UNITOR: Aspect Based Sentiment Analysis with Structured Learning

Giuseppe Castellucci$^{(\dagger)}$, Simone Filice$^{(\ddagger)}$, Danilo Croce$^{(\star)}$, Roberto Basili$^{(\star)}$

($\dagger$) Dept. of Electronic Engineering
($\ddagger$) Dept. of Civil Engineering and Computer Science Engineering
($\star$) Dept. of Enterprise Engineering
University of Roma, Tor Vergata, Italy
{castellucci,filice}@ing.uniroma2.it; {croce,basili}@info.uniroma2.it

Abstract

In this paper, the UNITOR system participating in the SemEval-2014 Aspect Based Sentiment Analysis competition is presented. The task is tackled exploiting Kernel Methods within the Support Vector Machine framework. The Aspect Term Extraction is modeled as a sequential tagging task, tackled through SVM^{hmm}. The Aspect Term Polarity, Aspect Category and Aspect Category Polarity detection are tackled as a classification problem where multiple kernels are linearly combined to generalize several linguistic information. In the challenge, UNITOR system achieves good results, scoring in almost all rankings between the 2^{nd} and the 8^{th} position within about 30 competitors.

1 Introduction

In recent years, many websites started offering a high level interaction with users, who are no more a passive audience, but can actively produce new contents. For instance, platforms like Amazon[1] or TripAdvisor[2] allow people to express their opinions on products, such as food, electronic items or clothes. Obviously, companies are interested in understanding what customers think about their brands and products, in order to implement corrective strategies on products themselves or on marketing solutions. Performing an automatic analysis of user opinions is then a very hot topic. The automatic extraction of subjective information in text materials is generally referred as *Sentiment Analysis* or *Opinion Mining* and it is performed via natural language processing, text analysis and computational linguistics techniques. Task 4 in SemEval 2014 edition[3] (Pontiki et al., 2014) aims at promoting research on *Aspect Based Opinion Mining* (Liu, 2007), which is approached as a cascade of 4 subtasks. For example, let us consider the sentence:

The fried rice is amazing here. (1)

The *Aspect Term Extraction* (ATE) subtask aims at finding words suggesting the presence of aspects on which an opinion is expressed, e.g. `fried rice` in sentence 1. In the *Aspect Term Polarity* (ATP) task the polarity evoked for each aspect is recognized, i.e. a `positive` polarity is expressed with respect to `fried rice`. In the *Aspect Category Detection* (ACD) task the category evoked in a sentence is identified, e.g. the `food` category in sentence 1). In the *Aspect Category Polarity* (ACP) task the polarity of each expressed category is recognized, e.g. a `positive` category polarity is expressed in sentence 1.

Different strategies have been experimented in recent years. Classical approaches are based on machine learning techniques and rely on simple representation features, such as unigrams, bigrams, Part-Of-Speech (POS) tags (Pang et al., 2002; Pang and Lee, 2008; Wiebe et al., 1999). Other approaches adopt sentiment lexicons in order to exploit some sort of prior knowledge about the polar orientation of words. These resources are usually semi-automatically compiled and provide scores associating individual words to sentiments or polarity orientation.

In this paper, the UNITOR system participating to the SemEval-2014 *Aspect Based Sentiment Analysis* task (Pontiki et al., 2014) is presented. The ATE task is modeled as a sequential labeling problem. A sentence is considered as a sequence of tokens: a Markovian algorithm is adopted in

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

[2] http://www.tripadvisor.com

[3] http://alt.qcri.org/semeval2014/task4/

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 761–767,
Dublin, Ireland, August 23-24, 2014.

order to decide what is an aspect term . All the remaining tasks are modeled as multi-kernel classification problems based on Support Vector Machines (SVMs). Various representation have been exploited using proper kernel functions (Shawe-Taylor and Cristianini, 2004a). Tree Kernels (Collins and Duffy, 2001; Moschitti et al., 2008; Croce et al., 2011) are adopted in order to capture structural sentence information derived from the parse tree. Moreover, corpus-driven methods are used in order to acquire meaning generalizations in an unsupervised fashion (e.g. see (Pado and Lapata, 2007)) through the analysis of distributions of word occurrences in texts. It is obtained by the construction of a Word Space (Sahlgren, 2006), which provides a distributional model of lexical semantics. Latent Semantic Kernel (Cristianini et al., 2002) is thus applied within such space.

In the remaining, in Section 2 and 3 we will explain our approach in more depth. Section 4 discusses the results in the SemEval-2014 challenge.

2 Sequence Labeling for ATE

The Aspect Term Extraction (ATE) has been modeled as a *sequential tagging* process. We consider each token representing the *beginning* (B), the *inside* (I) or the *outside* (O) of an argument. Following this IOB notation, the resulting ATE representation of a sentence like "*The* [*fried rice*]$_{\text{ASPECTTERM}}$ *is amazing here*" can be expressed by labeling each word according to its relative position, i.e.: [*The*]$_{\text{O}}$ [*fried*]$_{\text{B}}$ [*rice*]$_{\text{I}}$ [*is*]$_{\text{O}}$ [*amazing*]$_{\text{O}}$ [*here*]$_{\text{O}}$.

The ATE task is thus a labeling process that determines the individual (correct IOB) class for each token. The labeling algorithm used is SVM^{hmm} (Altun et al., 2003)[4]: it combines both a discriminative approach to estimate the probabilities in the model and a generative approach to retrieve the most likely sequence of tags that explains a sequence. Given an input sequence $\mathbf{x} = (x_1 \dots x_l) \in \mathcal{X}$ of feature vectors $x_1 \dots x_l$, the model predicts a tag sequence $\mathbf{y} = (y_1 \dots y_l) \in \mathcal{Y}$ after learning a linear discriminant function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ over input-output pairs. The labeling $f(\mathbf{x})$ is thus defined as: $f(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w})$ and it is obtained by maximizing F over the response variable, $\mathbf{y}$, for a specific given input $\mathbf{x}$. F is linear in some combined feature representation of inputs and outputs $\Phi(\mathbf{x}, \mathbf{y})$, i.e. $F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$.

In SVM^{hmm} the observations $x_1 \dots x_l$ can be naturally expressed in terms of feature vectors. In particular, we modeled each word through a set of lexical and syntactic features, as described in the following section.

2.1 Modeling Features for ATE

In the discriminative view of SVM^{hmm}, each word is represented by a feature vector, describing its different observable properties. For instance, the word *rice* in the example 1 is modeled through the following features: *Lexical features*: its lemma (*rice*) and POS tag (*NN*); *Prefixes and Suffixes*: the first n and the last m characters of the word ($n = m = 3$) (e.g. *ric* and *ice*); *Contextual features*: the left and right lexical contexts represented by the 3 words before (*BEGIN*::BB *the*::DT *fried*::JJ) and after (*is*::VBZ *amazing*::JJ *here*::RB); the left and right syntactic contexts as the POS bi-grams and tri-grams occurring *before* (i.e. BB_DT DT_JJ BB_DT_JJ) and *after* (i.e. VBZ_JJ JJ_RB VBZ_JJ_RB) the word; *Grammatical features*: features derived from the dependency graph associated to the sentence, i.e. boolean indicators that capture if the token is involved in a *Subj*, *Obj* or *Amod* relation in the corresponding graph.

3 Multiple Kernel Approach for Polarity and Category Detection

We approached the remaining three subtasks of the pipeline as classification problems with multiple kernels, in line with (Castellucci et al., 2013). We used Support Vector Machines (SVMs) (Joachims, 1999), a maximum-margin classifier that realizes a linear discriminative model. The kernelized version of SVM learns from instances x_i exploiting rich similarity measures (i.e.the kernel functions) $K(x_i, x_j) = \langle \phi(x_i) \cdot \phi(x_j) \rangle$. In this way projection functions $\phi(\cdot)$ can be implicitly used in order to transform the initial feature space into a more expressive one, where a hyperplane that separates the data with the widest margin can be found. Kernels can directly operate on variegate forms of representation, such as feature vectors, trees, sequences or graphs. Then, modeling instances in different representations, specific kernels can be defined in order to explore different linguistic information. These variety of kernel functions

[4] www.cs.cornell.edu/People/tj/svm_light/svm_hmm.html

$K_1 \dots K_n$ can be independently defined and the combinations $K_1 + K_2 + \dots$ of multiple functions can be integrated into SVM as they are still kernels. The next section describes the representations as well as the kernel functions.

3.1 Representing Lexical Information

The **Bag of Word** (BoW) is a simple representation reflecting the lexical information of the sentence. Each text is represented as a vector whose dimensions correspond to different words, i.e. they represent a boolean indicator of the presence or not of a word in the text. The resulting kernel function is the cosine similarity (or linear kernel) between vector pairs, i.e. $\mathbf{lin_{BoW}}$. In line with (Shawe-Taylor and Cristianini, 2004b) we investigated the contribution of the Polynomial Kernel of degree 2, $\mathbf{poly^2_{BoW}}$ as it defines an implicit space where also feature pairs, i.e. words pairs, are considered.

In the polarity detection tasks, several polarity lexicons have been exploited in order to have useful hints of the intrinsic polarity of words. We adopted MPQA Subjectivity Lexicon[5] (Wilson et al., 2005) and NRC Emotion Lexicon (Mohammad and Turney, 2013): they are large collection of words provided with the underlying emotion they generally evoke. While the former considers only positive and negative sentiments, the latter considers also eight primary emotions, organized in four opposing pairs, joy-sadness, anger-fear, trust-disgust, and anticipation-surprise. We define the **Lexicon Based** (LB) vectors as follows. For each lexicon, let $E = \{e_1, ..., e_{|E|}\}$ be the emotion vocabulary defined in it. Let $w \in s$ be a word occurring in sentence s, with $I(w, i)$ being the indicator function whose output is 1 if w is associated to the emotion label e_i, or 0 otherwise. Then, given a sentence s, each e_i, i.e. a dimension of the emotional vocabulary E, receives a score $s_i = \sum_{w \in s} I(w, i)$. Each sentence produces a vector $\vec{s} \in \mathbb{R}^{|E|}$, for each lexicon, on which a linear kernel $\mathbf{lin_{LB}}$ is applied.

3.2 Generalizing Lexical Information

Another representation is used to generalize the lexical information of each text, without exploiting any manually coded resource. Basic lexical information is obtained by a co-occurrence **Word Space** (WS) 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* (Landauer 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. This approximation supplies a way to project a generic word w_i into the k-dimensional space using $W = U_k S_k^{1/2}$, where each row corresponds to the representation vector $\vec{w_i}$. The result is that every word is projected in the reduced Word Space and a sentence is represented by applying an additive model as an unbiased linear combination. We adopted these vector representations using a linear kernel, as in (Cristianini et al., 2002), i.e. $\mathbf{lin_{WS}}$ and a Radial Basis Function Kernel $\mathbf{rbf_{WS}}$.

In Aspect Category Detection, and more generally in topic classification tasks, some specific words can be an effective indicator of the underlying topic. For instance, in the restaurant domain, the word *tasty* may refer to the quality of food. These kind of word-topic relationships can be automatically captured by a Bag-of-Word approach, but with some limitations. As an example, a BoW representation may not capture synonyms or semantically related terms. This lack of word generalization is partially compensated by the already discussed Word Space. However, this last representation aims at capturing the sense of an overall sentence, and no particular relevance is given to individual words, even if they can be strong topic indicators. To apply a modeling more focused on topics, we manually selected m seed words $\{\sigma_1, \dots, \sigma_m\}$ that we consider reliable *topic-indicator*s, for example *spaghetti* for `food`. Notice that for every seed σ_i, as well as for every word w the similarity function $sim(\sigma_i, w)$ can be derived from the Word Space representations $\vec{\sigma_i}$ and $\vec{w}$, respectively. What we need is a specific seed-based representation reflecting the similarity between topic indicators and sentences s. Given the words w occurring in s, the **Seed-Oriented** (SO) representation of s is an m-dimensional vector $\vec{so}(s)$ whose components are: $so_i(s) = \max_{w \in s} sim(\sigma_i, w)$. Alternatively, as

[5] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon

seeds σ refer to a set of evoked topics (i.e. aspect categories such as `food`) $\Sigma_1, ..., \Sigma_t$, we can define a t-dimensional vector $\vec{to}(s)$ called **Topic-Oriented** (TO) representation for s, whose features are: $to_i(s) = \max_{w \in s, \sigma_k \in \Sigma_i} \;\; sim(\sigma_k, w)$.

The adopted word similarity function $sim(\cdot, \cdot)$ over $\vec{so}(s)$ and $\vec{to}(s)$ depends on the experiments. In the unconstrained setting, i.e. the Word Space Topic Oriented **WSTO** system, $sim(\cdot, \cdot)$ consists in the dot product over the Word Space representations $\vec{\sigma_i}$ and $\vec{w}$. In the constrained case $sim(\cdot, \cdot)$ corresponds to the Wu & Palmer similarity based on WordNet (Wu and Palmer, 1994), in the so called WordNet Seed Oriented **WNSO** system. The Radial Basis Function (RBF) kernel is then applied onto the resulting feature vectors $\vec{to}(s)$ and $\vec{so}(s)$ in the $\mathbf{rbf_{WSTO}}$ and $\mathbf{rbf_{WNSO}}$, respectively.

3.3 Generalizing Syntactic Information

In order to exploit the syntactic information, *Tree Kernel* functions proposed in (Collins and Duffy, 2001) are adopted. Tree kernels exploit syntactic similarity through the idea of convolutions among syntactic tree 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. Many tree representations can be derived to represent the syntactic information, according to different syntactic theories. For this experiment, dependency formalism of parse trees is employed to capture sentences syntactic information. As proposed in (Croce et al., 2011), the kernel function is applied to examples modeled according the Grammatical Relation Centered Tree representation from the original dependency parse structures, shown in Fig. 1: non-terminal nodes reflect syntactic relations, such as NSUBJ, pre-terminals are the Part-Of-Speech tags, such as nouns, and leaves are lexemes, such as *rice::n* and *amazing::j*[6]. In each example, the aspect terms and the covering nodes are enriched with a `a` suffix and all lexical nodes are duplicated by the node `asp` in order to reduce data sparseness. Moreover, prior information derived by the lexicon can be injected in the tree, by duplicating all lexical nodes annotated in the MPQA Subjectivity Lexicon, e.g. the adjective *amazing*, with a node expressing the polarity (`pos`).

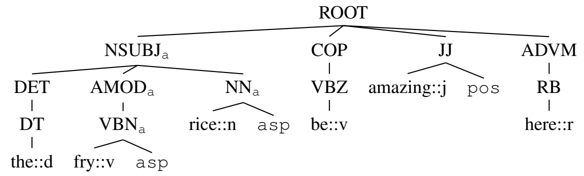

Figure 1: Tree representation of the sentence 1.

Given two tree structures T_1 and T_2, the Tree Kernel formulation is reported hereafter: $TK(T_1, T_2) \;=\; \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2)$ 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 constrained case the Partial Tree Kernel formulation (Moschitti, 2006) is used, i.e. $\mathbf{ptk_{GRCT}}$. In the unconstrained setting the Smoothed Partial Tree Kernel formulation (Croce et al., 2011) is adopted to emphasizes the lexicon in the Word Space, i.e. the $\mathbf{sptk_{GRCT}}$. It computes the similarity between lexical nodes as the similarity between words in the Word Space. So, this kernel allows a generalization both from a syntactic and lexical point of view.

4 Results

In this Section the experimental results of the `UNITOR` system in the four different subtasks of Semeval 2014 competition are discussed. Teams were allowed to submit two different outcomes for each task: *constrained* submissions (expressed by the suffix C in all the tables) are intended to measure systems ability to learn sentiment analysis models only over the provided data; *unconstrained* (expressed by the suffix U in all the tables) submissions allows teams to exploit additional training data. The first two tasks, i.e. ATE and ATP, are defined on the *laptop* and *restaurant* domains, while the last two tasks, i.e. ACD and ACP, are defined for the *restaurant* dataset only.

The unconstrained versions are derived by exploiting word vectors derived in an unsupervised fashion through the analysis of large scale corpora. All words in a corpus occurring more than 100 times (i.e. the *targets*) are represented through vectors. The original space dimensions are generated from the set of the 20,000 most frequent words (i.e. *features*) in the corpus. One dimension describes the Point-wise Mutual Information score between one feature, as it occurs on a left or right window of 3 tokens around a target. Left contexts of targets are distinguished from the right ones, in order to capture asymmetric syntactic behaviors

[6] Each word is lemmatized to reduce data sparseness, but they are enriched with POS tags.

(e.g., useful for verbs): 40,000 dimensional vectors are thus derived for each target. The Singular Value Decomposition is applied and the space dimensionality is reduced to $k = 250$. Two corpora are used for generating two different Word Spaces, one for the laptop and one for the restaurant domain. The Opinosis dataset (Ganesan et al., 2010) is used to build the electronic domain Word Space, while the restaurant domain corpus adopted is the TripAdvisor dataset[7]. Both provided data and in-domain data are first pre-processed through the Stanford Parser (Klein and Manning, 2003) in order to obtain POS tags or Dependency Trees.

A modified version of LibSVM has been adopted to implement Tree Kernel. Parameters such as the SVM regularization coefficient C, the kernel parameters (for instance the degree of the polynomial kernel) have been selected after a tuning stage based on a 5-fold cross validation.

4.1 Aspect Term Extraction

The Aspect Term Extraction task is modeled as a sequential labeling problem. The feature representation described in Section 2.1, where each token is associated to a specific target class according to the IOB notation, is used in the SVM^{hmm} learning algorithm. In the constrained version of the UNITOR system only the training data are used to derive features. In the unconstrained case the UNITOR system exploits lexical vectors derived from a Word Space. Each token feature representation is, in this sense, augmented through distributional vectors derived from the Word Spaces described above. Obviously, the Opinosis Word Space is used in the laptop subtask, while the TripAdvisor Word Space is used in the restaurant subtask. These allow the system to generalize the lexical information, enabling a smoother match between words during training and test phases, hopefully capturing similarity phenomena such as the relation between *screen* and *monitor*.

In Table 1 results in the laptop case are reported. Our system performed quite well, and ranked in 6^{th} and 10^{th} position over 28 submitted systems. In this case, the use of the Word Space is effective, as noticed by the 4 position gain in the final ranking (almost 2 points in F1-measure). In Table 2 results in the restaurant case are reported. Here, the use of Word Space does not give an improvement in the final performance.

Table 1: Aspect Term Extraction Results - Laptop.

System (Rank)	P	R	F1
UNITOR-C (10/28)	.7741	.5764	.6608
UNITOR-U (6/28)	.7575	.6162	.6795
Best-System-C (1/28)	.8479	.6651	.7455
Best-System-U (2/28)	.8251	.6712	.7403

Table 2: Aspect Term Extraction - Restaurants.

System (Rank)	P	R	F1
UNITOR-C (5/29)	.8244	.7786	.8009
UNITOR-U (6/29)	.8131	.7865	.7996
Best-System-C (2/29)	.8624	.8183	.8398
Best-System-U (1/29)	.8535	.8271	.8401

In both cases, we observed that most of the errors were associated to aspect terms composed by multiple words. For example, in the sentence *The portions of the food that came out were mediocre* the gold aspect term is `portions of the food` while our system was able only to retrieve *food* as aspect term. The system is mainly able to recognize single word aspect terms and, in most of the cases, double words aspect terms.

4.2 Aspect Term Polarity

The Aspect Term Polarity subtask has been modeled as a multi-class classification problem: for a given set of aspect terms within a sentence, it aims at determining whether the polarity of each aspect term is *positive*, *negative*, *neutral* or *conflict*. It has been tackled using multi-kernel SVMs in a One-vs-All Schema. In the constrained setting, the linear combination of the following kernel functions have been used: ptk_{GRCT}, $poly^2_{BoW}$ that consider all the lemmatized terms in the sentence, a $poly^2_{BoW}$ that considers only the aspect terms, $poly^2_{BoW}$ of the terms around the aspect terms in a window of size 5, lin_{LB} derived from the Emolex lexicon. In the unconstrained setting the $sptk_{GRCT}$ replaces the ptk counterpart and the rbf_{WS} is added by linearly combining Word Space vectors for verbs, nouns adjective and adverbs. Results in Table 3 show that the proposed kernel combination allows to achieve the 8^{th} position with the unconstrained system in the *restaurant* domain. The differences with the constrained setting demonstrate the contribution of the Word Space acquired from the TripAdvisor corpus. Unfortunately, it is not true in the laptop domain, as shown in Table 4. The use of the Opinosis corpus lets to a performance drop of the unconstrained setting. An error analysis shows that the main lim-

[7] http://sifaka.cs.uiuc.edu/˜wang296/Data/index.html

itation of the proposed model is the inability to capture deep semantic phenomena such as irony, as in the negative sentence "*the two waitress's looked like they had been sucking on lemons*".

Table 3: Aspect Term Polarity Results - Restaurant.

System (Rank)	Accuracy
UNITOR-C (12/36)	.7248
UNITOR-U (8/36)	.7495
Best-System-C (1/36)	.8095
Best-System-U (5/36)	.7768

Table 4: Aspect Term Polarity Results - Laptop.

System (Rank)	Accuracy
UNITOR-C (10/32)	.6299
UNITOR-U (17/32)	.5856
Best-System-C (1/32)	.7048
Best-System-U (5/32)	.6666

4.3 Aspect Category Detection

The Aspect Category Detection has been modeled as a multi-label classification task where 5 categories (*ambience*, *service*, *food*, *price*, *anecdotes/miscellaneous*) must be recognized. In the constrained version, each class has been tackled using a binary multi-kernel SVM equipped with a linear combination of $poly^2_{BoW}$ and rbf_{WNSO}. A category is assigned if the SVM classifiers provides a positive prediction. The anecdotes/miscellaneous acceptance threshold has been set to 0.3 (it has been estimated over a development set) due to its poor precision observed during the tuning phase. Moreover, considering each sentence is always associated to at least one category, if no label has been assigned, then the sentence is labelled with the category associated to the highest prediction.

In the unconstrained case, each class has been tackled using an ensemble of a two binary SVM-based classifiers. The first classifier is a multi-kernel SVM operating on a linear combination of rbf_{WS} and $poly^2_{BoW}$. The second classifier is a SVM equipped with a rbf_{WSTO}. A sentence is labelled with a category if at least one of the two corresponding classifiers predicts that label. The first classifier assigns a label if the corresponding prediction is positive. A more conservative strategy is applied to the second classifier, and a category is selected if its corresponding prediction is higher than 0.3; again this threshold has been estimated over a development set. As in the constrained version, we observed a poor precision in the anecdotes/miscellaneous category, so we increased the first classifier acceptance threshold to 0.3, while the second classifier output is completely ignored. Finally, if no label has been assigned, the sentence is labelled with the category associated to the highest prediction of the first classifier.

Table 5: Aspect Category Detection Results.

System (Rank)	P	R	F1
UNITOR-C (6/21)	.8368	.7804	.8076
UNITOR-U (2/21)	.8498	.8556	.8526
Best-System-C (1/21)	.9104	.8624	.8857
Best-System-U (4/21)	.8435	.7892	.8155

Table 5 reports the achieved results. Considering the simplicity of the proposed approach, the results are impressive. The ensemble schema, adopted in the unconstrained version, is very useful in improving the recall and allows the system to achieve the second position in the competition.

4.4 Aspect Category Polarity

The Aspect Category Polarity subtask has been modeled as a multi-class classification problem: given a set of pre-identified aspect categories for a sentence, it aims at determining the polarity (*positive*, *negative*, *neutral* or *conflict*) of each category. It has been tackled using multi-kernel SVMs in a One-vs-All Schema. In the constrained setting, the linear combination of the following kernel functions has been used: ptk_{GRCT}, $poly^2_{BoW}$ that consider all the lemmatized terms in the sentence, a $poly^2_{BoW}$ that considers only verbs, nouns adjective and adverbs in the sentence, lin_{LB} derived from the MPQA sentiment lexicon. In the unconstrained case the $sptk_{GRCT}$ replaces the ptk counterpart and the rbf_{WS} is added by linearly combining Word Space vectors for verbs, nouns adjective and adverbs.

Again, results shown in Table 6 suggest the positive contribution of the lexical generalization provided by the Word Space (in the $sptk_{GRCT}$ and rbf_{WS}) allows to achieve a good rank, i.e. the 4^{th} position with the unconstrained system in the *restaurant* domain. The error analysis underlines that the proposed features do not capture irony.

Table 6: Aspect Category Polarity Results.

System (Rank)	Accuracy
UNITOR-C (7/25)	.7307
UNITOR-U (4/25)	.7629
Best-System-C (1/25)	.8292
Best-System-U (9/25)	.7278

References

Yasemin Altun, I. Tsochantaridis, and T. Hofmann. 2003. Hidden Markov support vector machines. In *Proceedings of the International Conference on Machine Learning*.

Giuseppe Castellucci, Simone Filice, Danilo Croce, and Roberto Basili. 2013. Unitor: Combining syntactic and semantic kernels for twitter sentiment analysis. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 369–374, Atlanta, Georgia, USA, June. ACL.

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 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. ACL.

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

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 340–348. ACL.

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: Series B, Numerical Analysis*, 2(2):pp. 205–224.

Thorsten Joachims. 1999. *Making large-Scale SVM Learning Practical*. MIT Press, Cambridge, MA.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL'03*, pages 423–430.

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.

Bing Liu. 2007. *Web data mining*. Springer.

Saif Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

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

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML*, Berlin, Germany, September.

Sebastian Pado and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2).

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. ACL.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*.

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

John Shawe-Taylor and Nello Cristianini. 2004a. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA.

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

Janyce M. Wiebe, Rebecca F. Bruce, and Thomas P. O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the ACL on Computational Linguistics*, pages 246–253, Stroudsburg, PA, USA. ACL.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, Vancouver, CA.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting of ACL*, ACL '94, pages 133–138, Stroudsburg, PA, USA. ACL.