Engineering Applications of Artificial Intelligence 51 (2016) 50–57

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A comparison between semi-supervised and supervised text mining techniques on detecting irony in greek political tweets

Basilis Charalampakis, Dimitris Spathis *, Elias Kouslis, Katia Kermanidis

Department of Informatics, Ionian University, Corfu, Greece

ARTICLE INFO

Available online 2 March 2016

Keywords:
Irony detection
Text mining
Twitter
Politics

ABSTRACT

The present work describes a classification schema for irony detection in Greek political tweets. Our hypothesis states that humorous political tweets could predict actual election results. The irony detection concept is based on subjective perceptions, so only relying on human-annotator driven labor might not be the best route. The proposed approach relies on limited labeled training data, thus a semi-supervised approach is followed, where collective-learning algorithms take both labeled and unlabeled data into consideration. We compare the semi-supervised results with the supervised ones from a previous research of ours. The hypothesis is evaluated via a correlation study between the irony that a party receives on Twitter, its respective actual election results during the Greek parliamentary elections of May 2012, and the difference between these results and the ones of the preceding elections of 2009.

1. Introduction

Irony as a para-linguistic element is used to figuratively express a concept with a semantic meaning that is very different from its actual initial purpose. It is a challenging field for computational linguistics and natural language processing due to the high ambiguity and the difficulty to detect it, objectively. Language use is vigorous and creative; there is no pre-defined consensual agreement on how to recognize an ironic expression, due to the high subjectivity involved.

In the last decade, irony expression has been thriving on social networks and particularly Twitter, because of the 140 characters restraint on the status updates, being a perfect fit for good old one-liners. As a public social medium, users realize that their writings may be read and reproduced by potentially everyone, gaining popularity and followers. But this publicity, contrary to Facebook's real name policy, often has no direct consequences to their everyday lives, since the majority participate anonymously, using an avatar and a nickname.

This no-censorship state contributes to the freedom of expressing personal thoughts on tough, taboo, unpopular or controversial issues, part of which contains the political satire. Political satire is a significant part of comedy which specializes in drawing entertainment from politics. Most of the times, it aims just to please. By nature, it does not offer a constructive view by itself; when it is used as part of criticism, it tends to simply pinpoint the unexpected or different.

The high topicality of Twitter, combined with the ephemerality of political news, forms a state which is described as'echo chamber', a group-thinking effect on virtually enclosed spaces, amplified by repetition (Colleoni et al., 2014). As a result, the occasional user might write something political just to 'jump on the bandwagon', without an initial conscious aim to criticize. Adding to that, politics is a topic that almost everybody is familiar with and makes more sense from the engagement and attention side to write about Obama instead of an obscure book you just read.

Studies focus on the simultaneous usage of Twitter and the TV on circumstances like a political debate, where meta-talk tweets reveal critical scrutiny of the agenda or 'the debate about the debate' (Kalsnes et al., 2014).

In the rapidly changing web, there is a plethora of available text, especially from social networks, which is unlabeled, raw or unprocessed. Adding to the traditional supervised methods, there are quite a few techniques that enable us to take these huge unstructured data into account. An insight from our previous work was the subjectivity involved during the tagging of a text as ironic. Three of our authors who took up the tedious task of annotation could not agree on what should be considered as ironic or not. As a result, there cannot be a *gold standard corpus* of ironic tweets. This was our main motivation to explore semi-supervised techniques, since they take

* Corresponding author. Tel.: +30 6946354612.
(D. Spathis).
E-mail addresses: p11char@ionio.gr (B. Charalampakis), p11spat@ionio.gr, sdimitris@csd.auth.gr (D. Spathis), p11kous@ionio.gr (E. Kouslis), kerman@ionio.gr (K. Kermanidis).

http://dx.doi.org/10.1016/j.engappai.2016.01.007
0952-1976/

into account both train and test data. To be specific, the technique we chose is collective classification: a type of semi-supervised learning that presents an interesting method for optimizing the classification of partially-labeled data.

Considering the above, our empirical study tries to detect irony on a corpus of Greek political tweets by training a classifier, using appropriate linguistic features, some of which are proposed for the first time herein for irony detection. Our goal is to find a relation between the ironic tweets that refer to the political parties and leaders in Greece in the pre-election period of May 2012, and their actual election results. We compare the semi-supervised results with the supervised ones from a previous research of ours. Regarding the novelty of our study, this is a first exploration on the field of irony detection with semi-supervised learning and an application in politics.

The remainder of this paper is organized as follows: In Section 2, we present the related literature on the topics of irony detection, Twitter sentiment analysis and political expression. The next Sections 3 and 4 are dedicated to data preprocessing and its representation schema through the set of linguistic features that affect irony detection. The Section 5 describes the training procedure, the evaluation of the algorithms' performance and their test procedure on a large unlabeled dataset. An overview of the study limitations, future research prospects and a summary of the empirical study are described in Section 6.

2. Related work

The greater part of the literature on irony detection in computational linguistics is focused on English, but this is a first attempt to explore this area in the Greek language, to the authors' knowledge.

Reyes et al. (2013) attempt to detect irony by examining the corpus on the following features: signatures (concerning pointedness, counter-factuality, and temporal compression), unexpectedness (concerning temporal imbalance and contextual imbalance), style (as captured by character-grams (c-grams), skip-grams (s-grams), and polarity skip-grams (ps-grams)) and emotional scenarios (concerning activation, imagery, and pleasantness). These features work better when they are used as part of a coherent framework rather than used individually. They used multiple datasets in order to evaluate their hypothesis and achieved a precision of 0.79 at best. Classification is performed by Naïve Bayes and Decision Trees. Also a crisis management case study of the hashtag #Toyota is described.

A study by Rajadesingan and Liu (2014) discovered an interesting aspect of Twitter usage, an 'orientation phase' in which the user is gradually introduced to irony as one gains followers. The threshold of this phase is one's 30 initial tweets. The top features in decreasing order of importance for sarcasm detection are the following: Percentage of emoticons in the tweet, percentage of adjectives in the tweet, percentage of past words with sentiment score, number of polysyllables per word in the tweet, lexical density of the tweet. They evaluate using a J48 decision tree, logistic regression, and SVM to obtain an accuracy of 78.06%, 83.46%, and 83.05%, respectively.

The usual approach on similar irony detection studies on Twitter is to identify the two classes by hashtag analysis. However, this method creates noisy results with low accuracy (González-Ibánez et al., 2011; Liebrecht et al., 2013). Features used by Gonzalez were Lexical (unigrams, affective language, interjections and punctuation) and Pragmatic (positive smileys, negative smileys, and "@toUser" signs if a twitter is directed to another user). Algorithms used were SVM with SMO and Logistic Regression. Overall SMO outperformed LogR, with the best accuracy of 57% being an indication of the difficulty of the task. On the other hand, Liebrecht approached the same problem with the *Balanced Winnow* algorithm for classification. The strongest linguistic markers of sarcastic utterances were markers that can be seen as synonyms for #sarcasm hashtag. Testing the classifier on the top 250 of the tweets it ranked as most likely to be sarcastic, it attains a 30% average precision.

Twitter lexical analysis on Greek tweets has been the main subject of the research by Kermanidis and Maragoudakis (2013), examining the sentimental tagging in a supervised environment. Their hypothesis is focused on the positive / negative distinction, using statistical metrics such as count and frequency distributions. The alignment between actual political results and web sentiment in both directions was investigated and confirmed that there is a relation between political results and web sentiment. We use the same corpus of tweets in our study.

Apart from Twitter, similar techniques have been applied on Amazon reviews as well, making use of structured information of reviews versus the unstructured nature of Twitter. The accuracy results are encouraging due to the *semi-supervised* technique and the huge dataset, requiring human-annotator labor though, Davidov et al. (2010); Tsur et al. (2010). Features used were high-frequency words, content words, sentence length and punctuation. Results on the Twitter dataset are better than those obtained on the Amazon dataset, with accuracy of 0.947 with a k-nearest neighbors implementation.

Semi-supervised techniques on text mining were applied by Fangzhong and Markert (2009). Their approach involves Wordnet, like us, and they propose a subjectivity measure of each Wordnet entry. They suggest a semi-supervised minimum-cut framework that makes use of both WordNet definitions and its relation structure. Minimum-cut is a technique used at graph theory, which uses pairwise relationships between the data points in order to learn from both labeled and unlabeled data. The semi-supervised approach achieves the same results as the supervised framework with less than 20% of the training data.

In the emerging area of *active learning*, where the learning algorithm is able to interactively query the researcher to obtain the desired outputs at new data points, there is some ongoing research. Gokhan et al. (2005) wanted to reduce the labeling effort for spoken language understanding from data gathered at AT&T call centers. The examples that are classified with higher confidence scores (not selected by active learning) are exploited using two semi-supervised learning methods. This enables them to exploit all collected data and alleviates the data imbalance problem caused by employing only active or semi-supervised learning. Their results indicate that it is possible to reduce human labeling effort significantly. Similar technique, namely *collective learning*, was followed by Santos et al. (2011), where they propose a new method that adopts a collective learning approach to detect unknown malware. Their empirical research demonstrates that the labeling efforts are lower than when supervised learning is used, while maintaining high accuracy rates. Collective classification is an approach that uses the relational structure of the combined labeled and unlabeled dataset to enhance classification accuracy (Neville and Jensen, 2003).

Research by de-la-Peña-Sordo et al. (2013) studied the comparison between collective learning and supervised techniques, pretty similar with our methodology. Apart from that, quite similar was their topic of detecting trolling comments on a Spanish platform like Digg or Reddit and their lexical features selection, since irony and trolling may seem indistinguishable in some cases. Their approach obtains nearly the same accuracy than the best supervised learning approaches.

Another study dealing with online opinion and reviews, again by Reyes and Rosso (2011), examined Amazon and Slashdot.com

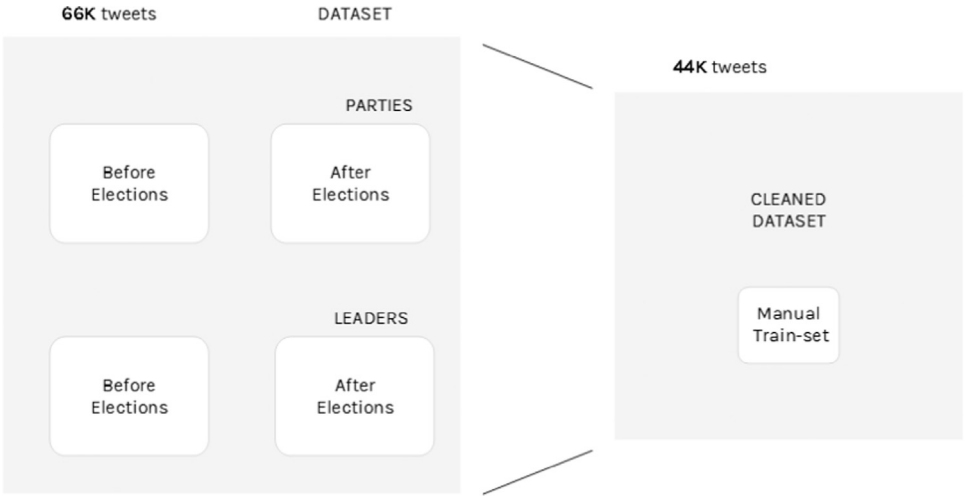

Fig. 1. Structure and data preprocessing of the initial dataset and the cleaned one after preprocessing.

customer reviews trained on Naive Bayes, Decision Trees and Support Vector Machines. Accuracy results were satisfying and feature selection ranked as top the features of *POS 3-gram* (frequent sequence of trigrams) and *Pleasantness* (dictionary approach to pleasant and unpleasant words).

Considering the above attempts in the field of computational linguistics, the novelty of our study lies on the leverage of these above techniques in Greek, based on the hypothesis that sarcasm and irony in Twitter messages may be linked to actual elections results. We conduct a comparison between traditional supervised and semi-supervised learning. Regarding the tools we developed, we used NLTK, Weka and Wordnet, but the feature extraction / text mining was done in Python code developed by us, and the methodology is described in the following sections (Fig. 1).

3. Data preprocessing

The dataset contains 61.427 Greek tweets collected on the week before and the week after the May 2012 parliamentary elections in Greece (Figs. 1 and 2). The dataset is divided in 2 sub-datasets: parties and leaders. For each one, there are two sub-sections: before and after the elections. The dataset structure before the clean-up is presented below:

The dataset is available for research purposes[1] (Table 1 and 2).

The first step was to eliminate the duplicate tweets in order to avoid frequency bias. Duplicate tweets are identical tweets and "retweets" (RTs) that do not contribute to our linguistic goals.

The second step was to delete the useless, unstructured artifacts (unformatted tweets) that were fetched by the Twitter API. In order to form a unibody test set, we merged the above sub-datasets. We decided to keep the tweets that contain links, because our hypothesis supports that tweets with links, for instance newspaper article tweets, are neutral, not ironic. After the cleanup, the size of tweets was 44.438.

The semantic analysis was assigned to Balkanet (Tufis et al., 2004), a Greek edition of the WordNet (OMW: Open Multilingual Wordnet), a popular lexical database that groups words into sets of cognitive synonyms (synsets), each expressing a distinct concept.

Also, the Python natural language package, NLTK (Bird et al., 2009), was used in order to support Wordnet. The machine learning and training process was performed using the Weka software (Fig. 2).[2]

4. Features

We approached irony detection as a text classification problem. The decision if a tweet is ironic-or-not is a binary decision. We tag each tweet with five features, taking into consideration structural sentence formations and unexpectedness occurrences. Some of the features are designed to detect imbalance and unexpectedness, others to detect common patterns in the structure of the ironic tweets (like type of punctuation, length, and emoticons). (Barbieri and Saggion, 2014) Our features are grouped into the following model:

- Spoken (spoken style applied in writings)
- Rarity (the frequency occurrences of the most rare words)
- Meanings (the number of Wordnet synsets as a measure of ambiguity)
- Lexical (punctuation, prosodic repeated letters, metaphors)
- Emoticons (smiley faces etc)

We analyzed the features with the pearson correlation and found low correlation between the variables. To be precise, the highest correlations were between the dependent and the independent variables: rarityScore and isIronic (0.398), lexicalScore and isIronic (0.350). These relations are confirmed by the Feature Selection process as well (see 5.3). The only correlation between independent variables (features) was between rarityScore and emoticonScore (0.329) which is considered relatively low, on statistical terms.

4.1. Spoken

The verbal irony in Twitter is often expressed as everyday-life chats between potentially real characters, using heavily dashes (-) and asterisks (*). Their occurrences in tweets count positively in our classifier. The use of spoken language is often related to unexpectedness. In political context, dashes may be used to quote an actual quote, but the

[1] http://di.ionio.gr/hilab/doku.php?id=start:websent.

[2] http://www.cs.waikato.ac.nz/ml/weka.

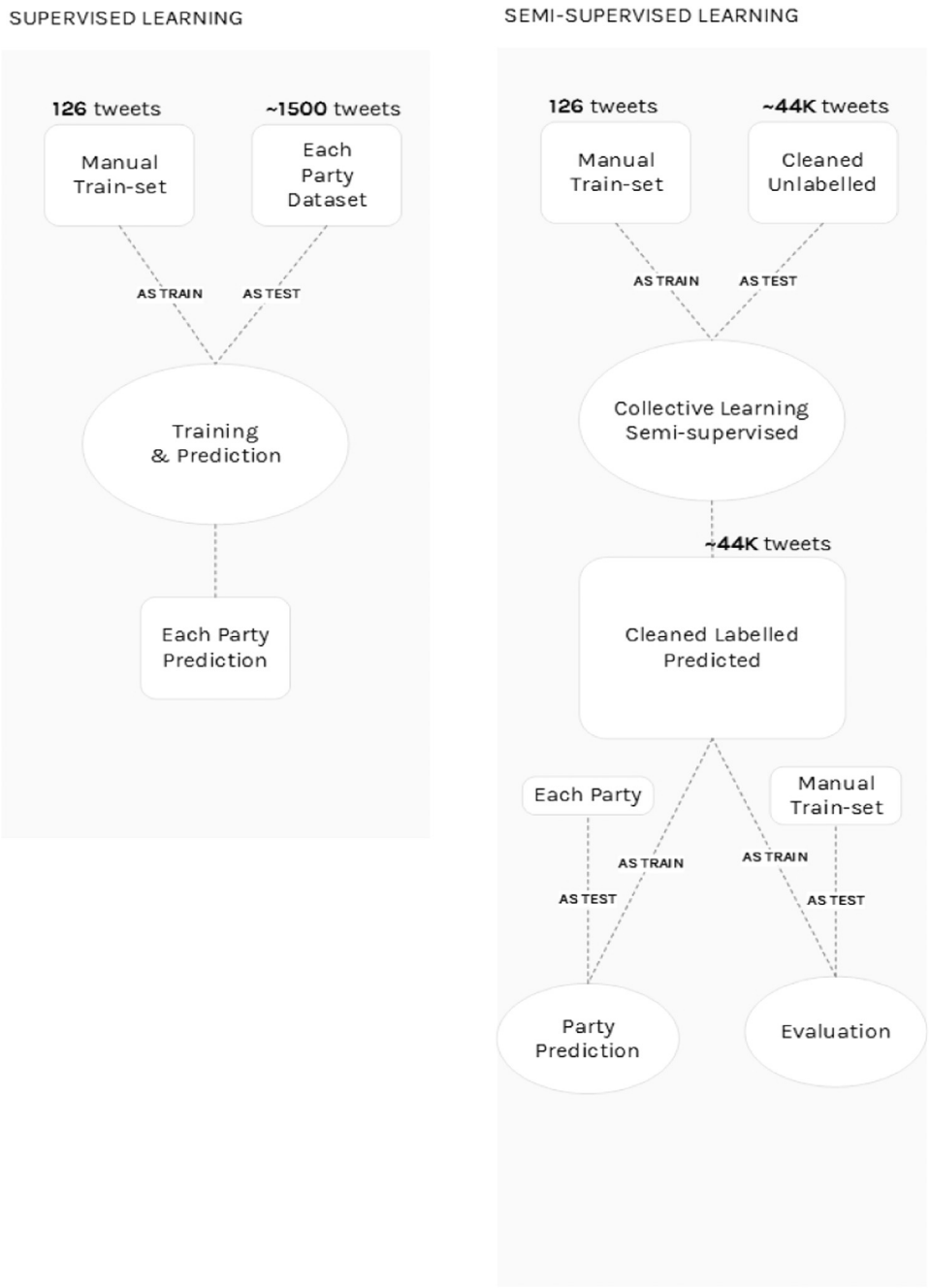

Fig. 2. Schematic flowchart of the workflow we followed, regarding the datasets, the training techniques and the operations.

Table 1
Number of tweets mentioning political parties.

Party	ANTARSYA	PASOK	DHMAR	KKE	ND	SYRIZA	XA	ANEL
Before elections	1615	2909	1373	1588	792	5159	2731	791
After elections	1628	9984	506	1594	1272	2294	3300	1109

reason is usually to add a sarcastic comment. The asterisk character showcases movements or non-verbal actions in tweets, such as *sigh* or *faints*, adding an emotional level. If there is at least one of the above characters in the tweet, the value of the feature is 'true', otherwise it is'false'. Thus, the spoken feature is binary. We grouped the attributes as one variable because the continuous score does not check the existence or not of spoken speech.

4.2. Rarity

A frequency dictionary for all the words of the original dataset was created. The tweets are split into tokens, and token occurrences (excluding URL links) are counted. Thus, we isolated the rarest words and limited the upper bound to three occurrences. The resulted frequency dictionary consists of 25.898 words. The

Table 2
Number of tweets mentioning party leaders.

Leader	Tsipras	Kammenos	Samaras	Michaloliakos
Before elections	3282	3511	4513	1059
After elections	2992	1721	3171	2530

rarest cases had 1 occurrence. If a word had 3 occurrences, it was less rare. In order to invert this scale, proper weights were attached to each token: weights 10, 5 and 2 were assigned to frequencies 1, 2 and 3 respectively. The distance of the first weight (10) and the second (5) shows the significance of the most rare words. The final score is the following equation:

$$\frac{\sum_{n=1}^{m} \text{word}_n(\text{weight})}{m}$$

The formula that estimates the rarity score of each word of the examined tweet, where: m=number of words (tokens) of each tweet and n=current word (token) under examination.

This formula looks up every word of the given tweet in our frequency dictionary. If the word is found there, it attaches a weight according to the scale we described above. For normalization purposes, we divide the weight scores with the number of words of each tweet. The three occurrences threshold is over the whole dataset. We qualitatively examined the dataset with 4+ occurrences and there were many frequent words inside that do not provide linguistic value, so we set the limit to three. The descriptive statistics of this variable distribution show high concentration around the 0.4 score with a maximum value of 10. Almost half of the dataset tweets have a score of zero.

4.3. Meanings

We used the Balkanet packet of Wordnet to extract the meanings of each word, because the use of a word with multiple meanings implies ambiguity and eventually irony. For instance, the one-liner 'Change is inevitable, except from a vending machine' exploits the ambiguity, and consequently wrong expectations, induced by the word change (Mihalcea and Strapparava, 2006). Our algorithm looks up in Balkanet every word of each tweet. If a word has multiple synsets (meanings), we count their number and add them to the score. This process is repeated for every tweet. The descriptive statistics of this variable distribution show high concentration between scores 0.2 and 3 with a maximum value of 85. Almost a third of the dataset tweets have a score of zero.

4.4. Lexical

The lexical attributes of each tweet were: repeated letters, metaphor words and punctuation.

Twitter's users use repeated letters to express a spoken-verbal emotion. This phenomenon is called prosody, altering the intonation of speech like singing (Cheang and Pell, 2008). Also, we track the occurrences of words that showcase figurative language. For the example, the word “like” in Greek is written as σαν, σα & σάν.

The punctuation feature is the aggregation of exclamation marks, question marks, dots and semicolons. The semicolon is used in Greek instead of the '?' symbol. The descriptive statistics of this variable distribution show high concentration of scores 1 and 2 with a maximum value of 5. Two thirds of the dataset tweets have a score of zero.

```
'Ο Καμμένος σαν λιοκαμένος μου φαίνεται'
```

```
Spoken Score    →    0
Rarity Score    →    1.67
Meaning Score   →    0
Lexical Score   →    1
Emoticon Score  →    0
```

Fig. 3. Example tweet assigned with feature scores.

4.5. Emoticons

The Emoticon feature detects all the possible variations of smiley, sad and mocking faces such as :), :-(, :P etc. Irony can be detected by the existence of emoticons, due to emotional charge. The value of the Emoticon feature is binary: 'true' if at least one emoticon appears in the tweet, 'false' otherwise.

The above example (Fig. 3) displays a random political Greek tweet about a party leader and translates roughly to “Kammenos looks like sunburnt”. This tweet exploits a wordplay with the name of the party leader, which in Greek sounds like the word “sunburnt”. As a result, the lexical score is above zero, since it uses figurative speech (“like”), as well as the rarity score, expressing a colloquial term of the word sunburnt in Greek.

5. Test and results

5.1. Supervised technique

5.1.1. Training

After the data preprocessing and the automatic feature scoring, we have a complete dataset, ready for labeling. We labeled a small amount of tweets manually (n=126) in order to train the classifier. The distribution of the dependent variable is: 74 ironic and 52 non-ironic tweets, gathered by randomly sampling the big dataset. The resulting set was loaded on Weka and was trained on multiple algorithms according to the 10-fold-cross validation technique. Apart from probabilistic algorithms, we involved decision trees as well, in order to be able to rank the significance of the features. The training algorithms with the best performance were: J48-the Weka version of C4.5 (Quinlan, 1993), SVM (Chang and Lin, 2011), Neural Networks, Naive Bayes (John and Langley, 1995), Functional Trees, KStar (Cleary and Trigg, 1995) and Random Forests (Breiman, 2001). The best performing algorithm on average was the Functional Trees (Precision=82.4). Functional Trees combine a univariate decision tree with a linear function by means of constructive induction. Decision trees created from the model are able to use decision nodes with multivariate tests, and leaf nodes that make predictions using linear functions (Gama, 2004).

5.1.2. Testing

The classification model created from the Functional Trees algorithm was applied to the unlabeled datasets of each party in order to get the irony predictions. Due to the fact that our test dataset is unlabeled, we can't evaluate the model's validity directly. An indirect, qualitative evaluation is attempted in the following sections, comparing the volume of irony in tweets with the actual election results.

5.2. Semi-supervised technique

5.2.1. Training-collective classification

Collective classification is a combinatorial optimization problem, in which we are given a set of documents, or nodes, D={d1,

…,dn} and a neighborhood function N, where $Ni \subseteq D/\{Di\}$, which describes the underlying network structure. Being D a random collection of documents, it is divided into two sets X and Y where X corresponds to the documents for which we know the correct values and Y are the documents whose values need to be found (Santos et al., 2011; Namata et al. 2009)

We applied the *collective-tree* algorithm, which is similar to the Random Tree classifier, but takes into account both labeled and unlabeled data. This algorithm combines the training and prediction phase, so it receives as training the manual small dataset and the big unlabeled one as testing. According to the documentation[3], the collective-tree algorithm splits the attribute at that position that divides the current subset of instances (of training and test instances) into (roughly) two halves. The tree is stopped from growing, if one of the following conditions is met:

a. only training instances would be covered (the labels for these instances are already known!)
b. only test instances in the leaf taking the distribution from the parent node
c. only training instances of one class all test instances are considered to have this class

5.2.2. Testing

The resulting predicted dataset is used as *gold corpus* training dataset against each unlabeled party dataset.

5.2.3. Validation

In order to evaluate the semi-supervised predictions, we used the resulting predicted dataset as train-set against the manual small dataset, enabling us to compare the performance of supervised vs semi-supervised techniques. The same algorithms as above were used (see Fig. 4). The best performing algorithm is Random Forest (precision=83.1), while Naive Bayes once again behaves the worst.

5.3. Hypothesis evaluation

In this section, we count the ironic and non-ironic tweets that were picked by our supervised and semi-supervised classifiers. Interestingly, the actual election results are not directly correlated but there is a trend between the parties that receive irony and their election votes' percentage fluctuation retrospectively. Table 3 shows that precision in both cases is quite similar so that the positive predicted outcome matches the manual positive tagging. On the other hand, recall is quite lower on the semi-supervised, meaning that the false negative rate is higher. Namely, semi-supervised classifies more frequently as non-ironic what human tagged as ironic. Probabilistic and instance-based methods perform worse with more data, while decision trees in some cases outperform the traditional models.

We cannot claim that one algorithm is better than another, but that it performs better on the given data. As we mention in Section 5.1, we include Decision Trees (J48, FT, Random Forest), Probabilistic (Naive Bayes), Instance-based (K-Star), Kernel-based (SVM) and Neural Networks. From theory, we know the considerations when choosing a ML algorithm are: *accuracy*, *training time*, *linearity*, *number of parameters* and *number of features*. Accuracy is covered thoroughly in our results tables and figures. In training time, our metrics show that naive bayes is the fastest while NNs expectedly require more training time. Linearity is a characteristic available only in SVMs and Neural Networks in our case. Qualitatively, our feature scatterplots do not present linearity, that is why those algorithms do not perform the best. Also, each algorithm

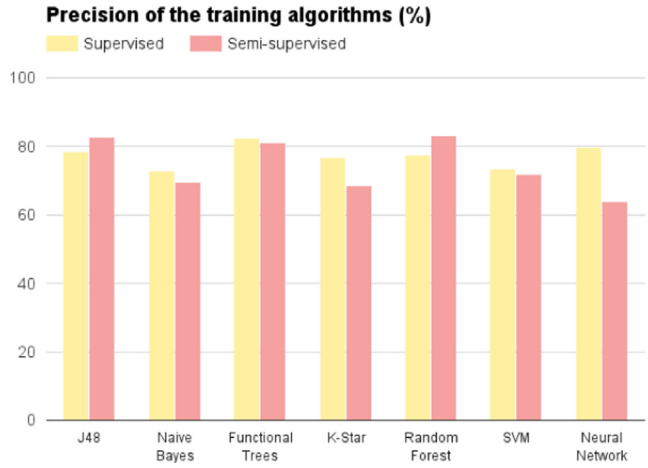

Fig. 4. Performance of the training algorithms, supervised against semi-supervised techniques. The semi-supervised precision is evaluated indirectly by using the predicted dataset as train-set against the human-annotated manual small dataset.

needs different fine-tuning regarding its parameters, with NNs being the most complex ones (many parameters) and probabilistic ones being more simple. Finally, the number of features remains the same (five *Scores*) in each experiment.

Another observation, the semi-supervised ironic tweets are significantly fewer, but their relative distance to the supervised ones is pretty much the same. We guess that the unlabeled dataset contributed to that, so that the smaller percent of ironic tweets of the semi-technique might be more representative and closer to the reality. Reasonably, the supervised ironic results seem a little too high; for instance, 70% of one party's tweets cannot be humorous.

Table 4 consists of the sub-dataset of 'parties before elections'. One should note that the irony percent shows a trend that may be interpreted as the *hype* for every party. The 'Election results' column refers to the actual results of the May 2012 elections. The fluctuation (last column) describes the difference between the May 2012 election results and the previous of 2009. The fluctuation percent shows *a trend on the edges*, so that the 'loser' parties of ND and PASOK are getting the most ironic tweets as well as the'winner' parties SYRIZA and XA. Both machine-learning techniques predict roughly the same result.

5.4. Feature selection

The features significance rank, ordered by decreasing significance, based on Information Gain, is the following: 1) Rarity (0.1732), 2) Lexical (0.0841), Emoticon (0.0451), Meanings (0) and Spoken (0). As mentioned in Section 4, we do not have adequate scores for meanings and spoken, so Information Gain does not consider them to be significant in feature selection.

6. Discussion

Twitter in Greece is not as popular and established compared to other countries. Greek Twitter users are just 3.7% of the total population[4]. The average user is usually young, well-educated and liberal, something important for our political context. As a result, our findings are filtered through this demographic.

[3] https://github.com/fracpete/collective-classification-weka-package.

[4] statistic from social media analysis site trending.gr.

Table 3
Performance measures of the training algorithms. Green indicates the best performance, while red the worst.

Algorithms	Supervised			Semi supervised		
	Precision	Recall	*F*-measure	Precision	Recall	*F*-measure
J48	78.4	77.8	77.0	82.7	73.8	73.3
Naive bayes	72.8	71.4	71.6	69.5	59.5	57.8
Functional trees	82.4	80.2	79.1	81.2	73.0	72.6
K-Star	76.9	77.0	76.7	68.6	61.1	60.2
Random forest	77.4	77.0	76.3	83.1	74.6	74.2
SVM	73.6	73.8	73.5	71.7	65.9	65.6
Neural network	79.9	78.6	77.6	63.8	55.6	53.8

Table 4
Ironic tweets that received every party and their election results. The fluctuation describes the difference between the May 2012 election results and the previous.

Party	Ironic		Total	Ironic/total %		2012 Greek election results %	Fluctuation % of 2012 results from 2009 greek election results
	Supervised	Semi-supervised		Supervised	Semi-supervised		
ANTARSYA	511	175	1240	41	14	1.19	0.83
PASOK	1440	592	2240	64	26	13.18	−30.74
DHMAR	497	224	1223	41	18	6.11	New
KKE	504	218	1262	40	17	8.48	0.94
ND	252	104	560	45	18	18.80	−14.60
SYRIZA	2649	1054	4371	61	24	16.70	12.10
XA	1040	410	1821	57	22	6.90	6.68
ANEL	280	77	680	41	11	10.60	New

On the technical side, we did not use a stemmer or a lemmatizer, because our hypothesis is depending on rare words or wordplays which would be eliminated. Furthermore, the informal nature of the text would render the performance of such tools rather useless for a morphologically fluent language such as Greek. Another restraint for our study was the shortage of tested NLP tools for the Greek language. Some available tools are not well documented or not accessible. As a result, our study is focused mainly on self-developed tools for mining the features from the text, which we are going to open source in the near future. On the semantic analysis, the *meanings score* was not effective due to the fact that the Balkanet framework does not support grammatical conjugation, resulting to fewer results. It accepted only the nominative case. Comparing the two machine learning techniques, semi- and supervised learning, we note that their performance is in some cases quite similar. Even semi-supervised techniques are based on a small seed train-set, which is why they are called that way after all. There is some literature on unsupervised text classification but it is more useful when the main interest is clustering, not explicit classification on pre-defined classes (irony or not). We discussed earlier the value of semi-supervised learning in irony annotation, due to the fact that manual labeling is very subjective and not easily available. Humor per se is one the most disputed and personal virtues. As future work we could attempt an approach with word vectors or deep learning.

We researched on the theoretical ground of the Twitter use-cases, especially on what influences and motivates the individual to criticize and joke about politics. The empirical study, attempts to detect irony on Greek political tweets, to automatically label a big unlabeled dataset of them and to seek underlying relations between the irony that the parties receive and their actual election results. The performance of the two machine learning techniques is reasonably acceptable (supervised ~82%, semi-supervised ~83%) and produces similar results on predicting the fluctuation from previous election results, establishing our initial hypothesis. The collective learning approach detected fewer ironic tweets, that in our opinion is closer to the reality. The big unlabeled dataset assisted and contributed to this result.

The real-world application of irony detection could be useful to polling companies to get the pulse of social media in election periods as well as to the parties to get feedback. Another business-oriented aspect could be its use by brands in crisis management situations to leverage the opinion of the web. Zooming out, humor detection was always one of the desired targets of computational intelligence, where the machines will be able to empathize with humans in all aspects of speech, figurative or literal.

References

Barbieri, Francesco, Saggion, Horacio, 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.

Bird, Steven, Loper, Edward, Klein, Ewan, 2009. Natural Language Processing with Python. O'Reilly Media Inc.

Breiman, Leo, 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Chang, C.C., Lin, C.J., 2011. LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (3), 27.

Cheang, Henry S., Pell, Marc D., 2008. The sound of sarcasm. Speech Commun. 50 (5), 366–381.

Cleary, John G., Leonard, E. Trigg, 1995. K*: an instance-based learner using an entropic distance measure. In: Proceedings of the 12th International Conference on Machine learning. Vol. 5.

Colleoni, Elanor, Rozza, Alessandro, Arvidsson, Adam, 2014. Echo chamber or public sphere? Predicting political orientation and measuring political homophily in Twitter using big data. J. Commun. 64 (2), 317–332.

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

de-la-Peña-Sordo, Jorge, et al., 2013. "Filtering Trolling Comments through Collective Classification". Network and System Security. Springer, Berlin, Heidelberg, pp. 707–713.

Fangzhong, Su, Markert, Katja, 2009. Subjectivity recognition on word senses via semi-supervised mincuts. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics.

Gama, João, 2004. Functional trees. Mach. Learn. 55 (3), 219–250.

Gokhan, Tur, Hakkani-Tür, Dilek, Schapire, Robert E., 2005. Combining active and semi-supervised learning for spoken language understanding. Speech Commun. 45 (2), 171–186.

González-Ibánez, Roberto, Muresan, Smaranda, Wacholder, Nina, 2011. Identifying sarcasm in Twitter: a closer look. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2. Association for Computational Linguistics.

John, George H., Langley, Pat, 1995. Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.

Kalsnes, Bente, Krumsvik, Arne H., Storsul, Tanja, 2014. Social media as a political backchannel: Twitter use during televised election debates in Norway. Aslib J. Inf. Manag. 66 (3), 313–328.

Kermanidis, Katia Lida, Maragoudakis, Manolis, 2013. Political sentiment analysis of tweets before and after the Greek elections of May 2012. Int. J. Soc. Netw. Min. 1 (3), 298–317.

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

Mihalcea, Rada, Strapparava, Carlo, 2006. Learning to laugh (automatically): computational models for humor recognition. Comput. Intell. 22 (2), 126–142.

Namata, Galileo, et al., 2009. Collective classification for text classification. Text Min., 51–69.

Neville, Jennifer, Jensen, David, 2003. Collective classification with relational dependency networks. In: Proceedings of the Second International Workshop on Multi-Relational Data Mining.

Quinlan, J. Ross, 1993. C4.5: Programming for Machine Learning. Morgan Kauffmann, San Mateo.

Rajadesingan, Ashwin, Zafarani, Reza, Liu, Huan, 2014. Sarcasm detection on twitter: a behavioral modeling approach (Dissertation). Arizona State University, Arizona.

Reyes, Antonio, Rosso, Paolo, Veale, Tony, 2013. A multidimensional approach for detecting irony in twitter. Lang. Resour. Eval. 47 (1), 239–268.

Reyes, Antonio, Rosso, Paolo, 2011. Mining subjective knowledge from customer reviews: a specific case of irony detection. In: Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis. Association for Computational Linguistics.

Santos, Igor, Laorden, Carlos, Bringas, Pablo Garcia, 2011. Collective Classification for Unknown Malware Detection. SECRYPT.

Tsur, Oren, Davidov, Dmitry, Rappoport, Ari, 2010. ICWSM-A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews. ICWSM.

Tufis, Dan, Cristea, Dan, Stamou, Sofia, 2004. BalkaNet: ams, methods, results and perspectives. a general overview. Rom. J. Inf. Sci. Technol. 7 (1–2), 9–43.