Incorporating Emoji Descriptions Improves Tweet Classification

Abhishek Singh, **Eduardo Blanco** and **Wei Jin**
Computer Science and Engineering
University of North Texas
Denton, TX, 76203
abhisheksingh@my.unt.edu, {eduardo.blanco,wei.jin}@unt.edu

Abstract

Tweets are short messages that often include specialized language such as hashtags and emojis. In this paper, we present a simple strategy to process emojis: replace them with their natural language description and use pretrained word embeddings as normally done with standard words. We show that this strategy is more effective than using pretrained emoji embeddings for tweet classification. Specifically, we obtain new state-of-the-art results in irony detection and sentiment analysis despite our neural network is simpler than previous proposals.

1 Introduction

Tweets are short messages shared on Twitter, one of the most popular social networking services with 326 million monthly active users word wide (Twitter, 2018). Tweets often use specialized language such as abbreviations (e.g., *TBH: To be honest*), hashtags (e.g., *#NBAFinals*), emoticons and emojis. The Oxford Dictionary defines an emoticon as "a facial expression such as a smile or frown, formed by various combinations of keyboard characters" (e.g., ":)", ":-("), and an emoji as "a small digital image or icon used to express an idea or emotion" (e.g., 😀, 👏🏼, 😷). While the number of emoticons is relatively small, the Unicode Standard includes over 2,800 emojis.

Emojis are interesting because they succinctly encode meaning that otherwise would require more than one word to convey (e.g., *grinning face*, *clapping hands* and *face with medical mask* for the emojis above). Additionally, emojis have become popular in social media. 5 billion emojis are sent daily on Facebook (Burge, 2018). While only 6% of the top-100 Facebook headlines used emojis in 2015, 52% did so in 2017 (Boland, 2017). Over 14% of tweets and 50% of Instagram posts contain at least one emoji (Cruse, 2015; Moon, 2015).

	Irony?
I just love being ignored😊|#not 😒	Yes
Love it when my mans on a cleaning spree.. Saves me doing it 👌🏼😘	No

	Sentiment
@Paul_OConnor187 hi we going to see ted 2 at the Odeon cinemas at Glasgow on Wednesday 😊	Positive
Serato DJ isn't compatible with Windows 10 yet 😞 ...got to spin on my old laptop Saturday.	Negative

Table 1: Sample tweets with their irony and sentiment judgements. Note that the emojis help to determine irony usage and the author's sentiment.

Irony detection and sentiment analysis in tweets are two popular tasks. Sentiment analysis has received substantially more attention than irony detection. Irony, however, is a major error source in sentiment analysis (0.71 F1 overall but 0.29 F1 with ironic tweets (Hee et al., 2018)), and natural language understanding in general does not generalize well with ironic texts (Liu et al., 2012; Maynard and Greenwood, 2014).

In this paper, we tackle both irony and sentiment analysis in tweets—two classification tasks. In particular, we focus on modeling emojis. Consider the examples in Table 1. Understanding the emojis is critical to making irony and sentiment judgements. In the first example, the contrast between the emojis helps determining that irony is present (the hashtag *#not* also helps). In the second tweet, the *OK hand sign* and *face blowing a kiss* emojis help reinforcing that the author is praising somebody and not being ironic. Similarly, the smiling and sad emojis in the last two examples are a clear

Proceedings of NAACL-HLT 2019, pages 2096–2101
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

sign of the author's sentiment towards the movie *Ted 2* and the incompatibility issue.

The main contributions of this paper are twofold. First, we present a simple strategy to model emojis: replace them with their textual description. Second, we show that this strategy outperforms previous methods and yields a new state-of-the-art in two tweet classification tasks: irony detection and sentiment analysis.

2 Related Work

Irony is closely related to sarcasm. The Oxford Dictionary defines irony as "The expression of one's meaning by using language that normally signifies the opposite, typically for humorous or emphatic effect", and sarcasm as "The use of irony to mock or convey contempt." Given these definitions, it is not surprising that many researchers do not distinguish between them (Maynard and Greenwood, 2014). The top-3 systems to detect irony are built with neural networks and pretrained word embeddings. Baziotis et al. (2018) build an ensemble of two stacks of BiLSTMs (word and character level) with attention. Wu et al. (2018) propose a BiLSTM and a multitask learning framework (hashtag, irony presence and irony type prediction), and complement the input text with sentiment features extracted from lexicons. Vu et al. (2018) propose a multilayer perceptron taking as input an embedding for the input text (average of word embeddings) as well as manually crafted lexical, syntactic, semantic and polarity features. Our strategy to incorporate emojis outperforms all of them (Table 3).

Sentiment analysis in tweets has been studied for years (Nakov et al., 2013). At its core, it is the task of classifying a tweet into expressing positive, neutral or negative sentiment (Rosenthal et al., 2017). Initial systems were primarily based on sentiment lexicons and manually extracted features, but the state of the art uses neural networks and word embeddings. Baziotis et al. (2017) propose a stack of two BiLSTMs at the word level and do not use any lexicons. Cliche (2017) presents a CNN and BiLSTM ensemble and experiment with three pretrained embeddings. Rouvier (2017) also presents a CNN and BiLSTM ensemble but incorporates manually defined features (e.g., word presence in emotion lexicons, all-caps). The strategy presented here to incorporate emojis outperforms all these systems (Table 4).

Within natural language processing and social media, emojis have received considerable attention. Barbieri et al. (2016) train emoji embeddings with word2vec and discover that the closest words are sound (e.g., ☕: coffee, roasters, caffeine, latte). Eisner et al. (2016) propose a complementary approach to train emoji embeddings (Section 3). Emojis have also been used as labels for distant supervision to improve tweet classification (Felbo et al., 2017). The strategy presented here to incorporate emojis is simpler and more effective than previous ones, does not require additional pretraining or domain specific corpora, and can be used with any neural architecture that takes text as input without any modifications. Simply put, we replace emojis with their textual descriptions and leverage existing pretrained word embeddings.

3 Strategies to Incorporate Emojis

Neural networks that take as input text usually transform the input tokens into pretrained embeddings. When the input text are tweets, it is common to use embeddings pretrained with large collections of tweets as opposed to general purpose text (Li et al., 2017; Pennington et al., 2014).

Emojis as Regular Tokens. The simplest option to incorporate emojis into a neural network is to consider them as any other token in the input text (Barbieri et al., 2016). This strategy relies on having seen enough instances of each emoji in the texts with which embeddings were pretrained—otherwise the embeddings will not capture the semantics of emoji tokens properly.

Emoji Embeddings. Another strategy is to use separate embeddings for emojis. Eisner et al. (2016) pretrain emoji embeddings using positive and negative (randomly sampled) emoji descriptions. Descriptions are transformed into a vector by adding the corresponding word2vec embeddings (Mikolov et al., 2013). Emoji embeddings are tuned quickly because only a positive and a negative description per emoji are considered.

We refer to this strategy as EMJ-EMBED.

Our Strategy: Emoji Descriptions. Our strategy is simple: replace emojis with their textual descriptions. Effectively, this eliminates all emojis in the input and incorporates a rather detailed description—several tokens—of the emojis (see examples in Table 2). Our rationale is as follows. First, lists of emojis and their textual descriptions

Emoji	Description
😂	Face with tears of joy
😘	Face blowing a kiss
😄	Grinning face with smiling eyes
😌	Relieved face
😝	Squinting face with tongue
😥	Sad but relieved face
😠	Angry face
😭	Loudly crying face
😓	Downcast face with sweat
😰	Anxious face with sweat

Table 2: Emojis and their textual description.

are readily available.[1] Second, while emojis are common (Section 1), words are more common. Thus, it is reasonable to expect word embeddings to capture the meaning of words better than emoji embeddings capture the meaning of emojis. Consider the last two examples in Table 2, 😓 and 😰. These emojis are relatively uncommon, and pretrained emoji embeddings do not leverage the fact that both of them are *faces* with *sweat*. Using the textual descriptions bypasses both issues.

Finally, this strategy is straightforward, fast to implement and run, and can be used regardless of the neural network architecture. Indeed, it could be considered a preprocessing step.

We refer to this strategy as EMJ-DESC.

4 Experiments and Results

We experiment with two tweet classification tasks: irony detection and sentiment analysis. We use standard corpora and compare with previous work using the same set up (i.e., we train and test with exactly the same instances they did).

4.1 Experimental Setup

Corpora. For irony detection, we use the corpus released by Hee et al. (2018). It includes two tasks: binary (Task A: yes or no) and 4-way multiclass irony detection (Task B: verbal irony realized through polarity contrast, other verbal irony, situational irony or non-irony). The corpus consists of 3,000 tweets (yes: 2396, no: 604; verbal irony with polarity contrast: 1,728, other verbal irony: 267, situational irony: 401, non-irony: 604). Here are some examples: *I love waking up with migraines #not* 😣 (verbal, polarity contrast), *I cared for 8 seconds, then I got distracted.* 😂 (other verbal), *I wonder what Professor Iaukea has to say about the new Disney Princess...?* 😒 (situational), and *Is Obamacare Slowing Health Care Spending? #NOT* (non-irony).

For sentiment analysis, we use the corpus released by Rosenthal et al. (2017), which has 62,617 tweets (positive: 22,277, neutral: 28,528, negative: 8,982). Table 1 shows examples of positive and negative sentiment, and here is an example of neutral sentiment: *I'm switching to T-Mobile tomorrow and I'm getting a new number.*

Evaluation Metrics. We follow the metrics used by previous work. Regarding irony detection, we report accuracy and macro-average F1 (all labels weighted equal regardless of frequency). Regarding sentiment analysis, we report accuracy, average recall and F1. Following previous work, we calculate accuracy and average recall using all labels (positive, negative and neutral) but F1 using only positive and negative instances.

Preprocessing. We preprocess the input text following standard steps. Specifically, we tokenize with the NLTK's TweetTokenizer (Bird, 2006), lowercase all text, and use regular expressions to remove stop words, numbers, urls, consecutive repeated words and Twitter users (i.e., tokens whose first character is '@'). We also expand hashtags (e.g., *#PickANewSong*: *Pick a new song*) with ekphrasis (Baziotis et al., 2017).

Regarding emojis, we either (a) do nothing special and use pretrained emoji embeddings (EMJ-EMBED strategy), or (b) replace emojis with their textual description and use pretrained word embeddings for the words in their descriptions (EMJ-DESC strategy).

Let us consider the following tweet: "@Paul_OConnor187 hi we going to see ted 2 at the Odeon cinemas at Glasgow on Wednesday 😊". After preprocessing, we transform it into "hi we going see ted odeon cinemas glasgow wednesday 😊" or "hi we going see ted odeon cinemas glasgow wednesday smiling face" (EMJ-EMBED and EMJ-DESC strategies respectively).

Neural Network Architecture. We experiment with a stack of two BiLSTMs (Dyer et al., 2015) with attention (Zhou et al., 2016) to generate distributed representations of the input, and a softmax layer as the output layer. This architecture is simpler than previous proposals, but as we shall see,

[1] https://pypi.org/project/emoji-unicode/, https://github.com/uclmr/emoji2vec, https://emojipedia.org/

		Task A		Task B	
		Acc.	F1	Acc.	F1
Previous Work (Top 3)	Vu et al. (2018)	0.7015	0.6476	0.6594	0.4437
	Wu et al. (2018)	0.7347	0.7054	0.6046	0.4947
	Baziotis et al. (2018)	0.7883	0.7856	0.6888	0.5358
This paper	EMJ-EMBED	0.7864	0.7814	0.6940	0.5434
	EMJ-DESC	**0.8056**	**0.8031**	**0.7187**	**0.5565**

Table 3: Results on irony detection (Accuracy and Macro F1). Task A is a binary classification (yes / no) and Task B is a four-way classification (verbal irony with polarity contrast, other verbal irony, situational irony, non-irony).

		Avg. Rec.	Acc.	F1
Previous Work (Top 3)	Baziotis et al. (2017)	0.681	0.651	0.677
	Cliche (2017)	0.681	0.658	0.685
	Rouvier (2017)	0.676	0.661	0.674
This paper	EMJ-EMBED	0.703	0.689	0.691
	EMJ-DESC	**0.728**	**0.704**	**0.703**

Table 4: Results on sentiment analysis (three-way classification: positive, neural or negative).

outperforms previous work when coupled with our strategy to incorporate embeddings. Regarding word embeddings, we use the ones trained by Baziotis et al. (2018) using word2vec (Mikolov et al., 2013) and 550 million tweets. Regarding emoji embeddings, we use emoji2vec (Eisner et al., 2016). Note that EMJ-EMBED uses both word and emoji embeddings whereas EMJ-DESC only uses word embeddings.

4.2 Results

Irony Detection. Table 3 presents the results for irony detection. EMJ-EMBED obtains virtually the same results than the state of the art, although the neural architecture is much simpler (Section 2). EMJ-DESC, however, obtains the best results to date: (Task A: 0.80 vs. 0.78 F1, Task B: 0.55 vs. 0.53 F1). These results show that replacing embeddings with their textual descriptions and using the corresponding word embeddings is more effective than using emoji embeddings. As discussed earlier, there is a large amount of emojis (over 2,800), and some of them are infrequent. Many have, however, words in common in their descriptions thus leveraging the descriptions is beneficial.

Sentiment Analysis. Table 4 presents results for sentiment analysis. The standard evaluation metric in this task is average recall (Rosenthal et al., 2017), but we also provide accuracy and F1. Both EMJ-EMBED and EMJ-DESC outperform the state of the art despite we experiment with a simpler neural architecture. Indeed, EMJ-DESC outperforms previous work by a substantial margin (+0.047, +6.9% avg. recall). The reason for these results is the same than for irony detection: modeling emojis is key for tweet classification and our strategies to incorporate emojis are better suited than the ones used by previous work, which primarily treat them as any other token.

5 Conclusions

We have presented a strategy to incorporate emojis into any neural network: replace them with their textual descriptions. This strategy does not require any additional pretraining or component in the network. Instead, it leverages pretrained word embeddings, which are readily available and pretrained using massive amounts of text (Mikolov et al., 2013; Pennington et al., 2014).

Experimental results show that our strategy is more effective than previous ones (either consider emojis as regular tokens or use specialized emoji embeddings). Indeed, we obtain new state-of-the-art results in two tweet classification tasks: irony detection and sentiment analysis. We hypothesize that this is due to two reasons. First, while emojis are common, the words in their descriptions are more common. Thus, there is more data to pretrain word embeddings than emoji embeddings. Simply put, there is more proper English texts available than social media text with emojis. Second, emoji descriptions have many words in common (Table

2), thus many emojis benefit from a single word embedding (e.g., the textual descriptions of 😌 and 😥 share the words *face* and *relieved*).

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants Nos. 1734730, 1832267 and 1845757. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The Titan Xp used for this research was donated by the NVIDIA Corporation.

References

Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2016. What does this emoji mean? A vector space skip-gram model for twitter emojis. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*

Christos Baziotis, Athanasiou Nikolaos, Pinelopi Papalampidi, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, and Alexandros Potamianos. 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, SemEval@NAACL-HLT, New Orleans, Louisiana, June 5-6, 2018*, pages 613–621.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Steven Bird. 2006. NLTK: the natural language toolkit. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006.*

Gabriele Boland. 2017. How emoji usage has exploded by 766% on social. `https://www.newswhip.com/2017/12/emoji-on-social/`. [Online; accessed Nov 17th, 2018].

Jeremy Burge. 2018. 5 billion emojis sent daily on messenger. [Online; accessed Nov 17th, 2018].

Mathieu Cliche. 2017. Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 573–580, Vancouver, Canada. Association for Computational Linguistics.

Joe Cruse. 2015. Emoji usage in TV conversation. `https://blog.twitter.com/official/en_us/a/2015/emoji-usage-in-tv-conversation.html`. [Online; accessed Dec 5th, 2018].

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 334–343.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bosnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media, SocialNLP@EMNLP 2016, Austin, TX, USA, November 1, 2016*, pages 48–54.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1615–1625.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT, New Orleans, Louisiana, June 5-6, 2018*, pages 39–50.

Quanzhi Li, Sameena Shah, Xiaomo Liu, and Armineh Nourbakhsh. 2017. Data sets: Word embeddings learned from tweets and general data. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017.*, pages 428–436.

Kun-Lin Liu, Wu-Jun Li, and Minyi Guo. 2012. Emoticon smoothed language models for twitter sentiment analysis. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*

Diana Maynard and Mark A. Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 4238–4243.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Mariella Moon. 2015. Instagram takes a serious look at how people use emojis. `https://www.engadget.com/2015/05/02/instagram-emoji-study/`. [Online; accessed Dec 5th, 2018].

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013*, pages 312–320.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, SemEval '17, Vancouver, Canada. Association for Computational Linguistics.

Mickael Rouvier. 2017. Lia at semeval-2017 task 4: An ensemble of neural networks for sentiment classification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 760–765, Vancouver, Canada. Association for Computational Linguistics.

Twitter. 2018. Q3 2018 Earnings Report. `https://investor.twitterinc.com/static-files/4bfbf376-fefd-43cc-901e-aedd6a7f1daf`. [Online; accessed Dec 5th, 2018].

Thanh Vu, Dat Quoc Nguyen, Xuan-Son Vu, Dai Quoc Nguyen, Michael Catt, and Michael Trenell. 2018. NIHRIO at semeval-2018 task 3: A simple and accurate neural network model for irony detection in twitter. *CoRR*, abs/1804.00520.

Chuhan Wu, Fangzhao Wu, Sixing Wu, Junxin Liu, Zhigang Yuan, and Yongfeng Huang. 2018. Thu_ngn at semeval-2018 task 3: Tweet irony detection with densely connected lstm and multi-task learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 51–56. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.