Leveraging Transitions of Emotions for Sarcasm Detection

Ameeta Agrawal, Aijun An, Manos Papagelis
Department of Electrical Engineering and Computer Science, York University, Toronto, Canada
{ameeta,aan,papaggel}@eecs.yorku.ca

ABSTRACT

One popular thread of research in computational sarcasm detection involves modeling sarcasm as a contrast between positive and negative sentiment polarities or exploring more fine-grained categories of emotions such as *happiness*, *sadness*, *surprise*, and so on. Most current models, however, treat these affective features independently, without regard for the sequential information encoded among the affective states. In order to explore the role of transitions in affective states, we formulate the task of sarcasm detection as a sequence classification problem by leveraging the natural shifts in various emotions over the course of a piece of text. Experiments conducted on datasets from two different genres suggest that our proposed approach particularly benefits datasets with limited labeled data and longer instances of text.

CCS CONCEPTS

• **Computing methodologies → Natural language processing**.

KEYWORDS

sarcasm detection; emotion detection

ACM Reference Format:
Ameeta Agrawal, Aijun An, Manos Papagelis. 2020. Leveraging Transitions of Emotions for Sarcasm Detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401183

1 INTRODUCTION

The strong correlation between affective states and sarcasm has been well-highlighted by various research studies in psychology [4–6, 9, 10, 14, 17, 18, 25] as well as well-supported by recent works in computational sarcasm detection [1, 8, 12, 26]. In addition to the commonly used lexical features such as n-grams, punctuation, number of words, etc., some sarcasm detection models also employ affective features such as the frequency of positive, negative or emotion words. To further increase the vocabulary coverage, some models also leverage word embedding features [22]. A severe limitation of the current methods, however, is that they all employ features (e.g., minimum, average, sum, binary representation, etc.) independently without regard for the sequential information among the affective states. We hypothesize that additional advantages may be acquired by considering the sequences or transitions of affective states when designing sarcasm detection systems.

We formulate sarcasm detection as a sequence classification problem to design our model, which we call Emotion Transitions (EmoTrans). Each instance of text such as a paragraph or sentence is first divided into a number of consecutive non-overlapping chunks. Then, for each chunk, a vector of emotion features is computed by employing various emotion resources. Finally, the emotion vectors are fed into sequence classification models to learn patterns of affective transitions. Experiments on two sarcasm datasets (debate corpus and news headlines) demonstrate the potential of using emotion transitions, with the proposed model outperforming several baseline models.

Our main contributions can be summarized as follows:

- To the best of our knowledge, we present the first analysis of *emotion transitions* in sarcasm versus non-sarcasm text;
- We describe a novel method, EmoTrans, to leverage the inherent transitions of emotions within text for automatically detecting sarcasm using sequence classification;
- We demonstrate the effectiveness of the proposed approach by comparing against several baselines on two datasets.

2 SARCASM DETECTION USING EMOTIONS

As our proposed approach, EmoTrans, relies on identifying the transitions between emotions in text, we first segment the text into smaller chunks, then represent each chunk with affective features, and subsequently, utilize these features to train a model to detect sarcasm.

2.1 Chunking

In the first step, the input text (e.g., a sentence or paragraph) $s = \{w_1, w_2, ..., w_l\}$ of length l is split into a set of n non-overlapping segments called chunks, $C = \{c_1, c_2, ..., c_n\}$, where $2 \leq n \leq l$ such that a chunk consists of one or more consecutive words and there are at least 2 chunks in a sequence. We use k_i to denote the length of chunk c_i, i.e., the number of words in c_i.

We investigate three possible methods of obtaining such chunks:

- **Phrase-based chunking**: Intuitively, a sentence can be split into a sequence of phrases. We create a shallow parser using NLTK's chunking module and extract non-overlapping and non-embedded syntactical subtree phrases, yielding a sequence composed of a variable number of chunks with variable numbers of words.
- **Fixed-n chunking**: Since the unconventional language typically expressed in natural language text can be challenging for shallow parsing, we explore the method of dividing all the instances into a fixed number (i.e., n) of chunks.
- **Fixed-k chunking**: This variant explores having a fixed number (i.e., k) of consecutive words per chunk.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401183

2.2 Computing Emotion Scores

Each chunk c_i is, then, represented by an emotion vector $\mathbf{e}(c_i) = \langle s_{e_1}, s_{e_2}, ..., s_{e_m} \rangle$, where $s_{e_j} \in (0, 1)$ is a normalized real-valued degree of an emotion e_j from the set of emotions $E = \{e_1, e_2, ..., e_m\}$. We choose Ekman's model of emotions [7] consisting of $m = 6$ basic emotion categories, namely, *anger*, *disgust*, *fear*, *happiness*, *sadness* and *surprise*, as this model is well-represented in previous works of computational sarcasm detection [1, 26].

To obtain an emotion vector of a chunk c_i, we first compute an emotion vector for each word w in c_i using one of the following three resources/methods:

- **WordNet Affect (WNA)**: For each emotion category in Ekman's theory of emotions, WNA [28] specifies a list of words that are associated with that emotion. The emotion vector of a word w is represented as $\langle s_1, s_2, ..., s_6 \rangle$, where s_i is either 0 or 1, depending on whether w is associated with the emotion or not.
- **NRC EmoLex (NRC)**: NRC EmoLex [21] contains about 14,200 unigrams annotated with one or more of ten affect categories (eight emotions as well as positive and negative sentiments). For instance, the binary association between the word *"awful"* and 10 affect categories (anger, anticipation, disgust, fear, joy, negative, positive, sadness, surprise, trust) is represented as: 1, 0, 1, 1, 0, 1, 0, 1, 0, 0. For a given word w, we extract its binary association scores corresponding to the six categories of Ekman's model.
- **Word-Embedding based Emotion Scores (WEES)**: Using the above two lexicons, most of the words are considered neutral with all zeros in their vectors. However, some of these words may bear some degree of emotions. In this method, we compute the emotion vector for each word in an unsupervised manner using cosine similarity between a pre-trained word embedding [1] of the word and the embedding of a seed word per emotion category. The list of seed words is "angry", "disgust", "fear", "happy", "sad" and "surprise" representing the six different emotion categories. For example, the emotion vector for the word *"awful"* is $\langle 0.33, 0.29, 0.15, 0.26, 0.45, 0.19 \rangle$, where the values are the cosine similarities between *"awful"* and each of the above seed words.

After obtaining the emotion vector for each word with one of the above methods, the emotion vector of a chunk c_i is the average of the emotions vectors of all the words in c_i: $\mathbf{e}(c_i) = \frac{1}{k_i} \sum_{j=1}^{k_i} \mathbf{e}(w_j)$, where $w_j \in c_i$ and $|c_i| = k_i$. Then, the input text with n chunks is represented as $\langle \mathbf{e}(c_1), \mathbf{e}(c_2), \ldots, \mathbf{e}(c_n) \rangle$.

2.3 Classifying Sequences for Sarcasm

To build a classifier using sequences of multidimensional emotion vectors, we train a Long Short-Term Memory network (LSTM) to model sequence data with a set of training data.

Given a multidimensional sequence $\mathbf{E} = \langle \mathbf{e}(c_1), \mathbf{e}(c_2), \ldots, \mathbf{e}(c_T) \rangle$, where $\mathbf{e}(c_i)$ is a vector of 6 scores, one for each emotion, and T is the number of chunks (i.e., time steps), the LSTM model processes it sequentially. In other words, the input data is reshaped as T timestamps, each with 6 features. We used binary cross entropy as the loss function for the two class classification, and Adam algorithm [16] for optimization.

[1] https://code.google.com/archive/p/word2vec/

Table 1: Statistics of sarcasm datasets

Dataset	Genre	Avg. Length	Sarcasm	Non-sarcasm	Total
IAC	debate	41	1630	1630	3260
Onion	headlines	12	13634	14985	28619

3 EXPERIMENTS

3.1 Evaluation Datasets

We employ two sarcasm datasets, IAC debate corpus [23] and Onion news headlines [20] for evaluating the performance of our proposed approach. IAC contains response utterances annotated for sarcasm[2], whereas Onion news headlines[3] is a collection of sarcastic versions of current events from The Onion and non-sarcastic news headlines from HuffPost. Table 1 summarizes the statistics of the evaluation datasets while Table 2 presents some sample instances.

3.2 Baselines

We compare our proposed approach against several baselines described below.

(1) **LSTM with word embeddings (LSTM WordEmbed)**: Given a sentence s, this method sequentially inputs its word vectors $\langle \mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x}_{|\mathbf{s}|} \rangle$ to an LSTM model for text classification. As pre-trained word embeddings have shown to be useful in sarcasm detection with text [11, 29], this baseline consists of pre-trained word2vec word embeddings [19] (dimensions = 300, trained on the Google News corpus) [24].

(2) **Rule-based methods (Rule-based PosNeg)**: In [27], the authors popularized the notion of sarcasm as a contrast between positive and negative sentiment. We re-implement one of their rule-based algorithms for comparison: an instance is labeled as sarcastic if it contains both a positive sentiment term and a negative sentiment term, in any order. The NRC EmoLex lexicon [21] is used to identify the positive and negative terms.

(3) **LSTM with sentiment scores but no chunking (PosNeg No Chunk)**: Using sentiment polarities as features has been shown effective for sarcasm detection [3, 13, 15, 27]. For this baseline method, we extract positive and negative sentiment scores of an input text using a method similar to WEES described in §2.2. That is, we compute a sentiment vector for a word that contains two components $\langle s_{pos}, s_{neg} \rangle$, where s_{pos} or s_{neg} are the cosine similarity between embeddings of the word and a seed word "good" or "bad", respectively. The sentiment vector of the input text is the average of the sentiment vectors of all the words in the text[4]. Then LSTM is trained to predict sarcasm based on the sentiment scores.

(4) **LSTM with sentiment transitions (PosNeg Trans)**: This method is the same as "PosNeg No Chunk" except that we apply the same chunking methods described earlier for modeling emotion transitions (§2.3), for comparison to see how sentiment transitions affect sarcasm detection.

[2] https://nlds.soe.ucsc.edu/sarcasm2

[3] https://github.com/rishabhmisra/News-Headlines-Dataset-For-Sarcasm-Detection

[4] Note that we compute the sentiment scores in this way instead of using NRC EmoLex as in the second baseline because this produces better results.

Table 2: Sample instances from evaluation datasets

Dataset	Text Instance	Label
IAC	*Anyone wanna make any bets on who the newbie is here???*	sarcasm
	Haaaaa Haaaaa Haaaaaaah haahhhha HAhaaaa. My GOD I can't stop laughing Heeee Haaaa.	sarcasm
	To the extent that working toward social change on any moralistic issue is an 'imposition', sure. But I think one could say that about anything that has a social effect.	non-sarcasm
Onion	*inclement weather prevents liar from getting to work*	sarcasm
	handshake comes in at unusually high angle, velocity	sarcasm
	10 big space-saving ideas for small kitchens	non-sarcasm

Table 3: Comparison of four emotion detection methods on sarcasm detection with Fixed-*n* chunking method on the IAC and Onion datasets. The values of classification accuracy are in percentage.

n	IAC				Onion			
	WEES+NRC	WEES	NRC	WNA	WEES+NRC	WEES	NRC	WNA
2	62.6	**65.8**	57.2	51.5	60.7	**61.9**	54.2	52.8
3	63.0	**64.4**	56.4	51.4	61.5	**62.6**	54.5	52.5
4	62.7	**65.3**	56.7	50.8	52.9	**63.9**	54.3	52.7
5	63.0	**66.4**	55.1	50.0	58.6	**62.5**	55.2	52.4

(5) **LSTM with emotion vector but no chunking (Emotion No Chunk)**: This method does not divide the input text into chunks and instead uses a single emotion vector for the whole input text as the input to LSTM. It is equivalent to the proposed method with the number of chunks being 1.

All the approaches are evaluated using 80%-20% train-test split and the results are reported in terms of accuracy.

3.3 Results

We first investigate which method (WNA, NRC or WEES) described in 2.2 works the best for obtaining the word emotion vectors. Table 3 shows the accuracy scores of sarcasm detection with the three emotion vector computation methods using the Fixed-*n* chunking method for different number of chunks, together with a method that combines the two better methods (i.e., WEES and NRC), denoted as WEES+NRC. In WEES+NRC, the emotion vector for a word is the average of the emotion vectors for the word from WEES and NRC. From the results, it is clear that WEES alone consistently produces the best results among the four methods on both datasets. As such, we only use WEES as the emotion detection method for the rest of the experiments.

Table 4 presents the accuracy of sarcasm detection of all the baselines and our proposed method, EmoTrans. We notice the following observations. First, between sentiment and emotion models, the emotion-based methods perform better on both datasets. This can be seen by comparing "PosNeg No Chunk" with "Emotion No Chunk" in the non-sequence situation, and comparing "PosNeg-Trans fixed-*n* chunks" with "EmoTrans fixed-*n* chunks" for the sequence classification case[5]. These results suggest the usefulness of a richer spectrum of emotions in sarcasm detection. Second, we note that chunking (especially fixed-*n* chunking for EmoTrans) yields improvement over no chunking, validating the effectiveness of using emotion transitions. Third, comparing the three types of chunking, fixed-*n* chunking appears to be the best option overall on both the datasets. Note that while phrase-based chunking works well for the IAC dataset which contains proper sentences, it is unsurprisingly not suitable for Onion dataset's headlines text. Finally, on the IAC dataset, EmoTrans (with fixed-*n* and phrase-based chunking) achieves the best results, whereas on the Onion dataset, EmoTrans' results are comparable to that of the LSTM WordEmbed, which outperforms other methods.

Table 4: Results of sarcasm detection of different methods, shown in accuracy (%). Top two best results for each dataset are shown in bold.

Methods	IAC	Onion
LSTM WordEmbed	61.9	**66.2**
Rule-based PosNeg	43.8	51.2
PosNeg No Chunk	61.2	55.4
PosNegTrans, fixed-*n* chunks ($n = 2$)	60.6	56.2
PosNegTrans, fixed-*n* chunks ($n = 3$)	61.3	57.0
PosNegTrans, fixed-*n* chunks ($n = 4$)	60.9	56.6
PosNegTrans, fixed-*n* chunks ($n = 5$)	60.4	56.1
PosNegTrans, phrase-based chunks	59.8	54.2
Emotion No Chunk	64.6	61.6
EmoTrans, fixed-*n* chunks ($n = 2$)	**65.8**	61.9
EmoTrans, fixed-*n* chunks ($n = 3$)	64.4	62.6
EmoTrans, fixed-*n* chunks ($n = 4$)	65.3	**63.9**
EmoTrans, fixed-*n* chunks ($n = 5$)	**66.4**	62.5
EmoTrans, fixed-*k* chunks ($k = 2$)	64.6	52.1
EmoTrans, fixed-*k* chunks ($k = 3$)	60.1	54.7
EmoTrans, fixed-*k* chunks ($k = 4$)	49.8	60.4
EmoTrans, fixed-*k* chunks ($k = 5$)	49.8	53.7
EmoTrans, phrase-based chunks	**65.8**	60.1

Looking at the characteristics of the two datasets, we notice that IAC has longer input texts (debate responses) but a smaller number of training instances (3,260 instances), while the Onion

[5]The results of PosNegTrans fixed-*k* chunks are also worse than those for EmoTrans, which were not presented due to space limitation.

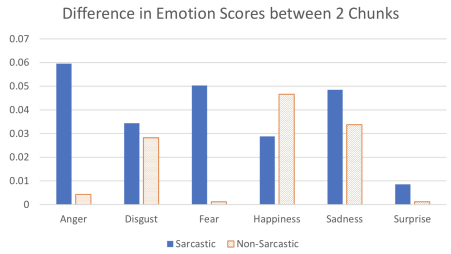

Figure 1: Difference in emotion scores between two consecutive chunks (transitions) in the IAC dataset

dataset consists of a large number (28,619 instances) of shorter texts (headlines). Based on our results, we conjecture that EmoTrans is more effective with long input texts and limited labeled training data than LSTM with word embeddings.

To see how emotions change in sarcastic compared to non-sarcastic texts, Figure 1 shows the average of absolute differences between emotions of two consecutive chunks of a piece of text in the IAC dataset. Note that these emotion scores were obtained through unsupervised emotion labeling algorithm (described earlier in §2.2), and thus, likely contain some degree of noise. However, even this noisy distribution lends a discriminative observation: on 5 of the 6 emotion categories, the difference between emotion scores in adjacent chunks of text is higher in sarcastic texts than that in non-sarcastic texts, which supports our hypothesis that the transitions of emotions can be a useful feature for detecting sarcasm.

4 CONCLUSIONS

In this paper, we investigated whether emotion transitions in text can be leveraged for detecting sarcasm from texts, and introduced a novel methodology for detecting sarcasm by formulating it as an emotion sequence classification problem. To demonstrate the potential of our approach, we conducted experiments on two datasets of different genres and sizes, where the proposed approach outperformed several baselines. In particular, the results indicate that exploiting fine-grained categories of emotions such as *anger*, *happiness*, etc., yield better results than binary polarities comprising of positive and negative sentiment. Furthermore, chunks of emotion transitions offer significant improvement over no chunking. The results suggest that our proposed approach particularly benefits datasets with limited labeled data and longer instances of text which are generally more challenging to annotate. As future work, we would like to explore more taxonomies of emotion modeling and investigate whether combining EmoTrans with emotion-specific word embeddings [2] can improve its performance further.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and Nastaran Babanejad for their helpful feedback. This work is funded by the Big Data Research, Analytics and Information Network (BRAIN) Alliance established by Ontario Research Fund - Research Excellence Program (ORF-RE), and the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] Ameeta Agrawal and Aijun An. 2018. Affective representations for sarcasm detection. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1029–1032.
[2] Ameeta Agrawal, Aijun An, and Manos Papagelis. 2018. Learning emotion-enriched word representations. In *Proceedings of the 27th International Conference on Computational Linguistics*. 950–961.
[3] Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014. Modelling sarcasm in twitter, a novel approach. *ACL 2014* (2014), 50.
[4] James Boylan and Albert N Katz. 2013. Ironic expression can simultaneously enhance and dilute perception of criticism. *Discourse Processes* 50, 3 (2013).
[5] John D Campbell and Albert N Katz. 2012. Are there necessary conditions for inducing a sense of sarcastic irony? *Discourse Processes* (2012).
[6] Herbert L Colston. 1997. Salting a wound or sugaring a pill: The pragmatic functions of ironic criticism. *Discourse Processes* 23, 1 (1997), 25–45.
[7] Paul Ekman. 1992. An argument for basic emotions. *Cognition & Emotion* 6, 3-4 (1992). http://dx.doi.org/10.1080/02699939208411068
[8] Delia Irazú Hernańdez Farías, Viviana Patti, and Paolo Rosso. 2016. Irony Detection in Twitter: The Role of Affective Content. *ACM Transactions on Internet Technology (TOIT)*, Article 19 (2016), 24 pages. https://doi.org/10.1145/2930663
[9] Ruth Filik, Christian Mark Hunter, and Hartmut Leuthold. 2015. When language gets emotional: Irony and the embodiment of affect in discourse. *Acta psychologica* 156 (2015), 114–125.
[10] Ruth Filik, Alexandra Turcan, Dominic Thompson, Nicole Harvey, Harriet Davies, and Amelia Turner. 2016. Sarcasm and emoticons: Comprehension and emotional impact. *The Quarterly Journal of Experimental Psychology* 69, 11 (2016), 2130–2146. https://doi.org/10.1080/17470218.2015.1106566 PMID: 26513274.
[11] Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan. 2015. Sarcastic or Not: Word Embeddings to Predict the Literal or Sarcastic Meaning of Words. In *EMNLP*. http://aclweb.org/anthology/D/D15/D15-1116.pdf
[12] Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying Sarcasm in Twitter: A Closer Look. In *Human Language Technologies: Short Papers*. ACL. http://dl.acm.org/citation.cfm?id=2002736.2002850
[13] Irazú Hernández-Farías, José-Miguel Benedí, and Paolo Rosso. 2015. *Applying Basic Features from Sentiment Analysis for Automatic Irony Detection*. Springer.
[14] Julia Jorgensen. 1996. The functions of sarcastic irony in speech. *Journal of Pragmatics* 26, 5 (1996), 613–634.
[15] Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing Context Incongruity for Sarcasm Detection. In *ACL and IJCNLP, 2015, Short Papers*. 757–762. http://aclweb.org/anthology/P/P15/P15-2124.pdf
[16] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). http://arxiv.org/abs/1412.6980
[17] Roger J Kreuz, Debra L Long, and Mary B Church. 1991. On being ironic: Pragmatic and mnemonic implications. *Metaphor and symbol* 6, 3 (1991), 149–162.
[18] Ida Unmack Larsen, Tua Vinther-Jensen, Anders Gade, Jørgen Erik Nielsen, and Asmus Mejling Vogel. 2016. Do I misconstrue?: Sarcasm detection, emotion recognition, and Theory of Mind in Huntington disease. *Neuropsychology* 30, 2 (2016), 181–189. https://doi.org/10.1037/neu0000224
[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). http://arxiv.org/abs/1301.3781
[20] Rishabh Misra and Prahal Arora. 2019. Sarcasm Detection using Hybrid Neural Network. *arXiv preprint arXiv:1908.07414* (2019).
[21] Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a Word-Emotion Association Lexicon. 29, 3 (2013), 436–465.
[22] Aytuğ Onan. 2019. Topic-enriched word embeddings for sarcasm identification. In *Computer Science On-line Conference*. Springer, 293–304.
[23] Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2017. Creating and characterizing a diverse corpus of sarcasm in dialogue. *arXiv preprint arXiv:1709.05404* (2017).
[24] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*.
[25] Louise H Phillips, Roy Allen, Rebecca Bull, Alexandra Hering, Matthias Kliegel, and Shelley Channon. 2015. Older adults have difficulty in decoding sarcasm. *Developmental psychology* 51, 12 (2015), 1840.
[26] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. 2016. A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks. In *COLING 2016, Osaka, Japan*. http://aclweb.org/anthology/C/C16/C16-1151.pdf
[27] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In *EMNLP*.
[28] Carlo Strapparava and Alessandro Valitutti. 2004. WordNet-Affect: An affective extension of WordNet. In *LREC*. 1083–1086.
[29] Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet Sarcasm Detection Using Deep Neural Network. In *COLING*. http://aclweb.org/anthology/C/C16/C16-1231.pdf