IIIDYT at SemEval-2018 Task 3: Irony detection in English tweets

Edison Marrese-Taylor[1*], Suzana Ilic[2*], Jorge A. Balazs[1*], Helmut Prendinger[2], Yutaka Matsuo[1]
Graduate School of Engineering, The University of Tokyo, Japan[1]
emarrese,jorge,matsuo@weblab.t.u-tokyo.ac.jp
National Institute of Informatics, Tokyo, Japan[2]
suzana,helmut@nii.ac.jp
[*]Authors contributed equally to this work.

Abstract

In this paper we introduce our system for the task of Irony detection in English tweets, a part of SemEval 2018. We propose representation learning approach that relies on a multi-layered bidirectional LSTM, without using external features that provide additional semantic information. Although our model is able to outperform the baseline in the validation set, our results show limited generalization power over the test set. Given the limited size of the dataset, we think the usage of more pre-training schemes would greatly improve the obtained results.

1 Introduction

Sentiment analysis and emotion recognition, as two closely related subfields of affective computing, play a key role in the advancement of artificial intelligence (Cambria et al., 2017). However, the complexity and ambiguity of natural language constitutes a wide range of challenges for computational systems.

In the past years irony and sarcasm detection have received great traction within the machine learning and NLP community (Joshi et al., 2016), mainly due to the high frequency of sarcastic and ironic expressions in social media. Their linguistic collocation inclines to flip polarity in the context of sentiment analysis, which makes machine-based irony detection critical for sentiment analysis (Poria et al., 2016; Van Hee et al., 2015). Irony is a profoundly pragmatic and versatile linguistic phenomenon. As its foundations usually lay beyond explicit linguistic patterns in reconstructing contextual dependencies and latent meaning, such as shared knowledge or common knowledge (Joshi et al., 2016), automatically detecting it remains a challenging task in natural language processing.

In this paper, we introduce our system for the shared task of Irony detection in English tweets, a part of the 2018 SemEval (Van Hee et al., 2018). We note that computational approaches to automatically detecting irony often deploy expensive feature-engineered systems which rely on a rich body of linguistic and contextual cues (Bamman and Smith, 2015; Joshi et al., 2015). The advent of Deep Learning applied to NLP has introduced models that have succeeded in large part because they learn and use their own continuous numeric representations (Hinton, 1984) of words (Mikolov et al., 2013), offering us the dream of forgetting manually-designed features. To this extent, in this paper we propose a representation learning approach for irony detection, which relies on a bidirectional LSTM and pre-trained word embeddings.

2 Data and pre-processing

For the shared task, a balanced dataset of 2,396 ironic and 2,396 non-ironic tweets is provided. The ironic corpus was constructed by collecting self-annotated tweets with the hashtags *#irony*, *#sarcasm* and *#not*. The tweets were then cleaned and manually checked and labeled, using a fine-grained annotation scheme (Van Hee et al., 2015). The corpus comprises different types of irony:

- Verbal irony (polarity contrast): 1,728 instances
- Other types of verbal irony: 267 instances.
- Situational irony: 401 instances

Verbal irony is often referred to as an utterance that conveys the opposite meaning of what of literally expressed (Grice, 1975; Wallace, 2015), e.g. *I love annoying people*. Situational irony appears

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 537–540
New Orleans, Louisiana, June 5–6, 2018.

in settings, that diverge from the expected (Lucariello, 1994), e.g. *an old man who won the lottery and died the next day*. The latter does not necessarily exhibit polarity contrast or other typical linguistic features, which makes it particularly difficult to classify correctly.

For the pre-processing we used the Natural Language Toolkit (Loper and Bird, 2002). As a first step, we removed the following words and hashtagged words: *not*, *sarc*, *sarcasm*, *irony*, *ironic*, *sarcastic* and *sarcast*, in order to ascertain a clean corpus without topic-related triggers. To ease the tokenizing process with the NLTK TweetTokenizer, we replaced two spaces with one space and removed usernames and urls, as they do not generally provide any useful information for detecting irony.

We do not stem or lowercase the tokens, since some patterns within that scope might serve as an indicator for ironic tweets, for instance a word or a sequence of words, in which all letters are capitalized (Tsur et al., 2010).

3 Proposed Approach

The goal of the subtask A was to build a binary classification system that predicts if a tweet is ironic or non-ironic. In the following sections, we first describe the dataset provided for the task and our pre-processing pipeline. Later, we lay out the proposed model architecture, our experiments and results.

3.1 Word representation

Representation learning approaches usually require extensive amounts of data to derive proper results. Moreover, previous studies have shown that initializing representations using random values generally causes the performance to drop. For these reasons, we rely on pre-trained word embeddings as a means of providing the model the adequate setting. We experiment with GloVe[1] (Pennington et al., 2014) for small sizes, namely 25, 50 and 100. This is based on previous work showing that representation learning models based on convolutional neural networks perform well compared to traditional machine learning methods with a significantly smaller feature vector size, while at the same time preventing over-fitting and accelerates computation (e.g (Poria et al., 2016).

[1] nlp.stanford.edu/projects/glove

GloVe embeddings are trained on a dataset of 2B tweets, with a total vocabulary of 1.2 M tokens. However, we observed a significant overlap with the vocabulary extracted from the shared task dataset. To deal with out-of-vocabulary terms that have a frequency above a given threshold, we create a new vector which is initialized based on the space described by the infrequent words in GloVe. Concretely, we uniformly sample a vector from a sphere centered in the centroid of the 10% less frequent words in the GloVe vocabulary, whose radius is the mean distance between the centroid and all the words in the low frequency set. For the other case, we use the special *UNK* token.

To maximize the knowledge that may be recovered from the pre-trained embeddings, specially for out-of-vocabulary terms, we add several token-level and sentence-level binary features derived from simple linguistic patterns, which are concatenated to the corresponding vectors.

Word-level features

1. If the token is fully lowercased.
2. If the Token is fully uppercased.
3. If only the first letter is capitalized.
4. If the token contains digits.

Sentence-level features

1. If any token is fully lowercased.
2. If any token is fully uppercased.
3. If any token appears more than once.

3.2 Model architecture

Recurrent neural networks are powerful sequence learning models that have achieved excellent results for a variety of difficult NLP tasks (Ian Goodfellow, Yoshua Bengio, 2017). In particular, we use the last hidden state of a bidirectional LSTM architecture (Hochreiter and Urgen Schmidhuber, 1997) to obtain our tweet representations. This setting is currently regarded as the state-of-the-art (Barnes et al., 2017) for the task on other datasets. To avoid over-fitting we use Dropout (Srivastava et al., 2014) and for training we set binary cross-entropy as a loss function. For evaluation we use our own wrappers of the the official evaluation scripts provided for the shared tasks, which are based on accuracy, precision, recall and F1-score.

4 Experimental setup

Our model is implemented in PyTorch (Paszke et al., 2017), which allowed us to easily deal with the variable tweet length due to the dynamic nature of the platform. We experimented with different values for the LSTM hidden state size, as well as for the dropout probability, obtaining best results for a dropout probability of 0.1 and 150 units for the the hidden vector. We trained our models using 80% of the provided data, while the remaining 20% was used for model development. We used Adam (Kingma and Ba, 2015), with a learning rate of 0.0001 and early stopping when performance did not improve on the development set. Using embeddings of size 100 provided better results in practice. Our final best model is an ensemble of four models with the same architecture but different random initialization.

To compare our results, we use the provided baseline, which is a non-parameter optimized linear-kernel SVM that uses TF-IDF bag-of-word vectors as inputs. For pre-processing, in this case we do not preserve casing and delete English stopwords.

5 Results

To understand how our strategies to recover more information from the pre-trained word embeddings affected the results, we ran ablation studies to compare how the token-level and sentence-level features contributed to the performance. Table 1 summarizes the impact of these features in terms of F1-score on the validation set.

Feature	Yes	No
Token-level	0.6843	0.7008
Sentence-level	0.6848	0.6820

Table 1: Results of our ablation study for binary features in terms of F1-Score on the validation set.

We see that sentence-level features had a positive yet small impact, while token-level features seemed to actually hurt the performance. We think that since the task is performed at the sentence-level, probably features that capture linguistic phenomena at the same level provide useful information to the model, while the contributions of other finer granularity features seem to be too specific for the model to leverage on.

Table 2 summarizes our best single-model results on the validation set (20% of the provided data) compared to the baseline, as well as the official results of our model ensemble on the test data.

Split	Accuracy	Precision	Recall	F1-score
Baseline Valid	0.6375	0.6440	0.6096	0.6263
Ours Valid	0.6610	0.6369	0.8447	0.7262
Ours Test	0.3520	0.2568	0.3344	0.2905

Table 2: Summary of the obtained best results on the valid/test sets.

Out of 43 teams our system ranked 421st with an official F1-score of 0.2905 on the test set. Although our model outperforms the baseline in the validation set in terms of F1-score, we observe important drops for all metrics compared to the test set, showing that the architecture seems to be unable to generalize well. We think these results highlight the necessity of an ad-hoc architecture for the task as well as the relevance of additional information. The work of Felbo et al. (2017) offers interesting contributions in these two aspects, achieving good results for a range of tasks that include sarcasm detection, using an additional attention layer over a BiLSTM like ours, while also pre-training their model on an emoji-based dataset of 1246 million tweets.

Moreover, we think that due to the complexity of the problem and the size of the training data in the context of deep learning better results could be obtained with additional resources for pre-training. Concretely, we see transfer learning as one option to add knowledge from a larger, related dataset could significantly improve the results (Pan and Yang, 2010). Manually labeling and checking data is a vastly time-consuming effort. Even if noisy, collecting a considerably larger self-annotated dataset such as in Khodak et al. (2017) could potentially boost model performance.

6 Conclusion

In this paper we presented our system to SemEval-2018 shared task on irony detection in English tweets (subtask A), which leverages on a BiLSTM and pre-trained word embeddings for representation learning, without using human-engineered features. Our results showed that although the generalization capabilities of the model are limited, there are clear future directions to improve. In particular, access to more training data and the deployment of methods like transfer learning seem to be promising directions for future research in representation learning-based sarcasm detection.

References

David Bamman and Noah A Smith. 2015. Contextualized sarcasm detection on twitter. *Icwsm (International AAAI Conference on Web and Social Media)*, pages 574–577.

Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. 2017. Assessing state-of-the-art sentiment models on state-of-the-art sentiment datasets. pages 2–12.

Erik Cambria, Dipankar Das, Sivaji Bandyopadhyay, and Antonio Feraco. 2017. *Guide to Sentiment Analysis*. Springer International Publishing AG 2017, Cham, Switzerland.

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.

H. Paul Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics*, volume 3, pages 41–58. Academic Press, New York.

Geoffrey E Hinton. 1984. Distributed representations.

Sepp Hochreiter and J Urgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Aaron Courville Ian Goodfellow, Yoshua Bengio. 2017. *Deep Learning*, volume 521. MIT Press.

Aditya Joshi, Pushpak Bhattacharyya, and Mark James Carman. 2016. Automatic sarcasm detection: A survey. *ACM Computing Surveys*, V.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, 51(4):757–762.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2017. A large self-annotated corpus for sarcasm.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*, San Diego.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit.

Joan Lucariello. 1994. Situational irony: A concept of events gone awry. *Journal of Experimental Psychology: General*, 123(2):129–145.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Adam Paszke, Gregory Chanan, Zeming Lin, Sam Gross, Edward Yang, Luca Antiga, and Zachary Devito. 2017. Automatic differentiation in pytorch. *Advances in Neural Information Processing Systems 30*, (Nips):1–4.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Oren Tsur, Ari Rappoport, and Dmitry Davidov. 2010. Icwsm a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. *International AAAI Conference on Weblogs and Social Media*, (9):162–169.

Cynthia Van Hee, Els Lefever, and Veronique Hoste. 2015. Guidelines for annotating irony in social media text. *LT3 Technical Report*, 15(2).

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-2018, New Orleans, LA, USA. Association for Computational Linguistics.

Byron C. Wallace. 2015. Computational irony: A survey and new perspectives. *Artificial Intelligence Review*, 43(4):467–483.