Special Collection: Social and Collaborative Networks and their Applications

INTERNATIONAL JOURNAL OF ENGINEERING BUSINESS MANAGEMENT

International Journal of Engineering Business Management
Volume 9: 1–18

DOI: 10.1177/1847979017744916
journals.sagepub.com/home/enb
SAGE

Dynamic polarity lexicon acquisition for advanced Social Media analytics

Roberto Basili, Danilo Croce and Giuseppe Castellucci

Abstract

Social media analytics tool aims at eliciting information and knowledge about individuals and communities, as this emerges from the dynamics of interpersonal communications in the social networks. Sentiment analysis (SA) is a core component of this process as it focuses onto the subjective levels of this knowledge, including the agreement/rejection, the perception, and the expectations by which individual users socially evolve in the network. Analyzing user sentiments thus corresponds to recognize subjective opinions and preferences in the texts they produce in social contexts, gather collective evidence across one or more communities, and trace some inferences about the underlying social phenomena. Automatic SA is a complex process, often enabled by hand-coded dictionaries, called *polarity lexicons*, that are intended to capture the a priori emotional aspects of words or multiword expressions. The development of such resources is an expensive, and, mainly, language and task-dependent process. Resulting *polarity lexicons* may be inadequate at fully covering Social Media phenomena, which are intended to capture global communities. In the area of SA over Social Media, this article presents an unsupervised and language independent method for inducing large-scale polarity lexicons from a specific but representative medium, that is, Twitter. The model is based on a novel use of Distributional Lexical Semantics methodologies as these are applied to Twitter. Given a set of heuristically annotated messages, the proposed methodology transfers the known sentiment information of subjective sentences to individual words. The resulting lexical resource is a large-scale polarity lexicon whose effectiveness is measured with respect to different SA tasks in English, Italian, and Arabic. Comparison of our method with different Distributional Lexical Semantics paradigms confirms the beneficial impact of our method in the design of very accurate SA systems in several natural languages.

Keywords

Social network analysis, social media analytics, sentiment analysis, natural language processing, machine learning

Date received: 7 April 2017; accepted: 13 October 2017

Motivations

Social and collaborative networks interest more and more the professional, social, as well as public sphere, whereas individual's communication practices are allowed to spread across local and global communities. Notice that this is at the basis of the huge increase in the availability of large-scale Web data sets that specifically emerge and characterize, at the same time, large-scale communities on the Social Media. Although this opens the way to novel professional collaborative practices, it is also a crucial amplifying factor for the complexity of the underlying network structures, fostering new research opportunities.

One major challenge in modern social network analysis is the pervasive role played by unstructured data that characterize the nature and content of the interactions. In general, modeling precise predictions in complex networks

Department of Enterprise Engineering, University of Roma Tor Vergata, Rome, Italy

Corresponding author:
Roberto Basili, Department of Enterprise Engineering, University of Roma, Tor Vergata, Via del Politecnico 1, 00133 Roma, Italy.
Email: basili@info.uniroma2.it

always rely on strong abstractions about the interactions observable among individuals (i.e. nodes of the network). Binary links among nodes are a precise mathematical notion able to support dynamic models about a variety of network phenomena, such as the emergence of independent communities or the spreading of information across the networks. However, social networks, such as Web communities or Interest groups, are not artificial ones but are made of people and thus they are much more complex. The nature and depth of the interactions among members of the network live in a variety of semantic dimensions characterized by the content of the exchanged messages. Although messages are mainly textual, their content is not explicit and machine-readable in some standard formalism. If no account is provided for the content of individual messages, any analysis about their interactions and the way these influence the entire network remain too vague or even arbitrary.

For this reason, a variety of current studies concentrate on the ways linguistic content propagates across social networks and how this can be studied to capture and reuse dynamic core network properties for a variety of predictive tasks.[1–5] These include detecting and predicting habits, trends, and expectations within the social socioeconomic systems emerging from the Web.[6] As trends, preferences, and expectations emerge from the subjective perceptions of members of a social network, the analysis of produced text for the recognition of phenomena inside this subjective sphere, including sentiments and emotions, is a strictly necessary activity.

Opinion mining (OM)[7] specifically focuses on such dimension. It aims at tracking the opinions expressed in texts with respect to specific topics, for example, products or people. In particular, sentiment analysis (SA) deals with the problem of deciding whether a portion of text, for example, a sentence or a phrase, is expressing a polarity trend toward specific feeling. It is clear that OM and SA have a huge impact over the user-generated contents that are typical in blogs and microblogs.

Opinionated lexicons and their social dynamics

In SA, polarity lexicons are special purpose dictionaries, listing positively and negatively polarized words that help in characterizing the text where they appear with respect to the attitude of the writer. They are defined to support the development of automatic systems that match terms or phrases in the incoming texts to decide the polarity of the overall text.[8,9] In these resources, entries are associated with their *prior* polarity, that is, whether their uses tend to evoke a positive or negative sentiment. For example, "good" can be associated with a prior positive sentiment in contrast to "sad," considered negative in every domain. These lexicons are often hand-compiled, as by Stone et al.[10] or by Hu and Liu,[11] and they reflect the polarity of individual words outside the contexts in which they tend to occur. However, from a linguistic point of view, a priori membership of words to polarity classes can be considered too restrictive, as sentiment expressions are often context-dependent, for example, the occurrences of the word *mouse* are mostly neutral in the consumer electronics domain, as an example consider the review "Whether you're blowing chunks out of opponents or demolishing those TPS reports, it's essential that you have the right mouse under your hand," while it can be negatively biased in a restaurant domain, as an example consider the portion of a restaurant review "Did he want extra cheese? Horrifying moment passerby snaps mouse dining on crumbs at Pizza Hut restaurant hours before it opened." Accounting for topic-specific phenomena would require manual revisions of the lexicon. Moreover, these resources exist for English, but they are less common in other languages: compiling a lexicon for a new language represents a very expensive process.

The complexity and costs associated with the development of annotated resources are not trivial for inductive approaches to SA. In line with other natural language semantic tasks, semisupervised approaches can be applied to integrate unsupervised (such as distributional analysis of large text collections) and supervised processes (e.g. support vector machine (SVM)) in order to increase applicability and reduce costs. This integration shows performances comparable with purely supervised algorithms with much smaller training data sets. Examples are lexical generalization as promoted by the distributional models[12] (DMs) or distant supervision as applied in the study by Mintz et al.[13] for relation extraction.

In this article, we promote a semisupervised perspective to SA in Social Media, by applying an unsupervised process for the acquisition of sentiment lexicons. These can be then adopted within supervised language learning systems in order to leverage on prior polarity information of individual words.

The proposed approach is based on DMs of lexical semantics. These allow to represent both words and sentences into high-dimensional geometrical spaces where it is possible to approximate a sort of semantic equivalence between them. As an example, in the technique known as latent semantic analysis (LSA), introduced by Landauer and Dumais,[12] words and texts can be represented into the same geometrical space, the so-called dual space.

As entire sentences can be clearly related to a given polarity, a classifier can always be trained in the document/term spaces and used to transfer sentiment information from sentences to words. Specifically, a polarity classifier is trained by observing sentences and it is used to classify words to populate the polarity lexicon. Annotated messages are derived from Twitter (http://www.twitter.com) and their polarity is determined by simple heuristics. It means that words in specific domains can be related to sentiment classes by looking at their semantic closeness to emotionally biased sentences. The resulting approach

is highly applicable, as the DM can be acquired without any supervision, and the provided heuristics do not have any bias with respect to languages or domains. The above methodology enables the acquisition of a polarity lexicon almost in any language and in any domain with a limited human effort.

In this work, we demonstrate the effectiveness and generality of our methodology by acquiring polarity lexicon in multiple languages, that is, English, Italian, and Arabic. Moreover, we will provide an extensive analysis aiming at verifying whether different distributional methods, such as LSA,[12] word spaces (WSs),[14] and neural word embeddings,[15] can capture different aspects of the polarity of individual words. We will provide evaluations over data sets coming from largely participated international benchmarks, such as the Association for Computational Linguistics SA in Twitter challenges. SemEval[16,17] data sets will be adopted or in English. The Evalita (http://www.evalita.it) Sentipolc[18] data will be adopted for the Italian language polarity lexicon. Finally, the Arabic language polarity lexicon will be measured against the recently released Arabic Sentiment Twitter (AST) data set.[19]

In the rest of this article, related works are discussed in section "Related work." Section "Polarity lexicon generation through distributional approaches" presents the proposed methodology, while section "Polarity lexicons acquisition" describes the experimental evaluations. Finally, conclusions and future works are discussed in section "Conclusions."

Related work

Polarity lexicons have been seen as fundamental resources both for the manual inspection of lexical and sentiment phenomena and for the acquisition of statistical sentiment and emotional models. Their appearance can be dated back to the 60s with the work of Stone et al.[10] It is worth noting that during the decades, a plethora of techniques has been developed by the researchers to compile such lexicons. We can point out three main methodologies and areas for the acquisition of polarity lexicons, that is, *manually annotated lexicons*, *lexicons acquired over graphs*, and *corpus-based lexicons*. The three areas can be thought of three different basic methodologies, where, on the one hand, the aim is to automatize as much as possible the process of lexicon acquisition; on the other hand, the aim is to exploit different semantic/sentiment information between words (as for example, relationships in graphs or co-occurrences) to improve the lexicon quality. In the following, the main works in these three areas are pointed out.

Manually annotated lexicons

Earlier works are based on manual annotations of terms with respect to emotional categories. For example, in the study by Stone et al.,[10] sentiment labels are manually associated with 3600 English terms. In the study by Hu and Liu,[11] a list of positive and negative words is manually extracted from customer reviews. The MPQA Subjectivity Lexicon (SBJ)[9] contains words, each with its prior polarity (positive or negative) and discrete strength (strong or weak). The National Research Council Canada Emotion Lexicon[20] is composed of frequent English nouns, verbs, adjectives, and adverbs annotated through Amazon Mechanical Turk with respect to eight emotions (e.g. joy, sadness, trust) and sentiment. However, the manual development and maintenance of lexicons may be expensive, and coverage issues can arise.

Lexicons acquired over graphs

Graph-based approaches exploit an underlying semantic structure that can be built upon words. In the study by Esuli and Sebastiani,[21] the WordNet[22] synset glosses are exploited to derive three scores describing the positivity, negativity, and neutrality of the synsets through a PageRank-style algorithm. The work of Rao and Ravichandran[23] generates a lexicon through a graph label propagation process. Each node in the graph represents a word. Each weighted edge encodes a relation between words derived from WordNet.[22] The graph is constructed starting from a set of manually defined seeds. The polarity for the other words is determined by exploiting graph-based methods.

Corpus-based lexicons

Statistics-based approaches are more general as they mainly exploit corpus processing techniques. For example, Turney and Littman[8] proposed a minimally supervised approach to associate a polarity tendency with a word by determining whether it co-occurs more with positive words than negative ones. More recently, Zhang and Singh[24] proposed a semisupervised framework for generating a domain-specific sentiment lexicon. Their system is initialized with a small set of labeled reviews, from which segments whose polarity is known are extracted. It exploits the relationships between consecutive segments to automatically generate a domain-specific sentiment lexicon. In the study by Kiritchenko et al.,[25] a minimally supervised approach based on Social Media data is proposed by exploiting emotion evoking words, such as hashtags or emoticons, that are related to positivity and negativity, for example, #happy, #sad, ☺, or ☹. They compute a score, reflecting the polarity of a target word, through a pointwise mutual information-based measure between the target and the words evoking emotions. In the study by Saif et al.,[26] word contexts are adopted to generate sentiment orientation for words. In particular, the sentiment of context words, available in an already built lexicon, is shown to contribute in deriving the sentiment orientation of a target word. As a result, the so-called *SentiCircle* is derived for each target word by considering the contexts in which they

appear. Among their advantages, corpus-driven methods are appealing for the acquisition of sentiment lexicons as they can be applied to input texts even when no sense disambiguation has been (or can be) applied: in this case, polarity is an emerging property of all lexical items in a text and coverage is a major concern. The approach here presented can be seen as more general, as it does not rely on any existing lexicon, but it could be used to build a *SentiCircle*.

Polarity lexicon generation through distributional approaches

In order to rely on comparable representations for words and sentences to transfer sentiment information from the former to the latter, DMs of lexical semantics are exploited. DMs are intended to express semantic relationships between lexical entries, mainly by looking at the words usage. The foundation for these models is the *Distributional Hypothesis*,[27] that is, words that are used and occur in the same "contexts" tend to have similar meanings. A context is here a set of words that appear in the neighborhood of a target word of interest. In this sense, if two words share many contexts, then they can be considered somehow similar. Although different ways for modeling the semantics of words exist, they all derive vector representations for words from more or less complex processing stages of large-scale text collections.

This kind of approaches is effective as it enables the estimation of semantic relationships in terms of vector similarity. From a linguistic perspective, such vectors allow to geometrically model some aspects of lexical semantics and to provide a useful way to represent this information in a machine-readable format. Distributional methods can model different semantic relationships, for example, *topical* similarities (if vectors are built considering the occurrence of a word in documents) or *paradigmatic* similarities (if vectors are built considering the occurrence of a word in the context of another word[14]). In such models, words like *run* and *walk* are close in the space, while *run* and *read* are projected in different subspaces. Here, we concentrate on DMs that are mainly adopted to model *paradigmatic* relationships, as we are more interested in capturing phenomena of synonymy, that is, when two words can be substituted in a sentence without significantly changing its meaning.

Word representations for lexical semantics

Two main families for the acquisition of distributional representations can be pointed out: *count*-based methods, where the co-occurrences between words are considered[14] and *prediction*-based approaches, where word representations are acquired through a supervised learning setting and correspond to distributions useful to trigger lexical prediction tasks (e.g. lexical substitution). Notice that a *word* here should be considered as a generic term that can potentially indicate a simple token, a stem, or a lemma. In this article, we will preprocess a text to extract morphological and grammatical information, and target words here correspond to (*lemma, part of speech* (pos)) pairs. In both approaches, the distributional hypothesis is exploited but with different methodologies. In this section, we briefly review the LSA[12] model and the Skip-gram model.[15] Leaving aside the computational aspects, these *semantic spaces* give rise to word representations that have been used traditionally in learning algorithms to reduce data sparseness and to obtain better generalization capability of the learned functions.[28–31] An in-depth comparison of these methods is discussed in the work of Baroni et al.[32]

Counting co-occurrences: The LSA approach

In a word-based *count* co-occurrence model, contexts correspond to all words appearing in an n-position window to the left and to the right of a target word. In the sentence,

> *Counting methods depend on the occurrence of words in documents*

the target *words* has a two-position window including the four words $con(word) = \{occurrence, of, in, documents\}$. Here, n is the parameter that extends or restricts the window and allows the resulting space to capture different semantic aspects of words. A word-by-context matrix is built by counting the co-occurrences of words and its contexts in a corpus. Each row of this matrix represents a target word, while each column represents a context, that is made of other words, co-occurring with the target one. (Often in the construction of the matrix, left and right contexts of a target word t are separated in sets $con^l(t)$ and $con^r(t)$ for which every word w describes t through two counts, i.e. $|\{con^l(t)|w \in con^l(t)\}|$ and $|\{con^r(t)|w \in con^r(t)\}|$.) The obtained row vectors provide weights for each pair in the dictionary of all target words t and all context words w: this corresponds to the word-by-context matrix $\mathbf{M}$. $\mathbf{M}$ is a first estimation of the semantic relationships between every t and w. However, a further processing step is often applied: the LSA[12] technique is adopted to acquire meaningful generalization of this lexical model. It can be seen as a variant of the principal component analysis idea applied to $\mathbf{M}$. LSA finds the best lower dimensional approximation of the original $\mathbf{M}$, in the sense of minimizing the global reconstruction error, by projecting data along the directions of maximal variance. It captures term (semantic) dependencies by applying a matrix decomposition process called *Singular Value Decomposition* (SVD).[33] The original word-by-context matrix $\mathbf{M}$ is transformed into the product of three new matrices: U, S, and V so that S is diagonal and $\mathbf{M} = USV^T$. Matrix $\mathbf{M}$ is then approximated by $M_d = U_d S_d V_d^T$ in which only the first d columns of U and V are used, and only the first d greatest singular values of

M are considered. The SVD approximation (notice that usually the dimensionality d of the resulting space is much smaller than the size of the context word vocabulary, as usually d is in the $[50 - 500]$ range, while the vocabulary is always made of about 25 K words) supplies a way to project term vectors into the d-dimensional space using $Y_{\text{terms}} = U_d S_d^{1/2}$. These d newly derived features may be thought of as artificial concepts, each one representing an emerging meaning component as a linear combination of many different context words. Notice how sentences can be represented as linear combinations of the word vectors they are made of. This property will allow to adopt documents (or sentences as micro-documents made of just one sentence) for the acquisition of new information about words, by exploiting the relationships that are made available in the d-dimensional space.

Predicting words through vector representations: The Skip-gram model

Prediction-based word vectors have been recently proposed, as an alternative to count-based methods.[32] They mostly rely on the development of more or less complex neural networks, whose aim is to learn a language model.[34] These methods have been successfully applied to different problems according to the renewed interests around the neural networks inspired by the deep learning methodology. In the study by Mikolov et al.,[15] a very efficient model is proposed for deriving these representations, which are able to capture both syntactic and semantic properties.[15] Two main neural network architectures are discussed by Mikolov et al.,[15] the *Contextual Bag of Word* (BoW) and the *Skip-gram* models. The former models the relationship between a context (input of the network) and its target word (output of the network): In other words, given a representation of all words in a given window around a target position (the context), the network predicts the best target word t. In this way, the vectors of words w cooperate to estimate the most likely word t.

In this article, we will adopt the Skip-gram model defined in the same work.[15] It models the inverse task, as it tries to predict the context $w_{t-c}w_{t-c+1}...w_{t-1}w_{t+1}...w_{t+c}$ of a target word w_t, given the representation for w_t. The input layer of this neural network is fed with the target word vector representation. The triggered network (accomplishing the *forward* step) outputs a multinomial probability distribution over the vocabulary in the output layer. This is used to derive the context words of the target. The network is trained by adjusting the weights of both the neural network and the word vectors, in order to optimally explain the relationships between target words and contexts as observed in a very large training corpus. More formally, given the output layer modeling the multinomial distribution over the vocabulary, the average log probability is defined as the training objective function

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-c \le j \le c \\ j \neq 0}} \log \quad p(w_{t+j}|w_t) \tag{1}$$

where c is the context size, w_{t+j} is a word in the context of w_t, and the probability in the log term is computed through a softmax function. Equation (1) is thus optimized during training through backpropagation, and an efficient formulation is obtained by applying hierarchical softmax and negative sampling.[15]

Acquiring polarity lexicons in word networks within distributional spaces

Despite the specific algorithm used for the acquisition of the vectors underlying the WS, all the above approaches allow to derive a projection function $\Phi(\cdot)$ for a (target) word from a dictionary into a metric space. The d-dimensional vector representation for a generic word $w_k \in \mathbb{W}$ is obtained through $\vec{w}_k = \Phi(w_k)$. Notice that we can exploit geometrical regularities in the vector space to determine the prior sentiment for words. Our assumption is that polarized words are neighbors in a word graph, that is, they lie in specific subspaces. Let us consider, for example, Figure 1 where a two-dimensional projection of word embeddings is obtained by computing the similarity between a target word (adequate::j; j is a marker to indicate the adjective category) and the other words represented in the space. One can notice that words with the same polarity are near in the space according to the cosine similarity metrics (the value indicated in Figure 1) that is they lie in specific subspaces. However, in the underlying semantic space, opposite polarity words are often similar too, as they tend to share the same contexts in a corpus, thus resulting in similar word vector representations. Notice that in Figure 1, opposite polarity adjectives (inadequate::j or insufficient::j) are similar to their antonyms (inadequate::j). By construction, the WSs built according to the Distributional Hypothesis is not able to distinguish between contexts of different polarities.

However, as a large set of documents and words represented in the same space is available, we can try to detect specific subspaces where polarity is preserved. The final aim is to leverage on the DMs because of their ability to represent both sentences and words in the same space. In other words, we can hope to establish arcs between words only if these lie in subspaces where polarity is homogeneous. In the following, we discuss how observable sentence polarity is carrier of useful information about such subspaces: these can be expected to preserve word and sentence polarity as well. When transferred to single words, polarity information will help in confirming or rejecting high similarity arcs connecting opposite polarity words and adjust misleading similarities.

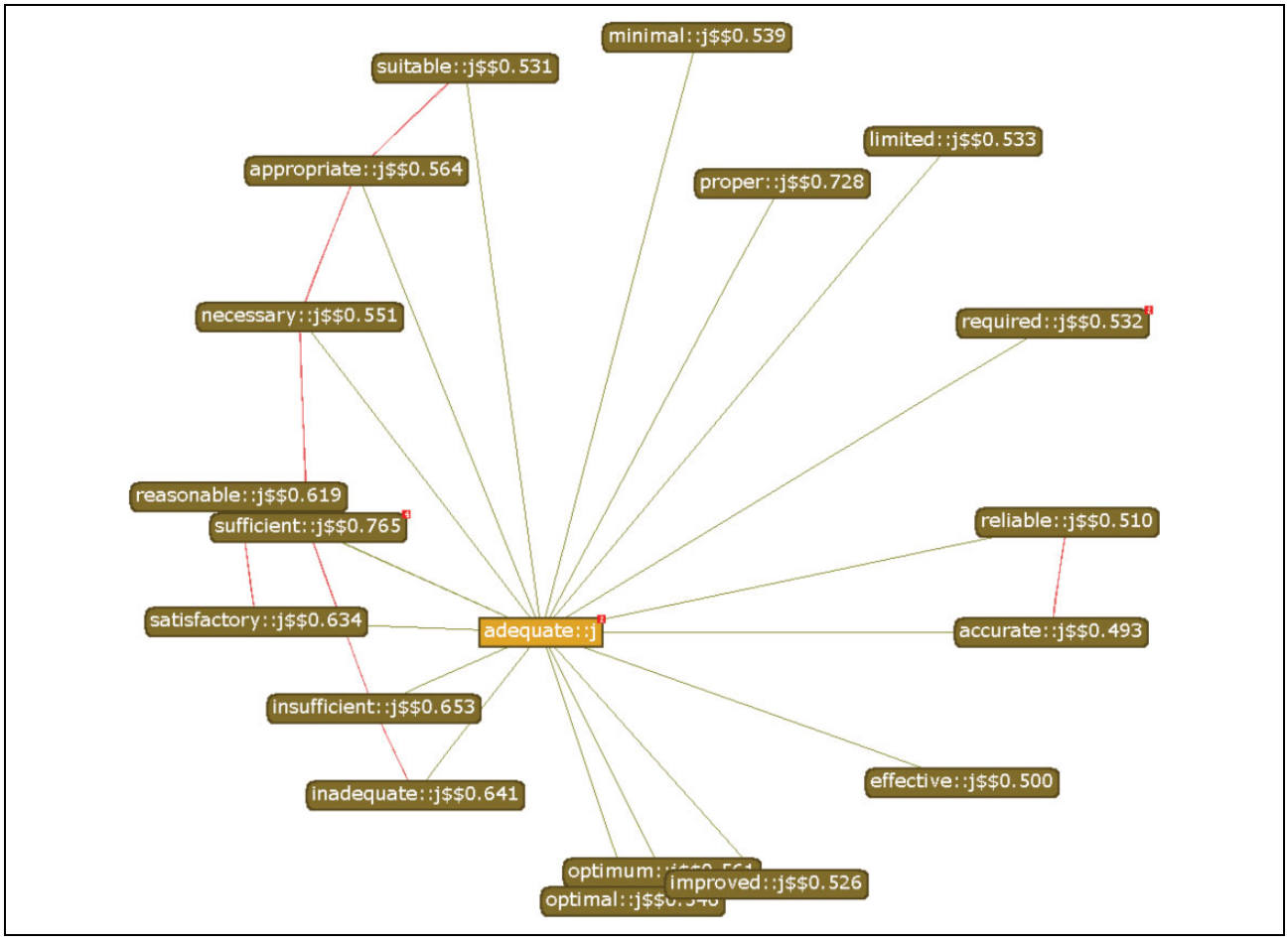

Figure 1. A word graph generated in the neighborhood of the word adequate, that is, a positive polarity word. As emerge from the network, several other words with opposite polarity (such as insufficient::j or inadequate::j) are generally near to adequate::j in the WS, according to the similarity measure, such as the cosine similarity (reported in brackets). Notice how several neutral words are also present, such as necessary::j or minimal::j. WS: word space.

Lexicon generation through classification

The semantic similarity (that is the closeness between words, as established in the originating DM space) does not completely reflect emotional similarity. Sentiment or emotional differences between words must be captured into representations that are able to coherently express the underlying sentiment. In this perspective, we promote to acquire a discriminant function using DM-based representations as a source. Let us consider a space $\mathbb{R}^d$ where a given geometrical representation for a possibly large set of annotated examples can be derived. In general, a discriminative linear classifier can be seen as a separating hyperplane $\theta \in \mathbb{R}^d$ used to classify new examples from the same space into distinct classes. Notice that the parameters of θ, in particular the individual components θ_i, correspond to a specific ith dimension, that is, feature, whose numerical value (the weight) depends on the annotated examples. In a classification setting, the magnitude of each θ_i reflects the importance of the feature i with respect to a target phenomenon, that is, the target classes to which instances in the space should be assigned. In this sense, when applied to Distributional Lexical Semantic vectors, a linear classifier is expected to learn selectively the dimensions (i.e. regions of the space $\mathbb{R}^d$) useful to discriminate examples with respect to the individual target classes (in our case, the sentiment categories such as *positive*, *negative*, and *neutral*). Classes reflecting the sentiment expressed by words should be mapped by one such classifier into those subspaces better modeling the associations of source examples with sentiment classes. In this perspective, training examples could be gathered as any set of words $w_i \in \mathbb{W}$ whose associated polarity is known a priori and suitable for training the classifier. In fact, given such set of seed words w_k^{seed} (each assigned to a polarity class) and their projection in the WS model $\vec{w}_k^{\text{seed}} = \Phi(w_k^{\text{seed}})$, it is straightforward for acquiring a linear classifier. The outcome will correspond to dimensions in $\mathbb{R}^d$ able to characterize the different polarities. In this way, classification corresponds to the transfer of knowledge regarding sentiment from the seed words to the remaining words.

Unfortunately, a number of limitations affects this view. First, the definition and annotation of seed words could be expensive and certainly not portable across natural languages. Second, lexical items do change emotional flavor across domains, and the knowledge embodied by the seed lexicons may not generalize when different domains are faced. Notice that selecting lexical seeds is not the only possible solution for training a polarity lexicon classifier as the nature of DMs can be emphasized. The vector representations of sentences and words lie in the same distributional space, where closeness can be established between sentences, texts, as well as individual words. In this perspective, entire sentences can be seemingly adopted as source of evidences for the training of the classifier: notice how these sentences embody a specific sentiment in a more explicit (and unambiguous) manner than words. For example, sentences including strong sentiment markers can be gathered in a rather cheap manner, thus providing a large-scale seed resource. The training of the classifier over sentences and the availability of similarity metrics among sentences and words allow to transfer the polarity from a limited pool of sentences to large-scale lexicons. The training process detects the regions of the space that are strongly related to specific sentiment classes, and the resulting classifier can be used to emphasize them across the lexicon.

In more detail, we have words $w_k \in \mathbb{W}$ and their vector representation $\vec{w}_k \in \mathbb{R}^d$ obtained by projecting them in a WS, that is, $\vec{w}_k = \Phi(w_k)$. We also have a training set $\mathbb{T}$, including sentences associated with a polarity class. The projection of an entire sentence in the space $\mathbb{R}^d$ is carried out as a simple linear combination of vectors of words occurring in the sentence. For each sentence $t \in \mathbb{T}$, we derive the vector representation $\vec{t} \in \mathbb{R}^d$ by combining all the word vectors involved by sentence t, that is

$$\vec{t} = \sum_{w_i \in t} \Phi(w_i)$$

It is one of the simpler, but still expressive, methods that is used to derive a representation that accounts for the underlying meaning of a sentence, as discussed by Landauer and Dumais.[12] Having projected an entire sentence in the space, we can find all the dimensions of the space that are related to a sentiment class. Sentence representations are fed to a linear learning algorithm that induces the discriminant function f expected to capture the sentiment-related subspaces by properly weighting each dimension i of the original space. The lexicon is finally generated by applying f to the entire $\mathbb{W}$. As we deal with multiple sentiment classes, f corresponds to m distinct real-valued functions $(f_1, \ldots, f_m)$, one for each sentiment class. Each word $w_k \in \mathbb{W}$ is classified with all f_i, thus receiving m distinct scores s_i^k, each one reflecting the classifier confidence in the membership of w_k to the ith class. Each s_i^k is normalized through a softmax function obtaining the final polarity score

$$o_i^k = \frac{e^{s_i^k}}{\sum_{j=1}^{m} e^{s_j^k}}$$

As a consequence, each word w_k can be represented by two distinct representations. One vector expresses its distributional (semantic) properties, that is, $\vec{w}_k = \Phi(w_k)$, and a second vector, that is, $\vec{o}^k$, expresses its scores across sentiment classes.

Regarding the second vector, such representation is different with respect to other works concerning the definition of polarity lexicons where, for example, polarity is represented with one numeric value: In the study by Kiritchenko et al.,[25] the polarity of a word is represented with a real number ranging between $[-1, 1]$: -1 indicates strong negativity, 1 indicates strong positivity, and the values in between define the shades of the polarity (with 0 indicating neutrality). In this work, we adopted a three-dimensional representation, where each value indicates the degree with respect to a polarity dimension. We believe that such apparently redundant representation is more appropriate to express the polarity of words whose contribution depends on their context: As an example, the adjective *high* when modifying a noun such as *cost* suggests a negative polarity, while this turns to positive when modifying the noun *performance*. Our approach would express this case through three values, for example, 0.4 for positivity, 0.4 for negativity, and 0.2 for neutrality, meaning that this word can express both polarities. On the contrary, a singly score risks to mislead such case with a neutral word.

Generating a training data set through emoticons

An annotated data set of sentences $\mathbb{T}$ is needed to acquire a linear classifier that emphasizes specific subspaces. Although different data sets of such kind exist, our aim is to use a general methodology that can enable the use of this technique in different domains or languages. We are going to use (language independent) heuristic rules to select sentences by exploring Twitter messages and the emoticons that can be found in them. The method is based on a distant supervision approach.[35] In order to derive messages belonging to the positive or negative classes, we select Twitter messages whose last token is a smile either positive, for example, ☺ or :D or negative, for example, ☹ or :-(. Neutral messages are filtered by looking at those messages that end with a URL; as in many cases, these are written by newspaper accounts and they use mainly nonpolar words to announce an article. In order to improve the quality of the data set, we further filter out those messages that contain elements from other classes: if a message ends with a positive smile and it contains either a negative smile or a URL it will be discarded. The process of

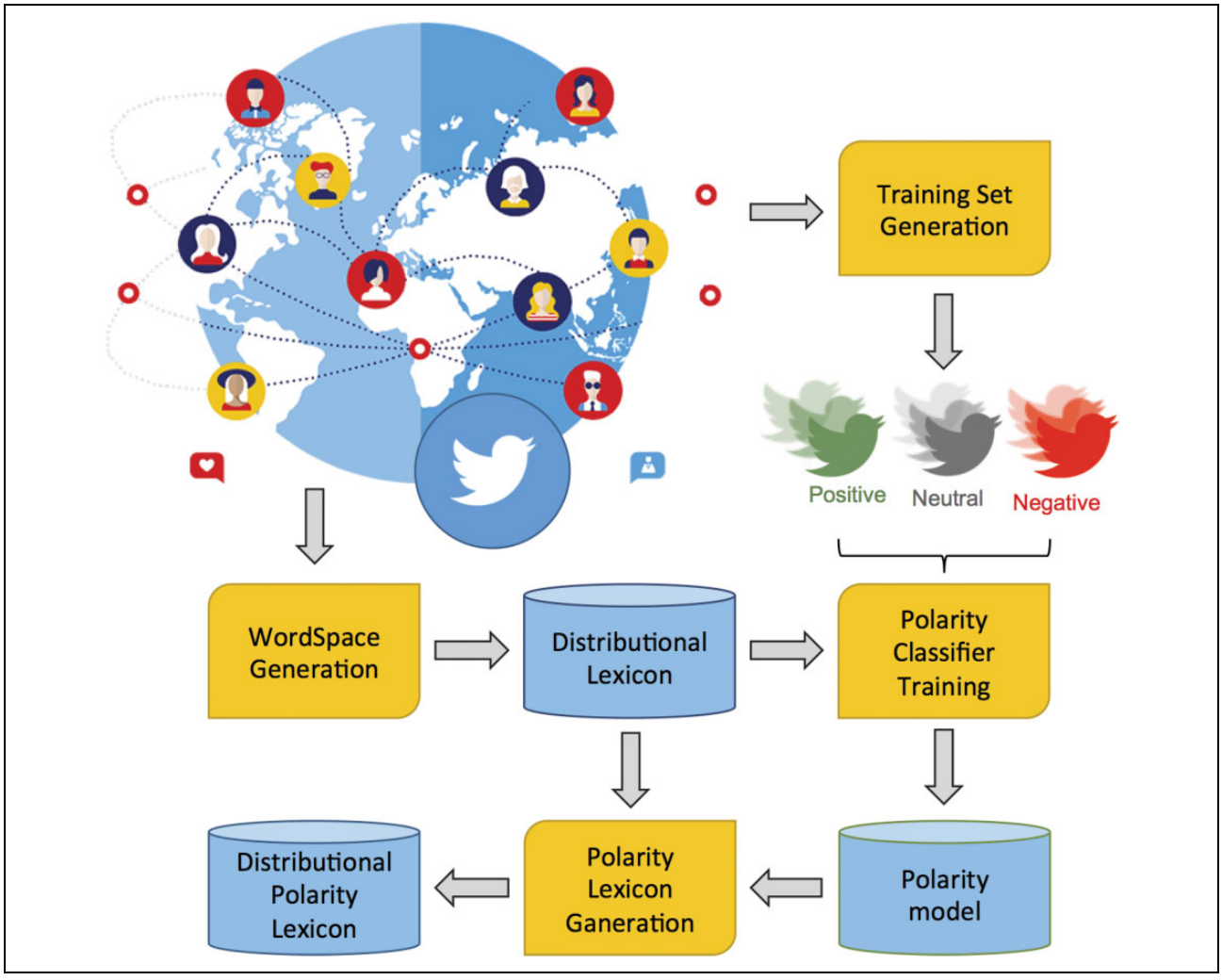

Figure 2. The architecture of the DPL acquisition process. DPL: Distributional Polarity Lexicon.

selecting emoticons is crucial for the data set construction. In fact, cultural- or language-dependent factors risk to introduce errors in the polarity assignment heuristic. In this work, the set of emoticons used to filter data has been selected to be as much independent as possible from these factors. We conducted a manual analysis on a small sample of the heuristically annotated data and it confirms that the selection process is sufficiently robust. It is worth noting that if a more fine-grained emoticons classification is available,[36] it will be possible to derive a data set made by even more heterogeneous data to observe finer grain phenomena.

Polarity lexicons acquisition

In this section, details about the acquisition of polarity lexicons are provided, and different SA tasks in three different languages are evaluated with these resources to prove the effectiveness of the proposed methodology, depicted in Figure 2.

WS generation

As discussed in section "Lexicon generation through classification," distributional representations for words are needed for the acquisition of a Distributional Polarity Lexicon (DPL). In the WS generation stage, a Skip-gram model[15] (described in section "Predicting words through vector representations: The Skip-gram model") is applied to an incoming large-scale collection of unannotated tweets. The word2vec (https://code.google.com/p/word2vec/) tool is adopted to acquire the WS according to the Skip-gram model and 250-dimensional vectors are derived for the majority of words appearing in a corpus (in particular, the following settings are adopted: *min-count=50*, *window=5*, *iter=10* and *negative=10)* in the so-called *Distributional Lexicon*.

Training set generation

A set of annotated tweet is derived in this stage by applying the heuristics described in section "Lexicon generation

through classification" to large-scale tweet collections. In particular, we select positive, neutral, and negative messages through the use of emoticons and URLs. The original tweet collection can be the same used for the WS generation or originating from different collections.

Polarity classifier training

SVMs[37] are among the most effective classifiers applied in many different fields. In Natural Language Processing, they have been used for their capability to learn both linear and nonlinear functions (by exploiting the notion of kernel function[38]). In this stage, a linear polarity classification function (called *Polarity Model*) is acquired to partition streams of tweets in three sentiment classes of interest: positive, neutral, and negative tweets. We adopted the LibLinear[39] formulation of SVM that can be found in KeLP (http://www.kelp-ml.org)[40] to learn the resulting Polarity Model: it corresponds to three real-valued functions that output the independent confidence scores for the three classes. Each sentence in the training set $\mathbb{T}$ is represented as the linear combination of vectors of the occurring words (in order not to be biased by the terms t used to retrieve the individual positive, negative, or neutral tweets, the token t is never considered in the training and is thus removed from the linear combination), whereas only the verb, noun, adjective, and adverb grammatical classes are considered. A one-versus-all strategy is applied to derive the optimal classifiers. Classifier hyperparameters are tuned by splitting the training set $\mathbb{T}$ through an 80/20% split: for each parameter configuration, a learning phase is performed on the 80% of the data and the accuracy, that is, the percentage of correctly classified examples, is computed over the remaining 20%.

Polarity lexicon generation

The three acquired classifiers are then used to compile the *DPL*: each word from the Distributional Lexicon is classified through the Polarity Model and the resulting three polarity scores are normalized into lexical three-dimensional vectors, as described in section "Lexicon generation through classification." A synthetic view of the process is described in Algorithm 1.

Measuring the impact of the DPL

In recent years, the interest in mining the sentiment expressed in the Web is growing, and different Twitter-based international benchmarking campaigns have been proposed in the computational linguistics area. We want to verify whether the polarity lexicon acquisition approach proposed in the previous sections is actually beneficial to the achievable quality on a Twitter-based SA task. Moreover, we aim at showing that the approach is language independent, in the sense that the strength of its benefits does not depend on the involved natural language.

Algorithm 1. The algorithm of the Distributional Polarity Lexicon generation process.

function LEXGEN (Annotated Sentences T, Polarities $p_1 \ldots p_m \in P$, Words W, DistributionalModel Φ)
  trainingSetVectors = []
  for t in T **do**
    $\vec{t} = \Sigma_{w \in t} \Phi(w)$
    trainingSetVectors = trainingSetVectors $\cup \{\vec{t}\}$
  end for
  for polarity $p_i \in P$ **do**
    f_i = trainBinaryClassifiers(*trainingSetVectors*, P, p_i)
  end for
  DPL = []
  for w in W **do**
    $\vec{p}(w) = [f_1(\Phi(w)), f_2(\Phi(w)), \ldots, f_m(\Phi(w))]$
    $\vec{p}(w)$ = softmaxNormalization($\vec{p}(w)$)
    $DPL = DPL \cup \{\vec{p}(w)\}$
  end for
  return the Distributional Polarity Lexicon DPL
end function

For this reason, we will evaluate automatically generated DPLs against tasks related to different benchmarking campaigns, held in three different languages. Starting from the 2013 and 2014 SemEval editions,[16,17] (for English data) we will investigate also the 2014 Evalita challenge on Twitter[18] over tweets written in Italian and finally, after generating a polarity lexicon for the Arabic language, we will also investigate the contribution against the AST data set.[19]

All the above campaigns focus on the task of assigning a sentiment to a target tweet. For example, the tweet "*Porto amazing as the sun sets…http://bit.ly/c28w*" should be recognized as *positive*, while "@knickfan82 Nooo;(they delayed the knicks game until Monday!" as *negative*. Notice that, as our method is based on automatically generated polarity lexicons, differences in performance obtained by classifiers that use the polarity lexicons against other uninformed classifiers (i.e. classifiers that do not use any polarity lexicon) will be assumed as quantitative indicators of the advantage produced by our DPL acquisition process.

All experiments reported in the rest of the article are performed by exploiting the kernelized formulation of the SVM algorithm[37] that can be found in the KeLP framework.[40] Kernels allow representing data at an abstract level, while their computation still refers to core informative properties. Moreover, kernel functions can be combined, for example, the contribution of kernels can be summed, in order to account at the same time for different linguistic properties. In the targeted tasks, multiple kernels are combined to verify the contribution of each representation: in particular, an independent kernel will be made dependent on one DPL to prove its effectiveness.

As presented in section 'Polarity lexicon generation through distributional approaches', an m = three-

dimensional vector $\vec{o}^k$ is available for each word w_k in the vocabulary, each expressing a positivity, negativity, and neutrality score for w_k. In order to represent an entire sentence t for SVM, we propose to adopt a very simple feature representation by summing up all the polarity lexicon vectors $\vec{o}^k$ corresponding to the words w_k in t (we apply a normalization on the resulting vector $\vec{t}$), that is, $\vec{t} = \sum_{w_k \in t} \vec{o}^k$. This should be able to capture a sort of agreement of many words with respect to a given polarity class; the dimension associated with a particular sentiment should have a higher score. This representation is very flexible and portable, as it can be adopted for different languages without changing the feature extraction process. Obviously, it has some limitations, for example, it doesn't account for the scope of negation, but, in principle, it can be modified to accommodate negation specific aspects. However, negation-related aspects are often domain and language dependent; in this work, we are interested in a general model that can be adopted for multiple languages.

In the remaining part of this section, we will first measure the impact of the DPL in the polarity classification task in three languages: English, Italian, and Arabic. These results will be obtained by deriving the lexicons from a distributional space generated by a neural network, that is, a *prediction-based* methodology. For completeness, the contribution of lexicons generated according to other *count-based* methodologies[41] will be then discussed in the last subsection.

Twitter SA in English

The English DPL is generated starting from a WS acquired over a corpus of more than 20 million tweets downloaded during the last months of 2014. We processed the corpus with a custom version of the Chaos parser[42]: lemmatization and pos tagging are applied to derive *lemma::pos* input for the word vector generation. We obtained about 190,000 words that have been classified to generate the polarity lexicon.

In Table 1, an excerpt of the English lexicon can be found, where *pos*, *neg*, and *neu* refer, respectively, to *positivity*, *negativity*, and *neutrality* scores. The approach seems to be able to transfer the polarity to words, given the sentence-based classifiers. Qualitatively, it seems that polarized words (e.g. the adjectives "good" and "bad") tend to lie in specific independent subspaces, which are well separated by the linear classification strategy: the word *good* in fact receives a positive score of 0.74, while *bad* receives a score of 0.12; on the contrary, the former word has a negative score of 0.11, while the latter 0.80. Notice that in Table 1, words that are mostly domain independent are shown, while the lexicon generation process could be biased by the sampling process of the training messages. For example, the word Mario Monti has a different polarity signature in different domains, changing from (0.15, 0.53, 0.32) in data sampled from 2014 to (0.09, 0.13, 0.78) in data sampled from 2016.

Table 1. Examples of polarity lexicon terms and relative sentiment scores (English language).[a]

Term	Positive	Negative	Neutral
good::j	0.74	0.11	0.15
☺	0.86	0.04	0.10
bad::j	0.12	0.80	0.08
pain::n	0.13	0.76	0.11
#apple	0.14	0.16	0.70
#ibm	0.07	0.04	0.89
#microsoft	0.09	0.09	0.82
#google	0.14	0.17	0.69
#dell	0.13	0.20	0.67
#barackobama	0.19	0.07	0.74
#mccain	0.22	0.16	0.62
article::n	0.16	0.09	0.75
government::n	0.09	0.09	0.82
friend::n	0.37	0.31	0.32
surprise::n	0.40	0.31	0.29

[a] *::j* and *::n* indicate, respectively, adjectives and nouns.

The quantitative evaluations focus on the quality that an SVM classifier can achieve with and without the adoption of DPL. In this setting, tweets are first modeled through two basic feature representations: a BoW and a WS. The former BoW captures the lexical information directly, whereas each binary dimension of the vector represents the presence (or absence) of a particular word in a sentence. The latter WS relies on a Distributional Lexicon acquired by automatically processing a large-scale tweet collection and it is able to generalize the meaning of single words: in particular, it is used to smooth the lexical overlap measure between messages obtained from the pure occurrence model expressed by the BoW vectors. The WS representation of the sentence is obtained by summing the vectors of all its verbs, nouns, adjectives, and adverbs.

Then, lexical representation of the involved words is further augmented by the representation with the polarity scores as derived from the DPL. Again, only verbs, nouns, adjectives, and adverbs are augmented so that other categories are neglected.

The SVM learning algorithm is then applied on different representations by devoting a different kernel function to each vector. In this way, each feature vector (e.g. the three-dimensional polarity lexicon) contributes independently through its own specific kernel function: the overall kernel function is computed as the normalized sum of the kernels over the different feature vectors. For example, the BoW + WS + DPL system makes use of three kernels: the first linear kernel operates on BoW binary vectors, the second on the WS vectors, and finally the third kernel is fed with three-dimensional polarity scores of the DPL; all kernels correspond to the cosine similarity function between vector pairs.

Table 2. SA in Twitter 2013 results.[a]

Kernel	F1pn	F1pnn
BoW	59.72	63.53
BoW + SBJ	61.46	64.95
BoW + DPL	60.78	64.09
BoW + WS	66.12	68.56
BoW + WS + SBJ	65.20	67.93
BoW + WS + DPL	66.40	68.68
Best-System	69.02	—

SA: sentiment analysis; BoW: bag-of-word; DPL: Distributional Polarity Lexicon; WS: word space.
[a]*Best-System* refers to the top scoring system in SemEval 2013.

Table 3. SA in Twitter 2014 results.[a]

Kernel	F1pn	F1pnn
BoW	58.74	61.38
BoW + SBJ	60.82	62.85
BoW + DPL	62.49	64.01
BoW + WS	65.20	66.35
BoW + WS + SBJ	64.29	66.13
BoW + WS + DPL	66.11	67.07
Best-System	70.96	—

SA: sentiment analysis; BoW: bag-of-word; SBJ: Subjectivity Lexicon; DPL: Distributional Polarity Lexicon; WS: word space.
[a]*Best-System* refers to the top scoring system in SemEval 2014.

In Tables 2 and 3, the experimental outcomes for the 2013 and 2014 SemEval data sets are reported, as well as the Best-System in the two challenges. Performance measures are the *F1pn* and the *F1pnn*. The former is the arithmetic mean between the F1 measures of the positive and negative classes, that is, the official score adopted by the SemEval challenges. The latter is the arithmetic mean between the F1 measures of the positive, negative, and neutral classes. The WS representation is based on the same WS used to generate the polarity lexicon. Here, we compare the contribution of DPL with a well-known lexicon, that is, the SBJ by Wilson et al.[9] It is composed of words manually annotated with subjective polarity information (positive, negative, neutral) and a strength (weak or strong) value. For each tweet, we generate a new feature representation SBJ where each dimension refers to a polarity value with its relative strength, as found in the message. For example, the SBJ representation of "Getting better!" is a feature vector whose only nonzero element is strong_pos. In Table 2, results are shown for the 2013 test data set, which is composed of 3814 examples. First, the baseline performance achievable with a linear kernel applied to the simple BoW (63.53% *F1Pnn*) representation is shown. Then, the combination of the other representations is experimented. When applying the WS, an improvement can be noticed, as demonstrated by the *F1pnn* score of 68.56% in the BoW + WS kernel. It means that distributional representations are useful to capture the semantic phenomena behind sentiment-related expressions, even in short texts and to alleviate data sparseness problems of the pure BoW kernel, as demonstrated by the approximately five point increment in *F1Pnn* in this setting. When combining also DPL, further improvements are obtained for both performance measures (66.40% *F1pn* and 68.68% *F1pnn*). It seems that DPL effectively acts as a smoothing of the contribution of the pure lexical semantics information provided by WS. It is noticeable that the BoW + WS + DPL system would have ranked in second position in the 2013 ranking, where the *Best-System* (the best system measured during the official competition adopted many polarity lexicons and ad hoc features.) achieved the *F1pn* score of 69.02%.

Similar trends are observable for the 2014 test set, as shown in Table 3. In this case, we were not able to rely on the complete test set, as, at the time of this experimentation, some of the messages were no longer available for download. Our evaluation is carried out on 1562 test examples, while the full test set was composed of 1853. It makes a direct comparison with the in-challenge systems impossible, but it still can give an idea of the achievable performances. Again, performances are measured with the BoW and WS representation combined with SBJ and DPL. As it can be noticed, the use of distributed word representations is also beneficial in this scenario, as demonstrated by the BoW + WS row of Table 3, where a 65.20% in *F1pnn* and 66.35% in *F1pnn* are reported. Again, when using the automatically acquired polarity lexicon, improvements in both the performance scores are noticeable, as demonstrated by 66.11% in the *F1pn* and 67.07% in the *F1pnn* of the BoW + WS + DPL setting. These results should be compared with the Best-System both in 2013 and 2014 considering that no hand-coded resource has been here adopted. Instead, the best systems measured during the official competition adopted many polarity lexicons (both automatically and manually derived) as well as different syntactic (char-ngrams and word-ngrams) and semantic features (word senses and word clusters).

Twitter SA in Italian

The Italian DPL is generated starting from a WS acquired over a corpus of more than nine million tweets. We processed, again, the corpus with a custom version of the Chaos parser[42]: lemmatization and pos tagging are applied to derive *lemma::pos* input for the word vector generation. We obtained about 99,000 words that have been classified to generate the polarity lexicon.

In Table 4, an excerpt of the Italian lexicon can be found. Again, *pos*, *neg*, and *neu* refer, respectively, to *positivity*, *negativity*, and *neutrality* scores: again, intrinsic positive adjectives such as "buono" (i.e. "good") are significantly separated from intrinsically negative words, such as the noun "sofferenza" (i.e. "pain").

Table 4. Example of polarity lexicon terms and relative sentiment scores (Italian language).[a]

Term	Positive	Negative	Neutral
buono::j (good::j)	0.77	0.12	0.11
☺	0.73	0.08	0.19
cattivo::j (bad::j)	0.23	0.63	0.14
sofferenza::n (pain::n)	0.17	0.48	0.35
#apple	0.17	0.12	0.71
#ibm	0.15	0.13	0.72
#microsoft	0.14	0.12	0.74
#google	0.20	0.07	0.73
#dell	0.13	0.20	0.67
#barackobama	0.24	0.09	0.67
#mccain	0.13	0.02	0.85
articolo::n (article::n)	0.19	0.05	0.76
governo::n (government::n)	0.12	0.12	0.76
amico::n (friend::n)	0.44	0.24	0.32
sorpresa::n (surprise::n)	0.40	0.22	0.38

[a]*::j* and *::n* indicate, respectively, adjectives and nouns.

Table 5. Twitter polarity classification in Italian.

Kernel	F1pn	F1pnn
BoW	62.49	58.58
BoW + STX	63.50	59.20
BoW + DPL	65.38	60.75
BoW + WS	68.26	63.13
BoW + WS + STX	68.46	63.33
BoW + WS + DPL	68.28	63.35

BoW: bag-of-word; DPL: Distributional Polarity Lexicon; WS: word space.

The impact of the Italian lexicon has been measured against the data provided by the Evalita 2014 Sentipolc[18] challenge. Here, Twitter messages are annotated with respect to subjectivity, polarity, and irony. We selected only those messages annotated with polarity and that were not expressing any ironic content in order not to have been biased by particular polarity inversion phenomena typical of ironic texts. Thus, our evaluations are pursued on 2566 and 1175 messages, used respectively for training and testing.

In Table 5, performance measures for this setting are reported. Again, the F1 mean between the positive and negative classes (*F1pn*), as well as the mean between all the involved classes are reported *F1pnn*. Notice that in the Sentipolc challenge, a slightly different evaluation has been carried out; however, in the challenge the best system obtained an F1 of 67.71% in the polarity classification subtask.

We compare DPL with another Italian polarity lexicon, called SENTIX, in the study by Basile and Nissim.[43] It consists of words automatically annotated with four sentiment scores, that is, *positive*, *negative*, *polarity*, and *intensity*. In our evaluation, features correspond to the sum of the four scores across words appearing in a message (STX kernel). The benefits of using a polarity lexicon for augmenting the BoW representation are more evident, and the improvement in using the two resources is very similar. In fact, the BoW kernel alone reaches a performance of 58.58% in *F1pnn*, and when augmented with the STX and the DPL, the performance increases, respectively, to 59.20% and 60.75. The DPL is able to provide more information to the learning algorithm, as demonstrated by the higher performance that is measured. When adopting the WS representation, performances increase up to 63.13% in *F1pnn*. When using also the DPL lexicon, it seems that the interaction with the WS features is beneficial in deciding the polarity of a tweet, as demonstrated by the further improvement up to 63.35%.

We also carried out a qualitative evaluation of the lexicons in the Italian language, that is, SENTIX and DPL. In Table 6, Italian words along with their scores from the DPL and SENTIX are compared. They have been selected by looking at the accordance/discordance (given the relative scores) in the two lexicons. For example, "vantaggioso" (*profitable*) has been selected as it is considered positive in both lexicons, while "inestimabile" (*priceless*) has been selected as the two lexicons disagree about its polarity. The examples in the table have been then manually inspected and selected to point out some linguistic phenomena. For example, it is interesting to notice that for some word, the two lexicons give similar judgments for their polarity. Let us consider the words "abile" (*expert*) or "benefico" (*beneficial*). In Italian, these can be considered almost unambiguous words from a polarity point of view, and the lexicons agree as well. The role of the DPL is more evident for words that can be considered more ambiguous. For example, the word "pentimento" (*regret*) can be considered as a positive status that follows from a negative situation. This outcome makes explicit the strong dependence of corpus-based methods onto the nature of the used text material. In the SENTIX lexicon, it has been assigned to a negative polarity, while in the DPL, it is biased toward positivity. We thus retrieved from the tweets selected via distant supervision all messages containing "pentimento." The following tweets confirm the positive bias assigned to this word: *anche io carnivora . ma in via di pentimento . a volte* ☺ (*I am also a carnivore . but in repentance . sometimes* ☺), *é stato difficile ma alla fine con molto pentimento ce l'ho fatta* !! ☺ (*It was difficult but in the end with much repentance I did it* !! ☺), and *bene io ora ho 2 min di riflessione e pentimento sul divano* ☺ (*Well I have 2 min of reflection and repentance on the couch* ☺).

Again, the word "inestimabile" (*priceless*) is considered negative, while in the DPL, it is biased toward positivity. We can argue that in the modern language of Social Media, if something is "inestimabile" (*priceless*), that is, whose value cannot be easily measured, it is used more with a positive connotation in the data, for example, in the tweet *un complimento di inestimabile valore ed importanza per*

Table 6. Comparison of polarity judgment of Italian words in the SENTIX lexicon and in the DPL.[a]

	SENTIX			DPL		
Word	Positive	Negative	Polarity	Positive	Negative	Neutral
abile (clever)	0.125	0.000	1.000	0.304	0.108	0.588
intelligente (smart)	0.125	0.000	1.000	0.624	0.120	0.256
incapace (incompetent)	0.125	0.750	−1.000	0.294	0.280	0.426
tollerabile (tolerable)	0.625	0.000	1.000	0.323	0.318	0.359
inaccettabile (unacceptable)	0.125	0.375	−0.590	0.107	0.280	0.613
benefico (beneficial)	0.625	0.000	−1.000	0.442	0.139	0.419
vantaggioso (profitable)	0.625	0.000	−1.000	0.618	0.117	0.265
terribile (terrible)	0.000	0.875	−1.000	0.138	0.611	0.251
inestimabile (priceless)	0.125	0.625	−0.749	0.524	0.171	0.305
pentimento (regret)	0.000	0.250	−1.000	0.403	0.110	0.487
sacrificare (sacrifice)	0.000	0.125	−1.000	0.188	0.172	0.640
elemosinare (beg)	0.375	0.000	1.000	0.215	0.314	0.471
imprecare (swear)	0.250	0.125	0.410	0.225	0.578	0.197
politico (politic)	0.125	0.000	1.000	0.139	0.190	0.671
desiderio (wish)	0.250	0.500	−0.410	0.356	0.302	0.342
logica (logic)	0.750	0.000	1.000	0.357	0.334	0.309

DPL: Distributional Polarity Lexicon.
[a]For the SENTIX lexicon, *polarity* ranges from −1 (totally negative) to 1 (totally positive) and it is a function of positive and negative scores. DPL scores are derived as described in the previous sections.

me ! ☺ (*a priceless and important compliment for me*! ☺). When the lexicons disagree, it can be the case that the SENTIX judgment is correct or, alternatively, the DPL is correct. For example, the DPL is wrong in assigning polarity scores to "incapace" (*incompetent*). On the other hand, the DPL is correct with respect to SENTIX for "logica" (*logic*) or "imprecare" (*swear*), respectively, neutral and negative. DPL scores are dependent from the data used to acquire the classifiers, so it is sensible to the real usage of the words in the contexts of Social Media. A lexicon as SENTIX can be considered instead *static*, as it does not directly depends on real examples of the usage of words in contexts. Moreover, the DPL lexicon contains polarity judgments for some *meta* word, such as hashtags or users, as demonstrated in Table 4, that can be useful for analyzing how people use such kind of words. For example, a hashtag indicating an event that is biased toward positivity in the DPL can be an indicator that the event was mainly associated with positive evaluations.

Generating an Arabic lexicon

Recently, the interest in the automatic analysis of the Arabic language has seen a rapid growth. Many different systems have been released for processing the Arabic language,[44,45] but SA systems as well as SA resources are not easily available. It makes the processing of Arabic texts from a sentiment point of view not an easy process. We aim at automatically generating a sentiment lexicon for Arabic by following the same methodology adopted both for English and Italian and showing that it can be adopted in existing SA system with low effort. Again, we generated a WS through word2vec, by downloading a corpus of about two millions of Arabic tweets. A preprocessing step is adopted by applying word segmentation and pos tagging to each tweet through the Stanford Arabic Parser.[45] We adopted the same settings as for the English and Italian WSs (given the reduced size of this corpus, we reduced the except the word2vec parameter called *min-count* to 10). The lexicon has been generated starting from a further corpus of Arabic tweets that we heuristically classified with respect to the *positive*, *negative*, and *neutral* classes by adopting the same emoticons set of the English and Italian languages.

Table 7. Example of polarity lexicon terms and relative sentiment scores (Arabic language).

Term	Positive	Negative	Neutral
سعيد (happy)	0.87	0.08	0.05
طاري (emergency)	0.04	0.92	0.04
عافية (health)	0.90	0.04	0.06
يتبادل (exchange)	0.01	0.02	0.97
يدمع (not crying)	0.87	0.06	0.07
☺	0.72	0.11	0.17
اصيح (shout)	0.06	0.91	0.03

In Table 7, a portion of the lexicon is shown. Again, the lexicon is able to capture the main sentiment attitudes of the highly polar words, such as "سعيد" that is, the adjective "happy." We conclude that the proposed methodology can be effectively considered language independent, as even in such different languages, it is able to extract meaningful polarity scores for the words.

In order to quantitatively evaluate the lexicon, we tested its contribution against the AST data set.[19] It is a recently released data set for SA over Twitter. It contains about

Table 8. AST data set statistics over the different classes.

Polarity	Positive	Negative	Neutral	Objective	Total
N°	799	1684	832	6691	10,006

AST: Arabic Sentiment Twitter.

Table 9. Evaluation of a kernel based SA system over the AST data set with the DPL: positive, negative, and neutral classes only.

Three way Kernel	Balanced		Unbalanced	
	Accuracy	F1	Accuracy	F1
BoW	50.9	50.4	58.1	54.2
BoW + DPL	52.2	51.7	57.2	55.0
BoW + WS	56.6	56.1	59.0	57.4
BoW + WS + DPL	55.3	54.9	60.5	57.9

SA: sentiment analysis; BoW: bag-of-word; DPL: Distributional Polarity Lexicon; WS: word space; AST: Arabic Sentiment Twitter.

10,000 Twitter messages in Arabic that have been manually annotated with respect to four classes: *positive*, *negative*, *neutral*, and *objective*. In Table 8, the number of messages in each class is reported.

An SVM classifier with multiple kernels is adopted to train a sentiment classifier over two different settings, *balanced* and *unbalanced*. In the first case, the number of examples is balanced with respect to the different classes provided by Nabil et al.[19] The *unbalanced* scenario instead works with training and testing data sets where the number of examples is not balanced with respect to the different polarity classes. We are going first to test the lexicon in a setting similar to the English and Italian cases, that is, a three-way classification task where we filtered out the objective class from the data set. Then, an evaluation over the four-way classification task as reported by Nabil et al.[19] is discussed.

In Table 9, the three-way task performances are reported, in terms of accuracy and F1, which are the measures used by Nabil et al.[19] We trained the SVM learning algorithm with different combination of kernel functions to test the contribution of each representation. Again, the DPL is evaluated both with a simple BoW representation and with a more complex BoW and WS representation. As it can be noticed, even in this language, the DPL lexicon is able to provide useful information to train an SA system for tweet messages, as demonstrated by the performances in the unbalanced settings (60.5% in accuracy and 57.9% in F1 with the BoW + WS + DPL kernel). In fact, the system seems to benefit more from the adoption of the lexicon when enriching the BoW + WS kernel with respect to the balanced scenario, where the performance instead decreases (56.6% and 54.9% down to 55.3% and 54.9%, respectively, in accuracy and F1). Notice that the unbalanced scenario is a more realistic setting for a final production system, as the data in real operational conditions are far from being balanced.

Table 10. Evaluation of a kernel-based SA system over the AST data set with the DPL: positive, negative, neutral, and objective classes.

Four way Kernel	Balanced		Unbalanced	
	Accuracy	F1	Accuracy	F1
BoW	47.0	46.8	66.1	62.5
BoW + DPL	46.4	46.4	67.6	61.9
BoW + WS	50.2	50.2	68.9	64.2
BoW + WS + DPL	52.5	52.6	69.1	63.0
[19] SVM	42.5	42.1	64.4	61.1
[19] Best	49.1	49.3	69.1	62.6

SA: sentiment analysis; BoW: bag-of-word; DPL: Distributional Polarity Lexicon; WS: word space; AST: Arabic Sentiment Twitter; SVM: support vector machine.

In Table 10, the four-way task results are reported with a comparison over a similar system by Nabil et al.,[19] that is, an SVM-based system, and the best system reported by Nabil et al.[19] Again, the DPL features are useful in the prediction of the sentiment expressed in short messages. Notice that, except one case (the F1 measures in the unbalanced setting with a BoW + WS + DPL kernel), the lexicon always provides a beneficial impact over the performance. It is remarkable that the lexicon is able to further generalize the WS contribution, as demonstrated by the accuracy (52.5%) and F1 (52.6%) in this data set in the balanced setting. In the unbalanced case, the contribution of the lexicon is noticeable only in the accuracy measures, even if it provides a score of 63.0% in the F1. In the unbalanced four-way task, the DPL is giving too much bias to the subjective classes, resulting in worse performances in the prediction of the *objective* class, that is, instead, the more populated (see Table 8) in the data set. In the balanced scenario, the *objective* class is instead treated similarly to the others, given that all the examples are equally distributed over them.

Impact of the methods for acquiring WSs on the DPL generation

In all previous evaluations, we considered DPLs that have been acquired starting from WSs obtained with the so-called *prediction*-based methods, that is, the *Skip-gram* model.[15] Here, we aim at verifying whether a WS acquired through the *count-based* methodology can provide similar results, in terms of lexicon acquisition. We are going to acquire a WS with the LSA[12] approach described in section "Counting co-occurrences: The LSA approach." We adopted the same set of tweet messages used to build the *Skip-gram* model. In order to have a comparable WS, we built the word-by-context matrix by considering a window of five words to the left and to the right of each target word, discarding the words appearing less than 50 times in

Table 11. Twitter SA in multiple languages.[a]

Kernel	En-2013	En-2014	It	Ar-bal	Ar-unbal
BoW	63.5	61.4	58.6	47.0	66.1
BoW + DPL	64.1	64.0	60.8	46.4	67.6
BoW + WS	68.6	66.4	63.1	50.2	68.9
BoW + WS + DPL	68.7	67.1	63.4	52.5	69.1
BoW + LDPL	64.7	63.6	60.0	47.8	68.2
BoW + LSA	68.7	67.6	62.5	52.0	68.5
BoW + WS + LSA + DPL + LDPL	69.6	68.7	63.5	51.9	69.3

BoW: bag-of-word; DPL: Distributional Polarity Lexicon; WS: word space; LSA: latent semantic analysis; SA: sentiment analysis; LDPL: LSA-Based DPL.

[a]Reported measures are the *F1pnn* for English and Italian cases and four-way *Accuracy* for the Arabic language.

English and Italian (this limit is set to 10 for Arabic) and by maintaining the most frequent contexts, that is, discarding those appearing less than 50 times in English and Italian (this limit is set to 20 for the Arabic language). We applied a point-wise mutual information weighting scheme.[14] It means that the value of a cell of the initial matrix M is computed according to the mutual information of a target word and a context in which it appears. It measures how much a target word and a context are related but penalizing most frequent words and contexts. Then, we applied the SVD algorithm, and we approximated the space with a $k = 250$ value, finally obtaining 250-dimensional vectors by considering the word projection $Y_{\text{terms}} = U_k S_k^{1/2}$.

Given this space, we acquired English, Italian, and Arabic DPLs with the same settings described in the previous sections. We applied, again, the new lexicon over the SemEval Twitter SA 2013 and 2014 tasks, the Italian tweet data set and over the AST data set with the same experimental setup previously adopted as well as the same performance metrics. We applied different linear kernel combinations to verify the contribution of the newly generated sentiment representation. In the following experiments, the WS derived with the LSA method will be denoted as LSA, while the lexicon generated starting from this space will be called LSA-Based DPL.

In Table 11, the measures (F1Pnn) for all the data set in all languages are reported. First, notice that for the English data sets (En-2013 and En-2014), the LSA space is able to provide good generalization capabilities leading to performances that are comparable to the ones obtained with the *prediction*-based space in section "Twitter SA in English." The outcomes when using only the two versions of DPLs with a BoW (BoW + DPL and BoW + LDPL) are quite similar in all cases, for example, 64.1% versus 64.7% in the 2013 English data set and 64.0% versus 63.6% in the 2014 English data set. It shows that the two methodologies induce very similar semantic and sentiment representations.

Moreover, we tested whether the two spaces could provide complementary information to the learning algorithm. We combined the *WS* and *LSA* kernels also with their relative lexicons DPL and LDPL. The outcome is quite interesting, as the combination BoW + WS + DPL + LSA + LDPL shows a good increment, about one point in the *F1pnn* for both the English Twitter data sets.

Similar trends can be observed in the Italian case (column it). The LSA lexicon (LDPL) in combination with the BoW is beneficial, as demonstrated by the 60.0% measure that is higher than 58.6% of the pure BoW kernel. The BoW + WS + LSA + DPL + LDPL kernel combination confirms its positive effects, as it seems to provide additional useful information to the learning process, as demonstrated by the score of 63.5% with this configuration.

Finally, in the Arabic language scenario (columns Arabic-balanced and Arabic-unbalanced), we measured the system with the LSA and LDPL representations against the four-way classification task. Again, we can notice a positive impact of the LSA WS and of the LSA derived sentiment lexicon, both in the *balanced* and *unbalanced* scenarios. Notice how the adoption of the BoW + WS + LSA + DPL + LDPL kernel gives an improvement in the accuracy of the unbalanced scenario reaching the score of 69.3%. The balanced setting does not benefit of the same improvement. Nevertheless, the combination of both kinds of WSs (i.e. *count based* and *prediction based*) with their DPL lexicons is beneficial. It suggests that these are capturing slightly different linguistic information, and, thus, they should be both adopted in language learning systems to capture these differences in SA tasks.

Conclusions

Subjective phenomena, such as polarity, represent crucial issues in the modeling of complex social networks that are increasingly influent on modern decision-making and business process. In this article, an unsupervised learning methodology to generate large-scale polarity lexicons (the lexicons and the emoticons used for generating them are available on: (http://sag.art.uniroma2.it/demo-software/distributional-polarity-lexicon/)) is presented to automatically acquire such precious resources for SA across social networks. The methodology is simple and allows to be easily replicated for multiple languages. We show how polarity-related aspects can be observed across streams of microblogs as they are observed in the Social Media. Through the use of simple heuristics, large data sets including annotated examples can be easily derived in terms of individual sentences that are representative of certain polarity classes. These sentences are then used to train a classifier and transfer polarity information to individual lexical items. This transfer is made possible as both sentences and words are represented in the same vector space based on DMs of lexical semantics, and therefore training the linear polarity classifier becomes straightforward. The method proved to be quite general, as it does not rely on any hand-coded resource, but mainly uses simple cues, for example, emoticons, for generating a large corpus of labeled sentences. It

turns out to be largely applicable to resource poor languages, such as the Italian or Arabic languages. The generated lexicons have been in fact shown beneficial on SA tasks in three different languages. In particular, DPLs have been adopted for predictive tasks, that is, the classification of polarity in short texts. However, a DPL can be also used for different applications. For example, such a resource could be adopted to support the analysis of the words *coloring* in specific domains, for example, through the automatic generation of polarized tag clouds. Moreover, the generality of the lexicon generation process allows to acquire different version of the lexicons in different time periods. In fact, as we demonstrated in the study by Castellucci et al.,[46] the usage of words can change over time. This is evident for such words that refer to events or people that can be used to communicate positive or negative biases. In the study by Castellucci et al.,[46] we acquired different lexicons in 2014 and 2016 in Italian. For example, in these lexicons, the word referring to the former Italian prime minister *Mario Monti* shifted its polarity from negativity to neutrality (Mario Monti was the author of some unpopular law in 2013, resulting in one of most criticized person in Social Media.), that is, DPL vector in 2014 was (0.15, 0.53. 0.32), while in 2016, it was (0.09, 0.13, 0.78). Notice how the three-dimensional representation of the DPL allows to easily track also the neutrality of each term. In fact, differently from other lexicons, where polarity is represented only with one or two dimensions (only for positivity and negativity), we decided to track neutrality with a separated classifier. The neutral classifier is responsible to track the neutral contexts in which words appear, thus resulting in a more fine-grained representation. Moreover, this richer representation schema is flexible in combination with machine learning algorithms, such as SVMs, that can automatically select the most expressive dimension for a targeted task. In the article, we provided an analysis of the lexicon generation process by studying two different distributional methods. On the one side, we explored *prediction* methods, that is, method inspired by neural language models whose lexical vectors correspond to predictors of the context of individual words. On the other side, we also applied *count*-based methods whose vectors express for co-occurrence counts as these are found in large corpora. The two methodologies provide representations that are morphologically very similar though expressing quite different information. The acquired lexicons seem to have a comparable impact on the polarity scores generation and in sentiment classification tasks, as test over different SA data sets provides quite similar performances. Moreover, the combination of the two different representations, that is, the adoption of the resource derived by the application of both paradigms, results in further improvements. It seems that the two WSs provide slightly different contributions resulting in different and independent information about the task: it is probably due to the fact that the two WSs are built with quite diverse methods, each looking at different information of the texts. Future investigations will systematically address the problem of combining multiple lexicons when available. In fact, in English, multiple affective resources are available[25,26]: it could be an interesting direction to combine them to improve the performances of SA classifiers. Again, it could be interesting to combine the DPL representations obtained by different distributional methods. For example, all these could be adopted in combination with novel and promising convolutional neural networks used within sentence classification tasks.[47] One possibility is the investigation of the impact of these lexicons in augmenting the representation of individual words in pure neural network architectures. In this case, we expect the network should automatically learn the suitable representations for the classification of the polarity of messages, according to the different facets of the individual word semantics. A further direction is the investigation about the use of more complex grammatical features in the stage of the polarity lexicon acquisition. All the adopted classification algorithms did make no use of negation or other grammatical markers in the texts. Irony is another neglected phenomenon, so that a further extension of our method should be focusing on the management of ironic phenomena.[48,49] Distributional polarity vectors capture the main usage of words but not their ironic or metaphorical senses. It should be interesting to verify if an approach similar to the one suggested by Castellucci et al.[50] could be beneficial. In that work, deviation from standard semantic usages of words provides effective information on the irony detection task. A similar method applied on to distributional polarity vectors could provide interesting features for modeling irony in even more complex contexts.

Author contribution

Giuseppe Castellucci mainly contributed to this work while working at the University of Roma, Tor Vergata (Italy). Now at Almawave srl, Roma (Italy).

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

1. Backstrom L, Huttenlocher D, Kleinberg J, et al. Group formation in large social networks: membership, growth, and evolution. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '06*. New York, NY, USA 2006. pp. 44–54. New York, NY: ACM. DOI: 10.1145/1150402.1150412.
2. Potthast M, Hagen M, Völske M, et al. Crowdsourcing interaction logs to understand text Reuse from the Web.

In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013*, 4–9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers; 2013. pp. 1212–1221. http://aclweb.org/anthology/P/P13/P13-1119.pdf.

3. Kramer ADI, Guillory JE and Hancock JT. Experimental evidence of massive-scale emotional contagion through social networks. *Proc National Acad Sci* June 2, 2014; 111 (24): 8788–8790. http://www.pnas.org/content/111/24/8788.abstract.
4. Plank B and Hovy D. Personality Traits on Twitter -or- How to Get 1500 Personality Tests in a Week. In: *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Lisboa, Portugal, September 17, 2015, pp. 92–98. Lisboa, Portugal: Association for Computational Linguistics. http://aclweb.org/anthology/W15–2913.
5. Ho SM, Hancock JT, Booth C, et al. Computer-mediated deception: strategies revealed by language-action cues in spontaneous communication. *J Manage Inform Syst* 2016; 33(2): 393–420. DOI: 10.1080/07421222.2016.1205924.
6. Hall W, Hendler J, and Staab S. A manifesto for web science @ 10. CoRR. 2017;abs/1702.08291. http://arxiv.org/abs/1702.08291.
7. Pang B and Lee L. Opinion mining and sentiment analysis. *Found Trend Inform Retr* January, 2008; 2(1–2): 1–135. http://portal.acm.org/citation.cfm?id=1454712.
8. Turney PD and Littman ML. Measuring praise and criticism: inference of semantic orientation from association. *ACM Trans Inform Syst (TOIS)* 2003; 21(4): 315–346.
9. Wilson T, Wiebe J, and Hoffmann P. 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*. Stroudsburg, PA, USA, 2005. pp. 347–354. Stroudsburg, PA, USA: Association for Computational Linguistics. DOI: 10.3115/1220575.1220619.
10. Stone PJ, Dunphy DC, Smith MS, et al. *The General Inquirer: A Computer Approach to Content Analysis*. Cambridge, USA: MIT Press, 1966.
11. Hu Minqing and Liu Bing. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '04)*, New York, NY, USA, pp. 168–177. ACM. DOI=http://dx.doi.org/10.1145/1014052.1014073
12. Landauer T and Dumais S. A solution to plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychol Rev* 1997; 104(2): 211–240.
13. Mintz M, Bills S, Snow R, et al. Distant supervision for relation extraction without labeled data. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, *ACL '09*. Stroudsburg, PA, USA, 2009, pp. 1003–1011. Stroudsburg, PA, USA: Association for Computational Linguistics.
14. Sahlgren M. *The Word-space model*. Sweden: University of Stockholm, 2006.
15. Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. CoRR. 2013; abs/1301.3781. http://arxiv.org/abs/1301.3781.
16. Nakov P, Rosenthal S, Kozareva Z, et al. SemEval-2013 task 2: sentiment analysis in twitter. In: *Proceedings of Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, USA, June 2013, pp. 312–320. Atlanta, USA: Association for Computational Linguistics.
17. Rosenthal S, Ritter A, Nakov P, et al. SemEval-2014 task 9: sentiment analysis in twitter. In: Nakov P and Zesch T (ed) *SemEval@COLING*. The Association for Computer Linguistics, August 2014, pp. 73–80. Available from: http://dblp.uni-trier.de/db/conf/semeval/semeval2014.html#RosenthalRNS14.
18. Valerio Basile, Andrea Bolioli, Malvina Nissim, et al. Overview of the Evalita 2014 SENTIment POLarity classification task. *Proceedings of the 4th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'14)*, Pisa, Italy, 2014.
19. Nabil M, Aly M, and Atiya A. ASTD: Arabic sentiment tweets dataset. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, 2015, pp. 2515–2519. Lisbon, Portugal.
20. Mohammad SM and Turney PD. Emotions evoked by common words and phrases: using mechanical turk to create an emotion lexicon. In: *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. CAAGET '10*, Stroudsburg, PA, USA, June, 2010, pp. 26–34. Stroudsburg, PA, USA: Association for Computational Linguistics. Available from: http://dl.acm.org/citation.cfm?id=1860631.1860635.
21. Esuli A and Sebastiani F. Sentiwordnet: a publicly available lexical resource for opinion mining. In: *Proceedings of 5th LREC*, Genoa, Italy, 2006, pp. 417–422. ELRA.
22. Miller GA. WordNet: a lexical database for English. *Commun ACM* 1995; 38(11): 39–41.
23. Delip Rao and Deepak Ravichandran. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL '09)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 675–682.
24. Zhang Z and Singh MP. ReNew: a semi-supervised framework for generating domain- specific lexicons and sentiment analysis. In: *Proceedings of 52nd Annual Meeting of the ACL (Volume 1: Long Papers)*, Baltimore, Maryland, June, 2014, pp. 542–551. Association for Computational Linguistics.
25. Kiritchenko S, Zhu X, and Mohammad SM. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research* 2014; 50: 723–762. DOI: 10.1613/jair.4272.
26. Saif H, Fernandez M, He Y, et al. Senticircles for contextual and conceptual semantic sentiment analysis of twitter. In: Presutti Valentina, d'Amato Claudia, Gandon Fabien, et al

(eds). *The Semantic Web: Trends and Challenges. vol. 8465 of LNCS*. Springer International, 2014, pp. 83–98.
27. Harris Z. Distributional structure. *Word* 1954; 10(23): 146–162.
28. Cristianini N, Shawe-Taylor J and Lodhi H. Latent semantic kernels. In: Brodley C and Danyluk A (ed) *Proceedings of ICML-01*. USA: Williams College, 2001, pp. 66–73.
29. Bhat V, Oates T, Shanbhag V, et al. Finding aliases on the web using latent semantic analysis. *Data Knowl Eng* 2004; 49(2): 129–143.
30. Croce D, Moschitti A, and Basili R. Structured lexical similarity via convolution kernels on dependency trees. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing. EMNLP '11*. Stroudsburg, PA, USA, July 2011, pp. 1034–1046. Stroudsburg, PA, USA: Association for Computational Linguistics. Available from: http://dl.acm.org/citation.cfm?id=2145432.2145544.
31. Croce D, Castellucci G, and Bastianelli E. Structured learning for semantic role labeling. *Int Artif* 2012; 6(2): 163–176.
32. Baroni M, Dinu G, and Kruszewski G. Don't count, predict! A systematic comparison of contextcounting vs. context-predicting semantic vectors. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 238–247.
33. Golub G and Kahan W. Calculating the singular values and pseudo-inverse of a matrix. *J Soc Industr Appl Math B Num Anal* 1965; 2(2): 205–224.
34. Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model. *J Mach Learn Res* 2003; 3: 1137–1155.
35. Go A, Bhayani R and Huang L. Twitter sentiment classification using distant supervision. *Processing* 2009; 1–6.
36. Suttles J and Ide N. Distant supervision for emotion classification with discrete binary values. In: Gelbukh A (ed) *Computational Linguistics and Intelligent Text Processing. vol. 7817 of LNCS*. Berlin, Heidelberg: Springer, 2013, pp. 121–136.
37. Vapnik VN. *Statistical Learning Theory*. Hoboken, NJ: Wiley-Interscience, 1998.
38. Shawe-Taylor J and Cristianini N. *Kernel Methods for Pattern Analysis*. Cambridge: Cambridge University Press, 2004.
39. Fan RE, Chang KW, Hsieh CJ, et al. Liblinear: a library for large linear classification. *J Mach Learn Res* 2008; 9: 1871–1874.
40. Filice S, Castellucci G, Croce D, et al. KeLP: a kernel-based learning platform for natural language processing. In: *Proceedings of ACL-IJCNLP 2015 System Demonstrations. Association for Computational Linguistics and The Asian Federation of Natural Language Processing*, July 2015, pp. 19–24. Available from: http://aclanthology.coli.uni-saarland.de/pdf/P/P15/P15-4004.pdf.
41. Turney PD and Pantel P. From frequency to meaning: vector space models of semantics. *J Artif Int Res* 2010; 37: 141–188.
42. Basili R, Pazienza MT and Zanzotto FM. Efficient parsing for information extraction. In *Proceedings of The 17th European Conference on Artificial Intelligence (ECAI)*, Riva del Garda, Italy, August 1998, pp. 135–139. IOS Press.
43. Basile V and Nissim M. Sentiment analysis on Italian tweets. In: *Proceedings of the 4th WS: Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Atlanta, USA, June 2013, p. 100. Association for Computational Linguistics.
44. Diab M. Second generation tools (AMIRA 2.0): fast and robust tokenization, POS tagging, and base phrase chunking. In: Choukri K and Maegaard B (eds) *Proceedings of the Second International Conference on Arabic Language Resources and Tools*. Cairo, Egypt: The MEDAR Consortium, 2009, pp. 285–288.
45. Green S and Manning CD. Better Arabic parsing: baselines, evaluations, and analysis. In: *Proceedings of the 23 rd International Conference on Computational Linguistics. COLING '10*. Stroudsburg, PA, USA, 2010, pp. 394–402. Stroudsburg, PA, USA: Association for Computational Linguistics.
46. Castellucci G, Croce D, De Cao D, et al. User mood tracking for opinion analysis on twitter. In: Adorni Giovanni, Cagnoni Stefano, Gori Marco, et al editors, *AI* IA 2016 Advances in Artificial Intelligence*, Genova, Italy, December 2016, pp. 76–88. Springer International Publishing.
47. Kim Y. Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1746–1751.
48. Farías DIH, Patti V and Rosso P. Irony detection in twitter: the role of affective content. *ACM Trans Int Technol* 2016; 16 (3): 19:1–19:24.
49. Farias DIH and Rosso P. Chapter 7 - irony, sarcasm, and sentiment analysis. In: Pozzi FA, Fersini E, Messina E and Liu B (eds) *Sentiment Analysis in Social Networks*. Burlington: Morgan Kaufmann, 2017, pp. 113–128.
50. Castellucci G, Croce D, De Cao D, et al. A multiple kernel approach for twitter sentiment analysis in Italian. In: *Proceedings of the 4th EVALITA*, 2014, pp. 98–103.