Modelling Irony in Twitter

Francesco Barbieri
Pompeu Fabra University
Barcelona, Spain
francesco.barbieri@upf.edu

Horacio Saggion
Pompeu Fabra University
Barcelona, Spain
horacio.saggion@upf.edu

Abstract

Computational creativity is one of the central research topics of Artificial Intelligence and Natural Language Processing today. Irony, a creative use of language, has received very little attention from the computational linguistics research point of view. In this study we investigate the automatic detection of irony casting it as a classification problem. We propose a model capable of detecting irony in the social network Twitter. In cross-domain classification experiments our model based on lexical features outperforms a word-based baseline previously used in opinion mining and achieves state-of-the-art performance. Our features are simple to implement making the approach easily replicable.

1 Introduction

Irony, a creative use of language, has received very little attention from the computational linguistics research point of view. It is however considered an important aspect of language which deserves special attention given its relevance in fields such as sentiment analysis and opinion mining (Pang and Lee, 2008). Irony detection appears as a difficult problem since ironic statements are used to express the contrary of what is being said (Quintilien and Butler, 1953), therefore being a tough nut to crack by current systems. Being a creative form of language, there is no consensual agreement in the literature on how verbal irony should be defined. Only recently irony detection has been approached from a computational perspective. Reyes et al. (2013) cast the problem as one of classification training machine learning algorithms to sepatare ironic from non-ironic statements. In a similar vein, we propose and evaluate a new model to detect irony, using seven sets of lexical features, most of them based on our intuitions about "unexpectedness", a key component of ironic statements. Indeed, Lucariello (1994) claims that irony is strictly connected to surprise, showing that unexpectedness is the feature most related to situational ironies.

In this paper we reduce the complexity of the problem by studying irony detection in the micro-blogging service Twitter[1] that allows users to send and read text messages (shorter than 140 characters) called tweets.

We do not adopt any formal definition of irony, instead we rely on a dataset created for the study of irony detection which allows us to compare our findings with recent state-of-the-art approaches (Reyes et al., 2013).

The contributions of this paper are as follows:

- a novel set of linguistically motivated, easy-to-compute features
- a comparison of our model with the state-of-the-art; and
- a novel set of experiments to demonstrate cross-domain adaptation.

The paper will show that our model outperforms a baseline, achieves state-of-the-art performance, and can be applied to different domains.

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.

[1] https://twitter.com/

Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 56–64,
Gothenburg, Sweden, April 26-30 2014. ©2014 Association for Computational Linguistics

2 Related Work

Verbal 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 Veale and Hao (2010b). 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 scarce. 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 (2010a) 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.

There are also a few computational model that detect sarcasm ((Davidov et al., 2010); (González-Ibáñez et al., 2011); (Liebrecht et al., 2013)) on Twitter and Amazon, but even if one may argue that sarcasm and irony are the same linguistic phenomena, 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).

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.

3.1 Text Processing

In order to process the tweets we use the freely available *vinhkhuc* Twitter Tokenizer[3] which allows us to recognise words in each tweet. To part-of-speech tag the words, we rely on the Rita WordNet API (Howe, 2009) that associates to a word with its most frequently used part of speech. We also adopted the Java API for WordNet Searching

[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)

[3] https://github.com/vinhkhuc/Twitter-Tokenizer/blob/master/src/Twokenizer.java

(Spell, 2009) to perform some operation on WordNet synsets. It is worth noting that although our approach to text processing is rather superficial for the moment, other tools are available to perform deeper tweet linguistic analysis (Bontcheva et al., 2013; Derczynski et al., 2013).

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 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 two tree-based classifiers: Random Forest and Decision tree (the latter allows us to compare our findings directly to Reyes et. al (2013)). We use the implementations available in the Weka toolkit (Witten and Frank, 2005).

To represent each tweet we use six groups of features. 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 describe the theoretical motivations behind the features and how them have been implemented.

4.1 Frequency

As said previously unexpectedness can be a signal of irony and 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. We have verified that the mean of this gap in each tweet of the irony corpus is higher than in the other corpora.

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: for example they are often longer than the tweets in the other corpora, they contain certain kind of punctuation and they use only specific emoticons. This group includes several features that we describe below.

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. Some of them seem to be significant; for example the average length of an ironic tweet is 94.8 characters and the average length of education, humour, and politics tweets are respectively 82.0, 86.6, and 86.5. The words used in the irony corpus are usually shorter than in the other corpora, but they amount to more.

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*. As the previous features, punctuation and laughing occur more frequently in the ironic tweets than in the other topics.

The **emoticon** feature is the sum of the emoticons *:)*, *:D*, *:(* and *;)* in a tweet. This feature works well in the humour corpus because is the one that presents a very different number of them, it has four times more emoticons than the other corpora. The ironic corpus is the one with the least emoticons (there are only 360 emoticons in the Irony corpus, while in Humour, Education, and Politics tweets they are 2065, 492, 397 respectively). In the light of these statistics we can argue that ironic authors avoid emoticons and leave words to be the central thing: the audience has to understand the irony without explicit signs, like emoticons. Another detail is the number of winks *;)*. In the irony corpus one in every five emoticon is a wink, whereas in the Humour, Education and Politics corpora the number of winks are 1 in every 30, 22 and 18 respectively. Even if the wink is not a usual emoticon, ironic authors use it more often because they mean *something else* when writing their tweets, and a wink is used to suggest that something is hidden behind the words.

4.4 Intensity

A technique ironic authors may employ is saying the opposite of what they mean (Quintilien and Butler, 1953) using adjectives and adverbs to, for example, describe something very big to denote something very small (e.g. saying “Do we hike that *tiny* hill now?” before going on top of a very high mountain). In order to produce an ironic effect some authors might use an expression which is antonymic to what they are trying to describe, 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 example if the sky is grey and it is about to rain, someone with ironic intents may say "sublime weather today", choosing *sublime* over many different, more common, synonyms (like nice, good, very good and so on, that according to ANC are more used in English) to advise the listener that the literal meaning may not be the only meaning present. A listener will grasp this hidden information when he asks himself why a rare word like *sublime* was used in that context where other more common synonyms were available to express the same *literal* meaning.

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 the arithmetic average of the number of WordNet synsets in the irony corpus is greater than in all the other corpora; this indicates that ironic tweets presents words with more meanings. 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, de-

	Training Set								
	Education			**Humour**			**Politics**		
Test set	P	R	F1	P	R	F1	P	R	F1
Education	.85/.73	.84/.73	.84/.73	.57/.61	.53/.61	.46/**.61**	.61/.67	.56/.67	.51/**.67**
Humour	.64/.62	.51/.62	.58/**.62**	.85/.75	.85/.75	.85/.75	.65/.61	.59/.61	.55/**.60**
Politics	.61/.67	.58/.67	.55/**.67**	.55/.61	.60/.60	.56/**.60**	.87/.75	.87/.75	.87/.75

Table 1: Precision, Recall and F-Measure of each topic combination for word based algorithm and our algorithm in the form "Word Based / Ours". Decision Tree has been used as classifier for both algorithms. We marked in **bold** the results that, according to the t-test, are significantly better.

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

4.8 Bag of Words Baseline

Based on previous work on sentiment analysis and opinon classification (see (Pang et al., 2002; Dave et al., 2003) for example) we also investigate the value of using bag of words representations for irony classification. In this case, each tweet is represented as a set of word features. Because of the brevity of tweets, we are only considering presence/absence of terms instead of frequency-based representations based on $tf * idf$.

5 Experiments and Results

In order to carry out experimentation and to be able to compare our approach to that of (Reyes et al., 2013) 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. The task at hand it to train a classifier to identify ironic and non-ironic tweets.

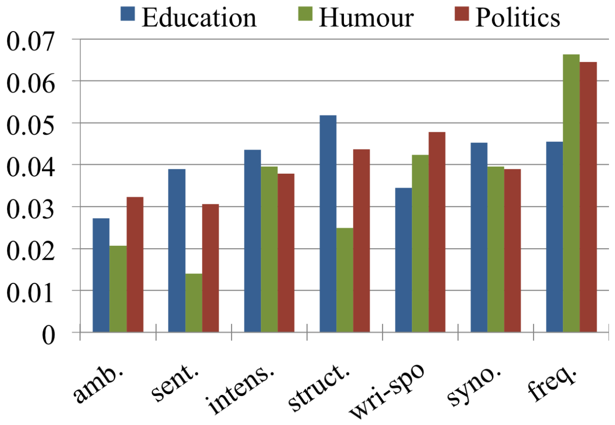

Figure 1: Information gain value of each group (mean of the features belonged to each group) over the three balanced corpus.

We perform two types of experiments:

- we run in each of the datasets a 10-fold cross-validation classification;
- across datasets, we train the classifier in one dataset and apply it to the other two datasets. To perform these experiments, we create three balanced datasets containing each one third of the original 10.000 ironic tweets (so that the datasets are disjoint) and one third of the original domain tweets.

The experimental framework is executed for the word-based baseline model and our model. In Table 1 we present precision, recall, and F-measure

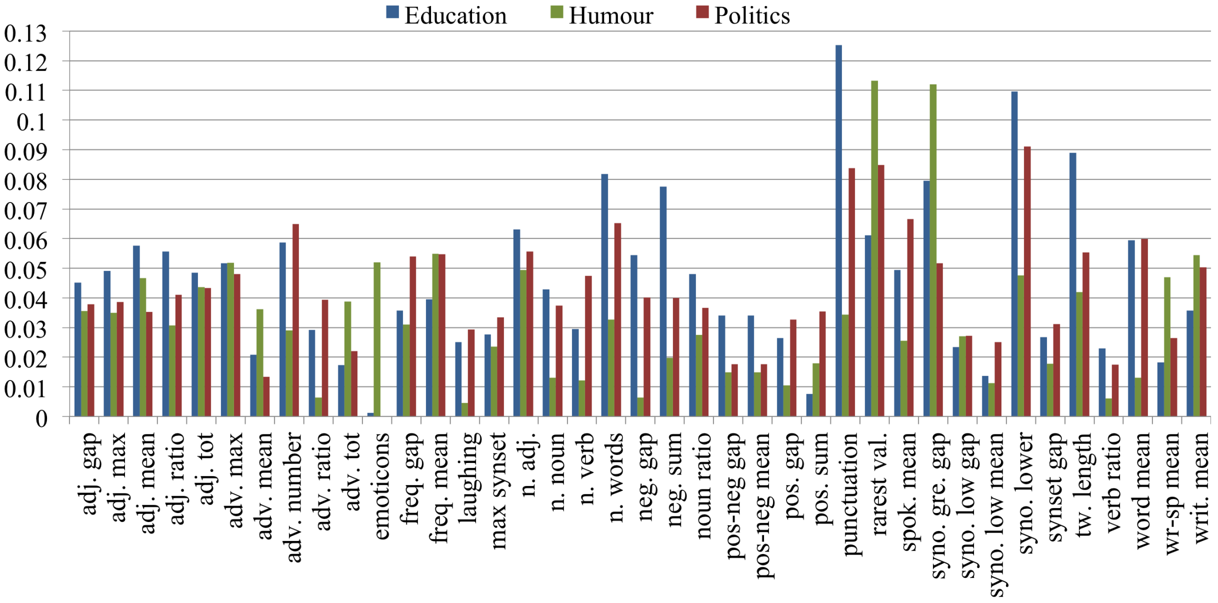

Figure 2: Information gain of each feature of the model. Irony corpus is compared to Education, Humor, and Politics corpora. High values of information gain help to better discriminate ironic from non-ironic tweets.

figures for the different runs of the experiments. Table 3 shows precision, recall, and F-measure figures for our approach compared to (Reyes et al., 2013). Table 2 compares two different algorithms: Decision Tree and Random Forest using our model.

In order to have a clear understanding about the contribution of each set of features in our model, we also studied the behaviour of information gain in each dataset. We compute information gain experiments over the three balanced corpora and present the results in Figure 1. The graphic shows the mean information gain for each group of features. We also report in Figure 2 the information gain of each single feature, where one can understand if a feature will be important to distinguish ironic from non-ironic tweets.

6 Discussion

The results obtained with the bag-of-words baseline seem to indicate that this approach is working as a topic-based classifier and not as an irony detection procedure. Indeed, within each domain using a 10 fold cross-validation setting, the bag-of-words approach seems to overtake our model. However, a clear picture emerges when a cross-domain experiment is performed. In a setting where different topics are used for training and testing our model performs significantly better than the baseline. t-tests were run for each experiment and differences between baseline and our model were observed for each cross-domain condition (with a 99% confidence level). This could be an indication that our model is more able to capture ironic style disregarding domain.

Analysing the data on Figure 2, we observe that 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 (this latter statement can be appreciated in Figure 1 as well). Note, however, that 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 imbalance may cause issues in a not controlled environment (e.g. no preselected topics, only random generic tweets). In spite of this, information gain values are fairly high with four features having information gain values over 0.1. Finding features that are significant for any non-ironic topic is hard, this is why our system includes several feature sets: they aim to distinguish irony from as many different topics as possible.

	Training Set								
	Education			**Humour**			**Politics**		
Test set	P	R	F1	P	R	F1	P	R	F1
Education	.78/.73	.78/.73	**.78**/.73	.65/.61	.63/.61	.62/.61	.71/.67	.71/.67	.70/.67
Humour	.64/.62	.61/.62	.60/.62	.80/.75	.80/.75	**.80**/.75	.64/.61	.62/.61	.60/.60
Politics	.71/.67	.70/.67	.69/.67	.63/.61	.51/.60	.59/.60	.79/.75	.79/.75	**.79**/.75

Table 2: Precision, Recall and F-Measure for each topic combination of our model when Decision Tree and Random Forest are used. Data are in the format "Random Forest / Decision Tree". We marked in **bold** the F-Measures that are better.

	Education			**Humour**			**Politics**		
Model	P	R	F1	P	R	F1	P	R	F1
Reyes et. al	.76	.66	.70	.78	.74	**.76**	.75	.71	.73
Our model	.73	.73	**.73**	.75	.75	.75	.75	.75	**.75**

Table 3: Precision, Recall, and F-Measure over the three corpora Education, Humour, and Politics. Both our and Reyes et al. results are shown; the classifier used is Decision Tree for both models. We marked in **bold** the F-Measures that are better compared to the other model.

With respect to results for two different classifiers trained with our model (Random Forest (RF) and Decision Trees (DT)) we observe that (see Table 2) RF is better in cross-validation but across-domains both algorithms are comparable.

Turning now to the state of the art we compare our approach to (Reyes et al., 2013), the numbers presented in Table 3 seem to indicate that (i) our approach is more balanced in terms of precision and recall and that (ii) our approach performs slightly better in terms of F-Measure in two out of three domains.

7 Conclusion and Future Work

In this article we have proposed a novel linguistically motivated set of features to detect irony in the social network Twitter. The features take into account frequency, written/spoken differences, sentiments, ambiguity, intensity, synonymy and structure. We have designed many of them to be able to model "unexpectedness", a key characteristic of irony.

We have performed controlled experiments with an available corpus of ironic and non-ironic tweets using classifiers trained with bag-of-words features and with our irony specific features. We have shown that our model performs better than a bag-of-words approach across-domains. We have also shown that our model achieves state-of-the-art performance.

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. Finally, we have relied on very basic tools for linguistic analysis of the tweets, so in the near future we intend to incorporate better linguistic processors. A final aspect we want to investigate is the use of n-grams from huge collections to model "unexpected" word usage.

Acknowledgments

We are greatful 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.

References

Kalina Bontcheva, Leon Derczynski, Adam Funk, Mark A. Greenwood, Diana Maynard, and Niraj Aswani. 2013. TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. In *Proceedings of Recent Advances in Natural Language Processing Conferemce*.

Paula Carvalho, Luís Sarmento, Mário J Silva, and

Eugénio de Oliveira. 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.

Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 519–528, New York, NY, USA. ACM.

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

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*.

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

Elena Filatova. 2012. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Proceedings of Language Resources and Evaluation Conference*, pages 392–398.

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

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: A closer look. In *ACL (Short Papers)*, pages 581–586. Citeseer.

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

Daniel C Howe. 2009. Rita wordnet. java based api to access wordnet.

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

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

Joan Lucariello. 1994. Situational irony: A concept of events gone awry. *Journal of Experimental Psychology: General*, 123(2):129.

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

Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

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

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

Brett Spell. 2009. Java api for wordnet searching (jaws).

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

Tony Veale and Yanfen Hao. 2010a. Detecting ironic intent in creative comparisons. In *ECAI*, volume 215, pages 765–770.

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

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

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

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