Deep and Dense Sarcasm Detection

A Preprint

Devin Pelser
The School of Mathematics, Statistics and Computer Science
University of KwaZulu-Natal
South Africa
215023955@stu.ukzn.ac.za

Hugh Murrell
The School of Mathematics, Statistics and Computer Science
University of KwaZulu-Natal
South Africa
murrellh@ukzn.ac.za

Abstract

Recent work in automated sarcasm detection has placed a heavy focus on context and meta-data. Whilst certain utterances indeed require background knowledge and commonsense reasoning, previous works have only explored shallow models for capturing the lexical, syntactic and semantic cues present within a text. In this paper, we propose a deep 56 layer network, implemented with dense connectivity to model the isolated utterance and extract richer features therein. We compare our approach against recent state-of-the-art architectures which make considerable use of extrinsic information, and demonstrate competitive results whilst using only the local features of the text. Further, we provide an analysis of the dependency of prior convolution outputs in generating the final feature maps. Finally a case study is presented, supporting that our approach accurately classifies additional uses of clear sarcasm, which a standard CNN misclassifies.

***K*eywords** sarcasm · dense connectivity · low-level features

1 Introduction

Sarcasm is a complex linguistic phenomenon in which the intended meaning of an utterance is not the same as its literal meaning [1]. Sarcasm's figurative and nuanced nature presents a challenging task within opinion mining and sentiment analysis [2]. With the rise of social media use, and the need for understanding comments therein, affective computing has gained an increase in popularity. Existing systems for summarising reviews, or monitoring brand sentiment, often fail to detect the implicit meaning behind sarcastic remarks [3] and consequently mislead the analysis of sarcastic phrases, with interpretations often being taken as literal. Consider the sentence "*Great! I love waking up sick!*". We are easily able to recognise the sarcasm due to the presence of strong polarity shifts and common-sense; no one *loves* waking up sick. The literal meaning of the sentence has been discounted, with the speaker expecting the listener to understand the implied intent. This, however, is not as easily identified within the context of machine learning due to the overall positive sentiment formed by the words 'Great!' and 'love'. Whilst certain sarcastic utterances are easily identified based solely on lexical and pragmatic cues dependencyring within, the need for extrinsic information has been identified in prior works [4]. In the realms of social media, comments are often in response to a previous comment or in reference to a world event; suggesting context plays a key role in sarcasm detection. Further, the commonality of informal language and slang has been shown to diminish the reliance of grammatical hints [5]. With past research finding difficulty in correctly classifying these particular phrases [6].

The key aim of this work is to explore whether, perhaps, additional local cues could be extracted from the isolated utterance. In particular, we propose a deep neural network implemented with dense connectivity to build rich feature maps purely from the linguistic structure of the text. We do not seek to outperform the state-of-the-art, but rather, to determine if our model is able to rival past approaches, all of which make extensive use of both context and meta-data (such as user profiling to determine an author's sarcastic tendencies). This considerable usage of external information presents an issue for deploying real-world systems. Modelling the trends and opinions of each user, prior to categorising their comments, may not always be possible given privacy settings, or a lack of data [7]. Additionally it introduces a large overhead, and places the focus on generating detailed profiles of users or forums, rather than the sarcastic characteristics of the utterance. Hence our work focuses solely on the language facet of sarcasm. Our approach uses multiple convolution layers - with direct connections between each - to facilitate the use of both low-level, simple features, as well as complex hierarchical ones, in order to develop a deeper understanding of the sarcastic text. We hypothesise that these diverse feature-maps would give further understanding of the sarcastic intent, or lack thereof, within a given phrase. Through empirical evaluations, we demonstrate our approach yields competitive results against existing models incorporating extrinsic material. We provide a demonstration of the model[1], as well as all datasets and source code using the NextJournal platform, readers may run the demonstration by creating an account and clicking 'Remix'. In summary, the overall contributions of this work are as follows:

- Proposing a novel deep and dense sarcasm detection system to model the isolated utterance for classification.
- Examining the role of low level features in enhancing the accuracy of sarcasm classification.
- Presenting benchmark results on a new formal sarcasm dataset in the form of on ablation study.

2 Related Work

Whilst computational sarcasm is still a relatively new field - receiving an increased focus from researchers in recent times due to the massive growth of social media and the need for sentiment analysis therein - various approaches have been presented and examined. Tepperman et al. [8] studied detection within speech through the use of prosodic, spectral (average pitch of utterance, duration of utterance etc.) and contextual cues (gender, laughter etc.). Carvalho et al. [9] analysed comments on a Portuguese news site and found certain linguistic features were indicative of irony; such as emoticons, excessive punctuation and quotation marks. This statistical approach based on surface-level features has been further studied, with González-Ibàñez et al. [10] exploring unigrams, dictionary-based lexical features and pragmatic features (punctuation, emojis etc.). Liebrecht et al. [11] extended the idea of n-grams as features through the use of unigrams, bigrams and trigrams, together with intensifiers to classify Dutch tweets. Given that a sarcastic utterance often implies the opposite to what is said, sentiment and semantic incongruity as a feature has also been investigated: Riloff et al. [12] studied the presence of positive sentiment co-occurring with negative situational phrases and Buschmeier et al. [13] used the polarity between written words and the star rating on Amazon reviews. To build large self-annotated corpora Twitter API has been used to scrape comments containing tokens implying sarcastic intent (e.g. #sarcasm) [14]. This approach was followed by Ptácek et al. [15] with a feature set consisting of skipgrams and character n-grams to classify Czech and English tweets. The need for context was investigated by Wallace et al. [16], showing humans battle to infer ironic intent without it. The inclusion of extrinsic meta-knowledge has been increasingly used, with Bamman and Smith [17] making extensive use of contextual information regarding the author and audience. While Ghaeini et al. [7] reported competitive results by incorporating context together with Bi-LSTMs and attention mechanisms. A manually annotated Reddit dataset was analysed by Wallace et al. [18], proposing features based on subreddit, sentiment, named entities and interactions between these. Automatic feature extraction through convolutions within NLP domains was first proposed by Collobert and Weston [19], demonstrating it's effectiveness across multiple tasks. Amir et al. [4] extended this to sarcasm to mitigate the effort of handcrafting features; implementing a CNN to learn user embeddings based on an author's prior texts. Further deep learning methods have been reported with Poria et al. [6] yielding state-of-the-art results on a ensemble of an SVM preceded by four CNNs, three of which were pre-trained to extract emotion, sentiment and personality respectively. Combining a CNN, with an LSTM and DNN was also shown to improve performance against a recursive SVM [20]. In recent times, Hazarika et al. [21] extensively explored meta-data through profiling discussion forums based on all previously written comments, as well as capturing both stylometric and personality traits of the author through user embeddings.

Within the field of computer vision, the application of residual learning [22] resulted in extremely deep networks which led to state-of-the-art results on multiple image datasets. This notion was further enhanced through dense connectivity by Huang et al. [23], showing that the dense variants outperformed the residual counterparts. Schwenk et al. [24] researched the application of deep residual archetypes to NLP domains, correlating an increase in depth with performance gain across numerous tasks such as; news categorisation, ontology classification and sentiment analysis.

In this paper, deep densely connected networks for sarcasm classification are studied; noting that prior approaches make use of shallower models. Previous results indicate the necessity of contextual information and the insufficiency of

[1] https://nextjournal.com/Anon-Dem/dwenet-56

purely textual cues. However, no contextual information is used in this work, with the primary research goal to ascertain whether our deeper network is able to extract richer sarcastic patterns intrinsic to the text. Furthermore, a formal dataset - free of noisy labels and colloquial language - is used to conduct analysis of the proposed approach compared to a CNN.

3 Method

Inspired by the promising results of deep residual based archetypes on multiple NLP domains [24], together with the superior performance of dense connectivity within the field of computer vision; a deep, densely connected model is proposed for sarcasm classification. We hypothesise that the low-level features extracted through initial convolutions, in conjunction with abstracted hierarchical representations formed deeper in the network, will provide a richer understanding of the lexical, syntactic and semantic cues present within an isolated utterance. Through empirical results we demonstrate that our deep model is capable of rivalling the performance of more complex networks which benefit from the use of contextual and extrinsic information.

3.1 Dense Connectivity

The use of shortcut connections dependencyring between convolution layers was first proposed by Het et al [22], facilitated by the element-wise addition of input and output tensors. While this was shown to be necessary in constructing very deep networks, the continual convolutions of low-level features resulted in distorted, high-level representations forming the final feature map. To allow the network to potentially use these low-level features, Huang et al. [23] proposed the concatenation of prior convolution layer outputs to all sequential inputs; effectively allowing later layers to incorporate simple features in the production of their feature maps. We note that a convolution layer is defined as a Conv, BatchNorm, ReLU sequence.

Multiple densely connected convolution layers (dense-layers), with a filter size of 3, are stacked to form a dense-block. Each dense-layer within contributes k channels to the overall block output, defined as the growth rate. The aforementioned concatenation consequently results in the input of consecutive dense-layers having k additional channels. Hence, a dense-block of n input channels, with i dense-layers, will produce $n + (i \times k)$ outputs, noting that the signal dimension remains unchanged throughout. Downsampling is facilitated through the transition layers dependencyring between each block; halving the channel and signal dimension through convolutions of size 1 and average pooling with kernel size 2 respectively.

3.2 Proposed Architecture

The design of our network is shown in Fig. 1. The input text $W = [w_1, w_2, ..., w_s]$ is initially passed to an embedding, responsible for generating vector representations, $\vec{u_j} \in \mathbb{R}^d$, for each word $w_j \in W$, where d is the embedding dimension. Resulting in a single channel output tensor $[1, s, d]$, where s is the number of tokens in input text W. An initial convolution layer follows, producing 64 channels by convolving along 3 embedding vectors at a time, effectively performing a 1-dimensional convolution through a 2-dimensional kernel which spans the entire embedding dimension (which has been padded once on both sides) - i.e a kernel of size $3 \times (d + 2)$. This yields an output tensor, $[64, s, 1]$, formed from the 3-gram representations of the input text. Four dense-blocks follow - outputting C$_i$ channels - with transition layers in between responsible for halving the signal and channel count. Consequently resulting in 4 hierarchical levels

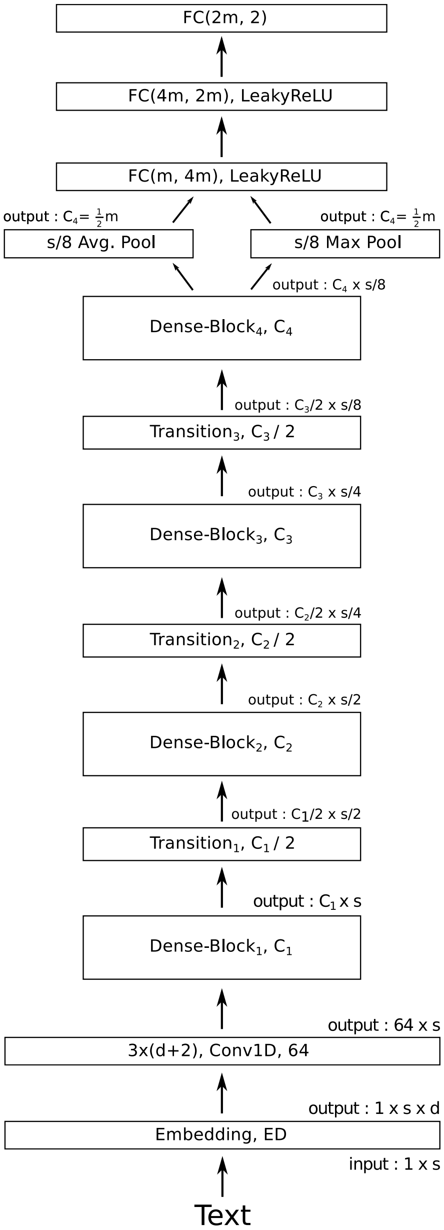

Figure 1: Overall network architecture.

generating feature maps from complex combinations of the resulting signal, $s_i = \frac{s}{2^{i-1}}$, where i is the i^{th} dense-block. The final feature maps undergo a combination of both max and average pooling over the resultant signal. Ouputs are concatenated - yielding m features - before being passing to three linear layers with leaky ReLU and dropout in between. Finally, classification dependencys through the softmax function at the head of the network. The final model was implemented with a growth rate of 32 and dense-blocks of size 6, 12, 24 and 16 respectively.

4 Experimental Results

The proposed approach is tested on the binary classification of sarcastic text over several datasets. A formal dataset is used to conduct an analysis of the model compared to a CNN; in order to investigate the effect of dense connectivity and demonstrate the use of low-level features. Finally, the network is compared against several recent models, each incorporating contextual and meta-knowledge, on a dataset designed for context investigations. Thus, we demonstrate competitive results using only the lexical details within the isolated utterance.

4.1 Datasets

The Kaggle dataset *Newsheadlines for Sarcasm Detection*[2], was constructed by collecting headlines from two news websites: TheOnion[3] and HuffPost[4]. The former is well known for producing satirical adaptations of world news and comprises the sarcastic entries, whilst the latter is a reputable news site and the headlines form the nonsarcastic entries. Consequently the resulting data is free of spelling mistakes, informal language usage and noisy labels; that is, the dataset can be considered as *formal*. Furthermore, as opposed to datasets from Twitter and Reddit, the headlines are self-contained, they are not sent in response to a prior post or discussion. This allows us to focus entirely on the sarcastic characteristics without local context needing consideration. We analyse the performance of the proposed dense model in comparison to a baseline CNN to determine whether additional intrinsic features are indeed being detected. An outline of the dataset is seen in Table 1.

To evaluate the proposed approach against models incorporating extrinsic information, the self-annotated corpus for sarcasm - SARC[5] (V2.0) - presented by Khodak et al. [25], is used. This dataset was designed for contextual investigations, with related works making considerable usage of said context. The dataset was constructed by scraping Reddit comments; with sarcastic entries being self-annotated by authors through the use of the \s token, which indicates sarcastic intent on the website. Posts on Reddit are often in response to another comment; SARC incorporates this information through the addition of the parent comment and further child comments surrounding a post. Additional details about the author and which subreddit the post appeared on are also provided. We make use of only the original comments in the datasets, discarding the parent and additional child comments. Two variants of this dataset are considered for benchmarking: Main balanced and Political balanced - the latter consisting of comments obtained only from the political subreddit. Details of both datasets are presented in Table 1.

	Training		Testing	
	non-sarc	*sarc*	*non-sarc*	*sarc*
Newsheadlines	11988	9379	2997	2345
SARC Main Balanced	104209	109713	26173	27520
SARC Pol Balanced	6834	6834	1703	1703

Table 1: Dataset statistics for Headlines and SARC

4.2 Training and Testing Details

Pre-trained GloVe[6] vectors of dimension 50 were found to be optimal and are used to initialise the non-static embedding layer. That is the word representations are updated during training to better align them for sarcasm detection. A batch size of 64 with a learning rate of 1e-03, together with the one cycle learning policy introduced by Smith [26] is adopted. A weight decay of 1e-02 is used throughout the investigations - with dropout at a rate of 0.2 added to the final 56 layer dense network for additional regularisation. Weights are intialised using the method outlined by He et al. [27] and are updated through minimisation of the log-loss with the Adam optimiser [28] having momentum range between 0.8 and

[2] https://www.kaggle.com/rmisra/news-headlines-dataset-for-sarcasm-detection

[3] https://www.theonion.com

[4] https://www.huffpost.com/

[5] http://nlp.cs.princeton.edu/SARC/

[6] https://nlp.stanford.edu/projects/glove/

0.7. Furthermore, input texts are padded to a size of 64 for the Headlines dataset, whilst both Reddit datasets are padded to a size of 128 to facilitate the batched-learning within the convolutional layers. Texts larger than the indicated lengths are removed from the dataset. It must be noted that the padding token used has constant zero entries as its embedding and gradients are not tracked. All experiments are run 20 times with results presented as the average. The proposed model and investigations are implemented using Pytorch and the FastAI framework.

4.3 Baselines

Prior state-of-the-art approaches - for both Reddit datasets - which are compared against our work, are detailed below.

- **Bag-of-words:** An SVM network which receives the word-counts of the text as a vector of the size V, where V is the vocabulary size.
- **CNN:** A simple CNN with 3 different filter sizes to extract n-gram features, as detailed by Hazarika et al. [21].
- **CNN-SVM:** An ensemble of 4 CNN's in which 3 are pretrained to extract sentiment, emotion and personality features from the given comment proposed by Poria et al. [6]. The outputs are concatenated and passed to an SVM for the final classification.
- **CUE-CNN:** Proposed by Amir et al. [4], user embeddings are modelled to obtain stylometric features which are then combined with a CNN.
- **CASCADE:** Stylometric and personality features of the authors are modelled and fused using canonical correlation analysis to obtain comprehensive user embeddings. Further, for each forum, all previous comments are combined to obtain discourse representations surrounding the topics and patterns therein. This approach was proposed by Hazarika et al. [21] and holds the record results on both datasets.
- **CASCADE** (no personality features) **:** The above CASCADE model where user embeddings are generated without personality features.
- **AMR:** An RNN-based model incorporating BiLSTMs to model input comments and responses thereof. Attention mechanisms, projection and re-reading are also used to provide deeper representations [7].

4.4 Results

Here, we present the results of our model, dweNet, compared to that of other state-of-the-art approaches. Table 2 presents the yielded results on the two SARC datasets. We can see that all proposed approaches - including ours - outperform the simple BOW and CNN baselines. Entries written in blue represent results which are rivaled by our network. Each model - except the full CASCADE implementation - is challenged by our approach; this is interesting considering our method makes no use of context, nor meta-data. The full CASCADE model makes the most use of these extrinsic features; supporting the need for background information. However, the competitive results obtained by our model suggests that additional information is indeed available in the isolated utterance, allowing it to outperform a network which made use of pretrained models extracting emotion, sentiment and personality features.

	Main		Pol	
	Accuracy	*F1*	*Accuracy*	*F1*
Bag-of-words	0.63	0.64	0.59	0.60
CNN	0.65	0.66	0.62	0.63
CNN-SVM [6]	0.68	0.68	0.65	0.67
CUE-CNN [4]	0.70	0.69	0.69	0.70
CASCADE [21]	0.77	0.77	0.74	0.75
CASCADE (no personality features)	0.68	0.66	0.68	0.70
AMR [7]	0.68	0.70	-	-
dweNet	0.69	0.69	0.69	0.69

Table 2: Comparison of our approach, dweNet, with state-of-the-art models and simple baselines on two versions of the SARC dataset. Blue entries indicate results rivaled by our network and dashed lines where no results were reported.

4.5 Ablation Study

Experiments surrounding architectural designs are conducted on the proposed dense model to evaluate various features. Table 3 details the results of all structural changes investigated on the Headlines dataset. Residual archetypes marginally outperformed the base CNN but resulted in lower accuracies and increased parameter counts when compared to the dense variant. This suggests low-level feature reuse is indeed significant, given that a ResNet is unable to take advantage of this. Further, depth was found to be beneficial in classification accuracy, resulting in an increase of 1.74%. This, however, was expected based on promising depth investigations across several NLP tasks conducted by Schwenk et al. [24]; similar results were observed when increasing the growth rate. Both FastText[7][29] embeddings - standard and subword - were found to be suboptimal, with the 50D Glove representations increasing testing accuracy by over 2%. Keeping the pretrained Glove embeddings static resulted in a notable drop of 2.99%, which can be attributed to the nuanced nature of sarcasm. Pretrained embeddings are trained on large-scale typical language datasets, which will rarely include sarcastic phrases, resulting in word embeddings which do not align with the semantics of sarcasm.

	Headlines
	Accuracy
8 Layer CNN	83.88
8 Layer ResNet	84.21
8 Layer DenseNet	85.55
8 Layer DenseNet	85.55
28 Layer DenseNet	87.29
8 Layer DenseNet GR = 4	85.55
8 Layer DenseNet GR = 32	86.85
dweNet FastText-1M 300D	86.51
dweNet FastText-1M-subword 300D	86.15
dweNet GLoVe 50 static	85.68
dweNet GLoVe 50 non-static	88.67

Table 3: Comparison of our proposed approach, the final entry, to structural variants.

4.6 Low-Level Feature Use

A similar approach as Huang et al. [23] is taken to support the notion that low-level features play a key role in increasing sarcasm detection when in combination with abstract hierarchical features. A 16 Layer DenseNet with block sizes (4, 4, 4, 4) and a growth rate of 4 is trained on the Headline dataset. The $L1$ norm of the weights connecting the preceding layers to the final layer in each block is taken in order to determine the dependency of the final layer with those prior to it. A detailed heatmap is shown in Fig 2 to visualise the intensity of this dependency for the final layer in the final block. From this we see the network assigns relatively strong importance to the input planes, indicating low-level features are indeed being used to form the final feature map. Furthermore the weights connecting the preceding 3 layers also exhibit high values, suggesting the abstract features also influence classification. The yellow blocks indicate features with little precedence and can be seen scattered throughout the heatmap; hinting that the features extracted by these layers are of little significance, or may be contained within another feature map.

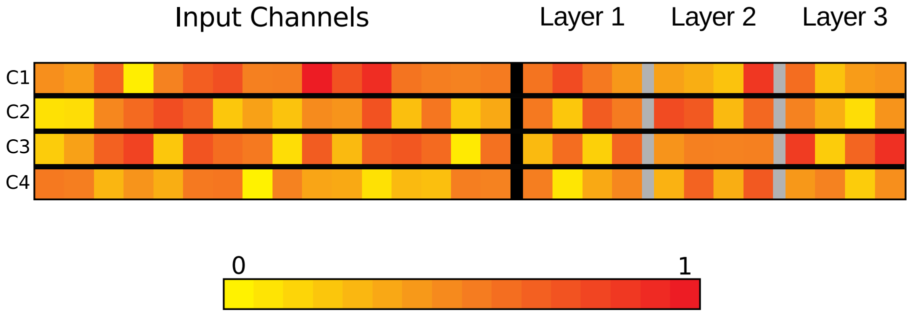

Figure 2: The average absolute filter weights of the convolution layers connected to the final layer in the fourth denseblock, channel by channel (C1 to C4). The colour encodes the normalised magnitude of these weights. The input feature maps are placed behind the vertical black bar, representing low-level features found earlier on in the network. Layer 1, 2 and 3 - separated by the grey rectangles - represent the initial three layers in the block.

4.7 Case Study

Results demonstrate that dense connectivity indeed allowed our model to rival related works which made extensive use of contextual information and meta-data, such as an author's post history and trends within a specific subforum. The 8

[7] https://fasttext.cc/

layer CNN was trained on the headlines dataset with incorrect classifications recorded. Similarly, we captured all errors made by our proposed model and the set difference was taken, that is, headlines which our model correctly identified, but the CNN did not. Analysis of these misclassifications was performed to determine whether our model was able to correctly classify additional headlines in which the sarcasm - or lack thereof - is clear based solely on the utterance. Below, we present a couple of these cases, paired with their actual label - sarcastic or nonsarcastic.

- *Efforts of world's 16 billion chickens still not adding up to much.* - sarcastic
- *CEO unveils bold new plan to undo damage from last year's bold new plan.* - sarcastic
- *Like boxes of shit in your house? Get a cat.* - sarcastic
- *United airlines temporarily suspends cargo travel for pets.* - nonsarcastic
- *There have been more mass shootings this year than there have been days.* - nonsarcastic

The initial three headlines in the above list are all clearly sarcastic, with no background knowledge required to classify it as such. The latter two are seemingly normal news headlines with no sarcastic cues present. All five of which were incorrectly classified by the CNN, but identified correctly by our approach. This suggests that additional cues are available in the isolated text of the utterance than a standard CNN is able to extract. We do however note that our approach fails to classify text which require a clear understanding of the topic or background, such as:

> *Cops cleared on corruption charges after implicating decorated police dog.*

This statement satirises cases where law enforcement were not held accountable for corruption, and further requires an understanding that a dog cannot be culpable of such an offense.

5 Conclusion

In this paper, we introduce a deep and dense network for extracting additional intrinsic information from a standalone utterance. Low-level features are shown to be used during the formation of the final feature maps. These, in combination with abstracted hierarchical features, enabled our model to rival state-of-the-art approaches which incorporated considerably more information on the SARC 2.0 datasets - such as user profiling and topic trends within a specific subforum. Our results demonstrate that whilst context is often needed to classify sarcasm; there is additional local information present that previous approaches have not taken advantage of.

References

[1] Jihen Karoui, Benamara Farah, Véronique Moriceau, Viviana Patti, Cristina Bosco, and Nathalie Aussenac-Gilles. Exploring the impact of pragmatic phenomena on irony detection in tweets: A multilingual corpus study. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 262–272, 2017.

[2] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135, 2007.

[3] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *CoNLL 2010*, 2010.

[4] Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Silva. Modelling context with user embeddings for sarcasm detection in social media. In *CoNLL*, 2016.

[5] Ranjan Satapathy, Claudia Guerreiro, Iti Chaturvedi, and Erik Cambria. Phonetic-based microtext normalization for twitter sentiment analysis. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 407–413, 2017.

[6] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. A deeper look into sarcastic tweets using deep convolutional neural networks. In *COLING*, 2016.

[7] Reza Ghaeini, Xiaoli Z. Fern, and Prasad Tadepalli. Attentional multi-reading sarcasm detection. *ArXiv*, abs/1809.03051, 2018.

[8] Joseph Tepperman, David R. Traum, and Shrikanth Narayanan. "yeah right": Sarcasm recognition for spoken dialogue systems. In *INTERSPEECH*, 2006.

[9] Paula Carvalho, Luís Sarmento, Mário J. Silva, and Eugénio de Oliveira. Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-). In *CIKM 2009*, 2009.

[10] Roberto I. González-Ibáñez, Smaranda Muresan, and Nina Wacholder. Identifying sarcasm in twitter: A closer look. In *ACL*, 2011.

[11] Christine Liebrecht, Florian Kunneman, and Antal van den Bosch. The perfect solution for detecting sarcasm in tweets #not. In *WASSA@NAACL-HLT*, 2013.

[12] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. Sarcasm as contrast between a positive sentiment and negative situation. In *EMNLP*, 2013.

[13] Konstantin Buschmeier, Philipp Cimiano, and Roman Klinger. An impact analysis of features in a classification approach to irony detection in product reviews. In *WASSA@ACL*, 2014.

[14] Antonio Reyes, Paolo Rosso, and Tony Veale. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47:239–268, 2013.

[15] Tomás Ptácek, Ivan Habernal, and Jun Hong. Sarcasm detection on czech and english twitter. In *COLING*, 2014.

[16] Byron C. Wallace, Do Kook Choe, Laura Kertz, and Eugene Charniak. Humans require context to infer ironic intent (so computers probably do, too). In *ACL*, 2014.

[17] David Bamman and Noah A. Smith. Contextualized sarcasm detection on twitter. In *ICWSM*, 2015.

[18] Byron C. Wallace, Do Kook Choe, and Eugene Charniak. Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In *ACL*, 2015.

[19] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*, 2008.

[20] Aniruddha Ghosh and Tony Veale. Fracking sarcasm using neural network. In *WASSA@NAACL-HLT*, 2016.

[21] Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. Cascade: Contextual sarcasm detection in online discussion forums. *ArXiv*, abs/1805.06413, 2018.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[23] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2016.

[24] Holger Schwenk, Loïc Barrault, Alexis Conneau, and Yann LeCun. Very deep convolutional networks for text classification. In *EACL*, 2016.

[25] Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A large self-annotated corpus for sarcasm. *ArXiv*, abs/1704.05579, 2017.

[26] Leslie N. Smith. Cyclical learning rates for training neural networks. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472, 2015.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[29] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.