UNITOR: Combining Syntactic and Semantic Kernels for Twitter Sentiment Analysis

Giuseppe Castellucci(†), **Simone Filice**(‡), **Danilo Croce**(⋆), **Roberto Basili**(⋆)

(†) Dept. of Electronic Engineering
(‡) Dept. of Civil Engineering and Computer Science Engineering
(⋆) Dept. of Enterprise Engineering
University of Rome, Tor Vergata
Rome, Italy

{castellucci,filice,croce,basili}@info.uniroma2.it

Abstract

In this paper, the UNITOR system participating in the SemEval-2013 *Sentiment Analysis in Twitter* task is presented. The polarity detection of a tweet is modeled as a classification task, tackled through a Multiple Kernel approach. It allows to combine the contribution of complex kernel functions, such as the Latent Semantic Kernel and Smoothed Partial Tree Kernel, to implicitly integrate syntactic and lexical information of annotated examples. In the challenge, UNITOR system achieves good results, even considering that no manual feature engineering is performed and no manually coded resources are employed. These kernels in-fact embed distributional models of lexical semantics to determine expressive generalization of tweets.

1 Introduction

Web 2.0 and Social Networks technologies allow users to generate contents on blogs, forums and new forms of communication (such as micro-blogging) writing their opinion about facts, things, events. The analysis of this information is crucial for companies, politicians or other users in order to learn what people think, and consequently to adjust their strategies. In such a scenario, the interest in the analysis of the sentiment expressed by people is rapidly growing. Twitter[1] represents an intriguing source of information as it is used to share opinions and sentiments about brands, products, or situations (Jansen et al., 2009).

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

On the other hand, tweet analysis represents a challenging task for natural language processing systems. Let us consider the following tweets, evoking a *positive* (1), and *negative* (2) polarity, respectively.

Porto amazing as the sun sets... http://bit.ly/c28w (1)

@knickfan82 Nooooo ;(they delayed the knicks game until Monday! (2)

Tweets are short, informal and characterized by their own particular language with "Twitter syntax", e.g. retweets ("*RT*"), user references ("@"), hashtags ("*#*") or other typical web abbreviations, such as emoticons or acronyms.

Classical approaches to sentiment analysis (Pang et al., 2002; Pang and Lee, 2008) are not directly applicable to tweets: most of them focus on relatively large texts, e.g. movie or product reviews, and performance drops are experimented in tweets scenario. Some recent works tried to model the sentiment in tweets (Go et al., 2009; Pak and Paroubek, 2010; Kouloumpis et al., 2011; Davidov et al., 2010; Bifet and Frank, 2010; Croce and Basili, 2012; Barbosa and Feng, 2010; Agarwal et al., 2011). Specific approaches and feature modeling are used to achieve good accuracy levels in tweet polarity recognition. For example, the use of n-grams, POS tags, polarity lexicon and tweet specific features (e.g. hashtag, retweet) are some of the component exploited by these works in combination with different machine learning algorithms (e.g. Naive Bayes (Pak and Paroubek, 2010), k-NN strategies (Davidov et al., 2010), SVM and Tree Kernels (Agarwal et al., 2011)).

In this paper, the UNITOR system participating

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 369–374, Atlanta, Georgia, June 14-15, 2013.

in the SemEval-2013 *Sentiment Analysis in Twitter* task (Wilson et al., 2013) models the sentiment analysis stage as a classification task. A Support Vector Machine (SVM) classifier learns the association between short texts and polarity classes (i.e. *positive*, *negative*, *neutral*). Different kernel functions (Shawe-Taylor and Cristianini, 2004) have been used: each kernel aims at capturing specific aspects of the semantic similarity between two tweets, according to syntactic and lexical information. In particular, in line with the idea of using convolution tree kernels to model complex semantic tasks, e.g. (Collins and Duffy, 2001; Moschitti et al., 2008; Croce et al., 2011), we adopted the *Smoothed Partial Tree Kernel* (Croce et al., 2011) (SPTK). It is a state-of-the-art convolution kernel that allows to measure the similarity between syntactic structures, which are partially similar and whose nodes can differ but are nevertheless semantically related. Moreover, a Bag-of-Word and a Latent Semantic Kernel (Cristianini et al., 2002) are also combined with the SPTK in a multi-kernel approach.

Our aim is to design a system that exhibits wide applicability and robustness. This objective is pursued by adopting an approach that avoids the use of any manually coded resource (e.g. a polarity lexicon), but mainly exploits distributional analysis of unlabeled corpora: the generalization of words meaning is achieved through the construction of a Word Space (Sahlgren, 2006), which provides an effective distributional model of lexical semantics.

In the rest of the paper, in Section 2 we will deeply explain our approach. In Section 3 the results achieved by our system in the SemEval-2013 challenge are described and discussed.

2 System Description

This section describes the approach behind the `UNITOR` system. Tweets pre-processing and linguistic analysis is described in Section 2.1, while the core modeling is described in 2.2.

2.1 Tweet Preprocessing

Tweets are noisy texts and a pre-processing phase is required to reduce data sparseness and improve the generalization capability of the learning algorithms. The following set of actions is performed before applying the natural language processing chain:

- fully capitalized words are converted in their lowercase counterparts;
- reply marks are replaced with the pseudo-token `USER`, and POS tag is set to `$USR`;
- hyperlinks are replaced by the token `LINK`, whose POS is `$URL`;
- *hashtags* are replaced by the pseudo-token `HASHTAG`, whose POS is imposed to `$HTG`;
- characters consecutively repeated more than three times are cleaned as they cause high levels of lexical data sparseness (e.g. "*nooo!!!!!*" and "*nooooo!!!*" are both converted into "*noo!!*");
- all emoticons are replaced by `SML_CLS`, where `CLS` is an element of a list of classified emoticons (113 emoticons in 13 classes).

For example, the tweet in the example 2 is normalized in '*user noo sml_cry they delayed the knicks game until monday*'. Then, we apply an almost standard NLP chain with *Chaos* (Basili and Zanzotto, 2002). In particular, we process each tweet to produce *chunks*. We adapt the POS Tagging and Chunking phases in order to correctly manage the pseudo-tokens introduced in the normalization step. This is necessary because tokens like `SML_SAD` are tagged as nouns, and they influence the chunking quality.

2.2 Modeling Kernel Functions

Following a summary of the employed kernel functions is provided.

Bag of Word Kernel (BOWK) A basic kernel function that reflects the lexical overlap between tweets. Each text is represented as a vector whose dimensions correspond to different words. Each dimension represents a boolean indicator of the presence or not of a word in the text. The kernel function is the cosine similarity between vector pairs.

Lexical Semantic Kernel (LSK) A kernel function is obtained to generalize the lexical information of tweets, without exploiting any manually coded resource. Basic lexical information is obtained by a co-occurrence Word Space built accordingly to the methodology described in (Sahlgren, 2006) and (Croce and Previtali, 2010). A word-by-context matrix M is obtained through a large scale corpus analysis. Then the *Latent Semantic Analysis* (Lan-

dauer and Dumais, 1997) technique is 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, i.e. distribution noise. The result is that every word is projected in the reduced Word Space and an entire tweet is represented by applying an *additive linear combination*. Finally, the resulting kernel function is the cosine similarity between vector pairs, in line with (Cristianini et al., 2002).

Smoothed Partial Tree Kernel (SPTK) In order to exploit the syntactic information of tweets, the *Smoothed Partial Tree Kernel* proposed in (Croce et al., 2011) is adopted. Tree kernels exploit syntactic similarity through the idea of convolutions among substructures. Any tree kernel evaluates the number of common substructures between two trees T_1 and T_2 without explicitly considering the whole fragment space. Its general equation is reported hereafter:

$$TK(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2), \qquad (3)$$

where N_{T_1} and N_{T_2} are the sets of the T_1's and T_2's nodes, respectively and $\Delta(n_1, n_2)$ is equal to the number of common fragments rooted in the n_1 and n_2 nodes. The function Δ determines the nature of the kernel space. In the SPTK formulation (Croce et al., 2011) this function emphasizes lexical nodes. It computes the similarity between lexical nodes as the similarity between words in the Word Space. So, this kernel allows a generalization both from the syntactic and the lexical point of view.

However, tree kernel methods are biased by parsing accuracy and standard NLP parsers suffer accuracy loss in this scenario (Foster et al., 2011). It is mainly due to the complexities of the language adopted in tweets. In this work, we do not use a representation that depends on full parse trees. A syntactic representation derived from tweets chunking (Tjong Kim Sang and Buchholz, 2000) is here adopted, as shown in Figure 1.

Notice that no explicit manual feature engineering is applied. On the contrary we expect that discriminative lexical and syntactic information (e.g. negation) is captured by the kernel in the implicit feature space, as discussed in (Collins and Duffy, 2001).

A multiple kernel approach Kernel methods are appealing as they can be integrated in various machine learning algorithms, such as SVM. Moreover a combination of kernels is still a kernel function (Shawe-Taylor and Cristianini, 2004). We employed a linear combination αBOWK + βLSK + γSPTK in order to exploit the lexical properties captured by BOWK (and generalized by LSK) and the syntactic information of the SPTK. In our experiments, the kernel weights α, β and γ are set to 1.

3 Results and Discussion

In this section experimental results of the UNITOR system are reported.

3.1 Experimental setup

In the *Sentiment Analysis in Twitter* task, two subtasks are defined: Contextual Polarity Disambiguation (Task A), and Message Polarity Classification (Task B). The former deals with the polarity classification (*positive*, *negative* or *neutral*) of a marked occurrence of a word or phrase in a tweet context. For example the adjective "*amazing*" in example 1 expresses a positive marked word. The latter deals with the classification of an entire tweet with respect to the three classes *positive*, *negative* and *neutral*. In both subtasks, we computed a fixed (80%-20%) split of the training data for classifiers parameter tuning. Tuned parameters are the *regularization parameter* and the *cost factor* (Morik et al., 1999) of the SVM formulation. The former represents the trade off between a training error and the margin. The latter controls the trade off between positive and negative examples. The learning phase is made available by an extended version of SVM-LightTK[2], implementing the smooth matching between tree nodes.

We built a Word Space based on about 1.5 million of tweets downloaded during the challenge period using the topic name from the trial material as

[2] http://disi.unitn.it/moschitti/Tree-Kernel.htm

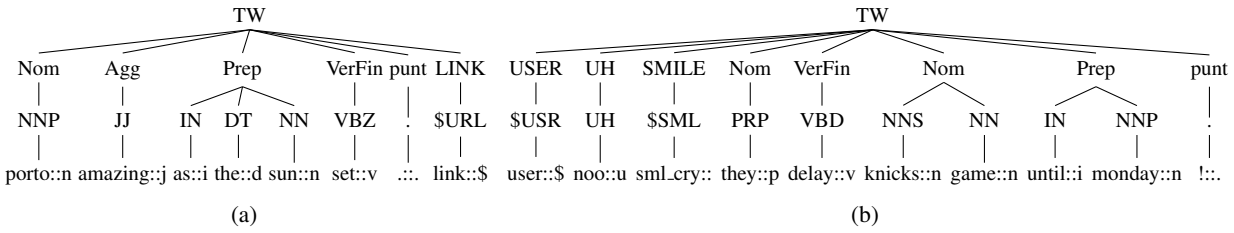

Figure 1: Chunk-based tree derived from examples (1) and (2)

query terms. We normalized and analyzed tweets as described in section 2.1. Words occurring more than 100 times in the source corpus are represented as vectors. The $10,000$ most frequent words in the corpus are considered as contexts and the co-occurrence scores are measured in a window of size $n = \pm 5$. Vector components are weighted through the Pointwise Mutual Information (PMI), and dimensionality reduction is applied through SVD with a cut of $k = 250$.

The task requires to classify two different texts: tweets and sms. Sms classification is intended to verify how well a system can scale on a different domain. In the testing phase two types of submissions are allowed. *Constrained* results refer to the case where systems are trained only with the released data. *Unconstrained* results refer to the case where additional training material is allowed. Evaluation metrics adopted to compare systems are *Precision*, *Recall* and *F1-Measure*. *Average F1* of the *positive* and *negative* classes is then used to generate ranks. Further information about the task is available in (Wilson et al., 2013).

3.2 Results over Contextual Polarity Disambiguation

We tackled Task A with a multi-kernel approach combining the kernel functions described in Section 2.2. The final kernel is computed as the linear combination of the kernels, as shown in Equation 4.

$$\begin{aligned} k(t_1, t_2) = & SPTK(\phi_A(t_1), \phi_A(t_2)) \\ & + BOWK(\psi_A(t_1), \psi_A(t_2)) \\ & + LSK(\tau_A(t_1), \tau_A(t_2)) \end{aligned} \quad (4)$$

where t_1, t_2 are two tweet examples. The $\phi_A(x)$ function extracts the 4-level chunk tree from the tweet x; nodes (except leaves) covering the marked instance in x are highlighted in the tree with `-POL`. The $\psi_A(x)$ function extracts the vector representing the Bag-of-Word of the words inside the marked instance of x, while τ_A builds the LSA vectors of the words occurring within the marked span of x. Referring to example 1, both $\psi_A(x)$ and τ_A point to the "*amazing*" adjective. Finally, $k(t_1, t_2)$ returns the similarity between t_1 and t_2 accordingly to our modeling. As three polarity classes are considered, we adopt a multi-classification schema accordingly to a *One-Vs-All* strategy (Rifkin and Klautau, 2004): the final decision function consists in the selection of the category associated with the maximum SVM margin.

Rank	4/19	class	precision	recall	f1
		positive	.8375	.7750	.8050
Avg-F1	.8249	negative	.8103	.8822	.8448
		neutral	.3475	.3082	.3267

Table 1: Task A results for the sms dataset

Rank	7/21	class	precision	recall	f1
		positive	.8739	.8844	.8791
Avg-F1	.8460	negative	.8273	.7988	.8128
		neutral	.2778	.3125	.2941

Table 2: Task A results for the twitter dataset

Tables 1 and 2 report the results of the `UNITOR` system in the Task A. Only the constrained setting has been submitted. The performance of the proposed approach is among the best ones and we ranked 4^{th} and 7^{th} among about 20 systems.

The system seems to be able to generalize well from the provided training data, and results are remarkable, especially considering that no manual annotated lexical resources were adopted and no manual feature engineering is exploited. It demonstrates that a multi-kernel approach, with the proposed shallow syntactic representation, is able to correctly classify the sentiment in out-of-domain contexts too. Syntax is well captured by the SPTK and the lexical generalization provided by the Word Space allows to generalize in the sms scenario.

3.3 Results over Message Polarity Classification

A multi-kernel approach is adopted for this task too, as described in the following Equation 5:

$$k(t_1, t_2) = SPTK(\phi_B(t_1), \phi_B(t_2)) + BOWK(\psi_B(t_1), \psi_B(t_2)) + LSK(\tau_B(t_1), \tau_B(t_2)) \quad (5)$$

The $\phi_B(x)$ function extracts a tree representation of x. In this case no nodes in the trees are marked. The $\psi_B(x)$ function extracts Bag-of-Word vectors for all the words in the tweet x, while $\tau_B(x)$ extracts the linear combination of vectors in the Word Space for adjectives, nouns, verbs and special tokens (e.g. hashtag, smiles) of the words in x. Again, a *One-Vs-All* strategy (Rifkin and Klautau, 2004) is applied.

Constrained run. Tables 3 and 4 report the result in the constrained case. In the sms dataset our system suffers more with respect to the tweet one. In both cases, the system shows a performance drop on the *negative* class. It seems that the multi-kernel approach needs more examples to correctly disambiguate elements within this class. Indeed, *negative* class cardinality was about 15% of the training data, while the *positive* and *neutral* classes approximately equally divided the remaining 85%. Moreover, it seems that our system confuses polarized classes with the *neutral* one. For example, the tweet "*going Hilton hotel on Thursday for #cantwait*" is classified as *neutral* (the gold label is *positive*). In this case, the hashtag is the sentiment bearer, and our model is not able to capture this information.

Rank	13/29	class	precision	recall	f1
		positive	.5224	.7358	.6110
Avg-F1	.5122	negative	.6019	.3147	.4133
		neutral	.7883	.7798	.7840

Table 3: Task B results for the sms dataset in the constrained case

Rank	13/36	class	precision	recall	f1
		positive	.7394	.6514	.6926
Avg-F1	.5827	negative	.6366	.3760	.4728
		neutral	.6397	.8085	.7142

Table 4: Task B results for the twitter dataset in the constrained case

Unconstrained run. In the unconstrained case we trained our system adding 2000 *positive* examples and 2000 *negative* examples to the provided training set. These additional tweets were downloaded from Twitter during the challenge period using *positive* and *negative* emoticons as query terms. The underlying hypothesis is that the polarity of the emoticons can be extended to the tweet (Pak and Paroubek, 2010; Croce and Basili, 2012). In tables 5 and 6 performance measures in this setting are reported.

Rank	10/15	class	precision	recall	f1
		positive	.4337	.7317	.5446
Avg-F1	.4888	negative	.3294	.6320	.4330
		neutral	.8524	.3584	.5047

Table 5: Task B results for the sms dataset in the unconstrained case

Rank	5/15	class	precision	recall	f1
		positive	.7375	.6399	.6853
Avg-F1	.5950	negative	.5729	.4509	.5047
		neutral	.6478	.7805	.7080

Table 6: Task B results for the twitter dataset in the unconstrained case

In this scenario, sms performances are again lower than the twitter case. This is probably due to the fact that the sms context is quite different from the twitter one. This is not true for Task A: polar expressions are more similar in sms and tweets. Again, we report a performance drop on the *negative* class. However, using more negative tweets seems to be beneficial. The F1 for this class increased of about 3 points for both datasets. Our approach thus needs more examples to better generalize from data.

In the future, we should check the redundancy and novelty of the downloaded material, as early discussed in (Zanzotto et al., 2011). Moreover, we will explore the possibility to automatically learn the kernel linear combination coefficients in order to optimize the balancing between kernel contributions (Gönen and Alpaydin, 2011).

Acknowledgements

This work has been partially funded by the Italian Ministry of Industry within the "Industria 2015" Framework, under the project DIVINO (MI01_00234).

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *COLING*, pages 36–44, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Albert Bifet and Eibe Frank. 2010. Sentiment knowledge discovery in twitter streaming data. In *Proceedings of the 13th international conference on Discovery science*, pages 1–15, Berlin, Heidelberg. Springer-Verlag.

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Proceedings of Neural Information Processing Systems (NIPS'2001)*, pages 625–632.

Nello Cristianini, John Shawe-Taylor, and Huma Lodhi. 2002. Latent semantic kernels. *J. Intell. Inf. Syst.*, 18(2-3):127–152.

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

Danilo Croce and Daniele Previtali. 2010. Manifold learning for the semi-supervised induction of framenet predicates: an empirical investigation. In *GEMS 2010*, pages 7–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of EMNLP*, Edinburgh, Scotland, UK.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *COLING*, pages 241–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jennifer Foster, Özlem Çetinoglu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. #hardtoparse: Pos tagging and parsing the twitterverse. In *Analyzing Microtext*.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision.

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

Mehmet Gönen and Ethem Alpaydin. 2011. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268.

Bernard J. Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. *J. Am. Soc. Inf. Sci. Technol.*, 60(11):2169–2188, November.

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.

Katharina Morik, Peter Brockhausen, and Thorsten Joachims. 1999. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *ICML*, pages 268–277, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Alessandro Moschitti, Daniele Pighin, and Robert Basili. 2008. Tree kernels for semantic role labeling. *Computational Linguistics*, 34.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*.

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

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP*, volume 10, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ryan Rifkin and Aldebaro Klautau. 2004. In defense of one-vs-all classification. *J. Mach. Learn. Res.*, 5:101–141, December.

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, New York, NY, USA.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: chunking. In *ConLL '00*, pages 127–132, Stroudsburg, PA, USA. Association for Computational Linguistics.

Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Alan Ritter, Sara Rosenthal, and Veselin Stoyonov. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

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