On the Automatic Learning of Sentiment Lexicons

Aliaksei Severyn
DISI, University of Trento
38123 Povo (TN), Italy
severyn@disi.unitn.it

Alessandro Moschitti
Qatar Computing Research Institue
5825 Doha, Qatar
amoschitti@qf.org.qa

Abstract

This paper describes a simple and principled approach to automatically construct sentiment lexicons using distant supervision. We induce the sentiment association scores for the lexicon items from a model trained on a weakly supervised corpora. Our empirical findings show that features extracted from such a machine-learned lexicon outperform models using manual or other automatically constructed sentiment lexicons. Finally, our system achieves the state-of-the-art in Twitter Sentiment Analysis tasks from Semeval-2013 and ranks 2nd best in Semeval-2014 according to the average rank.

1 Introduction

One of the early and rather successful models for sentiment analysis (Pang and Lee, 2004; Pang and Lee, 2008) relied on manually constructed lexicons that map words to their sentiment, e.g., *positive*, *negative* or *neutral*. The document-level polarity is then assigned by performing some form of averaging, e.g., majority voting, of individual word polarities found in the document. These systems show an acceptable level of accuracy, they are easy to build and are highly computationally efficient as the only operation required to assign a polarity label are the word lookups and averaging. However, the information about word polarities in a document are best exploited when using machine learning models to train a sentiment classifier.

In fact, most successful sentiment classification systems rely on supervised learning. Interestingly, a simple bag of words model using just unigrams and bigrams with an SVM has shown excellent results (Wang and Manning, 2012) performing on par or beating more complicated models, e.g., using neural networks (Socher et al., 2011).

Regarding Twitter sentiment analysis, the top performing system (Mohammad et al., 2013) from Semeval-2013 Twittter Sentiment Analysis task (Nakov et al., 2013) follows this recipe by training an SVM on various surface form, sentiment and semantic features. Perhaps, the most valuable finding is that sentiment lexicons appear to be the most useful source of features accounting for over 8 point gains in the F-measure on top of the standard feature sets.

Sentiment lexicons are mappings from words to scores capturing the degree of the sentiment expressed by a given word. While several manually constructed lexicons are made available, e.g., the MPQA (Wilson et al., 2005), the Bing and Liu (Hu and Liu, 2004) and NRC Emoticon (Mohammad and Turney, 2013) lexicons, providing high quality word-sentiment associations compiled by humans, still their main drawback is low recall.

For example, the largest NRC Emoticon lexicon contains only 14k items, whereas tweets with extremely sparse surface forms are known to form very large vocabularies. Hence, using larger lexicons with better recall has the potential of learning more accurate models. Extracting such lexicons automatically is a challenging and interesting problem (Lau et al., 2011; Bro and Ehrig, 2013; Liu et al., 2013; Tai and Kao, 2013; Yang et al., 2014; Huang et al., 2014). However, different from previous work our goal is not to extract human-interpretable lexicons but to use them as a source of features to improve the classifier accuracy.

Following this idea, the authors in (Mohammad et al., 2013) use features derived from the lexicons to build a state-of-the-art sentiment classifier for Twitter. They construct automatic lexicons using noisy labels automatically inferred from emoticons and hashtags present in the tweets. The word-sentiment association scores are estimated using pointwise mutual information (PMI) computed between a word and a tweet label.

While the idea to model statistical correlations between the words and tweet labels using PMI or any other metric is rather intuitive, we believe there is a more effective way to exploit noisy labels for estimating the word-sentiment association scores. Our method relies on the idea of distant supervision (Marchetti-Bowick and Chambers, 2012). We use a large distantly supervised Twitter corpus, which contains noisy opinion labels (positive or negative) to learn a supervised polarity classifier. We encode tweets using words and multi-word expressions as features (which are also entries in our lexicon). The weights from the learned model are then used to define which lexicon items to keep, i.e., items that constitute a good sentiment lexicon. The scores for the lexicon items can be then directly used to encode new tweets or used to derive more advanced features. Using machine learning to induce the scores for the lexicon items has an advantage of learning the scores that are directly optimized for the classification task, where lexicon items with higher discriminative power tend to receive higher weights.

To assess the effectiveness of our approach, we reimplemented the state-of-the-art system ranking 1st in Semeval-2013 Twitter Sentiment Analysis challenge and used it as our baseline. We show that adding features from our machine-learned sentiment lexicon yields better results than any of the automatic PMI lexicons used in the baseline and all of them combined together. Our system obtains new state-of-the-art results on the SemEval-2013 message level task with an F-score of 71.32 – a 2% of absolute improvement over the previous best system in SemEval-2013. We also evaluate the utility of the ML lexicon on the five test sets from a recent Semeval-2014 task showing significant improvement over a strong baseline. Finally, our system shows high accuracy among the 42 systems participating in the Semeval-2014 challenge ranking 2nd best according to the average rank across all test sets.

2 Our model

We treat the task of sentiment analysis as a supervised learning problem, where we are given labeled data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ and the goal is to estimate a decision function $f(\mathbf{x}) \to \mathbf{y}$ that maps input examples to labels. In particular, we use a linear SVM model with the prediction function of the following form: $f = sign(\mathbf{w}^T\mathbf{x} + b)$, where the model weights $\mathbf{w}$ are estimated from the training set.

In the following we describe our approach to construct sentiment lexicons by learning an SVM model on the the distant supervised dataset. Finally, we describe our baseline model.

2.1 Distant Supervision for Automatic Lexicon Construction

Our sentiment lexicon consists of words and word sequences (we only use word unigrams and bigrams). To select lexicon items from a set of all unigrams and bigrams, we propose the following process:

1. Collect a large unlabelled corpus of tweets C.
2. For each tweet $t_i \in C$ use cues (hashtags or emoticons) to automatically infer its label (positive or negative): $y_i \in \{-1, +1\}$. For example, positive or negative emoticons, such as ' `:-)` ' or ' `:(`' are good indicators of the general sentiment expressed by a tweet.
3. Extract unigram and bigram features to encode a tweet t_i into a feature vector $\mathbf{x}_i \in \mathbb{R}^{|L|}$, where the lexicon L is a set of unigrams and bigrams.
5. Train an SVM model $\mathbf{w} = \sum_{i=1..N} \alpha_i y_i \mathbf{x}_i$ on the encoded corpus $C = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. The model $\mathbf{w} \in \mathbb{R}^{|L|}$ is a dense vector whose components are obtained from a weighted combination of training examples $\mathbf{x}_i$ (support vectors) and their labels y_i (only those instances with $\alpha_i > 0$ contribute to the components of $\mathbf{w}$).
6. Given that the each component w^j of the model $\mathbf{w}$ directly corresponds to the lexicon entry $l_j \in L$ its raw score is used as a sentiment association score.

Different from manually constructed lexicons compiled by humans where each item is assigned with an interpretable sentiment score, the scores in the automatic lexicon are learned automatically on a weakly supervised task. We use the weights from an SVM model whose weights are formed by the support vectors, i.e., the most difficult instances close to the decision boundary, hence most useful for the classification task. Additionally, due to its regularisation properties, SVM is known to select only the most robust features, which is important in the case of noisy labeled data. Hence, our method is a more principled way grounded in the statistical learning theory to exploit the noisy labels for estimating the word-sentiment association scores for the lexicon entries. Moreover, feature engineering with our lexicon appears to be more helpful (see Sec. 3) on a supervised task.

2.2 Baseline model

We re-implement the state-of-the-art NRC model from (Mohammad et al., 2013), which ranked 1st in the Semeval-2013, and use it as our baseline. This system relies on various n-gram, surface form and lexicon features. Briefly, we engineered the following feature sets:[1]

- **Word and character grams**: we use 1,2,3 n-grams for words and 3,4,5 n-grams for character sequences;
- **Negation**: the number of negated contexts – a span of words between a negation word (*not, never*), and a punctuation mark.
- **Lexicons**: given a word, we lookup its sentiment polarity score in the lexicon: $score(w)$. The following *aggregate* features are produced for the lexicon items found in a tweet: the total count, the total sum, the maximal score and the score of the last token. These features are produced for unigrams, bigrams, each part-of-speech tag, hashtags and all-caps tokens.
- **Other**: number of hashtags, capitalized words, elongated words, positive and negative emoticons, punctuation.

3 Experiments

In the following experiments our goal is to assess the value of our distant supervision method to automatically extract sentiment lexicons. We compare its performance with other automatically constructed lexicons extracted from large Twitter corpora, e.g., auto lexicons built using the PMI approach from (Mohammad et al., 2013).

Negative	Positive
(disappointing,)	(no, problem)
(depressing,)	(not, bad)
(bummer,)	(not, sad)
(sadly,)	(cannot, wait)
(passed, away)	(no, prob)

Table 1: Lexicon items learned from Emoticon140 corpus with top negative and positive scores.

3.1 Lexicon learning

We extract our lexicon from a freely available Emoticon140 Twitter corpus (Go et al., 2009), where the sentiment labels are automatically inferred from emoticons contained in a tweet[2]. The major advantage of such corpora is that it is easy to build as emoticons serve as fairly good cues for the general sentiment expressed in a tweet, thus they can be used as noisy labels. Hence, large datasets can be collected without incurring any annotation costs.

Tweets with positive emoticons, like ':)', are assumed to be positive, and tweets with negative emoticons, like ':(', are labeled as negative. The corpus contains 1.6 million tweets with equal distribution between positive and negative tweets. We use a tokeniser from the CMU Twitter tagger (Gimpel et al., 2011) extracting only unigrams and bigrams[3] to encode training instances. To make the extraction of word-sentiment association weights from the model straight-forward, we ignore neutral labels thus converting the task to a binary classification task. We use LibLinear (Fan et al., 2008) with L2 regularization and default parameters to learn a model. Preprocessing, feature extraction and learning is very fast taking only a few minutes. As the number of unique unigrams and bigrams can be very large and we would like to keep our sentiment lexicon rea-

[1] our baseline system, lexicon and the code to construct it are freely available at: `https://github.com/yyy`

[2] unfortunately, the corpus to build the NRC Hashtag lexicon (Mohammad et al., 2013) is not freely available due to Twitter data distribution policies.

[3] Adding tri-grams yielded a very minor improvement, yet the size of the dictionary exploded, so to keep the size of the dictionary relatively small we use only uni- and bi-grams.

Dataset	Size	Pos	Neg.	Neu.
Train'13	9,728	38%	15%	47%
Dev'13	1,654	35%	21%	45%
Twitter'13	3,813	41%	16%	43%
SMS'13	2093	24%	19%	58%
Twitter'14	1,853	53%	11%	36%
Sarcasm'14	86	38%	47%	15%
LiveJournal'14	1,142	37%	27%	36%

Table 2: Datasets.

sonably small, we filter entries with small weights. In particular, we found that selecting items with a weight greater than $1e-6$ did not cause any drop in accuracy, while the resulting lexicon is reasonably compact — it contains about 3 million entries.

Table 1 gives an example of top 10 lexicon entries with highest positive and negative scores. Interestingly, one would expect to find words such as *amazing*, *cool*, etc. as having the highest positive sentiment score. However, an SVM model assigns higher scores to bigrams containing negative words *problem, bad, worries*, to outweigh their negative impact. This helps to handle the inversion of the sentiment due to negations.

It is important to note that our goal is different from constructing sentiment lexicons that are interpretable by humans, e.g., manually built lexicons, but, similar to (Mohammad et al., 2013), we build automatic lexicons to derive highly discriminative features improving the accuracy of our sentiment prediction models.

3.2 Setup

Task. We focused on the Twitter Sentiment Analysis (Task 2) from Semeval-2013 (Nakov et al., 2013) and its rerun (Task 9) from Semeval-2014 (Rosenthal et al., 2014). Both tasks include two subtasks: an expression-level and a message-level subtasks. Being more general, we focus only on predicting the sentiment of tweets at the message level, where given a tweet, the goal is to classify whether it expresses *positive*, *negative*, or *neutral* sentiment.

Evaluation. We used the official scorers from the Semeval 2013 & 2014, which compute the average between F-measures for the positive and negative classes.

Data. We evaluated our models on both Semeval-2013 and Semeval-2014 tasks with 44 and 42 participating systems correspondingly. The Semeval-2013 task released the training set containing 9,728 tweets, dev and two test sets: *Twitter'13* and *SMS'13*. We train our model on a combined train and dev sets[4]. The Semeval-2014 re-uses the same training data and systems are evaluated on 5 test sets: two test sets from Semeval-2013 and three new test sets: *LiveJournal'14*, *Twitter'14* and *Sarcasm'14*. The datasets are summarized in Table 3.1.

n-grams	Manual			PMI		ML		Twitter'13
	M	B	N	hash	s140	raw	agg	
●								63.53
●	●							64.96 (+1.43)
●		●						66.74 (+3.21)
●			●					64.21 (+0.68)
●	●	●	●					67.44 (+3.91)
●	●	●	●	●				68.47 (+4.94)
●	●	●	●		●			69.08 (+5.55)
●	●	●	●	●	●			70.06 (+6.53)
●	●	●	●			●		69.47 (+5.94)
●	●	●	●				●	69.89 (+6.36)
●	●	●	●			●	●	70.93 (+7.40)
●	●	●	●	●	●	●	●	**71.32** (+7.79)
best Semeval'13 system								69.06

Table 3: Results on Semeval-2013 test set. Used feature sets: n-grams; features from *Manual* lexicons using MPQA (M), BingLiu (B) and NRCEmoticon (N) lexicons; PMI lexicon extracted from NRC-hashtag and Emoticon140 (s140) datasets; our ML lexicon using *raw* and aggregate (*agg*) features. The numbers in parenthesis indicate absolute improvement w.r.t. baseline *n-grams* model.

3.3 Results

We report the results on two runs of the Twitter Sentiment Analysis challenge organized by Semeval from 2013 and 2014.

3.3.1 Semeval-2013

The *n-grams* model includes word and character n-grams, negation and various surface form features as described in Section 2. We use this feature set as a yardstick to assess the value of adding features

[4] While in the real setting it is also possible to include additional weakly labeled data, e.g. Emoticon140, for training a model, we stick to the **constrained** setting of the Semeval tasks, where training is allowed only on the train and dev sets.

from various lexicons. Firstly, we note that using three manual lexicons: MPQA (M), BingLiu (B), and NRC (N) results in almost 4 points of absolute improvement. Notably, among all manual lexicons the *BingLiu* lexicon accounts for the largest improvement. Next, we explore the value of automatically generated lexicons using PMI scoring extracted from two large Twitter datasets: Emoticon140 (s140) and hashtag (hash). Both lexicons rely on PMI scoring formula to derive word-sentiment association scores. Adding features from these automatically generated lexicons results in further improvement over the *n-grams* feature set and yields F-score: 70.06.

Next, we explore the value of features derived from our ML based lexicon. We use the lexicon in two modalities: (i) including the raw scores (raw) of each lexicon entry (unigrams and bigrams) found in the given tweet; (ii) deriving aggregate features (*agg*) from the *raw* scores as described in Sec. 2; and (iii) using both. We note that the features from our ML-based lexicon yield superior performance to any of the PMI lexicons providing at least 2% gains and is even better when the two PMI lexicons are combined. Finally, adding the ML-based lexicon on top of the models including manual and auto lexicons provides the new state-of-the-art result on Semeval-2013 with an improvement of almost 8 points w.r.t. to the *basic* model. Our model achieves the score of 71.32 vs. 69.06 for the previous best system.

3.3.2 Semeval-2014

Table 4 shows that adding features from our ML-based vocabulary provides a substantial improvement over the previous best NRC system on 4 out of 5 test sets. Interestingly, we observe a strong drop on the Sarcasm'14 test set. One possible reason is that the labels for Emoticon140 corpus are inferred automatically using emoticons, which may strongly bias our model to incorrectly predict sentiment for those tweets containing sarcasm. With more than 40 systems participating in Semeval-2014 challenge, we note that the majority of systems perform well only on few test sets at once while failing on the others[5]. The performance of our system is rather high across all the test sets with an average rank of 3.4, which is the second best result in Semeval-2014 message-level task (the best system is from the NRC team with an ave-rank 2.4, whereas the closest follow up system has an ave-rank 6).

Table 4: Semeval-2014. Numbers in parenthesis is the absolute rank of a system on a given test set. Bold scores compares using our ML lexicon on top of the NRC system. Results marked with † are statistically significant at $p > 0.05$ (via the paired t-test).

System	NRC	NRC + ML lex.	best score
LJournal'14	75.28 (1)	**76.54**† (1)	74.84
SMS'13	66.86 (5)	**67.20** (5)	70.28
Twitter'13	70.06 (5)	**71.32**† (2)	72.12
Twitter'14	68.71 (6)	**70.51**† (2)	70.96
Sarcasm'14	**59.20** (1)	55.08 (7)	58.16
ave-rank	3.8	3.4 (2)	2.4 (1)

4 Conclusions

We demonstrated a simple and principled approach grounded in machine learning to construct sentiment lexicons. We show that using off-the-shelf machine learning tools to automatically extract lexicons greatly outperforms other automatically constructed lexicons that use pointwise mutual information to estimate sentiment scores for the lexicon items.

We have shown that combining our machine-learned lexicon with the previous best system yields state-of-the-art results in Semeval-2013 gaining over 2 points in F-score and ranking our system 2nd according to the average rank over the five test sets of Semeval-2014. Finally, our ML-based lexicon shows excellent results when added on top of the current state-of-the-art NRC system. While our experimental study is focused on Twitter, our method is general enough to be applied to sentiment classification tasks on other domains. In the future, we plan to experiment with constructing ML lexicons from larger Twitter corpora also using hashtags.

Recently, deep convolutional neural networks for sentence modelling (Kalchbrenner et al., 2014; Kim, 2014) have shown promising results on several NLP tasks. In particular, (Tang et al., 2014) showed that learning sentiment-specific word embeddings and using them as features can boost the accuracy of existing sentiment classifiers. In the future work we plan to explore such approaches.

[5] `http://alt.qcri.org/semeval2014/task9/`

References

Jrgen Bro and Heiko Ehrig. 2013. Automatic construction of domain and aspect specific sentiment lexicons for customer review mining. In *CIKM*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for Twitter: annotation, features, and experiments. In *ACL*.

Alex Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. In *CS224N Project Report, Stanford*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD*.

Sheng Huang, Zhendong Niu, and Chongyang Shi. 2014. Automatic construction of domain-specific sentiment lexicon based on constrained label propagation. *Knowl.-Based Syst.*

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *ACL*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Raymond Yiu-Keung Lau, Chun Lam Lai, Peter Bruza, and Kam-Fai Wong. 2011. Leveraging web 2.0 data for scalable semi-supervised learning of domain-specific sentiment lexicons. In *CIKM*.

Lizhen Liu, Mengyun Lei, and Hanshi Wang. 2013. Combining domain-specific sentiment lexicon with hownet for chinese sentiment analysis.

Micol Marchetti-Bowick and Nathanael Chambers. 2012. Learning for microblogs with distant supervision: Political forecasting with twitter. In *EACL*.

Saif Mohammad and Peter Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 39(3):555–590.

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Semeval*.

Preslav Nakov, Zornitsa Kozareva, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Theresa Wilson. 2013. In semeval-2013 task 2: Sentiment analysis in twitter. In *Semeval*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*.

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

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. In semeval-2014 task 9: Sentiment analysis in twitter. In *Semeval*.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP*.

Yen-Jen Tai and Hung-Yu Kao. 2013. Automatic domain-specific sentiment lexicon generation with label propagation. In *iiWAS*.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL*.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *ACL*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *EMNLP*.

Min Yang, Dingju Zhu, Rashed Mustafa, and Kam-Pui Chow. 2014. Learning domain-specific sentiment lexicon with supervised sentiment-aware lda. In *ECAI*.