Monday mornings are my fave :) #not
Exploring the Automatic Recognition of Irony in English tweets

Cynthia Van Hee, Els Lefever and Véronique Hoste
LT^3, Language and Translation Technology Team
Department of Translation, Interpreting and Communication – Ghent University
Groot-Brittanniëlaan 45, 9000 Ghent, Belgium
Firstname.Lastname@UGent.be

Abstract

Recognising and understanding irony is crucial for the improvement natural language processing tasks including sentiment analysis. In this study, we describe the construction of an English Twitter corpus and its annotation for irony based on a newly developed fine-grained annotation scheme. We also explore the feasibility of automatic irony recognition by exploiting a varied set of features including lexical, syntactic, sentiment and semantic (Word2Vec) information. Experiments on a held-out test set show that our irony classifier benefits from this combined information, yielding an F_1-score of 67.66%. When explicit hashtag information like *#irony* is included in the data, the system even obtains an F_1-score of 92.77%. A qualitative analysis of the output reveals that recognising irony that results from a polarity clash appears to be (much) more feasible than recognising other forms of ironic utterances (e.g., descriptions of situational irony).

1 Introduction

With the emergence of the social web, a large part of our daily communication has moved online. As a result, the past decade has seen an increased research interest in text mining on social media data. The frequent use of irony in this genre (Ghosh and Veale, 2016; Maynard and Greenwood, 2014) has important implications for tasks such as sentiment analysis and opinion mining (Liu, 2015), which aim to extract positive and negative opinions automatically from online text. Irony detection is therefore crucial if we want to push the state of the art in sentiment analysis or, more broadly, any task involving text interpretation (e.g., cyberbullying detection).

Most computational approaches to date model irony by relying solely on categorical labels like irony hashtags (e.g., *#irony, #sarcasm*) assigned by the author of the text. To our knowledge, no guidelines presently exist for the more fine-grained annotation of irony in social media content without exploiting this hashtag information. In order to understand how irony is linguistically realised and how it can be recognised in text, we developed a set of annotation guidelines for identifying specific aspects and forms of irony that are susceptible to computational analysis. We collected a Twitter corpus containing 3,000 English tweets with an irony hashtag and, based on these guidelines, manually annotated them for the presence of irony. We explored the feasibility of automatic irony detection by relying on a varied set of features, including lexical, shallow syntactic, sentiment and lexical semantic information.

The remainder of this paper is structured as follows. Section 2 includes an overview of existing work on defining and modelling irony. Section 3 zooms in on the corpus construction and annotation. Section 4 outlines the experiments, and Section 5 presents the results. Section 6 concludes the paper with some prospects for future research.

2 Related Research

Since many years, irony has been a frequent topic of discussion in linguistics and philosophy. More recently, researchers in the field of natural language processing (NLP) have shown an increasing interest in the subject, trying to formalise the concept of irony, and detect it automatically. Different types of irony can be distinguished. Kreuz and Roberts (1993) define four types of irony: (i) Socratic irony and (ii) dramatic irony, both explained as a tension between what the hearer knows and what the speaker pretends

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2730–2739, Osaka, Japan, December 11-17 2016.

to know (with the latter entailing a performance aspect), (iii) Irony of fate, which involves an incongruence between two situations, similarly to what is commonly understood as situational irony (Lucariello, 1994), and (iv) verbal irony, which implies a speaker who intentionally says the opposite of what they believe.

In this research, we focus on verbal irony and (written forms of) situational irony. A popular definition of verbal irony is saying the opposite of what is meant (Grice, 1975). Though often criticised (e.g., Giora (1995); Sperber and Wilson (1981)), this definition is commonly used in contemporary research on the automatic modelling of irony (Kunneman et al., 2015). When describing how irony works, many studies struggle to distinguish between verbal irony and sarcasm. Some consistently use one of the two terms (e.g., Grice (1975); Sperber and Wilson (1981)), or consider both as essentially the same phenomenon (e.g., Attardo (2000); Reyes et al. (2013)). Other studies claim that sarcasm and verbal irony do differ in some respects, stating that ridicule (Lee and Katz, 1998), hostility and denigration (Clift, 1999), and the presence of a victim (Kreuz and Glucksberg, 1989) play a more important role in sarcasm than in irony. To date, however, experts do not formally agree on the distinction between irony and sarcasm. For the present research, we elaborated a working definition (Section 3.1) that does not distinguish between the two either, and refer to this linguistic form as (verbal) irony.

Automatic irony detection has received increased research interest in the past few years, with one of the main motivations being the improvement of sentiment analysis systems. In effect, as irony implicitly alters the polarity of an utterance, its recognition is important for the development and refinement of sentiment classification systems. Computational approaches to irony detection involve supervised or semi-supervised learning. More recently, researchers have been investigating deep learning by defining neural networks for irony detection. Twitter is a popular data genre when training supervised models for irony detection, one of the main advantages being that it minimises (or even discards) the annotation effort as it contains self-describing hashtags like *#sarcasm* and *#irony*. Davidov et al. (2010) experimented with 6 million tweets and 66,000 Amazon product reviews. They built an algorithm (*SASI*) based on semi-supervised learning and exploited syntactic patterns and punctuation as features, yielding F-scores of 0.79 (Amazon) and 0.83 (Twitter). Reyes et al. (2013) built a corpus of 40,000 tweets and divided it into four topics based on hashtag information (*irony, education, humour* and *politics*). They made use of Decision trees and Naïve Bayes exploiting a rich set of features capturing *style* (e.g., n-grams), *emotional scenarios* (e.g., imagery), *signatures* (e.g., pointedness), and *unexpectedness* (e.g., temporal imbalance). Their approach yields F= 0.72 on recognising the irony topic. Barbieri and Saggion (2014) experimented with the same corpus as Reyes et al. (2013) and used Random Forest and Decision Tree as classifiers. They made use of a varied set of features (word frequency, writing style, punctuation, ambiguity, etc.), and performed binary classification experiments to distinguish irony from each one of the three other topics: education (F= 0.73), humour (F= 0.75), and politics (F= 0.75). Riloff et al. (2013) presented a bootstrapping algorithm to automatically learn sequences of positive sentiment and negative situation phrases from ironic tweets. When adding n-grams and sentiment lexicon features, their SVM classifier achieved an F-score of 0.51. Kunneman et al. (2015) collected a dataset of 800,000 tweets. They made use of Balanced Winnow, exploiting word uni-, bi- and trigrams as features, and obtained an accuracy of 87% on the ironic tweets. In line with Wallace's (2015) claim that text-based features are too shallow and that context and semantics are required for reliable irony detection, a recent study from Ghosh and Veale (2016) describes neural-network-based semantic modelling for irony detection. The researchers compared the performance of an SVM model exploiting shallow features to that of neural networks capturing semantic information and demonstrated that the latter outperformed the SVM model (F= 0.92 vs. 0.73).

It is important to note that, in the above-described papers, training data is often obtained by collecting tweets with hashtags like *#sarcasm* and *#irony* and labelling them accordingly (i.e., tweets containing such hashtags are labeled as ironic, whereas tweets devoid of such hashtags are considered non-ironic). An important contribution of this paper is that, after collecting data based on irony hashtags, all tweets were manually labeled for irony based on a fine-grained annotation scheme (Van Hee et al., 2016b). Furthermore, to estimate the impact of irony hashtags on a detection system, we evaluate the performance

of our classifier before and after removing them from the corpus.

3 Corpus Description

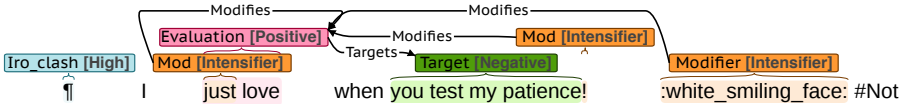
We collected a dataset of 3,000 English tweets by searching for the hashtags *#irony*, *#sarcasm* and *#not*. Unlike many other approaches, all tweets were manually annotated for irony presence based on a newly developed annotation scheme (Van Hee et al., 2016b), which resulted in 2,396 ironic and 604 non-ironic tweets. Some example annotations are presented in Section 3.1.

3.1 Corpus Annotation

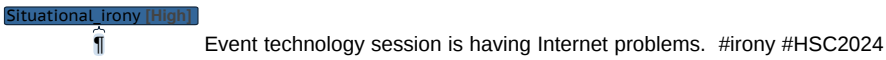
The main goal of our annotation guidelines was to develop a set of reproducible coding principles to mark irony in (social media) text. As we ultimately want to be able to model irony in text, without relying on additional (i.e., hashtag) information provided by the writer of the message, our main focus was on disentangling expressions of verbal irony. In accordance with the classic account of irony, i.e., 'saying the opposite of what is meant' (Grice, 1975), we define verbal irony as an *evaluative expression whose polarity (i.e., positive, negative) is inverted between the literal and the intended evaluation, resulting in an incongruence between the literal evaluation and its context*. While a detailed overview of the annotation procedure is provided in the guidelines (Van Hee et al., 2016b), we briefly discuss the main principles below. Brat was used as annotation environment (Stenetorp et al., 2012).

At the **tweet level**, annotators indicated whether the tweet was (i) ironic by means of a clash (example 1), (ii) contained another type of irony (e.g., situational irony, example 2), or was (iii) not ironic (example 3).

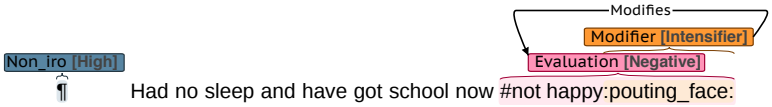
(1)



(2)



(3)



In order to better understand how this irony is realised, the tweets were also annotated **below tweet level**. In case of irony expressed by means of a clash, it was also indicated whether an irony-related hashtag (e.g., *#sarcasm*, *#irony*, *#not*) was required for recognising the irony. Furthermore, annotators were asked to signal variants of verbal irony that are particularly harsh (i.e., carrying a mocking or ridiculing tone with the intention to hurt someone), since it has been shown that harshness may be a useful feature to distinguish between irony and sarcasm (Barbieri and Saggion, 2014). Sentence 4 shows an example of such a harsh tweet.

(4) Shout outs to the guy who took a shower in 1 Million before heading out. The WHOLE BUS thanks you #Sarcasm

The annotators also marked all evaluations contained by the tweet and indicated text spans that contrast with the polarity expressed by that evaluation (i.e., *targets*) (See sentence 5 for an example). For each evaluation:

- the polarity was indicated;
- modifiers were annotated (if present);
- (in the case of ironic tweets) the target of the evaluation was indicated. Also, the implicit polarity of the target was defined based on context, world knowledge or common sense.

(5) The most hideous spider, that makes me feel sooo much better. #not
→ Evaluations: *most hideous* (neg. polarity), *makes me feel sooo much better* (pos. polarity).
→ Modifiers: *most*, and *sooo much*.
→ Targets: *the most hideous spider* (as target of *makes me feel sooo much better*), which holds a negative implicit polarity.

3.2 Annotation Statistics

To assess the validity of the annotations, inter-annotator agreement was measured between three independent annotators. Kappa scores for (i) the annotation of irony (ironic vs. not ironic) and (ii) the decision whether an irony hashtag was required to recognise the irony are 0.72 and 0.67, respectively.

Table 1 presents the distribution of the corpus as divided by the different annotation labels[1]. As can be inferred from the table, most instances that were labeled as ironic belong to the category *ironic by means of a clash*. When we zoom in on the category *other type of irony*, we can distinguish two subcategories: *situational irony* and *other verbal irony*. Whereas the former encompasses (written instances of) ironic situations and comprises the majority of this annotation class, the latter contains instances of irony that describe neither situational irony, nor a clash between two polarities (viz. the literal and the intended one). Nevertheless, they are still considered to be ironic. We refer to Van Hee et al. (2016a) for more details on the corpus statistics.

Ironic by means of a clash	**Other type of irony**		**Not ironic**	**Total**
	Situational irony	*Other verbal irony*		
1,728	401	267	604	3,000

Table 1: Statistics of the annotated corpus: number of instances per annotation category.

For our binary classification experiments, we merged the categories *ironic by means of a clash* and *other type of irony*. As we wanted a balanced dataset (ironic vs. not ironic), we added 1,792 non-ironic tweets to the current corpus. These tweets are from the same Twitter users from which we collected the initial 3,000 tweets (Table 1) and contain no irony-related hashtags. As such, our experimental corpus consists of 4,792 tweets, of which 2,396 are ironic and another 2,396 are not ironic.

4 Experiments

In our classification experiments, we evaluated the viability of automatically recognising verbal irony in tweets. To this end, we constructed a pipeline and ran a series of experiments while exploiting different feature groups. These include standard text classification features (i.e., bags-of-words, lexical and syntactic features), and features based on existing sentiment lexicons. Furthermore, we added semantic features based on Word2Vec clusters, which is, to our knowledge, novel in SVM-based approaches to irony detection.

For the classification experiments, we split the randomised corpus (4,792 instances) into an 80% training and 20% test set for evaluation, resulting in a training set of 3,834 instances and a held-out test set of 958 instances. Both sets show a balanced irony distribution (50% ironic vs. 50% not ironic). Furthermore, all annotation categories (Table 1), as well as the extra non-ironic instances that were added (Section 3.2) are equally distributed among the train and test sets.

4.1 Preprocessing

After constructing the corpus, all emoji were replaced by their name or a description using the Python Emoji module[2] to facilitate annotation and processing of the data. Furthermore, we normalised hyperlinks and @-replies or mentions to *http://someurl* and *@someuser*, respectively.

Other preprocessing steps involve tokenisation and PoS-tagging (Gimpel et al., 2011), lemmatisation (Van de Kauter et al., 2013) and named entity recognition (Ritter et al., 2011).

[1] Due to a refinement of the annotations, the corpus statistics are slightly different from a first version of the annotated corpus described in Van Hee et al. (2016a).

[2] https://github.com/carpedm20/emoji/.

4.2 Information Sources

For the automatic irony detection system, we implemented a variety of features that represent every instance within a (sparse) feature vector.

- As **lexical** features, we included bags-of-words (BoW) features that represent a tweet as a 'bag' of its words or characters. We incorporated token unigrams and bigrams and character trigrams and fourgrams. Furthermore, a set of numeric and binary features were included containing information about (i) character and (ii) punctuation flooding, (iii) punctuation and (iv) capitalisation, (v) hashtag frequency and (vi) the hashtag-to-word ratio, (vii) emoticon frequency, and (viii) tweet length. Where relevant, numeric features were normalised by dividing them by the tweet length in tokens.

- As **syntactic** features, we integrated four Part-of-Speech features for each of the 25 tags in the tagset. These indicate for each PoS-tag (i) whether it occurs in the tweet or not, (ii) whether the tag occurs 0, 1, or > 2 times, (iii) the frequency of the tag in absolute numbers and (iv) as a percentage. Also the number of interjections was added as a feature. Furthermore, we included a binary feature indicating a 'clash' between verb tenses in the tweet (see Reyes et al. (2013)). Finally, we integrated four features indicating the presence of named entities in a tweet: one binary feature and three numeric features, indicating (i) the number of named entities in the text, (ii) the number and (iii) frequency of tokens that are part of a named entity.

- Six **sentiment lexicon** features were implemented based on existing sentiment lexicons: AFINN (Nielsen, 2011), General Inquirer (GI) (Stone et al., 1966), MPQA (Wilson et al., 2005), the NRC Emotion Lexicon (Mohammad and Turney, 2013), Liu's opinion lexicon (Hu and Liu, 2004), and Bounce (Kökciyan et al., 2013). For each lexicon, five numeric and one binary feature were derived:

  - the number of positive, negative and neutral lexicon words averaged over text length;
  - the overall tweet polarity (i.e., the sum of the values of the identified sentiment words);
  - the difference between the highest positive and lowest negative sentiment values;
  - a binary feature indicating whether there is a polarity contrast (i.e., at least one positive and one negative sentiment word from the lexicon are present in the tweet).

  The sentiment lexicon features were extracted in two ways: (i) by considering all tokens in the instance and (ii) by considering only hashtag tokens (e.g., *lovely* from *#lovely*). We took negation cues into account by flipping the polarity of a sentiment word when it occurred in a negation relation.

- As **semantic** information, we used word embedding cluster features generated with Word2Vec (Mikolov et al., 2013). The word embeddings were generated from a separate background corpus of 45,251 English tweets, collected with the hashtags *#sarcasm*, *#irony* and *#not*. More precisely, we ran Word2Vec on this corpus, applying the CBoW model, a context size of 8, a word vector dimensionality of 200 features, and a cluster size of $k = 2{,}000$[3]. The following are two example clusters: [#chistecorto #dailysarcasm #fun #sarcastically #sarcastichumor] and [#exams #nosleep #10am editing essay grading psychology stress revision]. The clusters were implemented as binary features, indicating for each cluster whether a word contained by that cluster occurs in the tweet.

In total, we have four feature groups. Based on each of them, we trained a binary classifier which was then tested on the held-out set. After evaluating the performance of each feature group individually, another experiment was run with the combined feature groups (comprising 100,278 individual features).

[3]To define k, we performed 5-fold cross validation experiments on the training data, exploiting features based on different cluster sizes (100; 200; 500; 1,000 and 2,000).

4.3 Experimental Setup

As mentioned earlier, we conducted binary classification experiments for detecting ironic tweets by exploiting lexical, syntactic, sentiment lexicon and semantic features. One of the contributions of this paper is to measure the impact of irony-related hashtags (e.g., *#irony*) in the dataset. To this end, we ran two sets of experiments based on each feature group: one before and another after removing such hashtags from the corpus.

We used LIBSVM (Chang and Lin, 2011) with the standard RBF kernel as the classification algorithm. As shown by Keerthi and Lin (2003), a nonlinear kernel like RBF (or *Gaussian*) is at least as good as a linear one if it is properly tuned. The LIBSVM parameters C and γ were therefore optimised for each experiment exploiting a different feature group, and by means of a cross-validated grid search on the complete training data. During the parameterisation, γ is varied between 2^{-15} and 2^{3} (stepping by factor 4), while the cost parameter C is varied between 2^{-5} and 2^{15} (stepping by factor 4). In both setups, the optimised values for C and γ were 2^{3} and 2^{-11}, respectively. These optimal parameter settings were then used to build a model for each feature group using all the training data, which was evaluated on the held-out test set (Section 5).

5 Results

This section presents the experimental results. As mentioned earlier, we tested the validity of four different feature groups for automatic irony detection, comprising lexical, syntactic, sentiment and semantic features. Finally, all feature groups were combined to see whether they provide complementary information to the classifier. Each feature group was evaluated in two experimental setups: one where irony hashtags like *#irony*, *#sarcasm* and *#not* were removed from the corpus, and another where this hashtag information was included.

5.1 Experimental Results on the Held-out Test Set

Table 2 presents the results obtained with each feature group separately, and with the combined set. Besides accuracy, we report F_1-score, precision and recall (on the positive class) as the evaluation metrics.

For the baseline system, we included only bag-of-words features. No parameter tuning was done. As the system consistently predicts the positive class (i.e., *ironic*), its recall is 100% while its accuracy is equal to the positive class distribution.

	Setup 1: without hashtag information				*Setup 2: with hashtag information*			
Features	**Accuracy**	**F_1-score**	**Precision**	**Recall**	**Accuracy**	**F_1-score**	**Precision**	**Recall**
BoW (baseline)	48.96	65.73	48.96	100	48.96	65.73	48.96	100
Lexical	66.91	66.38	66.03	66.74	90.92	91.41	85.11	98.72
Syntactic	63.57	64.57	61.63	67.80	80.48	81.68	75.54	88.91
Sentiment	59.29	55.78	59.56	52.45	74.95	71.22	81.37	63.33
Semantic	64.41	63.53	63.73	63.33	88.73	89.45	82.52	97.65
Combined	68.16	**67.66**	67.30	68.02	92.48	**92.77**	87.67	98.51

Table 2: Experimental results on the held-out test set in both setups.

Evidently, the experimental setup with hashtags included (setup 2) performs best. To understand the performance of this system better, we compare its accuracy (92.48%) to a baseline that simply classifies all tweets containing an irony-related hashtag as ironic and all other tweets as not ironic, yielding an accuracy of 87%. In both setups, the best performance is achieved by the combined feature set (F= 67.66 and 92.77). This partly supports the findings of Wallace (2015) that verbal irony cannot be recognised through lexical clues alone. Nevertheless, lexical information does seem to be of key importance for this task, as the corresponding system obtained the second best score (F= 66.38 and 91.41). An explanation could be the nature of Twitter data: due to its limited length, a tweet is prone to be misunderstood, which may encourage people to use explicit lexical clues when speaking ironically. In both setups, sentiment features perform least well for this task (F= 55.78 and 71.22), which demonstrates that (explicit) polarity

information is not sufficient for decent irony recognition. Nevertheless, the annotations revealed that many ironic tweets showed a polarity contrast (mostly between evaluations and *targets*). In future work, we will therefore explore methods to model this clash between explicit and implicit (i.e., inferred from world knowledge) sentiment expressions.

5.2 Analysis of the Results

Judging from the raw performance results, irony detection in Twitter data benefits from a combined set of information sources (i.e., lexical, syntactic, sentiment and semantic). In a next step, we investigate whether the combined system is significantly better than the baseline and the systems we built based on each feature group. To this end, we applied 10 paired samples t-test (α= 0.05) after bootstap resampling (Noreen, 1989): one for each system (including the baseline) that we compare against the combined system, in the two experimental setups. Concretely, we drew samples (n= 10,000) with replacement from the output of each system and of equal size of the output (n= 958, the number of test instances). Each sample was then evaluated using macro-averaged F_1-score (on the positive class), after which we applied a paired samples t-test to compare the mean scores for both systems. We found a significant difference ($p<$ 0.05) for all system pairs.

For the experiments in this paper, we concentrated on two classification labels being ironic and not ironic, thus casting the problem into a detection (or binary classification) task. To this end, the annotation labels *ironic by clash*, *other type of irony* and *situational irony* were combined into the positive class in both our training and test sets. In the following paragraph, we zoom in on these 'subclasses' of irony and evaluate the performance of our system on each one of them in the test set. The subclasses are represented by 346 (*irony by clash*), 62 (*other irony*) and 61 (*situational irony*) instances in the test set. We will also take a closer look at the two types of non-ironic tweets (i.e., with and without an irony-related hashtag) in the dataset.

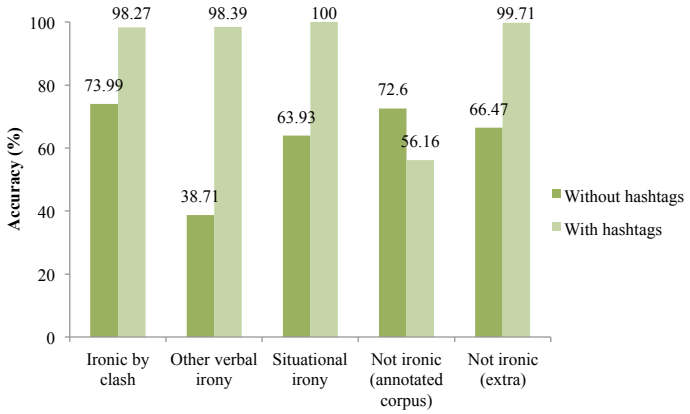


Figure 1: System accuracy on different types of ironic and non-ironic tweets.

As shown in Figure 1, the performance of the system increases significantly when hashtag information is included, reaching an accuracy of up to 100% for the recognition of tweets that describe situational irony. Without this hashtag information, the best performance is achieved on *ironic by clash* (acc.= 73.99%), followed by *situational irony* (acc.= 63.93%). The score on *other verbal irony* is low, however (acc.= 38.71%). This would suggest that detecting verbal irony is (much) more feasible when the irony results from contrasting evaluations, as opposed to other types of verbal irony.

We also had a closer look at the category 'not ironic'. Important to recapitulate is that 25% of these tweets contain an irony hashtag (they were part of the originally collected 3,000 tweets (cf. Table1)). When looking at the classification performance on these tweets, we observe that, when irony hashtags are included in the data, the accuracy obtained is 56%. This demonstrates that the system does not simply rely on hashtag information (since this would result in an accuracy of 0% on this category). Another category that is subject to a qualitative analysis, are the tweets for which annotators had indicated that an

explicit hashtag (e.g., *#irony*) was required to recognise the irony. Intuitively, the system would perform better on the instances where no such hashtag is needed, especially when these hashtags are removed from the data. Effectively, this hypothesis was confirmed by our experiments (setup 1), revealing a higher accuracy on ironic tweets where no hashtag was required to recognise the irony (83.43%) than on those where it was (63.64%). The accuracy on the latter still being relatively high, we can conclude that the irony in our dataset is moderately to strongly lexicalised. This conclusion is in line with the performance of the system exploiting lexical features (cf. Table 2).

We see that our combined system (F_1= 67.66) compares favourably to that of Riloff et al. (2013) (F_1= 0.51), who describe a similar experimental setup to the one presented here. However, comparison with state of the art is not trivial, given the size of our dataset and the different definitions of irony that are used among researchers. In effect, most studies make use of large corpora that are labeled based on hashtag information (e.g., Kunneman et al. (2015), Reyes et al. (2013)). Furthermore, some approaches (e.g., Riloff et al. (2013)) focus on one particular type of irony, whereas the present research takes different types into account.

6 Conclusion and Future Work

In this paper we developed and tested a system for irony detection in English Twitter data. We collected 3,000 tweets with irony hashtags (i.e., *#irony*, *#sarcasm*, and *#not*) and manually annotated them according to a newly developed annotation scheme for verbal irony. To balance the ironic vs. not ironic classes in the experiments, another 1,792 non-ironic tweets from a random Twitter sample were added, which resulted in an experimental corpus of 4,792 tweets. We explored the viability of automatic irony detection using different feature groups (lexical, syntactic, sentiment, semantic and combined). Additionally, we compared the system's performance on our dataset with and without removing the irony-related hashtags. The results on a held-out test set revealed that irony detection benefits from a combined feature set: our binary classifier yields an F_1-score of 67.66% on the dataset devoid of irony hashtags, while obtaining F_1= 92.77% with irony hashtags included in the dataset. Although lexical features are assumed insufficient for decent irony recognition (Wallace, 2015), we experimentally show that they do provide relevant information, as the corresponding system scored second best, after the combined system.

A qualitative analysis of the different types of ironic tweets revealed that our classifier performed best on tweets where the irony results from a polarity contrast (i.e., the polarity of the expressed sentiment is opposite to what is meant). Given that ironic tweets are prone to implicit sentiment, future research will focus on recognising and understanding such implicit evaluations by making use of world knowledge or common sense. This would allow to identify a polarity contrast in typically ironic utterances where a part of the evaluation is implicit, such as *monday mornings* in the sentence *Monday mornings are my fave! :)*.

Acknowledgements

The work presented in this paper was carried out in the framework of the AMiCA (IWT SBO-project 120007) project, funded by the government agency for Innovation by Science and Technology (IWT).

References

Salvatore Attardo. 2000. Irony as relevant inappropriateness. *Journal of Pragmatics*, 32(6):793–826.

Francesco Barbieri and Horacio Saggion. 2014. Modelling Irony in Twitter: Feature Analysis and Evaluation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC'14, Reykjavik, Iceland. European Language Resources Association (ELRA).

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27.

Rebecca Clift. 1999. Irony in conversation. *Language in Society*, 28(4):523–553.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised Recognition of Sarcastic Sentences in Twitter and Amazon. In *Proceedings of the 14th Conference on Computational Natural Language Learning*, CoNLL'10, pages 107–116, Uppsala, Sweden. Association for Computational Linguistics.

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

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, HLT'11, pages 42–47, Portland, Oregon. Association for Computational Linguistics.

Rachel Giora. 1995. On irony and negation. *Discourse Processes*, 19(2):239–264.

H. P. Grice. 1975. Logic and Conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics: Vol. 3*, pages 41–58. Academic Press, New York.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD'04, pages 168–177, New York, NY. Association for Computing Machinery.

Sathiya S. Keerthi and Chih-Jen Lin. 2003. Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel. *Neural Computation*, 15(7):1667–1689, July.

Nadin Kökciyan, Arda Çelebi, Arzucan Özgür, and Suzan Üsküdarl. 2013. Bounce: Sentiment classification in Twitter using rich feature sets. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the 7th International Workshop on Semantic Evaluation*, SemEval 2013, pages 554–561, Atlanta, Georgia, USA. Association for Computational Linguistics.

Roger J. Kreuz and Sam Glucksberg. 1989. How to be sarcastic: The echoic reminder theory of verbal irony. *Journal of Experimental Psychology: General*, 118(4):374.

Roger J. Kreuz and Richard M. Roberts. 1993. On Satire and Parody: The Importance of Being Ironic. *Metaphor and Symbolic Activity*, 8(2):97–109.

Florian Kunneman, Christine Liebrecht, Margot van Mulken, and Antal van den Bosch. 2015. Signaling sarcasm: From hyperbole to hashtag. *Information Processing & Management*, 51(4):500–509.

Christopher J. Lee and Albert N. Katz. 1998. The Differential Role of Ridicule in Sarcasm and Irony. *Metaphor and Symbol*, 13(1):1–15.

Bing Liu. 2015. *Sentiment Analysis - Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

Joan Lucariello. 1994. Situational Irony: A Concept of Events Gone Awry. *Journal of Experimental Psychology: General*, 123(2):129–145.

Diana Maynard and Mark Greenwood. 2014. Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, LREC'14, Reykjavik, Iceland.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Finn Årup Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses : An Introduction*. Wiley-Interscience.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in Twitter. *Language Resources and Evaluation*, pages 238–269.

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 Conference on Empirical Methods in Natural Language Processing*, EMNLP'13, pages 704–714. Association for Computational Linguistics.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP'11, pages 1524–1534, Edinburgh, United Kingdom. Association for Computational Linguistics.

Dan Sperber and Deirdre Wilson. 1981. Irony and the Use - Mention Distinction. In Peter Cole, editor, *Radical Pragmatics*, pages 295–318. Academic Press, New York.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: A Web-based Tool for NLP-assisted Text Annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL'12, pages 102–107, Avignon, France. Association for Computational Linguistics.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. The General Inquirer: A Computer Approach to Content Analysis. *MIT Press*.

Marjan Van de Kauter, Geert Coorman, Els Lefever, Bart Desmet, Lieve Macken, and Véronique Hoste. 2013. LeTs Preprocess: The multilingual LT3 linguistic preprocessing toolkit. *Computational Linguistics in the Netherlands Journal*, 3:103–120.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2016a. Exploring the Realization of Irony in Twitter Data. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, LREC'16, pages 1794–1799, Portorož (Slovenia). European Language Resources Association (ELRA).

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2016b. Guidelines for Annotating Irony in Social Media Text, version 2.0. Technical Report 16-01, LT3, Language and Translation Technology Team–Ghent University.

Byron C. Wallace. 2015. Computational irony: A survey and new perspectives. *Artificial Intelligence Review*, 43(4):467–483.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT'05, pages 347–354, Vancouver, Canada. Association for Computational Linguistics.