Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 3829–3839
Marseille, 20-25 June 2022

SenticNet 7: A Commonsense-based Neurosymbolic AI Framework for Explainable Sentiment Analysis

Erik Cambria,[1] Qian Liu,[1] Sergio Decherchi,[2] Frank Xing,[3] Kenneth Kwok[4]
[1] School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore
[2] Fondazione Istituto Italiano di Tecnologia (IIT), Italy
[3] School of Computing, National University of Singapore (NUS), Singapore
[4] Agency for Science, Technology and Research (A*STAR), Singapore
{cambria,liu.qian}@ntu.edu.sg, sergio.decherchi@iit.it, xing@nus.edu.sg, kenkwok@ihpc.a-star.edu.sg

Abstract

In recent years, AI research has demonstrated enormous potential for the benefit of humanity and society. While often better than its human counterparts in classification and pattern recognition tasks, however, AI still struggles with complex tasks that require commonsense reasoning such as natural language understanding. In this context, the key limitations of current AI models are: dependency, reproducibility, trustworthiness, interpretability, and explainability. In this work, we propose a commonsense-based neurosymbolic framework that aims to overcome these issues in the context of sentiment analysis. In particular, we employ unsupervised and reproducible subsymbolic techniques such as auto-regressive language models and kernel methods to build trustworthy symbolic representations that convert natural language to a sort of protolanguage and, hence, extract polarity from text in a completely interpretable and explainable manner.

Keywords: Neurosymbolic AI, sentiment analysis, natural language processing

1. Introduction

In 2012, a large-scale, international public effort was launched with the goal of reconstructing the full record of neural activity across complete neural circuits (Alivisatos et al., 2012). Ten years later, we still do not understand much about how the human brain works: we know well its hardware, but we are still pretty clueless about its operating system. Much earlier, the field of AI was born as an attempt to emulate human intelligence into machines. Today, however, most of AI research has regressed to the mimicking of intelligent behavior. Rather than 'artificial', such research should probably go under the name of 'pareidoliac' intelligence, as it tends to develop expert systems while claiming that these embed some sort of intelligence. Intelligence, however, is a 'suitcase' word that encapsulates several cognitive processes such as learning, attention, memory, decision making, emotions, and many more we may not even know about.

While recent developments have generated AI models that achieve human-like performance in many classification and pattern recognition tasks, AI still struggles with complex tasks that require more than just encoding joint probabilities or co-occurrence frequencies. Moreover, even the finest AI models are vulnerable to adversarial examples (Goodfellow et al., 2015).

The five key problems with AI today are: dependency, reproducibility, trustworthiness, interpretability, and explainability. In this work, we propose a commonsense-based neurosymbolic AI framework that aims to overcome these issues in the context of sentiment analysis. The framework is neurosymbolic because it leverages both subsymbolic and symbolic AI to perform polarity detection from text. In particular, subsymbolic techniques such as auto-regressive language models and kernel methods are used to build a symbolic representation, i.e., a hierarchical commonsense knowledge graph (Fig. 1), which is later used in concomitance with linguistic patterns to extract polarity from natural language text. As a result, the proposed framework is:

- unsupervised, because it does not require training on labeled data and it is domain-independent;
- reproducible, because each reasoning step can be explicitly recorded and replicated through each iteration;
- interpretable, because the process that generalizes input words and multiword expressions into their corresponding primitives is fully transparent;
- trustworthy, because classification outputs (positive or negative) come with a confidence score;
- explainable, because classification outputs are explicitly linked to emotions and the input concepts that convey these.

For example, a sentence like "Roberta murdered Elmo" would be categorized by most statistical sentiment analysis models as negative simply because the word 'murdered' is usually contained in negative sentences. SenticNet 7, instead, recognizes 'murdered' as the level-3 primitive `MURDER`, which is defined as `KILL(PERSON)`. The level-2 primitive `KILL`, in turn, is defined as `DEACTIVATE(LIFE)`, while `DEACTIVATE` is a level-1 primitive that is defined as `TERMINATE(PROCESS)`.

Finally, `TERMINATE(`x`)` is a level-0 primitive (or superprimitive), which is defined as the transition from a state of existence to a state of inexistence ($\exists x \to \nexists x$). Such a transition is characterized by the emotions 'fear' and 'anger' (towards the agent) and 'sadness' (towards the experiencer), which correspond to negative polarity values according to the Hourglass model (Susanto et al., 2020). In other words, the input sentence is 'translated' from natural language into a sort of 'protolanguage' sentence "Roberta `TERMINATE(`Elmo`.LIFE.PROCESS)`", which generalizes words and multiword expressions in terms of primitives and, hence, connects these (in a semantic-role-labeling fashion) to their corresponding emotion and polarity labels. Here, we list the processing steps for this sentence:

– Roberta murdered Elmo
– Roberta `MURDER` Elmo
– Roberta `KILL(PERSON=`Elmo`)`
– Roberta `DEACTIVATE(`Elmo`.LIFE)`
– Roberta `TERMINATE(`Elmo`.LIFE.PROCESS)`
– Roberta ⇒ ∄Elmo`.LIFE.PROCESS`
– Roberta ⇒ `fear+anger` ∧ Elmo ⇒ `sadness`
– Roberta ⇒ `NEGATIVE` ∧ Elmo ⇒ `NEGATIVE`

While these polarity values are hashed into upper-level primitives (`MURDER:=NEGATIVE`) for fast processing, the reasoning behind such hashing can always be unfolded (as shown in the example above) in order to ensure the full interpretability and explainability of classification results. Additionally, such results are associated with a confidence score between 0 and 100% calculated by means of sentic paths (explained later).

The remainder of the paper is organized as follows: the next section briefly discusses related works at the crossroads of neurosymbolic AI and sentiment analysis; later, we describe in detail the framework structure, including explaining how primitives are discovered, named, and refined; the following section presents experimental results on 10 different datasets; finally, we provide concluding remarks.

2. Related Work

Neurosymbolic AI is a new kind of 'hybrid' AI that aims to leverage the strengths of both recent subsymbolic AI techniques, e.g., deep neural networks, and good old-fashioned symbolic AI, e.g., knowledge graphs. In the last couple of years, AI researchers have started investigating how neurosymbolic AI can be used for natural language processing (NLP) and natural language understanding, computer vision and image understanding, speech recognition and machine translation (Wang et al., 2019; Mao et al., 2019; Krishnaswamy and Pustejovsky, 2020; d'Avila Garcez and Lamb, 2020; Sarker et al., 2021). In this work, we employ neurosymbolic AI for sentiment analysis, an NLP task that aims to identify, extract, quantify, and study affective states and subjective information from text. Recently, sentiment analysis systems have achieved remarkable accuracy thanks to the advancements of deep learning techniques. For example, (Barnes et al., 2021) proposed a unified approach to improve structured sentiment analysis which used dependency graph parsing to jointly predict all elements of an opinion tuple and their relations, instead of dividing the task into subtasks. (Li et al., 2021), instead, proposed to consider the complementarity of syntax structures and semantic correlations simultaneously using dual graph convolutional networks. (Yan et al., 2021) proposed to convert all subtasks in aspect-based sentiment analysis into a unified generative formulation, and redefined every subtask target as a sequence mixed by pointer indexes and sentiment class indexes. (Tang et al., 2021) designed a hierarchical multimodal fusion architecture to improve multimodal sentiment analysis, and employed coupled-translation fusion networks to model bi-direction interplay via couple learning, ensuring the robustness with respect to missing modalities.

Despite achieving increasingly higher accuracy, mainstream approaches are still based on black-box models that do not provide any insights about their internal reasoning process. Despite this state of affairs, over the last few years some works have started trying to make sentiment analysis systems more explainable without affecting performance (Gunning and Aha, 2019; Arrieta et al., 2020). For example, (Bodria et al., 2020) explored attention-based techniques to extract meaningful sentiment scores and, hence, to shed light on the internal behavior of deep neural networks. (Yang et al., 2021) proposed to automatically generate counterfactual augmented data for enhancing the robustness of sentiment analysis models. Finally, (Bacco et al., 2021) employed a hierarchical transformer architecture on movie reviews to generate extractive summaries that serve as an explanation for the decisions taken by the system.

3. Framework Structure

In the next four sections, we describe in detail how SenticNet 7 is built. Firstly, primitive sets (groups of concepts with similar meaning) are discovered by means of lexical substitution. Secondly, these semantically-related sets are refined in terms of affective similarity. Next, each primitive set is named after its most representative term and paired with its semantic opposite (e.g., `ACCEPT` versus `REJECT`). Finally, primitive sets are further refined by studying the multidimensional path between each antithetic primitive pair.

3.1. Primitive Set Discovery

One of the main reasons why conceptual dependency theory (Schank, 1972) and many other decompositional methods for conceptualization (Minsky, 1975; Jackendoff, 1976; Rumelhart and Ortony, 1977; Wierzbicka, 1996) were abandoned in favor of subsymbolic techniques was the amount of time and effort required to come up with a comprehensive set of rules.

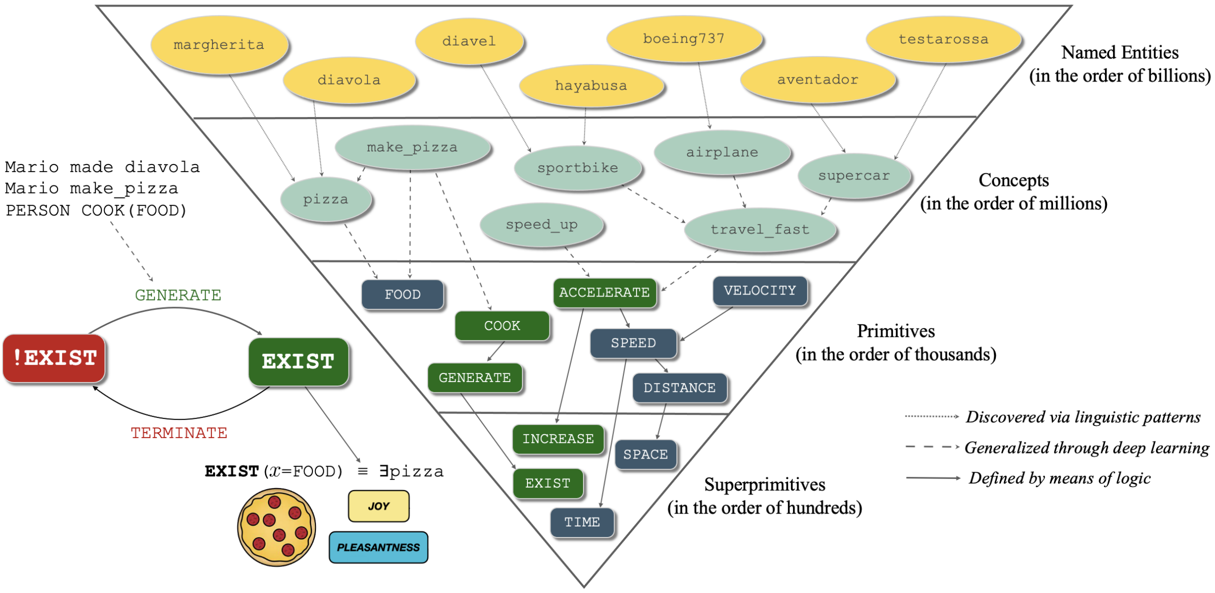

Figure 1: SenticNet 7's dependency graph structure.

In this work, we take inspiration from the field of semiotics (Peirce, 1902; Eco, 1984; Greimas, 1987) to perform symbol grounding in the context of sentiment analysis. In particular, we leverage the representation learning power of XLNet (Yang et al., 2019) to automatically discover primitive sets for affective reasoning. The aim is to get away from associating polarity to a static list of affect words by letting SenticNet 7 figure out such polarity on the fly based on the building blocks of meaning. Thus, given a target primitive like `ACCEPT`, for example, the goal is to find its synonym ring, i.e., words like `welcome`, `agree`, and `embrace` that are semantically equivalent. Firstly, we use word2vec's negative sampling objective function (Mikolov et al., 2013) to learn the appropriate representation of sentential context and target primitive. Here, a positive pair is described as a valid context and primitive pair and the negative pairs are created by sampling random words from a unigram distribution. Our aim is to maximize the following objective function:

$$Obj = \sum_{\mathbf{p},\mathbf{c}} (log(\sigma(\mathbf{p}.\mathbf{c})) + \sum_{i=1}^{z} log(\sigma(-\mathbf{p}_i.\mathbf{c}))), \quad (1)$$

where $\mathbf{p}$ is the target primitive and $\mathbf{c}$ is the sentential context representation. The overall objective is calculated across all the valid primitive and context pairs. We choose z invalid primitive-context pairs where each $-\mathbf{p}_i$ refers to an invalid primitive with respect to a context. Next, we use the XLNet architecture in order to obtain the sentential context embedding of a primitive. XLNet is a large bidirectional transformer that uses an improved training methodology, larger data and more computational power to achieve better prediction metrics on many NLP tasks. To improve the training, XLNet introduces permutation language modeling, where all tokens are predicted but in random order. This is in contrast to BERT's masked language model where only the masked tokens are predicted (Devlin et al., 2019). This is also in contrast to the traditional language models, where all tokens were predicted in sequential order instead of random order. This helps the model to learn bidirectional relationships and therefore better handles dependencies and relations between words. In particular, we utilize XLNet as it follows:

- First, we fine-tune the pre-trained XLNet network on the ukWaC corpus (Baroni et al., 2009).

- Next, we calculate the embedding for the context $\mathbf{c}$. For this, we first remove the target primitive $\mathbf{p}$, i.e., either the verb or noun from the sentence. The remainder of the sentence is then fed to the XLNet architecture which returns the context embedding.

- Finally, we adopt a new similarity measure in order to find the replacement of the primitive. For this, we need the embedding of the target primitive which we obtain by simply feeding the word to XLNet pre-trained network. Given a target primitive $\mathbf{p}$ and its sentential context $\mathbf{c}$, we calculate the cosine distance of all the other words in the embedding hyperspace with both $\mathbf{p}$ and $\mathbf{c}$. If $\mathbf{w}$ is a candidate word, the distance is then calculated as:

$$\begin{aligned} dist(\mathbf{w}, (\mathbf{p}, \mathbf{c})) = cos(\mathbf{w}, \mathbf{p}) \cdot cos(\mathbf{w}, \mathbf{c}) \cdot \\ cos(\text{XLNet}(\mathbf{c}, \mathbf{w}), \text{XLNet}(\mathbf{c}, \mathbf{p})), \end{aligned} \quad (2)$$

where XLNet$(\mathbf{c}, \mathbf{w})$ is the XLNet-produced embedding of the sentence formed by replacing primitive $\mathbf{p}$ with the candidate word $\mathbf{w}$ in the sentence. Similarly, XLNet$(\mathbf{c}, \mathbf{p})$ is the embedding of the original sentence which contains the primitive $\mathbf{p}$. We rank candidates as per their cosine distance and generate the list of possible lexical substitutes.

We apply the algorithm described above (code available on our github[1]) on ConceptNet 5.5 (Speer et al., 2017), a semantic network of commonsense knowledge developed through crowdsourcing. First, we extract all the verb-noun and adjective-noun concepts together with a sample sentence for each concept. Then, we select one word from the concept to be the target word while the remaining sentence serves as the context. Next, we obtain the context and target word embeddings (**c** and **p**) from the joint hyperspace of the network to find a substitute for the target word having the same part of speech in the given context. For all possible substitute words **w**, finally, we calculate their cosine similarity by Equation (2) and rank them using this metric for possible substitutes. This substitution leads to new verb-noun and adjective-noun pairs which bear the same conceptual meaning in the given context.

3.2. Affective Similarity Prediction

The lexical substitution algorithm described in the previous section allows for discovering synonym rings that group together concepts sharing similar meaning. Lexical substitution, however, tends to ignore affective differences between concepts. For example, verbs like `accept` and `reject` can be found in similar lexical contexts, e.g., "he accepted the job" versus "he rejected the job" or "she accepts the offer" versus "she rejects the offer", although they bear opposite polarity orientation. To solve this issue, we propose an affective similarity prediction algorithm that calculates the affective relatedness of concepts based on their links with 24 key emotion nodes (Susanto et al., 2020) in the graph representation of ConceptNet.

Such algorithm, inspired by another work of ours (Qiu et al., 2022), consists of two basic steps: 1) define an affective similarity index that assigns a score known as affective similarity score $Score_{x,y}$ for every pair of nodes (c_x, c_y); 2) rank the pairs of nodes in decreasing order based on their score and select links at the top of the ranking as the prediction results. The definition of the similarity index is the key to similarity-based link prediction. A well-defined similarity index can capture the link formation process, and then achieve high prediction accuracy, and vice versa. Table 1 lists a set of well-known similarity indexes along with a brief description and the mathematical definition of $Score_{x,y}$.

We define our semantic network as $G = (V, E)$, where $V = \{v_1, ..., v_n\}$ is the node set and n is the number of nodes, and $E = \{e_{i,j}\}$ is the link set. The adjacency matrix is denoted as $\mathbf{A} \in \mathbb{R}^{n\times n}$, where $\mathbf{a}_{i,j} \in [0, 1]$ is the link weight between nodes v_i and v_j. The value of $\mathbf{a}_{i,j}$ represents the connection strength between nodes v_i and v_j. If there is no link between v_i and v_j, then $\mathbf{a}_{i,j} = 0$. We define $\mathbf{U} \in \mathbb{R}^{n\times d}$ as the embedding matrix of the network G, where the i-th row of $\mathbf{U}$, $\mathbf{U}_{i,*}$, is the embedding vector of v_i and d is the embedding dimension (d is a preset constant and $d \ll n$). Our task is to propose a network embedding method to learn the $\mathbf{U}$ of G while preserving the affective properties of G. Let $\mathbf{S} \in \mathbb{R}^{n\times n}$ denote the affective similarity matrix, where $\mathbf{s}_{i,j}$ is the affective similarity between v_i and v_j. The similarity index vector $S_v = \{s_1, ..., s_i, ..., s_\gamma\}$ is a vector consisting of γ single similarity indexes. For example, s_i could be the CN index. For any node pair $\{v_i, v_j\}$, the similarity index vector is $S_v(v_i, v_j) = \{s_1(v_i, v_j), ..., s_\gamma(v_i, v_j)\}$. The index weight vector $\phi = \{\varphi_1, ..., \varphi_i, ..., \varphi_\gamma\}$ is the weight vector of S_v and φ_i denotes the weight of index s_i in S_v. Given S_v and ϕ, the comprehensive similarity index $S(v_i, v_j)$ between nodes v_i and v_j is defined as:

$$S(v_i, v_j) = \sum_{s_k \in S_v} \varphi_k \frac{s_k(v_i, v_j) - min(s_k)}{max(s_k) - min(s_k)}, \qquad (3)$$

where $min(s_k)$ is the minimum s_k value of all node pairs in the network, and $max(s_k)$ is the maximum value. To make the comprehensive similarity index S more consistent, we need to determine the optimal weight value of each single similarity index s_i in S_v. A ϕ whose corresponding S achieves the most accurate node similarity evaluation is defined as the optimal index weight, ϕ^*. To search for the optimal ϕ^*, we adopt the AUC (Hu et al., 2017) as the metric of prediction accuracy. Let $AUC(\phi)$ denote the prediction accuracy of the S corresponding to the ϕ.

The problem turns into searching for the ϕ^* such that the corresponding $AUC(\phi^*)$ is maximized: clearly, this is a random search problem. The Quantum-behaved Particle Swarm Optimization (QPSO) method (Tang et al., 2014) proved to be effective in random search problems, hence we use it to search the ϕ^* and construct the affective similarity matrix $\mathbf{S}$. In QPSO, the two polarization states of a qubit (the basic information storage unit) are $|0\rangle$ and $|1\rangle$. A qubit state is denoted as $P_{i,c}|0\rangle + P_{i,s}|1\rangle$, where $P_{i,c}$ and $P_{i,s}$ is the probability amplitudes of $|0\rangle$ and $|1\rangle$. The three steps of affective similarity prediction are as follows:

(1) Produce the initial quantum particle swarm. The coding method of each quantum particle in the swarm is

$$P_i = \left[\begin{vmatrix} cos(\theta_{i,1}) \\ sin(\theta_{i,1}) \end{vmatrix}, \begin{vmatrix} cos(\theta_{i,2}) \\ sin(\theta_{i,2}) \end{vmatrix}, ..., \begin{vmatrix} cos(\theta_{i,\gamma}) \\ sin(\theta_{i,\gamma}) \end{vmatrix}\right], \qquad (4)$$

where $\theta_{i,j} = 2\pi \times rnd$, rnd is a random number between 0 and 1, $i = 1, 2, ..., m$, and $j = 1, 2, ..., \gamma$. Here, m is the number of particles in the quantum particle swarm. γ is the size of S_v. Each quantum particle corresponds to two probability amplitudes $P_{i,s}$ and $P_{i,c}$:

$$\begin{array}{c} P_{i,s} = [sin(\theta_{i,1}), sin(\theta_{i,2}), ..., sin(\theta_{i,\gamma})] \\ P_{i,c} = [cos(\theta_{i,1}), cos(\theta_{i,2}), ..., cos(\theta_{i,\gamma})] \end{array}. \qquad (5)$$

[1] `https://github.com/senticnet`

Index	Definition	Description
CN	$\lvert\Gamma(u) \cap \Gamma(v)\rvert$	$\Gamma(u)$ denotes the neighbors set of node u, CN calculates the intersection set size of joint neighbors between nodes u and v (Liben-Nowell and Kleinberg, 2007).
AA	$\sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{log \Gamma \lvert z \rvert}$	The index measures the similarity between two nodes based on their shared neighbors. Each neighbor's weight is logarithmically penalized by its frequency (Adamic and Adar, 2003).
PA	$\lvert\Gamma(u)\rvert\lvert\Gamma(v)\rvert$	The index based on the observation that the probability of link formation between two nodes increases as the degree of these nodes dose (Barabási and Albert, 1999).
JA	$\frac{\lvert\Gamma(u) \cap \Gamma(v)\rvert}{\lvert\Gamma(u) \cup \Gamma(v)\rvert}$	The index measures the ratio of shared neighbors in the complete set of neighbors for two nodes (Martínez et al., 2017).
Kate	$\sum_{k=1}^{\infty} \beta^k (A^k)_{u,v}$	The A is the adjacency matrix and $0 < \beta < 1$. The index sums the influence of all possible paths between two pairs of nodes, incrementally penalizing paths by their length (Katz, 1953).
GLHN	$I_{u,v} + \sum_{k=1}^{\infty} \beta^k (A^k)_{u,v}$	The I is a identity matrix term, which indicates maximal self-similarity (Leicht et al., 2006).

Table 1: Popular similarity indexes

For each quantum particle, $P_{i,s}$ and $P_{i,c}$ can be transformed into index weight arrays $\phi_{i,s}$ and $\phi_{i,c}$. The ϕ can be $\phi_{i,s}$ or $\phi_{i,c}$.

$$\begin{aligned} \phi_{i,s} &= \left[\frac{sin(\theta_{i,1})}{\sum\limits_{\eta=1}^{\gamma} sin(\theta_{i,\eta})}, ..., \frac{sin(\theta_{i,\gamma})}{\sum\limits_{\eta=1}^{\gamma} sin(\theta_{i,\eta})} \right], \\ \phi_{i,c} &= \left[\frac{cos(\theta_{i,1})}{\sum\limits_{\eta=1}^{\gamma} cos(\theta_{i,\eta})}, ..., \frac{cos(\theta_{i,\gamma})}{\sum\limits_{\eta=1}^{\gamma} cos(\theta_{i,\eta})} \right]. \end{aligned} \tag{6}$$

(2) Weight array update. We update ϕ iteratively. Let $\phi_{i,l}$ denotes the index weight array for which $AUC(\phi_{i,l})$ is maximized during the current search for particle i, and $P_{i,l} = [cos(\theta_{i,l,1}), ..., cos(\theta_{i,l,\gamma})]$ (we assume the optimal sites are cosine sites) be the probability amplitude for $\phi_{i,l}$. Let ϕ^* denote the index weight vector for which $AUC(\phi^*)$ is maximized for the entire search process, and $P_g = [cos(\theta_{g,1}), ..., cos(\theta_{g,\gamma})$ be the probability amplitude for ϕ^*. The new value of ϕ can be obtained by updating $P_{i,s}$ and $P_{i,c}$. In each iteration, $\bar{P}_{i,s}$ and $\bar{P}_{i,c}$ are obtained by the following equations. Then $P_{i,s} = \bar{P}_{i,s}$, $P_{i,c} = \bar{P}_{i,c}$.

$$\begin{aligned} \bar{P}_{i,s} &= [sin(\theta_{i,1}(t) + \triangle\theta_{i,1}(t+1)), \\ &\quad ..., sin(\theta_{i,\gamma}(t) + \triangle\theta_{i,\gamma}(t+1))], \\ \bar{P}_{i,c} &= [cos(\theta_{i,1}(t) + \triangle\theta_{i,1}(t+1)), \\ &\quad ..., cos(\theta_{i,\gamma}(t) + \triangle\theta_{i,1}(t+1))], \end{aligned} \tag{7}$$

where $\triangle\theta_{i,j}(t+1) = \triangle\theta_{i,j} + c_1 r_1 \triangle\theta_l + c_2 r_2 \triangle\theta_g$, c_1 and c_2 are scaling parameters, r_1 and r_2 are uniform random number between 0 and 1, and $\triangle\theta_{i,j}(0) = 0$. The value of $\triangle\theta_l$ and $\triangle\theta_g$ are determined by the following equation:

$$\begin{aligned} \triangle\theta_l &= \begin{Bmatrix} 2\pi + \theta_{i,l,j} + \theta_{i,j}(\theta_{i,l,j} - \theta_{i,j} < -\pi) \\ \theta_{i,l,j} - \theta_{i,j}(-\pi \leq \theta_{i,l,j} - \theta_{i,j} \leq \pi) \\ \theta_{i,l,j} - \theta_{i,j} - 2\pi(\theta_{i,l,j} - \theta_{i,j} > \pi) \end{Bmatrix} \\ \triangle\theta_g &= \begin{Bmatrix} 2\pi + \theta_{g,j} + \theta_{i,j}(\theta_{g,j} - \theta_{i,j} < -\pi) \\ \theta_{g,j} - \theta_{i,j}(-\pi \leq \theta_{g,j} - \theta_{i,j} \leq \pi) \\ \theta_{g,j} - \theta_{i,j} - 2\pi(\theta_{g,j} - \theta_{i,j} > \pi) \end{Bmatrix}. \end{aligned} \tag{8}$$

(3) After the iterative search, we can obtain the optimal ϕ^*. Lastly, we calculate the affective similarity between each pair of nodes (v_i, v_j) and construct **S** by S. Summarizing, said g_{max} be the number of iterations, the process of affective similarity prediction is detailed in Algorithm 1.

Algorithm 1 Affective Similarity Prediction

```
Input: Semantic network $G$, similarity index vector $S_v$.
Output: Affective similarity matrix $\mathbf{S}$.
Initialize the index weight vector $\phi$ with the indexes listed in Table 1;
Generate $m$ quantum state particles by Equation (4);
for $r = 1$ to $g_{max}$ do
    for $i = 1$ to $m$ do
        Transform $P_{i,s}$ and $P_{i,c}$ of each particle into index weigh vector $\phi_{i,s}$ and $\phi_{i,c}$ by Equation (6);
        if $AUC(\phi_{i,c}) > AUC(\phi_{i,l})$ then $P_{i,l} = P_{i,c}$ end
        if $AUC(\phi_{i,s}) > AUC(\phi_{i,l})$ then $P_{i,l} = P_{i,s}$ end
        if $AUC(\phi_{i,l}) > AUC(\phi_g)$ then $P_g = P_{i,l}$ end
        Update $P_{i,s}$ and $P_{i,c}$ by Equation (7)
    end for
end for
Convert $P_g$ to $\phi^*$ by Equation (6);
for each node pair $(v_i, v_j) \in G$ do
    Calculate affective similarity $\mathbf{s}_{i,j}$ by Equation (3);
end for
return $\mathbf{S}$.
```

3.3. Primitive Definition and Pairing

A recent big shift in NLP research has been the upgrade from the bag-of-words (BOW) model to the continuous-bag-of-words (CBOW) model, which allowed NLP systems to take into account context in the same way one can tell what is the role of a pixel in an image based on its neighbors (Cambria and White, 2014). This same shift, however, is what had slowly turned NLP systems into black-box systems (Adadi and Berrada, 2018). Since they are better than CBOW at preserving meaning, multiword expressions are a possible solution to reverse this trend. Nevertheless, multiword expressions are hard to discover and can cause the size of a lexicon to increase exponentially (Rajagopal et al., 2013; Xing et al., 2019).

Instead of assigning polarity to millions of multiword expressions, SenticNet 7 allows polarity to be inferred on the fly by combining verb primitives (e.g., `SUPPORT` and its semantic opposite `OBSTRUCT`) and noun primitives (e.g., `FRIEND` and its semantic opposite `ENEMY`), so that expressions like `help_buddy`, `assist_pal`, or `stand_up_for_homeboy` are all generalized as `SUPPORT(FRIEND)` and, thus, categorized as positive.

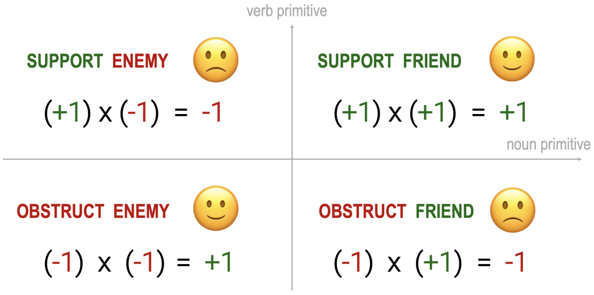

Figure 2: An example of sentic algebra.

Besides reducing lexicon size and processing time, this approach also ensures higher accuracy as compared to many statistical approaches that simply classify text based on word occurrence frequencies. For example, a BOW model would classify expressions like stand_in_the_way_of_foe, slow_down_rival or stall_adversary as negative because of the statistically negative words that compose them. In our framework, instead, such expressions are all generalized as OBSTRUCT(ENEMY) and thus correctly classified as positive (Fig. 2). This way, SenticNet 7 reduces the symbol grounding problem and, hence, gets one step closer to natural language understanding.

After discovering primitive sets by means of XLNet and splitting each set into positive subset and negative subset by means of affective similarity prediction, we assign a label to each subset by selecting the most typical of the terms. In the positive subset {add, soar, increase, escalate, mount_up, ...}, for example, the term with the highest occurrence frequency is increase. Hence, the subset is named after it, i.e., INCREASE, and later defined manually using logic, i.e., INCREASE(x):= x + 1. Likewise, the corresponding negative subset is termed DECREASE and defined as DECREASE(x):= x - 1. Primitives like INCREASE and DECREASE are Level-0 primitives (or superprimitives) because they are 'grounded' using logic. Primitives defined in terms of these, e.g., GROW:= INCREASE(SIZE), are Level-1 primitives. Primitives defined in terms of Level-1 primitives, e.g., LENGTHEN:= GROW(LENGTH), are Level-2 primitives and so on (Fig. 3).

3.4. Sentic Paths

Lexical substitution and affective similarity prediction enable the discovery of primitive sets that are both semantically and affectively related. However, they do not ensure that the intersection between different sets in null, i.e., they sometimes generate overlapping primitive clusters which may share some words and multiword expressions. In order to force the mutual exclusiveness of primitive sets, we introduce sentic paths, a cognitive-inspired algorithm that takes into account the topology of affective data in a multidimensional vector space of commonsense knowledge.

Sentic paths are an affective version of the principal path method (Ferrarotti et al., 2019), a kernel method conceived to find smooth paths between objects in space through a number of waypoints (N_c). The main feature of the method is that the obtained path aims to move through high probability regions of the space, searching for a geodetic whose underlying topology is ruled by the samples probability. This method aspires to mimic the cognitive intuition for which thinking is the process of moving from one concept to another through regions of the space where there is a high probability of finding other concepts (Ragusa et al., 2019).

In particular, in this work we take advantage of a recently refined version of the algorithm (Gardini et al., 2021) and we employ the plain feature space (linear kernel, primal problem). Rather than a distance, sentic paths calculate a discrete path between a primitive concept p_0 and its semantic opposite p_{N_c+1} throughout the vector space manifolds. While the shortest path (through the pure Euclidean distance) between two antithetic primitives risks to include many irrelevant concepts, a path that follows the topological structure of the vector space from a positive primitive (e.g., p_0=ACCEPT) to its semantic antithesis (e.g., p_{N_c+1}=REJECT) is more likely to contain concepts that are both semantically and affectively relevant.

Because positive and negative concepts are found in diametrically opposite zones of the space (Cambria et al., 2015), sentic paths always traverse the vector space from one end to the other (Fig. 4). This ensures the discovery of concepts that are both semantically and affectively related to both the positive primitive p_0 (e.g., welcome, agree, and take_in) and the negative one p_{N_c+1} (e.g., refuse, turn_down, and deny).

To adapt the algorithm to the context of sentiment analysis, we employ a metric based on the Hourglass model (Susanto et al., 2020), a biologically-inspired and psychologically-motivated emotion categorization model based on four independent but concomitant affective dimensions. The core steps of the algorithm can be summarized as it follows:

1. *Sentic path initialization:* given the starting and the ending primitives p_0 and p_{N_c+1}, the Dijkstra algorithm is run over a penalized graph obtained by computing the penalized distance matrix among all the concepts c_i in $\mathbf{C}$ as follows:

$$d_p^2(c_i, c_j) = \begin{cases} d^2(c_i, c_j), & c_i \in \text{nn}_k(c_j) \\ td^2(c_i, c_j), & \text{otherwise} \end{cases}$$

where $\text{nn}_k(c_j)$ is the nearest neighbors set and t is a penalization factor. This approach allows to capture the manifold and avoid shortcuts.

2. *Waypoint concept positioning:* the Dijkstra algorithm is run on the penalized distance matrix and some intermediate concepts are returned. This path is then reparameterized to obtained equally distanced points.

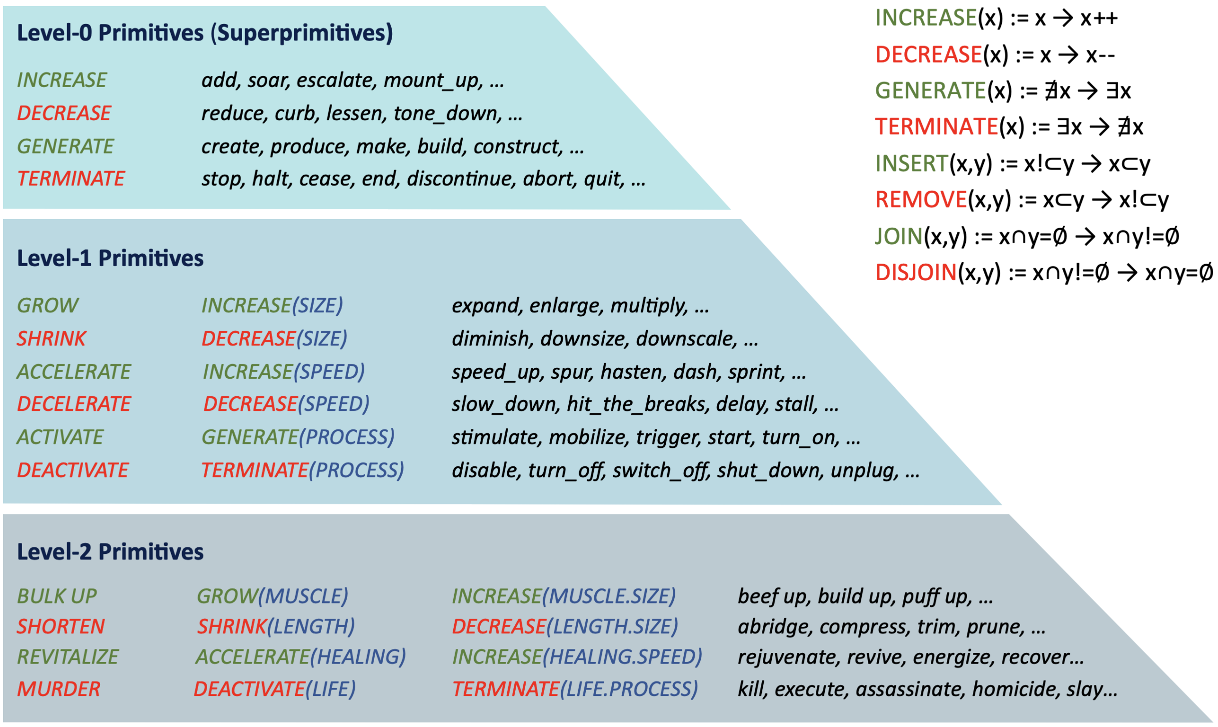
Level-0 Primitives (Superprimitives)
INCREASE add, soar, escalate, mount_up, …
DECREASE reduce, curb, lessen, tone_down, …
GENERATE create, produce, make, build, construct, …
TERMINATE stop, halt, cease, end, discontinue, abort, quit, …
INCREASE(x) := x → x++
DECREASE(x) := x → x--
GENERATE(x) := ∄x → ∃x
TERMINATE(x) := ∃x → ∄x
INSERT(x,y) := x!⊂y → x⊂y
REMOVE(x,y) := x⊂y → x!⊂y
JOIN(x,y) := x∩y=∅ → x∩y!=∅
DISJOIN(x,y) := x∩y!=∅ → x∩y=∅
Level-1 Primitives
GROW INCREASE(SIZE) expand, enlarge, multiply, …
SHRINK DECREASE(SIZE) diminish, downsize, downscale, …
ACCELERATE INCREASE(SPEED) speed_up, spur, hasten, dash, sprint, …
DECELERATE DECREASE(SPEED) slow_down, hit_the_breaks, delay, stall, …
ACTIVATE GENERATE(PROCESS) stimulate, mobilize, trigger, start, turn_on, …
DEACTIVATE TERMINATE(PROCESS) disable, turn_off, switch_off, shut_down, unplug, …
Level-2 Primitives
BULK UP GROW(MUSCLE) INCREASE(MUSCLE.SIZE) beef up, build up, puff up, …
SHORTEN SHRINK(LENGTH) DECREASE(LENGTH.SIZE) abridge, compress, trim, prune, …
REVITALIZE ACCELERATE(HEALING) INCREASE(HEALING.SPEED) rejuvenate, revive, energize, recover…
MURDER DEACTIVATE(LIFE) TERMINATE(LIFE.PROCESS) kill, execute, assassinate, homicide, slay…

Figure 3: Primitives hierarchy.

Lexicon	Year	CR	MR	Amazon	IMDb	Sanders	SST	STS	SE13	SE15	SE16
General Inquirer*	1966	56.56%	53.76%	59.63%	59.43%	46.81%	54.39%	54.59%	47.82%	51.12%	40.88%
LIWC	1993	52.42%	41.84%	57.33%	63.53%	52.45%	44.88%	67.68%	44.85%	43.96%	39.75%
ANEW	1999	51.55%	51.13%	50.66%	51.94%	50.08%	51.33%	47.35%	50.44%	48.26%	49.93%
WordNet-Affect*	2004	04.61%	05.05%	18.87%	28.99%	17.81%	4.812%	24.23%	15.92%	16.35%	10.54%
Opinion Lexicon	2004	72.98%	62.90%	65.76%	70.91%	67.89%	66.50%	74.09%	72.65%	64.01%	73.42%
Opinion Finder*	2005	62.05%	59.98%	59.48%	58.75%	51.22%	61.86%	60.72%	50.28%	53.57%	44.21%
Micro WNOp*	2007	20.39%	18.73%	44.48%	49.17%	22.95%	17.64%	28.13%	24.89%	26.58%	18.41%
Sentiment140	2009	65.50%	61.52%	66.64%	68.64%	70.92%	64.67%	76.88%	66.78%	60.94%	62.55%
SentiStrength*	2010	45.69%	41.72%	59.09%	60.18%	47.87%	41.57%	58.49%	42.32%	45.60%	35.46%
SentiWordNet	2010	64.60%	59.07%	62.36%	64.13%	61.68%	61.55%	63.23%	50.03%	60.53%	46.80%
AFINN	2011	70.59%	63.78%	66.63%	71.27%	**71.90%**	66.85%	78.27%	59.04%	**67.08%**	53.82%
SO-CAL	2011	65.58%	**64.58%**	**75.86%**	**78.67%**	52.78%	**67.33%**	63.51%	41.15%	37.63%	41.02%
EmoLex	2013	61.10%	56.03%	52.73%	51.94%	56.86%	59.06%	60.17%	66.21%	64.21%	66.40%
NOVAD*	2013	64.88%	56.91%	57.06%	56.81%	51.06%	58.88%	61.55%	61.10%	57.87%	58.16%
NRC HS Lexicon	2014	65.26%	58.53%	59.39%	63.49%	59.31%	61.58%	64.07%	70.45%	60.53%	72.72%
VADER	2014	**75.18%**	61.37%	67.03%	69.24%	71.81%	65.94%	**78.83%**	**74.88%**	**69.53%**	**74.05%**
MPQA	2015	68.20%	64.03%	62.43%	64.33%	61.03%	66.66%	71.03%	56.35%	58.28%	54.70%
SentiWords*	2016	62.71%	58.65%	58.11%	57.29%	53.59%	60.57%	60.44%	58.82%	57.46%	54.38%
HSSWE*	2017	71.33%	60.61%	67.08%	65.27%	**73.94%**	63.15%	78.27%	68.67%	64.83%	66.62%
Lingmotif-lex	2018	**76.08%**	**66.52%**	**73.34%**	**74.08%**	70.59%	**70.58%**	**79.11%**	**74.70%**	64.62%	**74.91%**
SenticNet 7	2022	**83.60%**	**77.04%**	**81.53%**	**82.91%**	**80.54%**	**78.71%**	**90.08%**	**83.69%**	**81.67%**	**84.39%**

Table 2: Comparison with 20 popular lexica on 10 benchmark datasets for sentiment analysis (top 3 results for each dataset are in bold). When available, we tested lexica using their own polarity detection framework. The lexica followed by a star sign (*), instead, were tested using a standard set of linguistic patterns plus microtext normalization. Data and code of the evaluation process are available upon request.

3. *Optimize the cost function*: the path is smoothed through a cost function optimized via the expectation-maximization (EM) algorithm. The waypoint concept configuration $\mathbf{P}_{init}$ from the previous step is used as waypoint concept initialization and as input matrix $\mathbf{C}$ ($\mathbf{P}_{init} = \mathbf{C}$). The cost function, hence, is:

$$\min_{P,u} \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} \|c_i - p_j\|^2 \delta(u_i, j) + s \sum_{i=0}^{N_c} \|p_{i+1} - p_i\|^2 \tag{9}$$

where $\delta(u_i, j)$ is a Kronecker delta to rule the waypoint membership and s is a regularization coefficient. Hence, the method is an out-of-sample smooth extension of Dijkstra shortest path, where the underlying graph is ruled by a penalized Euclidean metric and whose smoothness is ruled by s. Sentic paths are not only used to refine the generalization capability of the framework by reducing the overlap between primitive sets but also to calculate a confidence score for each concept, which is defined as the normalized distance between concept c and primitive p along the path.

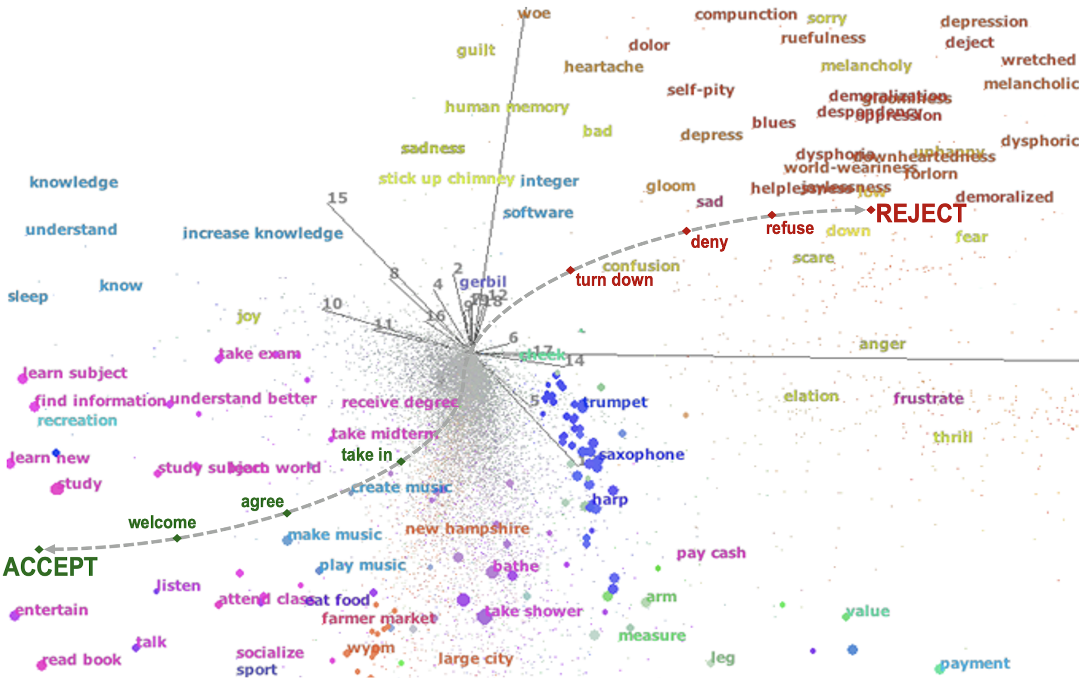

Figure 4: Sentic path between ACCEPT and REJECT.

4. Experiments

We evaluated SenticNet 7 (available both as a standalone XML repository[2] and as an API[3] in multiple languages) against 20 popular English lexica for sentiment analysis developed between 1966 and 2020, namely: General Inquirer (Stone et al., 1966), LIWC (Francis and Pennebaker, 1993), ANEW (Bradley and Lang, 1999), WordNet-Affect (Strapparava and Valitutti, 2004), Opinion Lexicon (Hu and Liu, 2004), Opinion Finder (Wilson et al., 2005), Micro WNOp (Cerini et al., 2007), Sentiment140 (Go et al., 2009), SentiStrength (Thelwall et al., 2010), SentiWordNet (Baccianella et al., 2010), AFINN (Nielsen, 2011), SO-CAL (Taboada et al., 2011), EmoLex (Mohammad and Turney, 2013), NOVAD (Warriner et al., 2013), NRC HS Lexicon (Zhu et al., 2014), VADER (Hutto and Gilbert, 2014), MPQA (Deng and Wiebe, 2015), SentiWords (Gatti et al., 2016), HSSWE (Wang and Xia, 2017), and Lingmotif-lex (Moreno-Ortiz et al., 2018). We tested these lexica on 10 well-known sentiment analysis datasets, namely: CR (Hu and Liu, 2004), MR (Pang and Lee, 2005), Amazon (Blitzer et al., 2007), IMDb (Maas et al., 2011), Sanders (Analytics, 2012), SST (Socher et al., 2013), STS (Saif et al., 2013), SE13 (Nakov et al., 2013), SE15 (Rosenthal et al., 2015), and SE16 (Nakov et al., 2016). We set the experiment as a binary classification problem (Table 2), i.e., we reduced the labels of all datasets and lexica to positive and negative (neutral entries were ignored).

SenticNet 7 was the best-performing of all 20 lexica, mostly because of its bigger size. Many of the classification errors made by other lexica, in fact, were due to missing entries. Beside single words, moreover, SenticNet 7 also contains multiword expressions which enable polarity disambiguation, e.g., dead vs dead_right, smart vs smart_ass, blind vs blind_date, or damn vs damn_good. Most sentences misclassified by SenticNet 7, instead, were using sarcasm or contained antithetic opinion targets. An ablation study showed that sentic paths enable a 6.8% average gain over using XLNet and affective similarity prediction alone. Finally, SenticNet 7 also stands tall against its predecessors, e.g., SenticNet 5 (Cambria et al., 2018) and SenticNet 6 (Cambria et al., 2020), and recent subsymbolic NLP models, e.g., Google's T5 (Raffel et al., 2019), which achieves slightly better accuracy on STS but it is supervised, hard to reproduce, uninterpretable, and not explainable.

5. Conclusion

AI systems are becoming more and more accurate but, at the same time, less and less transparent. In this work, we attempt to reverse the latter trend in the context of sentiment analysis by developing SenticNet 7, a neurosymbolic AI system that leverages subsymbolic models, such as auto-regressive language models and kernel methods, to build symbolic representations that convert natural language to a sort of protolanguage to better infer polarity from text. As a result, SenticNet 7 is unsupervised, reproducible, interpretable, trustworthy, and explainable while maintaining comparable accuracy to recent state-of-the-art subsymbolic models.

[2] https://sentic.net/downloads

[3] https://sentic.net/api

6. Bibliographical References

Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160.

Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social Networks*, 25(3):211–230.

Alivisatos, P., Chun, M., Church, G., Greenspan, R., Roukes, M., and Yuste, R. (2012). The brain activity map project and the challenge of functional connectomics. *Neuroview*, 74(6):970–974.

Analytics, S. (2012). Sanders dataset.

Arrieta, A. B., Rodríguez, N. D., Ser, J. D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, pages 2200–2204.

Bacco, L., Cimino, A., Dell'Orletta, F., and Merone, M. (2021). Extractive summarization for explainable sentiment analysis using transformers. In *Proceedings of International Workshop on Deep Learning meets Ontologies and Natural Language Processing*, volume 2918, pages 62–73.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *science*, 286(5439):509–512.

Barnes, J., Kurtz, R., Oepen, S., Øvrelid, L., and Velldal, E. (2021). Structured sentiment analysis as dependency graph parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistcs and the International Joint Conference on Natural Language Processing*, pages 3387–3402.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, pages 440–447.

Bodria, F., Panisson, A., Perotti, A., and Piaggesi, S. (2020). Explainability methods for natural language processing: Applications to sentiment analysis. In *Proceedings of Italian Symposium on Advanced Database Systems*, volume 2646, pages 100–107.

Bradley, M. and Lang, P. (1999). Affective norms for english words (ANEW): Stimuli, instruction manual and affective ratings. Technical report, The Center for Research in Psychophysiology, University of Florida.

Cambria, E. and White, B. (2014). Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2):48–57.

Cambria, E., Fu, J., Bisio, F., and Poria, S. (2015). AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis. In *AAAI*, pages 508–514.

Cambria, E., Poria, S., Hazarika, D., and Kwok, K. (2018). SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In *AAAI*, pages 1795–1802.

Cambria, E., Li, Y., Xing, F., Poria, S., and Kwok, K. (2020). SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. In *CIKM*, pages 105–114.

Cerini, S., Compagnoni, V., Demontis, A., Formentelli, M., and Gandini, C. (2007). Micro-WNOp: A gold standard for the evaluation of automatically compiled lexical resources for opinion mining. *Language resources and linguistic theory: Typology, Second Language Acquisition, English linguistics*, pages 200–210.

d'Avila Garcez, A. and Lamb, L. (2020). Neurosymbolic AI: The 3rd wave. *arXiv:2012.05876*.

Deng, L. and Wiebe, J. (2015). MPQA 3.0: An entity/event-level sentiment corpus. In *NAACL*, pages 1323–1328.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Eco, U. (1984). *Semiotics and Philosophy of Language*. Indiana University Press.

Ferrarotti, M., Rocchia, W., and Decherchi, S. (2019). Finding principal paths in data space. *IEEE Transactions on Neural Networks and Learning Systems*, 30(8):2449–2462.

Francis, M. and Pennebaker, J. (1993). LIWC: Linguistic inquiry and word count. Technical report, Southern Methodist University, Dallas.

Gardini, E., Cavalli, A., and Decherchi, S. (2021). An ab initio local principal path algorithm. In *International Joint Conference on Neural Networks, Accepted paper.*

Gatti, L., Guerini, M., and Turchi, M. (2016). SentiWords: Deriving a high precision and high coverage lexicon for sentiment analysis. *IEEE Transactions on Affective Computing*, 7(4):409–421.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12).

Goodfellow, I., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *ICLR*.

Greimas, A. J. (1987). *On Meaning: Selected Writings in Semiotic Theory*, volume 38 of *Theory and History of Literature*. University of Minnesota Press.

Gunning, D. and Aha, D. W. (2019). Darpa's explainable artificial intelligence (XAI) program. *AI Magazine*, 40(2):44–58.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *SIGKDD*, pages 168–177.

Hu, W., Wang, H., Qiu, Z., Nie, C., Yan, L., and Du, B. (2017). An event detection method for social networks based on hybrid link prediction and quantum swarm intelligent. *World Wide Web*, 20(4):775–795.

Hutto, C. J. and Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *International AAAI Conference on Web and Social Media*, pages 216–225.

Jackendoff, R. (1976). Toward an explanatory semantic representation. *Linguistic Inquiry*, 7(1):89–150.

Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43.

Krishnaswamy, N. and Pustejovsky, J. (2020). Neurosymbolic AI for situated language understanding. In *arXiv:2012.02947*.

Leicht, E. A., Holme, P., and Newman, M. E. (2006). Vertex similarity in networks. *Physical Review E*, 73(2):026120.

Li, R., Chen, H., Feng, F., Ma, Z., Wang, X., and Hovy, E. H. (2021). Dual graph convolutional networks for aspect-based sentiment analysis. In *Proceedings of ACL/IJCNLP*, pages 6319–6329.

Liben-Nowell, D. and Kleinberg, J. M. (2007). The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031.

Maas, A., Daly, R., Pham, P., Huang, D., Ng, A., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *ACL*, pages 142–150.

Mao, J., Gan, C., Kohli, P., Tenenbaum, J., and Wu, J. (2019). The neurosymbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *ICLR*.

Martínez, V., Berzal, F., and Talavera, J. C. C. (2017). A survey of link prediction in complex networks. *ACM Comput. Surv.*, 49(4):69:1–69:33.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Minsky, M. (1975). A framework for representing knowledge. In Patrick Winston, editor, *The psychology of computer vision*. McGraw-Hill, New York.

Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Moreno-Ortiz et al., A. (2018). Lingmotif-lex: a wide-coverage, state-of-the-art lexicon for sentiment analysis. In *LREC*, pages 2653–2659.

Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., and Wilson, T. (2013). SemEval-2013 task 2: Sentiment analysis in twitter. In *SemEval*, pages 312–320.

Nakov, P., Ritter, A., Rosentha, S., Sebastiani, F., and Stoyanov, V. (2016). Semeval-2016 task 4: Sentiment analysis in twitter. In *SemEval*.

Nielsen, F. (2011). A new anew: Evaluation of a word list for sentiment analysis in microblogs. *CoRR*, abs/1103.2903.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, pages 115–124, Ann Arbor.

Peirce, C. S. (1902). *Logic, Regarded As Semeiotic*. Arisbe.

Qiu, Z., Hu, W., Wu, J., Tang, Z., Jia, X., and Cambria, E. (2022). High order node similarity preserving method for robust network representation. In *under review*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ragusa, E., Gastaldo, P., Zunino, R., Ferrarotti, M. J., Rocchia, W., and Decherchi, S. (2019). Cognitive insights into sentic spaces using principal paths. *Cognitive Computation*, 11(5):656–675.

Rajagopal, D., Cambria, E., Olsher, D., and Kwok, K. (2013). A graph-based approach to commonsense concept extraction and semantic similarity detection. In *World Wide Web*, pages 565–570.

Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., and Stoyanov, V. (2015). Semeval-2015 task 10: Sentiment analysis in twitter. In *SemEval*, pages 451–463.

Rumelhart, D. and Ortony, A. (1977). The representation of knowledge in memory. In *Schooling and the acquisition of knowledge*. Erlbaum, Hillsdale, NJ.

Saif, H., Fernandez, M., He, Y., and Alani, H. (2013). Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold. In *International Conference of the Italian Association for Artificial Intelligence*.

Sarker, M. K., Zhou, L., Eberhart, A., and Hitzler, P. (2021). Neuro-symbolic artificial intelligence: Current trends. *arXiv:2105.05330*.

Schank, R. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3:552–631.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642.

Speer, R., Chin, J., and Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

Stone, P., Dunphy, D., Smith, M., and Ogilvie, D. (1966). *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, MA.

Strapparava, C. and Valitutti, A. (2004). WordNet-Affect: An affective extension of WordNet. In *LREC*, pages 1083–1086.

Susanto, Y., Livingstone, A., Ng, B. C., and Cambria, E. (2020). The hourglass model revisited. *IEEE Intelligent Systems*, 35(5).

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Tang, D., Cai, Y., Zhao, J., and Xue, Y. (2014). A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous non-linear large scale problems. *Inf. Sci.*, 289:162–189.

Tang, J., Li, K., Jin, X., Cichocki, A., Zhao, Q., and Kong, W. (2021). CTFN: hierarchical learning for multimodal sentiment analysis using coupled-translation fusion network. In *Proceedings of ACL/IJCNLP*, pages 5301–5311.

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American society for information science and technology*, 61(12):2544–2558.

Wang, L. and Xia, R. (2017). Sentiment lexicon construction with representation learning based on hierarchical sentiment supervision. In *EMNLP*, pages 502–510.

Wang, P.-W., Donti, P., Wilder, B., and Kolter, Z. (2019). SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *ICML*, pages 6545–6554.

Warriner et al., A. B. (2013). Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior Research Methods*, 45:1191–1207.

Wierzbicka, A. (1996). *Semantics: Primes and Universals*. Oxford University Press.

Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. (2005). Opinionfinder: A system for subjectivity analysis. In *HLT/EMNLP*, pages 34–35.

Xing, F., Pallucchini, F., and Cambria, E. (2019). Cognitive-inspired domain adaptation of sentiment lexicons. *Information Processing and Management*, 56(3):554–564.

Yan, H., Dai, J., Ji, T., Qiu, X., and Zhang, Z. (2021). A unified generative framework for aspect-based sentiment analysis. In *Proceedings of ACL/IJCNLP*, pages 2416–2429.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. (2019). XLNet: Generalized autoregressive pretraining for language understanding. In *NIPS*.

Yang, L., Li, J., Cunningham, P., Zhang, Y., Smyth, B., and Dong, R. (2021). Exploring the efficacy of automatically generated counterfactuals for sentiment analysis. In *Proceedings of ACL/IJCNLP*, pages 306–316.

Zhu, X., Kiritchenko, S., and Mohammad, S. (2014). NRC-Canada-2014: Recent improvements in the sentiment analysis of tweets. In *SemEval*, pages 443–447.