Soujanya Poria
School of Natural Sciences, University of Stirling, UK
Erik Cambria
School of Computer Engineering, Nanyang Technological University, SINGAPORE
Alexander Gelbukh
Centro de Investigación en Computación, Instituto Politécnico Nacional, MEXICO
Federica Bisio
DITEN, University of Genoa, ITALY
Amir Hussain
School of Natural Sciences, University of Stirling, UK

Sentiment Data Flow Analysis by Means of Dynamic Linguistic Patterns

Abstract—Emulating the human brain is one of the core challenges of computational intelligence, which entails many key problems of artificial intelligence, including understanding human language, reasoning, and emotions. In this work, computational intelligence techniques are combined with common-sense computing and linguistics to analyze sentiment data flows, i.e., to automatically decode how humans express emotions and opinions via natural language. The increasing availability of social data is extremely beneficial for tasks such as branding, product positioning, corporate reputation management, and social media marketing. The elicitation of useful information from this huge amount of unstructured data, however, remains an open challenge. Although such data are easily accessible to humans, they are not suitable for automatic processing: machines are still unable to effectively and dynamically interpret the meaning associated with natural language text in very large, heterogeneous, noisy, and ambiguous environments such as the Web. We present a novel methodology that goes beyond mere word-level analysis of text and enables a more efficient transformation of unstructured social data into structured information, readily interpretable by machines. In particular, we describe a novel paradigm for real-time concept-level sentiment analysis that blends computational intelligence, linguistics, and common-sense computing in order to improve the accuracy of computationally expensive tasks such as polarity detection from big social data. The main novelty of the paper consists in an algorithm that assigns contextual polarity to concepts in text and flows this polarity through the dependency arcs in order to assign a final polarity label to each sentence. Analyzing how sentiment flows from concept to concept through dependency relations allows for a better understanding of the contextual role of each concept in text, to achieve a dynamic polarity inference that outperforms state-of-the-art statistical methods in terms of both accuracy and training time.

Digital Object Identifier 10.1109/MCI.2015.2471215
Date of publication: 13 October 2015

Corresponding Author: Erik Cambria (Email: cambria@ntu.edu.sg).

 1556-603X/15©2015IEEE

©ISTOCKPHOTO.COM/LINEARCURVES

In this work, we propose a novel brain-inspired sentiment analysis framework to help machines emulate human inference of sentiment from natural language. By merging, for the first time, CI, linguistics, and common-sense computing [1], the proposed paradigm exploits the relations between concepts and linguistic patterns in text to reveal the flow of sentiment from concept to concept, and improve our understanding of how sentiment is conveyed in a sentence.

Besides tackling the challenge of emulating the way the human brain decodes semantic and affective meaning from text, the proposed paradigm aims to improve human-computer interaction in several real-world applications related to the fast-evolving Web environment, which is increasingly interwoven with unstructured social data. In recent years, the opportunity to capture opinions from the general public has raised increasing interest both in the scientific community, for the many exciting open challenges, and in the business world, for remarkable fallouts in marketing and financial prediction. However, efficient handling of opinion mining at the huge scale of the contemporary Web with all its diversity and noise require robust models and algorithms for the elicitation of opinions from text. Existing algorithms can efficiently find documents, split them into parts, check the spelling, and count their words. However, such algorithms are very limited when it comes to dynamically interpreting sentences and extracting meaningful information. When facing the open Internet where needles of useful information are buried in haystacks of ever-growing flood of data and texts, natural language processing (NLP) is in the need to fast "jump the curve" [2] to catch up with new challenges. Relying on arbitrary keywords, punctuation and word co-occurrence frequencies have worked fairly well so far, but the explosion of Web contents and the outbreak of deceptive phenomena such as Web trolling and opinion spam are increasingly exposing the inefficiency of conventional NLP algorithms. In order to effectively extract and dynamically manipulate text meanings, a truly intelligent NLP system must have access to a significant amount of knowledge about the real noisy ambiguous world and the domain of time-varying discourse.

In this paper, a novel rule-based sentiment flow algorithm is proposed. This algorithm extracts sentiment directly from existing lexical resources such as SenticNet for those words and

I. Introduction

Computational intelligence (CI) is an established research field that focuses on using brain- or nature-inspired models to solve complex real-world problems that cannot otherwise be effectively solved by traditional models. One of the biggest challenges of CI is the emulation of human intelligence, which includes not only human-like reasoning but also human interaction, e.g., human language and human emotions. Emotions, sentiment, and judgments on the scale of good—bad, desirable—undesirable, approval—disapproval are essential for human-to-human communication. Understanding human emotions and deciphering the way humans reason about them or express them in their language is key to enhancing human-machine interaction.

The electronic circuit metaphor: sentiment words are "sources" while other words are "elements", e.g., *very* is an amplifier, *not* is a logical complement, *rather* is a resistor, *but* is an OR-like element that gives preference to one of its inputs.

concepts that possess their own intrinsic polarity and are present in existing lexical resources. Then, the algorithm applies valence shifters and combines sentiment of individual words of the sentence in a way similar to electronic circuits, where the signal from the sources undergoes amplification, inversion, weakening, and signals from different sources are combined in a non-trivial way. In addition, the proposed method involves a back-up machine-learning technique, which works in those cases where no sufficient information is found in existing lexical resources.

The novelty of this paper consists in introducing new patterns as compared with our previous work [3], in an improved machine learning module, which shows better results, and, most importantly, in the radical change of the logical structure of how patterns are applied to the sentences. The novel algorithm assigns contextual polarity to the words and flows this polarity through the dependency arcs in order to assign the final polarity label to the sentence. This algorithm of activation of the rules, termed sentiment flow, is now considerably more general and clear, and leads to increased accuracy.

The rest of the paper is organized as follows: Section II presents related work in the context of using linguistic patterns for dynamic opinion mining in noisy ambiguous environments; Section III illustrates the dependency-based rules for sentence-level polarity detection; Section IV, the central section of the paper, presents the novel algorithm for activation of the rules, termed sentiment flow algorithm; Section V presents the CI techniques developed to overcome the limitations of the common-sense knowledge base; Section VI gives the evaluation results and discussion; finally, Section VII concludes the paper and suggests directions for future work.

II. Related Work

Sentiment analysis systems can be broadly categorized into knowledge-based [4] or statistics-based systems [5]. While, initially, the use of knowledge bases was more popular for the identification of emotions and polarity in text, recently sentiment analysis researchers have been increasingly using statistics-based approaches, with a special focus on supervised statistical methods. For example, Pang et al. [6] compared the performance of different machine learning algorithms on a movie review dataset: using a large number of textual features they obtained 82.90% of accuracy. A recent approach by Socher et al. [7] obtained even better accuracy (85%) on the same dataset using a recursive neural tensor network (RNTN).

Other unsupervised or knowledge-based approaches to sentiment analysis include Turney et al. [8], who used seed words to calculate the polarity and semantic orientation of phrases, Melville et al. [9], who proposed a mathematical model to extract emotional clues from blogs and then used those information for sentiment detection.

Sentiment analysis research can also be categorized as single-domain [6], [8] versus cross-domain [10]. The work presented in [11] discusses spectral feature alignment to group domain-specific words from different domains into clusters and reduce the gap between domain-specific words of two domains using domain independent words. Bollegala et al. [12] developed a sentiment-sensitive distributional thesaurus by using labeled training data from source domain and unlabeled training data from both source and target domain. Some recent approaches [13], [14] used SentiWordNet [15], a very large sentiment lexicon developed by automatically assigning a polarity value to WordNet [16] synsets. In SentiWordNet, each synset has three sentiment scores along three sentiment dimensions: positivity, negativity, and objectivity.

In this paper, we propose a novel sentiment analysis framework which incorporates CI, linguistics, and commonsense computing in an attempt to better understand sentiment orientation and flow in natural language text. One of the earliest ideas in the study of linguistic patterns was carried out by [17]. Their idea relies on discovering additional adjective sentiment words using some seed adjective sentiment words. A set of linguistic rules based on the connectives ('and', 'or', 'but', 'either/or', 'neither/nor') and discourse was proposed to find the additional sentiment bearing adjective words. Negation plays a major role in detecting the polarity of sentences [18]. In [18], Jia et al. carried out an experiment to identify negations in text using linguistic clues and showed a significant performance improvement over the state-of-the-art. However, when negations are implicit, i.e., cannot be recognized by an explicit negation identifier, irony detection should be considered as well [19]. In [19], three conceptual layers, each of which consists of eight textual features, were proposed to grasp the implicit negations. A method to exploit discourse relations for the detection of tweets polarity was proposed in [20]. The authors showed how conjunctions, connectives, modals and conditionals might affect sentiments in tweets.

Liu et al. [21] proposed a collection of opinion rules, defined as a concept that implies a positive or negative sentiment. Moilanen et al. [22] also introduced the notion of sentiment conflict. When two words having different sentiment polarity form a word together, e.g., 'terribly good' then it is called sentiment conflict. Narayanan et al. [23] aimed to analyze the sentiment polarity of conditional sentences, studying the linguistic structure of such sentences and applying supervised learning models for dynamic classification.

Text representation is the key task for any text classification framework. The bag-of-words (BoW) model looks for surface word forms and does not consider the semantic and contextual clues in the text. Most of the well-known techniques have focused on BoW representation for text classification [24], [25]. To overcome the problem of limited capability in grasping semantic clues, some existing related works relied on using knowledge bases [26], [27]. The bag-of-concepts (BoC) model leverages on representing text as a conceptual vector rather than relying on the terms in the text. For example, if a text contains "red" and "orange," then BoC models them as the concept "color," i.e., BoC looks for hyponym. The BoC model was first proposed by Sahlgren et al. [28] to enhance the performance of support vector machine (SVM) in text categorization tasks. According to their method, concepts are the synonym sets of BoW. Among recent approaches on the BoC model, the approach by Wang et al. [29] presented the idea of concept as a set of entities in a given domain, i.e., words belonging to similar classes have similar representation. If a text contains "Jeep" and "Honda", this can be conceptualized by the concept "car". On the basis of their study, we identify two major advantages of the BoC model:

- *Replacement of surface matching with semantic similarity*: the BoC model calculates semantic similarity between words and multi-word expressions from a higher concept level.
- *Tolerance to new terms*: In text, new terms are always encountered, but new concepts may not always arise. Once BoC models concepts for a category, it can handle new words under that category. This shows the strong adaptability of the BoC model with respect to word changes.

Zhang et al. [30] discussed semantic classification on a disease corpus. Though their approach does not focus on the BoC model, they attempted to capture semantic information from text at the highest level. According to them, the use of contextual semantic features along with the BoW model can be very useful for semantic text classification. Wu et al. [31] built a sentiment lexicon using a common-sense knowledge base, ConceptNet. Using the hypothesis that concepts pass their sentiment intensity to their neighbors based on the relations connecting them, they constructed an enriched sentiment lexicon able to produce better performance in the sentiment polarity classification task.

III. Sentic Pattern Rules

Sentic pattern rules are a recently suggested semantic technique for advanced analysis of text with the aim of determining its polarity [3]. In this section, we introduce the main idea of the sentiment flow algorithm based on such rules and give examples of novel rules, which are used in conjunction with the ones previously described in [3]. In the next section, the sentiment flow algorithm will be described in more detail.

A. Resource Used: SenticNet

The proposed polarity detection algorithm retrieves the polarity scores of concepts from SenticNet [32]. SenticNet is a sentiment lexicon, freely available as a knowledge base that contains polarity scores of individual words and multi-word expressions.

The version of SenticNet used was 3.0[1]. This common-sense knowledge base contains 30 thousand affective concepts, such as `celebrate_special_occasion`, `make_mistake`, and `feel_happy`. It was built using semantic multidimensional scaling [33]. The knowledge base is available in the form of RDF XML files, as well as through a web-based API[2].

B. Sentiment Flow Algorithm: An Informal Description

Consider the sentence

(1) The car is very old but it is rather not expensive.

A baseline approach to polarity detection consists in counting positive and negative words in the sentence, using a polarity lexicon. In this example, as it is typical for any text, most of the words, such as "car" or "very", do not have any intrinsic polarity. Suppose the polarity lexicon contains two words: "old" and "expensive", both with negative polarity, at least in the context of buying a car. Thus, the total, or average, polarity of the words in this sentence is negative.

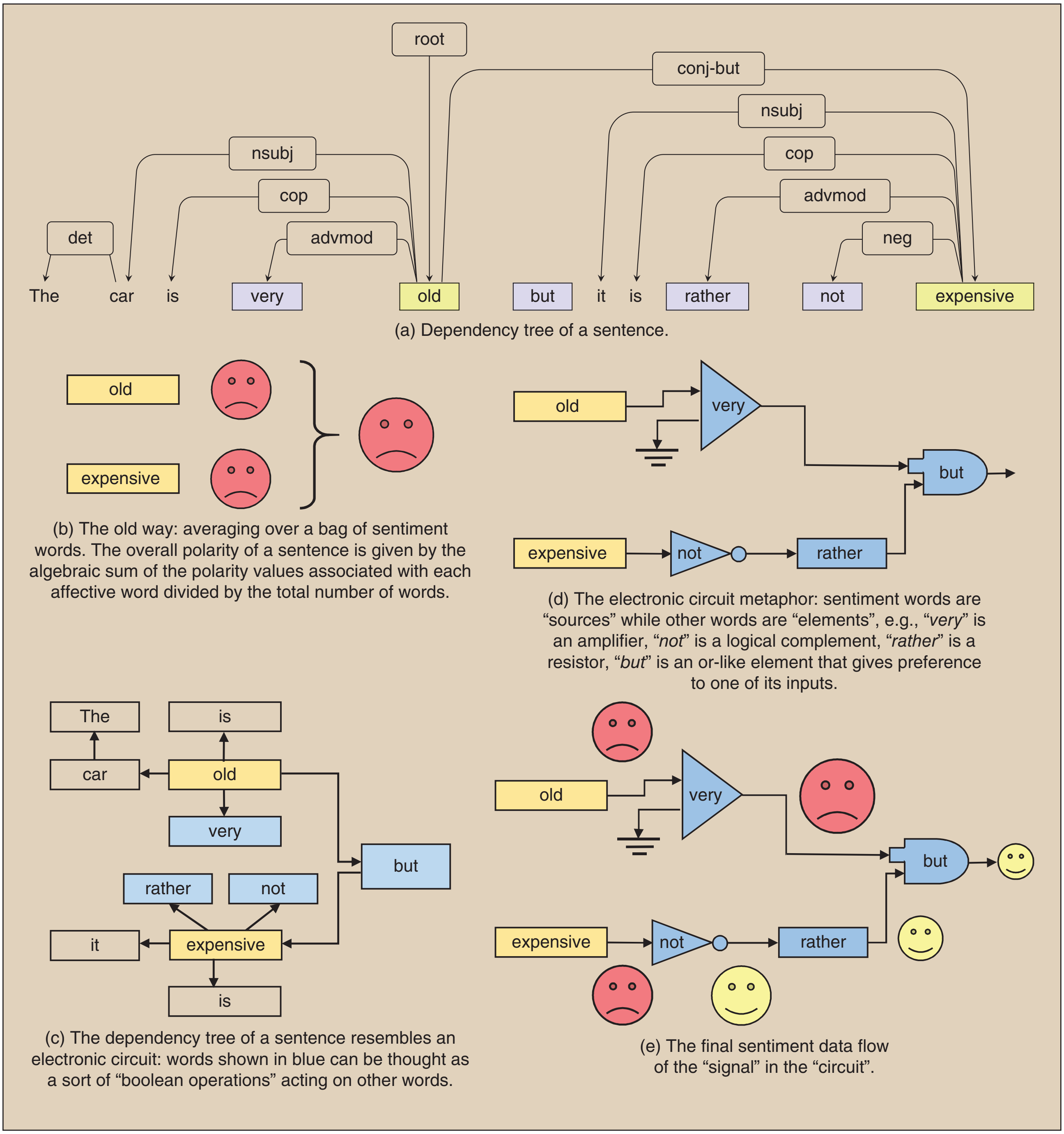
The main idea behind the use of sentic patterns [3] for calculating the overall polarity of a sentence can be best illustrated by analogy with an electronic circuit, in which few "elements" are "sources" of the charge or signal, while many elements operate on the signal by transforming it or combining different signals. This implements a rudimentary type of semantic processing, where the "meaning" of a sentence is reduced to only one value—its polarity.

Sentic patterns are applied to the dependency syntactic tree of the sentence, such as the one shown in Figure 1(a). The only two words that have intrinsic polarity are shown in yellow color; the words that modify the meaning of other words in the manner similar to contextual valence shifters [34] are shown in blue. A baseline that completely ignores the structure, as well as words that have no intrinsic polarity, is shown in Figure 1(b): the only two words in this sentence are negative, so the total is negative. However, the syntactic tree can be re-interpreted in the form of a "circuit" where the "signal" flows from one element, or subtree, to another, as shown in Figure 1(b): the word "combines" two parts of the sentence, each one, in this case having a polarity words combined with modifiers. Removing the words not used for calculation (shown by white rectangles) and re-structuring the tree, a sequence of transformations shown in Figure 1(d) is obtained. In blue color shown are elements resembling electronic amplifier, electronic logical negation, a resistor, and a kind of an asymmetric logical "and".

Figure 1(e) illustrates this idea at work: the sentiment flow from polarity words through shifters and combining words. The two polarity-bearing words in this example are negative. The negative effect of the word "old" is amplified by the intensifier "very". However, the negative effect of the word "expensive" is inverted by the negation, and the resulting positive value is decreased by the "resistor". Finally, the values of the two phrases are combined by the word "but", so that the overall polarity still has the same sign as that of the second component, but further decreased in value by the first component. Note that the effect of the conjunction would be opposite for, for example, "although":

[1] http://sentic.net/senticnet-3.0.zip

[2] http://sentic.net/api/



(a) Dependency tree of a sentence.

(b) The old way: averaging over a bag of sentiment words. The overall polarity of a sentence is given by the algebraic sum of the polarity values associated with each affective word divided by the total number of words.

(c) The dependency tree of a sentence resembles an electronic circuit: words shown in blue can be thought as a sort of "boolean operations" acting on other words.

(d) The electronic circuit metaphor: sentiment words are "sources" while other words are "elements", e.g., "*very*" is an amplifier, "*not*" is a logical complement, "*rather*" is a resistor, "*but*" is an or-like element that gives preference to one of its inputs.

(e) The final sentiment data flow of the "signal" in the "circuit".

FIGURE 1 The main idea behind sentic patterns: the structure of a sentence is like an electronic circuit where logical operators channel sentiment data-flows to output an overall polarity.

(2) The car is very old although it is rather not expensive.

would yield negative polarity, though softened by the second component.

C. Examples of Sentic Pattern Rules

The rules are represented in the form of a description of the context that triggers the rule and the action undertaken when such a context is found. Here are several examples. The algorithm, described in the next section, considers words of the sentence in a specific order. The word under consideration is referred to as the active token.

The simplest rules are those that treat the complementation relations. Examples of such rules for direct nominal objects, adjectival and clausal complements, and open clausal complements can be found in [3]. For example, the direct nominal object rule is applied in the following situations:

Example 1: "*You made me pissed off.*" Since the concept `(make, pissed)` is not found in SenticNet, the dynamic sentic patterns look up the polarity of "*make*" and "*pissed*" separately. As "*make*" is not in SenticNet, the polarity of "*pissed*" is considered as the polarity of the sentence (which is negative).

Example 2: "*John loves this boring movie.*" The polarity of "*love*" is positive and the polarity of "*movie*" is negative as it is modified by a negative modifier "*boring.*" Sentic patterns set the polarity of this sentence as negative as the speaker says it is a boring movie though the subject, John, loves it. This rule has an exception when the subject is first person, i.e., the subject is the same as the speaker.

Example 3: "*You have hurt the cute cat.*" The word "*hurt*" has negative polarity and the polarity of "*cat*" is positive as it has a positive modifier "*cute.*" Thus, the polarity of the sentence is negative.

In this work, we introduce new rules for complementation relation. Below are three examples of such rules.

1) *Subject nouns*:

Trigger: The active token d is a noun that is in the subject relation with a verb h.

Behavior: First, the complete concept consisting of both words, `(h,d)`, is looked up in SenticNet. If it is found there, then the resulting polarity of the construction is directly retrieved from SenticNet and is assigned to the relation. Otherwise the following options are tried, in the given order:

- ❏ If the sentence is in passive voice and h and d are both negative, then the subject noun relation between h and d yields positive sentiment. If the sentence is not in passive voice, then the sentiment of the relation is negative.
- ❏ If h is negative and d is positive and the speaker is a first person, then the expressed sentiment is positive, otherwise sentic patterns predict a negative sentiment.
- ❏ If h is positive and d is negative, then the expressed sentiment is detected as negative.
- ❏ If h and d are both positive, then the relation results in a positive sentiment.

Example 1: "*His troubles were relieved.*" The word "*relieve*" is in a subject noun relation with "*trouble.*" Here, the polarity of the word "*trouble*" is negative, while that of the word "*relieve*" is positive. The rule indicates that in this case the polarity is given by the second word. Therefore, the polarity assigned to the whole sentence is positive.

Example 2: "*My success has pissed him off.*" The word "*success*" is in subject noun relation with "*pissed.*" The polarity of "*success*" is positive while "*pissed*" has negative polarity. The final polarity of the sentence is negative according to this rule.

Example 3: "*Her gift was bad.*" The word "*gift*" is in subject noun relation with "*bad.*" The polarity of "*gift*" is positive and "*bad*" is negative. Therefore, sentic patterns extract the polarity of the sentence as negative.

2) Complement clause: This rule is fired when a sentence contains a finite clause which is subordinate to another clause: "That" and "whether" are complement clauses.

Trigger: When a complement clause is found in a sentence.

Behavior: The sentence is split into two parts based on the complement clause:

- ❏ The sentiment expressed by the first part is considered as the final overall sentiment.
- ❏ If the first part conveys no sentiment, the sentiment of the second part is taken as the result.
- ❏ If the first part conveys no sentiment but has a negation, the sentiment of the last part is inverted.

Example 1: "*I love that you did not win the match.*" The sentiment expressed by the part of the sentence before "that" is positive, so the overall sentiment of the sentence is considered positive.

Example 2: "*I do not know whether he is good.*" The first part of the sentence has no sentiment but contains a negation that alters the polarity of the last part. Thus the overall polarity of the sentence is negative.

3) Adverbial Clause:

Trigger: When a sentence contains an adverbial clause (i.e., "while").

Behavior: The role of "while" in a sentence is similar to the one of "but". Then, sentic patterns first split the sentence into two parts by recognizing the subject and the use of comma in the sentence. Then, the overall sentiment of the sentence is conveyed by the second part.

Example: "*While I'm sure the quality of the product is fine, the color is very different.*" Sentic patterns first identify the two parts of the sentence by recognizing the comma and the subject after the comma. The polarity of the first part (i.e., "*I'm sure the quality of the product is fine*") is positive but the polarity of the second part ("*the color is very different*") is neutral. Thus the polarity of the sentence is negative.

Another group of rules described in [3] treats the modifiers. These rules include treatment of adjectival, adverbial, and participial modification, relative clauses, prepositional phrases, adverbial clause modifiers, and untyped dependency.

Yet a third group represents heuristic rules, difficult to classify, applied in specific contexts. For example, one such rule used in the proposed method considers the treatment of the first person pronouns, which give important clues to the speaker's estimation of the situation. Here, we introduce a rule that deals with a specific preposition.

4) Rule for the preposition "against": In English, "*against*" is a preposition which carries a sentiment. Usually it is used as a negative sentiment expressing word. However, "*against*" can also be used in a sentence to express positive sentiment. Here, we give a few examples to explain the role of "against" in determining the

The polarity of the word *trouble* is negative, while that of the word *relieve* is positive. The rule indicates that in this case the polarity is given by the second word.

sentiment of a sentence. In (3), "*activity*" has negative sentiment as it is modified by a negative modifier, i.e., "*criminal*." Here, "*against*," attached to the target "*activity*," actually flips the polarity of "*activity*" and the overall sentiment of the sentence becomes positive.

(3) I am against all criminal activities.

In (4), "*against*" attaches to a positive target "*love*." Then, the overall sentiment of the sentence becomes negative.

(4) He is against me and your love.

If "*against*" attaches to a word with no polarity then the sentence sentiment turns negative.

IV. The Sentiment Flow Algorithm

In the previous work that used the idea of sentic pattern rules [3], no attention was paid there to the order of activation of the rules, that is, to the logic and order in which words of the sentence are considered as the active token. No algorithm of application of the rules was discussed in [3]. As the present research shows, this order is important. Here, we present an improved algorithm for activation of the rules, which gives better results as compared with [3].

The procedure can be considered as a tree painting algorithm operating on the nodes and arcs of the syntactic dependency tree. For those words or relations—concepts, or multiword expressions—for which the polarity can be determined directly from the existing lexical resources, the algorithm assigns it directly. Then it gradually extends the labels to other arcs and nodes, with the necessary transformations determined by sentic pattern rules, until it obtains the final label for the root element, which is the desired output. We term the extending of the polarity labels the flow of the sentiment.

Specifically, the algorithm dynamically operates over the dependency parse tree of the sentence. Starting from the first (leftmost) relation in the tree, the rules corresponding to relations are activated: for a relation $R(A, B)$, the rules of the form R_i are activated to assign polarity (not necessarily the same) to the relation itself and to the words A and B. The rules for relations that involve either A or B are scheduled to be activated next: the main idea of the algorithm is to use the polarity already assigned to the relations and words previously processed. However, a rule may alter the order of activation of other rules if it needs additional information before it can proceed. For example, while computing the polarity of a relation $R(A, B)$, if A and B have any modifier, negation and subject-noun relation, then those relations are computed immediately. The reason is that such relations may alter the polarity of A and B.

If there is no rule for a given relation $R(A, B)$, then it is left unprocessed and the new relations are scheduled for processing using the method described above. When there are no relations scheduled for processing, the process restarts from the leftmost relation not yet processed for which a rule exists.

The output of the algorithm is the polarity of the relation processed last. It accumulates the information of all relations in the sentence, because each rule relies on the result of the previous ones, so the information flows from the leftmost relation towards the rule executed last, which often corresponds to one of the rightmost relations.

Below, for (5), we describe the sentiment flow across the dependency arcs based on sentic patterns.

(5) My failure makes him happy.

See the dependency tree of this sentence in Figure 2.

- First the relation between "*my*" and "*failure*" is considered. This is a *possession modifier* relation which does not satisfy any rule, so nothing has to be done.
- Then, the algorithm computes the polarity of the subject-noun relation between "*make*" and "*failure*." The sentiment of this relation is negative according to sentic patterns. The rule also assigns negative polarity to "*make*," which is usually neutral. This is contextual polarity to be used to compute the polarity of subsequent relations.
- Next, the polarity of the relation between "*make*" and "*happy*" is computed. This computation also needs the polarity of the relation computed in the previous step. Before computing the polarity of this relation, the subject-noun relation between "*him*" and "*happy*" is computed and a positive polarity is obtained. This polarity value does not alter the polarity of "*happy*," which is positive according to SenticNet. The word "*make*" has a negative polarity according to the previous step. Then, there is a clausal complement relation between "*make*" and "*happy*." Based on the clausal complement rule, sentic patterns assign negative polarity to this relation. After this computation there is no more relation left which satisfies the rules, so the sentence is assigned negative polarity by the algorithm.

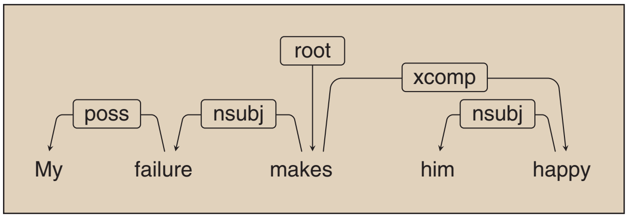


FIGURE 2 Dependency tree for the sentence "*My failure makes him happy*."

V. CI Classifier

The success of the proposed rule-based algorithm crucially relies on the completeness of the knowledge base used, in this case, SenticNet. However, no lexical resource can be absolutely complete, and in any text there will be some words or concepts still absent in the current version of the knowledge base. Instead of simply ignoring them, we resort to a backup processing method based on statistical machine learning. Namely, for the concepts that are absent in SenticNet, we extract features as described below and use a CI classifier trained on three well-known sentiment analysis datasets.

The common-sense knowledge features were the main component of our feature vectors. They were extracted from the AffectiveSpace lexical resource.

A. Datasets Used

We used the following three datasets to train the classifier, as well as to test the proposed method:

1) *Movie Review Dataset*: This dataset contains one thousand positive reviews and one thousand negative reviews extracted from the benchmark corpus of movie reviews created by Pang and Lee [35]. The corpus was collected from the website rottentomatos.com, where the texts are written by experts. The corpus has been pre-processed. The reviews were labeled by Pang and Lee at the document level, and later by Socher et al. [7], who labeled each individual sentence. The sentences are labeled with five labels, from strong or weak positive to neutral to weak or strong negative. For the experiments, however, we did not use neutral sentences, and for all others, we only considered binary polarity. The resulting corpus we used in the experiments includes 4,800 positive sentences and 4,813 negative sentences.
2) *Blitzer Dataset*: Another dataset we used in the experiments was developed by Blitzer et al. [10]. It includes one thousand positive documents and one thousand negative documents for each one of the seven domains, of which we only used the electronics domain. From this corpus, we selected at random 3,500 sentences marked other than neutral belonging to positive reviews, and the same amount of non-neutral sentences belonging to negative reviews. Then we manually labeled them with polarity, since we found that the polarity of an individual sentence not always corresponded to the polarity of the whole document, since in many cases the reviews were balanced and contained both positive and negative considerations. This procedure resulted in 3,800 positive sentences and 3,410 negative sentences.
3) *Amazon product review dataset*: We crawled the reviews of 453 mobile phones from http://amazon.com. Each review was split into sentences, and each sentence was then manually labelled by its sentiment labels. Finally, we obtained 115,758 sentences, out of which 48,680 were negative, 2,957 sentences neutral and 64,121 positive. In this experiment, we only employed positive and negative sentences. So, the final *Amazon dataset* contained 112,801 sentences annotated as either positive or negative.

B. Feature Set

As classification features, we used common-sense knowledge features, a sentic feature, a part-of-speech feature, a negation feature, and a modification feature.

Common-sense knowledge features were the main component of the feature vectors. They were extracted from the AffectiveSpace [33] for those concepts that were represented in the multidimensional vector space of commonsense knowledge. The latter resource assigns to each concept 100 real-number values. The common-sense knowledge feature vector for the whole sentence was obtained by coordinate-wise summation of individual 100-dimensional vectors for each concept present in the sentence and found in AffectiveSpace.

The sentic feature directly represented the polarity of each concept extracted from SenticNet; for the complete sentence the polarities of individual concepts were summed up. This represents the baseline "old way" shown in Figure 1(b).

Finally, the following three features reflected the formal characteristics of the sentence. The part-of-speech feature was a 3-dimensional vector reflecting the number of adjectives in the sentence, the number of adverbs, and the number of nouns. The binary negation feature indicated whether there was any negation in the sentence. In a similar way, the binary modification feature reflected whether in the dependency structure of the sentence there existed a word modified by a noun, an adjective, or an adverb. The latter feature, however, was not found to be useful in the experiments.

C. Classification

We selected 60% of the sentences from each of the three datasets as the training set for the classification. The sentences from each dataset were randomly drawn in such a way to balance the dataset with 50% negative sentences and 50% positive sentences. We used a novel CI technique called extreme learning machine (ELM) [36], [37], which is a recently developed type of single-hidden layer feedforward networks, with hidden layer that does not require tuning. ELM was found to outperform state-of-the-art methods such as SVM, in terms of both accuracy as well as training time. In the experiments, we obtained an overall 71.32% accuracy on the final dataset described in Table 1 using ELM and 68.35% accuracy using SVM.

TABLE 1 Datasets to train and test CI classifiers.

DATASET	NUMBER OF TRAINING SENTENCES	NUMBER OF TEST SENTENCES
MOVIE REVIEW DATASET	5,678	3,935
BLITZER-DERIVED DATASET	4,326	2,884
AMAZON DATASET	67,681	45,120
FINAL DATASET	77,685	51,939

We also trained the classifiers on each single dataset and tested it on both other datasets. In Table 2 we compare performance if the algorithm obtained in this experiment using different classifiers. It can be noted from Table 2 that the model trained on the Amazon dataset produced the best accuracy compared to the movie review and Blitzer-derived datasets. For each of these experiments, ELM outperformed SVM. The best performance of the ELM classifier was obtained on the Movie Review dataset, while the SVM classifier performed best on the Blitzer dataset. The training and test set collected from different datasets are shown in Table 1.

TABLE 2 Performance of the classifiers: SVM vs. ELM.

TRAINING DATASET	TEST DATASET: MOVIE REVIEWS SVM	MOVIE REVIEWS ELM	BLITZER SVM	BLITZER ELM	AMAZON SVM	AMAZON ELM
MOVIE REVIEWS	–		64.12%	72.12%	65.14%	69.21%
BLITZER	61.25%	68.09%	–		62.25%	66.73%
AMAZON	69.77%	70.03%	72.23%	73.30%	–	

TABLE 3 Feature analysis. Results of the ELM classifier on the final dataset from Table 1.

FEATURES USED	ACCURACY (ELM)
ALL	71.32%
ALL EXCEPT COMMON-SENSE KNOWLEDGE	40.11%
ALL EXCEPT SENTIC FEATURE	70.84%
ALL EXCEPT PART-OF-SPEECH FEATURE	70.41%
ALL EXCEPT NEGATION FEATURE	68.97%
ALL EXCEPT MODIFICATION FEATURE	**71.53%**

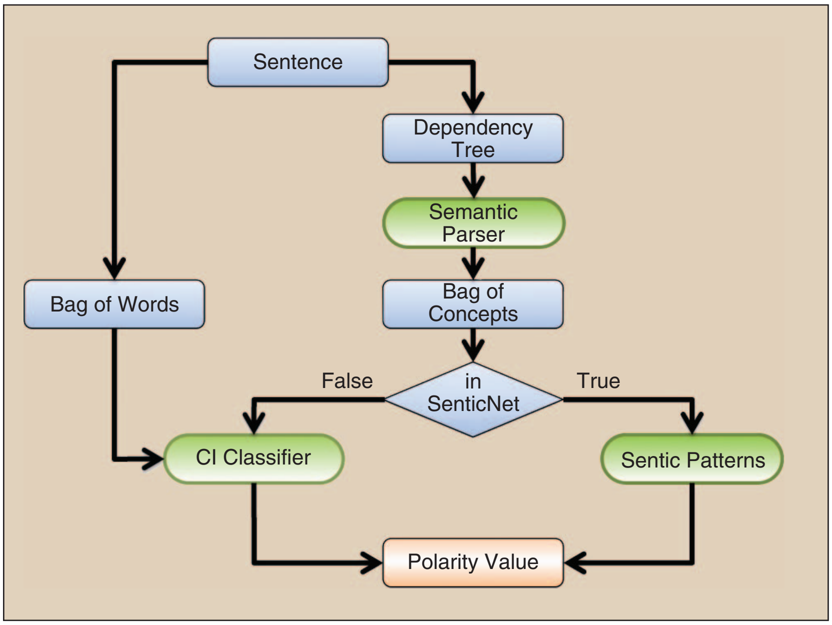


FIGURE 3 Sentence-level polarity detection algorithm. The semantic parser detects concepts in the text. The dynamic sentic patterns are applied to those concepts that are present in SenticNet, while other concepts are treated using statistical machine-learning methods.

Therefore, if a sentence cannot be processed using SenticNet and sentic patterns, it is possible to train an ELM classifier, which gives the polarity of the sentence basing on the features extracted from the text.

Though the ELM classifier was found to perform best when all features described in this paper were used together, common-sense-knowledge based features proved to be the most significant ones. From Table 3 one can notice that negation was also a useful feature. The other features were not found to play a significant role in the performance of the classifier; however, except for the modification feature, they were still found to be useful for producing the best accuracy. Since the ELM classifier provided the best accuracy, Table 3 shows the results obtained with this classifier. However, the purpose of this paper is to show the use of an ensemble of linguistic rules, thus a detailed study of features and their relative impact on CI classifiers will be the topic of our future work, to further enrich and optimize the performance of the ensemble framework.

VI. Experimental Results and Discussion

The proposed method is implemented in a live web demo available on http://sentic.net/demo. The live demo allows the user to enter one sentence and obtain its polarity according to the algorithm presented in this paper.

We tested the proposed method on three datasets: the Movie Review dataset described in Section V-A1, the Blitzer-derived dataset described in Section V-A2 and the Amazon dataset described in Section V-A3. As shown in the results, the best performance in terms of accuracy is observed using a combination of knowledge-based rules (Section III) and statistical machine-learning classification (Section V). Such classification is used as a back-up technique for the knowledge-based processing if no matching concepts are found in SenticNet (Figure 3). Table 4 shows a comparison of the experimental results.

A. Results

1) Results on the Movie Review Dataset: The accuracy we obtained in the experiments on the Movie Review dataset was 88.12%, which is better than the state-of-the-art accuracy of 85.40% achieved by Socher et al. [7]. In Table 4, we report results both using ensemble classification and not using ensemble classification. In addition, the table compares of the proposed system with well-known state-of-the-art previous results. The table shows that the proposed system performed better than [3] on the same Movie Review dataset. This is due to a new set of patterns and the use of a new training set, which helped to obtain better accuracy.

2) Results on the Blitzer-derived Dataset: Experiments on the Blitzer-derived dataset, as described in Section V-A2, showed an accuracy of 88.27% at the sentence level. We tested the performance of the other benchmark sentiment analysis systems on this dataset. As on Movie Review dataset, the new patterns and new training sets increased the accuracy over [3]. Further, the method by Socher et al. [7] was found to perform very poorly on the Blitzer dataset.

3) Results on the Amazon Dataset: The same table shows the results of dynamic sentic patterns on the Amazon dataset described in Section V-A3. Again, the proposed method outperforms the state-of-the-art approaches.

B. Discussion

The proposed approach outperforms the state-of-the-art methods on both the movie review and the Amazon datasets. The results achieved on the Blitzer-derived dataset are even more impressive. This can be considered as evidence of robustness of the proposed method, since its high performance is stable across very different datasets in different domains. Moreover, while standard statistical methods require extensive training, both in terms of resources (training corpora) and time (learning time), sentic patterns are mostly unsupervised, except for the dynamic CI module, which is, though, very fast, due to the use of ELM.

The addition and improvement of the patterns, as noted in [3], has helped the system improve its results. Results show performance improvement over [3]. On the other hand, [7] has failed to obtain consistently good accuracy over both Blitzer and amazon datasets but obtained good accuracy over the Movie Review dataset. This is because the classifier proposed in [7] was trained on the Movie Review dataset only.

The proposed approach has therefore obtained a better accuracy than the baseline system. We combined the three datasets described in Section V-A1, V-A2 and V-A3 to evaluate sentic patterns. From Section V-A, we can calculate the number of positive and negative sentences in the dataset, which shows 72,721 positive and 56,903 negative sentences. If the system predicts all sentences as positive, this would give a baseline accuracy of 56.10%. Clearly, the proposed system significantly outperformed the baseline system. Since the performance of the proposed method relies on the quality of dependency parsing, which in turn requires grammatically correct text, ungrammatical sentences present in all three datasets negatively affected results. It is worth noting that the accuracy of the system crucially depends on the quality of the output of the dependency parser, which relies on grammatical correctness of the input sentences.

On the other hand, compilation of a balanced training dataset has a strong impact on developing a more accurate classifier than the one reported by Poria et al. [3].

1) Results obtained using SentiWordNet: We carried out an extensive experiment using SentiWordNet instead of SenticNet on all the three datasets. The results showed SenticNet performed slightly better than SentiWordNet. A possible future direction of this work is the invention of a novel approach to combine SenticNet and SentiWordNet in the sentiment analysis framework. The slight difference in the accuracy reported in Table 5 confirmed that both the lexicons share similar knowledge but since SenticNet contains concepts, this helps increase accuracy. For example, in the sentence "The battery lasts little", proposed algorithm extracts the concept "last little" which exists in SenticNet but not in SentiWordNet. As a result, when SenticNet is used the framework labels the sentence with a "negative" sentiment but when using SentiWordNet the sentence is labeled with a "neutral" sentiment.

2) Examples of cases when a CI classifier was used: Table 6 presents some examples where the CI module was used to guess the polarity. For each of these sentences, no concept was found in SenticNet.

TABLE 4 Precision achieved with different classification algorithms on different datasets.

ALGORITHM	MOVIE REVIEW DATASET	BLITZER-DERIVED DATASET	AMAZON DATASET
RNN (SOCHER ET AL. 2012 [38])	80.00%	–	–
RNTN (SOCHER ET AL. 2013 [7])	85.40%	61.93%	68.21%
PORIA ET AL. 2014 [3]	86.21%	87.00%	79.33%
SENTIC PATTERNS	87.15%	86.46%	80.62%
ELM CLASSIFIER	71.11%	74.49%	71.29%
ENSEMBLE CLASSIFICATION (PROPOSED METHOD)	**88.12%**	**88.27%**	**82.75%**

TABLE 5 Results obtained using SentiWordNet.

DATASET	USING SENTICNET	USING SENTIWORDNET
MOVIE REVIEW	88.12%	87.63%
BLITZER	88.27%	88.09%
AMAZON	82.75%	80.28%

TABLE 6 Some examples where a CI classifier was used to obtain the polarity label.

SENTENCE	POLARITY
I HAD TO RETURN THE PHONE AFTER 2 DAYS OF USE.	NEGATIVE
THE PHONE RUNS RECENT OPERATING SYSTEM.	POSITIVE
THE PHONE HAS A BIG AND CAPACITIVE TOUCHSCREEN.	POSITIVE
MY IPHONE BATTERY LASTS ONLY FEW HOURS.	NEGATIVE
I REMEMBER THAT I SLEPT AT THE MOVIE HALL.	NEGATIVE

VII. Conclusions

The paper shows how computational intelligence and linguistics can be blended in order to understand sentiment associated with the text. The presented approach combines the use of various linguistic patterns based on the syntactic structure of the sentences. Similarly to the function of electronic logic gates, the algorithm determines the polarity of each word and flows, or extends, it through the dependency arcs in order to determine the final polarity label of the sentence. Overall, the proposed approach, relying on the novel sentiment flow algorithm, has outperformed the majority of the main existing approaches on the benchmark datasets, showing outstanding effectiveness.

The future work will aim to discover more linguistic patterns, generalizing the patterns and use of deep learning for the CI module. In the future, we plan to carry out additional experiments using diverse datasets to further evaluate the domain independence of the capabilities of linguistic patterns, and compare their performance with other benchmark state-of-the-art approaches (such as [12]). Further, whilst our presumption is that all linguistic patterns are equally important for calculating the polarity of natural language sentences, it would be interesting to carry out further detailed theoretical analysis to investigate relative contributions of sentic patterns across a range of datasets from different domains.

As discussed in Section VI-A, sarcasm detection plays an important role in sentiment analysis: irony usually inverts the polarity of the sentence. We plan to develop a sarcasm detection module trained on available datasets[3] [39]. Identifying implicit negation [19] when irony is detected, is also one of the key future tasks of our framework. Finally, we plan to develop other modules, e.g., microtext analysis and anaphora resolution, as part of our vision to develop a novel holistic approach for solving the multi-faceted dynamic sentiment analysis challenge [40].

References

[1] E. Cambria, A. Hussain, C. Havasi, and C. Eckl, "Common sense computing: From the society of mind to digital intuition and beyond," in *Biometric ID Management and Multimodal Communication* (Series Lecture Notes in Computer Science, vol. 5707), J. Fierrez, J. Ortega, A. Esposito, A. Drygajlo, and M. Faundez-Zanuy, Eds. Berlin, Germany: Springer, 2009, pp. 252–259.

[2] E. Cambria and B. White, "Jumping NLP curves: A review of natural language processing research," *IEEE Comput. Intell. Mag.*, vol. 9, no. 2, pp. 48–57, 2014.

[3] S. Poria, E. Cambria, G. Winterstein, and G.-B. Huang, "Sentic patterns: Dependency-based rules for concept-level sentiment analysis," *Knowl.-Based Syst.*, vol. 69, pp. 45–63, Oct. 2014.

[4] E. Cambria, B. Schuller, B. Liu, H. Wang, and C. Havasi, "Knowledge-based approaches to concept-level sentiment analysis," *IEEE Intell. Syst.*, vol. 28, no. 2, pp. 12–14, 2013.

[5] E. Cambria, B. Schuller, B. Liu, H. Wang, and C. Havasi, "Statistical approaches to concept-level sentiment analysis," *IEEE Intell. Syst.*, vol. 28, no. 3, pp. 6–9, 2013.

[6] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. Conf. Empirical Methods Natural Language Processing*, 2002, vol. 10, pp. 79–86.

[7] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Language Processing*, 2013, vol. 1631, pp. 1642–1654.

[8] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proc. 40th Annu. Meeting Association Computational Linguistics*, 2002, pp. 417–424.

[9] X. Hu, J. Tang, H. Gao, and H. Liu, "Unsupervised sentiment analysis with emotional signals," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 607–618.

[10] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th Annu. Meeting Association Computational Linguistics*, 2007, vol. 7, pp. 440–447.

[11] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 751–760.

[12] D. Bollegala, D. Weir, and J. Carroll, "Cross-domain sentiment classification using a sentiment sensitive thesaurus," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 8, pp. 1719–1731, 2013.

[13] K. Denecke, "Using SentiWordNet for multilingual sentiment analysis," in *Proc. 24th IEEE Int. Conf. Data Engineering Workshop*, 2008, pp. 507–512.

[14] B. Ohana and B. Tierney, "Sentiment classification of reviews using SentiWordNet," in *Proc. 9th Conf. Information Technology Telecommunications*, 2009, pp. 13–22.

[15] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *Proc. Language Resources Evaluation Conf.*, 2010, vol. 10, pp. 2200–2204.

[16] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[17] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Proc. 35th Annu. Meeting Association Computational Linguistics 8th Conf. European Chapter Association Computational Linguistics*, 1997, pp. 174–181.

[18] L. Jia, C. Yu, and W. Meng, "The effect of negation on sentiment analysis and retrieval effectiveness," in *Proc. 18th ACM Int. Conf. Information Knowledge Management*, 2009, pp. 1827–1830.

[19] A. Reyes and P. Rosso, "On the difficulty of automatically detecting irony: Beyond a simple case of negation," *Knowl. Inf. Syst.*, vol. 40, no. 3, pp. 595–614, 2014.

[20] K. Chawla, A. Ramteke, and P. Bhattacharyya, "IITB-sentiment-analysts: Participation in sentiment analysis in Twitter SemEval 2013 task," in *Proc. 2nd Joint Conf. Lexical Computational Semantics (*SEM)*, 2013, vol. 2, pp. 495–500.

[21] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lect. Human Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.

[22] K. Moilanen and S. Pulman, "Sentiment composition," in *Proc. Int. Conf. Recent Advances Natural Language Processing*, 2007, pp. 378–382.

[23] R. Narayanan, B. Liu, and A. Choudhary, "Sentiment analysis of conditional sentences," in *Proc. Conf. Empirical Methods Natural Language Processing*, 2009, pp. 180–189.

[24] Y.-H. Kim, S.-Y. Hahn, and B.-T. Zhang, "Text filtering by boosting naive Bayes classifiers," in *Proc. 23rd ACM SIGIR Conf. Research Development Information Retrieval*, 2000, pp. 168–175.

[25] A. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes," *Adv. Neural Inform. Process. Syst.*, vol. 14, pp. 841–846, 2002.

[26] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1138–1150, 2006.

[27] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 91–100.

[28] M. Sahlgren and R. Cöster, "Using bag-of-concepts to improve the performance of support vector machines in text categorization," in *Proc. 42nd Annu. Meeting Association Computational Linguistics*, 2004.

[29] F. Wang, Z. Wang, Z. Li, and J.-R. Wen, "Concept-based short text classification and ranking," in *Proc. Conf. Intelligent Computer Mathematics*, 2014, pp. 1069–1078.

[30] Y. Zhang and B. Liu, "Semantic text classification of emergent disease reports," in *Principles and Practice of Knowledge Discovery in Databases*, Berlin, Germany: Springer, 2007, pp. 629–637.

[31] C.-E. Wu and R. T.-H. Tsai, "Using relation selection to improve value propagation in a ConceptNet-based sentiment dictionary," *Knowl.-Based Syst.*, vol. 69, pp. 100–107, 2014.

[32] E. Cambria, D. Olsher, and D. Rajagopal, "SenticNet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis," in *Proc. 28th AAAI Conf. Artificial Intelligence*, 2014, pp. 1515–1521.

[33] E. Cambria, J. Fu, F. Bisio, and S. Poria, "AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis," in *Proc. 29th AAAI Conf. Artificial Intelligence*, 2015, pp. 508–514.

[34] L. Polanyi and A. Zaenen, "Contextual valence shifters," in *Computing Attitude and Affect in Text: Theory and Applications* (The Information Retrieval Series, vol. 20), Berlin, Germany: Springer, 2006, pp. 1–10.

[35] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. 43rd Annu. Meeting Association Computational Linguistics*, 2005, pp. 115–124.

[36] G.-B. Huang, E. Cambria, K.-A. Toh, B. Widrow, and Z. Xu, "New trends of learning in computational intelligence," *IEEE Comput. Intell. Mag.*, vol. 10, no. 2, pp. 16–17, 2015.

[37] G.-B. Huang, E. Cambria, K.-A. Toh, B. Widrow, and Z. Xu, "New trends of learning in computational intelligence (Part II)," *IEEE Comput. Intell. Mag.*, vol. 10, no. 3, p. 8, 2015.

[38] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proc. Joint Conf. Empirical Methods Natural Language Processing Computational Natural Language Learning*, 2012, pp. 1201–1211.

[39] E. Filatova, "Irony and sarcasm: Corpus generation and analysis using crowdsourcing," in *Proc. Language Resources Evaluation Conf.*, 2012, pp. 392–398.

[40] E. Cambria, S. Poria, F. Bisio, R. Bajpai, and I. Chaturvedi, "The CLSA model: A novel framework for concept-level sentiment analysis," in *Computational Linguistics and Intelligent Text Processing* (Series Lecture Notes in Computer Science, vol. 9042), Berlin, Germany: Springer, 2015, pp. 3–22.

[3] http://alt.qcri.org/semeval2015/task11