negative. The reviews were collected from the Internet Movie Database (IMDb) [36].

We also conducted more experiments using a product review dataset consisting of Amazon product reviews. This benchmark dataset, provided by Blitzer et al. [37], consists of reviews in various domains. We used product reviews of books, DVD, and electronics to evaluate the performance of our method. Each domain has 1,000 reviews labeled as positive and 1,000 reviews labeled as negative.

Evaluation Metrics

Usually, precision, recall, accuracy, and F-measure are used for evaluating performance of sentiment classification algorithms [35]. Precision for a class C is the fraction of total number of documents that are classified correctly in the total number of documents that were classified to the class C (the sum of true positives (TP) and false positives (FP)). Recall is the fraction of total number of correctly classified documents in the total number of documents that belong to class C: the sum of true positives and false negative (FN). F-measure is the combination of both precision and recall:

$$F\text{-measure} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (5)$$

F-measure is usually used as a cumulative measure to report the performance of classifiers for the sentiment classification.

Results and Discussion

From our experiments, we found that unigram features (F1) perform better than bigrams, bi-tagged, and dependency features for all datasets, see Table 5. For example, as shown in Table 5, the unigram feature set produces a F-measure of 84.2 %, in contrast to 78.8, 75.3, and 77.4 % for bigrams, bi-tagged, and dependency features, respectively, obtained with the SVM classifier on the movie review dataset. A possible reason for this could be that the bigram feature set contains many noise features, which are known to deteriorate the classification performance. In addition, the bigrams feature set is sparser than the unigram set; this again degrades the performance.

Next, bi-tagged features are polar in nature and important for sentiment classification, but these features are not able to include all the important information. They represent important information, but this information is not sufficient for sentiment classification; hence, they perform worst if used independently. Dependency features face the problem of over-generalization, which means that the features extracted with dependency rules are present in both classes, resulting in less discriminating features. Therefore, these features do not contribute much to classification. For example, 'this is not a very good movie' is a negative review and 'this is a good movie' is a positive review; however, both reviews contain many dependency features in common, such as 'movie good,' 'movie is,' and 'movie this.' These common features are less discriminating for classification purposes.

Besides, bigrams, bi-tagged, and dependency features are multiword features and thus suffer from the problem of sparsity, in contrast to unigrams [38]. Due to the data sparseness problem, the contribution of these features during classification by machine learning algorithms is not

Table 5 F-measure (in %) for various common-sense-based features on four datasets

	Movies	Books	DVD	Electronics
Unigrams	84.2	76.2	77.3	76.5
Bigrams	78.8	69.5	68.0	70.1
Bi-tagged	75.3	65.5	69.0	69.9
Dependency	77.4	71.3	69.2	71.2
Syntactic N-grams	87.3	84.6	84.1	85.8
With only Semantic parser	86.4	86.1	86.8	87.1
With Semantic parser and common-sense knowledge	88.9	86.9	86.2	87.0
With Semantic parser, common-sense knowledge, and mRMR feature selection	90.1 (+7.0 %)	88.5 (+16.1%)	89.2 (+15.3 %)	88.9 (+16.2 %)

as effective as expected [38]. In addition to data sparseness, simple bigram and dependency features contain many noise features; this reduces the performance of machine learning algorithms. Table 5 presents comparative results with respect to F-measure for all basic features and prominent features. The performance of unigram features with the SVM classifier is considered as baseline [15, 38].

Furthermore, syntactic N-grams were used as features for the machine learning algorithm. These features contain syntactic information for the words in a sentence. Syntactic N-grams are more informative and less arbitrary as compared to simple N-grams. These features give better results as compared to unigrams, bigrams, and dependency features. For example, syntactic N-gram features give F-scores of 87.3 and 84.6 % for movie and book review datasets, respectively, which is better than that for other features as shown in Table 5. Still, syntactic N-grams are not able to represent semantic information, which is intuitively useful for sentiment analysis.

Finally, our semantic parsing scheme was used to build machine learning model for sentiment classification. Our semantic parser produces better results as compared with other feature sets explored so far. Our semantic feature extraction method produces F-measure of 86.4 % with the SVM classifier on the movie review dataset, as shown in Table 5. Semantic parser-based features with common-sense knowledge improve the results, giving a F-measure of 88.9 % with the SVM classifier on the same dataset.

A possible reason for this observation is that ConceptNet includes important semantic information associated with the concepts. However, the feature set constructed with concepts extracted by the semantic parser in combination with the common-sense does not contribute as much as we expected to the sentiment classification of review documents. This is due to the fact that concepts extracted from ConceptNet with common-sense knowledge contain, along with important information, a significant amount of noise in the form of irrelevant and redundant features.

Therefore, when irrelevant and redundant features are eliminated from this feature vector using the mRMR feature selection technique, the obtained feature set outperforms all other features. For example, our concept feature set produces z F-measure of 90.1 % with the SVM classifier, for the movie review dataset. A possible reason for this is that we extract features that contain more useful sentiment information and also contain ConceptNet-based knowledge, which was not present in previous feature sets. In our approach, we present near-paraphrastic rules that simplify and normalize the dependency trees in order to reduce synonymous variation and remove insignificant details. These, in turn, improve similarity between feature vectors of semantically similar expressions and reduce data sparseness.

Another difference with previous methods is that syntactic N-grams convey all characteristics of basic N-gram, whereas our concept parser extracts semantics from the text. In addition, the mRMR feature selection technique was also used to eliminate redundant information and select only information important for classification.

Comparison with Existing Methods

A performance comparison of our approach with state-of-art results reported in the literature is shown in Table 6. Researchers have mostly used the movie review and product review datasets to evaluate their methods for sentiment analysis. As can be seen from Table 6, we achieve results competitive with previous works on both the movie review and product review datasets. Our proposed semantic parsing scheme with common-sense knowledge and the mRMR feature selection method outperformed other methods reported in the literature on the same dataset.

However, the results are not directly comparable due to different experimental settings used by the researchers with respect to evaluation measures, tools, and preprocessing methods. For example, Matsumoto et al. [15], as part of the preprocessing step, employed unigram patterns that appear in at least two distinct sentences in the dataset. They reported 87% accuracy for unigram features (simple bag-of-words features), whereas we report 84.2% accuracy in our experiment for the same unigram feature set with a similar linear SVM classifier. In addition, they reported a performance accuracy of 85.1% for bigram features, which is significantly better than the 78.8% produced by our approach, and 77.1% reported by Pang et al. [14] for the same bigram features.

The reason for this is the difference in experimental settings used by the respective researchers. For example, Matsumoto et al. considered only bigrams that occur at least twice in the dataset, in contrast to Pang et al. [14], who considered all bigrams that occur in at least seven documents. In addition, Matsumoto et al. [15] employed the Charniak parser to obtain dependency relations, whereas we used the Stanford dependency parser, whereas Ng. et al. [41] used the MINIPAR dependency parser. In addition, researchers use different machine learning tools to construct their models, for example, LibSVM, SVM*light*, or WEKA.

Further, Matsumoto et al. [15] constructed their sentiment analysis model using various features, such as unigrams, word subsequence, and a dependency sub-tree. They also used the Frequent Pattern Mining software FREQT, available at http://www.chasen.org/~taku/software/, to mine frequent dependency tree patterns, and developed an SVM classifier by combining all of these features for sentiment classification. Their approach is computationally very expensive, as the feature vector

Table 6 Comparison with existing methods

Paper	Approach	Machine learning algorithm	Dataset	Best accuracy, %
Pang et al. [36] (2004)	Minimum cut algorithm	SVM	Movie reviews	87.1
Prabowo et al. [39] (2009)	Hybrid (rules + closeness measure + SVM)	SVM	Movie reviews	87.3
Mullen et al. [22] (2004)	Hybrid method with Turney and Osgood values	Hybrid SVM	Movie reviews	86
O'keefee et al. [40] (2009)	SentiWordNet based features and feature selection	SVM, NB	Movie reviews	87.15
Xia et al. [18] (2011)	Ensemble features	Ensemble classifier	Movie reviews, books, DVD, electronics	88.0, 83.0, 83.8, 86.0
Xia et al. [38] (2010)	Ensemble and generalized dependency features	SVM, NB	Movie reviews	88.6
Ng et al. [41] (2006)	Various dependency features and polarity information	SVM	Movie reviews	90.5
Matsumoto et al. [15] (2005)	Various features such as unigrams, word subsequence and dependency subtree with frequent mining algorithm	SVM	Movie reviews	92.9
Tu et al. [42] (2012)	Dependency forest-based features	MaxEnt	Movie reviews	91.6
Dang et al. [25] (2010)	Selected content-free, content-specific and sentiment features	SVM	Books, DVD, electronics	78.85, 80.75, 83.75
Abbasi et al. [43] (2008)	Genetic algorithms (GA), Information gain (IG), IG + GA	SVM	Movie reviews	91.7
Abbasi et al. [44] (2010)	Stylistic and syntactic features	SVM	Movie reviews	87.9
Our approach	Dependency parsing-based semantic parser with common-sense knowledge; mRMR	SVM	Movie reviews, books, DVD, electronics	90.1, 88.5, 89.2, 88.9

length in their approach is huge due to the ensemble of variety of features. It is also computationally very expensive to apply the frequent pattern mining algorithm with dependency sub-tree features.

In contrast, we employed the mRMR feature selection technique to reduce the feature vector size; this resulted in a less computationally expensive procedure. In addition, Matsumoto et al. [15] reported that their method has the drawback of having overlap of features among word subsequence and dependency features. Feature set constructed with their method contains noisy and redundant information. On the other hand, the feature set constructed with our approach contains more important information due to the use of ConceptNet, while information redundancy is reduced through mRMR feature selection. Matsumoto et al. [15] reported a best performance of 92.9%, which improves the baseline by +6.6%, whereas our approach gives a best accuracy of 90.1%.

Ng et al. [41] considered top-ranked features using weighted log-likelihood ratio (WLLR) feature selection from unigrams, bigrams, trigrams, and dependency features. They employed a manually constructed term polarity lexicon to select more polar bigrams and dependency features. They also used 2000 additional non-review documents to increase the list of objective features. Finally, they used the objective feature list to remove objective features from their original feature set. Their approach is rather complex, and the main drawback of their approach is that it requires human intervention. In contrast, our approach is fully automatic and significantly simpler. Ng et al. [41] reported a best performance of 90.5%, which improves the baseline 87.1% by +3.4%, while our approach gives a best accuracy of 90.1%.

Xia et al. [18] experimented with various ensemble feature sets and ensemble classifiers to identify the best feature set and classifier for sentiment analysis on several datasets. Their approach faces the problem of noisy features due to the use of an ensemble of features. Xia and Zong [38] used various ensemble features and generalized dependency features for sentiment analysis. In particular, they focused on the problem of data sparseness by generalizing the dependency features. However, their approach is limited due to the problem of noisy features and overgeneralization of features.

Abbasi et al. [44] presented a sentiment analysis approach that filters irrelevant features with the help of a hybrid feature selection method. Their feature selection method is a combination of a genetic algorithm and IG. A drawback of their approach is that it is highly computationally expensive to use a GA for feature selection. Tu et al. [42] proposed a technique to extract complex features using dependency forest in combination with unigrams, which is, again, computationally expensive. They employed tenfold cross-validation to evaluate the performance of their method with the MaxEnt classifier.

In summary, we find that our proposed semantic parsing scheme with common-sense knowledge and the mRMR feature selection method outperforms other methods described in the literature on the same dataset.

Conclusions and Future Work

Performance of sentiment analysis crucially depends on the effectiveness of the feature extraction process. We have presented a novel feature extraction method that uses the dependency relation between words to extract features from text. The joint exploitation of extracted concepts and ConceptNet enabled us to acquire more knowledge and hence a better understanding of the text.

In addition, we employed the mRMR feature selection technique to select the important concepts and to eliminate redundant information. All experiments were performed on the movie review dataset and the product review datasets (books, DVDs, electronics). Our experimental results demonstrate that our approach outperforms other existing approaches to sentiment analysis.

Possible directions for future work include discovering more useful dependency relationships to mine the concepts. Along with ConceptNet, other ontologies can help further enrich the concept mining process. We also plan to use a supervised fuzzy learning algorithm to enhance the performance of the system [45].

References

1. Cambria E, Hussain A. Sentic album: content, concept, and context-based online personal photo management system. Cogn Comput. 2012;4(4):477–96.
2. Poria S, Cambria E, Hussain A, Huang G-B. Towards an intelligent framework for multimodal affective data analysis. Neural Netw. 2015;63:104–16.
3. Howard N, Cambria E. Intention awareness: improving upon situation awareness in human-centric environments. Hum Centric Comput Inf Sci. 2013;3(9):1–17.
4. Cambria E, Hussain A, Durrani T, Havasi C, Eckl C, Munro J. Sentic computing for patient centered applications. In: IEEE ICSP, Beijing; 2010. p. 1279–1282.
5. Cambria E, Hussain A, Havasi C, Eckl C, Munro J. Towards crowd validation of the UK national health service. In: ACM WebSci, Raleigh; 2010.
6. Liu B. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies. San Rafael: Morgan & Claypool Publishers; 2012.
7. Erik C, Amir H, Catheine H, Chris E. Common sense computing: From the society of mind to digital intuition and beyond. In: Lecture notes in computer science 5707, Springer; 2009. p. 252–259.
8. Cambria E, Gastaldo P, Bisio F, Zunino R. An ELM-based model for affective analogical reasoning. Neurocomputing. 2015;149: 443–55.
9. Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res. 2003;3(1):1157–82.
10. Agarwal B, Mittal N. Enhancing performance of sentiment analysis by semantic clustering of features. IETE J Res. 2014;60(6): 414–22.
11. Peng H, Long F, Ding C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell. 2005;27(8):1226–38.
12. Agarwal B, Mittal N. Prominent feature extraction for review analysis: an empirical study. J Exp Theor Artif Intell. 2014. doi:10.1080/0952813X.2014.97783.
13. Hougue N, Bhattacharyya DK, Kalita JK. MIFS-ND: a mutual information-based feature selection method. Expert Syst Appl. 2014;41(4):6371–85.
14. Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of EMNLP; 2002. p. 79–86.
15. Matsumoto S, Takamura H, Okumura M. Sentiment classification using word sub-sequences and dependency sub-trees. In: Proceedings of PAKDD; 2005. p. 301–311.
16. Pak A, Paroubek P. Text representation using dependency tree sub-graphs for sentiment analysis. DASFAA workshop; 2011. p. 323–332.
17. Nakagawa T, Inui K, Kurohashi S. Dependency tree-based sentiment classification using CRFs with hidden variables. In: Proceeding HLT '10 human language technologies: annual conference of the North American chapter of the association for computational linguistics; 2010. p. 786–794.
18. Xia R, Zong C, Li S. Ensemble of feature sets and classification algorithms for sentiment classification. J Inf Sci. 2011;181(6): 1138–52.
19. Riloff E, Patwardhan S, Janyce W. Feature subsumption for opinion analysis. In: EMNLP; 2006. p. 440–448.
20. Joshi M, Penstein-Rose C. Generalizing dependency features for opinion mining. In: ACL; 2009. p. 313–316.
21. Mejova Y, Srinivasan P. Exploring feature definition and selection for sentiment classifiers. In: Proceedings of the fifth international AAAI conference on weblogs and social media; 2011. p. 546–549.
22. Mullen T, Collier N. Sentiment analysis using support vector machines with diverse information sources. In: EMNLP; 2004. p. 412–418.
23. Osgood CE, Succi GJ, Tannenbaum PH. The measurement of meaning. Champaign: University of Illinois Press; 1957.
24. Turney PD. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. ACL-2002; 2002. p. 417–424.
25. Dang Y, Zhang Y, Chen H. A lexicon enhanced method for sentiment classification: an experiment on online product reviews. IEEE Intell Syst. 2010;25(4):46–53.
26. Gelfand B, Wulfekuler M, Punch WF. Automated concept extraction from plain text. In: AAAI workshop on text categorization; 1998. p. 13–17.
27. Hatzivassiloglou V, McKeown KR. Predicting the semantic orientation of adjectives. In: ACL; 1997. p. 174–181.
28. Sidorov G, Velasquez F, Stamatatos E, Gelbukh A, Chanona-Hernandez L. Syntactic n-grams as machine learning features for

natural language processing. Expert Syst Appl. 2014;41(3): 853–60.
29. Sidorov G. Non-continuous syntactic n-grams. Polibits. 2013;48(1): 67–75.
30. de Marneffe M-C, Manning CD. The stanford typed dependencies representation. In: Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation. Association for Computational Linguistics; 2008. p. 1–8.
31. Soujanya P, Basant A, Alexander G, Amir H, Newton H. Dependency-based semantic parsing for concept-level text analysis. In: CICLing 2014; 2014. p. 113–27.
32. Poria S, Cambria E, Winterstein G, Huang G-B. Sentic patterns: dependency-based rules for concept-level sentiment analysis. Knowl Based Syst. 2014;69:45–63.
33. Havasi C, Speer R, Alonso JB. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In: Recent advances in natural language processing; 2007. p. 27–29.
34. Wang Q-F, Cambria E, Liu C-L, Hussain A. Common sense knowledge for handwritten Chinese text recognition. Cogn Comput. 2013;5(2):234–42.
35. Manning CD, Raghvan P, Schutze H. Introduction to information retrieval. Cambridge: Cambridge University Press; 2008.
36. Pang B, Lee L. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the association for computational linguistics (ACL); 2004. p. 271–278.
37. Blitzer J, Dredze M, Pereira F. Biographies, bollywood, boomboxes and blenders: domain adaptation for sentiment classification. In: ACL; 2007. p. 440–447.
38. Xia R, Zong C. Exploring the use of word relation features for sentiment classification. In: COLING-2010; 2010. p. 1336–1344.
39. Prabowo R, Thelwall M. Sentiment analysis: a combined approach. J Informetr. 2009;3(2):143–57.
40. O'Keefe T, Koprinska I. Feature selection and weighting methods in sentiment analysis. In: Proceedings of the 14th Australasian document computing symposium, Sydney, Australia, ACL; 2009.
41. Ng V, Dasgupta S, Arifin SM. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In: Proceedings of the COLING/ACL 2006 main conference poster sessions; 2006. p. 611–618.
42. Tu Z, Jiang W, Liu Q, Lin S. Dependency forest for sentiment analysis. In: First CCF conference, natural language processing and Chinese computing; 2012. p 69–77.
43. Abbasi A. Intelligent feature selection for opinion classification. IEEE Intell Syst. 2010;25(4):75–9.
44. Abbasi A, Chen H, Salem A. Sentiment analysis in multiple languages: feature selection for opinion classification in Web forums. ACM Trans Inf Syst. 2008;26(3):12.
45. Poria S, Gelbukh A, Cambria E, Hussain A, Huang G-B. EmoSenticSpace: a novel framework for affective common-sense reasoning. Knowl Based Syst. 2014;69:108–23.