Disambiguating False-Alarm Hashtag Usages in Tweets for Irony Detection

Hen-Hsen Huang,[1] **Chiao-Chen Chen,**[1] and **Hsin-Hsi Chen**[12]
[1] Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
[2] MOST Joint Research Center for AI Technology and All Vista Healthcare, Taipei, Taiwan
hhhuang@nlg.csie.ntu.edu.tw, {b04902055,hhchen}@ntu.edu.tw

Abstract

The reliability of self-labeled data is an important issue when the data are regarded as ground-truth for training and testing learning-based models. This paper addresses the issue of false-alarm hashtags in the self-labeled data for irony detection. We analyze the ambiguity of hashtag usages and propose a novel neural network-based model, which incorporates linguistic information from different aspects, to disambiguate the usage of three hashtags that are widely used to collect the training data for irony detection. Furthermore, we apply our model to prune the self-labeled training data. Experimental results show that the irony detection model trained on the less but cleaner training instances outperforms the models trained on all data.

1 Introduction

Self-labeled data available on the Internet are popular research materials in many NLP areas. Metadata such as tags and emoticons given by users are considered as labels for training and testing learning-based models, which usually benefit from large amount of data.

One of the sources of self-labeled data widely used in the research community is Twitter, where the short-text messages tweets written by the crowd are publicly shared. In a tweet, the author can tag the short text with some hashtags such as #excited, #happy, #UnbornLivesMatter, and #Hillary4President to express their emotion or opinion. The tweets with a certain types of hashtags are collected as self-label data in a variety of research works including sentiment analysis (Qadir and Riloff, 2014), stance detection (Mohammad et al., 2016; Sobhani et al., 2017), financial opinion mining (Cortis et al., 2017), and irony detection (Ghosh et al., 2015; Peled and Reichart, 2017; Hee et al., 2018). In the case of irony detection, it is impractical to manually annotate the ironic sentences from randomly sampled data due to the relatively low occurrences of irony (Davidov et al., 2010). Collecting the tweets with the hashtags like #sarcasm, #irony, and #not becomes the mainstream approach to dataset construction (Sulis et al., 2016). As shown in (S1), the tweet with the hashtag #not is treated as a positive (ironic) instance by removing #not from the text.

> (S1) *@Anonymous doing a great job... #not What do I pay my extortionate council taxes for? #Disgrace #OngoingProblem http://t.co/FQZUUwKSoN*

However, the reliability of the self-labeled data is an important issue. As pointed out in the pioneering work, not all tweet writers know the definition of irony (Van Hee et al., 2016b). For instance, (S2) is tagged with #irony by the writer, but it is just witty and amusing.

> (S2) *BestProAdvice @Anonymous More clean OR cleaner, never more cleaner. #irony*

When the false-alarm instances like (S2) are collected and mixed in the training and test data, the models that learn from the unreliable data may be misled, and the evaluation is also suspicious.

The other kind of unreliable data comes from the hashtags not only functioning as metadata. That is, a hashtag in a tweet may also function as a content word in its word form. For example, the hashtag #irony in (S3) is a part of the sentence "the irony of taking a break...", in contrast to the hashtag #not in (S1), which can be removed without a change of meaning.

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 771–777
Melbourne, Australia, July 15 - 20, 2018.

> (S3) *The #irony of taking a break from reading about #socialmedia to check my social media.*

When the hashtag plays as a content word in a tweet, the tweet is not a good candidate of self-labeled ironic instances because the sentence will be incomplete once the hashtag is removed.

In this work, both kinds of unreliable data, the tweets with a misused hashtag and the tweets in which the hashtag serves as a content word, are our targets to remove from the training data. Manual data cleaning is labor-intensive and inefficient (Van Hee et al., 2016a). Compared to general training data cleaning approaches (Malik and Bhardwaj, 2011; Esuli and Sebastiani, 2013; Fukumoto and Suzuki, 2004) such as boosting-based learning, this work leverages the characteristics of hashtag usages in tweets. With small amount of golden labeled data, we propose a neural network classifier for pruning the self-labeled tweets, and train an ironic detector on the less but cleaner instances. This approach is easily to apply to other NLP tasks that rely on self-labeled data.

The contributions of this work are three-fold: (1) We make an empirically study on an issue that is potentially inherited in a number of research topics based on self-labeled data. (2) We propose a model for hashtag disambiguation. For this task, the human-verified ground-truth is quite limited. To address the issue of sparsity, a novel neural network model for hashtag disambiguation is proposed. (3) The data pruning method, in which our model is applied to select reliable self-labeled data, is capable of improving the performance of irony detection.

The rest of this paper is organized as follows. Section 2 describes how we construct a dataset for disambiguating false-alarm hashtag usages based on Tweets. In Section 3, our model for hashtag disambiguation is proposed. Experimental results of hashtag disambiguation are shown in Section 4. In addition, we apply our method to prune training data for irony detection. The results are shown in Section 5. Section 6 concludes this paper.

2 Dataset

The tweets with indication hashtags such as #irony are usually collected as a dataset in previous works on irony detection. As pointed out in Section 1, the hashtags are treated as ground-truth for training and testing. To investigate the issue of false-alarm self-labeled tweets, the tweets with human verification are indispensable. In this study, we build the ground-truth based on the dataset released for SemEval 2018 Task 3,[1] which is targeted for fine-grained irony detection (Hee et al., 2018).

Hashtag	False-Alarm	Irony	Total
#not	196	346	542
#sarcasm	46	449	495
#irony	34	288	322
Total	276	1,083	1,359

Table 1: Statistics of the Ground-Truth Data.

In the SemEval dataset, the tweets with one of the three indication hashtags #not, #sarcasm, and #irony, are collected and human-annotated as one of four types: verbal irony by means of a polarity contrast, other verbal irony, situational irony, and non-ironic. In other words, the false-alarm tweets, i.e., the non-ironic tweets with indication hashtags, are distinguished from the real ironic tweets in this dataset. However, the hashtag itself has been removed in the SemEval dataset. For example, the original tweet (S1) has been modified to (S4), where the hashtag #not disappears. As a result, the hashtag information, the position and the word form of the hashtag (i.e., not, irony, or sarcasm), is missing from the SemEval dataset.

> (S4) *@Anonymous doing a great job... What do I pay my extortionate council taxes for? #Disgrace #OngoingProblem http://t.co/FQZUUwKSoN*

For hashtag disambiguation, the information of the hashtag in each tweet is mandatory. Thus, we recover the original tweets by using Twitter search. As shown in Table 1, a total of 1,359 tweets with hashtags information are adopted as the ground-truth. Note that more than 20% of self-labeled data are false-alarm, and this can be an issue when they are adopted as training or test data. For performing the experiment of irony detection in Section 5, we reserve the other 1,072 tweets in the SemEval dataset that are annotated as real ironic as the test data.

In addition to the issue of hashtag disambiguation, the irony tweets without an indication hashtag, which are regarded as non-irony instances in previous work, are another kind of misleading data for irony detection. Fortunately, the occurrence of such "false-negative" instances is insignificant due

[1] https://competitions.codalab.org/competitions/17468

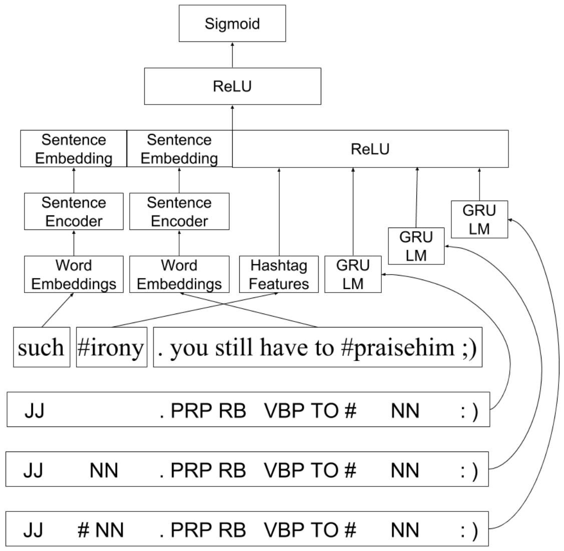

Figure 1: Overview of Our Model for Hashtag Disambiguation.

to the relatively low occurrence of irony (Davidov et al., 2010).

3 Disambiguation of Hashtags

Figure 1 shows our model for distinguishing the real ironic tweets from the false-alarm ones. Given an instance with the hashtag #irony is given, the preceding and the following word sequences of the hashtag are encoded by separate sub-networks, and both embeddings are concatenated with the handcrafted features and the probabilities of three kinds of part-of-speech (POS) tag sequences. Finally, the sigmoid activation function decides whether the instance is real ironic or false-alarm. The details of each component will be presented in the rest of this section.

Word Sequences: The word sequences of the context preceding and following the targeting hashtag are separately encoded by neural network sentence encoders. The Penn Treebank Tokenizer provided by NLTK (Bird et al., 2009) is used for tokenization. As a result, each of the left and the right word sequences is encoded as a embedding with a length of 50.

We experiments with convolution neural network (CNN) (Kim, 2014), gated recurrent unit (GRU) (Cho et al., 2014), and attentive-GRU for sentence encoding. CNN for sentence classification has been shown effective in NLP applications such as sentiment analysis (Kim, 2014). Classifiers based on recurrent neural network (RNN) have also been applied to NLP, especially for sequential modeling. For irony detection, one of the state-of-the-art models is based on the attentive RNN (Huang et al., 2017). The first layer of the CNN, the GRU, and the attenive-GRU model is the 300-dimensional word embedding that is initialized by using the vectors pre-trained on Google News dataset.[2]

Handcrafted Features: We add the handcrafted features of the tweet in the one-hot representation. The features taken into account are listed as follows. (1) Lengths of the tweet in words and in characters. (2) Type of the target hashtag (i.e. #not, #sarcasm, or #irony). (3) Number of all hashtags in the tweet. (4) Whether the targeting hashtag is the first token in the tweet. (5) Whether the targeting hashtag is the last token in the tweet. (6) Whether the targeting hashtag is the first hashtag in the tweet since a tweet may contain more than one hashtag. (7) Whether the targeting hashtag is the last hashtag in the tweet. (8) Position of the targeting hashtag in terms of tokens. If the targeting hashtag is the ith token of the tweet with $|w|$ tokens, and this feature is $\frac{i}{|w|}$. (9) Position of the targeting hashtag in all hashtags in the tweet. It is computed as $\frac{j}{|h|}$ where the targeting hashtag is the jth hashtag in the tweet that contains $|h|$ hashtags.

Language Modeling of POS Sequences: As mentioned in Section 1, a kind of false-alarm hashtag usages is the case that the hashtag also functions as a content word. In this paper, we attempt to measure the grammatical completeness of the tweet with and without the hashtag. Therefore, language model on the level of POS tagging is used. As shown in Figure 1, POS tagging is performed on three versions of the tweet, and based on that three probabilities are measured and taken into account: 1) $p_{\bar{h}}$: the tweet with the whole hashtag removed. 2) $p_{\bar{s}}$: the tweet with the hash symbol # removed only. 3) p_t: the original tweet. Our idea is that a tweet will be more grammatical complete with only the hash symbol removed if the hashtag is also a content word. On the other hand, the tweet will be more grammatical complete with the whole hashtag removed since the hashtag is a metadata.

To measure the probability of the POS tag sequence, we integrate a neural network-based language model of POS sequence into our model. RNN-based language models are reportedly capa-

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

ble of modeling the longer dependencies among the sequential tokens (Mikolov et al., 2011). Two millions of English tweets that are entirely different from those in the training and test data described in Section 2 are collected and tagged with POS tags. We train a GRU language model on the level of POS tags. In this work, all the POS tagging is performed with the Stanford CoreNLP toolkit (Manning et al., 2014).

4 Experiments

We compare our model with popular neural network-based sentence classifiers including CNN, GRU, and attentive GRU. We also train a logistic regression (LR) classifier with the hand-crafted features introduced in Section 3. For the imbalance data, we assign class-weights inversely proportional to class frequencies. Five-fold cross-validation is performed. Early-stop is employed with a patience of 5 epoches. In each fold, we further keep 10% of training data for tuning the model. The hidden dimension is 50, the batch size is 32, and the Adam optimizer is employed (Kingma and Ba, 2014).

Table 2 shows the experimental results reported in Precision (P), Recall (R), and F-score (F). Our goal is to select the real ironic tweets for training the irony detection model. Thus, the real ironic tweets are regarded as positive, and the false-alarm ones are negative. We apply t-test for significance testing. The vanilla GRU and attentive GRU are slightly superior to the logistic regression model. The CNN model performs the worst in this task because it suffers from over-fitting problem. We explored a number of layouts and hyperparameters for the CNN model, and consistent results are observed.

Our method is evaluated with either CNN, GRU, or attentive GRU for encoding the context preceding and following the targeting hashtag. By integrating various kinds of information, our method outperforms all baseline models no matter which encoder is used. The best model is the one integrating the attentive GRU encoder, which is significantly superior to all baseline models ($p < 0.05$), achieves an F-score of 88.49%,

To confirm the effectiveness of the language modeling of POS sequence, we also try to exclude the GRU language model from our best model. Experimental results show that the addition of language model significantly improves the performance ($p < 0.05$). As shown in the last row of Table 2, the F-score is dropped to 84.17%.

Model	Encoder	P	R	F
LR	N/A	91.43%	75.81%	82.89%
CNN	N/A	89.16%	56.97%	69.52%
GRU	N/A	90.75%	77.01%	83.32%
Att.GRU	N/A	87.97%	79.69%	83.62%
Our Method	CNN	90.35%	83.84%	86.97%
Our Method	GRU	90.90%	78.39%	84.18%
Our Method	Att.GRU	90.86%	86.24%	88.49%
w/o LM	Att.GRU	88.17%	80.52%	84.17%

Table 2: Results of Hashtag Disambiguation.

From the data, we observe that the instances whose $p_{\bar{s}} \gg p_{\bar{h}}$ usually contain a indication hashtag function as a content word, and vice versa. For instances, (S5) and (S6) show the instances with the highest and the lowest $\frac{p_{\bar{s}}}{p_{\bar{h}}}$, respectively.

(S5) *when your #sarcasm is so advanced people actually think you are #stupid ..*

(S6) *#mtvstars justin bieber #net #not #fast*

5 Irony Detection

We employ our model to prune self-labeled data for irony detection. As prior work did, we collect a set of tweets that contain indication hashtags as (pseudo) positive instances and also collect a set of tweets that do not contain indication hashtags as negative instances. For each positive instance, our model is performed to predict whether it is a real ironic tweet or false-alarm ones, and the false-alarm ones are discarded.

After pruning, a set of 14,055 tweets containing indication hashtags have been reduced to 4,617 reliable positive instances according to our model. We add an equal amount of negative instances randomly selected from the collection of the tweets that do not contain indication hashtags. As a result, the prior- and the post-pruning training data, in the sizes of 28,110 and 9,234, respectively, are prepared for experiments. The dataflow of the training data pruning is shown in Figure 2.

For evaluating the effectiveness of our pruning method, we implement a state-of-the-art irony detector (Huang et al., 2017), which is based on attentive-RNN classifier, and train it on the prior- and the post-pruned training data.

The test data is made by the procedure as follows. The positive instances in the test data are taken from the 1,072 human-verified ironic tweets

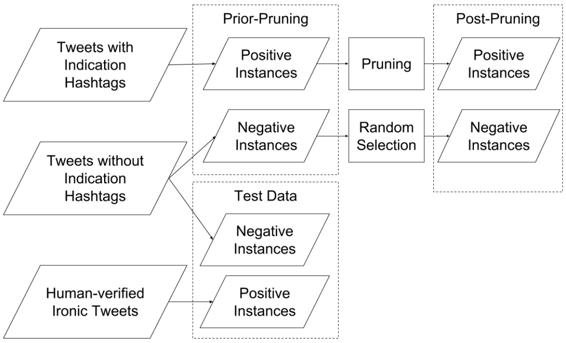

Figure 2: Dataflow of the Training Data Pruning for Irony Detection.

Training Data	Size	P	R	F
Prior-Pruning	28,110	79.04%	84.05%	81.46%
Post-Pruning	9,234	80.83%	85.35%	83.03%
Human Verified	2,166	86.35%	66.70%	75.26%

Table 3: Performance of Irony Detection.

that are reserved for irony detection as mentioned in Section 2. The negative instances in the test data are obtained from the tweets that do not contain indication hashtags. Note that the negative instances in the test data are isolated from those in the training data. Experimental results confirm the benefit of pruning. As shown in Table 3, the irony detection model trained on the less, but cleaner data significantly outperforms the model that is trained on all data ($p < 0.05$).

We compare our pruning method with an alternative approach that trains the irony detector on the human-verified data directly. Under this circumstances, the 1,083 ironic instances for training our hashtag disambiguation model are currently mixed with an equal amount of randomly sampled negative instances, and employed to train the irony detector. As shown in the last row of Table 3, the irony detector trained on the small data does not compete with the models that are trained on larger amount of self-labeled data. In other words, our data pruning strategy forms a semi-supervised learning that benefits from both self-labeled data and human annotation. Note that this task and the dataset are different from those of the official evaluation of SemEval 2018 Task 3, so the experimental results cannot be directly compared.

The calibrated confidence output by the sigmoid layer of our hashtag disambiguation model can be regarded as a measurement of the reliability of an instance (Niculescu-Mizil and Caruana, 2005; Guo et al., 2017). Thus, we can sort all self-labeled data by their calibrated confidence and control the size of training set by adjusting the threshold. The higher the threshold value is set, the less the training instances remain. Figure 3 shows the performances of the irony detector trained on the data filtered with different threshold values. For each threshold value, the bullet symbol (●) indicates the size of training data, and the bar indicates the F-score achieved by the irony detector trained on those data. The best result achieved by the irony detector trained on the 9,234 data filtered by our model with the default threshold value (0.5). This confirms that our model is able to select useful training instances in a strict manner.

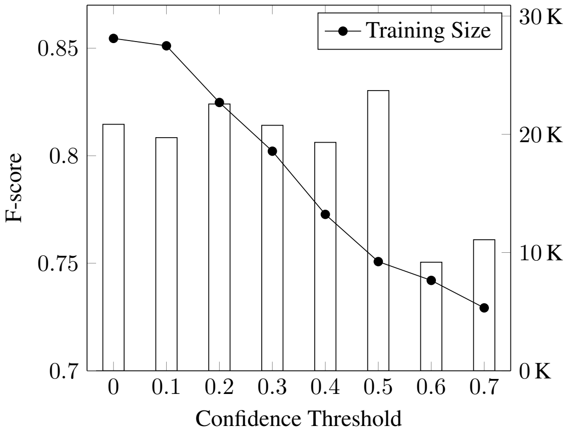

Figure 3: Performance of Irony Detection with Different Threshold Values for Data Pruning.

6 Conclusion

Self-labeled data is an accessible and economical resource for a variety of learning-based applications. However, directly using the labels made by the crowd as ground-truth for training and testing may lead to inaccurate performance due to the reliability issue. This paper addresses this issue in the case of irony detection by proposing a model to remove two kinds of false-alarm tweets from the training data. Experimental results confirm that the irony detection model benefits from the less, but cleaner training data. Our approach can be applied to other topics that rely on self-labeled data.

Acknowledgements

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST-105-2221-E-002-154-MY3, MOST-106-2923-E-002-012-MY3 and MOST-107-2634-F-002-011-.

References

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 519–535, Vancouver, Canada. Association for Computational Linguistics.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcasm in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116, Uppsala, Sweden. Association for Computational Linguistics.

Andrea Esuli and Fabrizio Sebastiani. 2013. Improving text classification accuracy by training label cleaning. *ACM Trans. Inf. Syst.*, 31(4):19:1–19:28.

Fumiyo Fukumoto and Yoshimi Suzuki. 2004. Correcting category errors in text classification. In *Proceedings of Coling 2004*, pages 868–874, Geneva, Switzerland. COLING.

Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan. 2015. Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1003–1012, Lisbon, Portugal. Association for Computational Linguistics.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330, International Convention Centre, Sydney, Australia. PMLR.

Cynthia Van Hee, Els Lefever, and Vronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Yu-Hsiang Huang, Hen-Hsen Huang, and Hsin-Hsi Chen. 2017. Irony detection with attentive recurrent neural networks. In *European Conference on Information Retrieval*, pages 534–540. Springer.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

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

Hassan H. Malik and Vikas S. Bhardwaj. 2011. Automatic training data cleaning for text classification. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, pages 442–449, Washington, DC, USA. IEEE Computer Society.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukar Burget, and Jan Cernocky. 2011. Rnnlm-recurrent neural network language modeling toolkit. In *Proc. of the 2011 ASRU Workshop*, pages 196–201.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 625–632, New York, NY, USA. ACM.

Lotem Peled and Roi Reichart. 2017. Sarcasm sign: Interpreting sarcasm with sentiment based monolingual machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1690–1700, Vancouver, Canada. Association for Computational Linguistics.

Ashequl Qadir and Ellen Riloff. 2014. Learning emotion indicators from tweets: Hashtags, hashtag patterns, and phrases. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1203–1209, Doha, Qatar. Association for Computational Linguistics.

Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. 2017. A dataset for multi-target stance detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 551–557, Valencia, Spain. Association for Computational Linguistics.

Emilio Sulis, Delia Irazú Hernández Farías, Paolo Rosso, Viviana Patti, and Giancarlo Ruffo. 2016. Figurative messages and affect in twitter. *Know.-Based Syst.*, 108(C):132–143.

Cynthia Van Hee, Els Lefever, and Veronique Hoste. 2016a. Exploring the realization of irony in twitter data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC '16)*, pages 1795–1799. European Language Resources Association (ELRA).

Cynthia Van Hee, Els Lefever, and Veronique Hoste. 2016b. Monday mornings are my fave :) #not exploring the automatic recognition of irony in english tweets. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2730–2739, Osaka, Japan. The COLING 2016 Organizing Committee.