Sarcasm Detection Using an Ensemble Approach

Jens Lemmens and **Ben Burtenshaw** and **Ehsan Lotfi**
and **Ilia Markov** and **Walter Daelemans**
CLiPS, University of Antwerp
Lange Winkelstraat 40
2000, Antwerp (Belgium)
firstname.lastname@uantwerpen.be

Abstract

We present an ensemble approach for the detection of sarcasm in Reddit and Twitter responses in the context of The Second Workshop on Figurative Language Processing held in conjunction with ACL 2020[1]. The ensemble is trained on the predicted sarcasm probabilities of four component models and on additional features, such as the sentiment of the comment, its length, and source (Reddit or Twitter) in order to learn which of the component models is the most reliable for which input. The component models consist of an LSTM with hashtag and emoji representations; a CNN-LSTM with casing, stop word, punctuation, and sentiment representations; an MLP based on Infersent embeddings; and an SVM trained on stylometric and emotion-based features. All component models use the two conversational turns preceding the response as context, except for the SVM, which only uses features extracted from the response. The ensemble itself consists of an adaboost classifier with the decision tree algorithm as base estimator and yields F1-scores of 67% and 74% on the Reddit and Twitter test data, respectively.

1 Introduction

In this paper, an ensemble approach for the detection of sarcasm in social media data is described. The ensemble was designed in the context of The Second Workshop on Figurative Language Processing held in conjunction with ACL 2020[1]. It was the goal of the shared task to create a robust sarcasm detection model for tweets and Reddit comments and investigate the role of conversational context in automatic sarcasm detection models.

Detecting sarcasm can be a challenging task, not only for machines, but also for humans, because sarcasm is subjective and culturally dependent, and because an utterance on its own can be both sarcastic and non-sarcastic (Ghosh et al., 2018; Joshi et al., 2017). Context is therefore vital for a correct interpretation of a comment on social media (Wallace et al., 2014). For example, "Well done, guys!" generally has a positive meaning, whereas it is used sarcastically in the context of a social media post about the governments mismanagement. Therefore, conversational context is used in our approach described below to identify sarcasm.

[1] https://sites.google.com/view/figlang2020/

2 Related research

In this section, recent advances and papers related to sarcasm detection are described. The first advance is related to automatic annotation methods where the annotators use computational methods to obtain the labels (Joshi et al., 2017), for instance by searching for "#sarcasm" in tweets (e.g. González-Ibáñez et al. (2011)). Automatic labelling is often preferred to manual labelling, because it is faster, cheaper, allows for the creation of larger data sets, and because the author of an utterance knows best whether it was meant sarcastically or not. Note that automatically annotated data can contain more false positives and/or false negatives than manually labeled data if the labeling method is not robust enough. An automatic method was used to label the data used in the present study and a more detailed description of that data can be found in Section 3.

A second advance in the field of sarcasm detection are pattern-based features (Joshi et al., 2017). This term refers to using linguistic patterns as features. For example, Riloff et al. (2013) use the pattern "presence of a positive verb and a negative situation" (e.g., "I love queueing for hours") as a feature. They hypothesized that this pattern is highly indicative of sarcasm. Their approach achieved an F1-score of 51%.

Similarly, Van Hee et al. (2018) hypothesized that sentiment incongruity within an utterance signifies sarcasm. They did not only consider explicit expressions of sentiment, but also attempted to deal with sentiment implicitly embedded in world

Proceedings of the Second Workshop on Figurative Language Processing, pages 264–269

https://doi.org/10.18653/v1/P17

knowledge. To achieve this, the annotators gathered all real-world concepts that carried an implicit sentiment and labeled them with either a "positive" or "negative" sentiment label (e.g., "going to the dentist", which is usually associated with a negative sentiment). Three approaches were then proposed that implemented these implicit sentiment labels. None of these approaches, however, outperformed the baseline (70% F1-score). Thus, although sarcasm can be seen as an expression of sentiment, this study showed that successfully implementing sentiment in a classifier is not trivial.

A third advance in the sarcasm detection field is using context as feature (Joshi et al., 2017). Three types of context can be distinguished: author context, e.g., (Joshi et al., 2016), conversational context, e.g., (Wang et al., 2015), and topical context, e.g., (Wang et al., 2015). Author context refers to the name of the author of the comment. The intuition behind using this type of context is that one individual uses sarcasm more regularly than another individual and can therefore improve the performance of sarcasm detection models. Conversational context, on the other hand, refers to the conversational turns preceding (or following) the relevant utterance. As mentioned in the introduction, this type of context can clarify whether an utterance is sarcastic or not if that utterance can be perceived as both. Finally, topical context is used, because it is hypothesized that certain topics (e.g., religion or politics) trigger more sarcastic responses than other topics.

Further, previous research has shown that for statistical models, SVM performs best (Joshi et al., 2017, 2016; Riloff et al., 2013). Conversely, the most successful deep learning algorithms are long-short term memory (LSTM) networks and convolutional neural networks (CNN) (Ghosh and Veale, 2016; Amir et al., 2016).

More recently, Ghosh et al. (2018) presented an LSTM network with sentence-level attention heads that achieved the state-of-the-art performance: they reported F1-scores of 84% for sarcastic comments and 83% for non-sarcastic comments. The goal of their study was to investigate the role of conversational context in sarcasm detection and their experiments suggested that conversational context significantly improves the performance of their model.

Joshi et al. (2017) provide a survey of previous sarcasm detection studies and can be consulted for a more extensive overview of related research.

3 Data

The training data comprises 5,000 tweets and 4,400 Reddit comments, and was annotated automatically (Khodak et al., 2018; Ghosh et al., 2018). Each comment or "response" is accompanied by its context, i.e., an ordered list of all previous comments in the conversation, and a binary label indicating whether the response is sarcastic. Both the Twitter and Reddit data are balanced.

The test data contains 1,800 tweets and the same number of Reddit comments. Similar to the training data, the test responses are accompanied by their conversational context, and are balanced. In all instances, user mentions and URLs are replaced with placeholders: "@USER" and "<URL>", respectively. All data that was used in the present study was provided by the organizers of the workshop. However, the participating teams were allowed to collect extra data if desired.

4 Methodology

Four component models were used to construct the ensemble classifier. All of these models use conversational context as feature, with the exception of the SVM model described in Section 4.1.3, which focuses only on stylometric and emotion-based properties of the response. All other models use the two conversational turns preceding the response as context, since this was the minimum amount of context that was provided for each response.

4.1 Component models

4.1.1 LSTM

Preliminary studies showed that non-word features have a noticeable effect on sarcasm transparency. For example, hashtags and emojis were used as signifiers to modify the rest of a sentence. A bidirectional LSTM model was used to recognize these modifications in relation to the main embedded vocabulary and predict binary sarcasm (Zhou et al., 2016).

Context and response words were vectorized using pretrained GloVe embeddings (Pennington et al., 2014). Emojis were embedded using Emoji2Vec (Eisner et al., 2016). All words were then further embedded using an RNN model, trained on tweets from the Chirps corpus to predict hashtags (Shwartz et al., 2017); comparable to a sentence summarization task (Jing et al., 2003), which contributed to the Reddit task as well as Twitter, by using base text alone.

These 3 embedding layers were combined for the bidirectional LSTM to iterate over. To mitigate overfitting, dropout was applied two times and optimized: (i) to the embedding layers, and (ii) within the LSTM layers. Finally the concatenated output was passed to a sigmoid layer for prediction.

4.1.2 CNN-LSTM

This model uses word embeddings of the response and context pretrained with GloVe embeddings (Pennington et al., 2014)), and punctuation, casing, sentiment and stop word features. The punctuation features contain the absolute and relative numbers of exclamation marks, quotation marks, question marks, periods, hashtags, and at-symbols in the response. Conversely, the casing features comprise the absolute and relative numbers of uppercase, lowercase and other characters (e.g. digits) in the response. The sentiment features, obtained with NLTK's Vader sentiment analyzer (Bird et al., 2009), are represented by a negative, neutral, positive, and global sentiment score of both the response and its context. Finally, stop word features were obtained by constructing a count vectorizer with scikit-learn (Pedregosa et al., 2011) out of NLTK's English stop word list.

The response and context word embeddings were twice fed to a sequence of a convolutional layer with max pooling, and to a bidirectional LSTM layer. The other feature vectors were each passed to a dense layer. To avoid overfitting, dropout was applied and optimized after each embedding, convolutional, LSTM, and dense layer. Finally, the outputs of all of the above were concatenated and passed to a sigmoid layer for prediction.

4.1.3 SVM

In this approach, the response messages were represented through a combination of part-of-speech (POS) tags (obtained using the StanfordNLP library (Qi et al., 2018)), function words (i.e., words belonging to the closed syntactic classes[2]), and emotion-based features from the NRC emotion lexicon (Mohammad and Turney, 2013). From this representation, n-grams (with n from 1 to 3) were built. Character n-gram features (with n from 1 to 3) were added as a separate feature vector. This approach captures the stylometric and emotion characteristics of a textual content and is described in detail in (Markov et al., 2020).

[2]https://universaldependencies.org/u/pos/

The features were weighted using the term frequency (tf) weighting scheme and fed to liblinear SVM with optimized parameters (the optimal liblinear classifier parameters were selected: penalty parameter (C), loss function (loss), and tolerance for stopping criteria (tol) (based on grid search). The liblinear scikit-learn (Pedregosa et al., 2011) implementation of SVM was used.

4.1.4 MLP

This model consists of simple multi-layer perceptron (MLP) classifier based on sentence embeddings from the Infersent model developed by Facebook (Conneau et al., 2017). Infersent is trained on natural language inference data, which is a motivation to use this model in our ensemble approach, since it might spot the logical discrepancies that often play a role in creating and detecting sarcasm. Infersent works with GloVe or Fasttext word embeddings as input and gives a 4092-dimensional sentence embedding. For this task we concatenated the response and context embeddings (with GloVe) and fed the resulting 8184-dimensional vector to an MLP with Relu non-linearity and a sigmoid at the end for classification. This was attempted with different architectures among which a [8184-2048-128-16-2] composition showed the best results.

Before they were converted to embeddings, the responses and their context were preprocessed as follows: hashtags were added as descriptions at the end of the string and links were removed.

4.2 Ensemble

We used 10-fold cross-validation to train the component models. For each fold, the predicted validation labels were stored in a dataframe. This allowed us to collect predictions for all comments in the training data without the models being trained on the comments for which they predicted the label. These predictions were then used to train the ensemble model, which consisted of a decision tree classifier implemented as the base estimator in a scikit-learn adaboost classifier. In addition to the predicted labels, the character length of the response and context, their source (Twitter or Reddit) and NLTK's Vader sentiment scores for the response and its two preceding turns were used as features, so that the ensemble could learn which component model was the most reliable and for which input (e.g., long positive tweet as response and short negative tweet as context).

5 Results

In this section, the performance of the component models and of the ensemble model are described. The models were evaluated on the Reddit test set and Twitter test set separately, and F1-score was used as the official evaluation metric.

In Table 1, the 10-fold cross-validation precision, recall, and F1-score of the component models on the training data can be found. The ensemble itself yields precision, recall, and F1 scores of 77.2%, 76.9% and 76.9% under 10-fold cross-validation (Reddit and Twitter combined). Table 2 represents an overview of the scores obtained on the held-out evaluation set by the different component models and the ensemble architecture.

From these results, it can be concluded that the ensemble model has higher precision, recall, and F1-score than the models in isolation. This suggests that the ensemble does not simply predict the same label as the overall best performing component model, but learns which model performs best and when. What component model is globally the most robust, depends on the type of data (Reddit or Twitter) and on the setting (training or test data). Nevertheless, each component model contributes to the results and therefore seems to capture different sarcasm characteristics, as evidenced by the increase in performance when all the models are combined through the ensemble and by an ablation study we conducted: removing any of the component models results in a decrease in performance.

Further, the results show that both official test sets contain comments that are, on average, more challenging to classify than the training data, since the 10-fold cross-validation scores (Table 1) are substantially higher than the scores on the official test sets (Table 2). Moreover, it can be observed that all models achieve lower scores on Reddit comments than on Twitter comments. Since not only recall, but also precision are lower for all models, this does not only suggest that sarcasm is more challenging to detect, but that it is generally more difficult to distinguish between non-sarcastic and sarcastic utterances in Reddit comments. One plausible explanation for this imbalance is that tweets are limited in length, whereas Reddit comments are not. Therefore, the models may have more difficulties with interpreting the context in the longer Reddit comments, resulting in a lower performance. However, more research is needed to determine why Reddit comments are more challenging to classify than tweets.

	Reddit			Twitter		
Model	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
LSTM	64.3	64.0	63.8	75.6	75.2	75.2
CNN	62.1	62.0	62.0	76.1	75.9	75.9
SVM	64.2	64.2	64.2	74.5	74.4	74.4
MLP	65.1	65.3	65.1	74.1	74.9	73.9

Table 1: Precision (%), recall (%), and F1-score (%) of the component models on the training data under 10-fold cross-validation.

	Reddit			Twitter		
Model	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
LSTM	63.6	63.7	63.5	67.7	68.0	67.5
CNN	59.1	59.1	59.1	67.1	67.2	67.0
SVM	62.0	62.0	62.0	66.6	66.7	66.5
MLP	60.2	61.9	58.6	68.3	68.3	68.3
Ens.	**67.0**	**67.7**	**66.7**	**74.1**	**74.6**	**74.0**

Table 2: Precision (%), recall (%), and F1-score (%) of the models on the official evaluation data.

6 Conclusion

We described an ensemble approach for sarcasm detection in Reddit and Twitter comments. The model consists of an adaboost classifier with the decision tree algorithm as base estimator and learns the sarcasm probabilities predicted by four different component models: an LSTM model that uses word, emoji and hashtag representations; a model that uses CNNs and LSTM networks to learn word embeddings, and dense networks to learn punctuation, casing, sentiment and stop word features; an MLP based on Infersent embeddings; and an SVM approach that captures the stylometric and emotional characteristics of sarcastic content. All component models (except SVM) use conversational context to make predictions, namely the two turns preceding the response.

The sarcasm probabilities used to train the ensemble were obtained by training the component models using 10-fold cross-validation and saving the labels predicted for the validation set in each fold. In order to learn which model performs best and for what input, the ensemble also uses the lengths, the source and sentiment scores of the response and context as features.

The ensemble yields F1-scores of 67% and 74% on the Reddit and Twitter test data, respectively. The imbalance between the Reddit and Twitter

scores is consistent in all component models, suggesting that the Reddit data is inherently more challenging to classify. However, more research on why this is the case is needed. Future work may also include experimenting with other component models to improve the overall performance of the ensemble.

Acknowledgments

This research received funding from the Flemish Government (AI Research Program).

References

Ilia Markov et al. 2020. Exploring stylometric and emotion-based features for multilingual cross-domain hate speech detection (in preparation).

Silvio Amir, Byron C. Wallace, Hao Lyu, and Paula Carvalho Mario J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 167–177, Berlin, Germany. Association for Computational Linguistics.

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

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, 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*, pages 48–54, Austin, Texas, USA. Association for Computational Linguistics.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169, San Diego, California. Association for Computational Linguistics.

Debanjan Ghosh, Alexander R. Fabbri, and Smaranda Muresan. 2018. Sarcasm analysis using conversation context. *Computational Linguistics*, 44(4):755–792.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in Twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 581–586, Portland, Oregon, USA. Association for Computational Linguistics.

Hongyan Jing, Daniel Lopresti, and Chilin Shih. 2003. Summarization of noisy documents: A pilot study. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*, pages 25–32.

Aditya Joshi, Pushpak Bhattacharyya, Mark Carman, Jaya Saraswati, and Rajita Shukla. 2016. How do cultural differences impact the quality of sarcasm annotation?: A case study of Indian annotators and American text. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH) 2016*, pages 95–99, Berlin, Germany. Association for Computational Linguistics.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys*, 50(5):73.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Saif Mohammad and Peter Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29:436–465.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(0):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

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 *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. Association for Computational Linguistics.

Vered Shwartz, Gabriel Stanovsky, and Ido Dagan. 2017. Acquiring predicate paraphrases from news tweets. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 155–160, Vancouver, Canada. Association for Computational Linguistics.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. We usually don't like going to the dentist: Using common sense to detect irony on Twitter. *Computational Linguistics*, 44(4):793–832.

Byron C. Wallace, Do Kook Choe, Laura Kertz, and Eugene Charniak. 2014. Humans require context to infer ironic intent (so computers probably do, too). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 512–516, Baltimore, Maryland, USA. Association for Computational Linguistics.

Zelin Wang, Zhijian Wu, Ruimin Wang, and Yafeng Ren. 2015. Twitter sarcasm detection exploiting a context-based model. In *Proceedings, Part I, of the 16th International Conference on Web Information Systems Engineering — WISE 2015 - Volume 9418*, page 77–91, Berlin, Heidelberg. Springer-Verlag.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3485–3495, Osaka, Japan. The COLING 2016 Organizing Committee.