Knowledge-Based Systems 69 (2014) 45–63

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Sentic patterns: Dependency-based rules for concept-level sentiment analysis

Soujanya Poria [a], Erik Cambria [b,*], Grégoire Winterstein [c], Guang-Bin Huang [a]

[a] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[b] School of Computer Engineering, Nanyang Technological University, Singapore
[c] Department of Linguistics, University of Paris Diderot, France

ARTICLE INFO

Article history:
Available online 6 June 2014

Keywords:
Artificial intelligence
Natural language processing
Opinion mining
Sentic computing
Linguistic rules

ABSTRACT

The Web is evolving through an era where the opinions of users are getting increasingly important and valuable. The distillation of knowledge from the huge amount of unstructured information on the Web can be a key factor for tasks such as social media marketing, branding, product positioning, and corporate reputation management. These online social data, however, remain hardly accessible to computers, as they are specifically meant for human consumption. The automatic analysis of online opinions involves a deep understanding of natural language text by machines, from which we are still very far. To this end, concept-level sentiment analysis aims to go beyond a mere word-level analysis of text and provide novel approaches to opinion mining and sentiment analysis that enable a more efficient passage from (unstructured) textual information to (structured) machine-processable data. A recent knowledge-based technology in this context is sentic computing, which relies on the ensemble application of common-sense computing and the psychology of emotions to infer the conceptual and affective information associated with natural language. Sentic computing, however, is limited by the richness of the knowledge base and by the fact that the bag-of-concepts model, despite more sophisticated than bag-of-words, misses out important discourse structure information that is key for properly detecting the polarity conveyed by natural language opinions. In this work, we introduce a novel paradigm to concept-level sentiment analysis that merges linguistics, common-sense computing, and machine learning for improving the accuracy of tasks such as polarity detection. By allowing sentiments to flow from concept to concept based on the dependency relation of the input sentence, in particular, we achieve a better understanding of the contextual role of each concept within the sentence and, hence, obtain a polarity detection engine that outperforms state-of-the-art statistical methods.

1. Introduction

Between the dawn of the Internet through year 2003, there were just a few dozens exabytes of information on the Web. Today, that much information is created weekly. The opportunity to capture the opinions of the general public about social events, political movements, company strategies, marketing campaigns, and product preferences has raised increasing interest both in the scientific community, for the exciting open challenges, and in the business world, for the remarkable fallouts in marketing and financial prediction. Keeping up with the ever-growing amount of unstructured information on the Web, however, is a formidable task and requires fast and efficient models for opinion mining. Hitherto, natural language processing (NLP) and online information retrieval have been mainly based on algorithms relying on the textual representation of web pages. Such algorithms are very good at retrieving texts, splitting them into parts, checking the spelling, and counting their words. But when it comes to interpreting sentences and extracting meaningful information, their capabilities are known to be very limited.

Early works aimed to classify entire documents as containing overall positive or negative polarity, or rating scores of reviews. Such systems were mainly based on supervised approaches relying on manually labeled samples, such as movie or product reviews where the opinionist's overall positive or negative attitude was explicitly indicated. However, opinions and sentiments do not occur only at document-level, nor they are limited to a single

* Corresponding author. Tel.: +65 67904328.
E-mail addresses: SPoria@ntu.edu.sg (S. Poria), cambria@ntu.edu.sg (E. Cambria), gregoire.winterstein@linguist.univ-paris-diderot.fr (G. Winterstein), EGBHuang@ntu.edu.sg (G.-B. Huang).

http://dx.doi.org/10.1016/j.knosys.2014.05.005
0950-7051/

valence or target. Contrary or complementary attitudes toward the same topic or multiple topics can be present across the span of a document. Later works adopted a segment-level opinion analysis aiming to distinguish sentimental from non-sentimental sections, e.g., by using graph-based techniques for segmenting sections of a document on the basis of their subjectivity, or by performing a classification based on some fixed syntactic phrases that are likely to be used to express opinions. In more recent works, text analysis granularity has been taken down to sentence-level, e.g., by using presence of opinion-bearing lexical items (single words or n-grams) to detect subjective sentences, or by exploiting association rule mining for a feature-based analysis of product reviews. These approaches, however, are still far from being able to infer the cognitive and affective information associated with natural language as they mainly rely on knowledge bases that are still too limited to efficiently process text at sentence-level. Moreover, such text analysis granularity might still not be enough as a single sentence may contain different opinions about different facets of the same product or service.

In a Web where user-generated content is drowning in its own output, NLP researchers are faced with the same challenge: the need to jump the curve [1] to make significant, discontinuous leaps in their thinking, whether it is about information retrieval, aggregation, or processing. Relying on arbitrary keywords, punctuation, and word co-occurrence frequencies has 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 causing standard NLP algorithms to be increasing less efficient. In order to properly extract and manipulate text meanings, a NLP system must have access to a significant amount of knowledge about the world and the domain of discourse.

In this work, we introduce a novel paradigm to sentiment analysis that merges linguistics, common-sense computing, and machine learning for properly deconstructing natural language text into concepts and opinion targets and, hence, for improving the accuracy of polarity detection. By augmenting the sentic computing framework [2] with dependency-based rules, in particular, we achieve a better understanding of the contextual role of each concept within a sentence and, hence, obtain a polarity detection accuracy that exceeds the state of the art.

The rest of the paper is organized as follows: Section 2 presents related work in the field of opinion mining and sentiment analysis; Section 3 describes how the sentic computing framework is further developed and applied for concept-level sentiment analysis; Section 4 illustrates the dependency-based rules designed for sentence-level polarity detection; Section 5 presents the machine learning techniques developed to overcome the limitedness of the common-sense knowledge base; Section 6 proposes an evaluation and a discussion of the overall framework; finally, Section 7 concludes the paper and suggests directions for future work.

2. Related work

Existing approaches to sentiment analysis can be grouped into four main categories: keyword spotting, lexical affinity, statistical methods, and concept-level approaches. Keyword spotting is the most naive approach and probably also the most popular because of its accessibility and economy. Text is classified into affect categories based on the presence of fairly unambiguous affect words like `happy`, `sad`, `afraid`, and `bored`. The weaknesses of this approach lie in two areas: poor recognition of affect when negation is involved and reliance on surface features. In relation to its first weakness, while the approach can correctly classify the sentence "today was a happy day" as being happy, it is likely to fail on a sentence like "today wasn't a happy day at all". Regarding its second weakness, the approach relies on the presence of obvious affect words that are only surface features of the prose. In practice, a lot of sentences convey affect through underlying meaning rather than affect adjectives. For example, the text "My husband just filed for divorce and he wants to take custody of my children away from me" certainly evokes strong emotions, but uses no affect keywords, and therefore, cannot be classified using a keyword spotting approach.

Lexical affinity is slightly more sophisticated than keyword spotting, as rather than simply detecting obvious affect words, it assigns arbitrary words a probabilistic 'affinity' for a particular emotion. For example, `accident` might be assigned a 75% probability of indicating a negative affect, as in `car accident` or `hurt by accident`. These probabilities are usually learnt using linguistic corpora. Though often outperforming pure keyword spotting, there are two main problems with the approach. First, lexical affinity, operating solely on the word-level, can easily be tricked by sentences like "I avoided an accident" (negation) and "I met my girlfriend by accident" (other word senses). Second, lexical affinity probabilities are often biased towards text of a particular genre, dictated by the source of the linguistic corpora. This makes it difficult to develop a reusable, domain-independent model.

Statistical methods, such as Bayesian inference, support vector machine (SVM) and artificial neural network (ANN), have been popular for affect classification of texts. By feeding a machine learning algorithm a large training corpus of affectively annotated texts, it is possible for the system to not only learn the affective valence of affect keywords (as in the keyword spotting approach), but also to take into account the valence of other arbitrary keywords (like lexical affinity), punctuation, and word co-occurrence frequencies. However, traditional statistical methods are generally semantically weak, i.e., with the exception of obvious affect keywords, other lexical or co-occurrence elements in a statistical model have little predictive value individually. As a result, statistical text classifiers only work with acceptable accuracy when given a sufficiently large text input. So, while these methods may be able to affectively classify user's text on the page- or paragraph- level, they do not work well on smaller text units such as sentences or clauses.

Concept-based approaches to sentiment analysis focus on a semantic analysis of text through the use of web ontologies or semantic networks, which allow the aggregation of conceptual and affective information associated with natural language opinions. By relying on large semantic knowledge bases, such approaches step away from blind use of keywords and word co-occurrence counts, but rather rely on the implicit features associated with natural language concepts. Unlike purely syntactical techniques, concept-based approaches are also able to detect sentiments that are expressed in a subtle manner, e.g., through the analysis of concepts that do not explicitly convey any emotion, but are implicitly linked to other concepts that do so.

The analysis at concept-level is intended to infer the semantic and affective information associated with natural language opinions and, hence, to enable a comparative fine-grained feature-based sentiment analysis. Rather than gathering isolated opinions about a whole item (e.g., iPhone5), users are generally more interested in comparing different products according to their specific features (e.g., iPhone5's vs Galaxy S3's touchscreen), or even sub-features (e.g., fragility of iPhone5's vs Galaxy S3's touchscreen). In this context, the construction of comprehensive common and common-sense knowledge bases is key for feature-spotting and polarity detection, respectively. Common-sense, in particular, is necessary to properly deconstruct natural language text into sentiments for example, to appraise the concept small room as negative for a hotel review and small queue as positive for a post office, or the concept go read the book as positive for a book review but negative for a movie review.

Current approaches to concept-level sentiment analysis mainly leverage on existing affective knowledge bases such as ANEW [3], WordNet-Affect [4], ISEAR [5], SentiWordNet [6], and SenticNet [7]. In [8], for example, a concept-level sentiment dictionary is built through a two-step method combining iterative regression and random walk with in-link normalization. ANEW and SenticNet are exploited for propagating sentiment values based on the assumption that semantically related concepts share common sentiment. Moreover, polarity accuracy, Kendall distance, and average-maximum ratio are used, in place of mean error, to better evaluate sentiment dictionaries. A similar approach is adopted in [9], which presents a methodology for enriching SenticNet concepts with affective information by assigning an emotion label to them. Authors use various features extracted from ISEAR, as well as similarity measures that rely on the polarity data provided in SenticNet (specifically those based on WordNet-Affect) and ISEAR distance-based measures, including point-wise mutual information, and emotional affinity. Another recently reported work that builds upon an existing affective knowledge base is that of [10], which proposes the re-evaluation of objective words in SentiWordNet by assessing the sentimental relevance of such words and their associated sentiment sentences. Two sampling strategies are proposed and integrated with SVM for sentiment classification. Their experiments show that the proposed approach significantly outperforms the traditional sentiment mining approach, which ignores the importance of objective words in SentiWordNet. In [11], the main issues related to the development of a corpus for opinion mining and sentiment analysis are discussed both by surveying existing works in this area and presenting, as a case study, an ongoing project in Italian, called Senti–TUT, where a corpus for the investigation of irony about politics in social media is developed.

Other works have explored the ensemble application of knowledge bases and statistical methods. In [12], for example, a hybrid approach to combine lexical analysis and machine learning is proposed in order to cope with ambiguity and integrate the context of sentiment terms. The context-aware method identifies ambiguous terms that vary in polarity depending on the context and stores them in contextualized sentiment lexicons. In conjunction with semantic knowledge bases, these lexicons help ground ambiguous sentiment terms to concepts that correspond to their polarity. Further machine-learning based works include [13], which introduces a new methodology for the retrieval of product features and opinions from a collection of free-text customer reviews about a product or service. Such methodologies rely on a language-modeling framework that can be applied to reviews in any domain and language provided with a seed set of opinion words. The methodology combines both a kernel-based model of opinion words (learned from the seed set of opinion words) and a statistical mapping between words to approximate a model of product features from which the retrieval is carried out.

Other recent works in the context of concept-level sentiment analysis include tasks such as domain adaptation [14], opinion summarization [15], and multimodal sentiment analysis [16,17]. In the context of domain adaptation, there are two distinct needs, namely labeling adaptation and instance adaptation. However, most of the current research focuses on the former attribute, whilst neglecting the latter one. In [14], a comprehensive approach, termed feature ensemble plus sample selection (SS-FE), is proposed. SS-FE takes both types of adaptation into account: a feature ensemble (FE) model is first adopted to learn a new labeling function in a feature re-weighting manner, and a PCA-based sample selection (PCA-SS) method is then used as an aid to FE. A first step towards concept-level summarization is performed by STARLET [15], a novel approach to extractive multi-document summarization for evaluative text that considers aspect rating distributions and language modeling as summarization features. Such features encourage the inclusion of sentences in the summary that preserves the overall opinion distribution expressed across the original reviews and whose language best reflects the language of reviews. The proposed method offers improvements over traditional summarization techniques and other approaches to multi-document summarization of evaluative text.

A sub-field of sentiment analysis that is becoming increasingly popular is multimodal sentiment analysis. [16], for example, considers multimodal sentiment analysis based on linguistic, audio, and visual features. A database of 105 Spanish videos of 2–8 min length containing 21 male and 84 female speakers was collected randomly from the social media website, YouTube, and annotated by two labellers for ternary sentiment. This led to 550 utterances and approximately 10,000 words (authors have made the data available upon request). The joint use of the three feature types leads to a significant improvement over the use of each single modality. This is further confirmed on another set of English videos. In [17], the authors introduce the ICT-MMMO database of personal movie reviews collected from YouTube (308 clips) and ExpoTV (78 clips). The final set contains 370 of these (1–3 min) English clips in ternary sentiment annotation with one to two coders. The feature basis is formed by 2 k audio features, 20 video features, and different textual features for selection. Then, different levels of domain-dependence are considered: in-domain analysis, cross-domain analysis based on the 100 k textual Metacritic movie review corpus for training, and use of on-line knowledge sources. This shows that cross-corpus training works sufficiently well, and language-independent audiovisual analysis is competitive with linguistic analysis.

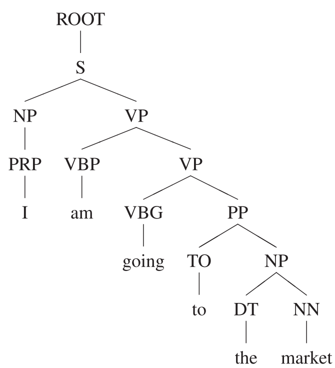
3. Concept-level sentiment analysis through sentic computing

Sentic computing is a multi-disciplinary approach to sentiment analysis at the crossroads between affective computing and common-sense computing, which exploits both computer and social sciences to better recognize, interpret, and process opinions and sentiments over the Web. In sentic computing, whose term derives from the Latin *sentire* (root of words such as sentiment and sentience) and *sensus* (intended both as capability of feeling and as common-sense), the analysis of natural language is based on affective ontologies and common-sense reasoning tools, which enable the analysis of text not only at document-, page- or paragraph-level, but also at sentence-, clause-, and concept-level. In particular, sentic computing involves the inter-disciplinary use of artificial intelligence and Semantic Web techniques, for knowledge representation and inference; mathematics, for carrying out tasks such as graph mining and multi-dimensionality reduction; linguistics, for discourse analysis and pragmatics; psychology, for cognitive and affective modeling; sociology, for understanding social network dynamics and social influence; and finally ethics, for understanding related issues about the nature of mind and the creation of emotional machines.

In this work, we further develop the sentic computing framework by using a semantic parser for deconstructing text into natural language concepts (Section 3.1), which then get represented and analyzed through a vector space of common-sense knowledge (Section 3.2) by means of an emotion categorization model (Section 3.3) and an ANN-based clustering technique (Section 3.4).

3.1. Semantic parsing

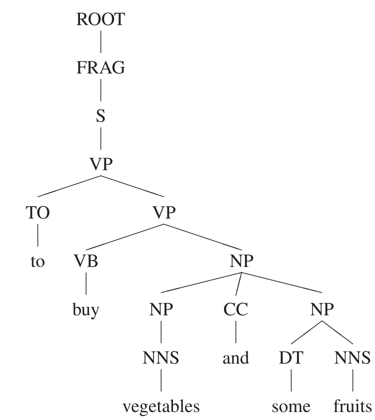
The aim of the semantic parser is to break text into clauses and, hence, deconstruct such clauses into concepts, to be later fed to a vector space of common-sense knowledge. For applications in fields such as real-time human–computer interaction and big social data analysis [18], deep natural language understanding is not strictly required: a sense of the semantics associated with text and some extra information (affect) associated with such semantics are often sufficient to quickly perform tasks such as emotion recognition and polarity detection.



First, the semantic parser breaks text into clauses. Each verb and its associated noun phrase are considered in turn, and one or more concepts is extracted from these. As an example, the clause “I went for a walk in the park”, would contain the concepts `go walk` and `go park`. The Stanford Chunker [19] is used to chunk the input text. A sentence like “I am going to the market to buy vegetables and some fruits” would be broken into “I am going to the market” and “to buy vegetables and some fruits”. A general assumption during clause separation is that, if a piece of text contains a preposition or subordinating conjunction, the words preceding these function words are interpreted not as events but as objects. The next step of the algorithm then separates clauses into verb and noun chunks, as suggested by the parse trees shown above.

Next, clauses are normalized in two stages. First, each *verb* chunk is normalized using the Lancaster stemming algorithm [20]. Second, each potential *noun* chunk associated with individual verb chunks is paired with the stemmed verb in order to detect multi-word expressions of the form ‘verb plus object’. Objects alone, however, can also represent a common-sense concept. To detect such expressions, a part-of-speech (POS) based bigram algorithm checks noun phrases for stopwords and adjectives. In particular, noun phrases are first split into bigrams and then processed through POS patterns, as shown in Algorithm 1. POS pairs are taken into account as follows:

1. ADJ + NOUN: The adj + noun combination and noun as a standalone concept are added to the objects list.
2. ADJ + STOPWORD: The entire bigram is discarded.
3. NOUN + ADJ: As trailing adjectives do not tend to carry sufficient information, the adjective is discarded and only the noun is added as a valid concept.
4. NOUN + NOUN: When two nouns occur in sequence, they are considered to be part of a single concept. Examples include *butter scotch*, *ice cream*, *cream biscuit*, and so on.
5. NOUN + STOPWORD: The stopword is discarded, and only the noun is considered valid.
6. STOPWORD + ADJ: The entire bigram is discarded.
7. STOPWORD + NOUN: In bigrams matching this pattern, the stopword is discarded and the noun alone qualifies as a valid concept.



Algorithm 1. POS-based bigram algorithm

Data: NounPhrase
Result: Valid object concepts
Split the NounPhrase into bigrams;
Initialize concepts to Null;
for *each NounPhrase* **do**
  while *For every* bigram *in the NounPhrase* **do**
    POS Tag the Bigram;
    if adj noun **then**
      add to Concepts: noun, adj + noun
    else if noun noun **then**
      add to Concepts: noun + noun
    else if stopword noun **then**
      add to Concepts: noun
    else if adj stopword **then**
      continue
    else if stopword adj **then**
      continue
    else
      Add to Concepts: entire bigram
    end
    repeat until no more bigrams left;
  end
end

The POS-based bigram algorithm extracts concepts such as *market*, *some fruits*, *fruits*, and *vegetables*. In order to capture event concepts, matches between the object concepts and the normalized verb chunks are searched. This is done by exploiting a parse graph that maps all the multi-word expressions contained in the knowledge bases (Fig. 1). Such an unweighted directed graph helps to quickly detect complex concepts, without performing an exhaustive search through all the possible word combinations that can form a common-sense concept.

Single-word concepts, e.g., *house*, that already appear in the clause as a multi-word concept, e.g., *beautiful house*, in fact, are

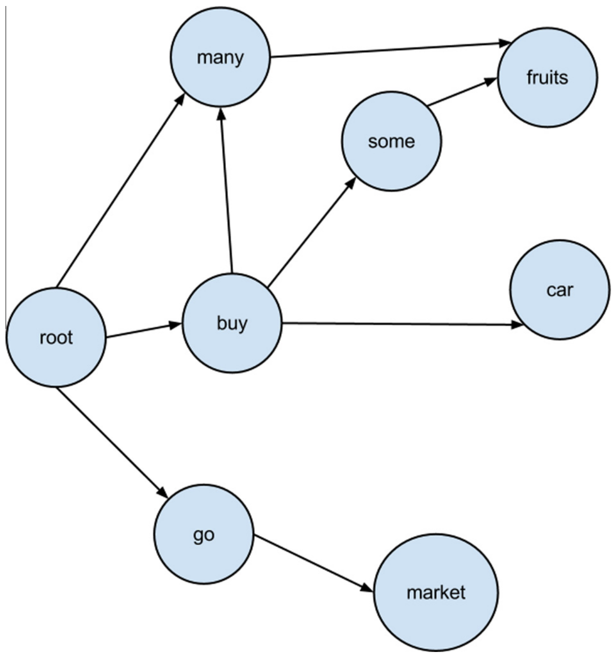


Fig. 1. Example parse graph for multi-word expressions.

pleonastic (providing redundant information) and are discarded. In this way, the Algorithm 2 is able to extract event concepts such as *go market*, *buy some fruits*, *buy fruits*, and *buy vegetables*, representing the concepts to be fed to a common-sense reasoning algorithm for further processing.

Algorithm 2. Event concept extraction algorithm

Data: Natural language sentence
Result: List of concepts
Find the number of verbs in the sentence;
for *every clause* **do**
  extract VerbPhrases and NounPhrases;
  stem VERB;
  for *every NounPhrase with the associated verb* **do**
    find possible forms of *objects*;
    link all *objects* to stemmed verb to get *events*;
  end
  repeat until no more clauses are left;
end

3.2. The vector space model

The best way to solve a problem is to already know a solution for it. If we have to face a problem we have never encountered before, however, we need to use our intuition. Intuition can be explained as the process of making analogies between the current problem and the ones solved in the past to find a suitable solution. Marvin Minsky attributes this property to the so called 'difference-engines' [21]. This particular kind of agents operates by recognizing differences between the current state and the desired state, and acting to reduce each difference by invoking K-lines that turn on suitable solution methods. This kind of thinking may be the essence of our supreme intelligence since in everyday life no two situations are ever the same and we have to perform this action continuously.

Human mind constructs intelligible meanings by continuously compressing over vital relations [22]. The compression principles aim to transform diffuse and distended conceptual structures to more focused versions so as to become more congenial for human understanding. To this end, principal component analysis (PCA) has been applied on the matrix representation of AffectNet. In particular, truncated singular value decomposition (TSVD) has been preferred to other dimensionality reduction techniques for its simplicity, relatively low computational cost, and compactness. TSVD is particularly suitable for measuring cross-correlations between affective common-sense concepts as it uses an orthogonal transformation to convert the set of possibly correlated common-sense features associated with each concept into a set of values of uncorrelated variables (the principal components of the SVD).

By using Lanczos' method [23], the generalization process is relatively fast (a few seconds), despite the size and the sparseness of AffectNet. Applying TSVD on AffectNet causes it to describe other features that could apply to known affective concepts by analogy: if a concept in the matrix has no value specified for a feature owned by many similar concepts, then by analogy the concept is likely to have that feature as well. In other words, concepts and features that point in similar directions and, therefore, have high dot products, are good candidates for analogies.

After performing TSVD on AffectNet, hereby termed A for the sake of conciseness, a low-rank approximation of it is obtained, that is, a new matrix $\widetilde{A} = U_M \Sigma_M V_M^T$. This approximation is based on minimizing the Frobenius norm of the difference between A and $\widetilde{A}$ under the constraint $rank(\widetilde{A}) = M$. For the Eckart–Young theorem [24], it represents the best approximation of A in the least-square sense, in fact:

$$\min_{\widetilde{A}|rank(\widetilde{A})=M} |A - \widetilde{A}| = \min_{\widetilde{A}|rank(\widetilde{A})=M} |\Sigma - U^* \widetilde{A} V| = \min_{\widetilde{A}|rank(\widetilde{A})=M} |\Sigma - S| \quad (1)$$

assuming that $\widetilde{A}$ has the form $\widetilde{A} = USV^*$, where S is diagonal. From the rank constraint, i.e., S has M non-zero diagonal entries, the minimum of the above statement is obtained as follows:

$$\min_{\widetilde{A}|rank(\widetilde{A})=M} |\Sigma - S| = \min_{s_i} \sqrt{\sum_{i=1}^{n} (\sigma_i - s_i)^2} \quad (2)$$

$$\min_{s_i} \sqrt{\sum_{i=1}^{n} (\sigma_i - s_i)^2} = \min_{s_i} \sqrt{\sum_{i=1}^{M} (\sigma_i - s_i)^2 + \sum_{i=M+1}^{n} \sigma_i^2}$$
$$= \sqrt{\sum_{i=M+1}^{n} \sigma_i^2} \quad (3)$$

Therefore, $\widetilde{A}$ of rank M is the best approximation of A in the Frobenius norm sense when $\sigma_i = s_i$ $(i = 1, \ldots, M)$ and the corresponding singular vectors are the same as those of A. If all but the first M principal components are discarded, common-sense concepts and emotions are represented by vectors of M coordinates.

These coordinates can be seen as describing concepts in terms of 'eigenmoods' that form the axes of AffectiveSpace, i.e., the basis $e_0, \ldots, e_{M-1}$ of the vector space (Fig. 2). For example, the most significant eigenmood, e_0, represents concepts with positive affective valence. That is, the larger a concept's component in the e_0 direction is, the more affectively positive it is likely to be. Concepts with negative e_0 components, then, are likely to have negative affective valence. Thus, by exploiting the information sharing property of TSVD, concepts with the same affective valence are likely to have similar features – that is, concepts conveying the same emotion tend to fall near each other in AffectiveSpace. Concept similarity does not depend on their absolute positions in the vector space, but rather on the angle they make with the origin. For example, concepts such as `beautiful day`, `birthday party`, and `make person happy` are found very close in direction in the vector space, while concepts like `feel guilty`, `be laid off`, and `shed tear` are found in a completely different direction (nearly opposite with respect to the center of the space).

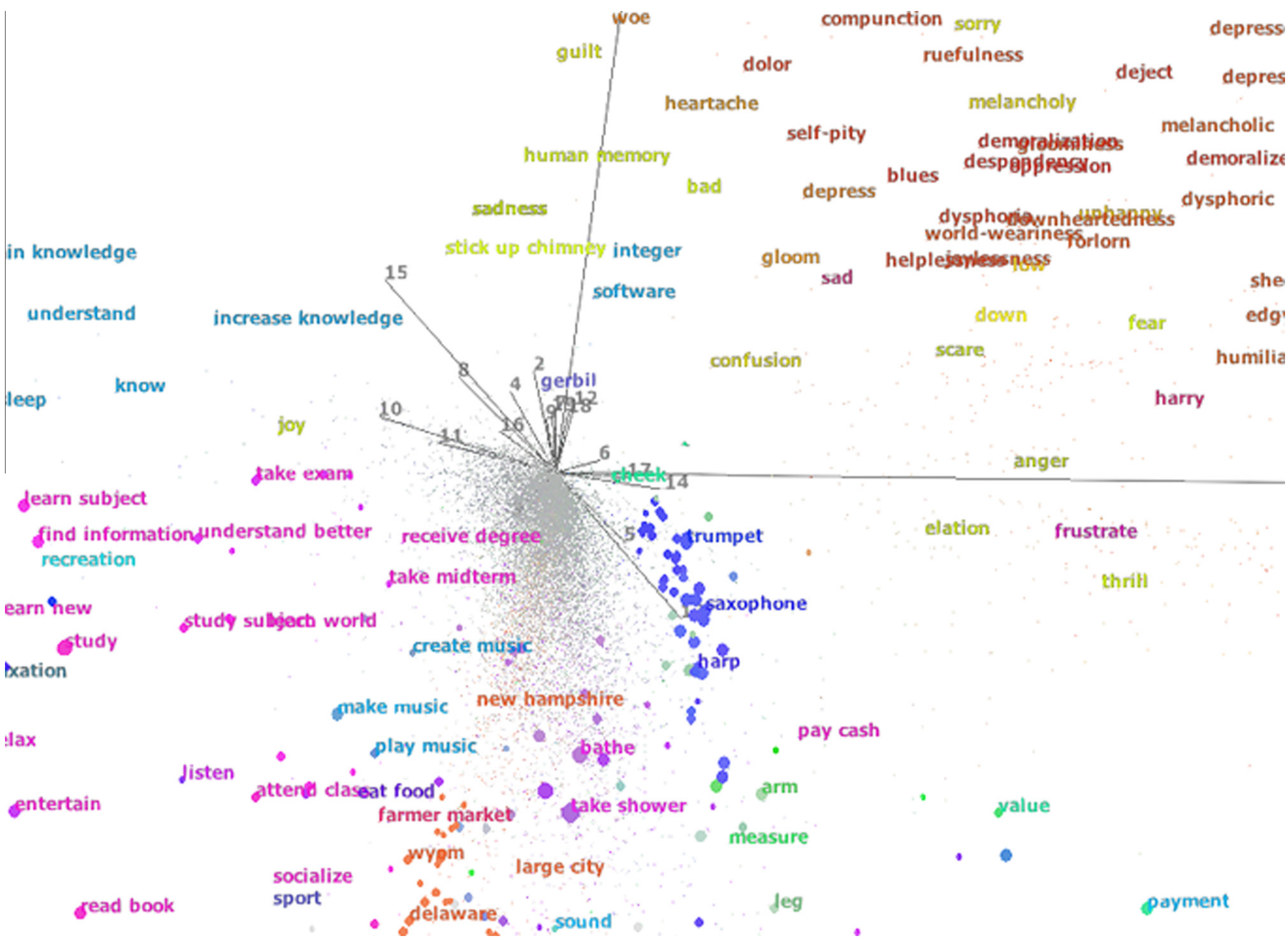


Fig. 2. A sketch of AffectiveSpace. Affectively positive concepts (in the bottom-left corner) and affectively negative concepts (in the up-right corner) are floating in the multi-dimensional vector space.

The key to perform common-sense reasoning is to find a good trade-off for representing knowledge. Since in reality two situations are never exactly the same, no representation should be too concrete, or it will not apply to new situations, but, at the same time, no representation should be too abstract, or it will suppress too many details. Within AffectiveSpace, this knowledge representation trade-off can be seen in the choice of the vector space dimensionality. The number M of singular values selected to build AffectiveSpace is a measure of the trade-off between precision and efficiency in the representation of the affective common-sense knowledge base. The bigger M is, the more precisely AffectiveSpace represents AffectNet's knowledge, but generating the vector space is slower, and so is computing dot products between concepts. The smaller M is, on the other hand, the more efficiently AffectiveSpace represents affective common-sense knowledge both in terms of vector space generation and dot product computation. However, too few dimensions risk not correctly representing AffectNet, as concepts defined with too few features tend to be too close to each other in the vector space and, hence, not easily distinguishable and clusterable.

3.3. The emotion categorization model

The Hourglass of Emotions [2] is an affective categorization model inspired by Plutchik's studies on human emotions [25]. It reinterprets Plutchik's model by organizing primary emotions around four independent but concomitant dimensions, whose different levels of activation make up the total emotional state of the mind. Such a reinterpretation is inspired by Minsky's theory of the mind, according to which brain activity consists of different independent resources and that emotional states result from turning some set of these resources on and turning another set of them off [26]. This way, the model can potentially synthesize the full range of emotional experiences in terms of Pleasantness, Attention, Sensitivity, and Aptitude, as the different combined values of the four affective dimensions can also model affective states we do not have a specific name for, due to the ambiguity of natural language and the elusive nature of emotions.

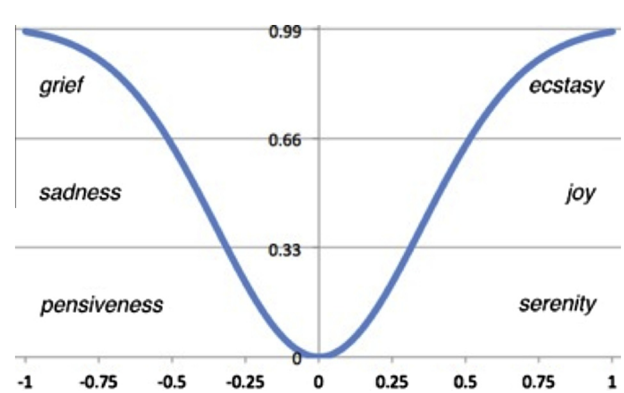


Fig. 3. Pleasantness emotional flow. $G(x)$ models how Pleasantness valence varies with respect to arousal (x), which spans from emotional void (null value) to heightened emotionality (unit value).

The primary quantity we can measure about an emotion we feel is its strength. But, when we feel a strong emotion, it is because we feel a very specific emotion. And, conversely, we cannot feel a specific emotion like fear or amazement without that emotion being reasonably strong. For such reasons, the transition between different emotional states is modeled, within the same affective dimension, using the function $G(x) = -\frac{1}{\sigma\sqrt{2\pi}}e^{-x^2/2\sigma^2}$, for its symmetric inverted bell curve shape that quickly rises up towards the unit value (Fig. 3). In particular, the function models how valence or intensity of an affective dimension varies according to different values of arousal or activation (x), spanning from null value (emotional void) to the unit value (heightened emotionality). Justification for assuming that the Gaussian function (rather than a step or simple linear function) is appropriate for modeling the variation of emotion intensity is based on research into the neural and behavioral correlates of emotion, which are assumed to indicate emotional intensity in some sense. Nobody genuinely knows what function subjective emotion intensity follows, because it has never been truly or directly measured [27].

For example, the so-called Duchenne smile (a genuine smile indicating pleasure) is characterized by smooth onset, increasing to an apex, and a smooth, relatively lengthy offset [28]. More generally, Klaus Scherer has argued that emotion is a process characterized by non-linear relations among its component elements – especially physiological measures, which typically look Gaussian [29]. Emotions, in fact, are not linear [25]: the stronger the emotion, the easier it is to be aware of it. Mapping this space of possible emotions leads to a hourglass shape (Fig. 4). It is worth to note that, in the model, the state of 'emotional void' is a-dimensional, which contributes to determine the hourglass shape. Total absence of emotion can be associated with the total absence of reasoning (or, at least, consciousness) [30], which is not an envisaged mental state as, in human mind, there is never nothing going on. Each affective dimension of the Hourglass model is characterized by six levels of activation (measuring the strength of an emotion), termed 'sentic levels', which represent the intensity thresholds of the expressed or perceived emotion.

These levels are also labeled as a set of 24 basic emotions [25], six for each of the affective dimensions, in a way that allows the model to specify the affective information associated with text both in a dimensional and a discrete form (Table 1). The dimensional form, in particular, is termed 'sentic vector' and it is a four-dimensional *float* vector that can potentially synthesize the full range of emotional experiences in terms of Pleasantness, Attention, Sensitivity, and Aptitude. In the model, the vertical dimension represents the intensity of the different affective dimensions, i.e., their level of activation, while the radial dimension represents K-lines [21] that can activate configurations of the mind, which can either last just a few seconds or years. The model follows the pattern used in color theory and research in order to obtain judgments about combinations, i.e., the emotions that result when two or more fundamental emotions are combined, in the same way that red and blue make purple.

Hence, some particular sets of sentic vectors have special names, as they specify well-known compound emotions (Fig. 5). For example, the set of sentic vectors with a level of Pleasantness $\in$ [G (2/3), G (1/3)), i.e., joy, a level of Aptitude $\in$ [G (2/3), G (1/3)), i.e., trust, and a minor magnitude of Attention and Sensitivity, are termed 'love sentic vectors' since they specify the compound emotion of love (Table 2). More complex emotions can be synthesized by using three, or even four, sentic levels, e.g., joy + trust + anger = jealousy. Therefore, analogous to the way

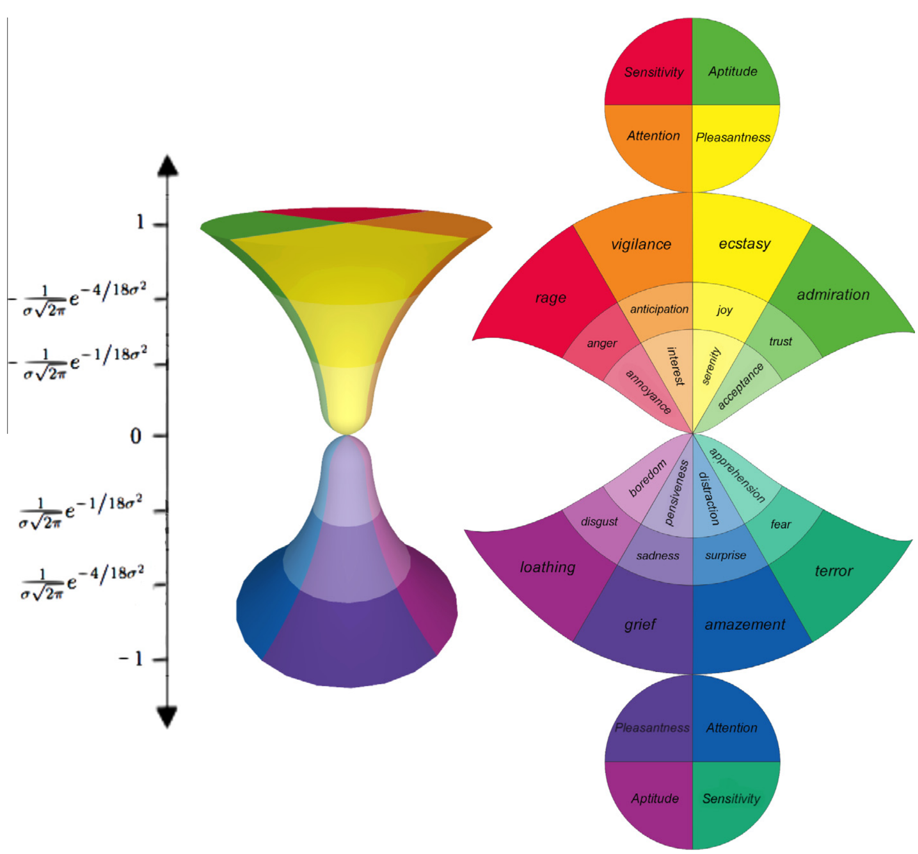


Fig. 4. The 3D model and the net of the Hourglass of Emotions. Since affective states go from strongly positive to null to strongly negative, the model assumes a hourglass shape.

Table 1
The sentic levels of the Hourglass model. Labels are organized into four affective dimensions with six different levels each, whose combined activity constitutes the 'total state' of the mind.

Interval	Pleasantness	Attention	Sensitivity	Aptitude
[G (1), G (2/3))	Ecstasy	Vigilance	Rage	Admiration
[G (2/3), G (1/3))	Joy	Anticipation	Anger	Trust
[G (1/3), G (0))	Serenity	Interest	Annoyance	Acceptance
(G (0), –G (1/3)]	Pensiveness	Distraction	Apprehension	Boredom
(–G (1/3), –G (2/3)]	Sadness	Surprise	Fear	Disgust
(–G (2/3), –G (1)]	Grief	Amazement	Terror	Loathing

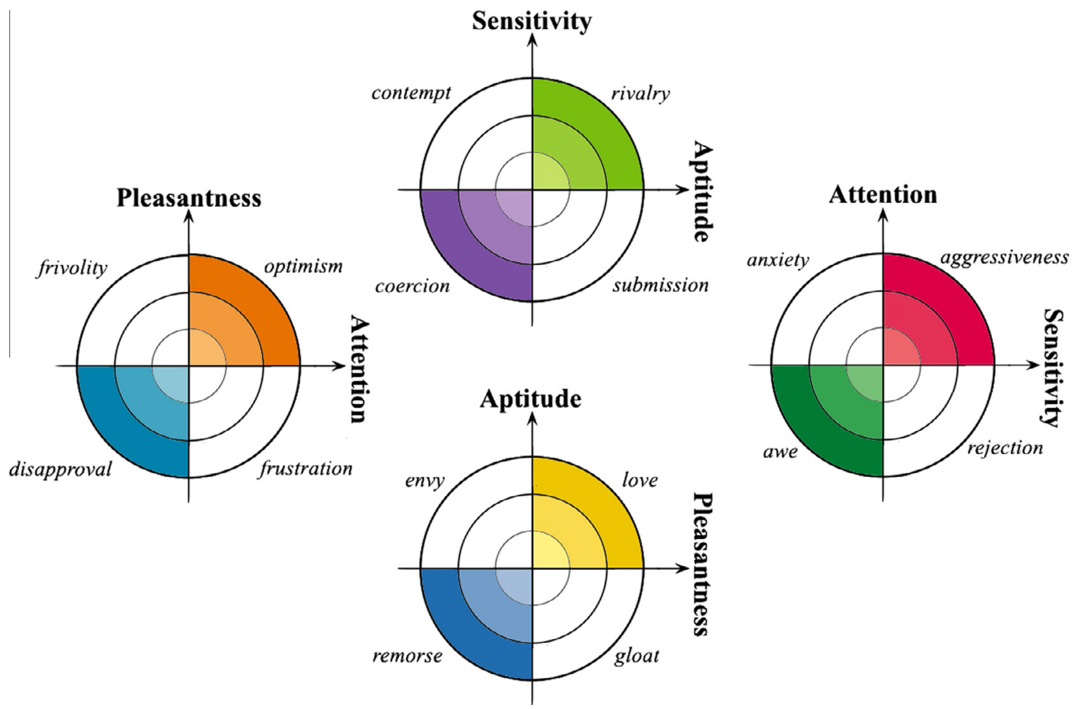


Fig. 5. Hourglass compound emotions of second level. By combining basic emotions pairwise, it is possible to obtain complex emotions.

Table 2
The second-level emotions generated by pairwise combination of the sentic levels of the Hourglass model. The co-activation of different levels gives birth to different compound emotions.

	Attention > 0	Attention < 0	Aptitude > 0	Aptitude < 0
Pleasantness > 0	Optimism	Frivolity	Love	Gloat
Pleasantness < 0	Frustration	Disapproval	Envy	Remorse
Sensitivity > 0	Aggressiveness	Rejection	Rivalry	Contempt
Sensitivity < 0	Anxiety	Awe	Submission	Coercion

primary colors combine to generate different color gradations (and even colors we do not have a name for), the primary emotions of the Hourglass model can blend to form the full spectrum of human emotional experience. Beyond emotion detection, the Hourglass model is also used for polarity detection tasks. Since polarity is strongly connected to attitudes and feelings, it is defined in terms of the four affective dimensions, according to the formula:

$$p = \sum_{i=1}^{N} \frac{Pleasantness(c_i) + |Attention(c_i)| - |Sensitivity(c_i)| + Aptitude(c_i)}{3N} \tag{4}$$

where c_i is an input concept, N the total number of concepts, and 3 the normalization factor (the Hourglass dimensions are defined as float $\in [-1,+1]$). In the formula, Attention is taken as an absolute value since both its positive and negative intensity values correspond to positive polarity values (e.g., `surprise` is negative in the sense of lack of Attention, but positive from a polarity point of view). Similarly, Sensitivity is taken as negative absolute value since both its positive and negative intensity values correspond to negative polarity values (e.g., `anger` is positive in the sense of level of activation of Sensitivity, but negative in terms of polarity). Besides practical reasons, the formula is important because it shows a clear connection between polarity (opinion mining) and emotions (sentiment analysis).

3.4. Analogical reasoning in AffectiveSpace

In order to cluster AffectiveSpace with respect to the Hourglass model (and, hence, reason on the semantic and affective relatedness of natural language concepts), ANNs were recently found to outperform standard clustering techniques, e.g., k-means or k-medoids, for grasping the non-linearities of the vector space [31]. In particular, a new version of SenticNet is built by means of extreme learning machine (ELM) [32] on account of its higher generalization performance (which can be useful for better matching conceptual and affective patterns), faster learning speed (for concept associations to be recalculated every time a new multi-word expression is inserted in the common-sense knowledge base), and lower computational complexity (to facilitate big social data analysis).

3.4.1. Extreme learning machine

The ELM approach was introduced to overcome some issues in back-propagation network [33] training, specifically, potentially slow convergence rates, critical tuning of optimization parameters [34], and the presence of local minima that call for multi-start and re-training strategies [35]. The ELM learning problem settings require a training set, X, of N labeled pairs, where $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathscr{R}^m$ is the ith input vector and $y_i \in \mathscr{R}$ is the associate expected 'target' value; using a scalar output implies that the network has one output unit, without loss of generality.

The input layer has m neurons and connects to the 'hidden' layer (having N_h neurons) through a set of weights $\{\hat{\mathbf{w}}_j \in \mathscr{R}^m; j = 1, \ldots, N_h\}$. The jth hidden neuron embeds a bias term, $\hat{b}_j$, and a nonlinear 'activation' function, $\varphi(\cdot)$; thus, the neuron's response to an input stimulus, $\mathbf{x}$, is:

$$a_j(\mathbf{x}) = \varphi(\hat{\mathbf{w}}_j \cdot \mathbf{x} + \hat{b}_j) \tag{5}$$

Note that (5) can be further generalized to a wider class of functions [36] but for the subsequent analysis this aspect is not relevant. A vector of weighted links, $\bar{\mathbf{w}}_j \in \mathscr{R}^{N_h}$, connects hidden neurons to the output neuron without any bias [37]. The overall output function, $f(\mathbf{x})$, of the network is:

$$f(\mathbf{x}) = \sum_{j=1}^{N_h} \bar{\mathbf{w}}_j a_j(\mathbf{x}) \tag{6}$$

It is convenient to define an 'activation matrix', $\mathbf{H}$, such that the entry $\{h_{ij} \in \mathbf{H}; i = 1, \ldots, N; j = 1, \ldots, N_h\}$ is the activation value of the jth hidden neuron for the ith input pattern. The $\mathbf{H}$ matrix is:

$$\mathbf{H} \equiv \begin{bmatrix} \varphi(\hat{\mathbf{w}}_1 \cdot \mathbf{x}_1 + \hat{b}_1) & \cdots & \varphi(\hat{\mathbf{w}}_{N_h} \cdot \mathbf{x}_1 + \hat{b}_{N_h}) \\ \vdots & \ddots & \vdots \\ \varphi(\hat{\mathbf{w}}_1 \cdot \mathbf{x}_N + \hat{b}_1) & \cdots & \varphi(\hat{\mathbf{w}}_{N_h} \cdot \mathbf{x}_N + \hat{b}_{N_h}) \end{bmatrix} \tag{7}$$

In the ELM model, the quantities $\{\hat{\mathbf{w}}_j, \hat{b}_j\}$ in (5) are set randomly and are not subject to any adjustment, and the quantities $\{\bar{\mathbf{w}}_j, \bar{b}\}$ in (6) are the only degrees of freedom. The training problem reduces to the minimization of the convex cost:

$$\min_{\{\bar{\mathbf{w}}, \bar{b}\}} \|\mathbf{H}\bar{\mathbf{w}} - \mathbf{y}\|^2 \tag{8}$$

A matrix pseudo-inversion yields the unique L_2 solution, as proven in [35]:

$$\bar{\mathbf{w}} = \mathbf{H}^{+}\mathbf{y} \tag{9}$$

The simple, efficient procedure to train an ELM therefore involves the following steps:

1. Randomly set the input weights $\hat{\mathbf{w}}_i$ and bias $\hat{b}_i$ for each hidden neuron;
2. Compute the activation matrix, $\mathbf{H}$, as per (7);
3. Compute the output weights by solving a pseudo-inverse problem as per (9).

Despite the apparent simplicity of the ELM approach, the crucial result is that even random weights in the hidden layer endow a network with a notable representation ability [35]. Moreover, the theory derived in [38] proves that regularization strategies can further improve its generalization performance. As a result, the cost function (8) is augmented by an L_2 regularization factor as follows:

$$\min_{\bar{\mathbf{w}}} \{\|\mathbf{H}\bar{\mathbf{w}} - \mathbf{y}\|^2 + \lambda\|\bar{\mathbf{w}}\|^2\} \tag{10}$$

3.4.2. The SenticNet framework

The emotion categorization framework is designed to receive as input a natural language concept represented according to an M-dimensional space and to predict the corresponding sentic levels for the four affective dimensions involved: Pleasantness, Attention, Sensitivity, and Aptitude. The dimensionality M of the input space stems from the specific design of AffectiveSpace. As for the outputs, in principle each affective dimension can be characterized by an analog value in the range $[-1, 1]$, which represents the intensity of the expressed or received emotion. Indeed, as discussed in Section 3.3, those analog values are eventually remapped to obtain six different sentic levels for each affective dimension.

The SenticNet framework spans each affective dimension separately, under the reasonable assumption that the various dimensions map perceptual phenomena that are mutually independent [2]. As a result, each affective dimension is handled by a dedicated ELM, which addresses a regression problem. Thus, each ELM-based predictor is fed by the M-dimensional vector describing the concept and yields as output the analog value that would eventually lead to the corresponding sentic level. Fig. 6 provides the emotion categorization scheme; here, g_X is the level of activation predicted by the ELM and l_X is the corresponding sentic level.

In theory, one might also implement the emotion categorization scheme showed in Fig. 6 by using four independent predictors based on a multi-class classification schema. In such a case, each predictor would directly yield as output a sentic level out of the six available. However, two important aspects should be taken into consideration. First, the design of a reliable multi-class predictor is not straightforward, especially when considering that several alternative schemata have been proposed in the literature without a clearly established solution. Second, the emotion categorization scheme based on sentic levels stem from an inherently analog model, i.e., the Hourglass of Emotions. This ultimately motivates the choice of designing the four prediction systems as regression problems.

In fact, the emotion categorization scheme shown in Fig. 6 represents an intermediate step in the development of the final emotion categorization system. One should take into account that every affective dimension can in practice take on seven different values: the six available sentic levels plus a 'neutral' value, which in theory correspond to the value $G(0)$ in the emotion categorization model discussed in Section 3.3. In practice, though, the neutral level is assigned to those concepts that are characterized by a level

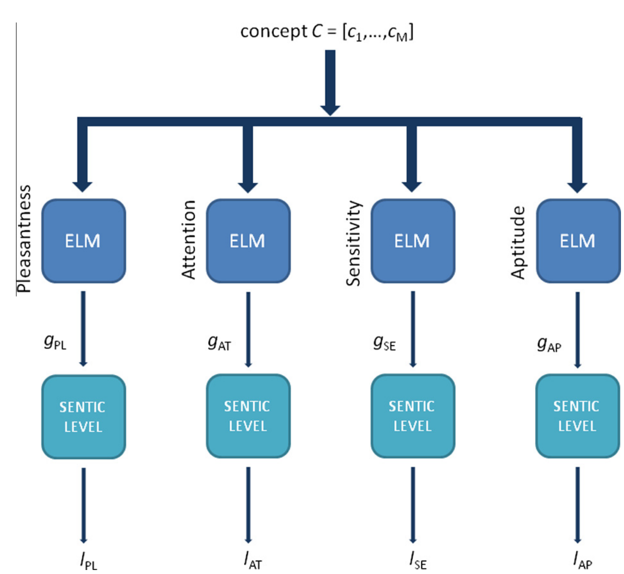


Fig. 6. The emotion categorization scheme describes common-sense concepts in terms of the four Hourglass model's dimensions.

activation that lies in an interval around $G(0)$ in that affective dimension. Therefore, the final framework should properly manage the eventual seven-level scale. To this end, the complete categorization system is set to include a module that is able to predict if an affective dimension is present or absent in the description of a concept. In the latter case, no sentic level should be associated with that affective dimension (i.e., I_x = null).

This task is addressed by exploiting the hierarchical approach presented in Fig. 7. Hence, given a concept and an affective dimension, first a SVM is entitled to decide if a sentic level should be assessed. Accordingly, the ELM-based predictor is asked to assess the level of activation only if the SVM-based classifier determines that a sentic level should be associated with that concept. Otherwise, it is assumed that the neutral level should be associated with that concept (i.e., the corresponding affective dimension is not involved in the description of that concept). Obviously, such a structure is replicated for each affective dimension. Fig. 8 schematizes the complete SenticNet framework for classifying emotional concepts in terms of Pleasantness, Attention, Sensitivity, and Aptitude (from which a polarity value can also be inferred according to Formula (4)).

4. Sentic patterns

Despite being more effective than word-based methods, sentic computing is limited by the fact that does not take into account discourse structure. The bag-of-concepts model [2] can represent the semantics associated to a natural language sentence much better than bags of words. In the bag-of-words model, in fact, a concept such as `cloud computing` would be split into two separate words, disrupting the semantics of the input sentence (in which, for example, the word `cloud` could wrongly activate concepts related to `weather`). The bag-of-concepts model, however, would not be able to correctly infer the polarity of a sentence such as "the phone is nice but slow", in which it would just extract the concepts `phone`, `nice`, and `slow` (which would hardly result in a negative polarity because `nice` and `slow` bear antithetic polarity values that annul each other).

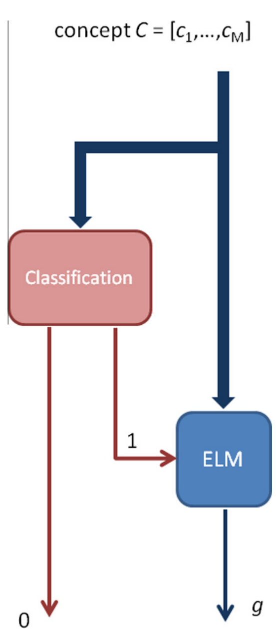


Fig. 7. The hierarchical scheme in which a SVM-based classifier first filters out unemotional concepts and an ELM-based predictor then classifies emotional concepts in terms of the involved affective dimension.

To this end, we further develop and apply linguistic patterns that were initially developed for affective common-sense knowledge acquisition [39]. Such patterns, termed sentic patterns, allow sentiment to flow from concept to concept based on the dependency relation of the input sentence and, hence, to generate a binary (positive or negative) polarity value reflecting the feeling of the speaker. It should be noted that, in some cases, the emotion attributed to a speaker can differ from his/her opinion. For example, (1) conveys a negative sentiment, even though the speaker conveys that he/she is satisfied. There is a gap between the informational and emotional contents of the utterance and we are interested in the latter.

(1) I am barely satisfied.

Similarly, a speaker can convey an objectively negative fact by presenting it in a positive way, as in (2).

(2) It is fortunate that Paul died a horrible death.

Irrespective of Paul's fate, the (possibly psychotic) speaker presents it as a good thing. Hence, the inferred polarity is positive. Nevertheless, in most product or service reviews, the sentiment attributed to the speaker coincides with the opinion expressed. For example, if a sentence attributes a positive property to an object (e.g., "The battery is very good"), the sentiment of the speaker is considered corresponding to his/her evaluation.

In order to compute polarity, sentic patterns leverage on the SenticNet framework (Section 3.4.2) and on the syntactic dependency relations found in the input sentence. It is therefore an explicit approach that rests on linguistic considerations rather than on less interpretable models, such as those produced by most machine learning approaches. The upshot of this approach is that, besides being interpretable, it can take into account complex linguistic structures in a straightforward manner and can easily be modified and adapted.

To the best of our knowledge, the ensemble application of linguistics and common-sense computing has never been explored before in this context. The literature on discourse structure for opinion mining only reports a few techniques based on the propagation of the polarity of seed words according to simple 'opposition constraints' or syntactic dependency relations indicating twists and turns in documents [40–43].

The general template we propose for sentence-level polarity detection is illustrated in Section 4.1, notably by describing how polarity gets inverted (Section 4.1.2) and the way the calculus of polarity takes advantage of the discursive structure of the sentence (Section 4.1.3). The rules associated with specific dependency types are given in Section 4.2. A concrete example is given in Section 4.3.

4.1. General rules

4.1.1. Global scheme

The polarity score of a sentence is a function of the polarity scores associated to its sub-constituents. In order to calculate those polarities, sentic patterns consider each of the sentence's tokens by following their linear order and look at the dependency relations they entertain with other elements. A dependency relation is a binary relation characterized by the following features:

- The **type** of the relation that specifies the nature of the (syntactic) link between the two elements in the relation.
- The **head** of the relation: this is the element that is the pivot of the relation. Core syntactic and semantics properties (e.g., agreement) are inherited from the head.

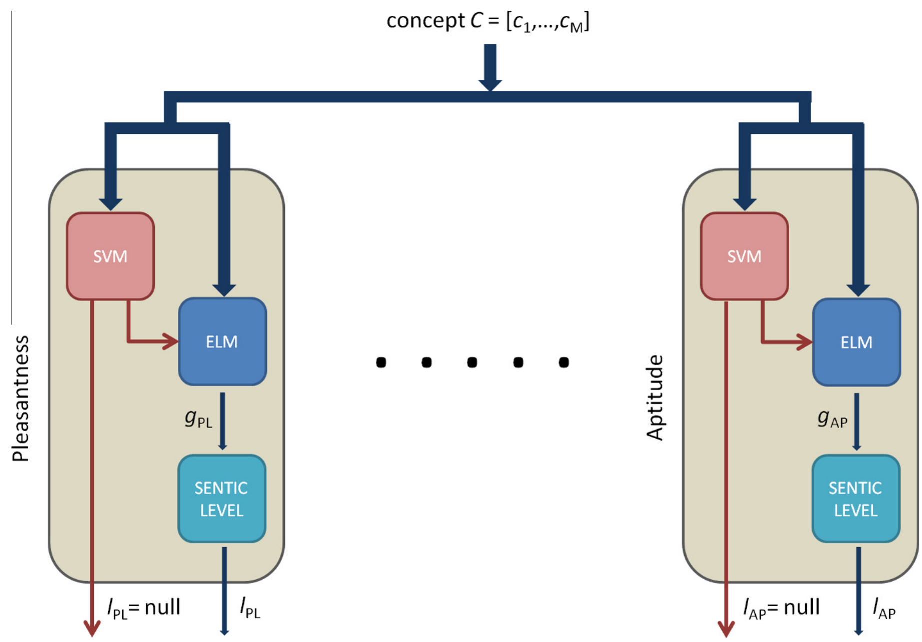


Fig. 8. The SenticNet framework: a hierarchical scheme is adopted to classify emotional concepts in terms of the four affective dimensions of the Hourglass model.

- The **dependent** is the element that depends on the head and which usually inherits some of its characteristics (e.g., number, gender in the case of agreement).

Most of the times, the active token is considered in a relation if it acts as the *head* of the relation, although some rules are an exception. Once the active token has been identified as the trigger for a rule, there are several ways to compute its contribution, depending on how the token is found in SenticNet. The preferred way is to consider the contribution not of the token alone, but in combination with the other element in the dependency relation.

This crucially exploits the fact that SenticNet is not just a polarity dictionary, but also encodes the polarity of complex concepts. For example, in (3), the contribution of the verb `watch` will preferably be computed by considering the complex concept `watch movie` rather than the isolated concepts `watch` and `movie`.

(3) I watched a movie.

If SenticNet has no entry for the multi-word concept formed by the active token and the element related to it, then the way individual contributions are taken into account depends on the type of the dependency relation. The specifics of each dependency type are given in 4.2.

Since SenticNet sometimes encodes sentiment scores for a token and a specific categorization frame, sentic patterns also check whether there is an entry for a frame corresponding to the active token and the part of speech of the other term in the dependency relation.

4.1.2. Polarity inversion

Once the contribution of a token has been computed, sentic patterns check whether the token is in the scope of any polarity switching operator. The primary switching operator is negation: the use of negation on a positive token (4-a) yields a negative polarity (4-b).

(4) a. I liked the movie.
b. I did not like the movie.

However, double negation can keep the polarity of the sentence intact by flipping the polarity twice. For example, (5-a) is positive and (5-b) inverts its polarity. However, (5-c) keeps the polarity of (5-a) identical because in (5-c) *dislike* conveys negative polarity and, hence, nullifies the negation word *not*.

(5) a. I like it.
b. I do not like it.
c. I do not dislike it.

Besides negation, other polarity switching operators include:

- exclusives such as *only*, *just*, *merely*…([44,45])
- adverbs that type their argument as being low, such as *barely*, *hardly*, *least*…

(6) Paul is the least capable actor of his time.

- upper-bounding expressions like *at best*, *at most*, *less than*…
- specific constructions such as the use of past tense along with a comparative form of an adjective as in (7) or counter-factuals expressed by expressions like *would/could have been*

(7) a. My old phone was better. ⇝Negative
b. My old phone was slower. ⇝Positive

Whenever a token happens to be in the scope of such an element, its polarity score is inverted. Finally, inversion also happens when some specific scopeless expressions occur in a sentence, such as *except me*.

A shortcoming of our treatment of negation is that it does not take into account the different effects of negation on various layers of meaning. It is a well known fact in linguistics that some items convey complex meanings on different layers. Presupposition is probably the most studied phenomenon of this kind: both versions of (8) convey that John killed his wife, even though the second version is the negation of the first one ([46,47]).

(8) a. John regrets killing his wife.
b. John does not regret killing his wife.

In the domain of sentiment related expressions, the class of *expressives* has a comparable behavior, even though these elements have been analyzed as conventional implicatures rather than presuppositions ([48]). For example, a verb like *waste* can be analyzed as conveying two distinct pieces of meaning: an event of money spending and a negative evaluation regarding this spending. In some cases, this negative component is not affected by negation: (9) convey that the phone is not worth the money, even though the verb *waste* is embedded under a negation.

(9) a. I will not waste my money on this phone.
b. I do not want to waste my money on this phone.
c. I did not waste my money on this phone.

Therefore, the current treatment of negation needs to be supplemented by a classification of expressions indicating whether their negative (or positive) character has to be analyzed as a *main content*, affected by negation and other operators, or as a *projective content*, i.e., content that 'survives' or is non-canonically affected by an embedding under operators that usually affect truth-conditional content. It might prove difficult to be exhaustive in our description since projection is not a purely semantic problem but is also affected by pragmatic contextual factors [49]. Nevertheless, it is conceivable to rely on a list of elements which convey sentiment on a clearly non-main level and to tune the algorithm to deal with them.

4.1.3. Coordinated and discourse structures

Coordination is an informationally rich structure for which sentic patterns have rules that do not specify which elements should be looked for in SenticNet, rather they indicate how the contributions of different elements should be articulated.

In some cases, a sentence is composed of more than one elementary discourse unit (in the sense of Asher and Lascarides [50]). In such cases, each unit is processed independently and the discourse structure is exploited in order to compute the overall polarity of the sentence, especially if an overt discourse cue is present.

At the moment, we only consider structures that use an overt coordination cue and limit ourselves to adversative markers like *but* and to the conjunctions *and* and *or*.

But *and adversatives*. Adversative items like *but*, *even though*, *however*, *although*, etc. have long been described as connecting two elements of opposite polarities. They are often considered as connecting two full-fledged discourse units in the majority of cases even when the conjuncts involve a form of ellipsis [51,52].

It has also long been observed that, in an adversative structure, the second argument "wins" over the first one [53,54]. For example in (10-a) the overall attitude of the speaker goes against buying the car, whereas just inverting the order of the conjuncts yields the opposite effect (10-b) while keeping the informational content identical.

(10) a. This car is nice but expensive.
b. This car is expensive but nice.

Therefore, when faced with an adversative coordination, sentic patterns primarily consider the polarity of the right member of the construction for the calculation of the polarity of the overall sentence. If it happens that the right member of the coordination is unspecified for polarity, sentic patterns invert the polarity of the left member. The various possibilities are summarized in Table 3.

As shown in the Table, the information retrieved from SenticNet overrides the semantics of adversative constructions when there is a conflict. If an adversative sentic pattern retrieves the same polarity for both conjuncts from SenticNet, it attributes this polarity to the overall sentence, ignoring the opposition marked by the adversative.

Specific heuristics triggered by tense are added to this global scheme. Whenever the two conjuncts share their topic and the second conjunct is temporally anterior to the first one, the overall polarity will be that of the first conjunct. Thus, in (11) since both conjuncts are about the director and since the first one is posterior, the first one drives the polarity calculus.

Table 3
Adversative sentic patterns.

Left conjunct	Right conjunct	Total sentence
Pos.	Neg.	Neg.
Neg.	Pos.	Pos.
Pos.	Undefined	Neg.
Neg.	Undefined	Pos.
Undefined	Pos.	Pos.
Undefined	Neg.	Neg.
Pos.	Pos.	Pos.
Neg.	Neg.	Neg.

(11) This director is making awful movies now, but he used to be good.

Another specific rule is implemented to deal with structures combining *not only* and *but also*, as in (12).

(12) The movie is not only boring but also offensive.

In such cases, *but* cannot be considered an opposition marker. Rather, both its conjuncts argue for the same goal. Therefore, when this structure is detected, the rule applied is the same as for conjunctions using *and* (cf. infra).

And. The conjunction *and* has been described as usually connecting arguments that have the same polarity and are partly independent [55]. Therefore, when a coordination with *and* is encountered, the overall polarity score of the coordination corresponds to the sum of both conjuncts. If only one happens to have a polarity score, this score is used with the addition of a small bonus to represent the fact that *and* connects independent arguments (i.e., the idea that speakers using *and* stack up arguments for their conclusions). In case of conflicts, the polarity of the second conjunct is used.

Or. A disjunction marked by *or* is treated in the same way as the *and* disjunction, i.e., by assuming that in the case one of the conjuncts is underspecified, its polarity is determined by the other. However, there is no added bonus to the polarity score since the semantics of disjunction do not imply independent arguments.

4.2. Dependency rules

In this section, we go over all of the rules that have been implemented to deal with specific dependency patterns. The main goal of these rules is to drive the way concepts are searched in SenticNet. One can roughly distinguish between two classes of dependencies:

- Relations of *complementation* where the dependent is an essential argument of the head.
- Relations of *modification* where the dependent is not sub-categorized by the head and acts as an adjunct.

We begin by focusing on essential arguments of verbs (Section 4.2.1), then move to modifiers (Section 4.2.2) and describe the rest of the rules in Section 4.2.3.

The default behavior of most rules is to build a multi-word concept formed by concatenating the concepts denoted by the head and the dependent of the relation (as exemplified in (3)). This multi-word concept is then looked for in SenticNet. If this is not found, the behaviors of the rule differ.

Therefore, in our descriptions of the rules, we systematically indicate:

- what triggers the rule,
- the behavior of the rule, i.e., the way it constructs complex concepts from the parts of the dependency relation that is under scrutiny.

To simplify the notation, we will use the following notation:

- R denotes the relation type,
- h the head of the relation,
- d the dependent of the relation.

Therefore, writing $R(h, d)$ means that the head h entertains a dependency relation of type R with the dependent d. We use typewriter font to refer to the concept denoted by a token, e.g., `movie` is the concept denoted by both tokens *movie* and *movies*. The concepts are the elements we look for in SenticNet.

4.2.1. Relations of complementation
We consider four relations of complementation, all centered on the verb as the head of the relation. One rule deals with the subject of the verb, the other three cover the different types of object a verb can take: noun phrases, adjective or full clauses.

4.2.1.1. Subject nouns.
Trigger: when the active token is found to be the syntactic subject of a verb.
Behavior: if the multi-word concept `(h,d)` is found it is used to calculate the polarity of the relation, otherwise nothing is done: subsequent relations will be activated later.
Example 1: In (13), *movie* is in a subject relation with *boring*.

(13) The movie is boring.

If the concept `(movie, boring)` is in SenticNet, its polarity will be used. Otherwise, sentic patterns perform a detailed analysis of the relation to obtain the polarity. In this case, sentiment of the `h` is treated as the sentiment of the relation. For example
Example 2: In (14), *relieved* is in a subject relation with *relieved*. Here, the polarity of *trouble* is negative and the polarity of *relieve* is positive. According to our rule sentiment is carried by the *relieve*. So, here the sentence expresses a positive sentiment.

(14) His troubles were relieved.

4.2.1.2. Direct nominal objects. This complex rule deals with direct nominal objects of a verb. Its complexity is due to the fact that the rule attempts to determine the modifiers of the noun in order to compute the polarity.

Trigger: when the active token is head verb of a direct object dependency relation.
Behavior: rather than searching directly for the binary concept `(h,d)` formed by the head and dependent, the rule first tries to look for richer concepts by including modifiers of the nominal object. Specifically, the rule looks for relative clauses and prepositional phrases attached to the noun and if these are found, it looks for multi-word concepts built with these elements. Thus, if the dependent d is head of a relation $R'(d, x)$ where R' is a relation of modification, then sentic patterns will consider the ternary concept `(h,d,x)`. If all fails and the binary concept `(h,d)` is not found either, the sign of the polarity is preferably driven by the head of the relation and in last resort by the dependent.

Example 1: In (15), sentic patterns first look for `(see,movie,in 3D)` and, if this is not found, they will look for `(see,movie)` and then `(see, in 3D)`.

(15) Paul saw the movie in 3D.

`(movie,in 3D)` will not be treated at this stage since it will later be treated by the standard rule for prepositional attachment. If this fails, then the polarity will be that of `see` and eventually `movie` assuming that `see` has not been found.

4.2.1.3. Adjective and clausal complements. 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: first, sentic patterns look for the binary concept `(h,d)`. If it is found, the relation inherits its polarity properties. If not found:

- if both elements `h` and `d` are independently found in SenticNet, then we take sentiment of `d` as the sentiment of the relation.
- if the dependent `d` alone is found in SenticNet, its polarity is attributed to the relation

Example: in (16), *smells* is the head of a dependency relation with *bad* as the dependent.

(16) This meal smells bad.

The relation inherits the polarity of `bad`.

4.2.1.4. 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 predicate of the relation[1]
Behavior: as for the case of direct objects, sentic patterns try to determine the structure of the dependent of the head verb. Here the dependent is itself a verb, therefore, sentic patterns attempt to establish whether there is a relation R' such that $R'(d, x)$ and where x is a direct object or a clausal complement of d. Sentic patterns are therefore dealing with three elements: the head/matrix verb (or predicate) h, the dependent predicate d, and the (optional) complement of the dependent predicate x. Once these have been identified, sentic patterns first test the existence of the ternary concept `(h,d,x)`. If this is found in SenticNet, the relation inherits its properties. If not found, sentic patterns then check for the presence of individual elements in SenticNet.

- If `(d,x)` is found as well as `h` or if all three elements `h`, `d` and `x` are independently found in SenticNet, then the final sentiment score will be that of `(d,x)` or that calculated from `d` and `x` by following the appropriate rule. The head verb then affects the sign of this score. The rules for computing the sign are summarized in Table 4 by indicating the final sign of the score in the function of the signs of individual scores of each of the three relevant elements.
- If the dependent verb `d` is not found in SenticNet but the head verb `h` and the dependent's complement `x` can be

[1] Usually the token will be a verb, although when the tensed verb is a copula, the head of the relation is rather the complement of the copula.

Table 4
Polarity algebra for open clausal complements.

Matrix predicate (h)	Dependent predicate (d)	Dep. comp. (x)	Overall polarity	Example
Pos.	Pos.	Pos.	Pos.	(17-a)
Pos.	Pos.	Neg.	Neg.	(17-b)
Pos.	Neg.	Pos.	Neg.	(17-c)
Pos.	Neg.	Neg.	Pos.	(17-d)
Neg.	Pos.	Pos.	Neg.	(17-e)
Neg.	Pos.	Neg.	Neg.	(17-f)
Neg.	Neg.	Pos.	Neg.	(17-g)
Neg.	Neg.	Neg.	Neg.	(17-h)
Pos.	Neutral	Pos.	Pos.	(17-i)
Pos.	Neutral	Neg.	Neg.	(17-j)
Neg.	Neutral	Pos.	Neg.	(17-k)
Neg.	Neutral	Neg.	Neg.	(17-l)

found, then they are used to produce a score with a sign again corresponding to the rules stated in Table 4.

Example: In order to illustrate every case presented in Table 4, we use the paradigm in (17). For each example, the final sign of the polarity is the one mentioned in Table 4. The examples assume the following:

- *h*, the matrix predicate, is either:
  - *perfect*, which has a positive polarity,
  - *useless*, which has negative polarity.
- *d*, the dependent verb, is either:
  - *gain*, which has a positive polarity,
  - *lose*, which has a negative polarity,
  - *talk*, which is not found isolated in SenticNet, i.e., is considered neutral here.
- *x*, the complement of the dependent verb, is either:
  - *money*, which has a positive polarity,
  - *weight*, which has a negative polarity[2]

It must be remembered that for such examples, we assume that the sentiment expressed by the speaker corresponds to his/her opinion on whatever *this* refers to in the sentence: if the speaker is positive about the thing he/she is talking about, we will consider that he/she is expressing positive sentiments overall.

(17) a. This is perfect to gain money.
b. This is perfect to gain weight.
c. This is perfect to lose money.
d. This is perfect to lose weight.
e. This is useless to gain money.
f. This is useless to gain weight.
g. This is useless to lose money.
h. This is useless to lose weight.
i. This is perfect to talk about money.
j. This is perfect to talk about weight.
k. This is useless to talk about money.
l. This is useless to talk about weight.

4.2.2. Modifiers

Modifiers, by definition, affect the interpretation of the head they modify. This explains why in most of the following rules the dependent is the guiding element for the computation of polarity.

[2] The negative score associated with *weight* does not reflect a deliberate opinion on the meaning of term. This score is extracted from Senticnet and as such has been automatically computed as explained in [7]. Thus, even though the term might not appear negative at first glance, its sentiment profile is nevertheless biased towards the negative.

4.2.2.1. 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: first, the multi-word concept `(h,d)` is looked for in SenticNet. If not found, then polarity is preferably driven by the modifier `d` if it is found in SenticNet, and `h` as a last resort.
Example: in (18) both sentences involve elements of opposite polarities. The rule ensures that the polarity of the modifiers is the one that is used and not that of the head of the relation: e.g., in (18-b) *beautifully* takes precedence over *depressed.*

(18) a. Paul is a bad loser.
b. Mary is beautifully depressed.

Unlike other NLP tasks such as emotion recognition, in fact, the main aim of sentiment analysis is to infer the polarity expressed by the speaker (i.e., the person who writes the review of a hotel, product, or service). Hence, a sentence such as (18-b) would be positive as it reflects the positive sentiment of the speaker.

4.2.2.2. Relative clauses.
Trigger: the rule is activated when the active token is modified by a relative clause, restrictive or not. The dependent usually is the verb of the relative clause.
Behavior: if the binary concept `(h,d)` is found in SenticNet, it assigns polarity to the relation, otherwise the polarity is assigned (in order of preference):

- By the value of the dependent verb `d` if it can be found.
- By the value of the active token `h` if it is found in SenticNet.

Example: in (19) is in relation with *love* which acts as a modifier in the relative clause.

(19) I saw the movie you love.

Assuming `(love, movie)` is not in SenticNet and that `love` is, then the latter will contribute to the polarity score of the relation. If neither of these two is in SenticNet, then the dependency will receive the score associated with `movie`. In the case of (19), the polarity will be inherited at the top level because the main verb *see* is neutral. However, the overall polarity of a sentence like (20) is positive since, in case the subject is a first person pronoun, the sentence directly inherits the polarity of the main verb, here *like* (see Section 4.2.3 for more details).

(20) I liked the movie you love.

Similarly, (21) will bear an overall negative sentiment because the main verb is negative.

(21) I disliked the movie you love.

4.2.2.3. Prepositional phrases. Although prepositional phrases (PPs) do not always act as modifiers, we introduce them in this section as the distinction does not really matter for their treatment (and also because the Stanford dependency parser on which we rely does not differentiate between modifier and non-modifier PPs).

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 the PP attaches, and the dependent is the head of the phrase embedded in the PP. This means that the active element is **not** one of the two arguments of the relation but participates in defining the type of the relation.
Behavior: instead of looking for the multi-word concept formed by the head *h* and dependent *d* of the relation, sentic patterns use the preposition *prep* (corresponding to the active token) to build a ternary concept `(h, prep, d)`. If this is not found, then they look for the binary concept `(prep, d)` formed by the preposition and the dependent and use the score of the dependent `d` as a last resort. This behavior is overridden if the PP is found to be a modifier of an NP that acts as the direct object.
Example 1: in (22), 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).

(22) Bob hit Mary with a hammer.

Therefore, sentic patterns first look for the multi-word concept `(hit, with, hammer)` and, if this is not found, they look for `(with, hammer)` and finally `hammer` itself.
Example 2: in (23), the PP headed by *in* is a modifier of the verb *complete* which is positive in SenticNet. *Terrible way* is however negative and because it directly modifies the verb, it is this element which gives the overall polarity.

(23) Paul completed his work in a terrible way.

Example 3: in (24), the PP introduced by *in* is attached to the direct object of the predicate *is a failure*.

(24) This actor is the only failure in an otherwise brilliant cast.

Here, sentic patterns will ignore the contribution of the PP since the main sentiment is carried by the combination of the verb and its object, i.e., it is negative.

4.2.2.4. 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 the binary concept `(h,d)` is found in SenticNet it is used for calculating the score. Otherwise, the rule assigns polarity by considering first the dependent `d` then the head `h`.
Example: in (25), *playing* modifies *slows*, if the multi-word concept `(slow, play)` is not in SenticNet, then `play` will be considered and finally `slow`.

(25) The machine slows down when the best games are playing.

4.2.2.5. Untyped dependency. Sometimes the dependency parser that we use detects two elements entertaining a dependency relation but is unable to type it properly. In this case, if the multi-word concept `(h,d)` is not found, then the polarity is computed by considering the dependent `d` alone.

4.2.3. Other rules

4.2.3.1. First person heuristics. On top of the rules presented so far, we implemented a specific heuristic when the subject of a sentence is the first person pronoun. When this is the case, the sentiment is essentially carried by the head verb of the relation. This is motivated by a contrast as the one in (26):

(26) a. Paul likes bad movies.
b. I like bad movies.

Whereas (26-a) is a criticism of Paul and his tastes, (26-b) is speaker-oriented as he/she conveys his/her (maybe peculiar) tastes. What matters is that the speaker of (26-b) is being positive and uses the verb *like*. This overrides the calculus that would yield a negative orientation as in (26-a) by considering the combination of `like` and `bad movies`.

Similarly, in (27) the use of the first person overrides the effect produced by the relative clause *which you like*. The overall sentiment is entirely driven by the use of the verb *hate* which is negative.

(27) I hate the movie which you like.

4.3. Walking through an example

Here we describe how the global sentiment for a complex example is computed. This is made in order to show how the sentiment flows in the treatment of a sentence. We will base our presentation on the (artificial) case of (28).

(28) The producer did not understand the plot of the movie inspired by the book and preferred to use bad actors.

The relevant dependency relations here are (with the concept arguments given between parentheses):

1. A general coordination with *and (understand, preferred)*.
2. Two subject relations *(understand, producer)* and *(preferred, producer)*.
3. A direct object relation *(understand, plot)*.
4. A prepositional attachment typed by *of (plot, movie)*.
5. A participial modification *(plot, inspired)*.
6. A open clausal complement relation *(preferred, use)*.
7. Another direct object relation *(use, actors)*.
8. An adjective modifying a noun *(actors, bad)*.

First, the discourse structure parser detects that the sentence has two discourse units conjoined by *and*. The final polarity will thus be a function of the elements π_1 = *The producer did not understand the plot of the movie based on the book* and π_2 = *[the producer] preferred to use bad actors*.

The computation of π_1 entails checking the relations in the following order:

- The subject relation *(understand, producer)* is treated to check whether the multi-word concept `(producer understand)` can be found in SenticNet. This is not the case so nothing is done.
- The relations having the verb *understand* as their head are looked at. Here there is only the direct object relation. In this relation the dependent object is modified in two ways:

– by a prepositional phrase,
– by a participial modifier.

Thus, sentic patterns will first try to find the multi-word concept `(understand, plot, of, movie)`. Since this one is not found, they will try `(understand, plot, inspired)` which is not in SenticNet either. In the end, sentic patterns fall back on the concept `(understand, plot)` which is found in SenticNet. Therefore, the polarity stack is set at the corresponding positive value.

- Since the previous polarity is in the scope of a sentential negation, the sign of the previous score is switched to give a negative value.

All relations having been used in the first conjunct, sentic patterns move on to π_2.

- The open clausal modification rule determines the dependent of the dependent. This case means identifying *actors* as the direct object of *use*.
- Since *actors* is modified by *bad*, it will inherit its negative orientation.
- The only relevant elements for the computation of the polarity due to the open clausal complement are *prefer* (which is positive) and *actor* (negative because of its adjectival modification). Therefore, the final polarity score is also negative.

In the end, both of the *and* conjuncts are negative meaning that the overall polarity of the sentence is also negative with a value equal to the sum of the scores of each conjunct.

5. Machine learning aid

Despite much more efficient than bag-of-words and bag-of-concepts models, the proposed approach is still limited by the richness of the knowledge base and the set of dependency-based rules. In order to be able to make a good guess even when no sentic pattern is matched or SenticNet entry found, we resort to machine learning. In particular, we use two well-known sentiment analysis datasets (Section 5.1), a set of four features per sentence (Section 5.2) and an ELM classifier (Section 5.3) to label stretches of text as positive or negative.

5.1. Datasets used

5.1.1. Movie review dataset

We used a dataset derived from the corpus developed by Pang and Lee [56]. This corpus includes 1000 positive and 1000 negative movie reviews authored by expert movie reviewers, collected from rottentomatos.com, with all text converted to lowercase and lemmatized, and HTML tags removed. Originally, Pang and Lee manually labeled each review as positive or negative. Later, Socher et al. [57] annotated this dataset at sentence level. They extracted 11855 sentences from the reviews and manually labeled them using a fine grained inventory of five sentiment labels: *strong positive*, *positive*, *neutral*, *negative*, and *strong negative*.

Since in this work we considered only binary classification, we removed from the dataset the sentences marked as neutral and reduced the labels on the remaining sentences to positive or negative. Thus, our final movie dataset contained 9613 sentences, of which 4800 were labeled as positive and 4813 negative. We divided this dataset into 6920 sentences for training and 2693 for testing.

5.1.2. Blitzer dataset

We also used the dataset introduced by Blitzer et al. [58], which consists of product reviews in seven different domains. For each domain there are 1000 positive and 1000 negative reviews. We only used reviews under the *electronics* category. We randomly extracted from them 7210 non-neutral sentences, 3505 from positive reviews and 3505 from negative ones, and manually annotated them as positive or negative. Note that the polarity of individual sentences does not always coincide with the overall polarity of the review: for example, in a negative review we found sentences such as "This is a good product – sounds great", "gets good battery life", "everything you 'd hope for in an iPod dock" or "It is very cheap", which we labeled as positive. Specifically, we obtained 3800 sentences marked as positive and 3410 as negative.

5.2. Feature set

5.2.1. Common-sense knowledge features

Common-sense knowledge features consist of concepts represented by means of AffectiveSpace. In particular, concepts extracted from text through the semantic parser are encoded as 100-dimensional real-valued vectors and then aggregated into a single vector representing the sentence by coordinate-wise summation:

$$x_i = \overset{N}{\underset{j=1}{m}} x_{ij},$$

where x_i is the ith coordinate of the sentences feature vector, $i = 1, \ldots, 100$; x_{ij} is the ith coordinate of its jth concepts vector, and N is the number of concepts in the sentence.

5.2.2. Sentic features

The polarity scores of each concept extracted from the sentence were obtained from SenticNet and summed up to produce a single scalar feature.

5.2.3. Part of speech features

This feature was defined by number of adjectives, adverbs, and nouns in the sentence, which gave three distinct features.

5.2.4. Modification features

This is a single binary feature. For each sentence, we obtained its dependency tree from the dependency parser. We analyzed this tree to determine whether there is any word modified by a noun, adjective, or adverb. The modification feature was set to 1 if we found any modification relation in the sentence; it was set to 0 otherwise.

5.2.5. Negation features

Similarly, the negation feature was a single binary feature determined by whether there was any negation in the sentence. It is important because the negation can invert the polarity of the sentence.

5.3. Classification

A SVM and a ELM classifiers are trained, over the training portion of the movie review dataset, using the sentence feature set described above. We found that ELM outperformed SVM in terms of both accuracy and training time. Specifically, on the testing portion of the movie review dataset, we obtained 67.35% accuracy with ELM and 65.67% with SVM. The same model trained on the movie dataset sentences but applied to our Blitzer-based dataset described in Section 5.1.2, achieved 72.00% accuracy with ELM but a much lower accuracy with SVM. Conversely, we trained the model on our Blitzer-derived dataset and evaluated it on our movie test set, obtaining 66.25% accuracy with ELM and 61.00% with SVM.

Hence, whenever we are unable to process a sentence through SenticNet and sentic patterns, we use the trained ELM classifier to make a good guess on the sentence polarity, based on the available features.

6. Experimental results and discussion

The proposed approach (available as a demo at http://sentic.net/demo) was tested on two datasets: the movie review dataset described in Section 5.1.1 and the Blitzer-derived dataset described in Section 5.1.2. As shown by results below, the best accuracy is achieved when applying an ensemble of knowledge-based analysis (Section 4) and machine-learning classification (Section 5), as the latter can act as reserve for the former when no match is found in SenticNet (Fig. 9).

6.1. Results

6.1.1. Results on the movie review dataset

We evaluated our approach on the Movie Review Dataset and obtained an accuracy of 86.21%, which is better than the state-of-the-art accuracy reported by Socher et al. [57] (85.40%). Table 5 shows our results with ensemble classification and without ensemble classification.

Table 6 presents the comparison of the proposed system with well-known state of the art.

6.1.2. Results on the Blitzer-derived dataset

On our Blitzer-derived dataset described in Section 5.1.2, an accuracy of 87.00% was achieved at sentence level (see Table 7). Since we developed this corpus in frame of the present work, there are no available results for comparative evaluation with other systems. However, the accuracy achieved is superior to the average results reported by state-of-the-art systems on other corpora.

Table 5
Results obtained using different algorithms on our movie review dataset.

Algorithm	Precision (%)
Sentic patterns	84.15
Machine learning	67.35
Ensemble classification	86.21

Table 6
Comparison with the state of the art.

System	Precision (%)
Socher et al. [59]	80.00
Socher et al. [57]	85.40
Proposed method	86.21

6.2. Discussion

The proposed approach outperforms state-of-the-art methods on the movie dataset and shows even better results on the Blitzer-derived dataset, which proves that our system is robust and not biased towards a particular domain.

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 sentence. Both datasets, however, contain ungrammatical sentences which penalize results.

Next, we discuss the performance of the proposed architecture on various linguistic patterns and types of sentence structure.

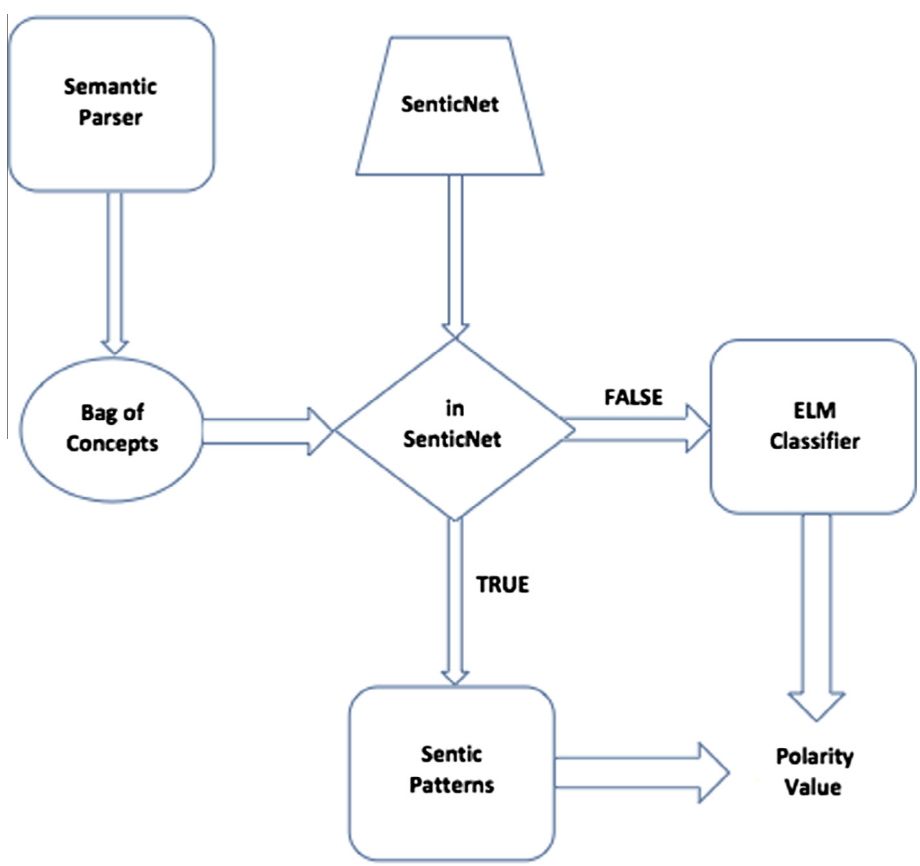


Fig. 9. Flowchart of the sentence-level polarity detection system. Natural language text is first deconstructed into concepts. If these are found in SenticNet, sentic patterns are applied. If none of the concepts are available in SenticNet, the ELM classifier is employed.

Table 7
Results obtained using different algorithms on our Blitzer-derived dataset.

Algorithm	Precision (%)
Sentic patterns	85.15
Machine learning	72.00
Ensemble classification	87.00

Table 8
Performance of the proposed system on sentences with conjunctions and comparison with state of the art.

System	AND	BUT
Socher et al. [57]	N/A	41.00%
Proposed method	88.15%	83.67%

Table 9
Comparison of the performance between the proposed system and state-of-the art systems on different sentence structures.

Sentence	Socher et al. [57]	Proposed system	Correct polarity
The room is so small to stay	Neutral	Negative	Negative
The tooth hit the pavement and broke	Positive	Negative	Negative
I am one of the least happy people in the world	Neutral	Negative	Negative
I love starbucks but they just lost a customer	Neutral	Negative	Negative
I doubt that he is good	Positive	Negative	Negative
Finally, for the beginner there are not enough conceptual clues on what is actually going on	Positive	Negative	Negative
I love to see that he got injured badly	Neutral	Positive	Positive
I love this movie though others say it's bad	Neutral	Positive	Positive
Nothing can be better than this	Negative	Positive	Positive
The phone is very big to hold	Neutral	Negative	Negative

6.2.1. Effect of conjunctions

Sentiment is often very hard to identify when sentences have conjunctions. We tested the performance of the proposed system on two types of conjunctions: *and* and *but*. High accuracy was achieved for both conjunctions. However, the accuracy on sentences containing *but* was somewhat lower because some such sentences do not match sentic patterns.

Table 8 shows the accuracy of the proposed system on sentences containing *but* and *and* and compares it with the state of the art. To the best of our knowledge, no state-of-the-art system reports accuracy on sentences containing *and*.

6.3. Effect of discourse markers

We used a discourse parser developed by Lin et al. [60] to analyze the discourse structure of sentences. Of the 251 sentences in the movie review and the Blitzer dataset that contain discourse markers (*though*, *although*, *despite*), we have correctly identified sentiment in 83.41% sentences. In some cases, the discourse parser failed to detect the discourse structure of sentences such as *So, although the movie bagged a lot, I give very low rating.*

6.4. Effect of negation

Through the linguistic rules described in Section 4.1.2, we detected negation and studied its impact on sentence polarity. Overall, we achieved 91% accuracy on polarity detection from sentences with negation.

Table 10
Performance of the system on the sentences bearing same meaning but with different words.

Sentence	Socher et al. [57]	Proposed system	Correct polarity
I feel **bad** when Messi scores fantastic goals	Neutral	Negative	Negative
I feel **bored** when Messi scores fantastic goals	Negative	Negative	Negative
I feel **upset** when Messi scores fantastic goals	Positive	Negative	Negative

Socher et al. [57] state that negation does not always reverse the polarity. According to their theory, the sentence “I do not like the movie” does not bear any negative sentiment but rather is neutral. Another example: ”The movie is not terrible”; their theory suggests that this sentence does not say that the movie is good but rather says that it is less bad, so this sentence bears negative sentiment. However, in our annotation we did not follow this theory. We believe that the expression “not bad” implies satisfaction; thus, we annotated such a sentence as positive. Conversely, “not good” implies dissatisfaction and, thus, bears negative sentiment. Based on this argument, we consider the sentence “The movie is not terrible” to be positive.

6.5. Examples of differences between the proposed system and state-of-the-art systems

Table 9 shows examples of various linguistic patterns and the performance of our system across different sentence structures.

Examples in Table 10 show that the proposed system gives consistent results on sentences carrying the same meaning although they use different words. In this example, we change the negative sentiment bearing word in the sentence: in the first variant it is **bad**, in the second variant it is **bored**, and in the third variant it is **upset**. In each case, our system detects the sentiment correctly. This analysis also illustrates inconsistency of the state-of-the-art approaches, given that the system [57] achieves the highest accuracy compared with other existing state-of-the-art systems.

7. Conclusion

Between the dawn of civilization through 2003, there were just a few dozens exabytes of information on the Web. Today, that much information is created weekly. The advent of the Social Web has provided people with new tools for creating and sharing, in a time and cost efficient way, their own contents, ideas, and opinions with virtually the millions of people connected to the World Wide Web. This huge amount of useful information, however, is mainly unstructured as specifically produced for human consumption and, hence, it is not directly machine-processable.

Concept-level sentiment analysis can help with this as, unlike other word-based approaches, it focuses on a semantic analysis of text through the use of web ontologies or semantic networks and, hence, allows for the aggregation of conceptual and affective information associated with natural language opinions. Concept-level sentiment analysis, however, is limited by the richness of the knowledge base and by the fact that the bag-of-concepts model, despite more sophisticated than bag-of-words, misses out important discourse structure information that is key for properly detecting the polarity conveyed by natural language opinions.

In this work, we introduced a novel paradigm to concept-level sentiment analysis that merges linguistics, common-sense computing, and machine learning for improving the accuracy of polarity detection. By allowing sentiments to flow from concept to concept based on the dependency relation of the input sentence, in particular, we achieve a better understanding of the contextual role of each concept within the sentence and, hence, obtain a polarity detection engine that outperforms state-of-the-art statistical methods.

There are a number of possible extensions of this work. One is to further develop sentic patterns, which we showed to play a key role in concept-level sentiment analysis. Another direction is to expand the common-sense knowledge base, as well as the accuracy of discourse and dependency parsing techniques.

In general, in fact, the accuracy of the ELM classifier is lower than the one obtained by means of sentic patterns because these better grasp the semantics conveyed by the input sentence. In the future, we aim to limit the use of machine learning techniques as much as possible, in order to shift to the exclusive use of semantics and, hence, better mimic the way we process language as human text processors.

Acknowledgements

This work was supported by a grant from Singapore Ministry of Defence (MINDEF) under Project MINDEF-NTU-JPP/12/02/05 (M406340000).

References

[1] E. Cambria, B. White, Jumping NLP curves: a review of natural language processing research, IEEE Comput. Intell. Mag. 9 (2) (2014) 48–57.
[2] E. Cambria, A. Hussain, Sentic Computing: Techniques, Tools, and Applications, Springer, Dordrecht, Netherlands, 2012.
[3] M. Bradley, P. Lang, Affective norms for English words (ANEW): stimuli, instruction manual and affective ratings, Tech. Rep., The Center for Research in Psychophysiology, University of Florida, 1999.
[4] C. Strapparava, A. Valitutti, WordNet-Affect: an affective extension of WordNet, in: LREC, Lisbon, 2004, pp. 1083–1086.
[5] C. Bazzanella, Emotions, language and context, in: E. Weigand (Ed.), Emotion in Dialogic Interaction, Advances in the Complex, Benjamins, Amsterdam, 2004, pp. 59–76.
[6] A. Esuli, F. Sebastiani, SentiWordNet: a publicly available lexical resource for opinion mining, in: LREC, 2006.
[7] E. Cambria, D. Olsher, D. Rajagopal, SenticNet 3:a common and common-sense knowledge base for cognition-driven sentiment analysis, in: AAAI, Quebec City, 2014, pp. 1515–1521.
[8] A. Tsai, R. Tsai, J. Hsu, Building a concept-level sentiment dictionary based on commonsense knowledge, IEEE Intell. Syst. 28 (2) (2013) 22–30.
[9] S. Poria, A. Gelbukh, A. Hussain, N. Howard, D. Das, S. Bandyopadhyay, Enhanced SenticNet with affective labels for concept-based opinion mining, IEEE Intell. Syst. 28 (2) (2013) 31–38.
[10] C. Hung, H.-K. Lin, Using objective words in SentiWordNet to improve sentiment classification for word of mouth, IEEE Intell. Syst. 28 (2) (2013) 47–54.
[11] C. Bosco, V. Patti, A. Bolioli, Developing corpora for sentiment analysis and opinion mining: a survey and the Senti-TUT case study, IEEE Intell. Syst. 28 (2) (2013) 55–63.
[12] A. Weichselbraun, S. Gindl, A. Scharl, Extracting and grounding context-aware sentiment lexicons, IEEE Intell. Syst. 28 (2) (2013) 39–46.
[13] L. García-Moya, H. Anaya-Sanchez, R. Berlanga-Llavori, A language model approach for retrieving product features and opinions from customer reviews, IEEE Intell. Syst. 28 (3) (2013) 19–27.
[14] R. Xia, C. Zong, X. Hu, E. Cambria, Feature ensemble plus sample selection: a comprehensive approach to domain adaptation for sentiment classification, IEEE Intell. Syst. 28 (3) (2013) 10–18.
[15] G. Di Fabbrizio, A. Aker, R. Gaizauskas, Summarizing on-line product and service reviews using aspect rating distributions and language modeling, IEEE Intell. Syst. 28 (3) (2013) 28–37.
[16] V. Perez-Rosas, R. Mihalcea, L.-P. Morency, Multimodal sentiment analysis of Spanish online videos, IEEE Intell. Syst. 28 (3) (2013) 38–45.
[17] M. Wollmer, F. Weninger, T. Knaup, B. Schuller, Congkai, K. Sagae, L.-P. Morency, YouTube movie reviews: in, cross, and open-domain sentiment analysis in an audiovisual context, IEEE Intell. Syst. 28 (3) (2013) 46–53.
[18] E. Cambria, D. Rajagopal, D. Olsher, D. Das, Big social data analysis, in: R. Akerkar (Ed.), Big Data Computing, Chapman and Hall/CRC, 2013, pp. 401–414 (Chapter 13).
[19] C. Manning, Part-of-speech tagging from 97% to 100%: is it time for some linguistics?, in: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 6608, Springer, 2011, pp. 171–189.
[20] C. Paice, Another stemmer, SIGIR Forum 24 (3) (1990) 56–61.
[21] M. Minsky, The Society of Mind, Simon and Schuster, New York, 1986.
[22] G. Fauconnier, M. Turner, The Way We Think: Conceptual Blending and the Mind's Hidden Complexities, Basic Books, 2003.
[23] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bureau Standards 45 (4) (1950) 255–282.
[24] C. Eckart, G. Young, The approximation of one matrix by another of lower rank, Psychometrika 1 (3) (1936) 211–218.
[25] R. Plutchik, The nature of emotions, Am. Scientist 89 (4) (2001) 344–350.
[26] M. Minsky, The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind, Simon & Schuster, New York, 2006.
[27] L. Barrett, Solving the emotion paradox: categorization and the experience of emotion, Personal. Soc. Psychol. Rev. 10 (1) (2006) 20–46.
[28] E. Krumhuber, A. Kappas, Moving smiles: the role of dynamic components for the perception of the genuineness of smiles, J. Nonverbal Behav. 29 (1) (2005) 3–24.
[29] M. Lewis, I. Granic, Emotion, Development, and Self-Organization: Dynamic Systems Approaches to Emotional Development, Cambridge University Press, 2002.
[30] M. Csikszentmihalyi, Flow: The Psychology of Optimal Experience, Harper Perennial, 1991.
[31] E. Cambria, P. Gastaldo, F. Bisio, R. Zunino, An ELM-based model for affective analogical reasoning, Neurocomputing Special Issue on Extreme Learning Machines.
[32] E. Cambria, G.-B. Huang, Extreme learning machines, IEEE Intell. Syst. 28 (6) (2013) 2–31.
[33] S. Ridella, S. Rovetta, R. Zunino, Circular backpropagation networks for classification, IEEE Trans. Neural Networks 8 (1) (1997) 84–97.
[34] T.P. Vogl, J. Mangis, A. Rigler, W. Zink, D. Alkon, Accelerating the convergence of the back-propagation method, Biol. Cybernet. 59 (4-5) (1988) 257–263.
[35] G.-B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybernet. 2 (2) (2011) 107–122.
[36] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Networks 17 (4) (2006) 879–892.
[37] G.-B. Huang, An insight into extreme learning machines: random neurons, random features and kernels, Cogn. Comput., doi: http://dx.doi.org/10.1007/s12559-014-9255-2.
[38] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 42 (2) (2012) 513–529.
[39] E. Cambria, Y. Xia, A. Hussain, Affective common sense knowledge acquisition for sentiment analysis, in: LREC, Istanbul, 2012, pp. 3580–3585.
[40] M. Joshi, C. Rose, Generalizing dependency features for opinion mining, in: ACL/IJCNLP, Singapore, 2009.
[41] B. Heerschop, F. Goossen, A. Hogenboom, F. Frasincar, U. Kaymak, F. de Jong, Polarity analysis of texts using discourse structure, in: CIKM, Glasgow, 2011.
[42] F. Wang, Y. Wu, L. Qiu, Exploiting discourse relations for sentiment analysis, in: COLING, Mumbai, 2012, pp. 1311–1320.
[43] B. Chardon, F. Benamara, Y. Mathieu, V. Popescu, N. Asher, Measuring the effect of discourse structure on sentiment analysis, Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 7817, Springer, 2013, pp. 25–37.
[44] E. Coppock, D. Beaver, Principles of the exclusive muddle, J. Semantics doi:10.1093/jos/fft007.
[45] G. Winterstein, Layered meanings and Bayesian argumentation. the case of exclusives, submitted for publication.
[46] L. Karttunen, Presuppositions of compound sentences, Linguistic Inquiry 4 (2) (1973) 169–193.
[47] D. Beaver, Presupposition and Assertion in Dynamic Semantics, CSLI Publications, 2008.
[48] C. Potts, The Logic of Conventional Implicatures, Oxford University Press, Oxford, 2005.
[49] M. Simons, J. Tonhauser, D. Beaver, C. Roberts, What projects and why, in: Semantics and Linguistic Theory (SALT) 20, 2010, pp. 309–327.
[50] N. Asher, A. Lascarides, Logics of Conversation, Cambridge University Press, Cambridge, 2003.
[51] S. Repp, Negation in Gapping, Oxford University Press, Oxford, 2009.
[52] L. Vicente, On the syntax of adversative coordination, Nat. Lang. Linguistic Theory 28 (2) (2010) 381–415.
[53] J. Anscombre, O. Ducrot, Deux mais en français, Lingua 43 (1977) 23–40.
[54] G. Winterstein, What but-sentences argue for: a modern argumentative analysis of but, Lingua 122 (15) (2012) 1864–1885.
[55] J. Jayez, G. Winterstein, Additivity and probability, Lingua 132 (2013) 85–102.
[56] B. Pang, L. Lee, Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, in: ACL, Ann Arbor, 2005, pp. 115–124.
[57] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank.
[58] J. Blitzer, M. Dredze, F. Pereira, Biographies, Bollywood, boom-boxes and blenders: domain adaptation for sentiment classification, in: ACL, Prague, 2007, pp. 440–447.
[59] R. Socher, B. Huval, C.D. Manning, A.Y. Ng, Semantic compositionality through recursive matrix-vector spaces, in: EMNLP/CoNLL, Jeju, 2012, pp. 1201–1211.
[60] Z. Lin, T. Hwee, M.-Y. Kan, A PDTB-styled end-to-end discourse parser, Natural Language Engineering.