Neural Computing and Applications (2020) 32:17309–17320
https://doi.org/10.1007/s00521-020-05102-3

S.I. : EMERGING APPLICATIONS OF DEEP LEARNING AND SPIKING ANN

A transformer-based approach to irony and sarcasm detection

Rolandos Alexandros Potamias[1] · Georgios Siolas[2] · Andreas - Georgios Stafylopatis[2]

Received: 12 November 2019 / Accepted: 4 June 2020 / Published online: 22 June 2020

Abstract

Figurative language (FL) seems ubiquitous in all social media discussion forums and chats, posing extra challenges to sentiment analysis endeavors. Identification of FL schemas in short texts remains largely an unresolved issue in the broader field of natural language processing, mainly due to their contradictory and metaphorical meaning content. The main FL expression forms are sarcasm, irony and metaphor. In the present paper, we employ advanced deep learning methodologies to tackle the problem of identifying the aforementioned FL forms. Significantly extending our previous work (Potamias et al., in: International conference on engineering applications of neural networks, Springer, Berlin, pp 164–175, 2019), we propose a neural network methodology that builds on a recently proposed pre-trained transformer-based network architecture which is further enhanced with the employment and devise of a recurrent convolutional neural network. With this setup, data preprocessing is kept in minimum. The performance of the devised hybrid neural architecture is tested on four benchmark datasets, and contrasted with other relevant state-of-the-art methodologies and systems. Results demonstrate that the proposed methodology achieves state-of-the-art performance under all benchmark datasets, outperforming, even by a large margin, all other methodologies and published studies.

Keywords Sentiment analysis · Natural language processing · Figurative language · Sarcasm · Irony · Deep learning · Transformer networks

1 Introduction

In the networked-world era, the production of (structured or unstructured) data is increasing with most of our knowledge being created and communicated via web-based social channels [96]. Such data explosion raises the need for efficient and reliable solutions for the management, analysis and interpretation of huge data sizes. Analyzing and extracting knowledge from massive data collections is not only a big issue per se, but also challenges the data analytics state-of-the-art [103], with statistical and machine learning methodologies paving the way, and deep learning (DL) taking over and presenting highly accurate solutions [29]. Relevant applications in the field of social media cover a wide spectrum, from the categorization of major disasters [43] and the identification of suggestions [74] to inducing users' appeal to political parties [2].

The raising of computational social science [56] and mainly its social media dimension [67] challenge contemporary computational linguistics and text-analytics endeavors. The challenge concerns the advancement of text analytics methodologies toward the transformation of unstructured excerpts into some kind of structured data via the identification of special passage characteristics, such as its emotional content (e.g., anger, joy, sadness) [49]. In this context, sentiment analysis (SA) comes into play, targeting the devise and development of efficient algorithmic processes for the automatic extraction of a writer's sentiment or emotion as conveyed in text excerpts. Relevant efforts focus on tracking the sentiment polarity of single utterances, which in most cases is loaded with a lot of subjectivity and a degree of vagueness [58]. Contemporary

Rolandos Alexandros Potamias: Work performed while at National Technical University of Athens.

✉ Rolandos Alexandros Potamias
r.potamias@imperial.ac.uk

Georgios Siolas
gsiolas@islab.ntua.gr

Andreas - Georgios Stafylopatis
andreas@cs.ntua.gr

[1] Department of Computing, Imperial College London, London, UK

[2] School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece

research in the field utilizes data from social media resources (e.g., Facebook, Twitter) as well as other short text references in blogs, forums, etc. [75]. However, users of social media tend to violate common grammar and vocabulary rules and even use various figurative language forms to communicate their message. In such situations, the sentiment inclination underlying the literal content of the conveyed concept may significantly differ from its figurative context, making SA tasks even more puzzling. Evidently, single turn text lacks in detecting sentiment polarity on sarcastic and ironic expressions, as already "signified in the relevant SemEval-2014 Sentiment Analysis task 9" [83]. Moreover, lacking of facial expressions and voice tone require context-aware approaches to tackle such a challenging task and overcome its ambiguities [31]. As sentiment is the emotion behind customer engagement, SA finds its realization in automated customer-aware services, elaborating over user's emotional intensities [13]. Most of the related studies utilize single turn texts from topic-specific sources, such as Twitter, Amazon and IMDB. Handcrafted and sentiment-oriented features, indicative of emotion polarity, are utilized to represent respective excerpt cases. The formed data are then fed traditional machine learning classifiers (e.g., SVM, random forest, multilayer perceptrons) or DL techniques and respective complex neural architectures, in order to induce analytical models that are able to capture the underlying sentiment content and polarity of passages [33, 42, 84].

The linguistic phenomenon of figurative language (FL) refers to the contradiction between the literal and the non-literal meaning of an utterance [17]. Literal written language assigns 'exact' (or 'real') meaning to the used words (or phrases) without any reference to putative speech figures. In contrast, FL schemas exploit non-literal mentions that deviate from the exact concept presented by the used words and phrases. FL is rich of various linguistic phenomena like 'metonymy' reference to an entity stands for another of the same domain, a more general case of 'synonymy'; and 'metaphors' systematic interchange between entities from different abstract domains [18]. Besides the philosophical considerations, theories and debates about the exact nature of FL, findings from the neuroscience research domain present clear evidence on the presence of differentiating FL processing patterns in the human brain [6, 13, 46, 60, 95], even for woman–man attraction situations! [23], a fact that makes FL processing even more challenging and difficult to tackle. Indeed, this is the case of pragmatic FL phenomena like irony and sarcasm that main intention of in most of the cases, are characterized by an oppositeness to the literal language context. It is crucial to distinguish between the literal meaning of an expression considered as a whole from its constituents' words and phrases. As literal meaning is assumed to be invariant in all context at least in its classical conceptualization [47], it is exactly this separation of an expression from its context that permits and opens the road to computational approaches in detecting and characterizing FL utterance.

We may identify three common FL expression forms, namely irony, sarcasm and metaphor. In this paper, figurative expressions, and especially ironic or sarcastic ones, are considered as a way of indirect denial. From this point of view, the interpretation and ultimately identification of the indirect meaning involved in a passage does not entail the cancellation of the indirectly rejected message and its replacement with the intentionally implied message (as advocated in [12, 30]). On the contrary, ironic/sarcastic expressions presuppose the processing of both the indirectly rejected and the implied message so that the difference between them can be identified. This view differs from the assumption that irony and sarcasm involve only one interpretation [32, 85]. Holding that irony activates both grammatical/explicit and ironic/involved notions provides that irony will be more difficult to grasp than a non-ironic use of the same expression.

Despite that all forms of FL are well-studied linguistic phenomena [32], computational approaches fail to identify the polarity of them within a text. The influence of FL in sentiment classification emerged both on SemEval-2014 sentiment analysis task [18, 83]. Results show that natural language processing (NLP) systems effective in most other tasks see their performance drop when dealing with figurative forms of language. Thus, methods capable of detecting, separating and classifying forms of FL would be valuable building blocks for a system that could ultimately provide a full-spectrum sentiment analysis of natural language.

In the literature, we encounter some major drawbacks of previous studies and we aim to resolve with our proposed method:

- Many studies tackle figurative language by utilizing a wide range of engineered features (e.g., lexical and sentiment-based features) [21, 28, 76, 78, 79, 87] making classification frameworks not feasible.
- Several approaches search words on large dictionaries which demand large computational times and can be considered as impractical [76, 87].
- Many studies exhaustively preprocess the input texts, including stemming, tagging, emoji processing, etc., that tend to be time-consuming especially in large datasets [52, 91].
- Many approaches attempt to create datasets using social media API's to automatically collect data rather than exploiting their system on benchmark datasets, with proven quality. To this end, it is impossible to be compared and evaluated [52, 57, 91].

To tackle the aforementioned problems, we propose an end-to-end methodology containing none handcrafted engineered features or lexicon dictionaries, a preprocessing step that includes only de-capitalization and we evaluate our system on several benchmark dataset. To the best of our knowledge, this is the first time that an unsupervised pre-trained transformer method is used to capture figurative language in many of its forms.

The rest of the paper is structured as follows: In Sect. 2, we present the related work on the field of FL detection; in Sect. 3, we shortly describe the background of recent advances in natural language processing that achieve high performance in a wide range of tasks and will be used to compare performance; in Sect. 4 we present our proposed method; the results of our experiments are presented in Sect. 5; and finally, our conclusion is in Sect. 6.

2 Literature review

Although the NLP community have researched all aspects of FL independently, none of the proposed systems were evaluated on more than one type. Related work on FL detection and classification tasks could be categorized into two main categories, according to the studied task: (a) irony and sarcasm detection and (b) sentiment analysis of FL excerpts. Even if sarcasm and irony are not identical phenomena, we will present those types together, as they appear in the literature.

2.1 Irony and sarcasm detection

Recently, the detection of ironic and sarcastic meanings from respective literal ones have raised scientific interest due to the intrinsic difficulties to differentiate between them. Apart from English language, irony and sarcasm detection have been widely explored on other languages as well, such as Italian [86], Japanese [36], Spanish [68] and Greek [10]. In the review analysis that follows, we group related approaches according to the their adopted key concepts to handle FL.

2.1.1 Approaches based on unexpectedness and contradictory factors

Reyes et al. [80, 81] were the first that attempted to capture irony and sarcasm in social media. They introduced the concepts of unexpectedness and contradiction that seems to be frequent in FL expressions. The unexpectedness factor was also adopted as a key concept in other studies as well. In particular, Barbieri and Saggion [4] compared tweets with sarcastic content with other topics such as, #politics, #education, #humor. The measure of unexpectedness was calculated using the *American National Corpus Frequency Data* source as well as the morphology of tweets, using random forests (RF) and decision trees (DT) classifiers. In the same direction, Buschmeir et al. [7] considered unexpectedness as an emotional imbalance between words in the text. Ghosh et al. [26] identified sarcasm using support vector machines (SVM) using as features the identified contradictions within each tweet.

2.1.2 Content and context-based approaches

Inspired by the contradictory and unexpectedness concepts, follow-up approaches utilized features that expose information about the content of each passage including: *N*-gram patterns, acronyms and adverbs [8]; semi-supervised attributes like word frequencies [16]; statistical and semantic features [79]; and *Linguistic Inquiry and Word Count* (LIWC) dictionary along with syntactic and psycholinguistic features [77]. LIWC corpus [70] was also utilized in [28], comparing sarcastic tweets with positive and negative ones using an SVM classifier. Similarly, using several lexical resources [87], and syntactic and sentiment related features [57], the respective researchers explored differences between sarcastic and ironic expressions. Affective and structural features are also employed to predict irony with conventional machine learning classifiers (DT, SVM, naïve Bayes/NB) in [20]. In a follow-up study [21], a knowledge-based *k*-NN classifier was fed with a feature set that captures a wide range of linguistic phenomena (e.g., structural, emotional). Significant results were achieved in [91], were a combination of lexical, semantic and syntactic features passed through an SVM classifier that outperformed LSTM deep neural network approaches. Apart from local content, several approaches claimed that global context may be essential to capture FL phenomena. In particular, in [93] it is claimed that capturing previous and following comments on Reddit increases classification performance. Users' behavioral information seems to be also beneficial as it captures useful contextual information in Twitter post [78]. A novel unsupervised probabilistic modeling approach to detect irony was also introduced in [66].

2.1.3 Deep learning approaches

Although several DL methodologies, such as recurrent neural networks (RNNs), are able to capture hidden dependencies between terms within text passages and can be considered as content-based, we grouped all DL studies for readability purposes. Word embeddings, i.e., learned mappings of words to real-valued vectors [62], play a key role in the success of RNNs and other DL neural architectures that utilize pre-trained word embeddings to tackle

FL. In fact, the combination of word embeddings with convolutional neural networks (CNN), so-called CNN-LSTM units, was introduced by Kumar et al. [53] and Ghosh and Veale [25] achieving state-of-the-art performance. Attentive RNNs exhibit also good performance when matched with pre-trained Word2Vec embeddings [39], and contextual information [102]. Following the same approach, an LSTM-based intra-attention was introduced in [89] that achieved increased performance. A different approach, founded on the claim that number present significant indicators, was introduced by Dubey et al. [19]. Using an attentive CNN on a dataset with sarcastic tweets that contain numbers, showed notable results. An ensemble of a shallow classifier with lexical, pragmatic and semantic features, utilizing a bidirectional LSTM model is presented in [51]. In a subsequent study [52], the researchers engineered a soft attention LSTM model coupled with a CNN. Contextual DL approaches are also employed, utilizing pre-trained along with user embeddings structured from previous posts [1] or, personality embeddings passed through CNNs [34]. ELMo embeddings [73] are utilized in [40]. In our previous approach, we implemented an ensemble deep learning classifier (DESC) [76], capturing content and semantic information. In particular, we employed an extensive feature set of a total 44 features leveraging syntactic, demonstrative, sentiment and readability information from each text along with Tf-idf features. In addition, an attentive bidirectional LSTM model trained with GloVe pre-trained word embeddings was utilized to structure an ensemble classifier processing different text representations. DESC model performed state-of-the-art results on several FL tasks.

2.2 Sentiment analysis on figurative language

The Semantic Evaluation Workshop-2015 [24] proposed a joint task to evaluate the impact of FL in sentiment analysis on ironic, sarcastic and metaphorical tweets, with a number of submissions achieving highly performance results. The ClaC team [69] exploited four lexicons to extract attributes as well as syntactic features to identify sentiment polarity. The UPF team [3] introduced a regression classification methodology on tweet features extracted with the use of the widely utilized SentiWordNet and DepecheMood lexicons. The LLT-PolyU team [99] used semi-supervised regression and decision trees on extracted unigram and bi-gram features, coupled with features that capture potential contradictions at short distances. An SVM-based classifier on extracted *n*-gram and Tf-idf features was used by the Elirf team [27] coupled with specific lexicons such as Affin, Patter and Jeffrey 10. Finally, the LT3 team [90] used an ensemble regression and SVM semi-supervised classifier with lexical features extracted with the use of WordNet and DBpedia11.

3 The background: recent advances in natural language processing

Due to the limitations of annotated datasets and the high cost of data collection, unsupervised learning approaches tend to be an easier way toward training networks. Recently, *transfer learning* approaches, i.e., the transfer of already acquired knowledge to new conditions, are gaining attention in several domain adaptation problems [22]. In fact, pre-trained embeddings representations, such as GloVe, ElMo and USE, coupled with transfer learning architectures were introduced and managed to achieve state-of-the-art results on various NLP tasks [37]. In the current section, we summarize those methods in order to introduce our proposed transfer learning system in Sect. 5. Model specifications used for the state-of-the-art models can be found in "Appendix".

3.1 Contextual embeddings

Pre-trained word embeddings proved to increase classification performances in many NLP tasks. In particular, global vectors (GloVe) [71] and Word2Vec [63] became popular in various tasks due to their ability to capture representative semantic representations of words, trained on large amount of data. However, in various studies (e.g., [61, 72, 73]), it is argued that the actual meaning of words along with their semantics representations varies according to their context. Following this assumption, researchers in [73] present an approach that is based on the creation of pre-trained word embeddings through building a bidirectional language model, i.e., predicting next word within a sequence. The ELMo model was exhaustingly trained on 30 million sentences corpus [11], with a two-layered bidirectional LSTM architecture, aiming to predict both next and previous words, introducing the concept of contextual embeddings. The final embeddings vector is produced by a task-specific weighted sum of the two directional hidden layers of LSTM models. Another contextual approach for creating embedding vector representations is proposed in [9], where complete sentences, instead of words, are mapped to a latent vector space. The approach provides two variations of universal sentence encoder (USE) with some trade-offs in computation and accuracy. The first approach consists of a computationally intensive transformer that resembles a transformer network [92], proved to achieve higher performance figures. In contrast, the second approach provides a lightweight model that averages input embedding weights for words and bigrams by utilizing of a deep average network (DAN) [41]. The output of the DAN is passed through a feed-forward neural network in order to produce the sentence embeddings. Both approaches take as input lowercased PTB tokenized[1] strings and output a 512-dimensional sentence embedding vectors.

3.2 Transformer methods

Sequence-to-sequence (seq2seq) methods using encoder-decoder schemes are a popular choice for several tasks such as machine translation, text summarization and question answering [88]. However, encoder's contextual representations are uncertain when dealing with long-range dependencies. To address these drawbacks, Vaswani et al. [92] introduced a novel network architecture, called transformer, relying entirely on self-attention units to map input sequences to output sequences without the use of RNNs. The transformer's decoder unit architecture contains a masked multi-head attention layer, followed by a multi-head attention unit and a feed-forward network, whereas the decoder unit is almost identical without the masked attention unit. Multi-head self-attention layers are calculated in parallel facing the computational costs of regular attention layers used by previous seq2seq network architectures. In [17] the authors presented a model that is founded on findings from various previous studies (e.g., [14, 38, 73, 77, 92]), which achieved state-of-the-art results on eleven NLP tasks, called BERT—bidirectional encoder representations from transformers. The BERT training process is split into two phases: the unsupervised pre-training phase and the fine-tuning phase using labeled data for down-streaming tasks. In contrast with previous proposed models (e.g., [73, 77]), BERT uses masked language models (MLMs) to enable pre-trained deep bidirectional representations. In the pre-training phase, the model is trained with a large amount of unlabeled data from Wikipedia, BookCorpus [104] and WordPiece [98] embeddings. In this training part, the model was tested on two tasks; on the first task, the model randomly masks 15% of the input tokens aiming to capture conceptual representations of word sequences by predicting masked words inside the corpus, whereas in the second task, the model is given two sentences and tries to predict whether the second sentence is the next sentence of the first. In the second phase, BERT is extended with a task-related classifier model that is trained on a supervised manner. During this supervised phase, the pre-trained BERT model receives minimal changes, with the classifier's parameters trained in order to minimize the loss function. Two models presented in [17], a "Base Bert" model with 12 encoder layers (i.e., transformer blocks), feed-forward networks with 768 hidden units and 12 attention heads, and a "Large Bert" model with 24 encoder layers 1024 feed-the pre-trained Bert model, an architecture almost identical with the aforementioned transformer network. A [CLS] token is supplied in the input as the first token, the final hidden state of which is aggregated for classification tasks. Despite the achieved breakthroughs, the BERT model suffers from several drawbacks. Firstly, BERT, as all language models using transformers, assumes (and pre-supposes) independence between the masked words from the input sequence, and neglects all the positional and dependency information between words. In other words, for the prediction of a masked token both word and position embeddings are masked out, even if positional information is a key-aspect of NLP [15]. In addition, the [MASK] token, which is substituted with masked words, is mostly absent in fine-tuning phase for down-streaming tasks, leading to a pre-training fine-turning discrepancy. To address the cons of BERT, a permutation language model was introduced, so-called XLnet, trained to predict masked tokens in a non-sequential random order, factorizing likelihood in an autoregressive manner without the independence assumption and without relying on any input corruption [100]. In particular, a query stream is used that extends embedding representations to incorporate positional information about the masked words. The original representation set (content stream), including both token and positional embeddings, is then used as input to the query stream following a scheme called "Two-Stream SelfAttention". To overcome the problem of slow convergence, the authors propose the prediction of the last token in the permutation phase, instead of predicting the entire sequence. Finally, XLnet uses also a special token for the classification and separation of the input sequence, [CLS] and [SEP], respectively; however, it also learns an embedding that denotes whether the two words are from the same segment. This is similar to relative positional encodings introduced in TrasformerXL [15], and extents the ability of XLnet to cope with tasks that encompass arbitrary input segments. Recently, a replication study [59], suggested several modifications in the training procedure of BERT which outperforms the original XLNet architecture on several NLP tasks. The optimized model, called robustly optimized BERT approach (RoBERTa), used 10 times more data (160 GB compared with the 16 GB originally exploited), and is trained with far more epochs than the BERT model (500 K vs. 100 K), using also 8 times larger batch sizes, and a byte-level BPE vocabulary instead of the character-level vocabulary that was previously utilized. Another significant modification was the dynamic masking technique instead of the single static mask used in BERT. In addition, RoBERTa model removes the next sentence prediction

[1] https://nlp.stanford.edu/software/tokenizer.html.

objective used in BERT, following advises by several other studies that question the NSP loss term [44, 55, 101].

4 Proposed method: recurrent CNN RoBERTA (RCNN-RoBERTa)

The intuition behind our proposed RCNN-RoBERTa approach is founded on the following observation: As pre-trained networks are beneficial for several down-streaming tasks, their outputs could be further enhanced if processed properly by other networks. Toward this end, we devised an end-to-end model that utilizes pre-trained RoBERTa [59] weights combined with a RCNN in order to capture contextual information. The RoBERTa network architecture is utilized in order to efficiently map words onto a rich embedding space. To improve RoBERTa's performance and identify FL within a sentence, it is essential to capture the dependencies within RoBERTa's pre-trained word-embeddings. This task can be tackled with an RNN layer suited to capture temporal reliant information, in contrast, to fully-connected and 1D convolution layers that are not able to delineate with such dependencies. In addition, aiming to enhance the proposed network architecture, the RNN layer is followed with a fully connected layer that simulates 1D convolution with a large kernel (see below), which is capable to capture spatiotemporal dependencies in RoBERTa's projected latent space. Actually, the proposed leaning model is based on a hybrid DL neural architecture that utilizes pre-trained transformer models and feed the hidden representations of the transformer into a recurrent convolutional neural network (RCNN), similar to [54]. In particular, we employed the RoBERTa base model with 12 hidden states and 12 attention heads, and used its output hidden states as an embedding layer to a RCNN. As already stated, contradictions and long-time dependencies within a sentence may be used as strong identifiers of FL expressions. RNNs are often used to capture temporal relationships between words. However they are strongly biased, i.e., later words are tending to be more dominant that previous ones. This problem can be alleviated with CNNs, which, as unbiased models, can determine semantic relationships between words with max-pooling [54, 65]. Nevertheless, contextual information in CNNs is depended totally on kernel sizes. Thus, we appropriately modified the RCNN model presented in [54] in order to capture unbiased recurrent informative relationships within text. In particular, we implemented a bidirectional LSTM (BiLSTM) layer, which is fed with RoBERTa's final hidden layer weights. The output of LSTM is concatenated with the embedded weights, and passed through a feed-forward network, acting as a 1D convolution layer with large kernel, and a max-pooling layer. Finally, softmax function is used for the output layer. Table 1 shows the parameters used in training, and Fig. 1 illustrates the proposed deep network architecture.

5 Experimental results

To assess the performance of the proposed method, we performed an exhaustive comparison with several advanced state-of-the-art methodologies along with published results. Nowadays trends in NLP community tend to explicitly utilize deep learning methodologies as the most convenient way to approach various semantic analysis tasks. In the past decade, RNNs such as LSTM and GRUs were the most popular choice, whereas the last years the impact of attention-based models such as transformers seems to outperform all previous methods, even by a large margin [17, 92]. On the contrary, classical machine learning algorithms such as SVM, *k*-nearest neighbors (kNN) and tree-based models (decision trees, random forest) have been considered inappropriate for real-world applications, due to their demand on hand-crafted feature extraction and exhaustive preprocessing strategies. In order to have a reasonable kNN or SVM algorithm, there should be a lot of effort to embed sentences on word level to a higher space that a classifier may recognize patterns. In support of the arguments made, in our previous study [76], classical machine learning algorithms supported with rich and informative features failed to compete deep learning methodologies and proved non-feasible to FL detection. To this end, in this study we acquired several state-of-the-art models to compare our proposed method. The used methodologies were appropriately implemented using the available codes and guidelines, and include: ELMo [73],

Table 1 Selected hyperparameters used in our proposed method RCNN-RoBERTa

Hyperparameter	Value
RoBERTa layers	12
RoBERTa attention heads	12
LSTM units	64
LSTM dropout	0.1
Batch size	10
Adam epsilon	1e−6
Epochs	5
Learning rate	2e−5
Weight decay	1e−5

The hyperparameters were settled following a grid search based on a fivefold cross-validation process; the finally selected parameters are the ones that exhibit the best performance

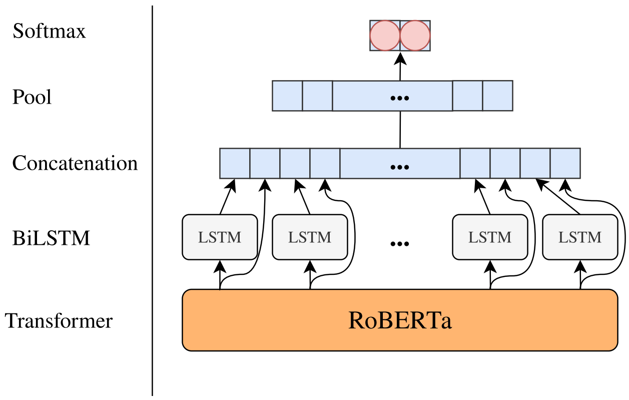

Fig. 1 The proposed RCNN-RoBERTa methodology, consisting of a RoBERTa pre-trained transformer followed by a bidirectional LSTM layer (BiLSTM). Pooling is applied to the representation vector of concatenated RoBERTa and LSTM outputs and passed through a fully connected softmax-activated layer. We refer the reader to [59, 92] for RoBERTa transformer-based architecture

Table 2 Comparison of RCNN-RoBERTa with state-of-the-art neural network classifiers and published results on SemEval-2018 dataset

Irony/SemVal-2018-Task 3.A [35]

System	Acc	Pre	Rec	$F1$	AUC
ELMo	0.66	0.66	0.67	0.66	0.72
USE	0.69	0.67	0.67	0.67	0.74
NBSVM	0.69	0.70	0.69	0.69	0.73
FastText	0.69	0.71	0.69	0.69	0.73
XLnet	0.71	0.71	0.72	0.70	0.80
BERT-Cased	0.70	0.69	0.70	0.69	0.77
BERT-Uncased	0.69	0.68	0.69	0.68	0.77
RoBERTa	0.79	0.78	0.79	0.78	0.89
Wu et al. [97]	0.74	0.63	0.80	0.71	–
Ilić et al. [40]	0.71	0.70	0.70	0.70	–
THU_NGN [97]	0.73	0.63	0.80	0.71	–
NTUA-SLP [5]	0.73	0.65	0.69	0.67	–
Zhang et al. [102]	–	–	–	0.71	–
DESC [76]	0.74	0.73	0.73	0.73	0.78
Proposed	**0.82**	**0.81**	**0.80**	**0.80**	**0.89**

Bold figures indicate superior performance

Table 3 Comparison of RCNN-RoBERTa with state-of-the-art neural network classifiers and published results on Reddit Politics dataset

Reddit SARC2.0 politics [48]

System	Acc	Pre	Rec	$F1$	AUC
ELMo	0.70	0.70	0.70	0.70	0.77
USE	0.75	0.75	0.75	0.75	0.82
NBSVM	0.65	0.65	0.65	0.65	0.68
FastText	0.63	0.65	0.61	0.63	0.64
XLnet	0.76	0.77	0.74	0.76	0.83
BERT-Cased	0.76	0.76	0.75	0.76	0.84
BERT-Uncased	0.77	0.77	0.77	0.77	0.84
RoBERTa	0.77	0.77	0.77	0.77	0.85
CASCADE [34]	0.74	–	–	0.75	–
Ilić et al. [40]	0.79	–	–	–	–
Khodak et al. [48]	0.77	–	–	–	–
Proposed	**0.79**	**0.78**	**0.78**	**0.78**	**0.85**

Bold figures indicate superior performance

USE [9], NBSVM [94], FastText [45], XLnet base cased model (XLnet) [100], BERT [17] in two setups: BERT base cased (BERT-Cased) and BERT base uncased (BERT-Uncased) models, and RoBERTa base model [59]. The settings and the hyper-parameters used for training the aforementioned models can be found in “Appendix”. The published results were acquired from the respective original publication (the reference publication is indicated in the respective tables). For the comparison we utilized benchmark datasets that include ironic, sarcastic and metaphoric expressions. Namely, we used the dataset provided in “Semantic Evaluation Workshop Task 3” (SemEval-2018) that contains ironic tweets [35]; Riloff’s high-quality sarcastic unbalanced dataset [82]; a large dataset containing political comments from Reddit [48]; and a SA dataset that contains tweets with various FL forms from “SemEval-2015 Task 11” [24]. All datasets are used in a binary classification manner (i.e., irony/sarcasm vs. literal), except from thec“SemEval-2015 Task 11” dataset where the task is to predict a sentiment integer score (from -5 to 5) for each tweet (refer to [76] for more details). For a fair comparison, we split the datasets on train/test stets as proposed by the authors providing the datasets or by following the settings of the respective published studies. The evaluation was made across standard five metrics, namely accuracy (Acc), precision (Pre), recall (Rec), $F1$-score ($F1$) and area under the receiver operating characteristics curve (AUC). For the SA task the cosine similarity metric (Cos) and mean squared error (MSE) metrics are used, as proposed in the original study [24].

The results are summarized in Tables 2, 3, 4 and 5; each table refers to the respective comparison study. All tables present the performance results of our proposed method (“Proposed”) and contrast them to eight state-of-the-art baseline methodologies along with published results using the same dataset. Specifically, Table 2 presents the results obtained using the ironic dataset used in SemEval-2018 Task 3.A, compared with recently published studies and two high performing teams from the respective SemEval shared task [5, 97]. Tables 3 and 4 summarize results obtained using Sarcastic datasets (Reddit SARC politics [48] and Riloff Twitter [82]). Finally, Table 5 compares the results from baseline models, from top two

Table 4 Comparison of RCNN-RoBERTa with state-of-the-art neural network classifiers and published results on Sarcastic Rillof's dataset

Riloff sarcastic dataset [82]					
System	Acc	Pre	Rec	*F*1	AUC
ELMo	0.85	0.85	0.86	0.85	0.89
USE	0.87	0.81	0.76	0.78	0.89
NBSVM	0.75	0.59	0.57	0.58	0.60
FastText	0.83	0.83	0.61	0.64	0.85
XLnet	0.86	0.88	0.86	0.86	0.92
BERT-Cased	0.86	0.87	0.85	0.86	0.91
BERT-Uncased	0.87	0.88	0.87	0.87	0.91
RoBERTa	0.89	0.85	0.84	0.85	0.91
Farías et al. [20]	–	–	–	0.75	–
Ilić et al. [40]	0.86	0.78	0.77	0.75	–
Tay et al. [89]	0.82	0.74	0.73	0.73	–
DESC [76]	0.87	0.86	0.87	0.87	0.86
Ghosh and Veale [25]	–	0.88	0.88	0.88	–
Proposed	**0.91**	**0.90**	**0.90**	**0.90**	**0.94**

Bold figures indicate superior performance

Table 5 Comparison of RCNN-RoBERTa with state-of-the-art neural network classifiers and published results on Task11—SemEval-2015 dataset (sentiment analysis of figurative language expression)

SemEval-2015 Task 11 [24]		
System	COS	MSE
ELMo	0.710	3.610
USE	0.71	3.17
NBSVM	0.69	3.23
FastText	0.72	2.99
XLnet	0.76	1.84
BERT-Cased	0.72	1.97
BERT-Uncased	0.79	1.54
RoBERTa	0.78	1.55
UPF [3]	0.71	2.46
ClaC [69]	0.76	2.12
DESC [76]	**0.82**	2.48
Proposed	0.81	**1.45**

Bold figures indicate superior performance

ranked task participants [3, 69], from our previous study with the DESC methodology [76] with the proposed RCNN-RoBERTa framework on a Sentiment Analysis task with figurative language, using the SemEval 2015 Task 11 dataset.

As it can be easily observed, the proposed RCNN-RoBERTa approach outperforms all approaches as well as all methods with published results, for the respective binary classification tasks (Tables 2, 3, 4). In particular, the RCNN architecture seems to reinforce RoBERTa model by 2–5% *F*1 score, increasing also the classification confidence, in terms of AUC performance. Note also that RoBERTa-RCNN show better behavior, compared to RoBERTa, on imbalanced datasets (Riloff [82], SemEval-2015 [24]). Also, one-way ANOVA Tukey test [64] revealed that RoBERTa-RCNN model outperforms by a statistical significant margin the maximum values of all metrics of previously published approaches, i.e., $p = 0.015; p < 0.05$ for ironic tweets and $p = 0.003; p < 0.01$ for Riloff sarcastic tweets. Furthermore, the proposed method increased the state-of-the-art performance even by a large margin in terms of accuracy, *F*1 and AUC score. Our previous approach, DESC (introduced in [76]), performs slightly better in terms of cosine similarity for the sentiment scoring task (Table 5, 0.820 vs. 0.810), with the RCNN-RoBERTa approach to perform better and managing to significantly improve the MSE measure by almost 33.5% (2.480 vs. 1.450).

6 Conclusion

In this study, we propose the first transformer based methodology, leveraging the pre-trained RoBERTa model combined with a recurrent convolutional neural network, to tackle figurative language in social media. Our network is compared with all, to the best of our knowledge, published approaches under four different benchmark dataset. In addition, we aim to minimize preprocessing and engineered feature extraction steps which are, as we claim, unnecessary when using overly trained deep learning methods such as transformers. In fact, handcrafted features along with preprocessing techniques such as stemming and tagging on huge datasets containing thousands of samples are almost prohibited in terms of their computation cost. Our proposed model, RCNN-RoBERTa, achieves state-of-the-art performance under six metrics over four benchmark dataset, denoting that transfer learning non-literal forms of language. Moreover, RCNN-RoBERTa model outperforms all other state-of-the-art approaches tested including BERT, XLnet, ELMo and USE under all metric, some by a large factor.

Appendix

In our experiments we compared our model with several seven different classifiers under different settings. For the ELMo system, we used the mean-pooling of all contextualized word representations, i.e., character-based embedding representations and the output of the two layer LSTM resulting with a 1024-dimensional vector, and passed it through two deep dense ReLu activated layers with 256 and 64 units. Similarly, USE embeddings are trained with a transformer encoder and output 512-dimensional vector for each sample, which is also passed through two deep dense ReLu activated layers with 256 and 64 units. Both ELMo and USE embeddings retrieved from TensorFlow Hub.[2] NBSVM system was modified according to [94] and trained with a 10^{-3} leaning rate for 5 epochs with Adam optimizer [50]. FastText system was implemented by utilizing pre-trained embeddings [45] passed through a global max-pooling and a 64 unit fully connected layer. System was trained with Adam optimizer with learning rate 0.1 for 3 epochs. XLnet model implemented using the base-cased model with 12 layers, 768 hidden units and 12 attention heads. Model trained with learning rate 4×10^{-5} using 10^{-5} weight decay for 3 epochs. We exploited both cased and uncased BERT-base models containing 12 layers, 768 hidden units and 12 attention heads. We trained models for 3 epochs with learning rate 2×10^{-5} using 10^{-5} weight decay. We trained RoBERTa model following the setting of BERT model. RoBERTa, XLnet and BERT models implemented using pytorch-transformers library[3] and were topped with two dense fully connected layers used as the output classifier.

References

1. Amir S, Wallace BC, Lyu H, Silva PCMJ (2016) Modelling context with user embeddings for sarcasm detection in social media. arXiv preprint arXiv:1607.00976
2. Antonakaki D, Spiliotopoulos D, Samaras CV, Pratikakis P, Ioannidis S, Fragopoulou P (2017) Social media analysis during political turbulence. PLoS ONE 12(10):1–23
3. Barbieri F, Ronzano F, Saggion H (2015) UPF-taln: SemEval 2015 tasks 10 and 11. Sentiment analysis of literal and figurative language in Twitter. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015). Association for Computational Linguistics, Denver, pp 704–708
4. Barbieri F, Saggion H (2014) Modelling irony in Twitter. In: EACL
5. Baziotis C, Nikolaos A, Papalampidi P, Kolovou A, Paraskevopoulos G, Ellinas N, Potamianos A (2018) NTUA-SLP at SemEval-2018 task 3: tracking ironic tweets using ensembles of word and character level attentive RNNs. In: Proceedings of the 12th international workshop on semantic evaluation. Association for Computational Linguistics, New Orleans, pp 613–621
6. Benedek M, Beaty R, Jauk E, Koschutnig K, Fink A, Silvia PJ, Dunst B, Neubauer AC (2014) Creating metaphors: the neural basis of figurative language production. NeuroImage 90:99–106
7. Buschmeier K, Cimiano P, Klinger R (2014) An impact analysis of features in a classification approach to irony detection in product reviews. In: Proceedings of the 5th workshop on computational approaches to subjectivity, sentiment and social media analysis. Association for Computational Linguistics, Baltimore, pp 42–49
8. Carvalho P (2009) Clues for detecting irony in user-generated contents: Oh...!! it's "so easy. In: International CIKM workshop on topic-sentiment analysis for mass opinion measurement, Hong Kong
9. Cer D, Yang Y, Kong SY, Hua N, Limtiaco N, John RS, Constant N, Guajardo-Cespedes M, Yuan S, Tar C et al (2018) Universal sentence encoder. arXiv preprint arXiv:1803.11175
10. Charalampakis B, Spathis D, Kouslis E, Kermanidis K (2016) A comparison between semi-supervised and supervised text mining techniques on detecting irony in greek political tweets. Eng Appl Artif Intell 51:50–57
11. Chelba C, Mikolov T, Schuster M, Ge Q, Brants T, Koehn P, Robinson T (2013) One billion word benchmark for measuring progress in statistical language modeling. arXiv preprint arXiv:1312.3005
12. Clark HH, Gerrig RJ (1984) On the pretense theory of irony. J Exp Psychol Gen 113:121–126
13. Cuccio V, Ambrosecchia M, Ferri F, Carapezza M, Piparo FL, Fogassi L, Gallese V (2014) How the context matters. Literal and figurative meaning in the embodied language paradigm. PLoS ONE 9(12):e115381
14. Dai AM, Le QV (2015) Semi-supervised sequence learning. In: Advances in Neural Information Processing Systems, pp 3079–3087
15. Dai Z, Yang Z, Yang Y, Cohen WW, Carbonell J, Le QV, Salakhutdinov R (2019) Transformer-xl: attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860
16. Davidov D, Tsur O, Rappoport A (2010) Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. In: Proceedings of the fourteenth conference on computational natural language learning, CoNLL '10. Association for Computational Linguistics, Stroudsburg, pp 107–116
17. Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American Chapter of the Association for Computational Linguistics: human language technologies, volume 1 (long and short papers). Association for Computational Linguistics, Minneapolis, pp 4171–4186
18. Dridi A, Recupero DR (2019) Leveraging semantics for sentiment polarity detection in social media. Int J Mach Learn Cybern 10(8):2045–2055
19. Dubey A, Kumar L, Somani A, Joshi A, Bhattacharyya P (2019) "When numbers matter!": detecting sarcasm in numerical portions of text. In: Proceedings of the tenth workshop on computational approaches to subjectivity, sentiment and social media analysis, pp 72–80
20. Farías DIH, Montes-y-Gómez M, Escalante HJ, Rosso P, Patti V (2018) A knowledge-based weighted KNN for detecting irony in Twitter. In: Mexican international conference on artificial intelligence. Springer, Berlin, pp 194–206
21. Farías DIH, Patti V, Rosso P (2016) Irony detection in Twitter: the role of affective content. ACM Trans Internet Technol (TOIT) 16(3):19

[2] https://tfhub.dev/s?module-type=text-embedding.

[3] https://huggingface.co/transformers/.

22. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016) Domain-adversarial training of neural networks. J Mach Learn Res 17(1):2096–2030
23. Gao Z, Gao S, Xu L, Zheng X, Ma X, Luo L, Kendrick KM (2017) Women prefer men who use metaphorical language when paying compliments in a romantic context. Sci Rep 7:40871
24. Ghosh A, Li G, Veale T, Rosso P, Shutova E, Barnden J, Reyes A (2015) SemEval-2015 task 11: sentiment analysis of figurative language in Twitter. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015). Association for Computational Linguistics, Denver, pp 470–478
25. Ghosh A, Veale T (2016) Fracking sarcasm using neural network. In: Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis, pp 161–169
26. Ghosh D, Guo W, Muresan S (2015) Sarcastic or not: word embeddings to predict the literal or sarcastic meaning of words. In: EMNLP
27. Giménez M, Pla F, Hurtado LF (2015) ELiRF: a SVM approach for SA tasks in Twitter at SemEval-2015. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015). Association for Computational Linguistics, Denver, pp 574–581
28. Gonzáilez-Ibáñez RI, Muresan S, Wacholder N (2011) Identifying sarcasm in Twitter: a closer look. In: ACL
29. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
30. Grice HP (2008) Further notes on logic and conversation. In: Adler JE, Rips LJ (eds) Reasoning: studies of human inference and its foundations. Cambridge University Press, Cambridge, pp 765–773
31. Gupta U, Chatterjee A, Srikanth R, Agrawal P (2017) A sentiment-and-semantics-based approach for emotion detection in textual conversations
32. Gibbs RW (1986) On the psycholinguistics of sarcasm. J Exp Psychol Gen 115:3–15
33. Hangya V, Farkas R (2017) A comparative empirical study on social media sentiment analysis over various genres and languages. Artif Intell Rev 47(4):485–505
34. Hazarika D, Poria S, Gorantla S, Cambria E, Zimmermann R, Mihalcea R (2018) Cascade: contextual sarcasm detection in online discussion forums. arXiv preprint arXiv:1805.06413
35. Hee CV, Lefever E, Hoste V (2018) SemEval-2018 task 3: irony detection in English tweets. In: SemEval@NAACL-HLT
36. Hiai S, Shimada K (2018) Sarcasm detection using features based on indicator and roles. In: International conference on soft computing and data mining. Springer, Berlin, pp 418–428
37. Howard J, Ruder S (2018) Universal language model fine-tuning for text classification. In: Proceedings of the 56th annual meeting of the Association for Computational Linguistics (volume 1: long papers). Association for Computational Linguistics, Melbourne, pp 328–339
38. Howard J, Ruder S (2018) Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146
39. Huang YH, Huang HH, Chen HH (2017) Irony detection with attentive recurrent neural networks. In: ECIR
40. Ilić S, Marrese-Taylor E, Balazs JA, Matsuo Y (2018) Deep contextualized word representations for detecting sarcasm and irony. arXiv preprint arXiv:1809.09795
41. Iyyer M, Manjunatha V, Boyd-Graber J, Daumé III H (2015) Deep unordered composition rivals syntactic methods for text classification. In: Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 1: long papers). Association for Computational Linguistics, Beijing, pp 1681–1691
42. Jianqiang Z, Xiaolin G, Xuejun Z (2018) Deep convolution neural networks for Twitter sentiment analysis. IEEE Access 6:23253–23260
43. Joseph JK, Dev KA, Pradeepkumar AP, Mohan M (2018) Chapter 16—Big data analytics and social media in disaster management. In: Samui P, Kim D, Ghosh CBTIDS (eds) Integrating disaster science and management. Elsevier, Amsterdam, pp 287–294
44. Joshi M, Chen D, Liu Y, Weld DS, Zettlemoyer L, Levy O (2019) Spanbert: improving pre-training by representing and predicting spans. arXiv preprint arXiv:1907.10529
45. Joulin A, Grave E, Bojanowski P, Douze M, Jégou H, Mikolov T (2016) Fasttext. zip: compressing text classification models. arXiv preprint arXiv:1612.03651
46. Kasparian K (2013) Hemispheric differences in figurative language processing: contributions of neuroimaging methods and challenges in reconciling current empirical findings. J Neuroling 26:1–21
47. Katz JJ (1977) Propositional structure and illocutionary force: a study of the contribution of sentence meaning to speech acts/ Jerrold J. Katz. The Language and thought series. Crowell, New York
48. Khodak M, Saunshi N, Vodrahalli K (2017) A large self-annotated corpus for sarcasm. arXiv e-prints
49. Kim E, Klinger R (2018) A survey on sentiment and emotion analysis for computational literary studies
50. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv e-prints
51. Kumar A, Garg G (2019) Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets. J Ambient Intell Humaniz Comput 1–16
52. Kumar A, Sangwan SR, Arora A, Nayyar A, Abdel-Basset M et al (2019) Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. IEEE Access 7:23319–23328
53. Kumar L, Somani A, Bhattacharyya P (2017) "Having 2 hours to write a paper is fun!": detecting sarcasm in numerical portions of text. arXiv e-prints
54. Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. In: Twenty-ninth AAAI conference on artificial intelligence
55. Lample G, Conneau A (2019) Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291
56. Lazer D, Pentland A, Adamic L, Aral S, Barabasi AL, Brewer D, Christakis N, Contractor N, Fowler J, Gutmann M, Jebara T, King G, Macy M, Roy D, Van Alstyne M (2009) Life in the network: the coming age of computational social science. Science (New York, N. Y.) 323(5915):721–723
57. Ling J, Klinger R (2016) An empirical, quantitative analysis of the differences between sarcasm and irony. In: European semantic web conference. Springer, Berlin, pp 203–216
58. Liu B (2015) Sentiment analysis—mining opinions, sentiments, and emotions. Cambridge University Press, Cambridge
59. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692
60. Loenneker-Rodman B, Narayanan S (2010) Computational approaches to figurative language. Cambridge Encyclopedia of Psycholinguistics. Cambridge University Press, Cambridge
61. McCann B, Bradbury J, Xiong C, Socher R (2017) Learned in translation: Contextualized word vectors. In: Advances in Neural Information Processing Systems, pp 6294–6305

62. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv e-prints
63. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. arXiv e-prints
64. Montgomery DC (2017) Design and analysis of experiments, 9th edn. Wiley, New York
65. Nguyen TH, Grishman R (2015) Relation extraction: perspective from convolutional neural networks. In: Proceedings of the 1st workshop on vector space modeling for natural language processing. Association for Computational Linguistics, Denver, pp 39–48
66. Nozza D, Fersini E, Messina E (2016) Unsupervised irony detection: a probabilistic model with word embeddings. In: KDIR, pp 68–76
67. Oboler A, Welsh K, Cruz L (2012) The danger of big data: social media as computational social science. First Monday 17(7)
68. Ortega-Bueno R, Rangel F, Hernández Farıas D, Rosso P, Montes-y-Gómez M, Medina Pagola JE (2019) Overview of the task on irony detection in Spanish variants. In: Proceedings of the Iberian languages evaluation forum (IberLEF 2019), co-located with 34th conference of the Spanish Society for natural language processing (SEPLN 2019). CEUR-WS.org
69. Özdemir C, Bergler S (2015) CLaC-SentiPipe: SemEval2015 subtasks 10 B, E, and task 11. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015). Association for Computational Linguistics, Denver, pp 479–485
70. Pennebaker J, Francis M (1999) Linguistic inquiry and word count. Lawrence Erlbaum Associates, Incorporated, Mahwah
71. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. EMNLP 14:1532–1543
72. Peters ME, Ammar W, Bhagavatula C, Power R (2017) Semi-supervised sequence tagging with bidirectional language models. arXiv preprint arXiv:1705.00108
73. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. arXiv preprint arXiv:1802.05365
74. Potamias RA, Neofytou A, Siolas G (2019) NTUA-ISLab at SemEval-2019 task 9: mining suggestions in the wild. In: Proceedings of the 13th international workshop on semantic evaluation. Association for Computational Linguistics, Minneapolis, pp 1224–1230
75. Potamias RA, Siolas G (2019) NTUA-ISLab at SemEval-2019 task 3: determining emotions in contextual conversations with deep learning. In: Proceedings of the 13th international workshop on semantic evaluation. Association for Computational Linguistics, Minneapolis, pp 277–281
76. Potamias RA, Siolas G, Stafylopatis A (2019) A robust deep ensemble classifier for figurative language detection. In: International conference on engineering applications of neural networks. Springer, Berlin, pp 164–175
77. Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training
78. Rajadesingan A, Zafarani R, Liu H (2015) Sarcasm detection on Twitter: a behavioral modeling approach. In: WSDM
79. Ravi K, Ravi V (2017) A novel automatic satire and irony detection using ensembled feature selection and data mining. Knowl Based Syst 120:15–33
80. Reyes A, Rosso P, Buscaldi D (2012) From humor recognition to irony detection: the figurative language of social media. Data Knowl Eng 74:1–12
81. Reyes A, Rosso P, Veale T (2013) A multidimensional approach for detecting irony in Twitter. Lang Resour Eval 47(1):239–268
82. Riloff E, Qadir A, Surve P, De Silva L, Gilbert N, Huang R (2013) Sarcasm as contrast between a positive sentiment and negative situation. In: EMNLP 2013—2013 conference on empirical methods in natural language processing, proceedings of the conference. Association for Computational Linguistics (ACL), pp 704–714
83. Rosenthal S, Ritter A, Nakov P, Stoyanov V (2014) SemEval-2014 task 9: sentiment analysis in Twitter. In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014). Association for Computational Linguistics, Dublin, pp 73–80
84. Singh NK, Tomar DS, Sangaiah AK (2020) Sentiment analysis: a review and comparative analysis over social media. J Ambient Intell Human Comput 11:97–117
85. Sperber D, Wilson D (1981) Irony and the use-mention distinction. In: Cole P (ed) Radical pragmatics. Academic Press, New York, pp 295–318
86. Stranisci M, Bosco C, Farias H, Irazu D, Patti V (2016) Annotating sentiment and irony in the online italian political debate on #labuonascuola. In: Tenth international conference on language resources and evaluation LREC 2016. ELRA, pp 2892–2899
87. Sulis E, Farías DIH, Rosso P, Patti V, Ruffo G (2016) Figurative messages and affect in Twitter: differences between #irony, #sarcasm and #not. Knowl Based Syst 108:132–143
88. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Advances in neural information processing systems, pp 3104–3112
89. Tay Y, Luu AT, Hui SC, Su J (2018) Reasoning with sarcasm by reading in-between. In: Proceedings of the 56th annual meeting of the Association for Computational Linguistics (volume 1: long papers). Association for Computational Linguistics, Melbourne, pp 1010–1020
90. Van Hee C, Lefever E, Hoste V (2015) LT3: sentiment analysis of figurative tweets—piece of cake #notreally. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015). Association for Computational Linguistics, Denver, pp 684–688
91. Van Hee C, Lefever E, Hoste V (2018) Exploring the fine-grained analysis and automatic detection of irony on Twitter. Lang Resour Eval 52(3):707–731
92. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
93. Wallace BC, Choe DK, Charniak E (2015) Sparse, contextually informed models for irony detection: exploiting user communities, entities and sentiment. In: ACL-IJCNLP 2015—53rd annual meeting of the Association for Computational Linguistics (ACL), proceedings of the conference, vol 1
94. Wang S, Manning CD (2012) Baselines and bigrams: simple, good sentiment and topic classification. In: Proceedings of the 50th annual meeting of the association for computational linguistics: short papers, vol 2. Association for Computational Linguistics, pp 90–94
95. Weiland H, Bambini V, Schumacher PB (2014) The role of literal meaning in figurative language comprehension: evidence from masked priming ERP. Front Hum Neurosci 8:583
96. Winbey JP (2019) The social fact. The MIT Press, Cambridge
97. Wu C, Wu F, Wu S, Liu J, Yuan Z, Huang Y (2018) THU_ngn at SemEval-2018 task 3: tweet irony detection with densely connected LSTM and multi-task learning. In: SemEval@NAACL-HLT
98. Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K et al (2016) Google's neural machine translation system: bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144

99. Xu H, Santus E, Laszlo A, Huang CR (2015) LLT-PolyU: identifying sentiment intensity in ironic tweets. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015). Association for Computational Linguistics, Denver, pp 673–678
100. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV (2019) Xlnet: generalized autoregressive pretraining for language understanding. arXiv preprint arXiv:1906.08237
101. You Y, Li J, Hseu J, Song X, Demmel J, Hsieh CJ (2019) Reducing bert pre-training time from 3 days to 76 min. arXiv preprint arXiv:1904.00962
102. Zhang S, Zhang X, Chan J, Rosso P (2019) Irony detection via sentiment-based transfer learning. Inf Process Manag 56(5):1633–1644
103. Zhou L, Pan S, Wang J, Vasilakos AV (2017) Machine learning on big data: opportunities and challenges. Neurocomputing 237:350–361
104. Zhu Y, Kiros R, Zemel R, Salakhutdinov R, Urtasun R, Torralba A, Fidler S (2015) Aligning books and movies: towards story-like visual explanations by watching movies and reading books. In: Proceedings of the IEEE international conference on computer vision, pp 19–27

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.