ISSN 2499-4553

IJCoL

Italian Journal of Computational Linguistics

Rivista Italiana di Linguistica Computazionale

Volume 1, Number 1
december 2015

Emerging Topics at the First Italian Conference on Computational Linguistics

aA accademia university press

registrazione in corso presso il Tribunale di Trento

Rivista Semestrale dell'Associazione Italiana di Linguistica Computazionale (AILC)

direttore responsabile
Michele Arnese

ISSN 2499-4553
ISBN 978-88-99200-63-3

www.aAccademia.it/IJCoL_01

Accademia University Press
via Carlo Alberto 55
I-10123 Torino
info@aAccademia.it

IJCoL

Volume 1, Number 1
december 2015

Emerging Topics at the First Italian Conference on Computational Linguistics

a cura di
Roberto Basili, Alessandro Lenci,
Bernardo Magnini, Simonetta Montemagni

CONTENTS

Context-aware Models for Twitter Sentiment Analysis

Giuseppe Castellucci*
Università di Roma, Tor Vergata

Andrea Vanzo**
Sapienza, Università di Roma

Danilo Croce†
Università di Roma, Tor Vergata

Roberto Basili‡
Università di Roma, Tor Vergata

Recent works on Sentiment Analysis over Twitter are tied to the idea that the sentiment can be completely captured after reading an incoming tweet. However, tweets are filtered through streams of posts, so that a wider context, e.g. a topic, is always available. In this work, the contribution of this contextual information is investigated for the detection of the polarity of tweet messages. We modeled the polarity detection problem as a sequential classification task over streams of tweets. A Markovian formulation of the Support Vector Machine discriminative model has been here adopted to assign the sentiment polarity to entire sequences. The experimental evaluation proves that sequential tagging better embodies evidence about the contexts and is able to increase the accuracy of the resulting polarity detection process. These evidences are strengthened as experiments are successfully carried out over two different languages: Italian and English. Results are particularly interesting as the approach is flexible and does not rely on any manually coded resources.

1. Introduction

In the Web 2.0 era, people write about their life and personal experiences, sharing contents about facts and ideas. Social Networks became the main place where sharing this information and now represent also a valuable source of evidences for the analysts. This data is crucial in the study of interactions and dynamics of subjectivity on the Web. Twitter[1] is one among these microblogging services that counts more than a billion of active users and more than 500 million of daily messages[2]. However, the analysis of this information is still challenging: Twitter messages are characterized by a very informal language, affected by misspelling, slang and special tokens as *#hashtags*, i.e. special user-generated tags used to contextualize a tweet around specific topics.

Researches focused on the computational study and automatic recognition of opinions and sentiments as they are expressed in free texts. It gave rise to the field of

* Dept. of Electronic Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: castellucci@ing.uniroma2.it
** Dept. of Computer Science, Control and Management Engineering - Via Ariosto 25, 00185 Rome, Italy.
E-mail: vanzo@diag.uniroma1.it
† Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: croce@info.uniroma2.it
‡ Dept. of Enterprise Engineering - Via del Politecnico 1, 00133 Rome, Italy.
E-mail: basili@info.uniroma2.it

1 http://www.twitter.com
2 http://expandedramblings.com/

Sentiment Analysis (SA), a set of tasks aiming at recognizing and characterizing the subjective attitude of a writer with respect to some topics. Many SA studies map sentiment detection in a Machine Learning (ML) setting (Pang and Lee 2008), where labeled data allow to induce a sentiment detection function. In general, sentiment detection in tweets has been generally treated as any other text classification task, as proved by most papers participating to the *Sentiment Analysis in Twitter* task in SemEval-2013, SemEval-2014 and Evalita-2014 challenges (Nakov et al. 2013; Rosenthal et al. 2014; Basile et al. 2014), where specific representations for a message are derived considering one tweet in isolation. The shortness of messages and the inherent semantic ambiguity are critical limitations and make these systems fail in many cases.

Let us consider the message, in which a tweet from `ColMustard` cites `SergGray`:

`ColMustard` : *@SergGray Yes, I totally agree with you about the substitutions! #Bayern #Freiburg*

The tweet sounds like to be a reply to the previous one. Notice how no lexical nor syntactic property allows to determine the sentiment polarity. However, if we look at the entire conversation preceding this message:

`ColMustard` : *Amazing match yesterday!!#Bayern vs. #Freiburg 4-0 #easyvictory*

`SergGray` : *@ColMustard Surely, but #Freiburg wasted lot of chances to score.. wrong substitutions by #Guardiola during the 2nd half!!*

`ColMustard` : *@SergGray Yes, I totally agree with you about the substitutions! #Bayern #Freiburg*

it is easy to establish that a first positive tweet has been produced, followed by a second negative one so that the third tweet is negative as well. Only by considering its context, i.e. the conversation, we are able to understand even such a short message and properly characterize it according to its author and posting time.

We aim at exploiting such a richer set of observations (i.e. conversations or, in general, contexts) and at defining a context-aware SA model along two lines: first, by enriching a tweet representation to include the conversation information, and then by introducing a more complex classification model that works over an entire tweet sequence and not only on a tweet (i.e. the target) in isolation. Accordingly, in the paper we will first focus on different representations of tweets that can be made available to a sentiment detection process. They will also account for contextual information, derived both from *conversations*, as chains of tweets that are *reply-to* the previous ones, and *topics*, built around hashtags. These are in fact topics explicitly annotated by users, such as events (*#easyvictory*) or people (*#Guardiola*). A hashtag represents a wider notion of conversation that enforces the sense of belonging to a community. From a computational perspective, the polarity detection of a tweet in a context is here modeled as a sequential classification task. In fact, both conversation and topic-based contexts are arbitrarily long sequences of messages, ordered according to *time* with the target tweet being the last. A variant of the SVM^{hmm} learning algorithm (Altun, Tsochantaridis, and Hofmann 2003) has been implemented in the KeLP framework (Filice et al. 2015) to classify an instance (here, a tweet) within an entire sequence. While SVM based classifiers allow to recognize the sentiments from one specific tweet at a time, the adopted sequence classifier jointly labels all tweets in a sequence. It is expected to capture patterns within a conversation and apply them in novel sequences through a standard decoding task.

While all the above contexts extend a tweet representation, they are still *local* to a specific notion of conversation. In this work, we also explore a more abstract notion of contexts, e.g. the history of messages from the same user, that embodies the emotional

attitude shown by each user in his overall usage of Twitter. In the above example, `ColMustard` exhibits a specific attitude while discussing about the Bayern Munchen. We can imagine that this feature characterizes most of its future messages at least about football. We suggest to enrich the tweet representation with features that *synthesize* a user's profile, in order to catch possible biases towards a particular sentiment polarity. This is quite interesting as it has been shown that communities behave in a coherent way and users tend to take stable standing points.

This work is an extension of (Vanzo, Croce, and Basili 2014) and (Vanzo et al. 2014). Here, the evaluation in the Italian setting is provided over a subset of the Evalita 2014 Sentipolc dataset (Basile et al. 2014). Moreover, we here provide a deeper evaluation of the contribution of different kernel functions as well as more insights about the phenomena covered by the contextual models.

In the remaining of the paper, a survey of the existing approaches is presented into Section 2. Then, Section 3 provides a description of context-based models: conversation, topic-based and user profiling. The experimental evaluation is presented in Section 4 and it proves the positive impact of social dynamics on the SA task.

2. Related Works

Sentiment Analysis (SA) has been described as a Natural Language Processing task at many levels of granularity. It has been mapped to *document level*, (Turney 2002; Pang and Lee 2004), *sentence level* (Hu and Liu 2004; Kim and Hovy 2004) and at the *phrase level* (Wilson, Wiebe, and Hoffmann 2005; Agarwal, Biadsy, and Mckeown 2009).

The spreading of microblog services, e.g. Twitter, where users post real-time opinions about "everything", poses newer and different challenges. Classical approaches to SA (Pang, Lee, and Vaithyanathan 2002; Pang and Lee 2008) are not directly applicable: tweets are very short and a fine-grained lexical analysis is required. Recent works tried to model the sentiment in tweets by taking into account these characteristics of the data (Go, Bhayani, and Huang 2009; Pak and Paroubek 2010; Davidov, Tsur, and Rappoport 2010; Bifet and Frank 2010; Barbosa and Feng 2010; Kouloumpis, Wilson, and Moore 2011; Zanzotto, Pennaccchiotti, and Tsioutsiouliklis 2011; Agarwal et al. 2011; Croce and Basili 2012; Si et al. 2013; Kiritchenko, Zhu, and Mohammad 2014). Specific approaches and feature modeling are used to improve accuracy levels in tweet polarity recognition. For example, the use of *n*-grams, POS tags, polarity lexicons (Kiritchenko, Zhu, and Mohammad 2014; Castellucci, Croce, and Basili 2015) and tweet specific features (e.g. hashtags, re-tweets) are some of the main properties exploited by these works, in combination with different machine learning algorithms: among these latter, probabilistic paradigms, e.g. Naive Bayes (Pak and Paroubek 2010), or Kernel-based machines, as discussed in (Barbosa and Feng 2010; Agarwal et al. 2011; Castellucci et al. 2014), are mostly adopted. An interesting perspective, where a kind of contextual information is studied, is presented in (Mukherjee and Bhattacharyya 2012): the sentiment detection of tweets is here modeled according to lexical features as well as discourse relations like the presence of connectives, conditionals and semantic operators like *modals* and *negations*. In (Speriosu et al. 2011) and (Tan et al. 2011), social information between users is exploited. (Speriosu et al. 2011) builds a graph of Twitter messages that are linked to words, emoticons and users. Users are connected if they are in a *following* relationship. A Label Propagation (Talukdar and Crammer 2009) framework is adopted to spread polarity label distributions and to classify messages with respect to polarity. The relationships between users constitute a sort of contextual information. Again, in (Tan et al. 2011), user relationships are exploited for the polarity classification of

messages in a transductive learning setting. The main motivation in (Tan et al. 2011) is that "users that are somehow connected may be more likely to hold similar opinions".

Nevertheless, in almost all the above approaches, features are derived only from lexical resources or from the tweet or users, and no contextual information, in terms of other related messages, is really exploited. However, given one tweet targeted, more awareness about its content and, thus, its sentiment, can be achieved by considering the entire stream of related posts immediately preceding it. In order to exploit this wider information, a Markovian extension of a Kernel-based categorization approach is presented in the next section.

3. A Context-aware Model for Sentiment Analysis in Twitter

As discussed in the introduction, contextual information about one tweet stems from various aspects: an explicit conversation, the overall set of recent tweets about a topic (for example a hastag like `#Bayern`), or the user attitude. The heterogeneity of this information requires the integration of different aspects that are heterogeneous. As individual perspectives on the context are independent, i.e. a conversation may or may not depend on user preference or cheer, and they also obey to different notion of analogies or similarity, we should avoid a unified representation for them. We are more likely to derive independent representations and make them interact in a proper algorithmic framework. We thus consider a tweet as a multifaceted entity where a set of vector representations, each one contributing to one aspect of the overall representation, exhibits a specific similarity metrics. This is exactly what Kernel-based learning supports, whereas the combination of different kernels can easily result in a kernel function itself (Shawe-Taylor and Cristianini 2004). Kernels are thus used to capture specific aspects of the semantic relatedness between two messages and are integrated in various machine learning algorithms, such as Support Vector Machines (SVMs).

3.1 Representing Tweets through Different Kernel Functions

Many ML approaches for Sentiment Analysis in Twitter benefits by complex modeling of individual tweets, as discussed in many works (Nakov et al. 2013). The representation we propose makes use of individual kernels as models of different aspects that are made available to a SVM algorithm. In the remaining of this Section, different kernel functions are presented for capturing different semantic and sentiment aspects of the data.

Bag of Word Kernel (BoWK). The simplest kernel function describes the lexical overlap between tweets, thus represented as vectors, i.e. Bag-Of-Words vectors, whose individual dimensions correspond to the different words. Components denote the presence or not of a word in the text and the kernel function corresponds to the *cosine similarity* between vector pairs. Even if very simple, the BoWK model is one of the most informative representation in SA, as emphasized since (Pang, Lee, and Vaithyanathan 2002).

Lexical Semantic Kernel (LSK). Lexical information in tweets can be very sparse. In order to extend the BoWK model, we provide a further representation aiming at generalizing the lexical information. It can be obtained for every term of a dictionary by a Word Space (WS) built according to a Distributional Model (Sahlgren 2006) of lexical semantics. These models have been successfully applied in several NLP tasks, such as Frame Induction (Pennacchiotti et al. 2008) or Semantic Role Labeling (Croce et al. 2010). In this work, we derive a vector representation $\vec{w_i}$ for each word w_i in the vocabulary by exploiting Neural Word Embeddings (Bengio et al. 2003; Mikolov et al. 2013). The result is that every word can be projected in the WS and a vector, i.e. WS vector, for each tweet

is derived through the linear combination of the occurring word vectors (also called *additive linear combination* in (Mitchell and Lapata 2010)). The resulting kernel function is the *cosine similarity* between tweet vector pairs, in line with (Cristianini, Shawe-Taylor, and Lodhi 2002). Notice that the adoption of a distributional approach does not limit the overall application, as it can be automatically applied without relying on any manually coded resource.

User Sentiment Profile Kernel (**USPK**). A source of evidence about a tweet is its author, with his attitude towards some polarities. In general, a person will show similar attitudes with respect to the same topics. Thus, we can think of specific features that should model the users' attitudes given its messages. Let $t_i \in \mathcal{T}$ be a tweet and $i \in \mathbb{N}^+$ its identifier. The *User Profile Context* can be defined as the set of the last tweets posted by the author u_i of t_i: we denote this set of messages as Λ^{u_i}. This information is a body of evidence about the opinion holder, and can be adopted to build a profile on which a further tweet representation can be defined. A tweet t_i is here mapped into a three dimensional vector, i.e. USP vector, $\vec{\mu_i} = \left(\mu_i^1, \mu_i^2, \mu_i^3\right)$, where each component μ_i^j is the indicator of a polarity trend, i.e. *positive*, *negative* and *neutral*, expressed through the conditional probability $P(j \mid u_i)$ for the polarity labels $j \in \mathcal{Y}$ given the user u_i. We can suppose that, for each $t_k \in \Lambda^{u_i}$, its corresponding label y_k is available either as a gold standard annotation or predicted in a semi-supervised fashion. The estimation of $\mu_i^j \approx P(j \mid u_i)$, is a σ-parameterized *Laplace smoothed* version of the observations in Λ^{u_i}:

$$\mu_i^j = \sum_{k=1}^{|\Lambda^{u_i}|} \frac{\mathbb{1}_{\{y_k=j\}}(t_k) + \sigma}{|\Lambda^{u_i}| + \sigma|\mathcal{Y}|} \tag{1}$$

where $\sigma \in \mathbb{R}$ is the smoothing parameter, $j \in \mathcal{Y}$, i.e. the set of polarity labels. A kernel function, in which we are interested in, should capture when two users $u_i, u_j, u_i \neq u_j$ expresses similar sentiment attitudes in their messages. We call this kernel function User Sentiment Profile Kernel (USPK), and it can be computed as the *cosine similarity* between the two vectors $(\vec{\mu_i}, \vec{\mu_m})$. As an example, let us consider a user u_1 whose timeline is composed by 100 messages, whose distribution with respect to the *positive*, *negative* and *neutral* classes is the following: 43 *positive*, 21 *negative* and 36 *neural*. If we adopt the Equation 1 with $\sigma = 1.0$, we obtain three values: $\mu_1^{positive} = \frac{43+1}{100+3} = 0.43$, $\mu_1^{negative} = \frac{21+1}{100+3} = 0.22$, $\mu_1^{neutral} = \frac{36+1}{100+3} = 0.35$. These values can be arranged into a 3-dimensional USP vector, $\vec{\mu_1} = [0.43, 0.22, 0.35]$ whose aim is to capture that u_1 writes with a-priori positive attitude. If another user, e.g. u_2, wrote 325 messages distributed as 145 *positive*, 65 *negative* and 115 *neutral*, it is easy to compute a USP vector $\vec{\mu_2} = [0.45, 0.20, 0.35]$. Then, the kernel operating on $\vec{\mu_1}, \vec{\mu_2}$ will capture that u_1 and u_2 write their messages with similar attitudes, and that they should be treated similarly.

The multiple kernel approach. Whenever the different kernels are available, we can apply a linear combination αBoWK+βLSK or αBoWK+βLSK+γUSPK in order to exploit lexical and semantic properties captured by BoWK and LSK, or user properties as captured by USPK. The combination is still a valid kernel, and can thus be adopted in a kernel-based learning framework.

3.2 Modeling Tweet Contexts in a Sequential Labeling Framework

The User Sentiment Profile Kernel (USPK) can be seen as an implicit representation of the context describing the writer. However, contextual information is usually embodied by the stream of messages in which a target tweet t_i is immersed. Usually, the stream is completely available to a reader. In all cases, the stream gives rise to a sequence on

which a sequence labeling algorithm can be applied: the target tweet is here always labeled within the entire sequence, where contextual constraints are provided by the preceding tweets. In this work we rely on two different types of context: *Conversational context* and *Topical context*. The former is based on the *reply-to* chain. In this case, the entire sequence is built by leveraging the *reply information* available for Twitter statuses, that basically represents a pointer to the previous tweet within the conversation chain. The latter takes into account hashtags that allow to aggregate different tweets around a specific topic specified by the users. Here, a tweet sequence can be derived including the n messages preceding the target t_i that contain the same hashtag set. This is usually the output of a search in Twitter and it is likely the source information that influenced the writer's opinion. A more formal definition of the above contexts is given below.

Definition 1 (Conversational context)
For every tweet $t_i \in \mathcal{T}$, let $r(t_i) : \mathcal{T} \rightarrow \mathcal{T}$ be a function that returns either the tweet to which t_i is a reply to, or $null$ if t_i is not a reply. Then, the *conversation-based context* $\Lambda_i^{C,l}$ of tweet t_i (i.e., the *target tweet*) is the sequence of tweet iteratively built by applying $r(\cdot)$, until l tweets have been selected or $r(\cdot) = null$. In other words, l allows to limit the size of the input context.

An example of conversation-based context is given in Section 1.

Definition 2 (Topical context)
Let $t_i \in \mathcal{T}$ be a tweet and $h(i) : \mathcal{T} \rightarrow \mathcal{P}(\mathcal{H})$ be a function that returns the entire hashtag set $H_i \subseteq \mathcal{H}$ observed into t_i. Then, the *hashtag-based context* $\Lambda_i^{H,l}$ for a tweet t_i (i.e., *target tweet*) is a sequence of the most recent l tweets t_j such that $H_j \cap H_i \neq \emptyset$, i.e. t_j and t_i share at least one hashtag, and t_j has been posted before t_i.

As an example, the following hashtag context has been obtained about `#Bayern`:

`MrGreen` : *Fun fact: #Freiburg is the only #Bundesliga team #Pep has never beaten in his coaching career. #Bayern*

`MrsPeacock` : *Young starlet Xherdan #Shaqiri fires #Bayern into a 2-0 lead. Is there any hope for #Freiburg?*
pic.twitter.com/krzbFJFJyN

`ProfPlum` : *It is clear that #Bayern is on a rampage leading by 4-0, the latest by Mandzukic...*
hoping for another 2 goals from #bayernmunich

`MissScarlet` : *Noooo! I cant believe what #Bayern did!*

`MissScarlet` expresses an opinion, but the corresponding polarity is easily evident only when the entire stream is available about the *#Bayern* hashtag. As well as in a conversational context, a specific context size n can be imposed by focusing only on the last n tweets of the sequence. Once different representations and contexts are available a structured learning-based approach can be applied to Sentiment Analysis. Firstly, we will discuss a discriminative multiclass learning approach adopted to classify tweets without considering the contextual information. Then a sequence labeling approach, inspired by the SVM^{hmm} learning algorithm (Altun, Tsochantaridis, and Hofmann

2003), will be introduced. It will be adopted to label sequence of messages coming both from conversation and hashtag contexts.

3.3 Context-unaware vs. Context-aware Classification

The multiclass approach for a context-unaware classification. A multi-classification schema is applied to detect the polarity of messages. We adopt Support Vector Machines (Vapnik 1998) within a One-Vs-All schema (Rifkin and Klautau 2004). In particular, given a training set D of tweet messages distributed across n classes, n binary classification functions f_p, where n is the number of classes, are acquired through the kernel functions above defined. These binary classifiers are used to decide the polarity of a message t_i, by choosing the class that maximizes the confidence of the classifier, i.e. $\arg\max_{p \in \{pos,neg,neu\}} f_p(t_i)$. This learning model is applied to tweet messages without considering the contexts in which they are immersed.

A sequential labeling approach for a context-aware classification. The sentiment prediction of a target tweet can be seen as a sequential classification task over a context. To this respect, we adopted an algorithm inspired by the *SVM*^{hmm} algorithm (Altun, Tsochantaridis, and Hofmann 2003).

Given an input sequence $\mathbf{x} = (x_1 \dots x_m) \subseteq \mathcal{X}$, where $\mathbf{x}$ is a tweet sequence, e.g. considering a *conversation* or *hashtag* context, and $x_i \in \mathbb{R}^n$ is a feature vector representing a tweet, the model predicts a tag sequence $\mathbf{y} = (y_1 \dots y_m) \in \mathcal{Y}^+$ (with $y \in \Sigma$ and $\|\Sigma\| = l$) after learning a linear discriminant function. The aim of a Markovian formulation of SVM is to make dependent the classification of a tweet x_i from the label assigned to the previous elements in a history of length k, i.e $x_{i-k}, \dots, x_{i-1}$. Given this history, a sequence of k labels can be retrieved, in the form $y_{i-k}, \dots, y_{i-1}$. In order to make the classification of x_i dependent also from the history, we augment the feature vector of x_i introducing a vector of transitions $\psi_{tr}(y_{i-k}, \dots, y_{i-1}) \in \mathbb{R}^l$: it is a boolean vector where the dimensions corresponding to the k labels preceding the target element x_i are set to 1. A projection function $\phi(x_i)$ is defined to consider both the observations, i.e. ψ_{obs} and the transitions ψ_{tr} in a history of size k by concatenating the two representation, i.e.:

$$x_i^k = \phi(x_i; y_{i-k}, \dots, y_{i-1}) = \psi_{obs}(x_i) \,||\, \psi_{tr}(y_{i-k}, \dots, y_{i-1})$$

with $x_i^k \in \mathbb{R}^{n+l}$ and $\psi_{obs}(x_i)$ leaves intact the original feature space. Notice that the vector concatenation is here denoted by the symbol $||$, and that the feature space operated by ψ_{obs} is the one defined by the kernel linear combination as described in Section 3.1. In fact, adopting linear kernels the space defined by the linear combination is equivalent to the space obtained by juxtaposing the vectors on which each kernel operates. More formally, assuming that K is a linear kernel, i.e. the inner product, and x_i, x_j are two instances whose vector representations are $x_{i_a}, x_{i_b}, x_{j_a}, x_{j_b}$, e.g. x_{i_a}, x_{j_a} are Bag-Of-Words vectors and x_{i_b}, x_{j_b} are WS vectors, $K(x_i, x_j) = K(x_{i_a}, x_{j_a}) + K(x_{i_b}, x_{j_b}) = \langle x_{i_a} || x_{i_b}, x_{j_a} || x_{j_b} \rangle$. In this case[3], thus, $\psi_{obs}(x_i) = x_{i_a} || x_{i_b}$.

At training time, we use the SVM learning algorithm implemented in LibLinear (Fan et al. 2008) in a One-Vs-All schema over the feature space derived by ϕ, so that for each y_j a linear classifier $f_j(x_i^k) = w_j \phi(x_i; y_{i-k}, \dots, y_{i-1}) + b_j$ is learned. The ϕ function is computed for each element x_i by exploiting the gold label sequences. At classification

3 Before concatenating, each vector composing the observation of an instance, i.e. $\psi_{obs}(x_i)$, is normalized to have unitary norm, so that each representation equally contributes to the overall kernel estimation.

time, all possible sequences $\mathbf{y} \in \mathcal{Y}^+$ should be considered in order to determine the best labeling $\hat{\mathbf{y}} = F(\mathbf{x}, k)$, where k is the size of the history used to enrich x_i, that is:

$$\hat{\mathbf{y}} = F(\mathbf{x}, k) = \arg\max_{\mathbf{y} \in \mathcal{Y}^+} \{ \sum_{i=1 \ldots m} f_j(x_i^k) \} = \arg\max_{\mathbf{y} \in \mathcal{Y}^+} \{ \sum_{i=1 \ldots m} w_j \phi(x_i; y_{i-k}, \ldots, y_{i-1}) + b_j \}$$

In order to reduce the computational cost, a *Viterbi-like decoding algorithm* is adopted[4] to derive the sequence, and thus build the augmented feature vectors through the ϕ function. In our setting, the markovian perspective allows to induce patterns across tweet sequences helpful to recognize sentiment even for truly ambiguous tweets.

4. Experimental Evaluation

The aim of the following evaluation is to estimate the contribution of the contextual models to the accuracy reachable in different scenarios, whereas rich contexts (e.g. popular hashtags) are possibly made available or when tweets with no context are targeted. Moreover, in order to prove the portability of the proposed approach, we experimented it on two different languages: English and Italian. In the first case, we adopted the *Sentiment Analysis in Twitter* dataset[5] as it has been made available in the *ACL SemEval-2013* (Nakov et al. 2013). Experiments for SA in Italian are carried out over the *Evalita 2014 Sentipolc* dataset (Basile et al. 2014).

Our experiments only require the availability of both conversation and hashtag contexts and these are gathered for both datasets by adopting the Twitter API, given the IDs of the target tweet in the datasets[6]. In the case of the Semeval2013 dataset, only tweets from the training and development datasets are characterized by IDs: we, thus, statically divided the training and development official datasets in 80/10/10, respectively for *Training/Held-out/Test*. As the performance evaluation is always carried out against one target tweet, the multi-classification may be applied when no context is available (i.e. there is no conversation nor hashtag to build the context) or when a rich conversational or topical context is available. Table 1 summarizes the number of tweets available for the *Semeval-2013* dataset. The entire corpus of 10,045 messages is shown in column 1, while columns 2-4 represent the subsets of target tweets for which conversational contexts, topical contexts or both were available, respectively. Conversational contexts are available only for 1,391 tweets (column 2), while topical contexts include 1,912 instances (column 3). Both contexts are available only for 128 tweets.

The Italian Evalita dataset consists of short messages annotated with the `subjectivity`, `polarity` and `irony` classes. We selected those messages annotated with polarity and that were not expressing any ironic content[7]. Again, we were able to gather the contexts only for a subset of this dataset due to cancelation or privacy restrictions. The final data used for our evaluations consists of a training set of $2,445$ messages and a testing set of $1,128$ messages. Table 2 summarizes the number of

4 When applying $f_j(x_i^k)$ the classification scores are normalized through a softmax function and probability scores are derived.

5 `http://www.cs.york.ac.uk/semeval-2013/task2/index.php?id=data`

6 We were able to download only a (still consistent) subset of the messages, as some of them have been deleted or the author changed its privacy settings.

7 We removed the ironic tweets to have similar datasets in English and Italian. In fact, ironic messages would have biased the final evaluations in Italian, making more difficult to interpret the results.

Table 1
Number of annotated messages within the Semeval 2013 Dataset. In parentheses the percentage of messages with respect to the size of the dataset.

Dataset (size)	w/ conv	w/ hashtag	w/ both
Training (8045)	1106 (13.74%)	1554 (19.31%)	100 (1.24%)
Development (1001)	150 (14.98%)	190 (18.98%)	12 (1.20%)
Testing (999)	135 (13.51%)	168 (16.81%)	16 (1.60%)

messages in this dataset, where the subsets of messages characterized by the considered contexts are again emphasized. In both languages, experiments are intended to classify the polarity of a message with respect to the three classes *positive*, *negative* and *neutral*.

Table 2
Number of annotated messages within the Evalita 2014 Sentipolc Dataset. In parentheses the percentage of messages with respect to the size of the dataset.

Dataset (size)	w/ conv	w/ hashtag	w/ both
Training (2445)	349 (14,27%)	987 (40.36%)	80 (3.27%)
Testing (1128)	169 (14.98%)	468 (41.48%)	47 (4.16%)

As tweets are noisy texts, a pre-processing phase has been applied to improve the quality of linguistic features observable and reduce data sparseness. In particular, a normalization step is applied to each post: fully capitalized words are converted in lowercase; reply marks are replaced with the pseudo-token `USER`, hyperlinks by `LINK`, *hashtags* by `HASHTAG` and emoticons by special tokens[8]. Afterwards, an almost standard multi-language NLP chain is applied through the *Chaos* parser (Basili, Pazienza, and Zanzotto 1998). In particular, each tweet, with its pseudo-tokens produced by the normalization step, is mapped into a sequence of POS tagged lemmas. In order to feed the LSK, lexical vectors correspond to a Word Space (WS) derived from a corpus of about 20 million and 10 million of tweets, respectively for English and Italian. Also these messages have been analyzed by applying the same normalization above, and ⟨lemma,pos⟩ pairs are fed in input to the `word2vec`[9] tool. Skip-gram models[10] are acquired from these datasets, resulting in two 250 dimensional vector spaces that are adopted in computing LSK. No existing dataset contains gold standard annotations for tweets belonging to contexts: USPK or the markovian approach would not be applicable. The solution we propose is to create a *semi-supervised Gold-Standard* by acquiring a multi-classifier. In particular, we derive a multi-classifier with the methodology described in Section 3.2 on the available labeled training data with a BoWK+LSK function. We then classify each tweet in contexts with this classifier. This is a noisy but realistic and portable solution across datasets to initialize tweets labels.

Performance scores report the classification accuracy in terms of Precision, Recall and standard F-measure. However, in line with SemEval-2013, we report the $F1Pn$ score as the arithmetic mean between the F_1 of *positive*, *negative* classes, and the $F1Pnn$ score as the mean between of all the involved polarity classes. The multi-class classifiers

8 We normalized 113 well-known emoticons in 15 classes.
9 `https://code.google.com/p/word2vec/`
10 `word2vec` settings are: *min-count=50*, *window=5*, *iter=10* and *negative=10*.

have been acquired with the SVM implementation that can be found in the KeLP (Filice et al. 2015) framework[11]. Also the Markovian sequential labeler has been implemented within KeLP. In the following experiments we adopted different kernel combinations to test the contribution of each kernel. When a kernel is the result of the combination of two or more kernels, the corresponding weights are set to 1 to equally consider their contribution. For example, when adopting the BoWK and the USPK their combination is given by α BoWK + β USPK where $\alpha = \beta = 1$.

Table 3
Results over the Semeval 2013 Twitter Sentiment Analysis Dataset.

	Ctx. size	Positive			Negative			Neutral			F1Pn	F1Pnn
		P	R	F1	P	R	F1	P	R	F1		
BoWK												
multi	-	.746	.661	.701	.478	.620	.540	.733	.736	.735	.621	.659
conv	3	.774	.656	.710	.550	.465	.504	.701	.821	.756	.607	.657
	6	.755	.693	.722	.618	.444	.516	.707	.815	.757	.619	.665
	16	.751	.680	.714	.604	.472	.530	.703	.804	.750	.622	.664
	31	.765	.680	.720	.595	.486	.535	.705	.809	.753	.627	.669
hash	3	.769	.654	.707	.567	.479	.519	.705	.826	.761	.613	.662
	6	.746	.651	.695	.565	.521	.542	.708	.798	.750	.619	.662
	16	.742	.677	.708	.567	.535	.551	.723	.787	.754	.629	.671
	31	.763	.690	.725	.578	.549	.563	.730	.798	.762	.644	.683
BoWK+LSK												
multi	-	.765	.690	.726	.500	.648	.564	.760	.753	.756	.645	.682
conv	3	.773	.703	.736	.603	.535	.567	.731	.811	.769	.652	.691
	6	.770	.708	.738	.584	.514	.547	.732	.806	.767	.642	.684
	16	.780	.705	.741	.591	.528	.558	.730	.811	.768	.649	.689
	31	.772	.716	.743	.603	.535	.567	.732	.800	.764	**.655**	**.691**
hash	3	.770	.708	.738	.563	.500	.530	.741	.815	.776	.634	.681
	6	.757	.693	.723	.579	.514	.545	.730	.806	.766	.634	.678
	16	.756	.705	.730	.578	.549	.563	.736	.787	.761	.647	.685
	31	.770	.682	.723	.577	.577	.577	.732	.800	.764	.650	.688
BoWK+USPK												
multi	-	.769	.669	.715	.481	.634	.547	.747	.755	.751	.631	.671
conv	3	.735	.680	.706	.569	.289	.383	.687	.832	.753	.545	.614
	6	.751	.661	.703	.551	.415	.474	.699	.819	.754	.589	.644
	16	.738	.654	.693	.523	.401	.454	.697	.811	.749	.574	.632
	31	.737	.674	.704	.555	.465	.506	.703	.787	.743	.605	.651
hash	3	.762	.672	.714	.590	.486	.533	.713	.821	.764	.624	.670
	6	.771	.669	.716	.580	.535	.557	.724	.819	.768	.637	.681
	16	.756	.680	.716	.569	.521	.544	.720	.798	.757	.630	.672
	31	.776	.682	.726	.578	.549	.563	.731	.815	.771	.645	.687
BoWK+LSK+USPK												
multi	-	.779	.685	.729	.511	.634	.566	.758	.779	.768	.648	.688
conv	3	.764	.703	.732	.619	.514	.562	.733	.819	.774	.647	.689
	6	.764	.703	.732	.612	.521	.563	.738	.819	.776	.647	.690
	16	.770	.685	.725	.623	.535	.576	.726	.823	.772	.650	.691
	31	.776	.690	.731	.582	.549	.565	.735	.815	.773	.648	.690
hash	3	.772	.690	.729	.588	.542	.564	.734	.815	.772	.646	.688
	6	.759	.693	.724	.591	.528	.558	.726	.802	.762	.641	.681
	16	.755	.693	.722	.581	.556	.568	.732	.791	.761	.645	.684
	31	.753	.700	.726	.596	.570	.583	.736	.787	.761	.654	.690

11 http://sag.art.uniroma2.it/demo-software/kelp/

4.1 Context-aware Classification of Twitter Messages

The experiments have been run to validate the impact of contextual information over generic tweets, independently from the availability of a context. In this case, the entire dataset is used. The different settings adopted are reported in independent rows, corresponding to different classification approaches:

- *multi* refers to the application of the multi-classification of SVM with the One-Vs-All approach, that does not require any context and can be considered as a baseline for the employed kernel combination;
- *conv* refers to the sequential labeler observing the conversation-based contexts. The training and testing of the classifier is here run with different *context size*s, by parameterizing l in $\Lambda_i^{C,l}$;
- likewise, *hash* refers to the sequential labeler observing the topic-based contexts, when hashtags are considered. Different *context size*s have been considered, by parameterizing l in $\Lambda_i^{H,l}$.

When no context is available, both *conv* and *hash* models act on a sequence of length one, and no transition is applied.

Table 3 shows the empirical results over the test set for the English language, while in Table 4 results for the Italian language are reported. The first general outcome is that algorithmic baselines, i.e. context-unaware models that use no contextual information (multi rows) are better performing whenever richer representations are provided. The lexical information provided by the LSK kernel is beneficial as it increases the performance significantly, as well as the user profiling. They are able to provide useful information with all kernels, but the BoWK benefits more from their adoption. English outcomes show that the *negative* and *neutral* classes are more positively influenced by the adoption of contextual models. It seems that the positive label is harder to manage, even if a slight improvement is measured. In many cases the classifiers faced messages for which no sufficient information was available. Let us consider the message "*Got my Dexter fix for the night. Until 2morw night Dexter Morgan*" that is annotated as *positive* in the gold standard and that has no context. All the classifiers predicts the *neutral* class, as no cue exists suggesting that the message is positively biased. The same phenomenon occurs for the message "*Comedy Central made my night tonight*" where the positive attitude is not directly expressed in neither linguistic nor contextual elements. Again, the multiclass and the sequence based classifiers predicts the *neutral* class.

Italian results (Table 4) shows similar trends, with good improvements with respect to all the adopted kernel functions. Again, the BoWK benefits more by the adoption of contextual models, as good increment are measured in both the $F1Pn$ and the $F1Pnn$. This is a clear effect on alleviating data sparsity that is inherent to a BoWK function. When richer kernel are adopted these improvements are less evident, even though the conversation model is able to reach a remarkable score of 69.6 in the $F1Pn$.

Almost all context-driven models provide an improvement with respect to their context-unaware counterpart. Notice that there are two different behaviors in the two languages. In fact, in English the conversation-based models are more reliable, obtaining better results with respect to the hashtag-based context classifiers. In Italian, the opposite situation is observed: the hashtag based models are more effective. In this last setting, we argue that the different availability of conversation and hashtag contexts plays a crucial role. In fact, hashtag contexts in Italian are far more populated with respect to the conversation contexts. In English, the number of messages in a conversa-

Table 4
Results over the Evalita 2014 Sentipolc Dataset.

	Ctx. size	Positive			Negative			Neutral			F1Pn	F1Pnn
		P	R	F1	P	R	F1	P	R	F1		
BoWK												
multi	-	.647	.647	.647	.646	.575	.609	.439	.513	.473	.628	.576
conv	3	.673	.649	.661	.634	.662	.648	.481	.470	.476	.654	.595
	6	.671	.644	.657	.613	.638	.625	.466	.460	.463	.641	.582
	16	.664	.666	.665	.634	.642	.638	.457	.447	.452	.651	.585
	31	.661	.663	.662	.623	.642	.633	.460	.437	.448	.647	.581
hash	3	.708	.616	.659	.630	.670	.649	.479	.507	.493	.654	.600
	6	.696	.638	.666	.655	.670	.662	.476	.507	.491	.664	.606
	16	.712	.671	.691	.697	.651	.673	.503	.590	.543	.682	.636
	31	.708	.652	.679	.694	.683	.688	.494	.553	.522	.684	.630
BoWK+LSK												
multi	-	.701	.707	.704	.686	.601	.641	.475	.560	.514	.672	.619
conv	3	.688	.688	.688	.671	.647	.659	.473	.500	.486	.673	.611
	6	.695	.723	.709	.679	.642	.660	.506	.523	.515	.684	.628
	16	.698	.696	.697	.671	.647	.659	.491	.520	.505	.678	.620
	31	.698	.721	.709	.676	.644	.660	.497	.513	.505	.684	.625
hash	3	.708	.704	.706	.673	.655	.664	.484	.507	.495	.685	.622
	6	.708	.696	.702	.689	.653	.670	.491	.540	.514	.686	.629
	16	.708	.696	.702	.689	.653	.670	.491	.540	.514	.686	.629
	31	.712	.704	.708	.700	.664	.681	.512	.560	.535	.695	**.641**
BoWK+USPK												
multi	-	.682	.611	.645	.616	.608	.612	.474	.543	.506	.628	.587
conv	3	.672	.622	.646	.614	.662	.637	.467	.453	.460	.641	.581
	6	.632	.655	.643	.626	.627	.626	.444	.423	.433	.635	.568
	16	.644	.638	.641	.616	.640	.628	.470	.447	.458	.634	.576
	31	.644	.679	.661	.609	.640	.624	.469	.400	.432	.643	.572
hash	3	.659	.619	.638	.613	.666	.638	.468	.440	.454	.638	.577
	6	.676	.636	.655	.630	.651	.641	.466	.477	.471	.648	.589
	16	.674	.630	.652	.624	.634	.629	.461	.487	.473	.640	.585
	31	.681	.649	.665	.640	.636	.638	.481	.513	.497	.651	.600
BoWK+LSK+USPK												
multi	-	.695	.712	.704	.693	.612	.650	.484	.557	.518	.677	.624
conv	3	.701	.718	.709	.666	.670	.668	.500	.480	.490	.689	.622
	6	.707	.726	.716	.683	.668	.675	.507	.507	.507	**.696**	.633
	16	.688	.707	.697	.678	.659	.669	.488	.493	.491	.683	.619
	31	.683	.710	.696	.681	.625	.652	.481	.520	.500	.674	.616
hash	3	.698	.685	.692	.676	.662	.669	.498	.527	.512	.680	.624
	6	.704	.690	.697	.669	.653	.661	.491	.520	.505	.679	.621
	16	.712	.699	.705	.664	.649	.656	.503	.533	.518	.681	.627
	31	.699	.688	.693	.677	.659	.668	.497	.527	.511	.681	.624

tion or in a hashtag context is similar, making the beneficial effects of the reply-to chain more evident. In fact, the reply-to chain provides a more coherent set of messages in the sequences, but in the Italian setting their effects are alleviated by data scarcity issues.

To further analyze what is happening when considering the contexts, let us consider some classification examples of the multiclass and sequential models. Let us consider, for example, the tweet "*@cewitt94 I'll see :S I have to go to Timmonsville tomorrow afternoon and Brandon's gonna be with me, so I'm not sure.*" It is incorrectly classified as *negative* by the multiclass `BOWK+LSK` classifier. It is, instead, correctly classified as *neutral* by the corresponding conversation sequential model, considering that it is immersed in a context of 3 previous messages whose polarity is *neutral*, *neutral* and *negative*. In

order to further show the importance of the context, let us consider the *positive* message "*@arrington Noticed that joke when you interviewed Reid Hoffman. Better the 2nd time around ;)*". It is characterized only by a conversation context, while it has no hashtag. In this case, the *hashtag* based classifier `BOWK+LSK` predicts a wrong class for that message, i.e. *negative*. The conversation context contains another message whose class is annotated as *positive*: "*This is by far the biggest TechCrunch Disrupt ever with 3,600 attendees. Clearly they're completely falling apart without me :-)*". The conversation-based classifier with `BOWK+LSK` observations is thus able to exploit the contextual information to correctly predict the *positive* class. In the Italian setting we observe similar outcomes. Let us consider the message "*@fioryrus ti do il numero in dm? :)*". This message seems neutral (despite of the smile), and the `BOWK+LSK` multiclassifier predicts such polarity label. In reality this message belongs to a context of 3 messages whose polarity is *neutral*, *neutral* and *positive*. The preceding *positive* message of the target one is thus informing the sequential classifier that, probably, the target message is positive as well.

5. Conclusions

In this work, the role of contextual information in supervised Sentiment Analysis over Twitter is investigated for two different languages, English and Italian. While the task is eminently linguistic, as resources and phenomena lie in the textual domain, other semantic dimensions are worth to be explored. In this work, three types of contexts for a target tweet have been studied. A markovian approach has been adopted to inject contextual evidence (e.g. the history of preceding posts) in the classification of the most recent, i.e. a target, tweet. An improvement of accuracy in the investigated tasks is measured. It is a straightforward result as the approach is free of language specific resources or manually engineered features. The different employed contexts show specific but systematic benefits. In these experiments, users have only been partially explored through the USPK. It seems to express a more static notion of context (i.e. the attitude of the user as observed across a longer period than individual conversations).

Future work will concentrate on the exploration of more sophisticated user models, whose contribution is expected to improve the overall impact. The user sentiment profile adopted in this work, through the USPK similarity, is in fact a first approximation in the direction of exploiting user information during training. Here, we analyzed messages without considering any existing sentiment resource. It could be interesting to adopt a polarity lexicon, e.g. (Mohammad and Turney 2010) or (Castellucci, Croce, and Basili 2015), to strengthen the final system within a context based framework. Moreover, this work explores a notion of context restricted to simple tweet sequences. In Social Networks, information flows according to richer structures, e.g. graph of messages and users: a user is exposed to messages whose streams in the community are very complex, i.e. not linear. Graph-based models of the context are appealing, as they provide more expressive ways to represent the messages and (other) users influencing the writer. This is an interesting direction to be further explored.

References

Agarwal, Apoorv, Fadi Biadsy, and Kathleen R. Mckeown. 2009. Contextual phrase-level polarity analysis using lexical affect scoring and syntactic n-grams. In *Proceedings of the 12th Conference of the EACL*, pages 24–32. Association for Computational Linguistics.

Agarwal, Apoorv, 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*, LSM '11, pages 30–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Altun, Y., I. Tsochantaridis, and T. Hofmann. 2003. Hidden Markov support vector machines. In *Proceedings of ICML*, pages 3–10.

Barbosa, Luciano and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING (Posters)*, pages 36–44. Chinese Information Processing Society of China.

Basile, Valerio, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. 2014. Overview of the evalita 2014 sentiment polarity classification task. In *Proc. of the 4th EVALITA*, pages 50–57.

Basili, Roberto, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. 1998. Efficient parsing for information extraction. In *Proc. of the European Conference on Artificial Intelligence*, pages 135–139.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March.

Bifet, Albert and Eibe Frank. 2010. Sentiment knowledge discovery in twitter streaming data. In *Proceedings of the 13th International Conference on Discovery Science*, DS'10, pages 1–15, Berlin, Heidelberg. Springer-Verlag.

Castellucci, Giuseppe, Danilo Croce, and Roberto Basili. 2015. Acquiring a large scale polarity lexicon through unsupervised distributional methods. In Chris Biemann, Siegfried Handschuh, Andrè Freitas, Farid Meziane, and Elisabeth Mètais, editors, *Natural Language Processing and Information Systems*, volume 9103. Springer International Publishing, pages 73–86.

Castellucci, Giuseppe, Danilo Croce, Diego De Cao, and Roberto Basili. 2014. A multiple kernel approach for twitter sentiment analysis in italian. In *4th International Workshop EVALITA 2014*, pages 98–103.

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

Croce, Danilo and Roberto Basili. 2012. Grammatical feature engineering for fine-grained ir tasks. In Giambattista Amati, Claudio Carpineto, and Giovanni Semeraro, editors, *IIR*, volume 835 of *CEUR Workshop Proceedings*, pages 133–143. CEUR-WS.org.

Croce, Danilo, Cristina Giannone, Paolo Annesi, and Roberto Basili. 2010. Towards open-domain semantic role labeling. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 237–246. Association for Computational Linguistics.

Davidov, Dmitry, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING (Posters)*, pages 241–249. Chinese Information Processing Society of China.

Fan, Rong-En, 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*, 9:1871–1874.

Filice, Simone, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2015. Kelp: a kernel-based learning platform for natural language processing. In *Proceedings of ACL2015: System Demonstrations*, pages 19–24, Beijing, China, July. Association for Computational Linguistics.

Go, Alec, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6.

Hu, Minqing and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Kim, Soo-Min and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, pages 1367–1374, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kiritchenko, Svetlana, Xiaodan Zhu, and Saif M. Mohammad. 2014. Sentiment analysis of short informal texts. *JAIR*, 50:723–762, Aug.

Kouloumpis, Efthymios, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! In Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, *ICWSM*, pages 538–541. The AAAI Press.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mitchell, Jeff and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Mohammad, Saif M. and Peter D. Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of CAAGET Workshop*, pages 26–34.
Mukherjee, Subhabrata and Pushpak Bhattacharyya. 2012. Sentiment analysis in twitter with lightweight discourse analysis. In *Proceedings of COLING*, pages 1847–1864.
Nakov, Preslav, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the SemEval 2013*, pages 312–320, Atlanta, Georgia, USA, June. Association for Computational Linguistics.
Pak, Alexander and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 1320–1326, Valletta, Malta. European Language Resources Association (ELRA).
Pang, Bo and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL2004*, ACL '04, pages 271–279, Stroudsburg, PA, USA. Association for Computational Linguistics.
Pang, Bo and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.
Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.
Pennacchiotti, Marco, Diego De Cao, Roberto Basili, Danilo Croce, and Michael Roth. 2008. Automatic induction of framenet lexical units. In *Proceedings of EMNLP2008*, pages 457–465. Association for Computational Linguistics.
Rifkin, Ryan and Aldebaro Klautau. 2004. In defense of one-vs-all classification. *J. Mach. Learn. Res.*, 5:101–141, December.
Rosenthal, Sara, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proc. SemEval*, pages 73–80. ACL and Dublin City University.
Sahlgren, Magnus. 2006. *The Word-Space Model*. Ph.D. thesis, Stockholm University.
Shawe-Taylor, John and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA.
Si, Jianfeng, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. 2013. Exploiting topic based twitter sentiment for stock prediction. In *ACL (2)*, pages 24–29.
Speriosu, Michael, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. 2011. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, EMNLP '11, pages 53–63, Stroudsburg, PA, USA. Association for Computational Linguistics.
Talukdar, Partha Pratim and Koby Crammer. 2009. New regularized algorithms for transductive learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, pages 442–457, Berlin, Heidelberg. Springer-Verlag.
Tan, Chenhao, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proc. of the 17th International Conference on Knowledge Discovery and Data Mining*, pages 1397–1405, New York, NY, USA. ACM.
Turney, Peter D. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA. Association for Computational Linguistics.
Vanzo, Andrea, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2014. A context based model for sentiment analysis in twitter for the italian language. In *First Italian Conference on Computational Linguistics CLiC-it*, volume 1, pages 379–383.
Vanzo, Andrea, Danilo Croce, and Roberto Basili. 2014. A context-based model for sentiment analysis in twitter. In *Proc. of 25th COLING*, pages 2345–2354. Dublin City University and Association for Computational Linguistics.
Vapnik, Vladimir N. 1998. *Statistical Learning Theory*. Wiley-Interscience.
Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA. Association for Computational Linguistics.
Zanzotto, Fabio M., Marco Pennaccchiotti, and Kostas Tsioutsiouliklis. 2011. Linguistic Redundancy in Twitter. In *Proc. of EMNLP*, pages 659–669, Edinburgh, Scotland, UK., July.