UNIBA - Integrating distributional semantics features in a supervised approach for detecting irony in Italian tweets

Pierpaolo Basile and **Giovanni Semeraro**
Department of Computer Science
University of Bari Aldo Moro
Via, E. Orabona, 4 - 70125 Bari (Italy)
{pierpaolo.basile,giovanni.semeraro}@uniba.it

Abstract

English. This paper describes the UNIBA team participation in the IronITA 2018 task at EVALITA 2018. We propose a supervised approach based on LIBLINEAR that relies on keyword, polarity, micro-blogging features and representation of tweets in a distributional semantic model. Our system ranked 3rd and 4th in the irony detection subtask. We participated only in the constraint run exploiting the training data provided by the task organizers.

Italiano. *Questo articolo descrive la partecipazione del team UNIBA al task IronITA 2018 organizzato durante EVALITA 2018. Nell'articolo proponiamo un approccio supervisionato basato su LIBLINEAR che sfrutta le parole chiave, la polarità, attributi tipici dei micro-blog e la rappresentazione dei tweet in uno spazio semantico distribuzionale. Il nostro sistema si è classificato terzo e quarto nel sotto task di identificazione dell'ironia. Abbiamo partecipato solamente nel constraing run utilizzando i dati di training forniti dagli organizzatori del task.*

1 Introduction

The irony is defined as "the use of words that say the opposite of what you really mean, often as a joke and with a tone of voice that shows this"[1]. This suggests us that when we are analyzing written text for detecting irony, we should focus our attention on those words that are used in an unconventional context. For example, given the tweet: *"S&P ha declassato Mario Monti da Premier a Badante #declassaggi"*[2], we can observe that the word "badante" *(caregiver)* is used in an unconventional context, since "caregiver" usually does not co-occur with words "Premier" or "Mario Monti".

Following this idea in our work we introduce a feature able to detect words used out of their usual context. Moreover, we integrate further features based on keywords, bigrams, trigrams, polarity and micro-blogging features as reported in (Basile and Novielli, 2014). Our idea is supported by best systems participating in the Semeval-2018 task 3 - Irony detection in English tweets (Van Hee et al., 2018), where the best systems not based on deep learning exploit features based on polarity contrast information and context incongruity.

We evaluate our approach in the context of the IronITA task at EVALITA 2018 (Cignarella et al., 2018). The goal of the task is to predict irony in Italian tweets. The task is organized in two sub-tasks: 1) irony detection and 2) different types of irony. In the second sub-task participates must identify if irony belongs to sarcasm or not. In this paper, we propose an approach which is able to detect the presence of irony without taking into account different types of irony. We evaluate the approach in a constrained setting using only the data provided by task organizers. The only external resources exploited in our approach are a polarity lexicon and a collection of about 40M tweets randomly extracted from TWITA(Basile and Nissim, 2013) (a collection of about 800M Italian tweets).

The paper is structured as follows: Section 2 describes our system, while evaluation and results are reported in Section 3. Final remarks are provided in Section 4.

2 System Description

Our approach adopts a supervised classifier based on LIBLINEAR (Fan et al., 2008), in particular we use the L2-regularized L2-loss linear SVM. Each tweet is represented using several sets of features:

[1] Oxford Learner Dictionary

[2] In English: "S&P has downgraded Mario Monti from Premier to Caregiver"

keyword-based : keyword-based features exploit tokens occurring in the tweets. Unigrams, bigrams and trigrams are considered. During the tokenization we replace the user mentions and URLs with two metatokens: "_USER_", "_URL_";

microblogging : microblogging features take into account some attributes of the tweets that are peculiar in the context of microblogging. We exploit the following features: the presence of emoticons, item character repetitions[3], informal expressions of laughters[4] and the presence of exclamation and interrogative marks. All microblobbing features are binary.

polarity : this block contains features extracted from the SentiWordNet (Esuli and Sebastiani, 2006) lexicon. We translate SentiWordNet in Italian through MultiWordNet (Pianta et al., 2002). It is important to underline that SentiWordNet is a synset-based lexicon while our Italian translation is a word based lexicon. In order to automatically derive our Italian sentiment lexicon from SentiWordNet, we perform three steps. First, we translate the synset offset in SentiWordNet from version 3.0 to 1.6[5] using automatically generated mapping file. Then, we transfer the prior polarity of SentiWordNet to the Italian lemmata. Finally, we expand the lexicon using Morph-it! (Zanchetta and Baroni, 2005), a lexicon of inflected forms with their lemma and morphological features. We extend the polarity scores of each lemma to its inflected forms. Details about the creation of the sentiment lexicon are reported in (Basile and Novielli, 2014). The obtained Italian translation of SentiWordNet is used to compute three features based on prior polarity of words in the tweets: 1) the maximum positive polarity; 2) the maximum negative polarity; 3) polarity variation: for each token occurring in the tweet a tag is assigned, according to the highest polarity score of the token in the Italian lexicon. Tag values are in the set {OBJ, POS , NEG}. The sentiment variation counts how many switches from POS to NEG, or vice versa, occur in the tweet.

distributional semantics features : we compute two kinds of distributional semantics features:

1. given a set of unlabelled downloaded tweets, we build a geometric space in which each word is represented as a mathematical point. The similarity between words is computed as their closeness in the space. To represent a tweet in the geometric space, we adopt the superposition operator (Smolensky, 1990), that is the vector sum of all the vectors of words occurring in the tweet. We use the tweet vector $\overrightarrow{t}$ as a semantic feature in training our classifiers;
2. we extract three features that taking into account the usage of words in an unconventional context. In particular, for each word w_i we compute a score ac_i that measures how the word is out of its conventional context. Finally, we compute three features: the average, the maximum and the minimum of all the ac_i scores. More details about the computation of the ac_i score are reported in Subsection 2.1.

2.1 Distributional Semantics Features

The distributional semantics model is built on a collection of tweets. We randomly extract 40M tweets from TWITA and build a semantic space based on the Random Indexing (RI) (Sahlgren, 2005) technique using a context windows equals to 2. Moreover, we consider only words occurring more than ten times[6]. The context window is dynamic and it does not take into account words that are not in the vocabulary. Our vocabulary contains 105,543 terms.

The mathematical insight behind the RI is the projection of a high-dimensional space on a lower dimensional one using a random matrix; this kind of projection does not compromise distance metrics (Dasgupta and Gupta, 1999).

Formally, given a $n \times m$ matrix A and an $m \times k$ matrix R, which contains random vectors, we define a new $n \times k$ matrix B as:

$$A^{n,m} \cdot R^{m,k} = B^{n,k} \quad k << m \tag{1}$$

[3]These features usually plays the same role of intensifiers in informal writing contexts.

[4]i.e., sequences of "ah".

[5]Since MultiWordNet is based on WordNet 1.6.

[6]We call this set of words: the vocabulary.

The new matrix B has the property to preserve the distance between points, that is if the distance between two any points in A is d; then the distance d_r between the corresponding points in B will satisfy the property that $d_r \approx c \times d$. A proof of that is reported in the Johnson-Lindenstrauss lemma (Dasgupta and Gupta, 1999).

Specifically, RI creates the *WordSpace* in two steps:

1. A context vector is assigned to each word. This vector is sparse, high-dimensional and ternary, which means that its elements can take values in {-1, 0, 1}. A context vector contains a small number of randomly distributed non-zero elements, and the structure of this vector follows the hypothesis behind the concept of Random Projection;

2. Context vectors are accumulated by analyzing co-occurring words. In particular, the semantic vector for any word is computed as the sum of the context vectors for words that co-occur with the analyzed word.

Formally, given a corpus C of n documents, and a vocabulary V of m words extracted from C, we perform two steps: 1) assign a context vector c_i to each word in V; 2) compute a semantic vector sv_i for each word w_i as the sum of all context vectors assigned to words co-occurring with w_i. The context is the set of m words that precede and follow w_i.

For example, considering the following tweet: *"siete il buono della scuola fatelo capire"*. In the first step we assign a random vector to each term as follows:

$$c_{siete} = (-1, 0, 0, -1, 0, 0, 0, 0, 0, 0)$$
$$c_{buono} = (0, 0, 0, -1, 0, 0, 0, 1, 0, 0)$$
$$c_{scuola} = (0, 0, 0, 0, -1, 0, 0, 0, 1, 0)$$
$$c_{fatelo} = (0, 1, 0, 0, 0, -1, 0, 0, 0, 0)$$
$$c_{capire} = (-1, 0, 0, 0, 0, 0, 0, 0, 0, 1)$$

In the second step, we build a semantic vector for each term by accumulating random vectors of its co-occurring words. For example fixing $m = 2$, the semantic vector for the word *scuola* is the sum of the random vectors *siete, buono, fatelo, capire*. Summing these vectors, the semantic vector for *scuola* results in $(-1, 1, 0, -2, 0, -1, 0, 1, 0, 1)$. This operation is repeated for all the sentences in the corpus and for all the words in V. In this example, we used very small vectors, but in a real scenario, the vector dimension ranges from hundreds to thousands of dimensions. In particular, in our experiment we use a vector dimension equals to 200 with 10 no-zero elements.

In order to compute the ac_i score for a word w_i in a tweet, we build a context vector c_{w_i} as the sum of random vectors assigned to words that co-occur with w_i in the tweet. Then we compare the cosine similarity between c_{w_i} and the semantic vector sv_i assigned to w_i. The idea is to measure how the semantic vector is dissimilar to the context vector. If the word w_i has never appeared in the context under analysis, its semantic vector does not contain the random vectors of the words in the context, this results in low cosine similarity. Finally, the divergence from the context is computed as $1 - cosSim(c_{w_i}, sv_i)$.

3 Evaluation

We perform the evaluation using the data provided by the task organizers. The number of tweets in the training set is 3,977, while the testing set consists of 872 tweets. The only parameter to set in LIBLINEAR is C (the cost), after a 5-fold cross validation on training we set C=1.

We submit two runs: UNIBA1 includes the semantic vector representing the tweet as a feature, while UNIBA2 does not include this vector. Nevertheless, features about the divergence are included in both the runs.

Official results are reported in Table 1. Our runs rank third and fourth in the final rank. Our team is classified as second since the first two runs in the rank belong to the team1. We can notice that runs are very close in the rank. The last run is ranked below the baseline *random*, while any system is ranked below the baseline *baseline-mfc* that assigns the most frequent class (*non-ironic*).

Results show that our system is not able to improve performance exploiting the distributional representation of tweets, since the two runs report the same average F1-score. We performed further experiments in order to understand the contribution of each feature. Some relevant outcomes are reported in Table 2, in particular:

- keyword-based features are able to achieve the best performance, in particular bigrams

team	precision (non-ironic)	recall (non-ironic)	F1-score (non-ironic)	precision (ironic)	recall (ironic)	F1-score (ironic)	average F1-score
team1	0.785	0.643	0.707	0.696	0.823	0.754	0.731
team1	0.751	0.643	0.693	0.687	0.786	0.733	0.713
UNIBA1	0.748	0.638	0.689	0.683	0.784	0.730	0.710
UNIBA2	0.748	0.638	0.689	0.683	0.784	0.730	0.710
team3	0.700	0.716	0.708	0.708	0.692	0.700	0.704
team6	0.600	0.714	0.652	0.645	0.522	0.577	0.614
random	0.506	0.501	0.503	0.503	0.508	0.506	*0.505*
team7	0.505	0.892	0.645	0.525	0.120	0.195	0.420
baseline-mfc	0.501	1.000	0.668	0.000	0.000	0.000	*0.334*

Table 1: Task results.

run	note	no-iro-F	iro-F	avg-F
run1	all	0.6888	0.7301	0.7095
run2	no DSM	0.6888	0.7301	0.7095
1	keyword	0.6738	0.6969	0.6853
2	keyword, bigrams	0.6916	0.7219	0.7067
3	keyword, bigrams, trigrams	0.6992	0.7343	0.7168
4	keyword, bigrams, trigrams, blog	0.7000	0.7337	0.7168
5	keyword, bigrams, trigrams, polarity	0.6906	0.7329	0.7117
6	keyword, bigrams, trigrams, context	0.6937	0.7325	0.7131
7	only DSM	0.6166	0.6830	0.6406
8	only context	0.4993	0.5587	0.5290

Table 2: Task results obtained combining different types of features.

and trigrams contribute to improve the performance (run 1 and 2);

- DSM features introduce some kind of noise when are combined with other features, in fact run 4, 5 and 6 achieve good performance without DSM;

- DSM alone without any other kind of features is able to achieve remarkable results, it is important to notice that in this run only the tweet vector is used as a feature;

- blog, polarity, and context features are not able to give a contribution to the overall system performance, however we can observe that using only context features (only three features for each tweet) we are able to overcome both the baselines.

Analyzing results we can conclude that a more effective way to combine distributional with no-distributional features is needed. We plan to investigate as a future work the combination of two different kernels for distributional and keyword-based features.

4 Conclusions

We propose a supervised system for detecting irony in Italian tweets. The proposed system exploits different kinds of features: keyword-based, microblogging features, polarity, distributional semantics features and a score that measure how a word is used in an unconventional context. The word divergence from its conventional context is computed exploiting the distributional semantics model build by the Random Indexing.

Results prove that our system is able to achieve good performance and rank third in the official ranking. However, a deep study on different combinations of features shows that keyword-based features alone are able to achieve the best result, while distributional features introduce noise during the training. This outcome suggests the need for a different strategy for combining distributional a no-distributional features.

References

Valerio Basile and Malvina Nissim. 2013. Sentiment analysis on italian tweets. In *Proc. of WASSA 2013*, pages 100–107.

Pierpaolo Basile and Nicole Novielli. 2014. Uniba at evalita 2014-sentipolc task: Predicting tweet sentiment polarity combining micro-blogging, lexicon and semantic features. In *Proc. of EVALITA 2014*, pages 58–63, Pisa, Italy.

Alessandra Teresa Cignarella, Simona Frenda, Valerio Basile, Cristina Bosco, Viviana Patti, and Paolo Rosso. 2018. Overview of the evalita 2018 task on irony detection in italian tweets (ironita). In Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*, Turin, Italy. CEUR.org.

Sanjoy Dasgupta and Anupam Gupta. 1999. An elementary proof of the Johnson-Lindenstrauss lemma. Technical report, Technical Report TR-99-006, International Computer Science Institute, Berkeley, California, USA.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proc. of LREC*, pages 417–422.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. Multiwordnet: developing an aligned multilingual database. In *Proc. 1st Intl Conf. on Global WordNet*, pages 293–302.

Magnus Sahlgren. 2005. An Introduction to Random Indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE*, volume 5.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1-2):159–216, November.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50.

Eros Zanchetta and Marco Baroni. 2005. Morph-it!: a free corpus-based morphological resource for the italian language. *Proc. of the Corpus Linguistics Conf. 2005*.