WLV at SemEval-2018 Task 3: Dissecting Tweets in Search of Irony

Omid Rohanian, Shiva Taslimipoor, Richard Evans and Ruslan Mitkov
Research Group in Computationa Linguistics
University of Wolverhampton
Wolverhampton, UK
{omid.rohanian,shiva.taslimi,r.j.evans,r.mitkov}@wlv.ac.uk

Abstract

This paper describes the systems submitted to SemEval 2018 Task 3 "Irony detection in English tweets" for both subtasks A and B. The first system leveraging a combination of sentiment, distributional semantic, and text surface features is ranked third among 44 teams according to the official leaderboard of the subtask A. The second system with slightly different representation of the features ranked ninth in subtask B. We present a method that entails decomposing tweets into separate parts. Searching for contrast within the constituents of a tweet is an integral part of our system. We embrace an extensive definition of contrast which leads to a vast coverage in detecting ironic content.

1 Introduction

In figurative language (also known as trope), there is a departure from literal use of words. In order to decode meaning, therefore, it is not enough to rely solely on the literal sense of individual words. Irony and sarcasm are two types of such language that exploit this technique in similar ways. They "both involve deliberately saying something that is incongruous or the opposite of what the speaker knows to be true" (Hanks, 2013). This is sometimes formulated as a transgression of the Gricean maxim of quality (Grice, 1975)[1].

Under this assumption it follows that the violation is only permissible thanks to shared knowledge between the speaker and the hearer. In order to achieve this goal, the speaker frames the message with some form of commentary or metamessage that signals the ironic or sarcastic nature of the message. This is usually realised through negation of the original meaning (Haiman, 1998).

Regardless of their similarities, irony and sarcasm are not technically the same as they might be employed for different purposes. It is widely accepted that sarcasm involves some degree of verbal aggression and ridicule directed at the hearer, whilst irony can simply be used for humorous or emphatic effect. It has been shown that computational processing of irony and sarcasm requires some knowledge of the context in which they appear, sometimes including paralinguistic information (Wallace et al., 2014).

Exploring ironicity has practical implications, since performance of sentiment analysis systems is directly affected by knowledge about irony and sarcasm (Pozzi et al., 2016).

As part of the 12th workshop on semantic evaluation (SemEval-2018), Shared Task 3 defines two subtasks with regards to irony detection in English tweets (Van Hee et al., 2018). Subtask A involves binary classification. The objective is to train a system that can label tweets as ironic or not. Subtask B is a multi-class classification problem with the objective to label tweets with one of the four specified labels describing the type of irony (verbal irony by means of a polarity contrast, situational irony, other verbal irony, and non-ironic).

To tackle these problems, in this paper we describe two rich feature-based systems addressing each subtask. Our systems use a combination of sentiment, distributional semantic, and text surface features. The code and data for this project is freely available[2].

The rest of this paper is organised as follows: Section 2 describes related work. Section 3 provides a comprehensive description of the overall methodology including pre-processing, feature representation, and system architecture. Sections 4 and 5 discuss experiments and results, Section 6 involves error analysis and finally Section 7 concludes the paper with some closing remarks.

[1] "Do not say what you believe to be false."

[2] https://github.com/omidrohanian/irony_detection

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 553–559
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

2 Related Work

There has been a recent surge of interest in the tasks of irony and sarcasm detection due in large part to increasing popularity of social media and the availability of data from websites like Twitter and Reddit. Some recent work focus exclusively on irony or sarcasm in isolation (Joshi et al., 2016), under the assumption that sarcasm has a stronger impact on changing the sentiment of the overall message. However in many cases, these terms are taken to be practically synonymous (Pozzi et al., 2016; Wallace et al., 2014; Ptáček et al., 2014). SemEval has a long-standing shared task on sentiment analysis that has also involved processing of figurative language including irony and sarcasm (Ghosh et al., 2015; Nakov et al., 2016). Results from recent tasks on sentiment analysis confirm that the top performing teams increasingly employ deep learning methodologies, while classical machine learning models like SVM and logistic regression remain popular (Ghosh et al., 2015; Rosenthal et al., 2017).

3 Methodology

We train our supervised systems using an ensemble soft voting classifier with logistic regression (LR) and support vector machine (SVM) as component models, and create our feature sets using a combination of sentiment, semantic, and surface features. We leverage these handcrafted features in combination with dense vector representations which differ in details between subtask A and B. The differences in feature engineering and representation between the two subtasks will be discussed in 3.2.

3.1 Pre-processing

Tweets were tokenised using NLTK's tweet tokeniser (Loper and Bird, 2002). Additional pre-processing was done to obtain a subset of the features that concerned surface orthographic features (e.g. all capitals, elongations, emoticons, etc) and pattern-based named entities (e.g. time, place, user, etc). For this we used the ekphrasis toolkit (Baziotis et al., 2017). It employs an XML-based annotation scheme that made it easy to extract this information.

For sentiment features and embeddings, however, pre-processing beyond tokenisation was deemed unnecessary as our emoji and word vectors were pre-trained on raw tweets.

3.2 Feature Representation

In our observation of the training data, we noticed that tweets often follow a fairly consistent spatial pattern. Informative words are more likely to cluster at both ends of a tweet. Hashtags, while scattered throughout the whole text, tend to occur at the end. In ironic tweets, negative sentiments are more likely to be preceded by neutral or positive ones. An example is given in (1).

(1) What a golden morning. 😒

In order for our models to capture these spatial patterns and to provide a more rigorous representation of a tweet's structure, we propose the idea of decomposing a tweet into separate chunks and extracting features for each one separately. By concatenating these features we are able to partially preserve information about linear precedence. To this end, we simply split the sentences to two sections as represented in example (2) and (3).

(2) 8ams are just | so LOVELY .
surface features: $time1$ | $allcaps2$

(3) SEEING @AlpEmiel ON | SATURDAY whaddddddup #legend
surface features: $allcaps1$, $user1$ | $elongated2$, $hashtag2$

In examples (2) and (3), the numbers '1' and '2' signify the first and second sections of the tweet respectively.

We use the same split structure for representation of other features, and pre-trained dense vectors.

Contrast is one of the most important properties of ironic language. One contribution of this work lies in the particular manner in which the notion of contrast is defined. Contrast is a marker of polarity shift and is usually seen as the presence of a positive sentiment referring to a negative situation, or vice versa (Riloff et al., 2013) which is sometimes referred to as "asymmetry of affect" (Clark and Gerrig, 1984).

Twitter language is non-standard and informal. Polarity shift can be realised through contrast between different elements of the tweet. The elements of a tweet are: text, hashtagged tokens, and emojis. We adopt a more inclusive stance with regards to the concept of contrast with the following scenarios:

1. Contrast between different parts of the same

element of a tweet

a. antithetical emojis
b. antithetical hashtagged tokens

2. Contrast between two different elements of a tweet

c. text and hashtagged tokens
d. text and emojis
e. hashtagged tokens and emojis

A sizable proportion of the tweets contain multiword hashtags, such as *#NotExcitedAboutThisAtAll* or *#goodluck*, that require segmentation. For this we used ekphrasis' hashtag segmentation tool (Baziotis et al., 2017).

We separate the tweet and its segmented hashtagged tokens and run each group through the sentiment analysis tool from Stanford CoreNLP (Manning et al., 2014). CoreNLP assigns to an input any of 5 sentiment classes from very negative to very positive (0 to 4). If the resulting hashtag and text scores are on opposite sides of this spectrum, we consider this as contrast type c. as defined in 2.

For d. and e. we follow a similar procedure. To approximate the sentiments present in emoji tokens, we use Emoji Sentiment Ranking (Kralj Novak et al., 2015). This is a lexicon of 751 emojis whose sentiments are ranked based on human annotation of 70,000 tweets in 13 European languages.

The resulting contrast feature is a binary value that is set to True if any one of the aforementioned forms of contrast is present in the tweet.

Relying on sentiment information from CoreNLP, we define an additional binary feature named Intensity. It checks whether the sentiment in a segment of the tweet is sharply positive/negative. This translates to a value of 0 or 4 in the sentiment scores for that particular segment. The rationale behind definition of this feature is that too much of a positive emotion can, in certain contexts, imply a negative sentiment. To a lesser extent, the opposite is also true of an excessively negative emotion.

To track the changes of sentiment expressed throughout the whole tweet, we define sentiment patterns of Rise (R), Fall (F), and Stable (S) on a word-by-word basis and encode this information in a vector representing the number of S, R, F, RF, and FR patterns. For these features, we rely on information from Vader sentiment lexicon (Gilbert, 2014).

For dense vectors we use word2vec embeddings pretrained on a large twitter corpus as described in Godin et al. (2015). One limitation of these embeddings is that they don't contain information on emojis. Therefore we have to complement this resource with additional embeddings specifically trained on emojis (Eisner et al., 2016).

3.3 Task-specific Selected Features

3.3.1 Subtask A

For subtask A, we found that the best way to combine embeddings is through averaging, separately for left and right parts[3]. Features we combine with these vectors are the following: Surface features, Intensity (for left and right), and Contrast.

3.3.2 Subtask B

For subtask B, concatenation of the embeddings was deemed more effective. Furthermore, we augment the combined embeddings with bigram tf-idf count vectors.

As a rhetorical trope, irony can often have subtle political and social dimensions, and is used frequently to express opinionated thoughts in general (Hutcheon, 1994). We noticed that adding topic modeling features to our system in subtask B slightly improves classification performance as these features can help the model capture more subtle forms of irony that tend to co-occur with certain topics and are not necessarily realised as polarity contrasts. Topic modeling of the tweets is done using Latent Dirichlet Allocation (LDA) and Non-negative matrix factorization (NMF).

Other features we add to the above are: Surface features (we consider these with regard to both the whole tweet and its left and right splits), Intensity (for left and right), Contrast, and Vader-based Rise and Fall sentiment patterns.

4 Experimental Settings

We use the data (text including emojis) as provided by the organisers of the shared task. We train our models on the training set using 10-fold cross-validation. Predictions were made on the held-out test data.

[3] word and tweet embeddings are averaged independently, and subsequently the averages are concatenated.

	ironic	non-ironic	total
train	1911 (49.84%)	1923 (50.15%)	3834
test	311 (39.66%)	473 (60.33%)	784

Table 1: Statistics of the data for subtask A

```
rightIntensity, contrast, date1,
sad1, surprise1, url1, date2,
elongated2, laugh2, sad2, shocking2,
url2, user2
```

Figure 1: The most informative features for subtask A

Train and test data in both subtasks A and B are the same and only differ in their annotation. Tables 1 and 2 present the breakdown of the classes and the number of their instances in each subtask.

The most informative features are selected using recursive feature elimination (RFE) (Guyon et al., 2002). As a result, the algorithm uses 13 features for subtask A as listed in figure 1. They are concatenated with the vectors that were derived by separately averaging the words and emoji vectors of the left and right parts of tweets.

The best features derived from RFE for subtask B did not improve the performance of the model. Therefore we use all of the 87 features which are consequently augmented with the concatenation of the word and and emoji vectors of tweets.

The baseline system provided by task organisers is an SVM classifier which uses tf-idf feature vectors. We consider this as the benchmark and report the results for 2 different settings of our system as follows:

- setting 1: average of word and emoji vectors of bi-sectioned tweets
- setting 2: concatenation of word and emoji vectors

In both settings we combine vectors with best features and feed them to the classifiers. To achieve the best system for subtask A (best system A) we apply a voting classifier with soft voting between LR and SVM whose model components are based on setting 1 plus the 13 best features that were selected using RFE for subtask A.

The best system for subtask B is a voting classifier between 3 LRs with 3 different class weights as shown in Table 3. The components of the models are based on setting 2 plus all features for subtask B.

5 Results and Discussion

Table 4 details the results for subtask A, and the results for subtask B are presented in Table 5. After cross-validation on the TRAIN set, the best system which is an ensemble voting classifier trained on models based on setting 1 + best features of subtask A achieves the highest record in F1-score and recall, but is outperformed in accuracy by its own component model. In terms of precision it also scores lower than the system based on setting 2 + all features.

When tested on the TEST data, our best system for subtask A ranked third overall on the shared tasks's official leaderboard among 44 teams with an F-score of 0.65. It has the second highest score for recall. This indicates that the coverage of the model is extensive.

For subtask B, our best system is an ensemble voting classifier comprised of three logistic regression models based on setting 2 + all features with the set-up indicated in Table 3.

As can be seen in Table 5, it gives the best F1-score, accuracy, and recall when cross-validated on the TRAIN data. On the held-out TEST data, the system ranked 9th in terms of F1-score, and with 0.6709 accuracy ranked second out of all participating systems in the shared task.

Table 6 shows the F1-scores for subtask B based on the system's performance on each individual label. In the case of irony by clash, our system achieves an F1-score of 0.6584. This confirms that our features are informative enough to help the model capture this type of irony fairly well, even though only 20.91% of the tweets belong to this class (Table 2).

However, in the case of situational irony the system performs much worse. There are several possible factors that collectively contribute to this poorer performance. Situational irony is less studied in the literature and designing effective features to model it is more difficult. By definition, it involves a situation that does not conform to the expectations of the speaker and elicits an emotional response (Shelley, 2001). Expectations differ among individuals and people often react differently to the same events and stimuli which further complicates the problem.

In the provided dataset, the number of instances of this type of irony is small (only 8.24% of the total in the TRAIN set), and there are no salient tex-

	non-ironic	clash	situational	other	total
train	1923 (50.15%)	1390 (36.25%)	316 (8.24%)	205 (5.34%)	3834
test	473 (60.33%)	164 (20.91%)	85 (10.84%)	62 (7.90%)	784

Table 2: Statistics of the data for subtask B

	non-ironic	clash	situational	other
LR1	1	1	1	1
LR2	1	1	2	2
LR3	1	1	3	3

Table 3: Weights each LR classifier assigns to the 4 classes in subtask B

tual characteristics that can signal their occurrence while distinguishing them from irony by clash.

6 Error Analysis

Vast coverage in subtask A also means that the model is quick to judge a tweet as ironic which translates to a large number of tweets getting tagged as 1. According to Table 1 the distribution of labels is slightly skewed towards non-ironic labels, but in our predictions 0.62% of the tweets are tagged as ironic (Table 4) which explains higher recall and lower precision. This can be traced back to the inclusive definition of contrast as defined in 3.2.

The gold standard provided is not without faults. As an example (4) is obviously an ironic tweet that is incorrectly labeled as 0 in the gold-standard[4]. Also in example (5) the word *tit* (altered in spelling for censorship), is being used in two ways; first in its literal sense, and the other to sarcastically refer to a politician as foolish. This was labeled as non-ironic in the dataset, which is subject to debate. Our system correctly identified both of these instances.

(4) Corny jokes are my absolute favorite

(5) #farage a t1t in public who doesnt agree with seeing t1ts in public #breastfeeding

Looking at the per-class performances in subtask B (Table 6), the best system is predicting non-ironic instances with a high F1-score of 0.7652. However the F1-scores for other classes remain low.

The numbers for situational is lower than irony by clash, which seems logical because in order to effectively pinpoint a tweet as ironic by situation it is sometimes necessary to have access to information beyond the text which could involve a broader context (social, cultural, political, etc) as exemplified in the following examples that are taken from the `TRAIN` set:

(6) Sure Staff... Now Hiring. http://t.co/HDgfxG7elF

(7) #mondaymorning pouring rain and i am singing 'the most wonderful time of the year' as i walk to the office

(8) Patrick Kielty hosting Radio 2's Comedy Awards...

In (6), textual information does not provide anything of significant value. If the user clicks on the link, it seems like the image is about an employment agency that is hiring. Normally, they supply staff to clients who are recruiting but in this case, it is the agency itself which is recruiting. This goes against expectation. Realisation of this instance as situational irony requires interpretation of the image which in turn requires linking the name *Sure Staff* to an agency, and the background knowledge about the role of employment agencies.

Example (7) involves the interpretation of a rainy day on Monday morning as unpleasant, which is subjective. Example (8) implies that the comedian is not particularly known to be funny, which again requires background knowledge and is also dependent on the opinion of the annotator, as it could also read as a non-ironic sentence if the reader does not share the same impression of the comedian.

7 Conclusion

In this paper, we have described our supervised systems to identify ironic tweets and categorise them into three types. Our systems leveraging a combination of word/emoji vectors and features related to polarity contrast, intensity and text surface features achieved competitive results for bi-

[4] *corny* has a negative connotation, implying that the joke is unfunny, and uninteresting

		Accuracy	Precision	Recall	F1-score
TRAIN	benchmark system	0.6375	0.6440	0.6096	0.6263
	LR with setting 1	0.6643	0.6543	0.6923	0.6728
	LR with setting 2	0.6502	0.6466	0.6578	0.6521
	LR with setting 1 + best features of subtask A	**0.6808**	0.6616	0.7357	0.6967
	LR with setting 2 + all features	0.6787	**0.6726**	0.6923	0.6823
	best system A	0.6742	0.6452	**0.7698**	**0.7020**
TEST	best system A	0.6429 (15)	0.5317 (20)	0.8360 (2)	0.6500 (3)

Table 4: Results for subtask A

		Accuracy	Precision	Recall	F1-score
TRAIN	benchmark system	0.6064	0.4359	0.3540	0.3470
	LR with setting 1	0.6142	0.4952	0.3449	0.3278
	LR with setting 2	0.6239	**0.5394**	0.3796	0.3817
	LR with setting 1 + all features	0.6325	0.4867	0.3696	0.3550
	LR with setting 2 + all features	0.6450	0.5308	0.4061	0.4134
	best system B	**0.6458**	0.5280	**0.4122**	**0.4215**
TEST	best system B	0.6709 (2)	0.4311 (11)	0.4149 (10)	0.4153 (9)

Table 5: Results for subtask B

	non-ironic	clash	situational	other
TRAIN	0.7064	0.6584	0.2768	0.0444
TEST	0.7652	0.4651	0.2595	0.0299

Table 6: Per-class F1-scores for the best system in subtask B

nary classification of tweets as ironic/non-ironic. The system is ranked third out of 44 participating systems due to its high coverage in identifying ironic-tweets. For the subtask of multi-class classification, we have also used topic modeling features and features related to the distribution of polarity. The system is ranked ninth out of 32 participating systems with a very competitive accuracy.

In future, we intend to extract more sophisticated features related to situational irony. Observation of the dataset confirms that in cases where the tweet involves a URL, the contents of the external web page can play an important role in discriminating between ironic and non-ironic tweets. Therefore introduction of multimodal features is one future direction to enhance performance of such models.

References

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Herbert H Clark and Richard J Gerrig. 1984. On the pretense theory of irony. *Journal of Experimental Psychology: General.*

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*, page 4854. Association for Computational Linguistics.

Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes. 2015. Semeval-2015 task 11: Sentiment analysis of figurative language in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 470–478.

CJ Hutto Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International Conference on Weblogs and Social Media (ICWSM-14).*

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153.

H. P. Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Se-*

mantics: Vol. 3: Speech Acts, pages 41–58. Academic Press, New York.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422.

John Haiman. 1998. *Talk is cheap: Sarcasm, alienation, and the evolution of language*. Oxford University Press on Demand.

Patrick Hanks. 2013. *Lexical analysis: Norms and exploitations*. MIT Press.

Linda Hutcheon. 1994. *Irony's edge: The theory and politics of irony*. Psychology Press.

Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, Mark Carman, Meghna Singh, Jaya Saraswati, and Rajita Shukla. 2016. How challenging is sarcasm versus irony classification?: A study with a dataset from English literature. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 123–127.

Petra Kralj Novak, Jasmina Smailović, Borut Sluban, and Igor Mozetič. 2015. Emoji sentiment ranking 1.0. Slovenian language resource repository CLARIN.SI.

Edward Loper and Steven Bird. 2002. Natural language processing toolkit. http://www.nltk.org/.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1–18.

Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, and Bing Liu. 2016. *Sentiment analysis in social networks*. Morgan Kaufmann.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on Czech and English twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223.

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.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Cameron Shelley. 2001. The bicoherence theory of situational irony. *Cognitive Science*, 25(5):775–818.

Cynthia Van Hee, Els Lefever, and Vronique 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.

Byron C Wallace, Laura Kertz, Eugene Charniak, et al. 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)*, volume 2, pages 512–516.