10.12871/clicit2014217

A Multiple Kernel Approach for Twitter Sentiment Analysis in Italian

Giuseppe Castellucci$^{(\dagger)}$, **Danilo Croce**$^{(\ddagger)}$, **Diego De Cao**$^{(\ddagger)}$ and **Roberto Basili**$^{(\ddagger)}$
(†) Dept. of Electronic Engineering,
(‡) Dept. of Enterprise Engineering,
University of Roma, Tor Vergata
Via del Politecnico 1, Rome, 00133, Italy
{castellucci}@ing.uniroma2.it, {croce,decao,basili}@info.uniroma2.it

Abstract

English. This paper describes the UNITOR system that participated to the *SENTIment POLarity Classification* task within the context of Evalita 2014. The system has been developed as a workflow of Support Vector Machine classifiers. Specific features and kernel functions have been used to tackle the different sub-tasks, i.e. Subjectivity Classification, Polarity Classification and the pilot task Irony Detection. The system won 3 of the 6 evaluations carried out by the task organizers, and in the worst case it ranked in 4^{th} position w.r.t. about 10 participants.

Italiano. *Questo articolo descrive il sistema UNITOR che è stato valutato nel task di SENTIment POLarity Classification ad Evalita 2014. Il riconoscimento del sentimento nei Tweet è basato su un workflow di classificatori di tipo Support Vector Machine (SVM), il cui flusso è stato studiato appositamente per risolvere i diversi task proposti nella competizione. Rappresentazioni vettoriali specifiche sono state definite per modellare i tweet al fine di applicare funzioni Kernel che vengono utilizzate dai classificatori SVM. Il sistema ha ottenuto risultati promettenti risultando vincitore di 3 dei 6 task proposti.*

1 Introduction

Modern Internet technologies allow users to generate new contents, writing their opinions about facts, things and events. The interest in the analysis of the user-generated contents is rapidly growing. In particular, Sentiment Analysis (SA) of web data produced by users is becoming a crucial component for companies or politicians in order to check the mood on the web, and consequently adjust their strategies. Twitter[1] is one of the most popular social networking service that allows people to express themselves with very short messages. SA in Twitter represents a challenging task, as messages are short, informal and characterized by their own particular language, e.g. retweets ("*RT*"), user references ("@"), hashtags ("#") or other typical web slang, e.g. emoticons. Classical approaches to Sentiment Analysis (Pang et al., 2002; Pang and Lee, 2008) mainly focus on longer texts, e.g. movie reviews, resulting in performance drops when applied on tweets. Examples of tweet modeling within Machine Learning settings for the Twitter SA can be found in (Pak and Paroubek, 2010; Zanzotto et al., 2011; Kouloumpis et al., 2011; Agarwal et al., 2011; Croce and Basili, 2012; Castellucci et al., 2013; Rosenthal et al., 2014).

In this paper, the UNITOR system participating in the *Sentiment Polarity Classification* (SENTIPOLC) task (Basile et al., 2014) within the Evalita 2014 evaluation campaign is described. The system faces three proposed subtasks: *Subjectivity Classification*, *Polarity Classification* and the pilot task called *Irony Detection*. As the specific labeling of the challenge is rich and complex, we decomposed the analysis in different stages. The labeling of each tweet is determined by the application of a workflow of Support Vector Machine (Vapnik, 1998) classifiers. In this work, several kernel functions have been exploited to tackle the different nature of each subtask. The UNITOR system ranked among the 1^{st} and 4^{th} position in all the submitted runs, resulting the winning system in 3 of 6 evaluations.

In the rest of the paper, in Section 2 the classifiers, in terms of features, kernels are described and the adopted workflow is presented. In Section 3 the performance measures of the system are reported while Section 4 derives the conclusions.

[1] http://www.twitter.com

2 System Description

The UNITOR system participated to all the subtasks proposed in the SENTIPOLC (Basile et al., 2014) challenge: *Subjectivity Classification*, *Polarity Classification* and the pilot task *Irony Detection*. The first task aims at evaluating the performance of systems in capturing whether a message conveys a subjective position. The second task is intended to verify if a system is able to detect the polarity of a message, in terms of positive, neutral or negative classes. The last one is intended to verify the presence of irony.

2.1 Feature engineering

In our Supervised Learning setting, a multiple-kernel based approach has been adopted to acquire the SVM classifiers (Shawe-Taylor and Cristianini, 2004): the similarity between training and testing example is measured by kernel functions, that are applied to different feature representations, each engineered to capture different properties of each message.

First, all tweets have been processed through an adapted version of a Chaos natural language parser (Basili and Zanzotto, 2002). A normalization step is exploited before applying the Natural Language Processing chain. The following set of actions is performed: fully capitalized words are converted in their lowercase counterparts; hyperlinks are replaced by the token LINK; any character repeated more than three times are cleaned, as they cause high levels of lexical data sparseness (e.g. "*nooo!!!!!*" is converted into "*noo!!*"); all emoticons are replaced by special tokens[2].

Then, a set of feature vector is generated to let the SVM classifiers capture semantic properties of each tweet. In the rest of this Section, the representations of tweets are described.

Bag-Of-Word (BOW) is a representation that aims at capturing the lexical overlap between examples. A feature vector in which each dimension represents a lemma and a part-of-speech is derived from a tweet message. A boolean weighting is applied, i.e. a feature has a 1.0 value if the corresponding lemma and part-of-speech pair appears in the message.

SentixSum (SSUM) is a feature vector that is obtained using the Sentix (Basile and Nissim, 2013) lexicon. It is obtained aligning different existing resources. It consists of about 60.000 entries, each characterized by an Italian lemma, part-of-speech, WordNet (Miller, 1995) synset ID, and different polarity scores. Given a tweet, we derived the SSUM vector, as a 4-dimensional vector where each feature corresponds to the sum, with respect to each word, of the polarity scores that are available in the Sentix lexicon: positivity, negativity, polarity and intensity scores. The final vector is then normalized.

SentixDifference (SDIFF) is a feature vector describing how discordant are the words in a message. Again, this vector is obtained using the Sentix resource (Basile and Nissim, 2013). The SDIFF vector is 4-dimensional, and it reflects the 4 scores that can be extracted from this lexicon. In particular, each dimension is the result from the difference computed between the vectors of the maximally polar word and the minimally polar word. Formally, given $\vec{w_1}$ and $\vec{w_2}$ as the vectors in Sentix, representing the words respectively with the *maximum* and *minimum* polarity score respectively, then the SDIFF vector is computed as $sd(\vec{w_1}, \vec{w_2}) = \vec{w_1} - \vec{w_2}$.

Latent Semantic Analysis (LSA) representation aims at generalizing lexical information available through the BOW model. A vector representation for words is obtained from a co-occurrence Word Space built accordingly to the methodology described in (Sahlgren, 2006). A word-by-context matrix M is obtained through the analysis of a large scale corpus of 3 million of tweets. Each dimension is weighted through the Pointwise Mutual Information between a word and its context in a window of 3 words before or after. The *Latent Semantic Analysis* (Landauer and Dumais, 1997) technique is then applied as follows. The matrix M is decomposed through Singular Value Decomposition (SVD) (Golub and Kahan, 1965) into the product of three new matrices: U, S, and V so that S is diagonal and $M = USV^T$. M is then approximated by $M_k = U_k S_k V_k^T$, where only the first k columns of U and V are used, corresponding to the first k greatest singular values. The original statistical information about M is captured by the new k-dimensional space, which preserves the global structure while removing low-variant dimensions. Every word of a tweet is projected in the reduced Word Space and a message is represented by applying an *additive linear combination*. Only verbs, adjectives, nouns and hashtags are considered.

[2] We normalized 113 well-known emoticons in 13 classes.

IronyVector (IV) is a specific vector designed to capture the irony of messages. It has been inspired by some recent works on irony detection (Carvalho et al., 2009; Reyes et al., 2012). This is a 7-dimensional vector in which each value aims at capturing some linguistic feature of ironic messages. The features are the following: *hasQuotationMarks*, if the tweet contains a quotation mark; *hasQuestionMarks* if the message contains a question mark; *hasExclamationMarks* if the tweet contains an exclamation mark; *lastTokenIsAPunctuation* if the last token of a message is a punctuation; *lastTokenIsAHappySmile* if a tweet ends with a smile belonging to the *happy* category with respect to our classification; *lastTokenIsASadSmile* if last token is a sad smile; *lastTokenIsASmile* if message ends with a smile. Each activated dimension is boolean weighted, i.e. the value is 1.0.

Out-of-Topic Weighted BOW (WBOW) is a Bag-Of-Word vector representing the words in a message. The main difference with respect to the previous BOW representation is the adopted weighting scheme. In fact, in this case we leverage on the Word Space previously described. For each dimension representing a lemma/part-of-speech pair, its weight is computed as the cosine similarity between the LSA vector of the considered word and the vector obtained from the linear combination of all the other words in the message. This vector aims at capturing how a word is out of context in a sentence, and therefore it should help in capturing unconventional use of words, and it should be an indicator of an ironic use of language.

LSAIrony (LSAIR) is a 4-dimensional vector specifically designed for the irony detection tasks. Its purpose is to compute a measure of dissimilarity between the words in a tweet, exploiting, again, the idea that an ironic message makes an unconventional use of words. Each dimension is a measure of how much words are dissimilar in a specific grammatical category. Thus, the first dimension measures the dissimilarity in the Word Space of the verbs, the second dimension considers nouns, the third look at the dissimilarities between adjectives, while the last dimension takes into account all the words of the message.

2.2 A Cascade of SVM classifiers for Sentiment Analysis

In Figure 1 the workflow of SVM classifiers developed for the SENTIPOLC task is shown. Each tweet is pre-processed and feature vectors are generated as described in the previous Section. Separated representations are considered in the *constrained* and *unconstrained* settings. In the constrained setting only feature vectors using tweet information or public available lexica are considered. In the unconstrained setting, feature vectors are derived also by exploiting other tweet messages, that are used in the acquisition of the Word Space (LSA and LSAIR).

Each tweet, in terms of its multi-vector representation, is then fed to the classifiers, and it flows over the cascade following the diagram in Figure 1. At the end of the workflow, 7 possible outputs are allowed according to the specification of the task. A binary code is used to express the different outputs: 4 bits are used to express the *subjectivity*, *positivity*, *negativity* and *irony* of a message. For example, a tweet that is subjective, and expresses both a positive and negative sentiment is labeled as 1110.

In the following, the specific kernel functions used in each classification stage are reported.

Subjective classifier. At the first stage of the workflow, the *Subjectivity* classifier is invoked. This is a crucial step, as an error in the classification of the subjectivity of the message compromises the entire cascade. At this stage, the linear combination of a linear kernel is applied over the BOW and the SSUM vectors. In the unconstrained case, a 2-degree polynomial kernel (Shawe-Taylor and Cristianini, 2004) is applied on the BOW representation in combination with a linear kernel on SSUM and a linear kernel on LSA.

Explicit polarity classifier. Here, the classifier adopts the same representations and kernels that have been used for the Subjective classifier. Consequently, the resulting classification function only depends on the labels of the training material.

Explicit positive/negative classifier. Again, the same setting used in the previous classifiers is exploited. Instead of a single binary classifier discriminating between two classes (i.e. positive and negative), here we have two binary classifiers. This is necessary to enable the labeling of tweets conveying both a positive and negative polarity in opposition of a neutral polarity. This last labeling is assigned when both the explicit positive and negative classifiers express a negative confidence of the classification.

Irony classifier. When a tweet does not explic-

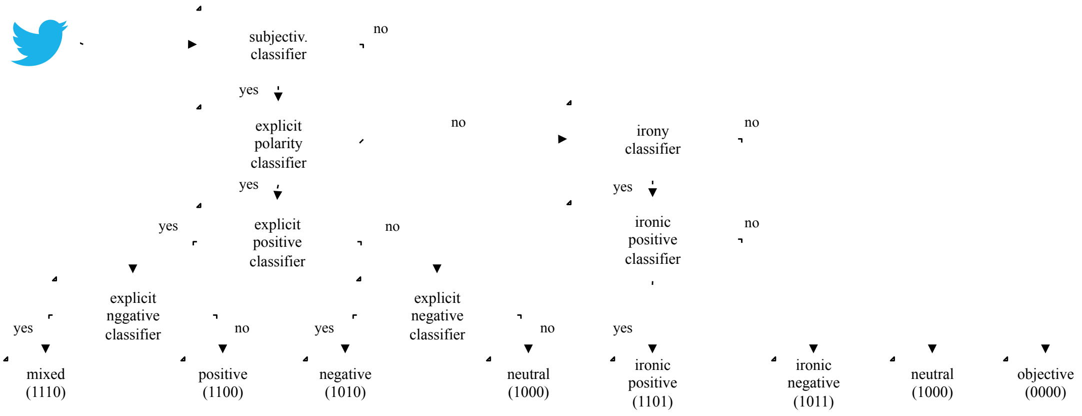

Figure 1: The UNITOR classifier workflow

itly express a sentiment, it may be ironic. It is reflected in the workflow as a classifier that separated ironic and neutral tweets. In the constrained case, the irony classifier adopts a `BOW` vector representation with a linear kernel combined with the `SDIFF` representation, again with a linear kernel. In the unconstrained case, a linear kernel applied on the `WBOW` representation is combined with a 2-degree polynomial kernel on the `BOW` vector and a linear kernel on the `SDIFF` vector.

Ironic positive/negative classifier. When a tweet is ironic, the last classification stage adopts more representations both in the constrained and in the unconstrained case. In the former, a linear kernel is applied on the `BOW`, `SDIFF` and `IV` vector. In the unconstrained case, the representations involved are: `BOW`, `SDIFF`, `IV`, `LSAIR` with a linear kernel, and the `LSA` with a RBF kernel (Shawe-Taylor and Cristianini, 2004). When training the explicit positive/negative and ironic positive/negative classifiers, the training material was split according the presence of irony as it affects also the way of expressing the polarity.

Each classifier is built by using a custom Java Support Vector Machine (SVM) implementation based on LibSVM (Chang and Lin, 2011). This implementation is specifically developed to support the combination of multiple representations and kernels. The Figure 1 reflects also the learning strategy that has been set up during the tuning phase: each classifier has been trained on the specific subset of the data of interest. Parameter tuning phase has been done by a fixed 80/20 split of the training data. Training data have been downloaded through the web interface proposed by the organizers[3], resulting in 4,033 tweet that were available at the time of the download. We lost 482 messages during the download phase due to Twitter policies. More information about the data, annotation process and evaluation metrics can be found in (Basile et al., 2014).

3 Results

In this Section the results of the UNITOR system are reported. Performance measures refer to the three subtasks proposed in the SENTIPOLC evaluation. Test data were downloaded through the same web interface provided by the organizers. Even for test data, some messages were no more available due to Twitter policies. Test data were supposed to be 1,938, while we downloaded 1,752 tweets. In Table 1 cumulative F1 scores and ranks for the UNITOR system are reported. Detailed performances are reported in the rest of the Section.

	C	U
Subjectivity Classification	68.7 (2)	69.0 (1)
Polarity Classification	63.0 (4)	65.5 (2)
Irony Detection	57.6 (1)	59.6 (1)

Table 1: UNITOR overall score and ranks. C and U refer to constrained and unconstrained runs

In Tables 2 and 3 the performances of the *Subjectivity Classification* subtask are reported. Both the constrained and unconstrained runs are here presented. UNITOR performances are remarkable as in the constrained run it ranks in 2^{nd} position, while in the unconstrained one is in 1^{st} position. In the constrained case, representations adopted are able to correctly determine whether a message is subjective with good precision, as demonstrated by the *Subjective* precision measure.

[3] `http://www.di.unito.it/~tutreeb/sentipolc-evalita14/tweet.html`

However, the winning system here was about 3 points ahead, in particular resulting more effective in the detection of non-subjective messages. The UNITOR system is not able to tackle messages that are too short. For example, some tweets were composed only by one or two words. In such messages there is not enough information for our classifiers. In the unconstrained case, the contribution of the LSA vector representation is demonstrated by the higher score obtained with respect to the constrained case. This makes the UNITOR system one of the best performing system in detecting the subjectivity of messages.

NotSubjective			Subjective		
P	R	F1	P	R	F1
57.7	**58.7**	58.2	**85.8**	73.6	79.2

Table 2: Subjectivity classification: constrained

NotSubjective			Subjective		
P	R	F1	P	R	F1
60.6	54.9	**57.6**	84.9	76.2	80.3

Table 3: Subjectivity classification: unconstrained

In Tables 4 and 5 the performances for the *Polarity Classification* are reported. In the constrained case, the results are comparable with the best systems, i.e. less than 5 points from the 1^{st} system. Analyzing the full results, our main problems are in the detection of the positive polarity classes, as we observed a 15 point drop of precision in the positive class. In the unconstrained case, the contribution of our tweet-specific Word Space derived vectors is again remarkable. In this case the UNITOR system is able to have the best performances in all the measures for the positive class (except the recall for the positive class). In the case of the negative class the system is not able to perform as well as the positive case. However, we consider this result very promising as the improvement w.r.t. our constrained run is about of 3 points. It means that the unsupervised analysis of a large tweet corpus is beneficial even for the polarity classification task. In this task, many misclassifications affect messages characterized by an implicit inversion of polarity. Moreover, messages that were not correctly recognized as ironic by the Explicit polarity classifier determine a more complex classification in the *Polarity Classification* stage, as we have a separated classifier for polarity in the ironic case.

In Tables 6 and 7 the performances of the UNITOR system on the pilot task *Irony Detection* are reported. In the constrained case, the UNITOR system reaches the 1^{st} position on the rank with a combined F1 score of 57.59. The system performs very well in detecting not-ironic messages, as demonstrated by the *NotIronic* columns. Probably this is due to the unbalanced dataset provided for this task. In fact, only 564 over 4515 messages in the training data were labelled as ironic. If the same ratio was in the test set, it can be seen as a bias for the evaluation. In the unconstrained case, the UNITOR system reaches again the 1^{st} position in the rank. The contribution of the unconstrained representations helped, as a gain of 2 points in the combined F1 score has been observed. Moreover, representations used in the unconstrained case allow to be more precise when a message is ironic, as the 4 points precision increment suggests. However, a drop in recall makes the two systems perform more or less the same in terms of Ironic F1 measure (about 35 points in F1 score in both cases).

Positivity						
P_0	R_0	$F1_0$	P_1	R_1	$F1_1$	F1
79.5	77.0	78.2	56.0	40.9	47.3	62.8
Negativity						
P_0	R_0	$F1_0$	P_1	R_0	$F1_1$	F1
72.2	60.1	65.6	61.4	60.2	60.8	63.2

Table 4: Polarity classification: constrained

Positivity						
P_0	R_0	$F1_0$	P_1	R_1	$F1_1$	F1
82.1	**77.5**	**79.7**	**60.8**	48.2	**53.7**	**66.7**
Negativity						
P_0	R_0	$F1_0$	P_1	R_0	$F1_1$	F1
73.8	59.9	66.2	62.1	**62.4**	62.2	64.2

Table 5: Polarity classification: unconstrained

NotIronic			Ironic		
P	R	F1	P	R	F1
93.1	69.6	79.6	26.6	**52.9**	**35.5**

Table 6: Irony detection: constrained

NotSubjective			Subjective		
P	R	F1	P	R	F1
92.1	**76.3**	**83.5**	**30.6**	**42.9**	**35.7**

Table 7: Irony detection: unconstrained

4 Conclusions

In this paper the description of the UNITOR system participating to the SENTIPOLC task at Evalita 2014 has been provided. The system won 3 of the 6 evaluations carried out in the task, and in the worst case it ranked in the 4^{th} position. Thus, the proposed classification strategy is one of the best performing in the Twitter Italian Sentiment Analysis scenario. The UNITOR system won the Irony Detection task both in constrained and unconstrained settings. Even if the evaluation dataset for this subtask was quite small, the irony specific features that were studied for this problem were able to detect irony in short messages. However, further work is needed to improve the overall (low) F1 scores. The nature of Twitter messages does not help, as tweets are very short and the amount of useful information for detecting irony is often out of the message. For these reasons, we think that more information can be extracted using message contexts, as demonstrated in (Vanzo et al., 2014b; Vanzo et al., 2014a) for the English and Italian languages.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Ws on Languages in Social Media*, pages 30–38. ACL.

Valerio Basile and Malvina Nissim. 2013. Sentiment analysis on italian tweets. In *Proceedings of the 4th Ws: Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 100–107.

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task. In *Proceedings of the 4th evaluation campaign of NLP and Speech tools for Italian (EVALITA)*, Pisa, Italy.

Roberto Basili and Fabio Massimo Zanzotto. 2002. Parsing engineering and empirical robustness. *Nat. Lang. Eng.*, 8(3):97–120.

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 *1st CIKM WS on Topic-sentiment Analysis for Mass Opinion*, pages 53–56. ACM.

Giuseppe Castellucci, Simone Filice, Danilo Croce, and Roberto Basili. 2013. Unitor: Combining syntactic and semantic kernels for twitter sentiment analysis. In *2nd Joint Conf. *SEM: Vol. 2: Proceedings of SemEval*, pages 369–374. ACL.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

Danilo Croce and Roberto Basili. 2012. Grammatical feature engineering for fine-grained ir tasks. In *IIR*, pages 133–143.

Gene Golub and W. Kahan. 1965. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Numerical Analysis*, 2(2):pp. 205–224.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! In *ICWSM*.

Tom Landauer and Sue Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104.

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

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of LREC*. ELRA.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP, vol. 10*, pages 79–86. ACL.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data and Knowledge Engineering*, 74(0):1 – 12.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th SemEval WS*, pages 73–80. ACL and Dublin City University.

Magnus Sahlgren. 2006. *The Word-Space Model*. Ph.D. thesis, Stockholm University.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Andrea Vanzo, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2014a. A context based model for twitter sentiment analysis in italian. In *Proceedings of CLIC (To Appear)*, Pisa, Italy, December.

Andrea Vanzo, Danilo Croce, and Roberto Basili. 2014b. A context-based model for sentiment analysis in twitter. In *Proceedings of COLING*, pages 2345–2354. ACL and Dublin City University.

Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. Wiley-Interscience.

Fabio Massimo Zanzotto, Marco Pennacchiotti, and Kostas Tsioutsiouliklis. 2011. Linguistic redundancy in twitter. In *EMNLP*, pages 659–669.