SenticNet 8:
Fusing Emotion AI and Commonsense AI for Interpretable, Trustworthy, and Explainable Affective Computing

Erik Cambria[1][0000−0002−3030−1280],
Xulang Zhang[1][0000−0002−2016−7575], Rui Mao[1][0000−0002−1082−8755],
Melvin Chen[1][0000−0003−1756−8666], and Kenneth Kwok[2][0000−0002−5387−8197]

[1] Nanyang Technological University, Singapore
{cambria,rui.mao,xulang.zhang,melvinchen}@ntu.edu.sg
https://sentic.net
[2] Agency for Science, Technology and Research (A*STAR), Singapore
kenkwok@ihpc.a-star.edu.sg

Abstract. ChatGPT has stunned the world with its ability to generate detailed, original, and accurate responses to prompts. While it unlocked solutions to problems that were previously considered unsolvable, however, it also introduced new ones. One of such problems is the phenomenon known as hallucination, the generation of content that is nonsensical or unfaithful to the provided source content. In this work, we propose SenticNet 8, a neurosymbolic AI framework leveraging an ensemble of commonsense knowledge representation and hierarchical attention networks, which aims to mitigate some of these issues in the context of affective computing. In particular, we focus on the tasks of sentiment analysis, personality prediction, and suicidal ideation detection. Results show that SenticNet 8 presents superior accuracy with respect to all four baselines, namely: bag-of-words, word2vec, RoBERTa, and ChatGPT. Unlike these baselines, moreover, SenticNet 8 is also fully interpretable, trustworthy, and explainable.

Keywords: Artificial Intelligence · Trustworthiness · Interpretability · Explainability · Affective Computing.

1 Introduction

Generative pretrained transformer (GPT) models enabled humanity to finally design an algorithm able to pass the famous machine intelligence test devised by Alan Turing some seventy years ago. With approximately 1 trillion parameters, ChatGPT has revolutionized the world of natural language processing (NLP) thanks to the high accuracy it can obtain on several information retrieval tasks [40, 35]. However, it still presents several issues that limit its widespread adoption, especially in contexts such as medical, ethical, or fail-safe applications [45, 2].

Some researchers defined large language models (LLMs) like ChatGPT as 'stochastic parrots' [4], i.e., systems that haphazardly stitch together sequences of linguistic forms it has observed in its vast training data, according to probabilistic information about how they combine, but without any reference to meaning. LLMs, in fact, are trained (mostly in a self-supervised manner) on 'broad' data, which leads to homogenization (i.e., using same model for fine-tuning and training for different downstream tasks) and emergence (i.e., LLMs can solve tasks they were not originally trained upon). This poses several risks [8], including 'hallucination' [26], which can lead to several ChatGPT failures, including reasoning, factual errors, math, coding, and bias [9]. ChatGPT, moreover, is not interpretable (because we do not get to see its true inner workings, e.g., how cause and effect are associated); it is not trustworthy (because it is only as good as its training data and it often lacks the commonsense knowledge required for disambiguation); and it is not explainable (because we do not get any explanation about the decision-making processes that produce its final results).

In this work, we aim to mitigate these issues in the context of affective computing. We propose SenticNet 8, a neurosymbolic AI framework leveraging an ensemble of commonsense knowledge representation and hierarchical attention networks, which automatically extracts important affective information (such as sentiment polarity, emotion labels, opinion targets, emotion-cause pairs, polarity intensity, personality traits, etc.) from both formal and informal natural language text with state-of-the-art accuracy. This is enabled by an approach to NLP that is both top-down and bottom-up: top-down for the fact that SenticNet 8 leverages symbolic models (namely, conceptual dependency theory and a semantic network of affective commonsense knowledge) to encode meaning; bottom-up because we use sub-symbolic paradigms (namely, hierarchical attention networks and LLMs) to infer syntactic patterns from data. We compare SenticNet 8 with ChatGPT, a robust language model (RoBERTa), pretrained embeddings (word2vec), and the bag-of-words (BoW) model. Results show that SenticNet 8 generally presents superior accuracy with respect to all four models. Unlike these baselines, moreover, SenticNet 8 is also interpretable, trustworthy, and explainable. The remainder of the paper is organized as follows: Section 2 lists recent related works; Section 3 describes the proposed framework; Section 4 presents experimental results; Section 5 discusses insights gained; finally, Section 6 provides concluding remarks.

2 Related Work

Since Ancient Greece, it has been widely acknowledged that humans seek explanations in an attempt to understand the world [30]. This ubiquitous search for answers and explanations is inherent to human nature and fundamental to integrate technology into everyday lives. As technology advances and human-computer interaction (HCI) becomes more prevalent, in fact, the need for understanding and explaining the decision-making processes of affective computing models has become paramount [16, 29].

Various studies have focused on developing machine learning models for emotion recognition from different modalities, such as facial expressions, speech, text, and physiological signals [20]. Traditional approaches include feature engineering and classical machine learning techniques. More recently, deep learning methods, especially transformers, have demonstrated remarkable performance in this domain [50]. However, the black-box nature of deep learning models has raised concerns about their interpretability, motivating researchers to delve into explainable machine learning techniques.

Explainable artificial intelligence (XAI) offers methodologies and tools to open the black box of machine learning models and make their decision-making processes understandable to humans [6, 23, 10, 12]. Interpretability techniques, such as saliency maps, feature visualization, and activation maximization, have been applied to emotion recognition systems to highlight the regions in input data that are influential in driving the model's predictions. XAI has enabled researchers to gain insights into how certain patterns or cues in facial expressions, speech, or text are associated with specific emotional states [18, 31].

Explainable affective computing represents an ongoing and significant area of research that seeks to bridge the gap between the powerful predictive capabilities of AI systems and the need for human-understandable decision-making processes [22, 27, 3]. By drawing from various fields, including XAI, HCI, and ethics, researchers aim to create emotionally intelligent systems that are transparent, trustworthy, and capable of enhancing human-computer interfaces in a more natural and empathetic manner [34, 19]. As the field continues to evolve, it is expected that advances in explainability will lead to more responsible and ethically-aware affective computing applications in various domains, such as healthcare, education, and human-robot interaction [13].

3 Proposed Framework

SenticNet 8 aims to mitigate one important issue with current AI models: the symbol grounding problem. Solving this problem is crucial for achieving XAI because it addresses the foundational challenge of connecting abstract symbols or representations to concrete real-world entities and experiences. By establishing a clear and meaningful connection between symbols and their referents, XAI systems can provide more understandable and interpretable explanations for their actions and decisions. This is done through a three-step normalization process.

Firstly, a "syntactic normalization" step leverages a graph-based approach [14] to replace inflections like `creates`, `built`, and `constructed` with their lemmas, e.g., `create`, `build`, and `construct`, respectively (Fig. 1).

Secondly, a "semantic normalization" step (explained in detail later) leverages conceptual dependency theory [44, 38, 25, 43, 48] and a commonsense knowledge graph [46] to replace words like `create`, `build`, `construct`, with their corresponding primitive, e.g., GENERATE(x), where x is the direct object indicating the person or thing acted upon by the primitive.

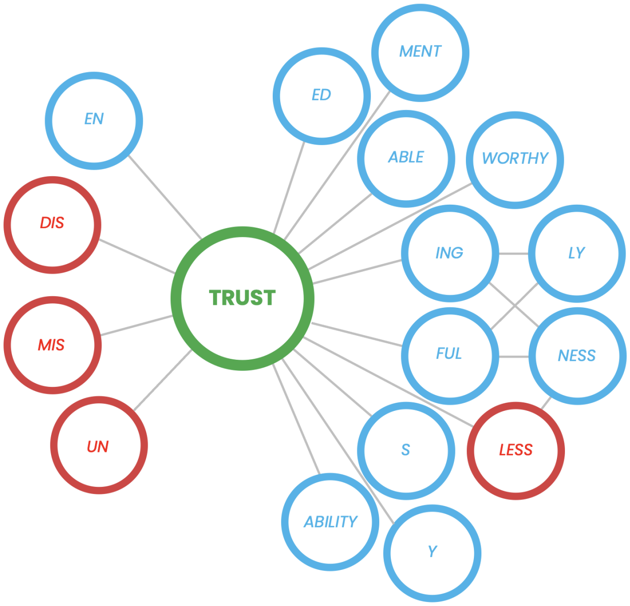


Fig. 1. An example of syntactic normalization graph, which allows words like `trustfulness`, `entrusted`, and `trustingly` to be all normalized as `trust`.

Finally, a "pragmatic normalization" step employs the logic definition associated with such primitive, e.g., the transition from a state of inexistence to a state of existence ($\nexists x \rightarrow \exists x$), for the downstream task. For example, if the downstream task is sentiment analysis, the polarity value associated with x will be the resulting output. Conversely, if the detected primitive is `TERMINATE` ($\exists x \rightarrow \nexists x$), the opposite polarity of x will be returned (Fig. 2).

A key novelty introduced by this paper is the design of the second step in the above-mentioned normalization process, i.e., the semantic normalization component (Fig. 3). Such component comprises of two main modules, namely polarity detection and lexical substitution (explained later). Given an input sentence $w = (w_1, w_2, \ldots, w_L)$, we first aim to identify the target word t that should be replaced by a primitive for the downstream prediction from w.

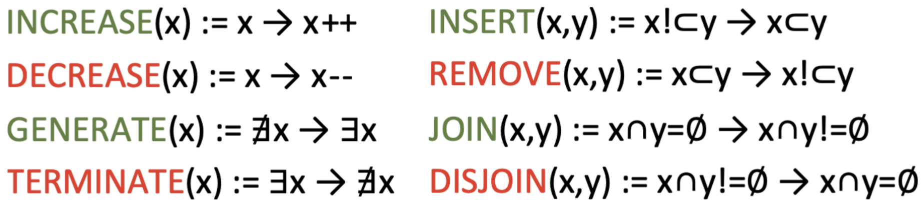

INCREASE(x) := x → x++ INSERT(x,y) := x!⊂y → x⊂y
DECREASE(x) := x → x-- REMOVE(x,y) := x⊂y → x!⊂y
GENERATE(x) := ∄x → ∃x JOIN(x,y) := x∩y=∅ → x∩y!=∅
TERMINATE(x) := ∃x → ∄x DISJOIN(x,y) := x∩y!=∅ → x∩y=∅

Fig. 2. Some examples of pragmatic normalization rules.

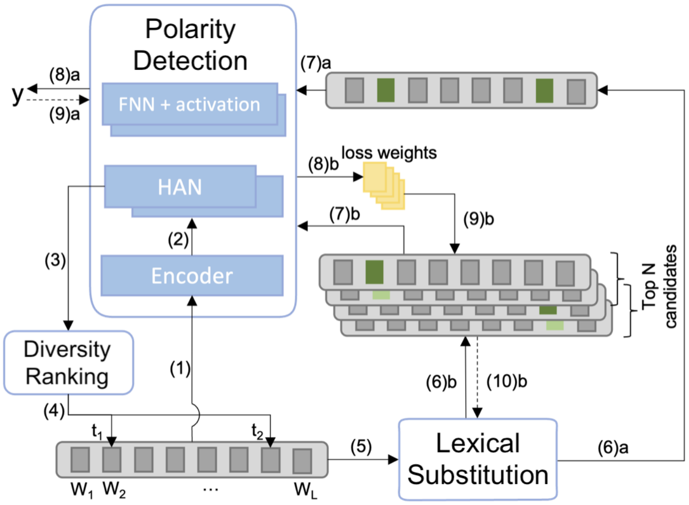


Fig. 3. Semantic normalization component. Dotted lines represent backpropagation. Green blocks indicate the original word being replaced by a word provided by the lexical substitution module. The darker the green, the higher probability it is assigned to by the model.

We extract K primitives from SenticNet 7 [11] as candidates $c = (c_1, \ldots, c_K)$, forming the input (s) as

$$\begin{aligned} s = &< s >, w_1, w_2, \ldots, t, \ldots, w_L, < /s >, \\ &c_1, < /s >, c_2, < /s >, \ldots, c_K, < /s > . \end{aligned} \tag{1}$$

$< s >$ and $< /s >$ are special tokens that were defined by the employed pre-trained language model. The lexical substitution module should identify the best candidate from c as the substitute $\hat{c}$ which retains the original meaning of t in the context by using contrast learning. Thus, $\hat{c}$ is the symbolic representation of t in context w. Then, the substitute input $w^f = (w_1, \ldots, \hat{c}, \ldots, w_L)$ is fed into a neural network classifier to predict a sentiment label. The objective is that the substitute input w^f increases the probability of correct sentiment prediction.

First, through steps (1) and (2) in Fig. 3, the original input is fed into the encoder and the interpretable attention module to obtain the top I tokens with the highest attention weights, which contribute the most to sentiment prediction. In step (3), the sense diversity of each token is computed as the average distance of the token's hidden state to those of its substitution candidates, i.e., the most relevant primitive. Then, through step (4), the top J tokens that are most likely to be replaced by a primitive are selected as targets, because these target tokens may be associated to different primitives in different contexts. Subsequently, the pre-trained lexical substitution module provides the best substitution to replace each target token, as shown in steps (5) and (6)a. The new input sentence is passed onto the polarity detection module for final prediction through steps (7)a and (8)a, which is used for the sentiment module backpropagation in step (9)a.

In order to fine-tune the lexical substitution module and, hence, provide better primitive substitutions that improve the accuracy of sentiment analysis, we implement a dynamic rewarding mechanism (explained later). As shown in steps (5) and (6)b, for each target word, the top N candidates are selected. Each of them is seen as a substitute candidate to calculate the probability of correct sentiment prediction after the replacement in steps (7)b and (8)b. Then, each probability is used to dynamically compute the loss weight when the corresponding primitive candidate is learned by the lexical substitution module as a ground truth as shown in (9)b and (10)b. If the replacement by a candidate leads to a higher chance of predicting the right sentiment, the candidate is more likely given a higher rank assigned by the lexical substitution module. Otherwise, the module would be less likely to select the candidate as the best replacement. The detailed training process can be seen in Algorithm 1.

3.1 Sentiment Analysis with Interpretability

Given input sentence $w = (w_1, \ldots, w_L)$, the goal is to predict the correct sentiment label $\tilde{y}$. The input is first fed into a pre-trained encoder:

$$V = Encoder(w), \tag{2}$$

where V is hidden states. Next, we aim to find the tokens that contribute the most to sentiment inference. We adopt an interpretable attention module called hierarchical attention network (HAN), which effectively encodes hidden states with multiple non-linear projections and ranks the most influential tokens based on attention [49]. We stack two blocks of HAN to form our attention module.

$$q, a = HAN(HAN(V)), \tag{3}$$

where vector q is the yielded hidden state and a is the attention weights, indicating the contribution of each token to the final sentiment prediction. To obtain the sentiment prediction, q is passed on to two layers of feedforward neural networks (FNN) to obtain the probability distribution of sentiment prediction, with the first one being activated by ReLU [1], and the second by softmax.

$$h = ReLU(FNN_1(q)) \tag{4}$$

$$\hat{y} = softmax(FNN_2(h)) \tag{5}$$

We denote the prediction of the sentiment analysis module as $\hat{y}^f$ when the input is w^f, which denotes a substitute w where all selected target tokens are replaced by relevant primitives. Thus, the sentiment analysis loss is computed as:

$$\mathcal{L}^{(sa)} = CrossEntropy(\hat{y}^f, \tilde{y}) \tag{6}$$

3.2 Generalization by Lexical Substitution

For the lexical substitution module, we employ a novel pre-training paradigm, termed anomalous language modeling (ALM), which was pre-trained to detect

Algorithm 1: Semantic normalization.

```
Initialize polarity detection module as $\varPhi$, pre-trained lexical substitution
  module as $\varPsi$;
Initialize hyperparameters $\beta$, $I$,$J$, $N$;
while not done do
    Sample a sentence $w = w_1, w_2, \ldots, w_L$;
    for $l$=1:$L$ do
        Compute the attention weight $a_l$ of token $w_l$;
    end
    $w^{att} \leftarrow$ Top $I$ of $w$ ordered by attention weights $a = a_1, a_2, \ldots, a_L$;
    for $i$=1:$I$ do
        $c \leftarrow$ all possible primitive candidates with the same part-of-speech type
          as $w_i^{att}$ from SenticNet 7;
        Compute the average Eulidean distance $d_i$ between the hidden states of
          $w_i^{att}$ and $c$, produced by the encoder in $\varPhi$;
    end
    $t = (t_1, t_2, \ldots, t_J) \leftarrow$ Top $J$ of $w^{att}$ ordered by $d = d_1, d_2, \ldots, d_I$;
    $w^f \leftarrow w$ ;
    for $j$=1:$J$ do
        Input $t_j$ into $\varPsi$ to produce top $N$ candidates $\hat{c} = (\hat{c}_1, \ldots, \hat{c}_N)$, ordered
          by probability;
        Replace $t_j$ in $w^f$ with $\hat{c}_1$;
        $s \leftarrow (< s >, w_1, \ldots, t_i, \ldots, w_L, < /s >,$
        $\hat{c}_1, < /s >, \ldots, \hat{c}_N, < /s >)$;
        for $n$=1:$N$ do
            $w^s \leftarrow$ Replace $t_j$ in $w$ with $\hat{c}_n$;
            Input $w^s$ into $\varPhi$ to obtain the probability of correct sentiment
              prediction $P(\hat{y} = \tilde{y})$ ;
            $\theta_{j,n} \leftarrow \beta P(\hat{y} = \tilde{y})^2$ ;
            Compute $\mathcal{L}_{j,n}^{(ls)}$ by feeding $s$ with $\hat{c}_n$ labeled as true substitute into $\varPsi$
              ;
            $\mathcal{L}_{j,n}^{(ls)} \leftarrow \theta_{j,n} \mathcal{L}_{j,n}^{(ls)}$;
        end
    end
    $\mathcal{L}^{(ls)} \leftarrow \mathcal{L}_{1,1}^{(ls)} + \cdots + \mathcal{L}_{1,N}^{(ls)} + \mathcal{L}_{J,1}^{(ls)} + \cdots + \mathcal{L}_{J,N}^{(ls)}$;
    Compute sentiment analysis loss $\mathcal{L}^{(sa)}$ using $w^f$ as input;
    $L \leftarrow \mathcal{L}^{(sa)} + \mathcal{L}^{(ls)}$ ;
end
```

anomalous substituted words from a sequence and retrieve the original words from a set of candidates that contains a positive sample (appropriate primitive of the original word according to the context) and multiple hard negative samples (other primitives associated with the original word) via contrastive learning. We use the candidates from SenticNet 7 to formulate our input s as in Formula 1. The candidate with the highest score is set as the ground truth substitution $\tilde{c}$.

Given the input s, the model encodes it as:

$$U, R = ALM(s), \tag{7}$$

where $U = [u_1, \ldots, u_L]$ is the hidden states of the input sentence, and $R = [r_1, \ldots, r_K]$ is the hidden states of the candidates. We denote the representation of the target word as u_t ($t \in \{1, \ldots, L\}$). Our training objective is to a) close the distance between the representation of the ground truth candidate r_k, ($k \in \{1, \ldots, K\}$) with u_t, and b) push away incorrect candidates representations r_j ($j \in \{1, \ldots, K | j \neq k\}$) from u_t. Namely, (r_k, u_t) will be regarded as a positive pair, while (r_j, u_t) will be regarded as a negative pair. We follow the InfoNCE loss [39] to achieve these goals, which can be formulated as :

$$\mathcal{L}^{(tune)} = -\sum_i \log \frac{exp(d(u_t, r_k)/\tau)}{\sum_j exp(d(u_t, r_i)/\tau)}, \tag{8}$$

where $i \in \{1, \ldots, K\}$, τ is a temperature hyper-parameter, and $d(\cdot)$ is Euclidean distance. During the inference stage, we choose the candidate $\hat{c}$ whose corresponding hidden state r_k is the most similar to u_t, measured by Euclidean distance:

$$\hat{c} = \arg\min(d(u_t, r_k)) \tag{9}$$

We then use the resulting lexical substitution module to find primitive replacements for the selected target words in sentiment analysis input. To determine which words are selected as primitive targets, we select the top I words in the input with the highest a produced by Equation 3, forming the set w^{att}. We denote their corresponding representations as $V^{att} = \{v_1^{att}, \ldots, v_I^{att}\}$. For each v_i^{att}, we compute its average Euclidean distance to all of its candidates' hidden states. The candidates consist of relevant primitives from SenticNet 7 under the same part-of-speech type, which are transformed into hidden states $G = \{g_1, \ldots, g_M\}$ using Equation 2. M represents the number of primitive candidates from SenticNet 7. The top J words in w^{att} with the largest corresponding average distance are considered to be the ones with the most diverse word meanings, and thus are more likely to be replaced. Hence, target words $t = (t_1, \ldots, t_J)$ are selected by finding each corresponding v_j^{att}:

$$v_j^{att} = \arg\max_i\left(\frac{1}{M}\sum_M d(v_i^{att}, g_m)\right). \tag{10}$$

3.3 Dynamic Rewarding Mechanism

To fine-tune the lexical substitution module on the downstream task of sentiment analysis, we utilize the top N candidates $\hat{c}_j = \{\hat{c}_{j,1}, \ldots, \hat{c}_{j,n}, \ldots, \hat{c}_{j,N}\}$ produced by the lexical substitution module, for each target word t_j.

The new input resulting from t_j being replaced by candidate $\hat{c}_{j,n}$ is denoted as $w^{j,n}$. Same with Equation 5, the probability distribution of a sentiment prediction from $w^{j,n}$ is:

$$P(\hat{y})_{j,n} = softmax(FNN_2(h^{j,n})), \tag{11}$$

where $h^{j,n}$ are the hidden states of $w^{j,n}$ produced by Equation 4 in the sentiment analysis module.

To adjust the model in such a way that a more accurate sentiment prediction results in a higher reward for the corresponding substitution output, we compute the loss weight for $\hat{c}_{j,n}$ being the correct substitution prediction as:

$$\theta_{j,n} = \beta P(\hat{y} = \tilde{y})^2_{j,n}, \tag{12}$$

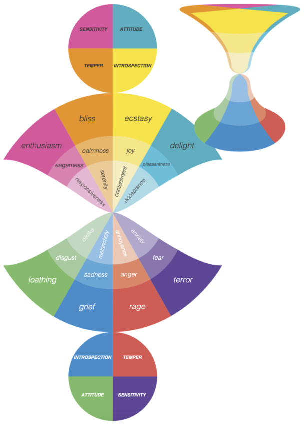
where β is a hyperparameter for balancing the sentiment analysis and lexical substitution losses, $\tilde{y}$ is the ground truth sentiment label as defined above. We formulate the input s^j to lexical substitution module with sentence w and candidates $\hat{c}_j$, using Formula 1. Similar to Equation 8, the loss ($\mathcal{L}^{(ls)}_{j,n}$) of the lexical substitution module when $\hat{c}_{j,n}$ is considered as gold standard is computed as follows:

$$\mathcal{L}^{(ls)}_{j,n} = -\sum_i \log \frac{exp(d(u_t, \hat{c}_{j,n})/\tau)}{\sum_j exp(d(u_t, \hat{c}_j)/\tau)}. \tag{13}$$

Finally, the total loss is computed as:

$$\mathcal{L} = \mathcal{L}^{(sa)} + \theta_{1,1}\mathcal{L}^{(ls)}_{1,1} + \cdots + \theta_{J,N}\mathcal{L}^{(ls)}_{J,N}. \tag{14}$$

The algorithm also works well with emoticons and emojis, which are very important sentiment indicators in social media text. These are aptly replaced with their corresponding primitive, which in most cases is an emotion primitive. In particular, we use the Hourglass of Emotions [47] as emotion categorization model for both verbal and nonverbal content (Fig. 4).



		INTROSPECTION			
ECSTASY	JOY	CONTENTMENT	MELANCHOLY	SADNESS	GRIEF
elation	happiness	satisfaction	pensiveness	unhappiness	desperation
jubilation	cheerfulness	gratification	abandonment	sorrow	gloom
exultation	joviality	fulfilment	emptiness	dejection	depression
glee	gaiety	light-heartedness	down-heartedness	heavy-heartedness	broken-heartedness
felicity	high-spiritedness	frivolity	nostalgia	low-spiritedness	woe
		TEMPER			
BLISS	CALMNESS	SERENITY	ANNOYANCE	ANGER	RAGE
placidity	tranquillity	quietude	disquietude	vexation	fury
peacefulness	equanimity	comfort	discomfort	exasperation	wrath
beatitude	composure	ease	unease	aggressiveness	ferocity
gladness	restfulness	imperturbability	perturbability	madness	enragement
relief	soothingness	carefreeness	frustration	acrimoniousness	vengeance
		ATTITUDE			
DELIGHT	PLEASANTNESS	ACCEPTANCE	DISLIKE	DISGUST	LOATHING
admiration	appreciation	approval	disapproval	disappointment	contempt
adoration	fondness	favorability	distaste	detestation	revulsion
glorification	predilection	propensity	rejection	disdain	scorn
devotion	respect	belief	disbelief	disrespect	repugnance
enthrallment	trust	worthiness	worthlessness	distrust	abhorrence
		SENSITIVITY			
ENTHUSIASM	EAGERNESS	RESPONSIVENESS	ANXIETY	FEAR	TERROR
zeal	keenness	decisiveness	indecisiveness	fright	horror
zest	willingness	receptiveness	apprehension	dread	panic
passion	motivation	agreeableness	helplessness	trepidation	appalment
avidity	inspiration	approachableness	agitation	angst	petrification
fervor	dedication	amenability	discouragement	scare	aghastness

Fig. 4. The Hourglass of Emotions. Unlike other emotion categorization models, the Hourglass represents antithetic emotions very efficiently. Its mirroring capability, in fact, enables the easy handling of negations and other variations of language that can change the sentiment of words otherwise taken in isolation.

4 Evaluation

We test SenticNet 8 against ChatGPT and three more NLP models on three different affective computing datasets. In the following three sections, we describe in detail baselines adopted, datasets used, and results obtained, respectively.

4.1 Baselines

In order to compare the performance of SenticNet 8 on the different tasks, we need to use baselines and train them on the Train portion (while validating on the Dev portion). Besides ChatGPT, we employ three more baselines, which serve as the specialized models specifically tailored for the corresponding downstream task: a robust language model (RoBERTa) trained on a large amount of text; a baseline which uses a word model by employing pretrained word2vec (W2V) embeddings; and a simple Bag-of-Words (BoW) model that utilizes a linear classifier. The hyperparameters of all models are optimized by selecting the hyperparameters yielding the best performance on the Dev portion. Such hyperparameters are tuned using the SMAC toolkit [32], which is based on Bayesian Optimization. The selected hyperparameters are listed in Table 1. Fig. 5 illustrates the pipelines of all methods.

Bag of Words BoW is a simple model that uses only in-domain data for training and no other data for either up- or downstreaming. We utilize the classical technique term frequency – inverse document frequency (TF-IDF), which tokenizes the sentences into words, then, a sentence is represented by a vector of the counts of the words it contains. The vector is then normalized by the term frequency across the entire Train set of the corresponding dataset. We tune the learning rate η of SGD using SMAC.

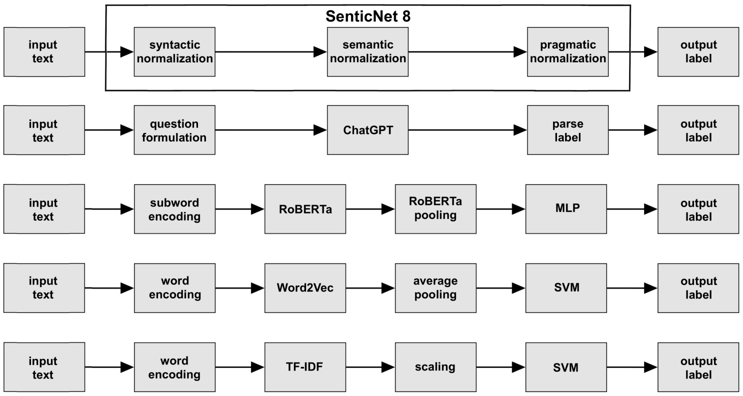


Fig. 5. Evaluation pipelines of SenticNet 8, ChatGPT, RoBERTa, word2vec, and BoW (from top to bottom).

Word2vec Embeddings The baseline word2vec [36, 37] makes use of pretrained word embeddings, which are trained on a large amounts of text from Google News. The model operates by tokenizing a given text into words, each word is assigned an embedding from the pretrained embeddings. The embeddings are then averaged for all words to give a static feature vector of size 300 for the entire string. An SVM model [7] is then used to predict the given task.

RoBERTa Language Model The baseline RoBERTa [33] is a pretrained BERT model, which has a transformer architecture. [33] trained two instances of RoBERTa; we use the smaller one, namely *RoBERTa-base*, consisting of 110 million parameters. The model starts by tokenizing a text using subword encoding, which is a hybrid representation between character-based and word-based encodings. The tokens are then fed to RoBERTa to obtain a sequence of embeddings.

ChatGPT We introduce the stages of querying ChatGPT as shown in Fig. 5. The general mechanism for collecting answers for each NLP task is as follows:

1. Reformat all the texts of the Test portion of the dataset, by using a format that asks ChatGPT what is their guess about the label of the text.
2. Chunk the examples into 25 examples per chunk.
3. For each chunk, open a new ChatGPT Conversation.
4. Ask ChatGPT (manually) the reformatted question for each example, one-by-one, and collect the answers.
5. Repeat the steps 3–4 until the predictions for the whole Test set are finished.
6. Postprocess the results in case they need some cleanup.

The formats used for the three NLP tasks are shown in the following snippets. The *{text}* part needs to be replaced with the sample text. Please note that quotation marks need to be kept since it specifies to ChatGPT that this a placeholder used by the question being asked. The formulations for the three NLP tasks are as follows:

	RoBERTa			W2V	BoW
	N	U	α	C	η
Polarity	3	420	2.97×10^{-5}	0.0144	5.25×10^{-6}
O	2	498	5.66×10^{-4}	0.0378	2.47×10^{-3}
C				0.0472	3.09×10^{-6}
E				0.0069	1.09×10^{-5}
A				0.0218	4.65×10^{-4}
N				0.0657	2.21×10^{-6}
Suicide	3	497	8.04×10^{-4}	10.00	4.71×10^{-6}

Table 1. Hyperparameters of the different baselines. N is the number of hidden layers, U is the number of neurons in the first hidden layer (which is halved for each subsequent layers), and α is the learning rate. Adam optimizer always yields the best results as compared to SGD. C is the SVM parameter for word2vec. η is the learning rate of the SGD in the BoW model.

1. For sentiment analysis, we formulated the question:
   "What is your guess for the sentiment of the text "{text}*", answer positive, neutral, or negative? it does not have to be correct. Do not show any warning after."*
2. For the Big-five personality traits, we asked:
   "What is your guess for the big-five personality traits of someone who said "{text}*", answer low or high with bullet points for the five traits? It does not have to be fully correct. You do not need to explain the traits. Do not show any warning after."*
3. For the suicidal ideation detection, we asked:
   "What is your guess if a person is saying "{text}*" has a suicide tendency or not, answer yes or no? it does not have to be correct. Do not show any warning after."*

The formulation of the question is of crucial importance to the answers ChatGPT generates. For anyone who would like to carry out similar investigations in the future, we report four important lessons learnt:

1. Asking the question directly without requesting ChatGPT to guess made it often answer that there is little information provided to answer the question, and it cannot answer it exactly.
2. It is important to ask *what* the guess is and not "*Can you guess*", because this can generate a response similar to 1., where ChatGPT responds with an answer that starts with "*No, I cannot accurately answer whether...*". Therefore, the question needs to be assertive and specific.
3. The questions for the suicide assessment task may trigger warnings in the responses of ChatGPT due to its sensitive content.
4. We need to specify the exact output format, because ChatGPT can get creative about the formatting of the answer, which can make it hard to collect answers for our experiment.

The responses of ChatGPT need to be parsed, since ChatGPT can give arbitrary formats for a given answer, even when the content is the same. This is predominant in the personality traits, since there are five traits. Sometimes the answers are listed as bullet points, other times they are all in one comma-separated line. Also, it used different delimiters or order, e.g., "Openness: Low", or "Low in Openness", and "Low: Openness". Additionally, in all problems, it sometimes gives an introduction for the answer, for example, "Here is my guess for ..", or "Based on the statement". We solve this issue by using regular expressions to find and edit such responses.

4.2 Datasets

In this section, we briefly introduce the three datasets we used. A summary of their statistics is presented in Table 2. We utilize publicly available datasets for reproducibility.

Dataset	Train	Dev	Test	Pos	Neg
Polarity	1,440,144	159,856	359	182	177
O	6,000	2,000	509	333	176
C				286	223
E				214	295
A				340	169
N				274	235
Suicide	138,479	6,270	496	165	331

Table 2. Statistics of the three datasets used for evaluation.

Polarity Dataset We adopt the Sentiment140 dataset [21] for the sentiment analysis task. The dataset is collected from Twitter, which makes the text very noisy and can pose a challenge against many models (especially word models). The dataset consists of tweets and the corresponding sentiment labels (positive or negative).

Personality Dataset We utilize the First Impressions dataset [41] for the personality task. Personality is represented by the Big-five personality traits (OCEAN), namely, *Openness (to experience)*, *Conscientiousness*, *Extraversion*, *Agreeableness*, and *Neuroticism.* The dataset consists of 15 seconds videos with one speaker, whose personality was manually labelled.

Suicide and Depression Dataset The Suicide and Depression dataset [17] is collected from the Reddit platform, under different subreddits categories, namely "SuicideWatch", "depression", and "teenagers". The texts of the posts from the "teenagers" category are labelled as negative, while the texts from the other two categories are labelled as positive.

4.3 Results

In this section, we review the results of our experiments. In summary, we evaluated the performance of SenticNet 8 (our commonsense knowledge base of 400,000 concepts, available for download at `https://sentic.net/downloads`) against four baselines (namely, ChatGPT, RoBERTa, word2vec, and BoW) on three downstream tasks (namely, sentiment analysis, personality recognition, and suicidal ideation detection). Results are shown in Table 3. We use classification accuracy and unweighted average recall (UAR) as performance measures. UAR has an advantage of exposing if a model is performing very well on a class on the expense of the other class, especially in imbalanced datasets. As also demonstrated by other recent works [24, 42], ChatGPT turned out to be jack of all trades but master of none [28] also in the context of affective computing: while the performance of ChatGPT is acceptable on many different NLP tasks, specialized models like SenticNet 8 (and even RoBERTa in most cases) still outperform it on specific tasks.

Unlike all baselines used in this work, moreover, SenticNet 8 is also interpretable (because the process that generalizes input words and multiword expressions into their corresponding primitives is fully transparent), trustworthy (because classification outputs always come with a confidence score), and explainable (because classification outputs are explicitly linked to emotions and the input concepts that convey these).

	Accuracy					Unweighted Average Recall				
[%]	SenticNet	CGPT	rBERT	W2V	BoW	SenticNet	CGPT	rBERT	W2V	BoW
Polarity	**88.80**	85.51	85.07	79.41	82.54	**88.67**	85.57	85.02	79.40	82.41
O	**67.91**	46.62	66.03	65.28	59.71	**78.27**	50.12	50.94	50.72	55.61
C	**65.97**	57.40	63.72	62.70	55.60	**79.92**	57.70	60.81	60.09	56.30
E	63.19	55.23	**66.09**	59.92	55.24	**72.76**	54.09	62.30	55.56	53.74
A	**67.86**	44.86	67.42	67.21	58.53	**79.93**	48.45	51.93	51.02	55.75
N	**64.53**	47.29	62.17	56.84	56.09	**77.54**	49.16	61.25	54.64	55.88
Suicide	**99.34**	92.71	97.43	92.16	92.78	**99.35**	91.26	97.40	91.23	90.97

Table 3. Classification accuracy and unweighted average recall (in %) of SenticNet 8 against four baselines (CGPT: ChatGPT; rBERT: RoBERTa; W2V: word2vec; BoW: Bag of Words) on three different NLP tasks (Polarity: sentiment analysis; OCEAN: personality prediction; Suicide: suicidal ideation detection). Bold values show the best method for a combination of specific performance metric and prediction target.

5 Discussion

Intuitively, even if real parrots or stochastic ones (LLMs) produce the appropriate linguistic response relative to the task-related prompts for the three abovementioned datasets, we would not deem their linguistic behavior trustworthy unless they possess the relevant natural language understanding. Meaning involves a relation between the linguistic form of data and an extralinguistic reality that is distinct from language. Where M denotes meaning, E denotes the form of natural language expressions, and I denotes communicative intent, this relation may be formally represented as $M \subseteq E \times I$ [5].

M contains ordered pairs (e, i) of natural language expressions (e) and communicative intents (i). Understanding may be interpreted as the process of retrieving i, given e. Since LLMs are pretrained on large datasets and meaning cannot be learnt from linguistic form (e) alone, however impressive their transformer and artificial neural network architecture might be, LLMs will necessarily lack the relevant intentionality. We do not claim that SenticNet 8 possesses either human-level or the requisite level of natural language understanding. However, as SenticNet 8 relies on commonsense knowledge representation as part of its ensemble, it is better able than ChatGPT to track the extralinguistic reality that is distinct from language.

For affective computing tasks, SenticNet 8 is ahead of ChatGPT, as its responses are more firmly grounded in an extralinguistic reality through its reliance on commonsense knowledge representation. SenticNet 8 leverages symbolic models (namely, conceptual dependency theory and a semantic network of commonsense knowledge) to encode meaning in a top-down fashion. Finally, we would like to clarify that this work does not aim to disdain ChatGPT but rather highlight issues to be addressed: we do hope future versions of ChatGPT will overcome at least some of the reported limitations. Given the non-interpretability of its constitutive models, however, it may not happen so soon. As shown in a recent study [15], in fact, ChatGPT seems prone to the “short blanket dilemma": while trying to improve its accuracy on some tasks, OpenAI researchers inadvertently made ChatGPT worse for tasks which it previously excelled at.

6 Conclusion

In this paper, we presented SenticNet 8, a neurosymbolic AI framework leveraging an ensemble of commonsense knowledge representation and hierarchical attention networks, which aims to mitigate the symbol grounding problem. We compared SenticNet 8 against ChatGPT and three more baselines on three downstream tasks. Results show that SenticNet 8's performance is generally superior to all baselines on all tasks. Unlike the other baselines, moreover, SenticNet 8 is interpretable, trustworthy, and explainable.

References

1. Abien Fred Agarap. Deep learning using rectified linear units (ReLU). *arXiv preprint arXiv:1803.08375*, 2018.
2. Mostafa Amin, Erik Cambria, and Björn Schuller. Can ChatGPT's responses boost traditional natural language processing? *IEEE Intelligent Systems*, 38(5):5–11, 2023.
3. Mostafa Amin, Erik Cambria, and Björn Schuller. Will affective computing emerge from foundation models and General AI? A first evaluation on ChatGPT. *IEEE Intelligent Systems*, 38(2):15–23, 2023.
4. Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021.
5. Emily M Bender and Alexander Koller. Climbing towards NLU: On meaning, form, & understanding in the age of data. In *ACL*, pages 5185–5198, 2020.
6. Or Biran and Courtenay Cotton. Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)*, 2017.
7. Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, New York City, NY, USA, 2006.
8. Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the Opportunities and Risks of Foundation Models. *arXiv preprint arXiv:2108.07258*, 2021.

9. Ali Borji. A Categorical Archive of ChatGPT Failures. *arXiv preprint arXiv:2302.03494*, 2023.
10. Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021.
11. Erik Cambria, Qian Liu, Sergio Decherchi, Frank Xing, and Kenneth Kwok. SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis. In *LREC*, pages 3829–3839, 2022.
12. Erik Cambria, Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, and Navid Nobani. A survey on XAI and natural language explanations. *Information Processing and Management*, 60(103111), 2023.
13. Erik Cambria, Rui Mao, Melvin Chen, Zhaoxia Wang, and Seng-Beng Ho. Seven pillars for the future of artificial intelligence. *IEEE Intelligent Systems*, 38(6):62–69, 2023.
14. Erik Cambria, Rui Mao, Sooji Han, and Qian Liu. Sentic parser: A graph-based approach to concept extraction for sentiment analysis. In *Proceedings of ICDM Workshops*, pages 413–420, 2022.
15. Lingjiao Chen, Matei Zaharia, and James Zou. How is chatgpt's behavior changing over time? *arXiv preprint arXiv:2307.09009*, 2023.
16. Karina Cortiñas-Lorenzo and Gerard Lacey. Toward explainable affective computing: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
17. Vishal Desu, Nikhileswar Komati, Sphoorthi Lingamaneni, and Fathimabi Shaik. Suicide and Depression Detection in Social Media Forums. In *Smart Intelligent Computing and Applications, Volume 2*, pages 263–270. Springer Nature, Singapore, 2022.
18. Arwa Diwali, Kawther Saeedi, Kia Dashtipour, Mandar Gogate, Erik Cambria, and Amir Hussain. Sentiment analysis meets explainable artificial intelligence: A survey on explainable sentiment analysis. *IEEE Transactions on Affective Computing*, 15, 2024.
19. Chunxiao Fan, Jie Lin, Rui Mao, and Erik Cambria. Fusing pairwise modalities for emotion recognition in conversations. *Information Fusion*, 106:102306, 2024.
20. Ankita Gandhi, Kinjal Adhvaryu, Soujanya Poria, Erik Cambria, and Amir Hussain. Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions. *Information Fusion*, 91:424–444, 2023.
21. Alec Go, Richa Bhayani, and Lei Huang. Twitter Sentiment Classification using Distant Supervision. *CS224N project report, Stanford*, page 2009, 2009.
22. J Górriz and et al. Computational approaches to explainable artificial intelligence: Advances in theory, applications and trends. *Information Fusion*, 100:101945, 2023.
23. Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):1–42, 2018.
24. Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Awadalla. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*, 2023.
25. Ray Jackendoff. Toward an explanatory semantic representation. *Linguistic Inquiry*, 7(1):89–150, 1976.
26. Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

27. David Johnson, Olya Hakobyan, and Hanna Drimalla. Towards interpretability in audio and visual affective machine learning: A review. *arXiv preprint arXiv:2306.08933*, 2023.
28. Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocoń, Bartłomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, Piotr Miłkowski, Marcin Oleksy, Maciej Piasecki, Łukasz Radliński, Konrad Wojtasik, Stanisław Woźniak, and Przemysław Kazienko. Chatgpt: Jack of all trades, master of none. *Information Fusion*, 99:101861, 2023.
29. Manohar Kumar, Aisha Aijaz, Omkar Chattar, Jainendra Shukla, and Raghava Mutharaju. Opacity, transparency, and the ethics of affective computing. *IEEE Transactions on Affective Computing*, 2023.
30. Jonathan Lear. *Aristotle: The Desire to Understand*. Cambridge University Press, 1988.
31. Zheng Lian, Licai Sun, Mingyu Xu, Haiyang Sun, Ke Xu, Zhuofan Wen, Shun Chen, Bin Liu, and Jianhua Tao. Explainable multimodal emotion reasoning. *arXiv preprint arXiv:1803.08375*, 2023.
32. Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization. *Journal of Machine Learning Research*, 23:1–9, 2022.
33. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019.
34. Yukun Ma, Khanh Linh Nguyen, Frank Xing, and Erik Cambria. A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70, 2020.
35. Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. GPTEval: A survey on assessments of ChatGPT and GPT-4. In *LREC-COLING*, 2024.
36. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*, 2013.
37. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Lake Tahoe, NV, USA, 2013. Curran Associates, Inc.
38. Marvin Minsky. A framework for representing knowledge. In Patrick Winston, editor, *The psychology of computer vision*. McGraw-Hill, New York, 1975.
39. Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
40. Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.
41. Víctor Ponce-López, Baiyu Chen, Marc Oliu, Ciprian Corneanu, Albert Clapés, Isabelle Guyon, Xavier Baró, Hugo Jair Escalante, and Sergio Escalera. Chalearn lap 2016: First Round Challenge on First Impressions - Dataset and Results. In *ECCV*, pages 400–418, 2016.

42. Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. Is ChatGPT a General-Purpose Natural Language Processing Task Solver? *arXiv preprint arXiv:2302.06476*, 2023.
43. David Rumelhart and Andrew Ortony. The representation of knowledge in memory. In C Anderson, R Spiro, and W Montague, editors, *Schooling and the acquisition of knowledge*. Erlbaum, 1977.
44. Roger Schank. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3:552–631, 1972.
45. Yiqiu Shen, Laura Heacock, Jonathan Elias, Keith D Hentel, Beatriu Reig, George Shih, and Linda Moy. Chatgpt & other large language models are double-edged swords. *Radiology*, 307(2):e230163, 2023.
46. Robert Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451, 2017.
47. Yosephine Susanto, Andrew Livingstone, Bee Chin Ng, and Erik Cambria. The hourglass model revisited. *IEEE Intelligent Systems*, 35(5):96–102, 2020.
48. Anna Wierzbicka. *Semantics: Primes and Universals*. Oxford University Press, 1996.
49. Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1480–1489, 2016.
50. Tan Yue, Rui Mao, Heng Wang, Zonghai Hu, and Erik Cambria. KnowleNet: Knowledge fusion network for multimodal sarcasm detection. *Information Fusion*, 100:101921, 2023.