Profiling Ironic Authors on Twitter
Language Processing 2 Project

Zuzanna Dubanowska
University of Copenhagen
vpz558@alumni.ku.dk

Abstract

Irony is a figurative speech device frequently employed by humans in informal settings. Due to its potential to improve many Natural Language Processing (NLP) tasks, there is a growing interest in finding computational approaches to automatic irony detection. While studies show promising results, the performance of traditional NLP systems is worse when figurative language is present in corpora. With the rise of big data generated by social media, Twitter became a data source for detecting irony. In this paper, we propose a system to profile ironic authors on Twitter. We explore different types of features to represent authors' posts and construct a traditional machine learning model for binary irony classification. A dataset consisting of collections of tweets coming from ironic and non-ironic authors is used for evaluating the performance of the ensemble. Results show, that the proposed system achieves 97% accuracy when evaluated on test data.

1 Introduction

Figurative language is an important component of human communications. Irony, characterised by the contrasting of the true meaning and the literal meaning of a message, is recognised as one of the most prevalent devices in non-literal and creative language.

Despite that figurative language is a well-understood linguistic phenomenon, it continues to be a perplexing task in Natural Language Processing (NLP). This is chiefly because in figurative speech words do not appear in their default meaning and decoding the sense of a message often requires contextual knowledge that machines do not possess. Figurative language classification [REY, 2012], and detection of its particular expressions like sarcasm [see e.g. Bamman and Smith, 2015], metaphor [e.g. Tsvetkov et al., 2013, Dodge et al., 2015] and irony [e.g. Van Hee et al., 2018, Frenda et al., 2018, Bueno et al., 2019] using computational approaches have shown promising results.

Nonetheless, studies show that systems effective in various NLP tasks experience performance drops when figurative language is blended into the corpora, as traditional NLP systems do not account for non-literal expressions [Weitzel et al., 2016]. Efforts are directed towards bridging the gap, however there is still need for improvement.

Successful irony detection has a large potential to improve many NLP tasks. In human-computer interaction, agents using irony appear more likeable, appealing and witty [Ritschel et al., 2019]. Similarly, systems correctly identifying irony would be able to interact with the user more naturally. Improving irony detection and ability to use irony could thus encourage more interactions with bots. This is especially important for successful implementation of, e.g., chatbot call centers. In the field of security, many false threat alarms are being raised due to automatic prevention systems mistaking irony for a real threat. Systems being able to identify irony could mitigate this issue. In the field of text mining, accurate irony detection, e.g. in news or customer reviews, is also of high importance.

This paper presents a model to automatically profile authors that frequently resort to irony in their Twitter posts. The proposed solution assigns a binary label to indicate whether an author is ironic or not based on features extracted from text. The rest of this paper is organised as follows. The remainder of this section introduces the reader to Background Work done in Automatic Irony Detection (1.1), Irony Detection from Twitter posts (1.2) and Transformers based methods (1.3). In Section 2 we introduce the methodology used in the experiments, and explain the experimental setting in Section 3. In Section 4 we present the results of the experiments. We discuss and conclude the paper in Section 5.

1.1 Automatic Irony Detection

Recent developments in the field of machine learning have led to increased popularity of machine learning based irony detection. In this formulation, the problem is treated as binary classification where a text is labeled as ironic or non-ironic, or as a multi-class classification where particular types of irony are detected [see e.g. Van Hee et al., 2018]. The problem continues to be popular amongst researchers due to its impact on other areas of Natural Language Processing (see Section 1) and abundance of available data.

Some of the earliest approaches focus on exploiting different kinds of features and traditional machine learning classifiers to detect irony. Buschmeier et al. [2014] propose a framework to detect irony in product reviews from an extensive set of features, ranging from Bag-of-Words features, through polarity, sentiment features, and hyperboles to star-ratings. Gianti et al. [2012] present a study exploring use of irony in online political discussion platform in Italian. Authors proposed four sets of features: statistical, lexical, sentiment of post and polarity value. Authors of both papers analyse the performance of a wide range of classifiers for binary irony classification.

1.2 Detecting Irony in Twitter posts

Social media platforms like Twitter or Reddit offer vast amounts of naturally-generated, opinionated text, where users often turn to rhetorical devices such as sarcasm or irony. Because of that, these social media have been gaining recognition as data sources for detecting figurative language expressions. In irony detection from Twitter posts, most research efforts have focused on English tweets [Van Hee et al., 2018], however analysing tweets in other languages: Arabic [Ghanem et al., 2019, Abu Farha et al., 2021], Italian [Frenda et al., 2018] or Spanish [Bueno et al., 2019] has also been gaining traction.

Reyes et al. [2013] provide one of the first insights into irony detection from Twitter data. Authors capture low-level and high-level properties of irony based on conceptual features such as: signatures, unexpectedness, style, and emotional scenarios and classify tweet passages as ironic or not using Naive Bayes and Decision Tree classifier. This initial investigation, achieving positive results, provided a lot of valuable insights and paved the way to automatic irony detection.

At EVALITA, a shared task to detect irony in tweets in Italian [Frenda et al., 2018] was proposed. Solutions used mostly traditional machine learning approaches (Support Vector Machines (SVM), Multinomial Naive Bayes and ensemble methods). Some systems were based on deep learning methods, with sequence learning networks (Long Short Term Memory (LSTM) and Gated Recurrent Units (GRUs)) being the most popular choice. Most widely used features were: n-grams, word embeddings, polarity of text and semantic and syntactic features, and some teams employed stylistic and structural features in their solution. The winning approach [Cimino et al., 2018] used a Bidirectional Long Short Term Memory (LSTM) network and various length n-grams, word2vec embeddings as their features and polarity-related features. Basile and Semeraro [2018] proposed an regularised SVM with hand-engineered features, such as: keyword based features, microblogging features, polarity and semantic features. The proposed approach scored second best in the binary irony classification task. Di Rosa and Durante [2018] proposed a Multinomial Naive Bayes and SVM ensemble. The framework used various length n-grams and special tokens and emoticon features and scored third in the irony classification task.

At SemEval-2018, a task focusing on irony detection in English tweets was organised [Van Hee et al., 2018]. The solutions encompassed mostly traditional machine learning approaches (SVMs, XGBoost classifier, Logistic Regression and SVM ensembles) and deep learning based approaches LSTMs, Convolutional Neural Networks (CNNs) and one solution implementing a Multilayer Perceptron model. The best performing team, Wu et al. [2018], used a Bidirectional LSTM trained on word embeddings with Part-of-Speech tags. The second best system proposed by Baziotis et al. [2018] was an ensemble of two Attentional LSTM models operating on words and characters respectively, with the word-level LSTM exploiting Twitter-pretrained word embeddings as features.

The winning approach at IroSvA [Bueno et al., 2019], a shared task concerning irony detection in Spanish tweets, was a BERT (Bidirectional Encoder Representations from Transformers) model with in-domain embeddings pre-trained on Spanish Twitter posts [Miranda-Belmonte and López-Monroy, 2019]. Iranzo-Sánchez and Ruiz-Dolz [2019] conducted a comparative evaluation of a BERT language model and traditional machine learning approaches. Authors proposed another BERT-based architecture, pre-trained on multilingual data (Spanish and English) and compared it's performance with traditional machine learning techniques recognised in irony detection such as Support Vector Machines (SVM), Naive Bayes and Gradient Tree Boosting (GTB). Evaluating the models showed that an ensemble of SVM and GTB performed best in binary irony classification, overperforming the BERT classifier by 10 pp.

At the IDAT @ FIRE2019 [Ghanem et al., 2019], a shared-task aiming to detect irony in Arabic Twitter posts, the systems used traditional machine learning approaches (SVM, Multimodal Naive Bayes, Logistic Regression, Ensemble) and deep learning methods (CNNs, RNNs, GRUs and BERT). Participants used Bag-of-words, n-grams, weighed TF-IDF features, emotion features and word embeddings to represent the Tweets. The winning solution [Khalifa and Hussein, 2019] used an ensemble model (Gradient Boost, Random Forest and Multilayer Perceptron) with bag-of-words, n-gram and emotion features. Authors submitted two other systems to the competition: one Bi-LSTM trained on the same feature set and a hybrid ensemble combining the systems from the winning and the second run. The classic ensemble outperformed the deep learning approaches by a significant margin. The second highest performing system used a BERT pretrained on a dialectal Twitter dataset. In order to train the model authors employed sentiment analysis, gender detection, age detection, dialect identification, and emotion detection. This system uniquely proposed different dialects as separate domains and leveraged that for pre-training.

1.3 Transformers based methods

Pretrained language models are an integral part of many NLP tasks nowadays. The Bidirectional Encoder Representations from Transformers [Devlin et al., 2018] (BERT) language model and its many variants, e.g.

RoBERTa [Liu et al., 2019] have recently emerged as very powerful tools for many language tasks, contributing to establishing new state-of-the-art results in many NLP tasks.

At WANLP2021, a shared task to detect sarcasm in Arabic tweets [Abu Farha et al., 2021] most teams used language models from BERT family. The approaches were very similar, employing these language models pretrained in Arabic and fine-tuned to the task, some hybrid approaches combining BERT and a deep neural network were also proposed. The best performing system was a multitask hybrid of MARBERT (Multilingual Arabic BERT) [Abdul-Mageed et al., 2021], where authors pretrained the model on in-domain knowledge in Arabic and English. The other top performing systems were also based on a combination of a BERT architecture with a neural network.

2 Proposed Methodology

Despite the success and growing popularity of BERT-based solutions in NLP, traditional machine learning approaches still outperform them on irony classification [Khalifa and Hussein, 2019, Iranzo-Sánchez and Ruiz-Dolz, 2019]. Moreover, the model comes with limitations: the complex architecture and abstracting intricacies from the user does not leave room for flexibility, training is computationally demanding and the model is unable to process text sequences longer than 512 tokens.

These limitations, together with a record of traditional machine learning methods having success in irony detection (see Subsection 1.2) formed the motivation for our methodology.

2.1 Features

2.1.1 N-grams

We employed word and character-level n-gram features to represent the data. N-grams have a long standing reputation for being golden standard features in NLP. Most traditional machine learning approaches proposed in irony detection literature have used n-gram features [Frenda et al., 2018, Van Hee et al., 2018, Bueno et al., 2019, Ghanem et al., 2019].

2.1.2 Statistical Features

Irony can be challenging to detect in text, even for humans. An analysis of the training data revealed, that users take advantage of extensive punctuation (e.g. exclamation marks, ellipsis or question mark), emoticons or capitalisation to accentuate the ironic meaning of the message. We extracted counts of these markers from the tweets and represented each author's usage of them by taking the mean amount used per tweet.

2.2 Modelling

Several classification methods are recognised for high performance in NLP tasks. To model ironic authors, we chose several models that recur in irony detection literature: AdaBoost [e.g. Van Hee et al., 2018], SVM (Linear and RBF Kernel)[e.g. Frenda et al., 2018], KNN [e.g. Van Hee et al., 2018], Decision Tree [e.g. Reyes et al., 2013], Random Forest [e.g. Khalifa and Hussein, 2019], Logistic Regression [e.g. Ghanem et al., 2019]. We applied the models to our respective feature sets and chose the best performing ones based on cross validated performance. The models were composed into a majority voting ensemble system that takes into account three different algorithms, one per each feature type. We got a number of inspirations for this methodology. Khalifa and Hussein [2019]'s ensemble system was the winning solution at IDAT @ FIRE 2019 Ghanem et al. [2019]. Iranzo-Sánchez and Ruiz-Dolz [2019] showed that traditional machine learning ensembles can outperform BERT language models on irony detection. Di Rosa and Durante [2018]'s system, based on a voting ensemble of classifiers, achieved second best performance in binary irony classification at EVALITA 2018 [Frenda et al., 2018].

3 Experimental Setting

3.1 Dataset

The data used in the experiment was provided by the organisers of the PAN at CLEF 2022 'Profiling Irony and Stereotype Spreaders on Twitter' shared task. The dataset is a set of Twitter posts in English, collected from 420 unique authors with 200 tweets per each of them. Each author was labeled as ironic or not. The dataset is balanced, there is an equal amount of ironic and non-ironic authors.

Split	Number of authors	Ironic / Non-ironic
train	336	50.8% / 49.2%
test	85	47.6% / 52.4%

Table 1: Detailed statistics of the dataset.

The goal was to profile ironic authors based on their number of ironic tweets. To achieve that, the authors were split into training and testing parts, Table 1 provides detailed statistics of the dataset. Due to the nature of the task and how the data was labeled, we treated each author's collection of Tweets as a whole rather than performing the analysis on singular tweets.

To preprocess the posts, we replaced emotion icons ('emojis') with word tokens using a Python library emoji and removed newline characters. The tweets were normalized by the dataset providers, by converting user mentions, hashtags and url links into special tokens: #USER#, #HASHTAG# and #URL#.

3.1.1 Statistical Features

To create the statistical features, the tweets were tokenized using NLTK's Tweet Tokenizer [Elhadad, 2010]. Then the average length of author's tweet, the average number of emojis in a post, use of punctuation signs and capitalisation were found by counting all occurrences in author's collection of posts and averaging over the number of posts.

3.1.2 N-grams

To create the character- and word-level n-grams, the entries were concatenated into single strings (one per author) containing all tweets and `sklearn`'s CountVectorizer [Pedregosa et al., 2011] was used to produce the bags-of-ngrams. We performed cross-validation to find which lengths of n-grams yield the best performance, $n \in \{1, 11\}$. The analysis showed that for both character- and word-level features 1- to 3-grams perform the best.

3.2 Training Procedure

The data were split into training and test 80% / 20%. The framework used for the experiment for constructing models, training, hyperparameter tuning and performance evaluation `sklearn`. We constructed the ensemble system using a Linear SVM, Decision Tree and Adaboost as classifiers, chosen as indicated in the methodology. The hyperparameters were tuned using a grid search and were: learning rate 0.1, number of estimators 500 for Adaboost, max depth 10, number of estimators 40 and max features 3 for Random Forest, C 0.1 and gamma 1 for Linear SVM. We performed each experiment using a 5-fold cross validation setting. After training the classifiers in parallel, we used each of them to predict the labels of test data. The final label of each sample was decided based on majority voting, i.e. if more than half of the classifiers chose some label x, then the final classification label for that sample was x. The ensemble predictions were compared to true labels and the performance of the system was evaluated in terms of binary accuracy.

The code for described procedure is included in our GitHub repository.

4 Results

Classifier	Feature Type	Accuracy
AdaBoost	char-level n-grams	0.9647
Linear SVM	word-level n-grams	0.9176
Random Forest	statistical	0.8705
Ensemble	all	**0.9765**

Table 2: Results of the irony detection experiment.

In Table 2 we present the results of our models on the test data. We present the accuracy of each individual classifier in the ensemble with its feature type, and the performance of the system as whole. The ensemble performed better overall than any of its components taken separately. The character-level n-gram AdaBoost achieved an average accuracy of 96.47%, the word-level n-gram Linear SVM achieved an accuracy of 91.76% and the Random Forest performed worst, with an accuracy of 87.05%. The ensemble reached an accuracy of 97.65%.

5 Discussion/Conclusions

As a preliminary remark, we would like to note that it is not possible to directly compare the performance of our study with other irony detection studies. This owes to our problem formulation: we aimed to profile ironic authors, not classify single ironic tweets. Nevertheless, we discuss some of the main findings and their alignment with other studies to the extent possible.

Our results offer compelling evidence that it is possible to model irony detection using traditional machine learning methods with considerable success. Broadly speaking, our results are consistent with previous research [Van Hee et al., 2018, Frenda et al., 2018, Ghanem et al., 2019, Bueno et al., 2019].

Our findings suggest that ensemble models can perform better than individual parts they are composed of. The classifiers have likely mislabeled different samples and implementing the voting ensemble contributed to overall improvement of accuracy of the system. These observations fall in line with Di Rosa and Durante [2018], Khalifa and Hussein [2019], Iranzo-Sánchez and Ruiz-Dolz [2019].

Our system classifies authors as ironic or not with a success rate of 97% on unseen data. To our best knowledge, this outperforms the current state-of-the-art, however a direct comparison is not possible, since previous approaches focused on classifying single tweets. Future work is needed to evaluate our proposed system on a new, possibly larger dataset to ensure robustness of the solution.

We are aware of possible sources of bias that could have positively influenced our results. Organisers of the shared task did not provide information about how data was collected or labelled. There is a possibility that some bias towards a specific type of irony was introduced in the labeling process.

To conclude, we have presented a model for profiling ironic authors based on their twitter posts. We have employed different types of features: character- and word-level n-grams and statistical features and performed binary classification using an ensemble of three traditional machine learning models. Results show that our system achieves 96% accuracy when evaluated against test data. To our best knowledge, this result improves the state of the art.

References

From humor recognition to irony detection: The figurative language of social media. *Data Knowledge Engineering*, 74:1–12, 2012. ISSN 0169-023X. doi: https://doi.org/10.1016/j.datak.2012.02.005. URL `https://www.sciencedirect.com/science/article/pii/S0169023X12000237`. Applications of Natural Language to Information Systems.

David Bamman and Noah Smith. Contextualized sarcasm detection on twitter. In *Proceedings of the Inter-*

national AAAI Conference on Web and Social Media, volume 9, pages 574–577, 2015.

Yulia Tsvetkov, Elena Mukomel, and Anatole Gershman. Cross-lingual metaphor detection using common semantic features. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 45–51, 2013.

Ellen K Dodge, Jisup Hong, and Elise Stickles. Metanet: Deep semantic automatic metaphor analysis. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 40–49, 2015.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-1005. URL `https://aclanthology.org/S18-1005`.

Simona Frenda, Alessandra Cignarella, Valerio Basile, Cristina Bosco, Viviana Patti, and Paolo Rosso. Ironita @ evalita 2018 irony detection in italian tweets task guidelines. 09 2018.

Reynier Ortega Bueno, Francisco Manuel Rangel Pardo, D. I. H. Farías, Paolo Rosso, Manuel Montes y Gómez, and José Eladio Medina-Pagola. Overview of the task on irony detection in spanish variants. In *IberLEF@SEPLN*, 2019.

Leila Weitzel, Ronaldo Prati, and Raul Aguiar. *The Comprehension of Figurative Language: What Is the Influence of Irony and Sarcasm on NLP Techniques?*, volume 639. 03 2016. ISBN 978-3-319-30317-8. doi: 10.1007/978-3-319-30319-2_3.

Hannes Ritschel, Ilhan Aslan, David Sedlbauer, and Elisabeth André. Irony man: augmenting a social robot with the ability to use irony in multimodal communication with humans. 2019.

Konstantin Buschmeier, Philipp Cimiano, and Roman Klinger. An impact analysis of features in a classification approach to irony detection in product reviews. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 42–49, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-2608. URL `https://aclanthology.org/W14-2608`.

Andrea Gianti, Cristina Bosco, Viviana Patti, Andrea Bolioli, and Luigi Di Caro. Annotating irony in a novel italian corpus for sentiment analysis. *Proceedings of the 4th Workshop on Corpora for Research on Emotion Sentiment and Social Signals, Istanbul, Turkey*, pages 1–7, 01 2012.

Bilal Ghanem, Jihen Karoui, Farah Benamara, Véronique Moriceau, and Paolo Rosso. Idat at fire2019: Overview of the track on irony detection in arabic tweets. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19, page 10–13, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450377508. doi: 10.1145/3368567.3368585. URL `https://doi.org/10.1145/3368567.3368585`.

Ibrahim Abu Farha, Wajdi Zaghouani, and Walid Magdy. Overview of the WANLP 2021 shared task on sarcasm and sentiment detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 296–305, Kyiv, Ukraine (Virtual), April 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.wanlp-1.36`.

Antonio Reyes, Paolo Rosso, and Tony Veale. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47, 03 2013. doi: 10.1007/s10579-012-9196-x.

Andrea Cimino, Lorenzo De Mattei, and Felice Dell'Orletta. Multi-task learning in deep neural networks at evalita 2018. *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*, pages 86–95, 2018.

Pierpaolo Basile and Giovanni Semeraro. Uniba-integrating distributional semantics features in a supervised approach for detecting irony in italian tweets. *EVALITA Evaluation of NLP and Speech Tools for Italian*, 12:152, 2018.

Emanuele Di Rosa and Alberto Durante. Irony detection in tweets: X2check at ironita 2018. *EVALITA Evaluation of NLP and Speech Tools for Italian*, 12: 157, 2018.

Chuhan Wu, Fangzhao Wu, Sixing Wu, Junxin Liu, Zhigang Yuan, and Yongfeng Huang. 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, 2018.

Christos Baziotis, Nikos Athanasiou, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. *arXiv preprint arXiv:1804.06658*, 2018.

Hairo Ulises Miranda-Belmonte and Adrián Pastor López-Monroy. Early fusion of traditional and deep features for irony detection in twitter. In *IberLEF@SEPLN*, 2019.

Javier Iranzo-Sánchez and Ramon Ruiz-Dolz. Vrain at irosva 2019: Exploring classical and transfer learning approaches to short message irony detection. In *IberLEF@ SEPLN*, pages 322–328, 2019.

Muhammad Khalifa and Noura Hussein. Ensemble learning for irony detection in arabic tweets. 12 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL https://arxiv.org/abs/1810.04805.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Muhammad Abdul-Mageed, AbdelRahim A. Elmadany, and El Moatez Billah Nagoudi. ARBERT & MARBERT: deep bidirectional transformers for arabic. *CoRR*, abs/2101.01785, 2021. URL https://arxiv.org/abs/2101.01785.

Michael Elhadad. Natural language processing with python steven bird, ewan klein, and edward loper (university of melbourne, university of edinburgh, and bbn technologies). *Computational Linguistics*, 36:767–771, 12 2010. doi: 10.1162/coli_r_00022.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.