Modelling Sarcasm in Twitter, a Novel Approach

Francesco Barbieri and **Horacio Saggion** and **Francesco Ronzano**
Pompeu Fabra University, Barcelona, Spain
`<firstName>.<lastName>@upf.edu`

Abstract

Automatic detection of figurative language is a challenging task in computational linguistics. Recognising both literal and figurative meaning is not trivial for a machine and in some cases it is hard even for humans. For this reason novel and accurate systems able to recognise figurative languages are necessary. We present in this paper a novel computational model capable to detect sarcasm in the social network Twitter (a popular microblogging service which allows users to post short messages). Our model is easy to implement and, unlike previous systems, it does not include patterns of words as features. Our seven sets of lexical features aim to detect sarcasm by its inner structure (for example unexpectedness, intensity of the terms or imbalance between registers), abstracting from the use of specific terms.

1 Introduction

Sarcasm is a mode of communication where literal and intended meanings are in opposition. Sarcasm is often used to express a negative message using positive words. Automatic detection of sarcasm is then very important in the sentiment analysis field, as a sarcastic phrase that includes positive words conveys a negative message and can be easily misunderstood by an automatic system.

A number of systems with the objective of detecting sarcasm have been designed in the past years (Davidov et al., 2010; González-Ibáñez et al., 2011; Riloff et al., 2013). All these computational models have in common the use of frequent and typical sarcastic expressions as features. This is of course a good approach as some words are used sarcastically more often than others.

Our research seeks to avoid the use of words as features, for two reasons. Firstly, we want to reduce the complexity of the computational model, decreasing drastically the number of features required for classification. Secondly, typical sarcastic expressions are often culturally specific (an expression that is considered sarcastic in British English is not necessary sarcastic in American English and vice-versa). For these reasons we have designed a system that aims to detect sarcasm without the use of words and patterns of words. We use simple features such as punctuation (Carvalho et al., 2009) and more sophisticated features, that for example detect imbalance between registers (the use of an "out of context" word may suggest sarcastic intentions) or the use of very intense terms.

We study sarcasm detection in the microblogging platform 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. The dataset we adopted contains positive examples tagged as sarcastic by the users (using the hashtag #sarcasm) and negative examples (tagged with a different hashtag). This methodology has been previously used in similar studies (Reyes et al., 2013; Lukin and Walker, 2013; Liebrecht et al., 2013).

We presented in Barbieri and Saggion (2014) a model capable of detecting irony, in this paper we add important features to this model and evaluate a new corpus to determine if our system is capable of detecting tweets marked as sarcastic (#sarcasm). The contributions of this paper are the following:

- Novel set of features to improve the performances of our model
- A new set of experiments to test our model's ability to detect sarcasm
- A corpus to study sarcasm in twitter

[1] `https://twitter.com/`

Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 50–58,
Baltimore, Maryland, USA. June 27, 2014. ©2014 Association for Computational Linguistics

We will show in the paper that results are positive and the system recognises sarcasm with good accuracy in comparison with the state-of-the-art. The rest of the paper is organised as follows: in the next Section we describe related work. In Section 3 we describes the corpus and text processing tools used and in Section 4 we present our approach to tackle the sarcasm 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

A standard definition for sarcasm seems not to exist. Sarcasm is often identified as irony or verbal irony (**?**). Irony has been defined in several ways over the years as for example "saying the opposite of what you mean" (Quintilien and Butler, 1953), or by Grice (1975) as a rhetorical figure that violates the maxim of quality: "Do not say what you believe to be false", or as any form of negation with no negation markers (Giora, 1995). Other definitions are the ones of Wilson and Sperber (2002) who states irony is an echoic utterance that shows a negative aspect of someone's else opinion, and 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".

Irony and sarcasm has been approached as computation problem recently by Carvalho et al. (2009) who 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) and also Barbieri and Saggion (2014) have recently proposed two approaches to detect irony in Twitter. There are also some computational model to detect sarcasm in Twitter. 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. Finally Riloff (2013) built a model to detect sarcasm with a bootstrapping algorithm that automatically learn lists of positive sentiments phrases and negative situation phrases from sarcastic tweet, in order to detect the characteristic of sarcasm of being a contrast between positive sentiment and negative situation.

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).

Some datasets exist for the study of sarcasm and irony. 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

We adopted a corpus of 60,000 tweets equally divided into six different topics: *Sarcasm*, *Education*, *Humour*, *Irony*, *Politics* and *Newspaper*. The Newspaper set includes 10,000 tweets from three popular newspapers (New York Times, The Economist and The Guardian). The rest of the tweets (50,000) were automatically selected by looking at Twitter hashtags #education, #humour, #irony, #politics and #sarcasm) added by users in order to link their contribution to a particular subject and community. These 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" (and sarcasm in the case of the hashtag #sarcasm). *Education*, *Humour* and *Politics* tweets were prepared by Reyes et al. (2013), we added *Irony*, *Newspaper* and *Sarcasm* tweets[2]. We obtained these data using the Twitter API.

Examples of tweets tagged with #sarcasm are:

- This script is superb, honestly.
- First run in almost two months. I think I did really well.
- Jeez I just love when I'm trying to eat lunch and someone's blowing smoke in my face. Yum. I love ingesting cigarette smoke.

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[3] (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, 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 sarcasm 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 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 several). 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 sarcastic tweets (like type of punctuation, length, emoticons), and some others to recognise sentiments and intensity of the terms used. 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)*

To the best of 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 the following sections we quickly describe all the features we used.

4.1 Frequency

Unexpectedness can be a signal of verbal 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

[2] To make possible comparisons with our system we published the IDs of these tweets at http://sempub.taln.upf.edu/tw/wassa2014/

[3] 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)

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.

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.).

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*.

Inspired by Davidov et al. (2010) and Carvalho (2009) we designed features related to punctuation. These features are: number of **commas**, **full stops**, **ellipsis**, **exclamation** and **quotation** marks that a tweet contain.

The **emoticon** feature is the sum of the emoticons *:)*, *:D*, *:(* and *;)* in a tweet.

The new features we included are *http* that simply says if a tweet includes or not an Internet link, and the entities features provided by TwitIE (Bontcheva et al., 2013). These features check if a tweet contains the following entities: *n. organisation*, *n. location*, *n. person*, *n. first person*, *n. title*, *n job title*, *n. date*. These last seven features were not available in the previous model, and some of them work very well when distinguishing sarcasm from newspaper tweets.

4.4 Intensity

In order to produce a sarcastic 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)). 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 (Riloff et al., 2013). 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

As previously said, sarcasm convey two messages to the audience at the same time. 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 Sarcasm corpus are higher than in the other topics, 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 sarcasm is ambiguity. We noticed that sarcastic 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 also evaluate the sentiment of the sarcastic tweets. 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 sarcasm. We explore ironic sentiments with two different views: the first one is the simple analysis of sentiments (to

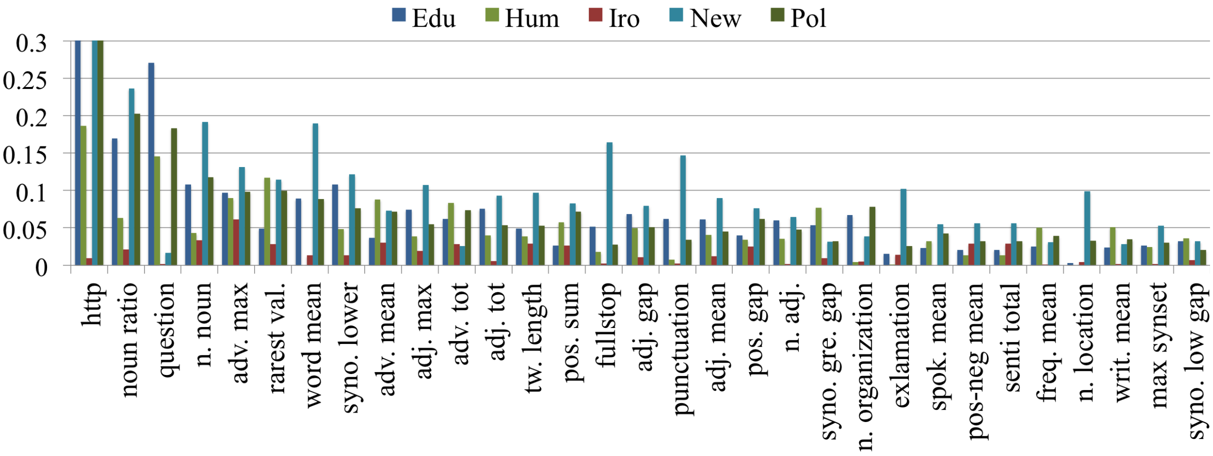

Figure 1: Information gain of each feature of the model. Sarcasm is compared to Education, Humor, Irony, Newspaper and Politics. High values of information gain help to better discriminate sarcastic from non-sarcastic tweets.

identify the main sentiment of a tweet) and the second one concerns sentiment imbalances between words.

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.

5 Experiments and Results

In order to evaluate our system we use five datasets, subsets of the corpus in Section 3: Sarcasm vs Education, Sarcasm vs Humour, Sarcasm vs Irony, Sarcasm vs Newspaper and Sarcasm vs Politics. Each combination is balanced with 10.000 sarcastic and 10.000 of non-sarcastic examples. We run the following two types of experiments:

1. We run in each datasets a 10-fold cross-validation classification experiment.

2. We train the classifier on 75% of positive examples and 75% of negative examples of the same dataset, then we use as test set the rest 25% positive and 25% negative. We perform this experiment for the five datasets.

In Figure 1 and Figure 2 we show the values of information gain of the five combinations of topics (Sarcasm versus each not-sarcastic topic). Note that, in the first figure the scale we chose to better visualise all the features truncates the scores of the feature **http** of Education, Newspaper, and Politics. These three values are respectively 0.4, 0.7 and 0.4. Table 1 and Table 2 includes Precision, Recall, and F-Measure results of Experiment 1 and Experiment 2.

6 Discussion

The best results are obtained when our model has to distinguish Sarcasm from Newspaper tweets. This was expected as the task was simpler than the others. In Newspaper tweets nine out of ten times present an internet link, and this aspect can be used to well distinguish sarcasm as internet links are not used often. Moreover the Newspaper tweets use a formal language easily distinguishable from sarcasm. In Newspaper tweets there are more nouns (average ratio of 0.5) than in sarcastic tweets (ratio

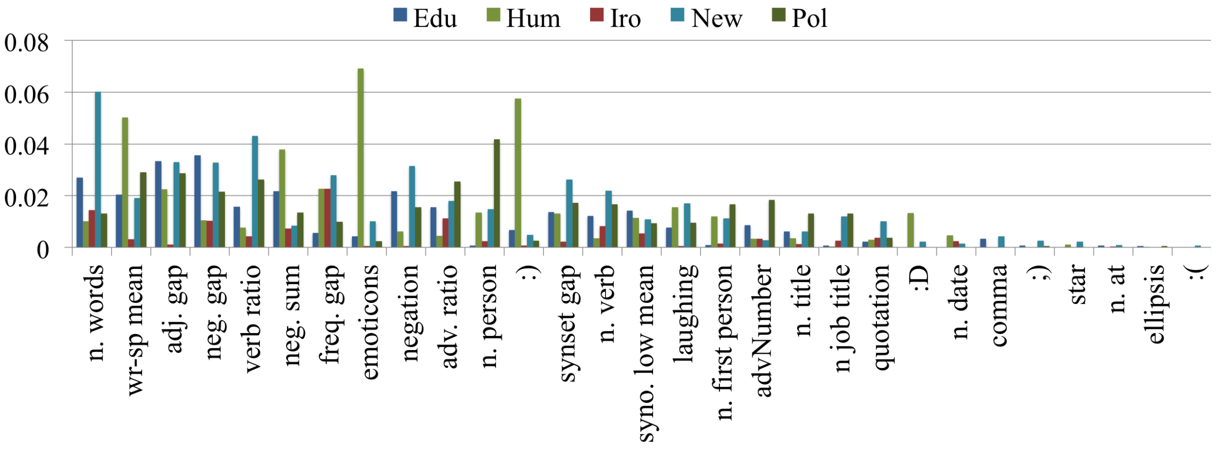

Figure 2: Information gain of each feature of the model. Sarcasm is compared to Education, Humor, Irony, Newspaper and Politics. High values of information gain help to better discriminate sarcastic from non-sarcastic tweets.

	Prec.	**Recall**	**F1**
Education	.87	.90	.88
Humour	.88	.87	.88
Irony	.62	.62	.62
Newspaper	.98	.96	.97
Politics	.90	.90	.90

Table 1: Precision, Recall and F-Measure of each topic combination for Experiment 1 (10 cross validation). Sarcasm corpus is compared to Education, Humour, Irony, Newspaper, and Politics corpora. The classifier used is Decision Tree

0.3), and Newspaper uses less punctuation marks than sarcasm. Overall Newspaper results are very good, the F1 is over 0.95.

Education and Politics results are very good as well, F1 of 0.90 and 0.92. Also in these topics the internet link is a good feature. Other powerful features in these two topics are **noun ratio** (as Newspaper they present more number of nouns than sarcasm), **question**, **rarest val.** (sarcasm includes less frequently used words) and **syno lower**.

Results regarding sarcasm versus Humour are positive, F-Measure is above 0.87. The most marked differences between Humour and sarcasm are the following. Humour includes more links (**http**), more question marks are used to mark jokes like: "Do you know the difference between...?", "What is an elephant doing...?" (**question**), sarcasm includes rarer terms and more intense adverbs than Humour (**rarest val.**, **adv. max**).

Our model struggles to detect tweets marked as sarcastic from the ones marked as ironic. Even if not very powerful, relevant features to detect sarcasm against irony are two: use of adverbs (sarcasm uses less but more intense adverbs) and sentiment scores (as expected sarcastic tweets are denoted by more positive sentiments than irony). Poor results in this topic indicate that irony and sarcasm have similar structures in our model, and that new features are necessary to distinguish them.

	Prec.	**Recall**	**F1**
Education	.87	.88	.87
Humour	.87	.86	.86
Irony	.60	.61	.60
Newspaper	.95	.96	.95
Politics	.89	.89	.89

Table 2: Precision, Recall and F-Measure of each topic combination for Experiment 2 (Test set). Sarcasm corpus is compared to Education, Humour, Irony, Newspaper, and Politics corpora.The classifier used is Decision Tree

The comparison with other similar systems is not easy. We obtain better results than Reyes et al. (2013) and than Barbieri and Saggion (2014), but the positive class in their experiments is irony. The system of Davidov et al. (2010) to detect sarcasm seems to be powerful as well, and their results can compete with ours, but in the mentioned study there is no negative topic distinction, the not-sarcastic topic is not a fixed domain (and our controlled experiments results show that depending on the negative example the task can be more or less difficult).

7 Conclusion and Future Work

In this study we evaluate our system to detect sarcasm in the social network Twitter. We tackle this problem as binary classification, where the negative topics are Education, Humour, Irony, Newspaper and Politics. The originality of our system is avoiding the use of pattern of words as feature to detect sarcasm. In spite of the good results, there is much space for improvement. We can still enhance our results by including additional features such as language models. We will also run new experiments with different negative topics and different kind of text, for example on Amazon reviews as Davidov et al. (2010). Finally, a very interesting but challenging issue will be distinguishing with better accuracy sarcasm from irony.

Acknowledgments

We are grateful to two 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 Innovación, Spain. We also acknowledge partial support from the EU project Dr. Inventor (FP7-ICT-2013.8.1 project number 611383).

References

Francesco Barbieri and Horacio Saggion. 2014. Modelling Irony in Twitter. In *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. Association for Computational Linguistics.

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 the International Conference on Recent Advances in Natural Language Processing*. Association for Computational Linguistics.

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

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.

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.

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.

Stephanie Lukin and Marilyn Walker. 2013. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. *NAACL 2013*, page 30.

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

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.

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.

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

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 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.

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

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.