Lancaster at SemEval-2018 Task 3: Investigating Ironic Features in English Tweets

Edward Dearden and Alistair Baron
School of Computing and Communications
Lancaster University
Lancaster, UK, LA1 4WA
initial.surname@lancaster.ac.uk

Abstract

This paper describes the system we submitted to SemEval-2018 Task 3. The aim of the system is to distinguish between irony and non-irony in English tweets. We create a targeted feature set and analyse how different features are useful in the task of irony detection, achieving an F_1-score of 0.5914. The analysis of individual features provides insight that may be useful in future attempts at detecting irony in tweets.

1 Introduction

With so many people using social media and microblogs such as Twitter, a huge amount of natural language data is available to be analysed. It is desirable to be able to accurately interpret what people are saying. An example of this is in the field of sentiment analysis where the aim is to determine whether the language being used by an author is positive or negative.

These kinds of tasks are made more difficult by the presence of figurative language. Figurative language is a type of language where the contents of the text are not literally true, making it difficult to ascertain the true meaning of the text purely from the content. Irony is a particular type of figurative language in which the meaning is often the opposite of what is literally said and is not always evident without context or existing knowledge. A system capable of accurately detecting irony would be a valuable addition to sentiment analysis systems, and other systems for natural language understanding which are confounded by irony.

In online discourse, examples of irony are very common. Social media platforms capture natural language which often includes sarcastic sentences. An example of a use for irony detection is in the area of online product reviews (Tsur et al., 2010) which can contain large amounts of ironic language. An irony detection system could be used to prevent ironic negative reviews being misinterpreted as positive and highlighted in advertising.

The system described in this paper aims to identify targeted features of irony and analyse how important they are in identifying ironic tweets. The annotated twitter data we use is that provided by the event organisers. The task is described in the task description paper (Van Hee et al., 2018).

2 Related Work

The task of irony detection is an inherently difficult one. Wallace (2015) suggests that to create a good system for irony detection, one cannot rely on lexical features such as Bag of Words, and one must consider also semantic features of the text.

There have been various methods employed to detect irony. Reyes et al. (2012) created a dataset generated by searching for user-created tags and attempted to identify humour and irony. The features used to detect irony were polarity, unexpectedness, and emotional scenarios. Their classifier achieved an F_1-score of 0.54 for general tweets of various topics, rising to 0.65 when the irony features were combined with the ambiguity features used to detect humour. The score also improved when looking at domain specific tweets, suggesting domain knowledge and context can be useful for identifying irony.

More recently, Van Hee et al. (2016a) investigated annotated ironic tweet corpora and suggested that looking at contrasting evaluations within tweets could be useful for detecting irony. Van Hee et al. (2016b) also created a system to detect ironic tweets, looking beyond text-based features, using a feature set made up of lexical, syntactic, sentiment, and semantic features. They achieved an F_1-score of 0.68. They also suggested

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 587–593
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

that irony by polarity clash was more simple to detect than other forms of irony, e.g. situational irony.

A number of works have looked at detecting sarcastic tweets. Sarcasm is a type of irony in which the meaning is the opposite of what is literally said. Maynard and Greenwood (2014) used a rule based system to supplement sentiment analysis systems by flipping the predicted polarity of a tweet if it contained a mismatch of sentiment between hashtag and text. They achieved an F_1-Score of 0.91. Davidov et al. (2010) used pattern-based and punctuation-based features, achieving an F_1-Score of 0.83. González-Ibáñez et al. (2011) combined lexical and pragmatic features, aiming to distinguish between sarcastic, positive, and negative utterances. Their classifiers achieved 70% accuracy distinguishing between sarcasm and positive tweets and between sarcasm and negative tweets. Barbieri et al. (2014) looked at distinguishing sarcasm from other domains of text including irony. They achieved an F_1-score of 0.60 when differentiating between sarcasm and irony compared to 0.89 when differentiating between sarcasm and politics. These findings suggest that it is easier to distinguish sarcastic tweets over irony in general. This could link to Van Hee et al. (2016b)'s finding that situational irony and other types of irony are harder to detect than irony by polarity clash. Given much of the irony in tweets is sarcasm, looking at some of these features may be useful.

One challenge for irony detection is that the understanding of irony often relies on context. There is certain contextual information that is required for a human to parse a sentence as being ironic. For example, the sentence "I am soooo happy to be going to the dentist tomorrow" would only be noticed as being ironic if the reader understood that the word 'dentist' has certain negative associations. Rajadesingan et al. (2015) tries to address this challenge by looking at more behavioural aspects of irony including looking at positive and negative associations of certain words, achieving an accuracy of 83%.

Other approaches have aimed to capture the context in which a tweet was posted for irony detection. Bamman and Smith (2015) used author and audience features on top of tweet features. These features looked at the past tweets of authors and the people to whom the tweet is responding. Their best performance was 85% accuracy. Wang et al. (2015) used three types of history of the tweet to try and bring in additional context: history of the conversation; history of the author; and history of the hashtag/topic. They improved the baseline F_1-Score of 0.55 to 0.60.

3 System Description

The developed system uses only the text data of the tweets provided with the task. It does not handle any contextual features as these would require gathering previous tweets and responses. The classifier we used was a standard, untuned SVM. As much of the code as possible has been made publicly available so it can be replicated[1]. There are certain parts of the system that cannot be shared, such as the USAS tagging system[2].

In the preprocessing stage the tweets were reduced to just the text, separately extracting emojis, hashtags, and user mentions, for use as features. Most of the processing of the text was performed with the python Natural Language Toolkit (NLTK) (Bird and Loper, 2004), including using the NLTK Tweet Tokeniser to tokenise the text.

For classification we used the Linear SVC implementation in the popular python machine learning package scikit-learn (Pedregosa et al., 2011). We performed no tuning of the model as we were more interested in features and their usefulness than gaining the maximum precision and we wanted to avoid overfitting. We also used a random forest classifier to compare results.

3.1 Features

The features used by this system fall into four main categories: Tweet-level features, Bag-of-X features, Sentiment features, and Complexity features. All feature values were normalised so they were between 0 and 1.

Tweet Features are the non-language features contained directly within the contents of the tweet. These features include punctuation, hashtags and emoji. These have been used in past research and found to be useful for the task. It is thought that these features are used to flag a tweet as ironic. As there were many different types of emoji used on Twitter, some very infrequently, the emojis, punctuation, and hashtags used were restricted to the

[1] `https://github.com/dearden/SemEval2018-Irony`

[2] `http://ucrel.lancs.ac.uk/usas/`

top 50 by frequency over the whole training set of each. Each of these top 50 were a feature and their value was 0 or 1 based on whether or not the token was present in the text. The tweets were tokenised such that repeated punctuation marks were counted as a single token. For example, "..." would be counted as a single token, not 3 instances of ".". We also count examples of repeated characters, e.g. in "Greeeeat!", and the proportion of the tweet that is capitalised. Other tweet features were: Number of links, number of mentions, number of hashtags, Tweet length, average word length, and amount of punctuation.

Bag-of-X Features is the set of features which contain the 1000 most frequent tokens of various types. The tokens used for these features were: word unigrams, word bigrams, character trigrams, POS tags, and Semantic tags. For the POS tagging, we used the NLTK POS tagger (Bird and Loper, 2004) and for Semantic tagging, the USAS semantic tagger (Rayson et al., 2004). Semantic tags put each word (or multi-word expression) into semantic categories, providing knowledge if some texts contain more emotion-based terms or more science and technology terms, for example. This should provide a higher level view of the text than achieved by bag of words. Using these techniques is like casting a wide net over the text that may find characteristics of irony not picked up by the more targeted features. They are also included to test the theory that lexical features are not useful for the task of irony detection.

For **Sentiment Features** we used a popular python package VaderSentiment (Pedregosa et al., 2011). Sentiment features may be important because if irony involves saying something positive to mean something negative, it may be that contrasts in sentiment or extreme values of sentiment are features of irony. The sentiment features gathered were: Positive sentiment score, negative sentiment score, mean score, standard deviation, range, average change of sentiment between adjacent words, number of positive to negative transitions, and emoji sentiment. When looking at changing sentiments, we modelled each sentence as a collection of words that were either positive, negative or neutral. We looked at the way the words in the sentence transitioned between positive and negative. For example, the sentence "I love how awful everything is right now", would have one transition between "love" and "awful". Emoji Sentiments were used from the work of Kralj Novak et al. (2015) on the sentiment of emojis. We included the mean, maximum, and minimum emoji sentiment, as well as the number of positive and negative emojis.

The **Complexity features** we gathered were: negations, function words, number of syllables, Automated Readability Index, ambiguity, lexical diversity, and lexical density. These features were included to examine the difference in complexity and style between ironic and non-ironic tweets. Ambiguity was calculated as the average number of meanings for each word in the sentence according to WordNet (Kilgarriff, 2000).

3.2 Feature Sets

We tested the system with seven feature sets. The first four feature sets (Tweet, Bag-of-X, Sentiment, and Complexity) are as described above. The other three are as follows:

Submission: A combination of Tweet, Complexity, and Sentiment Features as described above containing 126 features.

Reduced: A reduced version of the submission set with the token frequency features (Emoji, Punctuation) removed. The idea of this set is to see if keeping only a focussed set of features impacts performance. This set contains 27 features.

Combined: All the features combined together to see whether performance is increased by using all the features. Contains 3,537 features.

The features representing occurrences of individual hashtags were omitted as they did not repeat very often and we did not want the model to overfit. Also, as the data was gathered using hashtags we were concerned that certain hashtags would be used in conjunction with the hashtags used to gather the data and may not be representative of irony generally. We left the Bag-of-X features out of the submission feature set because we did not want this set being too large and overfitting. With our submission set, we aimed to use targeted features as opposed to data-driven features such as bag-of-words and character n-grams. This makes the system more explainable, with the reasoning pre-defined. With data-driven features, especially character n-grams, it can be difficult to explain

Feature set	Precision	Recall	F_1 Score
Baseline	0.5296	0.5756	0.5516
Tweet	0.5209	**0.7203**	**0.6046**
Bag-of-X	0.5153	0.5949	0.5522
Sentiment	0.4691	0.5370	0.5007
Complexity	0.4350	0.6238	0.5125
Submission	0.5321	0.6656	0.5914
Reduced	0.5385	0.6527	0.5901
Combined	**0.5387**	0.6495	0.5889

Table 1: Results of Linear SVC on test set.

why features are useful for the task. For the reduced set we went further, omitting all of the data-driven features. This meant removing Emoji and punctuation features so the feature set was more likely to capture the true features of irony as well as being more explainable.

One drawback of our feature extraction is the noisiness of the text. The NLTK POS tagger and VADER Sentiment only perform well with standard text. If we had performed better normalisation on the text before extracting Bag-of-X and Sentiment features, these feature sets may have performed better.

4 Results and Discussion

We ran the classifier with a number of different feature sets to assess the power of different types of feature. Bag-of-Words using all word tokens was used as a baseline to compare the other feature sets to. The bag-of-words implementation we used was slightly different to the baseline provided with the task in so far as it does not use Scikit-learn's in-built Vectoriser. As well as the baseline bag of words, seven feature sets were evaluated, as described in Section 3.2.

The results from our system can be seen in Table 1, with a Linear SVC. The tweet feature set was more useful than expected, with the highest F_1 Score. This suggests that twitter users communicate elements of what they mean via emojis, punctuation, and the way they present their tweets. The high recall means that this feature set is causing many tweets to be flagged as ironic. However, given in most real data there are likely to be fewer ironic tweets than non-ironic, high precision may be more desirable than high recall. Another concern with relying on these features is that an irony detection implementation that is clueless if a tweet contains no emojis or hashtags is not an effective system. This may also explain why the Tweet features did not achieve higher results. Many ironic tweets do not contain emojis or punctuation that flag their irony. The bag-of-x features and the baseline performed similarly, both getting an F_1 Score of around 0.55. The addition of Bigrams, Trigrams, POS and Semantic tags does not seem to have increased the accuracy. The performance of these feature sets is not surprising as the features were in no way targeted to irony. This supports the findings of Wallace (2015) that lexical features alone are not effective at identifying irony. The sentiment features performed the worst. This is in line with the results of Van Hee et al. (2016b), which also showed sentiment to be the weakest individual group. The reduced set performed almost as well as the submission set which is promising given it contained 99 fewer features, showing that specific targeted features are of most use.

With a random forest classifier, the results also found the Tweet feature set achieved the highest F_1 score of 0.5903 compared to the Baseline feature set score of 0.4957. The next highest was the Reduced feature set which achieved a score of 0.5657, outperforming both the Submission and Combined sets. This might be because it contains less sparse features which tree classifiers prefer, or could suggest that the emoji and punctuation features are causing the classifier to overfit. Both the Tweet and Reduced feature set achieved much higher precision than with the Linear SVC, 0.5922 and 0.5936 respectively, but lower recall, 0.5884 and 0.5401.

Next we looked at features individually, rather than looking at groups. To do this we used the coefficients for each feature in the Linear SVC model. These are the weights used by the model to decide how much each feature should weigh in on the final decision.

First we look at the features in the reduced feature set. The results are shown in Table 2. This set is interesting because it only contains the more information-dense features. Number of links, the number of mentions, and the number of hashtags are all ranked highly for identifying non-ironic tweets. This may be because tweets that have high values for these features are more focussed on sharing links, for example images, with their friends. As the focus is more on the thing they are sharing rather than in the text, these tweets may be less likely to be ironic. These features are un-

Feature	Weight	Class
Number of Links	-2.52	Non-ironic
Number of Syllables	1.89	Ironic
Number of Mentions	-1.84	Non-ironic
Punctuation Count	-1.50	Non-ironic
Repeated Characters	1.42	Ironic
Function Words	-1.20	Non-ironic
Lexical Density	-1.18	Non-ironic
Mean Sentiment	1.09	Ironic
Capitalisation	-1.04	Non-ironic
Number of Hashtags	-1.04	Non-ironic

Table 2: Top 10 LinearSVC weightings in Reduced Feature Set.

likely to be useful for identifying non-irony outside of the twitter domain. Duplicate characters are highly weighted for identifying irony. This feature is often used to signpost irony by overemphasising the emotion they are expressing ironically such as in the case of, "Loooovvveeeeeee when my phone gets wiped". Another common use is repeated punctuation to make the point clear, for example, "Gotta love being lied to....".

Looking into the rankings of individual words also provides some insights. Three of the top ranking words indicating irony for the Linear SVC with the combined feature set were "love", "great", and "fun". It is interesting that these are all positive emotional words. This could suggest that it is more common for such tweets to use positive language with a true negative meaning. A lot of tweets follow the format of "I love when...", going on to describe a negative experience. These tweets are often given a positive sentiment as the negative part of the tweet doesn't always use negative language. This highlights the effect a lack of irony detection can have on sentiment analysis. The system could be more effective if it took into account negative concepts, for example "going to the dentist", that are not explicitly negative.

Our findings suggest that social elements and structure of tweets are important for distinguishing ironic tweets from non-ironic tweets. The words that rank highly support the claim of Van Hee et al. (2016b) that irony by contrast is the easiest to detect. Irony by contrast is the most highly represented type of irony in the corpus so in most cases of irony, such features will be useful for detection. A future system would look at semantic and contextual features in more depth.

4.1 Subtask B

Subtask B involved a more complex classification task in which the system had to distinguish between different types of irony. In this task, the potential labels given by the classifier were: 0 – Non-irony, 1 – Verbal irony realised through a polarity contrast, 2 – Descriptions of situational irony, and 3 – Verbal irony without a polarity contrast. These categories are explained in detail in the task description (Van Hee et al., 2018).

We used the same feature set used for task A, aiming to test whether the same features could classify between the different types of irony. The submitted result achieved an F_1-score of 0.3130. This result was gained using a Linear SVC classifier with the submission feature set and compared to the bag-of-words baseline F_1-score of 0.3198. The random forest classifier achieved an F_1-score of 0.3465. These results suggest more complex, tailored features would be needed for this task.

Both classifiers labelled the majority of tweets as 0 – non-ironic. The next most frequent label was 1 – irony by polarity contrast. This is likely to be because the features were aimed at detecting irony from non-irony and did not take into account situational information. The results seem to support the idea that irony detection via polarity contrast is the easiest to detect. This is further supported by the fact that 76% of the examples of other irony and situational irony were classified as non-ironic.

To investigate this further, we looked at the results from subtask A and looked at how many ironic by polarity contrast tweets were correctly labelled as ironic compared to the other two forms of irony. This was to see if, when the classifier was dealing with a binary ironic/non-ironic decision, it still had the same problem of not detecting examples of irony with no polarity contrast. 82% of the ironic by polarity contrast tweets were correctly labelled as ironic by the classifier, compared to 50% of the other types. This suggests that the classifier was using the features of polarity contrast to make its decisions. This makes sense as some of the features, especially those from the sentiment set, were targeted at detecting this type of irony with features such as number of positive to negative sentiment transitions.

As a final experiment, we trained the classifier with the ironic by polarity contrast tweets removed. 164 tweets from the test set and 1,390 from the training set were removed, leaving a test set of 473 non-ironic and 147 ironic tweets and a training set of 1,923 non-ironic and 521 ironic tweets. The aim was seeing if the system could distinguish between non-irony and irony with the easiest to detect tweets removed. In this setup, the classifier only correctly labelled 4 out of the 147 ironic tweets as ironic. This suggests that it is much harder to distinguish between non-ironic text and these two forms of irony. More complex features that directly look at context may be needed for this task.

5 Conclusion

In this paper, we have described a system for the detection of irony using targetted features. The resulting F_1-score of 0.59 was an improvement over the baseline bag-of-words. The analysis of our findings provided insight into the features that are particularly useful for detecting irony. Tweet features performed well suggesting that Twitter users potentially broadcast their meaning using features such as emojis and structure. We also investigated how our system performed when distinguishing between different types of irony. Our findings suggest that deeper, more complex features will be needed to accurately identify situational irony and irony with no polarity contrast. We could improve our system by looking into contextual and semantic features. For example, we looked into sentiment of words, but not at words with positive and negative associations in certain contexts. Our analysis provides insights that may be useful for future research into irony detection.

References

David Bamman and Noah A Smith. 2015. Contextualized sarcasm detection on twitter. In *ICWSM*, pages 574–577.

Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014. Modelling sarcasm in twitter, a novel approach. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–58.

Steven Bird and Edward Loper. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 107–116, Stroudsburg, PA, USA. Association for Computational Linguistics.

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: Short Papers - Volume 2*, HLT '11, pages 581–586, Stroudsburg, PA, USA. Association for Computational Linguistics.

Adam Kilgarriff. 2000. Wordnet: An electronic lexical database.

Petra Kralj Novak, Jasmina Smailovi, Borut Sluban, and Igor Mozeti. 2015. Sentiment of emojis. *PLOS ONE*, 10(12):1–22.

Diana Maynard and Mark Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 97–106, New York, NY, USA. ACM.

Paul Rayson, Dawn Archer, Scott Piao, and Anthony M McEnery. 2004. The ucrel semantic analysis system.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data Knowl. Eng.*, 74:1–12.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2016a. Exploring the realization of irony in twitter data. In *LREC*.

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, 26th International Conference on Computational Linguistics*, pages 2730–2739. ACL.

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. *Artif. Intell. Rev.*, 43(4):467–483.

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*, pages 77–91, New York, NY, USA. Springer-Verlag New York, Inc.