The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

Exploring Machine Learning Techniques for Irony Detection

Chia Zheng Lin*1 Michal Ptaszynski*2 Fumito Masui*3

Kitami Institute of Technology, Kitami, Japan

Irony detection is considered a complex task in Natural Language Processing. This paper first introduce and cover the recently state of irony detection. Then we review and summarize previous related research on text-based irony detection. Finally we compare various classifiers including the proposed CNN model on three dataset of tweets, and analysis and discuss the results. We conclude that CNN is effective for irony detection under various situation with our model outperforming all the other classifiers.

1. Introduction

Irony is considered an important component of human communication recognized as one of the most prominent and pervasive device of figurative and creative language widely used dating back to ancient religious texts to modern time [Ghosh et al 2017]. Merriam Webster, a popular online dictionary, defines irony "as the use of words to express something other than and especially opposite off the literal meaning" [https://merriam-webster.com/].

Due to its nature, irony has important implications for Nature Language Processing (NLP) tasks, which aim to understand and produce human language. In fact, automatic irony detection has a large potential for various applications in the domain of text mining [Van Hee et al. 2018]. Rosenthal et al [2014] demonstrated the impact of irony on automatic sentiment classification by attempting to analyze a test set of irony tweets with standard sentiment analysis tools, and showing the inability of those tools to maintain high performance on irony texts.

In the recent years, studies in irony detection and classification have gained popularity and have been widely applied as sentiment analysis tasks. Various types of approaches were developed and improved to tackle the problem of irony detection. Some of the most popular approaches with better performance are rule-based, statistical, or Deep Learning-based approaches.

Among many Social Networking Services, the one that became one of the most popular for people to express their opinions, share their thoughts and report real-time events, etc., has been Twitter [https://twitter.com/] [Bouazizi and Otsuki 2016]. Many companies and organizations have been interested in these data for the purpose of studying the opinion of people. Therefore it has been suggested that data sets of tweets may be able to bring out the best performance of irony detection approaches.

Whilst many studies has been carried out on irony detection, there have been few empirical investigations into the best and optimal approach for the task. The aim of this paper is to evaluate whether it is possible to develop a classification model for irony detection with various Natural Language Processing (NLP) methods, with particular focus on recent developments in NLP and Artificial Intelligence (AI), such as Deep Neural Networks. In this paper we study, review and analyze previous related researches on text-based irony detection, investigate the potential of other methods,to compare them, and identify the applications of irony detection in various platforms.

The rest of this paper is organized in the following way. Firstly, we describe the problem of irony detection and present some of the previous research. Next, we describe the data set used in this research and approaches applied in experiment for comparison. Further, we explain the evaluation settings, followed by the analysis of experiment results and discussion.

Contact: Name, Affiliation, Address, Phone number, Facsimile number, and E-mail address

2. Research Background

2.1 Definition of Irony

The word irony originates from an Ancient Greek word ε ρ ω ν ε α, meaning dissimulation or feigned ignorance. Irony is often described as a rhetorical device, literary technique, or event in which what appears, on the surface, differs radically from what is actually the conveyed.

The relationship between irony and sarcasm have been confused in many studies. Van Hee [2017] concludes a number of differences between verbal irony and sarcasm, such as the level of aggressiveness, the presence of a target, the intention to hurt, and even some vocal clues. In this research, we will be performing experiments in irony detection, therefore, we will not distinguish between sarcasm and verbal irony, and instead we will be implementing the general term 'irony'.

2.2 Previous Research on Irony Detection

Some of the earliest research dealing with irony detection was a spoken dialogue system using feature extraction approach which included irony detection as a subtask [Tepperman et al. 2006]. As for the later research, Davidov et al. [2010] mainly focused on irony detection from tweets and Amazon product reviews, and Gonzalez-Ibanez et al. [2011] proposed a machine learning model composed of various features.

Numerous studies have attempted to describe the recent trend on approaches to irony detection, which can roughly be classified into three parts: rule-based, machine learning (statistical approach) and deep-learning approaches [Ku-

mar et al. 2017; Barbieri 2017]. Rule- based approaches attempt to identify irony through specific evidences which can be captured with specific rules. Barbieri [2017] reported that rule-based approaches which require no training mostly rely on lexical information and do not perform as well as statistical systems. Reyes et al. [2013] designed another system with different feature types exploiting lexical, syntactic and semantic information.

However, most of the work on irony detection apply statistical approaches. Statistical approaches vary in terms of features and learning algorithms, which mostly composed of two phases. Firstly the data is converted into a feature vector which will be calculated with various methods. Then a machine learning algorithm is used to classify them. Some of the most often used algorithms are Support Vector Machines and Naives Bayes. Liebrecht et al. [2013] implemented bi-gram and tri-gram based features and designed an irony detector that marks unseen tweets as being irony or not.

With the work of Amir et al. [2016] which used a standard binary classification with Convolutional Neural Network (CNN) and Poria et al. [2016] who used a combination of CNNs trained on different tasks, Deep Learning approaches have been brought into the scene of irony detection. Popular deep learning algorithms such as CNN [LeCun et al., 1998] and Long Short Term Memory (LSTM)[Hochreiter and Schmidhuber, 1997] have been widely used in recent works. Amir et al. [2016] and Poria et al. [2016] used CNN in irony detection. LSTM is also considered another popular deep learning algorithm in text classification. Ghosh and Veale [2016] proposed a network model composed of CNN and followed by a LSTM network. The model outperformed state-of-the-art text-based methods for irony detection at the publishing time.

Following the Semantic Evaluation 2018 Task 3: Irony Detection in English Tweets [Van Hee et al., 2018] which received submissions from 43 teams for the binary classification Task A, deep learning algorithms were further optimized for irony detection task. The best ranked system by team THU_NGN [Wu et al., 2018] consisted of densely connected LSTM network with multi-task learning strategy. One of the top teams, NTUA-SLP [Baziotis et al., 2018] ensembled two independent models, based on bi-directional LSTM networks. The systems that were submitted represent a variety of neural-network-based approaches and other popular classification algorithms include SVM, Maximum Entropy, Random Forest, and Naive Bayes [Van Hee et al., 2018]. Overall, there seems to be some evidence to indicate that approaches with ensemble learners are the current trend to further challenge the detection of irony however there is still no definitively best method for detecting irony automatically.

3. Proposed Methods

3.1 Data Preprocessing

Light normalization were applied to the data set. All of the tweets were transformed into lowercases and emojis were represented with their labels (e.g. :smileyface:). Furthermore, all URLs and tagged users are replaced with specific tokens "_url_" and "_tagged_" because they are not likely to be contributing to the classification.

3.2 Feature Extraction

Referring to Ptaszynski et al. [2017] work on data preparation, the following feature preprocessing was done after the normalization. Traditional weight calculation scheme, namely term frequency with inverse document frequency (TF-IDF) were applied to both dataset with and without hashtags. Term frequency t f (t,d) refers here to the traditional raw frequency, meaning the number of times a term t (word, token) occurs in a document d. Inverse document frequency idf (t,D) is the logarithm of the total number of documents containing the term nt. Finally tf*idf refers to the term frequency multiplied by inverse document frequency.

3.3 Classifiers

Several types of classifiers are applied for comparison in this research.

Naive Bayes classifier is a supervised learning algorithms applying Bayes' theorem which assign class labels to problem instance represented as vectors of feature values, often applied as a baseline in text classification task.

Next the k-Nearest Neighbors (kNN) classifier takes an input k-closest training samples and classifies them based on the majority vote . It is often used as a baseline after Naive Bayes. For the input sample to be assigned to the class of the first nearest neighbor, k=1 setting is used here.

JRip also known as Repeated Incremental Pruning to Produce Error Reduction (RIPPER) which is efficient in classifying noisy text [Sasaki and Kita, 1998], learns rules incrementally in order to optimize them. Also J48 which is implemented with C4.5 decision tree algorithm, builds decision trees from dataset and the optimal splitting criterion are further chosen from tree nodes to make the decision.

Support Vector Machines (SVM) is a supervised machine learning algorithm designed for classification or regression problems which uses a technique called kernel trick to transform data and finds an optimal boundary between the possible output. Two types of SVM functions are used here, linear and radial.

Lastly, Convolutional Neural Networks (CNN) which are a type of feed-forward neural network, were applied with Rectified Linear Units (ReLU) as neuron activation function. The proposed CNN method consisted of two hidden convolutional layers, containing 20 and 100 feature maps with both layers having 5x5 size of patch and 2x2 max-pooling, and Stochastic Gradient Descent [LeCun et al., 2012].

4. Experiment

4.1 Dataset

The dataset used in this research is the dataset provided by Semantic Evaluation 2018 Task 3: Irony Detection in English Tweets [Van Hee et al., 2018] which was constructed by searching Twitter for the hashtags #irony, #sarcasm

and #not, which could occur anywhere in the tweet that was finally included in the corpus. All tweets were collected between 2014/12/01 and 2015/01/04 and represent 2,676 unique users, and were manually labelled using a fine-grained annotation scheme for irony [Van Hee et al., 2016a]. The entire corpus was cleaned by removing retweets, duplicates and non-English tweets and replacing XML-escaped characters (e.g. &).

The dataset consists of 4,618 tweets (2,222 ironic + 2,396 non-ironic) that were manually labelled by three students using the brat rapid annotation tool with an inter annotator agreement study set up to assess the reliability of the annotations. Additionally, there are two duplicate sets of the data with all the ironic hashtags removed and with only hashtags.

4.2 Evaluation setup

Three separate datasets provided from the original preprocessed dataset are being performed in the experiment, with and without hashtags. Each of the classifiers mentioned in 3.4 was tested on both version of the dataset in a 10-fold cross validation procedure. The results were calculated using standard Accuracy (A), Precision (P), Recall (R) and balanced F-score (F1). The results were determined based on the highest achieved balance F-score.

4.3 Results discussion

Table 1 shows the summarization of all results. We can see that the results the from dataset with hashtags included are significantly higher than the other dataset without hashtags. As stated by Maynard and Greenwood [2014], even without considering ironic hashtags, the presence of hashtags greatly increase the results of irony detection.

The kNN scored the lowest result among the classifiers for both dataset and Naive Bayes barely came after it. Even though these classifiers may be able to do well in typical sentiment analysis, stemming and parsing are not applied to the dataset, hence the noisy language might be a challenge for them.

For the decision tree-based classifiers, J48 did better than Random Forest with hashtag included but scored as low as kNN when hashtags are removed. Random Forest scored third highest for both dataset but it is unfortunately impractical because it is time-inefficient when comparing to SVM. The rule learner algorithm, JRip scored highest when hashtags are included but just performed better than kNN and J48 when hashtags are removed.

The most used algorithms in irony detection are SVMs. As we can observe, the radial-SVM is comparable to the proposed CNN. They achieved the same F score on dataset with hashtags and SVM ranked second just after CNN for the dataset without hashtags. The linear-SVM, however, did not perform well enough in both condition.

When it comes to the proposed CNN with two hidden layers, 5x5 patch size, max-pooling, and Stochastic Gradient, it outperformed all of the classifiers in the harsh situation where all hashtags were removed (F-score= 0.66). While CNN is time-efficient comparing to other classifiers in small datasets, larger dataset might produce different result. One of the best irony detection system so far is also a network model composed of CNN, but applied to a data set of 39K tweets [Ghosh and Veale, 2016].

Table 1: Experiment result F-score

Classifiers	with hashtag	no hashtag	only hashtag
kNN	0.753	0.571	0.881
Naive Bayes	0.808	0.621	0.758
Random Forest	0.883	0.641	0.898
J48	0.883	0.641	0.884
JRip	0.899	0.616	0.897
SVM-linear	0.826	0.615	0.893
SVM-radial	0.844	0.644	0.833
CNN	0.844	0.660	N/A

The last column of Table 1 shows the results of the dataset which consists of only the hashtags. Besides CNN which the results could not be calculated due to the lack of suitable environment, all the remaining classifiers attain high F-score comparable to the dataset with hashtag. Together these results provide important insights into the presence of hashtags in a tweets especially ironic hashtag for irony detection.

These findings enhance our understanding of the impact of hashtag, which makes great difference in irony detection. In general, irony detection is still an unsolved problem, but it will be an easy task on Twitter thanks to the presence of deliberated hashtag. Taken together, these results also suggest that hashtag is the product of authors who realize that their ironic phrases alone may not be enough for their audience to understand. This redefines irony in textual communication especially on social network services from figurative speech to direct speech.

5. Conclusion

In this paper we reviewed and summarized previous related works on text-based irony detection. We covered various types of systems designed in the past works such as rule-based, statistical based, and deep learning based approaches. Then we compared a few different classifiers including the proposed optimized CNN model on two datasets with and without hashtags.

With minimal preprocessing done, the proposed CNN model outperformed all the other classifiers under the same condition even though the results are still far away from the known state-of-the-art system (F-score=0.92). We found that CNN is effective for irony detection under various situation.

In the future, we plan to evaluate the proposed method with other classifiers on larger corpus with more preprocessing. Future work will also focus on optimizing the feature extraction of the dataset.

References

[Ghosh 17] Aniruddha Ghosh and Tony Veale, Fracking Sarcasm using Neural Network, Proceedings of

NAACL-HLT 2016, Association for Computational Linguistics (2017)

[Van Hee 18] Cynthia Van Hee, Els Lefever and Veronique Hoste, SemEval-2018 Task 3: Irony Detection in English Tweets, Proceedings of the 12th International Workshop on Semantic Evaluation(SemEval-2018), Association for Computational Linguistics (2018)

[Poria 16] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Prateek Vij, A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks, COLING 2016 (2016)

[Kumar 17] Lakshya Kumar, Arpan Somani and Pushpak Bhattacharyya, Approaches for Computational Sarcasm Detection: A Survey, ACM CSUR (2017)

[Ptaszynski 17] Michal Ptaszynski, Juuso Kalevi Kristian Eronen, and Fumito Masui, Learning Deep on Cyberbullying is Always Better Than Brute Force, LaCATODA2017, (2017)

[Van Hee 17] Cynthia Van Hee, Can machines sense irony? Exploring automatic irony detection on social media, University Gent (2017)

[Barbieri 17] Francesco Barbieri, Machine Learning Methods for Understanding Social Media Communication: Modeling Irony and Emojis, Departament DTIC (2017)

[Wu 18] 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, Proceedings of the 12th International Workshop on Semantic Evaluation(SemEval-2018), Association for Computational Linguistics (2018)

[Vu 18] Thanh Vy, Dat Quoc Nguyen, Xuan-son Vu, Dai Quoc Nguyen, Michael Catt and Michael Trenell, NIHRIO at SemEval-2018 Task 3: A Simple and Accurate Neural Network Model for Irony Detection in Twitter, Proceedings of the 12th International Workshop on Semantic Evaluation(SemEval-2018), Association for Computational Linguistics (2018)

[Tepperman 06] Joseph Tepperman, David Traum, and Shrikanth Narayanan, "YEAH RIGHT": Sarcasm Recognition for Spoken Dialogue Systems, Interspeech 2006, ICSLP (2006)

[Bouazizi 16] Mondher Bouazizi and Tomoaki Otsuki, A Pattern-Based Approach for Sarcasm Detection on Twitter, Digital Object Identifier, IEEE Access (2016)

[Davidov 10] Dmitry Davidov, Oren Tsur and Ari Rappoport, Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon, Proceedings of the Fourteenth Conference on Computational Natural Language Learning, Association of Computational Linguistics (2010)

[Gonzalez-Ibanez 11] Roberto Gonzalez-Ibanez, Smaranda Muresan, and Nina Wacholder, Identifying Sarcasm in Twitter: A Closer Look, Proceedings of the 49th Annual Meeting of the Association For Computational Linguistics, Association for Computational Linguistic (2011)

[Reyes 13] Antonio Reyes, Paolo Rosso, and Tony Veale, A multidimensional approach for detecting irony in Twitter,Lang Resources & Evaluation (2013)

[Liebrecht 13] Christine Liebrecht, Florian Kunneman, and Antal Van den Bosch, The perfect solution for detecting sarcasm in tweets #not, Proceedings of the 4th Wiorkshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (2013)

[Amir 16] ilvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mario J. Silva, Modelling Context with User Embeddings for Sarcasm Detection in Social Media, Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL), Association for Computational Linguistic (2016)

[LeCun 98] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, Gradient-Based Learning Applied To Document Recognition, Proc of the IEEE (1998)

[Hochreiter 97] Sepp Hochreiter and Jurgen Schmidhuber, Long Short-Term Memory, Neural Computation 9(8) (1997)

[Baziotis 18] Christos Baziotis, Nikos Athanasiou, Pinelopu Papalampidi, Athanasia Kolovou, Georgios Paraskevopoulos, Mikolaos Ellinas, Alexandros Potamianos, NTUA-SLP at SemEval-2018 Task 3: Tracking Ironic Tweets using Ensembles of Word and Character Level Attentive RNNs, Proceedings of the 12th International Workshop on Semantic Evaluation(SemEval-2018), Association for Computational Linguistics (2018)

[Sasaki 98] Minoru Sasaki and Kenji Kita, Rule-Based Text Categorization Using Hierarchical Categories, Systems, Man, and Cybernetics, 1998 (1998)

[Maynard 14] Diana Maynard and Mark A. Greenwood, Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis, LREC 2014 Proceedings (2014)