Modelling Irony in Twitter: Feature Analysis and Evaluation

Francesco Barbieri, Horacio Saggion

Pompeu Fabra University
Barcelona, Spain
francesco.barbieri@upf.edu, horacio.saggion@upf.edu

Abstract

Irony, a creative use of language, has received scarce attention from the computational linguistics research point of view. We propose an automatic system capable of detecting irony with good accuracy in the social network Twitter. Twitter allows users to post short messages (140 characters) which usually do not follow the expected rules of the grammar, users tend to truncate words and use particular punctuation. For these reason automatic detection of Irony in Twitter is not trivial and requires specific linguistic tools. We propose in this paper a new set of experiments to assess the relevance of the features included in our model. Our model does not include words or sequences of words as features, aiming to detect inner characteristic of Irony.

Keywords: Computational Creativity, Irony Detection, Social Media

1. Introduction

Computational Creativity is of great importance to Computational Linguistics, and it becomes even more significant when studied in social networks, one of the most popular means of expression nowadays. In particular, irony is a very interesting phenomenon as it exposes the problems that current machines have in detecting the intended rather than the literal meaning of a sentence. By the use of irony, people hide the real meaning of a statement saying the opposite of what they mean (Quintilien and Butler, 1953), and this is what the current automatic systems still struggle to detect.

Only recently irony detection has been approached from a computational perspective. Reyes et al. (2013) cast the problem as a classification, designing a Machine Learning algorithm that separates ironic from non-ironic statements. In a similar vein, we proposed and evaluated a new model to detect irony (Barbieri and Saggion, 2014), using seven sets of lexical features. We study irony detection in the microblogging service Twitter[1] that allows users to send and read text messages (shorter than 140 characters) called tweets, which often do not follow the expected rules of the grammar. Indeed, Twitter users tend to truncate words and use particular punctuation. For these reason automatic detection of Irony in Twitter is not trivial and requires specific linguistic tools. We use the same dataset as Reyes et. al (2013). This dataset contains positive examples tagged as ironic by the users (using the hashtag #irony) and negative examples (tagged with a different hashtag). We illustrate with a few ironic examples the kind of irony contained in the dataset:

1. Good morning everyone. Yes, I could really, really, *really* get used to having only 3 hours sleep *sigh*
2. Bush sent more troops than Obama to create Peace in Afghanistan but Obama got the NOBEL!
3. I'll tell you a secret... I love Christmas!
4. It's easier to install Windows on a Mac than it is on a PC
5. Twitter was down and I couldn't tweet about it. addicted
6. DidYouKnow: The Bible is the most shoplifted book.

In Example 1 the user says that he can get used to sleep only three hours per night, but he is clearly meaning the opposite: three hours of sleep is almost not sleeping. In Example 2 the author ironically underlines that the use of military troops to create peace seems conflicting; indeed president Obama received the Nobel Peace price even though he used less troops than Bush to create peace in Afghanistan. Example 3 is ironic because most of the people like Christmas and loving it should not be a secret. In Example 4 the user finds using a Mac simpler than using a PC, even to install Windows. This is ironic because Windows is competitor of Mac, and it has not been designed to work on Macs. Example 6 is another ironic situation where the user would like to tell everyone via Twitter that Twitter is not working. And last, Example 6 is the ironic fact that even if the Bible should be the book that teaches the good behaviour (including "not stealing") it is the most stolen book in bookstores.

Overall we can distinguish two kinds of tweets. 1-3 are examples of verbal irony: these tweets are actually ironic sentences. On the other hand, examples 4-6 are not ironic utterances but descriptions of ironic situations (situational irony). The reader can also observe that in some examples detecting irony is not trivial.

Initial experiments on Irony detection are reported in Barbieri and Saggion (2014). In this paper we run further experiments to investigate the contribution of our features, studying them not only as single features (single information gain values) like in the previous research, but as collection of features. We also study redundancy and importance over different topics. The contributions of this paper is a set of new experiments in order to evaluate the features the computational model to detect Irony includes.

[1] https://twitter.com/

The rest of the paper is organised as follows: in the next Section we describe related work. In Section 3 we described the corpus and text processing tools used and in Section 4 we present our approach to tackle the Irony detection problem. Section 5 describes the experiments while Section 6 interprets the results. Finally we close the paper in Section 7 with conclusions and future work.

2. Related Work

Irony has been defined in several ways over the years but there is no consensual agreement on the definition. The standard definition is considered "saying the opposite of what you mean" (Quintilien and Butler, 1953) where the opposition of literal and intended meanings is very clear. Grice (1975) believes that irony is a rhetorical figure that violates the maxim of quality: "Do not say what you believe to be false". Irony is also defined (Giora, 1995) as any form of negation with no negation markers (as most of the ironic utterances are affirmative, and ironic speakers use indirect negation). Wilson and Sperber (2002) defined it as echoic utterance that shows a negative aspect of someone's else opinion. For example if someone states "the weather will be great tomorrow" and the following day it rains, someone with ironic intents may repeat the sentence "the weather will be great tomorrow" in order to show the statements was incorrect. Finally irony has been defined as form of pretence by Utsumi (2000) and by Veale and Hao (2010a). Veale states that "ironic speakers usually craft their utterances in spite of what has just happened, not because of it. The pretence alludes to, or echoes, an expectation that has been violated".

Past computational approaches to irony detection are not many. Carvalho et. al (2009) created an automatic system for detecting irony relying on emoticons and special punctuation. They focused on detection of ironic style in newspaper articles. Veale and Hao (2010b) proposed an algorithm for separating ironic from non-ironic similes, detecting common terms used in this ironic comparison. Reyes et. al (2013) have recently proposed a model to detect irony in Twitter, which is based on four groups of features: signatures, unexpectedness, style, and emotional scenarios. Their classification results support the idea that textual features can capture patterns used by people to convey irony. Among the proposed features, *skip-grams* (part of the Style group) which captures word sequences that contain (or skip over) arbitrary gaps, seems to be the best one.

Some computational model to detect sarcasm in Twitter have been designed in the past years. The systems of Gonzalez et. al (2011) and Davidov et. al (2010) detect sarcasm with good accuracy in English tweets (the latter model is also studied in the Amazon review context). Lukin and Walker (2013) used bootstrapping to improve the performance of sarcasm and nastiness classifiers for Online Dialogue, and Liebrecht et. al (2013) designed a model to detect sarcasm in Duch tweets. One may argue that sarcasm and irony are the same linguistic phenomena, but in our opinion the latter is more similar to mocking or making jokes (sometimes about ourselves) in a sharp and non-offensive manner. On the other hand, sarcasm is a meaner form of irony as it tends to be offensive and directed towards other people (or products like in Amazon reviews). Textual examples of sarcasm lack the sharp tone of an aggressive speaker, so for textual purposes we think irony and sarcasm should be considered as different phenomena and studied separately (Reyes et al., 2013).

Finally, a few corpus of Irony and Sarcasm has been created. Filatova (2012) designed a corpus generation experiment where regular and sarcastic Amazon product reviews were collected. Also Bosco et. al (2013) collected and annotate a set of ironic examples (in italian) for the study of sentiment analysis and opinion mining.

3. Data and Text Processing

The dataset used for the experiments reported in this paper has been prepared by Reyes et al. (2013). It is a corpus of 40.000 tweets equally divided into four different topics: *Irony*, *Education*, *Humour*, and *Politics* where the last three topics are considered non-ironic. The tweets were automatically selected by looking at Twitter hashtags (#irony, #education, #humour, and #politics) added by users in order to link their contribution to a particular subject and community. The hashtags are removed from the tweets for the experiments. According to Reyes et. al (2013), these hashtags were selected for three main reasons: (i) to avoid manual selection of tweets, (ii) to allow irony analysis beyond literary uses, and because (iii) irony hashtag may "reflect a tacit belief about what constitutes irony."

Another corpora is employed in our approach to measure the frequency of word usage. We adopted the Second Release of the American National Corpus Frequency Data[2] (Ide and Suderman, 2004), which provides the number of occurrences of a word in the written and spoken ANC. From now on, we will mean with "frequency of a term" the absolute frequency the term has in the ANC.

Processing microblog text is not easy because they are noisy, with little context, and often English grammar rules are violated. For these reasons, in order to process the tweets, we use the GATE Plugin TwitIE (Bontcheva et al., 2013) as tokeniser and Part of Speech Tagger. The POS tagger (adapted version of the Stanford tagger (Toutanova et al., 2003)) achieves 90.54% token accuracy, which is a very good results knowing the difficulty of the task in the microblogging context. This POS tagger is more accurate and reliable than the method we used in the previous research (Barbieri and Saggion, 2014), where the POS of a term was defined by the most commonly used (provided by WordNet). TwitIE also includes the best Named Entity Recognitions for Twitter (F1=0.8).

We adopted also Rita WordNet API (Howe, 2009) and Java API for WordNet Searching (Spell, 2009) to perform operations on WordNet synsets.

4. Methodology

We approach the detection of irony as a classification problem applying supervised machine learning methods to the Twitter corpus described in Section 3. When choosing the

[2]The American National Corpus (http://www.anc.org/) is, as we read in the web site, a massive electronic collection of American English words (15 million)

classifiers we had avoided those requiring features to be independent (e.g. Naive Bayes) as some of our features are not. Since we approach the problem as a binary decision (deciding if a tweet is ironic or not) we picked a tree-based classifiers: Decision Tree. We already studied the performance of another classifier (Random Forest) but even if Random Forest performed better in cross validation experiments, Decision Tree resulted better in cross domain experiments, suggesting that it would be more reliable in a real situation (where the negative topics are more than one). We use the Decision Tree implementation of the Weka toolkit (Witten and Frank, 2005).

Our model uses seven groups of features to represent each tweet. Some of them are designed to detect imbalance and unexpectedness, others to detect common patterns in the structure of the ironic tweets (like type of punctuation, length, emoticons). Below is an overview of the group of features in our model:

- Frequency *(gap between rare and common words)*
- Written-Spoken *(written-spoken style uses)*
- Intensity *(intensity of adverbs and adjectives)*
- Structure *(length, punctuation, emoticons)*
- Sentiments *(gap between positive and negative terms)*
- Synonyms *(common vs. rare synonyms use)*
- Ambiguity *(measure of possible ambiguities)*

In our knowledge Frequency, Written Spoken, Intensity and Synonyms groups have not been used before in similar studies. The other groups have been used already (for example by Carvalho et. al (2009) or Reyes et al. (2013)) yet our implementation is different in most of the cases.

In the following sections we quickly describe all the features we used. The reader can access the full description and the theoretical motivations behind the features in Barbieri and Saggion (2014).

4.1. Frequency

Unexpectedness can be a signal of irony, Lucariello (1994) claims that irony is strictly connected to surprise, showing that unexpectedness is the feature most related to situational ironies. In this first group of features we try to detect it. We explore the frequency imbalance between words, i.e. register inconsistencies between terms of the same tweet. The idea is that the use of many words commonly used in English (i.e. high frequency in ANC) and only a few terms rarely used in English (i.e. low frequency in ANC) in the same sentence creates imbalance that may cause unexpectedness, since within a single tweet only one kind of register is expected. We are able to explore this aspect using the ANC Frequency Data corpus.

Three features belong to this group: **frequency mean**, **rarest word**, **frequency gap**. The first one is the arithmetic average of all the frequencies of the words in a tweet, and it is used to detect the *frequency style* of a tweet. The second one, **rarest word**, is the frequency value of the rarest word, designed to capture the word that may create imbalance. The assumption is that very rare words may be a sign of irony. The third one is the absolute difference between the first two and it is used to measure the imbalance between them, and capture a possible intention of surprise.

4.2. Written-Spoken

Twitter is composed of written text, but an informal spoken English style is often used. We designed this set of features to explore the unexpectedness created by using spoken style words in a mainly written style tweet or vice versa (formal words usually adopted in written text employed in a spoken style context). We can analyse this aspect with ANC written and spoken, as we can see using this corpora whether a word is more often used in written or spoken English. There are three features in this group: **written mean**, **spoken mean**, **written spoken gap**. The first and second ones are the means of the frequency values, respectively, in written and spoken ANC corpora of all the words in the tweet. The third one, **written spoken gap**, is the absolute value of the difference between the first two, designed to see if ironic writers use both styles (creating imbalance) or only one of them. A low difference between written and spoken styles means that both styles are used.

4.3. Structure

With this group of features we want to study the structure of the tweet: if it is long or short (length), if it contains long or short words (mean of word length), and also what kind of punctuation is used (exclamation marks, emoticons, etc.). This is a powerful feature, as ironic tweets in our corpora present specific structures: they are often longer than the tweets in the other corpora, they contain certain kind of punctuation and they use only specific emoticons.

The **length** feature consists of the number of characters that compose the tweet, **n. words** is the number of words, and **words length mean** is the mean of the words length. Moreover, we use the number of verbs, nouns, adjectives and adverbs as features, naming them **n. verbs**, **n. nouns**, **n. adjectives** and **n. adverbs**. With these last four features we also computed the ratio of each part of speech to the number of words in the tweet; we called them **verb ratio**, **noun ratio**, **adjective ratio**, and **adverb ratio**. All these features have the purpose of capturing the style of the writer.

The **punctuation** feature is the sum of the number of commas, full stops, ellipsis and exclamation that a tweet presents. We also added a feature called **laughing** which is the sum of all the internet laughs, denoted with *hahah*, *lol*, *rofl*, and *lmao* that we consider as a new form of punctuation: instead of using many exclamation marks internet users may use the sequence *lol* (i.e. laughing out loud) or just type *hahaha*.

The **emoticon** feature is the sum of the emoticons *:)*, *:D*, *:(* and *;)* in a tweet. The ironic corpus is the one with the least emoticons probably because ironic authors avoid emoticons and leave words to be central: the audience has to understand the irony without explicit signs, like emoticons.

4.4. Intensity

In order to produce an ironic effect some authors might use an expression which is antonymic to what they are trying to

describe (saying the opposite of what they mean (Quintilien and Butler, 1953)). We believe that in the case the word being an adjective or adverb its intensity (more or less exaggerated) may well play a role in producing the intended effect. We adopted the intensity scores of Potts (2011) who uses naturally occurring metadata (star ratings on service and product reviews) to construct adjectives and adverbs scales. An example of adjective scale (and relative scores in brackets) could be the following: horrible (-1.9) → bad (-1.1) → good (0.2) → nice (0.3) → great (0.8).
With these scores we evaluate four features for adjective intensity and four for adverb intensity (implemented in the same way): **adj (adv) tot**, **adj (adv) mean**, **adj (adv) max**, and **adj (adv) gap**. The sum of the AdjScale scores of all the adjectives in the tweet is called **adj tot**. **adj mean** is **adj tot** divided by the number of adjectives in the tweet. The maximum AdjScale score within a single tweet is **adj max**. Finally, **adj gap** is the difference between **adj max** and **adj mean**, designed to see "how much" the most intense adjective is out of context.

4.5. Synonyms

Ironic authors send two messages to the audience at the same time, the literal and the figurative one (Veale, 2004). It follows that the choice of a term (rather than one of its synonyms) is very important in order to send the second, not obvious, message.
For each word of a tweet we get its synonyms with WordNet (Miller, 1995), then we calculate their ANC frequencies and sort them into a decreasing ranked list (the actual word is part of this ranking as well). We use these rankings to define the four features which belong to this group. The first one is **syno lower** which is the number of synonyms of the word w_i with frequency lower than the frequency of w_i. It is defined as in Equation 1:

$$sl_{w_i} = |syn_{i,k} \ : \ f(syn_{i,k}) < f(w_i)| \qquad (1)$$

where $syn_{i,k}$ is the synonym of w_i with rank k, and $f(x)$ the ANC frequency of x. Then we also defined **syno lower mean** as mean of sl_{w_i} (i.e. the arithmetic average of sl_{w_i} over all the words of a tweet).
We also designed two more features: **syno lower gap** and **syno greater gap**, but to define them we need two more parameters. The first one is *word lowest syno* that is the maximum sl_{w_i} in a tweet. It is formally defined as:

$$wls_t = \max_{w_i}\{|syn_{i,k} \ : \ f(syn_{i,k}) < f(w_i)|\} \qquad (2)$$

The second one is *word greatest syno* defined as:

$$wgs_t = \max_{w_i}\{|syn_{i,k} \ : \ f(syn_{i,k}) > f(w_i)|\} \qquad (3)$$

We are now able to describe **syno lower gap** which detects the imbalance that creates a common synonym in a context of rare synonyms. It is the difference between *word lowest syno* and **syno lower mean**. Finally, we detect the gap of very rare synonyms in a context of common ones with **syno greater gap**. It is the difference between *word greatest syno* and *syno greater mean*, where *syno greater mean* is the following:

$$sgm_t = \frac{|syn_{i,k} \ : \ f(syn_{i,k}) > f(w_i)|}{n.\,words\,of\,t} \qquad (4)$$

The arithmetic averages of **syno greater gap** and of **syno lower gap** in the irony corpus are higher than in the other corpora, suggesting that a very common (or very rare) synonym is often used out of context i.e. a very rare synonym when most of the words are common (have a high rank in our model) and vice versa.

4.6. Ambiguity

Another interesting aspect of irony is ambiguity. We noticed that ironic tweets presents words with more meanings (more WordNet synsets). Our assumption is that if a word has many meanings the possibility of "saying something else" with this word is higher than in a term that has only a few meanings, then higher possibility of sending more then one message (literal and intended) at the same time.
There are three features that aim to capture these aspects: **synset mean**, **max synset**, and **synset gap**. The first one is the mean of the number of synsets of each word of the tweet, to see if words with many meanings are often used in the tweet. The second one is the greatest number of synsets that a single word has; we consider this word the one with the highest possibility of being used ironically (as multiple meanings are available to say different things). In addition, we calculate **synset gap** as the difference between the number of synsets of this word (**max synset**) and the average number of synsets (**synset mean**), assuming that if this gap is high the author may have used that inconsistent word intentionally.

4.7. Sentiments

We think that sign of irony could also be found using sentiment analysis. The SentiWordNet sentiment lexicon (Esuli and Sebastiani, 2006) assigns to each synset of WordNet sentiment scores of positivity and negativity. We used these scores to examine what kind of sentiments characterises irony. We explore ironic sentiments with two different views: the first one is the simple analysis of sentiments (to identify the main sentiment that arises from ironic tweets) and the second one concerns sentiment imbalances between words, designed to explore unexpectedness from a sentiment prospective.
There are six features in the Sentiments group. The first one is named **positive sum** and it is the sum of all the positive scores in a tweet, the second one is **negative sum**, defined as sum of all the negative scores. The arithmetic average of the previous ones is another feature, named **positive negative mean**, designed to reveal the sentiment that better describe the whole tweet. Moreover, there is **positive-negative gap** that is the difference between the first two features, as we wanted also to detect the positive/negative imbalance within the same tweet.
The imbalance may be created using only one single very positive (or negative) word in the tweet, and the previous features will not be able to detect it, thus we needed to add two more. For this purpose the model includes **positive single gap** defined as the difference between most positive word and the mean of all the sentiment scores of all the

	Education			Humour			Politics		
Model	P	R	F1	P	R	F1	P	R	F1
Experiment 1	.72	.72	.72	.75	.75	.75	.76	.76	**.76**
Experiment 2	.73	.73	**.73**	.75	.75	.75	.74	.74	.74

Table 1: Experiments 1 and 2. Precision, Recall, and F-Measure over the three corpora Education, Humour. The classifier used is Decision Tree. Results are very similar, but the model of Experiment 2 uses less features.

words of the tweet and **negative single gap** defined in the same way, but with the most negative one.

5. Experiments and Results

To carry out our experiments we use three datasets derived from the corpus in Section 3: Irony vs Education, Irony vs Humour and Irony vs Politics. Each topic combination was balanced with 10.000 ironic and 10.000 of non-ironic examples. We perform two types of experiments:

1. We train the classifier on 7500 positive examples and 7500 negative examples of the same dataset, then we use as test set the rest 2500 positive and 2500 negative. We perform this experiment for the three datasets.

2. We apply to each training set used in Experiment 1 a feature selection (Correlation-based Feature Subset Selection method (Hall and Smith, 1998), Best First as search algorithm), leaving out the least relevant features for each topic. Then we test on the test set of Experiment 1.

In Table 1 we compare Experiment 1 and Experiment 2. Table 2 shows the three confusion matrices of Experiment 1. Even if the datasets are balanced we decided to add the confusion matrices to make possible future comparisons with other systems. Table 3 includes the Pearson correlations of the information gain of each feature between datasets. Table 4 illustrates the single features selected in Experiment 2 for each dataset.

6. Discussion

The features which are more discriminative of ironic style are **rarest value**, **synonym lower**, **synonym greater gap**, and **punctuation**, suggesting that Frequency, Structure and choice of the Synonym are important aspects to consider for irony detection in tweets. However, there is a topic or theme effect since features behave differently depending on the dataset used: the Humour corpus seems to be the least consistent. For instance **punctuation** well distinguishes ironic from educational tweets, but behaves poorly in the Humour corpus. This theme effect is seen in Table 3 where Education-Politics are strongly correlated but Humour-Education and Humour-Politics show respectively a weak and moderate correlation. Hence, the important features for Humour are not the same than for Education and Politics. Finding features that are significant for any non-ironic topic is hard, this is why we need to consider several features in our model.

From one side having more features may help to cover and detect more negative topics, but it also increases the complexity of the model, introducing possible redundancies between features. For this reason we applied a feature selection (Experiment 2) and studied the performances of models that were using less features. The performances of complete and filtered model are comparable (Table 1), then if the task was distinguishing Irony from only one negative topic, we could have used less features. Yet, as said previously, we wanted to design a model capable to detect irony in different circumstances. The features selected in each dataset are shown in Table 4. Some of the features result important for all the topics, others for none. One could argue that these latter should not be part of the model, but again, they can be redundant only in the three topics tested and they may be highly discriminative to other kind of text not yet considered.

			Predicted			
Actual	**Iro**	**Edu**	**Iro**	**Hum**	**Iro**	**Pol**
Iro	1777	723	**1876**	**624**	1857	643
NonIro	692	1808	621	1879	**583**	**1917**

Table 2: Confusion matrices of the three corpora Education, Humour and Politics of Experiment 1. The classifier used is Decision Tree.

	Education	**Humour**	**Politics**
Education	1	0.24	0.80
Humour	-	1	0.44
Politics	-	-	1

Table 3: Information gain Pearson Correlation of each feature over different topics when training Irony. Education and Politics are highly correlated, suggesting that similar features are used when trying to distinguish Irony vs Education and Irony vs Politics.

We can compare the behaviour of the classifier in the different topics looking at Table 2. We can see that Irony is well selected when the negative topic is Humour (best true positive score) and that Irony is not confused with Humour more than in the other topics (lowest number of false negatives). The confusion matrix shows also that Irony versus Politics obtains the lowest number of false positive, suggesting that politics tweets are not often misinterpreted as Ironic, confirmed also by the best number of correctly classified politics tweets (true negatives). The confusion matrices formalise also that Education is the most difficult topic for our classifier.

Regarding the use of TwitIE (Bontcheva et al., 2013) we found that it did not produce significant improvements in

	adj. gap	adj. max	adj. mean	adj. ratio	adj. tot	adv. max	adv. mean	adv. ratio	adv. tot	emoticons	freq. gap	freq. mean	laughing	max synset	n. adj.	n. adv.	n. noun	n. verb	n. words	neg. gap	neg. sum	noun ratio	pos-neg gap	pos-neg mean	pos. gap	pos. sum	punctuation	rarest val.	spok. mean	syno. gre. gap	syno. low gap	syno. low mean	syno. lower	synset gap	tw. length	verb ratio	word mean	wr-sp mean	writ. mean
Edu	X								X			X	X						X		X						X	X	X	X			X		X	X	X		
Hum			X			X				X																		X		X					X			X	
Pol	X		X			X							X					X				X				X	X	X		X			X	X		X		X

Table 4: Selected features by Correlation-based Feature Subset Selection (Best First as search algorithm), applied to each corpora (Irony vs Education/Humour/Politics)

the performance of the system. This is because the most important features are directly correlated to the structure (like punctuation) and not to POS for example. Nevertheless, the use of good linguistic tools give to our system reliability (i.e. if the POS of a term is more accurate, it will be so also the sentiment score calculated with SentiWordnet) and a better base for improvements.

7. Conclusion and Future Work

In this article we analysed with new experiments a novel model to detect Irony in the social network Twitter that we proposed in Barbieri and Saggion (2014). We also incorporated new linguistic tools to better deal with the complexity of Twitter messages, not always examples of correct and standard English. The features of our model take into account frequency, written/spoken differences, sentiments, ambiguity, intensity, synonymy and structure.

There is however much space for improvements. The ambiguity aspect is still weak in this research, and it needs to be improved. Also experiments adopting different corpora (Filatova, 2012) and different negative topics may be useful in order to explore the system behaviour in a real situation. Another aspect we want to investigate is the use of n-grams from huge collections to model "unexpected" word usage. Finally, we will run new experiments on sarcastic detection, as even if they may be different linguistic phenomena they may share some characteristics with irony and our model may have good chances to perform well on sarcasm too.

Acknowledgments

We are grateful to three anonymous reviewers for their comments and suggestions that help improve our paper. The research described in this paper is partially funded by fellowship RYC-2009-04291 from Programa Ramón y Cajal 2009 and project number TIN2012-38584-C06-03 (SKATER-UPF-TALN) from Ministerio de Economía y Competitividad, Secretaría de Estado de Investigación, Desarrollo e Innovacíon, Spain.

8. References

Barbieri, Francesco and Saggion, Horacio. (2014). Irony detection in Twitter. In *Proceedings of the European Chapter of the Association for Computational Linguistics (Student Workshop)*.

Bontcheva, Kalina, Derczynski, Leon, Funk, Adam, Greenwood, Mark A., Maynard, Diana, and Aswani, Niraj. (2013). TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. Association for Computational Linguistics.

Bosco, Cristina, Patti, Viviana, and Bolioli, Andrea. (2013). Developing corpora for sentiment analysis and opinion mining: the case of irony and senti-tut.

Carvalho, Paula, Sarmento, Luís, Silva, Mário J, and de Oliveira, Eugénio. (2009). Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM.

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

Esuli, Andrea and Sebastiani, Fabrizio. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of Language Resources and Evaluation Conference*, volume 6, pages 417–422.

Filatova, Elena. (2012). Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing. In *Proceedings of Language Resources and Evaluation Conference*, pages 392–398.

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

González-Ibáñez, Roberto, Muresan, Smaranda, and Wacholder, Nina. (2011). Identifying Sarcasm in Twitter: A Closer Look. In *ACL (Short Papers)*, pages 581–586. Citeseer.

Grice, H Paul. (1975). Logic and conversation. *1975*, pages 41–58.

Hall, Mark A and Smith, Lloyd A. (1998). Practical feature subset selection for machine learning.

Howe, Daniel C. (2009). Rita wordnet. Java based API to access Wordnet.

Ide, Nancy and Suderman, Keith. (2004). The American National Corpus First Release. In *Proceedings of the Language Resources and Evaluation Conference*.

Liebrecht, Christine, Kunneman, Florian, and van den Bosch, Antal. (2013). The perfect solution for detecting sarcasm in tweets# not. *WASSA 2013*, page 29.

Lukin, Stephanie and Walker, Marilyn. (2013). Really? well. apparently bootstrapping improves the per-

formance of sarcasm and nastiness classifiers for online dialogue. *NAACL 2013*, page 30.

Miller, George A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Potts, Christopher. (2011). Developing adjective scales from user-supplied textual metadata. *NSF Workshop on Restructuring Adjectives in WordNet. Arlington,VA*.

Quintilien and Butler, Harold Edgeworth. (1953). *The Institutio Oratoria of Quintilian. With an English Translation by HE Butler*. W. Heinemann.

Reyes, Antonio, Rosso, Paolo, and Veale, Tony. (2013). A multidimensional approach for detecting irony in Twitter. *Language Resources and Evaluation*, pages 1–30.

Spell, Brett. (2009). Java API for WordNet Searching (JAWS).

Toutanova, Kristina, Klein, Dan, Manning, Christopher D, and Singer, Yoram. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

Utsumi, Akira. (2000). Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony. *Journal of Pragmatics*, 32(12):1777–1806.

Veale, Tony and Hao, Yanfen. (2010a). An ironic fist in a velvet glove: Creative mis-representation in the construction of ironic similes. *Minds and Machines*, 20(4):635–650.

Veale, Tony and Hao, Yanfen. (2010b). Detecting Ironic Intent in Creative Comparisons. In *ECAI*, volume 215, pages 765–770.

Veale, Tony. (2004). The challenge of creative information retrieval. In *Computational Linguistics and Intelligent Text Processing*, pages 457–467. Springer.

Wilson, Deirdre and Sperber, Dan. (2002). Relevance theory. *Handbook of pragmatics*.

Witten, Ian H and Frank, Eibe. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.