Acquiring an Italian Polarity Lexicon through Distributional Methods

Giuseppe Castellucci[1], Danilo Croce[2], and Roberto Basili[2]

[1] Department of Electronic Engineering,
University of Roma Tor Vergata, Roma Italy
castellucci@ing.uniroma2.it,
[2] Department of Enterprise Engineering,
University of Roma Tor Vergata, Roma Italy
{croce,basili}@info.uniroma2.it

Abstract. Recent interests in Sentiment Analysis brought the attention on effective methods to detect opinions and sentiments in texts. Many approaches in literature are based on resources, such as Polarity Lexicons, which model the *prior* polarity of words or multi-word expressions. Developing such resources is expensive, language dependent, and linguistic sentiment phenomena are not fully covered in them. In this paper an automatic method for deriving polarity lexicons based on Distributional Models of Lexical Semantics is presented. Given a set of heuristically annotated messages from Twitter, we transfer sentiment information from sentences to words. As the approach is mostly unsupervised, it enables the acquisition of polarity lexicons for languages that are lacking these resources. We acquired a polarity lexicon in the Italian language, and experiments on Sentiment Analysis tasks show the benefit of the generated resources.

1 Introduction

Opinion Mining [16] aims at tracking the opinions expressed in texts with respect to specific topics, e.g. products or people. In particular, Sentiment Analysis (SA) deals with the problem of deciding whether an excerpt of text, e.g. a sentence or a phrase, is expressing a trend towards specific feelings. Tracking these phenomena can be crucial in different applications. As an example, a sentiment-aware retrieval or recommender system would produce more informative results for users interested in tracking opinions about specific topics or products.

In many approaches for SA, polarity lexicons have been adopted (see for example [23, 27]) to support the development of systems that automatically assign a polarity class to sentences (or texts in general) by matching words or phrases with the entries of the lexicon. In these resources, each term is associated to its prior polarity, under the assumption that some words can evoke something positive or something negative out of any context. For example, "*good*" can be associated to a prior positive sentiment in contrast to "*sad*", considered negative in every domain. These lexicons are often hand-compiled, as [22] or [9]. 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 topic dependent, e.g. the occurrences of the word *mouse* are mostly neutral

in the consumer electronics domain, while it can be negatively biased in a restaurant domain. Accounting for topic-specific phenomena would require manual revisions, and while these resources exist for English, they scarce for others.

In this paper, we propose an efficient and unsupervised methodology to derive large-scale polarity lexicons. It mainly exploits the extra-linguistic information within Social Media, e.g. the presence of emoticons in messages. The approach is based on Distributional Models of Lexical Semantics, by exploiting the equivalence in sentences and words representations available in some distributional models (e.g. the dual LSA space for words and texts introduced in [11]). As sentences can be clearly related to polarity, a classifier can always be trained in such spaces and used to *transfer* sentiment information from sentences to words. Specifically, we train polarity classifiers by observing sentences and we classify words to populate a polarity lexicon. Annotated messages are derived from Twitter[3] and their polarity is determined by simple heuristics. 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 distributional model can be acquired without any supervision, and the provided heuristics do not have any bias with respect to languages or domains.

We generated an Italian polarity lexicon, and its contribution is measured against different SA tasks. In particular, a first evaluation is based on Twitter Sentiment Analysis tasks within the context of Evalita[4] [2]. A second evaluation, consider a more complex setting, where tweet messages are classified within a context [25, 24].

In the remaining, related works are discussed in Section 2. Section 3 presents the proposed methodology, while Section 4 describes the experimental evaluations. Conclusions are derived in Section 5.

2 Related Work

Polarity lexicon generation has been tackled in many researches and three main areas can be pointed out.

Manually annotated lexicons. Earlier works are based on manual annotations of terms with respect to emotional categories. For example, in [22] sentiment labels are manually associated to 3600 English terms. In [9] a list of positive and negative words are manually extracted from customer reviews. The MPQA Subjectivity Lexicon [27] contains words, each with its prior polarity (positive or negative) and discrete strength (strong or weak). The NRC Emotion Lexicon [14] is composed by frequent English nouns, verbs, adjectives, and adverbs annotated through Amazon Mechanical Turk with respect to eight emotions (e.g. joy, sadness, trust) and positive or negative 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 [5] the WordNet [13] synset glosses are exploited to derive three scores describing the positivity, negativity and neutrality of

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

[4] http://www.evalita.it

the synsets. The work in [17] generates a lexicon as graph label propagation problem. Each node in the graph represents a word. Each weighted edge encodes a relation between words derived from WordNet [13]. 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. In [3] a polarity lexicon for Italian called SENTIX has been derived from existing lexical resources, such as [5, 13]: it consists of words automatically annotated with 4 sentiment scores, *positive*, *negative*, *polarity* and *intensity*.

Corpus-based lexicons. Statistics based approaches are more general as they mainly exploit corpus processing techniques. For example, [23] proposed a minimally supervised approach to associate a polarity tendency to a word by determining if it co-occurs more with positive words than negative ones. More recently, [28] proposed a semi-supervised 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 [10] a minimally-supervised approach based on Social Media data is proposed by exploiting hashtags or emoticons related to positivity and negativity, e.g., `#happy`, `#sad`, `:)` or `:(`. They compute a score, reflecting the polarity of each word, through a Point wise Mutual Information based measure between a word and an emotion. This work is close to [10], as we use tweets and emoticons to derive a labeled dataset. Differently, our approach exploits distributional models for training a classifier to acquire the lexicon.

3 Polarity Lexicon Generation through Distributional Approaches

In Section 3.1 we first describe how sentences and words can be represented through Distributional Models. In Section 3.2 the classification approach to transfer the sentiment information from sentences to words is presented. Section 3.3 describes a heuristic to generate a polarity annotated dataset of sentences.

3.1 Distributional Models

In order to rely on comparable representations for words and sentences, Distributional Models (DM) of Lexical Semantics are exploited. DMs are intended to acquire semantic relationships between words, mainly by looking at the word usage. The foundation for these models is the *Distributional Hypothesis* [8], that is words that are used and occur in the same contexts tend to purport similar meanings. Although DMs are similar in nature, as they all derive vector representations for words from more or less complex corpus processing stages, quite different methods have been proposed to derive them.

Main approaches estimate semantic relationships in terms of vector similarity. Different relationships can be modeled, e.g. *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 [20]. In such models, words like *run* and *walk* are close in the space, while *run* and *read* are projected in different subspaces. These representations can be derived mainly in two ways: *counting* the co-occurrences between words, e.g. [11], or *predicting* word representations in

a supervised setting. In particular, in [12] a simple Recursive Neural Network architecture is exploited to derive such representations. These show linguistic regularities at syntactic and semantic levels that allow to reason about analogy tasks, e.g. judging whether *king*:*man* $\sim$ *queen*:*woman*, as in [12]. Roughly speaking, these regularities are reflected in specific subspaces, that is specific dimensions of the generated vectors.

Despite the specific algorithms used for the space acquisition, these approaches allow to derive a projection function $\Phi(\cdot)$ of words into a geometrical space, i.e. the vector representation for a word $w_k \in \mathbb{W}$ is obtained through $\boldsymbol{w_k} = \Phi(w_k)$, into a d-dimensional geometric space. Geometrical regularities will be exploited to determine the prior sentiment for words: our assumption is that polarized words lie in specific subspaces. However, in DMs opposite polarity words are often similar, as they share the same contexts. In the following, we discuss how we can transfer known sentence polarity to single words by exploiting those subspaces.

3.2 Lexicon Generation through Classification

The semantic similarity (closeness) established by traditional DMs does not correspond well with 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, a discriminant function can be derived through machine learning over these representations. Let us consider a space $\mathbb{R}^d$ where some geometrical representations of a set of annotated examples can be derived. In general, a linear classifier can be seen as a separating hyper plane $\theta \in \mathbb{R}^d$ that is used to classify a new example represented in the same space. Each θ_i corresponds to a specific dimension, or feature i that has been extracted from the annotated examples. After a learning stage, the magnitude of each θ_i reflects the importance of the feature i with respect to a target phenomenon. In this sense, when applied on distributional vectors of word semantics, linear classifiers are expected to learn the regions useful to discriminate examples with respect to the target classes. If these classes reflect the sentiment expressed by words, a classifier should find those subspaces better correlating examples with the sentiment. In this way, any set of words $w_i \in \mathbb{W}$ associated with their prior polarity could be used to train a sentiment classifier. In fact, given a set of seed words whose prior polarity is known, their projection in the Word Space model $\boldsymbol{w_k^{seed}} = \Phi(w_k^{seed})$ is sufficient to train the linear classifier. This would find what dimensions in $\mathbb{R}^d$ are related to the different polarities. Classification, thus, corresponds to transferring the knowledge about sentiment implicit in the seed words to the other remaining words.

A number of limitations affect this view. First, the definition and annotation of seed words could be expensive and certainly not portable across languages, e.g. from English to Italian. 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. We suggest to avoid the selection of lexical seeds and emphasize the role of distributional models: the representations of both sentences and words are here capitalized to automatize the development of portable sentiment lexicons. We propose to make use of sentences as seeds of the classifier training, as these embody sentiment in a more explicit (and unambiguous) manner: for example sentences including strong

sentiment markers can be cheaply gathered and would provide a large scale seed resource. As these sentences and words (candidate entries for the polarity lexicon) lie into the same space (sentences and semantically related words belong to the same subspaces), we would be able to acquire a classifier over sentences and flexibly apply it to a very large lexicon. The subspaces strongly related to a sentiment class can be used to project it over the lexicon.

In details, we have words $w_k \in \mathbb{W}$ and their vector representation $\boldsymbol{w_k} \in \mathbb{R}^d$ obtained by projecting them in a Word Space, i.e. $\boldsymbol{w_k} = \Phi(w_k)$. We also have a training set $\mathbb{T}$, including sentences associated to a polarity class. In order to project an entire sentence in the same space, we apply a simple but effective linear combination operator. For each sentence $t \in \mathbb{T}$, we derive the vector representation $\boldsymbol{t} \in \mathbb{R}^d$ by summing all the word vectors composing the sentence, i.e. $\boldsymbol{t} = \sum_{w_i \in t} \Phi(w_i)$. It is one of the simpler, but still expressive, method that is used to derive a representation that accounts for the underlying meaning of a sentence[5], as discussed in [11]. 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 a discriminant function f, which is expected to capture the sentiment related subspaces by properly weighting each dimension i of the original space. The lexicon is generated by applying f to the entire $\mathbb{W}$. As we deal with multiple sentiment classes, f can be seen as m distinct binary functions $(f_1, \ldots, f_m)$, one for each sentiment class. Each word $w_k \in \mathbb{W}$ is classified with all the f_i, thus deriving m distinct scores s_i^k, each reflecting the classifier confidence in deciding whether w_k belongs to class i. Each s_i^k is normalized through a softmax function[6], obtaining the final polarity score o_i^k: each w_k is represented both with its distributional representation, i.e. $\boldsymbol{w_k} = \Phi(w_k)$, and with its sentiment representation, i.e. $\boldsymbol{o^k}$.

3.3 Generating a Dataset through Emoticons

As discussed in Section 3.2, an annotated dataset of sentences $\mathbb{T}$ is needed to acquire a linear classifier that emphasizes specific subspaces. Although different dataset of such kind exists, 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 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 [7].

In order to derive messages belonging to the positive or negative classes, we select Twitter messages whose last token is a smile either positive, e.g. `:)` or `:D` or negative, e.g. `:(` 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 that use mainly non-polar words to announce an article. We further filter out those messages that contain elements of other classes: if a message ends with a positive smile and it contains either a negative smile or a url it will be discarded. It is worth nothing that if a more fine-grained emoticon classification is available, it will be possible to derive a dataset made by even more heterogeneous data and observe finer grain phenomena.

[5] Notice that in this model word order is neglected.

[6] $o_i^k = e^{s_i^k} / \sum_{j=1}^{m} e^{s_j^k}$

4 Evaluating an Italian Polarity Lexicon

In this Section, details about the acquisition of an Italian polarity lexicon are provided, and different Sentiment Analysis tasks are evaluated.

Word vectors generation. As discussed in Section 3.2 the proposed approach for the polarity lexicon acquisition requires a distributional representation for words. We generated word vectors according to a Skip-gram model [12] through the `word2vec`[7] tool. In particular, we derived 250 dimensional word vectors[8], by using a corpus of more than 2 million tweets in Italian downloaded during the 2013 summer. We processed each tweet with a custom version of the Chaos parser [4]: lemmatization and part-of-speech (pos) tagging are applied. We obtained $16,579$ words that have been classified to generate the polarity lexicon.

Dataset generation. We applied the heuristics described in Section 3.3 to the same dataset used for word vector generation. Then, we filtered these data by randomly selecting $7,000$ tweets[9] for each class.

Acquisition of classification functions. In this work, both the acquisition of the polarity lexicon and the sentiment characterization of sentences/messages are modeled as classification problems. Classifiers are derived by applying the Support Vector Machine (SVM) learning algorithm [26]: in Natural Language Processing, SVM has been used for its capability to learn both linear and non-linear classification functions (by exploiting the notion of Kernels [21]). The polarity lexicon is acquired by a linear classifier that can realize a fuzzy assignment of words to the three sentiment classes of interest[10]. In the experimental evaluation hereafter presented, kernels based SVM has been also applied[11]. Kernels can be thought as similarity functions between data instances allowing to acquire non-linear classifiers. These are particularly interesting as the kernel combination is still a kernel, e.g. the contribution of kernels can be summed, thus capturing several linguistic properties of texts at the same time. In the targeted tasks, multiple kernels are combined to verify the contribution of each representation. In particular, one kernel function will be made dependent on the automatically generated polarity lexicon.

Distributional Polarity Lexicon generation. We represented each sentence in the training set $\mathbb{T}$ by linearly combining word vectors[12] considering verbs, nouns, adjectives and adverbs. As we are dealing with three sentiment classes, i.e. `positive`, `negative` and `neutral`, a One-Vs-All (OVA) strategy [18] is adopted to derive the optimal classifiers. A tuning phase is pursued on an 80/20 split of the training data $\mathbb{T}$ by optimizing the accuracy, i.e. the percentage of correctly classified examples. The lexicon is finally obtained by classifying the words of the distributional model, thus deriving the polarity scores as described in Section 3.2.

In Table 1 an excerpt of the Italian lexicon can be found. The approach seems able to transfer the polarity to words, given the sentence-based classifiers. Qualitatively, it

[7] `https://code.google.com/p/word2vec/`

[8] `word2vec` settings are: *min-count=50*, *window=5*, *iter=10* and *negative=10*.

[9] This number was selected through a validation phase.

[10] We adopted the LibLinear [6] formulation of SVM to acquire the classifiers.

[11] In this case the kernel based SVM implemented in KeLP is adopted, available at `http://sag.art.uniroma2.it/demo-software/kelp/`

[12] In order not to be biased by the query terms, the last token is not employed in the combination.

seems that polar words tend to lie in specific subspaces, which is captured by the linear classification strategy.

term	positivity	negativity	neutrality
ottimo::j (good::j)	0.71	0.11	0.18
:)	0.73	0.08	0.19
sofferenza::n (pain::n)	0.16	0.58	0.26
soffrire::v (suffer::v)	0.08	0.65	0.27
#apple::h (#apple::h)	0.17	0.12	0.71
articolo::n (article::n)	0.19	0.05	0.76

Table 1. Example of Italian polarity lexicon terms, and, in brackets, their English translation.

As presented in Section 3, a $m = 3$-dimensional vector $\boldsymbol{o^k}$ 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 $\boldsymbol{o^k}$ corresponding to the words w_k in t, i.e. $\boldsymbol{t} = \sum_{w_k \in t} \boldsymbol{o^k}$, and by finally normalizing $\boldsymbol{t}$. This should be able to capture when many words agree with respect to the polarity; the dimension associated to a particular sentiment should have a higher score. Obviously, this basic representation has some limitations, e.g. it doesn't account for the scope of negation.

4.1 Sentiment Analysis in Twitter

In recent years, the interest in mining the sentiment expressed in the Web is growing, and different Twitter based challenges have been proposed in the Computational Linguistics area, e.g. the 2013 and 2014 SemEval evaluations [15, 19]. In this paper we focus on the Italian language, by testing the automatically generated lexicon on the Evalita Italian challenge on Twitter Sentiment Analysis [2] and in a more complex setting that considers the stream in which a tweet is immersed [25, 24]. In both cases, we concentrate on the task of assigning a polarity class to a message. It means that a tweet as "*@andreaiannone29 stavi indiavolato... Bravo peccato per il rettilineo ma meglio di cosí non potevi fare!!! #Motomondiale*" should be recognized as `positive`, while "*La Yamaha é buona solo ad alzar polemiche.*" should be recognized as `negative`.

The SVM algorithm operates on messages represented according to a geometrical perspective, i.e. vectors. A Bag-Of-Word (BOW) vector captures directly the lexical information, whereas each binary dimension expresses the presence (or absence) of a particular word in a sentence. The Word Space (WS) vector relies on a word space to generalize the meaning of words in a message by smoothing the lexical overlap. The WS representation of a text is obtained by summing the vectors of all its verbs, nouns, adjectives and adverbs. Finally, the polarity information is modeled by adopting the Distributional Polarity Lexicon (DPL), through the 3-dimensional vector defined in the experimental setup. Again, only verbs, nouns, adjectives and adverbs are considered. The SVM learning algorithm is adopted along with kernel functions, which are applied on the different representations described above. Every information (e.g. the polarity

lexicon) is represented independently through a kernel function: the overall normalized sum of the different kernels is then adopted as the overall kernel function.

The first evaluation considers the data provided by the Evalita 2014 Sentipolc [2] challenge. The dataset consists of short messages annotated with the `subjectivity`, `polarity` and `irony` classes. We selected only those messages annotated with polarity and that were not expressing any ironic content. Thus, the datasets used for our evaluations consist of $2,566$ and $1,175$ messages, used respectively for training and testing. Linear kernel, 2-degree polynomial kernel (`poly`) and 1-gamma Gaussian kernel (`rbf`) are adopted in different combinations [13]. In Table 2 performance measures for each setting are reported. We measured the Precision and Recall of different systems, each using a specific combination of kernel functions. We then computed the F1 measure as the harmonic mean between Precision and Recall for each involved class. Finally, in Table 2 we report the mean between the F1 measures of the positive and negative classes (*F1-Pn*), as well as the mean of the F1 measures considering all the classes (*F1-Pnn*). In the linear kernel case the benefits of using the polarity lexicon for augmenting the BOW representation is more evident. When adopting the WS representation, performances increase, and when using also the DPL lexicon, it seems that the interaction with the WS features is beneficial in deciding whether a tweet is *positive* or *negative*, as demonstrated by the 66.04 *F1-Pn* measures. A different result is obtained when adopting a polynomial kernel over the BOW representation and a Gaussian kernel over the WS representation. In this case, the combination of a linear kernel over the polarity lexicon seems not to be beneficial. Instead, applying a Gaussian kernel also over the DPL lexicon allows to further push the performances about of 1 point in *F1-Pn*.

System	**F1-Pn**	**F1-Pnn**
BOW	61.58	57.97
BOW+DPL	62.35	58.30
BOW+WS	65.48	61.13
BOW+WS+DPL	66.04	60.99
poly_{BOW}+rbf_{WS}	68.52	63.24
poly_{BOW}+rbf_{WS}+DPL	68.45	63.14
poly_{BOW}+rbf_{WS}+rbf_{DPL}	**69.17**	**63.40**

Table 2. Twitter Polarity Classification in Italian.

4.2 Sentiment Analysis in Twitter with Contextual Information

The second experiment measures the contribution of the polarity lexicon in a further Sentiment Analysis classification task where *Contextual Information* of each message is considered, as defined in [25]. A context is a temporally ordered sequence of messages where a target tweet is the last element. This can be classified by considering the additional information of its preceding messages. Two kinds of contexts can be considered: *Conversation*, where a target tweet is immersed in a stream defined by the

[13] Notice that non-linear kernels are not adopted for the lexicon acquisition but they are used only on the final representation derived from the lexicon.

temporally ordered sequences of reply messages it appears in. The *Hashtag* context, instead, includes all temporally preceding messages sharing at least one hashtag with a target tweet. In [25], a sequence labeling algorithm is adopted to relate sentiment information of sequences with the sentiment of a target message in the English language. In particular, the classification is carried out through the SVM-HMM [1] algorithm. It learns a model isomorphic to a k-order Hidden Markov Model (HMM), and a sequence is classified by finding the sequence of HMM states that explains the given observations from a contextual sentiment point of view: the Viterbi algorithm is adopted to derive the sequence of sentiment states. These models have been demonstrated to be very effective, resulting in improvements with respect to alternatives where tweets are classified in isolation.

In the following, the experimental settings adopted in [24] for the Italian language are considered in order to verify the contribution of the automatically generated resource. Here, only the *Conversation* context is considered, where a tweet and all its preceding messages have been manually annotated by three annotators with respect to 4 different classes, `positive`, `negative`, `neutral` and `conflict`. The dataset is composed by 939, 201 and 296 instances, respectively for training, development and testing. All the data have been pre-processed as in the previous evaluations.

	Precision				**Recall**				$\mathbf{F_1}$				$\mathbf{F_1pnn}$	$\mathbf{F_1pnnc}$
	pos	*neg*	*neu*	*conf*	*pos*	*neg*	*neu*	*conf*	*pos*	*neg*	*neu*	*conf*		
BOW														
w/o conv	.532	.519	.403	.500	.610	.435	.621	.060	.568	.473	.489	.111	.510	.410
w conv	.542	.507	.401	.313	.565	.391	.632	.104	.553	.441	.491	.156	.495	.411
BOW+WS														
w/o conv	.585	.566	.439	.268	.551	.511	.540	.229	.567	.537	.485	.247	.530	.459
w conv	.566	.584	.435	.263	.623	.489	.621	.104	.593	.533	.512	.149	.546	.447
BOW+WS+DPL														
w/o conv	.579	.611	.462	.364	.638	.598	.632	.083	.607	.604	.534	.136	**.582**	**.470**
w conv	.583	.620	.453	.176	.609	.533	.667	.063	.596	.573	.540	.092	.569	.450

Table 3. Evaluation results of the Italian setting.

In Table 3 performance measures for the Italian contextual setting are reported. Precision, Recall and F1 measure are reported, as well as the F1 mean between the `positive`, `negative` and `neutral` classes (F_1pnn) and the F1 mean between all involved classes (F_1pnnc), thus including also the `conflict` class. The *w/o conv* rows refer to the case when a multi-classifier is used to decide the polarity of a message ignoring the previous message. The *w conv* rows refer to the case when the SVM-HMM algorithm is adopted, that is using the context during the classification phase. Results suggest that contextual information can be beneficial in the BOW and BOW+WS cases. However, when augmenting the data representation with the one derived from the lexicon, it seems that additional information derived from the context is not useful, as demonstrated by the drop in both F1 measures in the BOW+WS+DPL setting. It means that the contextual information is able to overcome the problems that arise when observing single tweets, e.g. lack of information in short messages. However, when interacting

with the polarity features here proposed, contextual information is not beneficial. In order to not be biased by the relative small size of this dataset, we conducted similar experiments on the English language following the settings of [25]. In fact, in this scenario, 8045, 1001 and 999 messages are available respectively for training, development and testing. We acquired a polarity lexicon on a 20 million English tweet corpus analyzed again with `word2vec` to derive the distributional representations. We obtained the English polarity lexicon by applying the same methodology adopted for the Italian one, obtaining $188,635$ words associated with polarity scores.

Both *Conversation* and *Hashtag* contexts are considered, and the sentiment class (`positive`, `negative` or `neutral`) for messages in context (different from the target) is assigned by a multi-classifier trained on a BOW and WS representations that does not use any context. This is necessary as these dataset have only the gold annotation for the target tweet, i.e. the last element of a context. Thus, this can be considered a more noisy setting, as contextual polarity classes are automatically derived.

Size	System	Base	+DPL
3	BOW	65.73	67.03
	BOW+WS	66.54	67.95
	BOW+WS+USP	68.88	68.54
6	BOW	65.24	65.49
	BOW+WS	67.10	67.10
	BOW+WS+USP	65.42	67.09
ALL	BOW	62.34	65.69
	BOW+WS	67.03	68.20
	BOW+WS+USP	67.91	68.59

Table 4. Twitter Conversation context results (English).

In Tables 4 and 5 performances for the two context-based settings are reported. As in [25] BOW, WS and User Sentiment Profile (USP) representations are used as basic features. WS is based on the Word Space used to generate the lexicon. The USP models the sentiment attitude of the user, acquired within the previous messages in its timeline, as defined in [25]. In addition, the Distributional Polarity Lexicon (DPL) representation is adopted. We report the *F1-Pnn* of the classification only of the target tweet, as in [25].

It can be noticed in Table 4 that the adoption of polarity lexicons is beneficial for the classification of tweet in conversation streams for the English language. In particular, the adoption of DPL is more evident when augmenting the BOW representation with small context size, i.e. 3. When using also WS or USP, improvements are less prominent for larger context sizes. It means that the distributional polarity lexicon is able to overcome data sparsity issues when less information is available, while its contribution is less important within richer contexts. When considering the hashtag context, performances trends are the same. Even in this setting, larger contexts seem to provide useful information for the sequence classification. The contribution of the lexicon is more evident when less data are available. In fact, at context size 3, augmenting the system BOW+WS+USP with DPL allows to obtain the state-of-the-art on this dataset,

Size	System	Base	+DPL
3	BOW	64.12	65.56
	BOW+WS	67.75	68.29
	BOW+WS+USP	69.32	**70.30**
6	BOW	63.72	64.92
	BOW+WS	68.89	67.90
	BOW+WS+USP	69.10	67.92
16	BOW	64.16	65.88
	BOW+WS	66.75	67.31
	BOW+WS+USP	66.66	67.79
31	BOW	65.13	64.38
	BOW+WS	67.63	67.90
	BOW+WS+USP	67.12	67.65

Table 5. Twitter Hashtag context results.

as compared to the best configuration (*69.32*) as measured by [25]. These results are quite interesting, as the acquired lexicon does not require any supervision.

5 Conclusions

In this paper, an unsupervised methodology to generate large-scale polarity lexicons is presented. Emotion related characteristics are observed over heuristically annotated sentences and are used to transfer the sentiment to lexical items. This transfer is made possible as both sentences and words lie in the same space, characterized by the underlying Distributional Model. The method is quite general, as it does not rely on any hand-coded resource. One major advantage is that it exploits sentiment at sentence level, that is a clearer emotional notion than polarity of a lexical entry: it is often impossible to provide a specific sentiment class to a word out of any context.

A large-scale polarity lexicon in Italian has been acquired and it has been shown beneficial on diverse Sentiment Analysis tasks. Moreover, English experiments further show the generality of the approach. In the future, we will investigate the integration of more complex grammatical features. In fact, the experimented classification algorithms were not sensitive to negation or other grammatical markers nor to ironic phenomena in the texts. Finally, specific work on multi-word expressions, e.g. *give up*, is needed as they have been almost neglected here.

References

1. Altun, Y., Tsochantaridis, I., Hofmann, T.: Hidden Markov support vector machines. In: Proc. of ICML (2003)
2. Basile, V., Bolioli, A., Nissim, M., Patti, V., Rosso, P.: Overview of the evalita 2014 sentiment polarity classification task. In: Proc. of the 4th EVALITA (2014)
3. Basile, V., Nissim, M.: Sentiment analysis on italian tweets. In: Proc. of the 4th WS: Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. pp. 100–107 (2013)

4. Basili, R., Pazienza, M.T., Zanzotto, F.M.: Efficient parsing for information extraction. In: ECAI. pp. 135–139 (1998)
5. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: In Proc. of 5th LREC. pp. 417–422 (2006)
6. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)
7. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Processing pp. 1–6 (2009)
8. Harris, Z.: Distributional structure. In: Katz, J.J., Fodor, J.A. (eds.) The Philosophy of Linguistics. Oxford University Press (1964)
9. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proc. of 10th Int. Conf. on Knowledge Discovery and Data Mining. pp. 168–177. ACM (2004)
10. Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. JAIR 50, 723–762 (Aug 2014)
11. Landauer, T., Dumais, S.: A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review 104 (1997)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013), http://arxiv.org/abs/1301.3781
13. Miller, G.A.: Wordnet: A lexical database for english. Commun. ACM 38(11), 39–41 (1995)
14. Mohammad, S.M., Turney, P.D.: Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In: Proceedings of NAACL 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. ACL (2010)
15. Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., Wilson, T.: Semeval-2013 task 2: Sentiment analysis in twitter. In: 2nd Joint Conference on Lexical and Computational Semantics (*SEM), Proc. of SemEval. pp. 312–320. ACL, Atlanta, USA (June 2013)
16. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. 2(1-2), 1–135 (Jan 2008)
17. Rao, D., Ravichandran, D.: Semi-supervised polarity lexicon induction. In: Proc. of the EACL. pp. 675–682. ACL (2009)
18. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. J. Mach. Learn. Res. 5, 101–141 (Dec 2004), http://dl.acm.org/citation.cfm?id=1005332.1005336
19. Rosenthal, S., Ritter, A., Nakov, P., Stoyanov, V.: Semeval-2014 task 9: Sentiment analysis in twitter. In: Proc. SemEval. ACL and Dublin City University (2014)
20. Sahlgren, M.: The Word-Space Model. Ph.D. thesis, Stockholm University (2006)
21. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
22. Stone, P.J., Dunphy, D.C., Smith, M.S., Ogilvie, D.M.: The General Inquirer: A Computer Approach to Content Analysis. MIT Press (1966)
23. Turney, P.D., Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Trans. Inf. Syst. 21(4), 315–346 (2003)
24. Vanzo, A., Castellucci, G., Croce, D., Basili, R.: A context based model for sentiment analysis in twitter for the italian language. In: First Italian Conference on Computational Linguistics CLiC-it. vol. 1 (2014)
25. Vanzo, A., Croce, D., Basili, R.: A context-based model for sentiment analysis in twitter. In: Proc. of 25th COLING: Best Paper. pp. 2345–2354. Dublin City University and Association for Computational Linguistics (2014)
26. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience (1998)
27. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proc. of EMNLP. ACL (2005)
28. Zhang, Z., Singh, P.M.: Renew: A semi-supervised framework for generating domain-specific lexicons and sentiment analysis. In: Proc. of ACL. pp. 542–551. ACL (2014)