# 10SENT: A Stable Sentiment Analysis Method Based on the Combination of Off-The-Shelf Approaches

**Philipe F. Melo**
*Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brazil. E-mail: philipe@dcc.ufmg.br*

**Daniel H. Dalip**
*Centro Federal de Educação Tecnológica de Minas Gerais, Brazil. E-mail: hasan@decom.cefetmg.br*

**Manoel M. Junior**
*Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brazil.*
*E-mail: manoelrmj@dcc.ufmg.br*

**Marcos A. Gonçalves**
*Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brazil.*
*E-mail: mgoncalv@dcc.ufmg.br*

**Fabrício Benevenuto**
*Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brazil.*
*E-mail: fabricio@dcc.ufmg.br*

**Sentiment analysis has become a very important tool for analysis of social media data. There are several methods developed, covering distinct aspects of the problem and disparate strategies. However, no single technique fits well in all cases or for all data sources. Supervised approaches may be able to adapt to specific situations, but require manually labeled training, which is very cumbersome and expensive to acquire, mainly for a new application. In this context, we propose to combine several popular and effective state-of-the-practice sentiment analysis methods by means of an unsupervised bootstrapped strategy. One of our main goals is to reduce the large variability (low stability) of the unsupervised methods across different domains. The experimental results demonstrate that our combined method (aka, 10SENT) improves the effectiveness of the classification task, considering thirteen different data sets. Also, it tackles the key problem of cross-domain low stability and produces the best (or close to best) results in almost all considered contexts, without any additional costs (e.g., manual labeling). Finally, we also investigate a transfer learning approach for sentiment analysis to gather additional (unsupervised) information for the proposed approach, and we show the potential of this technique to improve our results.**

Additional Supporting Information may be found in the online version of this article.

Received November 21, 2017; revised July 5, 2018; accepted July 14, 2018

 • Published online Month 00, 2018 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.24117

## Introduction

Online social media systems are places where people talk about everything, sharing their take or their opinions about noteworthy events. Not surprisingly, sentiment analysis has become an extremely popular tool in several analytic domains, but especially on social media data. The number of possible applications for sentiment analysis in this specific domain is growing fast. Many of them rely on monitoring what people think or talk about places, companies, brands, celebrities or politicians (Bollen, Mao, & Zeng, 2011; Hu & Liu, 2004; Oliveira, Cortez, & Areal, 2013).

Because of the enormous interest and applicability, many methods have been proposed in the last few years (e.g., SentiStrength (Thelwall, 2013), VADER (Hutto & Gilbert, 2014), Umigon(Levallois, 2013), SO-CAL (Taboada, Brooke, Tofiloski, Voll, & Stede, 2011), HSSWE (Wang and Xia., 2017)). In common, these methods are unsupervised[1] tools and have been applied to identify sentiments (i.e., positive, negative, and neutral) of short pieces of text such as tweets, in which the subject discussed in the text is known *a priori*. The importance of being unsupervised is that, in a real application of sentiment analysis, it can be very hard to get previous labeled data to train a classifier.

These tools are all currently acceptable by the research community as the state-of-the-art is not well-established yet. However, a recent effort (Ribeiro, Araújo, Gonçalves, Gonçalves, & Benevenuto, 2016) has shown that the prediction performance of these methods varies considerably from one data set to another. For instance, in that study, Umigon was ranked in the first position in five data sets containing tweets and was among the worst in a data set of news comments. Even among similar data sets, existing methods showed **low stability** in terms of their ranked positions. This suggests that existing unsupervised approaches should be used very carefully, especially for unknown data sets. More importantly, it suggests that novel sentiment analysis methods should not only be superior to existing ones in terms of predictive performance, but they should also be stable, that is, its relative prediction performance should vary minimally when used in many different data sets and contexts.

Accordingly, in this article, we propose 10SENT, an unsupervised learning approach for sentence-level sentiment classification that tells if a given piece of text (i.e. a tweet) is positive, negative, or neutral. To obtain better results than existing methods and guarantee stability across data sets, our approach exploits the combination of their classification outputs in a smart way. Our strategy relies on using a bootstrapped learning classifier that creates a training set based on a combination of answers provided by existing unsupervised methods. The intuition is that if most of the methods label an instance as positive, it is likely that it is positive, and it could be used to learn a classifier. This self-learning step provides to our method a level of adaptability to the current (textual) context, reducing prediction performance instability, a key aspect of an unsupervised approach.

We test our proposed approach by combining the top (best) ranked methods, according to a recent benchmark study (Ribeiro et al., 2016). We evaluate 10SENT with 13 gold standard data sets containing social media data from different sources and contexts. Those data sets consist of different sets of labeled data annotated for positive, negative and neutral texts from social networks messages and from comments of news articles, videos, websites, and blogs. Our approach showed to be statistically superior to (or at least ties with) the existing individual methods in most data sets. Therefore, our approach obtains the best mean rank position considering all data sets. Thus, our experimental results demonstrate that our combined method not only improves significantly the overall effectiveness in many data sets, but its cross-data set performance variability is minimal (maximum stability). In practical terms, this means that one can use our approach in any situation in which the base methods can be exploited, without any extra cost (since it is unsupervised) and without the need to discover the best method for a given context, and still obtain top-notch effectiveness in most situations.

We also show that 10SENT is superior to strong baseline combinations, such as a majority voting and combined lexical, with gains of up to 17% against such baselines. This highlights the importance of our bootstrapped strategy to improve the effectiveness of the sentiment classification task. It is important to stress that the number of methods to be combined is not necessarily restricted to 10. Our self-learning approach is very independent of the base methods, which means that it is highly extensible to incorporate any new additional method that can be created in the future.

To summarize, the main contribution of our work is an easily deployable and stable method that can produce results as good as or better than the best single method for most data sets in a completely unsupervised manner, being much superior than other unsupervised solutions such as majority voting and, in some cases, close to the best supervised ones. As far as we know, this is the first time nontrivial unsupervised learning used along with "state-of-the-practice" sentiment analysis methods to tackle the problem of providing an unsupervised approach able to obtain stable prediction performance across many domain-dependent data sets. The experimental results demonstrate that our combined method (aka, 10SENT) improves the effectiveness of the classification task. But more importantly, it tackles an important problem in the field—cross-domain low stability—10SENT produces the best (or close to best) results in almost all considered contexts, without any additional costs (e.g., manual labeling).

Finally, as a second contribution, we start an investigation into a important question of our research: whether we can "transfer" some knowledge to our method from a data set labeled with emoticons by Twitter users, which is easily available, meaning that no extra labeling effort is necessary. The main idea here is that such transfer of knowledge could provide additional (unsupervised) information to our method helping to improve it even further.

## Related Work

There are currently two distinct categories of sentiment analysis methods used in the social media domain: lexicon-based and those based on machine learning techniques. Machine learning methods comprise supervised classifiers trained with labeled data sets in which classes correspond to polarities (e.g. positive, negative or neutral) (Pang, Lee, & Vaithyanathan, 2002). One major challenge in this scenario is the difficulty in obtaining annotated data to train (supervised) methods due to issues such as cost

[1] They do not require explicit manually labeling data to be used in different domains.

and the inherent complexity of the labeling task. Accordingly, in here, we propose an unsupervised solution to deal with this sentiment analysis task.

Lexicon-based methods exploit lexical dictionaries, that is, word lists associated with sentiments or other specific features, which are usually not based on supervised learning. Some challenges with lexicon-based solutions including the construction of the lexicon itself (which is usually manually done) and difficulties in adapting for domains different from which they were originally designed.

Such issues naturally call for a combination of solutions that exploits their strengths while overcome their limitations. The idea of combining different sentiment analysis strategies, however, has been only recently explored. For instance, (Prabowo & Thelwall, 2009) proposes a new hybrid classification method based on the combination of different strategies. This work combines a rule-based classification and other supervised learning strategies into a new hybrid sentiment classifier. (Dang, Zhang, & Chen, 2010) combined machine learning and semantic-orientation that consider words expressing positive or negative sentiments.

(Zhang, Ghosh, Dekhil, Hsu, & Liu, 2011) explore an entity-level sentiment analysis method specific to the Twitter data. In that work, the authors combined lexicon and learning-based methods to increase the recall rate of individual methods. Differently from our work, this method was proposed for the entity-level, while ours focus on a sentence-level granularity. Similarly, Mudinas *et al.* (Mudinas, Zhang, & Levene, 2012) proposed *pSenti*, a method for sentiment analysis developed as a combination of lexicon and learning approaches for a different granularity level, the concept-level (semantic analysis of texts by means of web ontologies or semantic networks).

Moraes *et al.* (Moraes et al., 2013) investigated approaches to detect the polarity of FourSquare tips using supervised (SVM, Maximum Entropy and Naïve Bayes) and unsupervised (SentiWordNet) learning. They also investigate hybrid approaches, developed as a combination of the learning and lexical algorithms. All techniques were tested separately and combined, but the authors did not obtain significant improvements with the hybrid approaches over the best individual techniques for this domain.

As lexicon is usually a key part of many sentiment analysis methods, there are many efforts that focus on automatically building one or expanding an existing lexicon. In this direction, (Bravo-Marquez, Frank, & Pfahringer, 2015) present a supervised framework for expanding a sentiment lexicon for tweets. The authors trained a SVM classifier with a corpus of tweets labeled with semantic orientation using attributes based on part-of-speech tags and information computed from data streams containing emoticons. More recently, (Wang and Xia., 2017) proposed HSSWE, a method based on a sentiment-aware word representation learning approach. HSSWE uses a neural architecture to train a sentiment-aware word embedding in both document and word-level to enhance the quality of the sentiment lexicon. Those approaches usually require a supervised step and are used for online lexicon expansion in specific domains to discover terms potentially significant for an individual application. Our effort is complimentary to those as expanded lexicons can be used to create sentiment analysis methods for specific domains and combined with our proposed self-learning approach to provide an additional level of domain adaptation. *In any case, we use HHWSE as one of our baselines.*

There are other strategies for sentiment analysis that explore deep learning, well documented in a recent survey (Zhang, Wang, & Liu, 2018). Particularly, (Glorot, Bordes, & Bengio, 2011) uses a corpus from assorted domains to develop a deep learning approach that extracts meaningful representations of reviews to address the problem of domain adaptation. They explore transfer learning for reviews for aspect-level sentiment analysis by discovering relevant abstractions that are shared across different domains. Differently, reference (dos Santos & Gatti, 2014) uses a deep learning method to predict sentiment polarity on Twitter based on a convolutional neural network for sentiment analysis. They extract features from the character-level up to the sentence-level using word embedding to compute an opinion score for a given sentence. In a recent work, Felbo *et al.* (Felbo, Mislove, Søgaard, Rahwan, & Lehmann, 2017) proposed a method namely DeepMoji that uses millions of texts on Twitter containing emojis for training a deep learning model to learn representations of emotional content in texts. They used pre-trained classifiers to predict which emoji were originally part of the text. *We also use DeepMoji as a baseline.*

Overall, the increasing availability of huge amounts of data available might favor the development of deep learning methods on this field. In our work, we focus on complimentary framework to combine methods based on a self-learning ensemble step, aiming to provide performance stability across different data sets and domains. Ultimately, all forms of methods, including those based on deep learning, can be combined within our proposed framework.

Considering the combination of methods, Gonçalves *et al.* (Gonçalves, Dalip, Costa, Gonçalves, & Benevenuto, 2016) analyzed different data sets and considered supervised machine learning in the context of classifiers' ensembles. Their methodology also consisted of combining a set of different sentiment analysis methods in a "off-the-self" strategy to generate the ensemble method. Their results suggest that it is possible to obtain significant improvements with ensemble techniques depending on the domain. In here, we focus on a unsupervised solution enhanced with an automatic bootstrapping step.

Another effort on the ensemble direction, Gonçalves *et al.* (Gonçalves, Araújo, Benevenuto, & Cha, 2013) exploits the power of the combination of some of the state-of-the-art methods, showing that they can outperform individual methods. Their results show the potential of simple solutions such as majority voting, but the


DOI: 10.1002/asi

authors did not delve deep in more complex combination strategies.

Some approaches use a limited amount of labeled data (also known as weakly supervised classifier) to predict the sentiment in some domains. For example, (Siddiqua, Ahsan, & Chy, 2016) proposed a weakly supervised classifier for Twitter sentiment analysis. In this work, Naive-Bayes (NB) is combined with a rule-based classifier based on several publicly available sentiment lexicons to extract positive and negative sentiment words. After the rule-based classifier is applied, the NB is used to classify the remaining tweets as positive or negative. Deriu *et al.* (Deriu et al., 2017) also uses a weakly supervised approach to multi-language sentiment classification task. The developed method evaluates large amounts of weakly supervised data in various languages to train a multi-layer convolutional neural network, but its focus is on multilingual sentiment classification.

Wikisent, proposed by (Mukherjee & Bhattacharyya, 2012) also describes a weakly supervised system for sentiment analysis classification. They use text summarization focused on movie reviews domain to obtain knowledge about the various technical aspects of the movie. After that, the summary of the opinions are classified by using the SentiWordNet lexicon method.

To summarize, many efforts proposed supervised ensemble classifiers, but differently from those, we propose a novel approach by combining a series of "state-of-the-practice" existing methods in a totally unsupervised and in much more elaborated manner exploiting bootstrapping and (unsupervised) transfer learning. Another major difference of our effort is that we evaluate using multiple labeled data sets, covering multiple domains and social media sources. This is critical for an unsupervised approach given that the performance of the base methods varies significantly. As we shall see, our solution produced the most consistent results across all data sets and contexts.

## Combining Methods with 10SENT

Sentiment analysis can be applied to different tasks. We restrict our focus on combining those efforts related to detect the polarity (i.e. positivity, negativity, neutrality) of a given short text (i.e. sentence-level). In other words, given a set *S* of opinionated sentences, we want to determine whether each sentence *s in S* expresses a positive, negative or neutral opinion. We focus our effort on combining only unsupervised "off-the-shelf" methods. Our strategy consists of using the output label predicted by each individual method as input for a bootstrapping technique—a self-starting process supposed to proceed without external input. Next, we present the proposed technique.

Our **bootstrapping technique** is an unsupervised machine learning algorithm that uses the sentiment scores produced by each individual sentiment analysis method to create a training set for a supervised machine learning algorithm. With this algorithm, we can produce a final result regarding the sentiment of a sentence. Note that, we did not need to use any manually labeled data in order to produce the model.

We describe the method in Algorithm 1. Suppose we have access to a set of sentences $S = \{s_{tr0}, s_{tr1}, s_{tr3}, \ldots, s_{tr_n}\}$, which are candidates of being part of our training data. Our goal is to use the unlabeled data *S* to produce a training set *train* and, then, apply it to unseen sentences for which we want to predict (here represented as $test = \{s_{tst0}, s_{tst1}, s_{tst3}, \ldots, s_{tst_m}\}$), generating the set of predictions *P*. The training data *train* is represented by a set of pairs (*c*, *s*) where *c* is the class representing a sentiment (positive, negative or neutral) obtained by using the information of each sentiment analysis method described in Section  and *s* is a sentence represented by a set of features which, in our case, corresponds to the off-the-shelf sentiment methods' outputs.

**Algorithm 1** Bootstrapping Algorithm

**Require:** Minimum of Agreement $A$
**Require:** Minimum of Confidence $C$
**Require:** The set of $n$ sentences $S = s_{tr0}, s_{tr1}, s_{tr3}, ..., s_{tr_n}$, candidates of being part of our training data
**Require:** The set of $m$ sentences which we want to predict: $test = \{s_{tst0}, s_{tst1}, s_{tst3}, ..., s_{tst_m},\}$
Let $train$ = our training set represented by $(c, s)$ which $c$ is the target class and $s$ is the sentence
Let $P$ = our result which is represented by a set of triplet $(i, predicted\_class, confidence)$ which is the instance, the predicted class and its confidence
**for all** $s \in S$ **do**
  **if** $agree(s) \geq A$ **then**
    **Add** the pair $(agreeClass(s), s)$ to $train$
    **Remove** $s$ from $S$
**Create** a model $M$ using $train$
**Apply** the model $M$ in $S$ to obtain the predictions $P$
**for all** $(s, predicted\_class, confidence) \in P$ **do**
  **if** $confidence \geq C$ **then**
    **Add** the pair $(predicted\_class, s)$ to $train$
**Create** a model $M$ using $train$
**Apply** the model $M$ in $test$ to obtain the predictions $P$

The *test* is represented by a set of sentences $test = \{s_{tst0}, s_{tst1}, s_{tst3}, \ldots, s_{tst_n}\}$ and, the prediction *P*, contains a set of triplets (*s*, *predicted_class*, *confidence*) representing the sentence, the predicted class and the confidence (i.e., a score representing how confident the machine learning method is in its prediction), respectively.

We use the function *agree*(*s*), for each sentence *s*, which computes the Agreement level, in other words, the maximum number of sentiment analysis methods agreeing with each other regarding the sentiment in the sentence *s*. If this number is higher than the threshold *A*, we add the sentence *s* in the training set *train*, removing it from *S*. Note that, when adding a sentence to *train* we use the method *agreeClass*(*s*) in order to obtain the class *c* which will the sentiment assigned to *s*. Class *c* is obtained by using the class which has the majority of sentiment analysis methods assigned to the sentence *s*.

After doing this for all the sentences in *S*, only sentences for which we could not infer a label with enough agreement remain in *S*. Then, to increase our training data, we use our training set *train* to train a classification model and apply it to sentences in *S*, producing the predictions *P*. By doing so, we can use *P* to add more sentences to *train*. To avoid noise, we only add sentences for which the learned model produces a confidence higher than a threshold *C*. Finally, we retrain with the new set *train* and apply


DOI: 10.1002/asi

it to *test* to produce, for each sentence *s*, a single score *c* representing its final sentiment score.

As mentioned before, our approach consists of combining popular **"off-the-shelf" sentiment analysis methods** freely available for use. It is important to highlight that the number of methods to be combined is not necessarily restricted to ten. In fact, there is no limit on the number of methods we can include as part of our approach—thus, *we focus on the ones evaluated by (Ribeiro* et al., 2016*) as it provides the most recent and complete sentence-level benchmark of off-the-shelf sentiment analysis methods.*

There are few small adaptations on some methods to provide as output positive, negative and neutral decisions. For this, we have used the codes shared by the authors of (Ribeiro et al., 2016). More details about these implementations can be found there. The considered methods include: VADER (Hutto & Gilbert, 2014), AFINN(Nielsen, 2011), Opinion-Lexicon (Hu & Liu, 2004), Umigon (Levallois, 2013), SO-CAL (Taboada et al., 2011), Pattern.en (Smedt & Daelemans, 2012), Sentiment140 (Mohammad, Kiritchenko, & Zhu, 2013), EmoLex (Mohammad & Turney, 2013), Opinion Finder(Wilson et al., 2005), and SentiStrength (Thelwall, 2013). A brief description of these methods can also in found in (Ribeiro et al., 2016).

We also note that all methods exploit light-weight unsupervised approaches that rely on lexical dictionaries, usually implemented as a hash-like data structure. For this reason, the execution performance of our combined, as well as the individual methods, does not require any powerful hardware platform.

## Methods

### *Evaluation Metrics*

As **evaluation metric** we use the popular Macro-F1 score. Macro-F1 calculates the F1 score for each class separately and report the average of these scores for all classes. It is important in data sets with high skewness (as is the case here).

Our experiments were executed performing a 5-fold cross validation setup, with the best parameters for the learning methods found using cross-validation within the training set. This procedure was applied to all considered data sets. To compare the average results in the test sets of our experiments, we assess the statistical significance of our results by means of a paired *t*-test with 95% confidence.

We have checked the normality of the data distribution with a Shapiro–Wilk test before applying a Paired *t*-test. From output of test, the $p - value > 0.05$ we cannot reject the null hypothesis about the normal distribution implying that the data are not significantly different from normal distribution. To further investigate the normality verification and guarantee the data and the statistical significance of our results, we also performed Jarque Bera Test. As a result, we assume normality of the data distributions and we consider statistically significant only results in which the value of $p$ is less than 0.05 and any stated claim of superiority is based on these tests. Finally, we adapted the original outputs values of base methods to our corresponding polarities. An output equals to zero was considered as a neutral or "absence of opinion."

### *Data sets*

In our evaluation, we use 13 **data sets** of messages labeled as positive, negative and neutral from several domains, including messages from social networks, opinions and comments in news articles and videos. These data sets were kindly shared by the authors of (Ribeiro et al., 2016). We only consider those with three classes (positive, negative, and neutral). The number of messages vary from few hundreds to a few thousands. The data sets are usually very skewed, with usually one or two classes outnumbered the majority one by large margins. The median of the average number of phrases pier message is around 2 whereas the average number of words vary from around 15 to approximately 60. We refer the reader to their work for more details about the data sets. We emphasize that the diversity and amount of different data sets used in our evaluation allow us to accurately evaluate not only the prediction performance of the proposed method, but also measure the extent to which a method's result varies when it is tested for different social media sources.

### *Baselines*

As 10SENT explores the output of 10 other individual methods, Majority Voting is a natural baseline.[2] Voting is one of the simplest ways to combine several methods. By assuming that each individual method gives us a unique label as output for a sentence, the result of Majority Voting is the label that the majority of the base classifiers returned as output for that sentence.[3]

The major advantages of this approach are its simplicity and extensibility, i.e., it is very easy to include new (off-the-shelf ) methods. Also, no training data is necessary for this method, which fits well with our purpose of an unsupervised solution. On the other hand, majority voting is not as flexible as 10SENT in coping with all the diversity of the methods. This is because of the training phase of 10SENT that allows it to capture some idiosyncrasies of each one of them.

In addition to this baseline, we also explore a lexical combination, since several methods are lexical-based or have a dictionary of polarized terms for their prediction

[2] Notice that Weighted Majority Voting is not an option as a fair baseline, since to determine the weights we would need some type of supervision, something that our method does not exploit. In any case, we compare our solution to a version of the Weighted Majority Voting method in the 'Upperbound Comparison' Section.

[3] In this method, ties are possible. In this case, we assign a *Neutral* class to the sentence.


DOI: 10.1002/asi

task. As each method could fit well for different sources of data (social media, review, comments), a simple approach of merge their lexicon in a unique one is an alternative to compare with other results.

To accomplish this, we extracted the lexicons, containing a set of term and polarity score, from 9 out of the 10 methods we have used (except Umigon). Although some methods present extra information as (negation, intensifiers, POS tagging), we preferred to use just score as it is the common feature along all lists of terms.

Then, as they differ in the format and scale of measure, it was necessary a normalization step in all lexicon to fit them in a pattern ranging from -1 to +1 (those that only have the nominal category turned to [-1,0,1]). To merge them, two strategies were adopted: (a) Average Merged—when a term appeared in more than one different lexicon, its score was calculated by the average scores of these occurrences, varying the values from -1 to +1. (b) Majority Merged—When a term appeared in more than one lexicon, its score was made by vote among the occurrences to decide its polarity, in this lexicon the scores are discrete $[-1, 0, +1]$.

Finally, to obtain a unsupervised sentiment classification of a document, we sum up the scores of all sentiment terms in the document, according to the polarized terms in our merged lexicon. Also, for comparison, we use the Vader implementation but with our mean merged lexicon to assess the impact of the dictionary on classification.

Another approach considering lexicon-based methods is to automatically build a new lexicon for sentiment classification. HHSWE (Wang and Xia., 2017) does this by creating an expanded lexicon to enhance the quality of word embedding as well as the sentiment lexicon. Their method uses a neural architecture to train a sentiment-aware word embedding at both document and word level. HHWSE's authors shared with us a lexicon constructed by their method, trained with Sentiment140, one of the 10 methods of 10SENT. We used this expanded lexicon for sentiment prediction following the same methodology of the Merged Lexicon strategy. We sum individual polarities of each word to give the final score, as described in (Wang and Xia., 2017). For comparison purposes, we also use Vader implementation within this lexicon.

Last, we incorporate a deep learning method as one of our baselines for comparison. For this task, we used DeepMoji proposed in (Felbo et al., 2017). It is a method for sentiment analysis used primarily for predict emojis information for a text message. However, deep learning methods such DeepMoji usually need a large amount of training data to build a new model, hence we applied the pre-trained neural network model available by authors trained with millions of tweets containing emojis in order to adopt this method as an unsupervised tool. This pre-trained model is used to predict top 10 emojis for messages in our data set and each emoji is predicted with a probability. We map some of these emojis as positive and negative and finally sum these scores to define final prediction following a threshold for add an emoji to the result.

## 10SENT Configuration

Here, we discuss some decisions taken during the development of 10SENT. We also start an investigation on issues related to transfer learning for sentiment analysis, showing the potential of this technique to improve our results.

### *Choice of the Classifier*

10SENT is an unsupervised machine learning method as it does not exploit manually labeled data, only the agreement among the base methods. Given that the bootstrapping process adds a set of instances with high confidence into a training set, it is possible to perform a learning step exploiting such data in the usual format training/validation. Because of this, there is a need to investigate which classifier fits better this application. Thus, we perform a series of tests with our method using different classification algorithms to choose the best one for this task. In all these tests, we used all 10 methods of 10SENT. We tested three different and widely used algorithms in our approach: *Support Vector Machine* (SVM) (Chang & Lin, 2011), *Random Forest* (RF) (Breiman, 2001) and *k-Nearest Neighbors* (KNN) (Pedregosa et al., 2011). Here we used the implementations of RF and KNN provided in scikit-learn[4] and for SVM, we use LibSVM[5] package. Specifically, we use a *radial basis function* (RBF) kernel with a grid search for the best parameters. Overall, Random Forests produced the best results in most data sets, being the final choice for our bootstrapping method.

### *Choice of Number of Methods*

To verify the coherence with results obtained by Majority Voting, we perform a test with different number of methods used in the combination. In this test, we want to check how the addition of a method can impact the outcome. We evaluated results of 10SENT combining from 3 up to 10 methods. In these experiments, we included from the best to the worst method in each data set, according to (Ribeiro et al., 2016). We noted that adding a new method improves the overall results, but it is possible to note that improvements get smaller with new inclusions. Thus, after a certain number, the gain is minimal. Therefore, we fixed 10 as a good choice to number of methods in 10SENT core.

### *Choice of Parameters*

In our method, we need to define two important parameters: the agreement and the confidence level. Accordingly,

[4] Available at http://scikit-learn.org/stable/index.html
[5] Available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/


DOI: 10.1002/asi

we performed a study to better understand how our method performs when varying such parameters. In more details, the first tested parameter was the minimum number of agreements among the methods we should use in the first round of classification (Agreement Level).

Table 1 shows Macro-F1 results for each number of agreements. As we have a total of ten base methods, this table shows bootstrapping results when we use instances that at least 4 or more methods agree with each other, 5 and so on. We did not show results with less than 3 agreements since there were no instances in such scenario.

As we can see in this table, the extreme cases of agreement or disagreement produce the worst results. There is a small amount of instances with 100% of agreement, which harms the training of the algorithm. On the other hand, when the agreement is very low, there is a lot of noise in the training data. In sum, the Agreement level represents a tradeoff between the amount of available data for training and the amount of noise.

The second parameter was the RF confidence Level, defined in Algorithm 1. as the constant *C*. The confidence level is the confidence ratio of the Random Forest algorithm in its predictions. We use this in order to add more data to train during the bootstrapping step. Then, a similar variation of this parameter was tested, as shown in Table 2.

As a final conclusion of these experiments, we arrive at a value of 7 for agreement and 0.7 for confidence, in most data sets, as a way to achieve the "best" balance between quantity and quality for the training data.

### *Bag of Words Versus Predictions*

After the definitions of the parameters for the classification process, additional features can be extracted and combined with the predictions of other methods to improve results. One example is the text of messages itself. With the text, we can extract the Bag of Words representation (BoW) of the sentences included in the training. We used the traditional TF-IDF representation calculated for each sentence in each data set. This was concatenated with the results of each method, as in the traditional 10SENT.

In Table 3 we find the results comparing the use of these different sets of features, the predictions outputted by all base methods and bag of words. Here, we used all best parameters discovered in previous sections, including the random forest classifier. Note that the combination of these two set of features improves results compared with each single one separately. Although BoW individually presented better results in a few data sets it is not the best in all of them and alone, which suggests that using both sets of features is the best option for 10SENT. In the next experiments, we always use this joint representation (BoW + BaseMethods) when me mention 10SENT.

### *Transfer Learning Analysis*

Finally, we evaluate whether it is possible to explore some "easily available" knowledge from an external source. We do so by exploring data sets in which messages are labeled with "emoticons" by the systems' users themselves. To use an approach that transfers knowledge from one task to another, it is usually necessary to map characteristics from the source problem into the target one, identifying similarities and differences. Next we detail how we transfer knowledge from existing emoticons in the data sets to the task of sentence-level sentiment analysis.

Emoticons are representations of an expression in a faced-look set of characters. They became very popular nowadays and even the Oxford English Dictionary has recently chosen an "emoji" as word of year (in 2015) because of its notable and massive use around the world. In our case, they are used to give us an idea of feelings in the text, like happiness or sadness.

Previous works have demonstrated that such messages, though not available in large volumes, are very precise. In other words, labeling with emoticons used by the final user

TABLE 1. Comparative table of results (F1) for 10SENT bootstrapping by different agreement levels among the base methods in classification

| | #Concordants | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Data set | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| english_dailabor | 69.58 | 69.18 | 69.23 | 68.50 | 66.91 | 64.81 | 59.58 | 60.75 |
| aisopos_ntua | 60.93 | 60.54 | 56.87 | 57.74 | 64.14 | 59.65 | 54.58 | 58.93 |
| tweet_semevaltest | 63.54 | 63.58 | 63.64 | 63.47 | 63.82 | 61.56 | 59.36 | 59.15 |
| sentistrength_twitter | 56.75 | 58.32 | 59.13 | 58.34 | 57.58 | 55.35 | 54.27 | 57.44 |
| sentistrength_youtube | 55.63 | 55.22 | 55.67 | 56.39 | 56.65 | 55.44 | 54.69 | 54.50 |
| sanders | 56.23 | 55.72 | 55.94 | 55.64 | 55.27 | 52.73 | 50.77 | 48.14 |
| sentistrength_digg | 51.13 | 51.29 | 53.86 | 54.58 | 51.51 | 50.98 | 48.06 | 51.83 |
| sentistrength_myspace | 46.76 | 48.30 | 48.71 | 50.31 | 50.52 | 51.02 | 54.34 | 39.44 |
| sentistrength_rw | 48.96 | 50.09 | 46.62 | 49.73 | 49.00 | 46.52 | 46.03 | 47.58 |
| sentistrength_bbc | 49.15 | 50.62 | 46.95 | 47.41 | 46.31 | 45.04 | 45.75 | 46.57 |
| debate | 45.61 | 45.82 | 43.69 | 43.97 | 44.47 | 44.99 | 43.57 | 43.10 |
| nikolaos_ted | 46.29 | 44.71 | 48.13 | 46.43 | 46.97 | 47.52 | 44.50 | 47.24 |
| vader_nyt | 36.13 | 36.19 | 36.32 | 37.35 | 38.21 | 36.89 | 32.54 | 34.02 |


DOI: 10.1002/asi

TABLE 2. Comparative table of results (F1) for 10SENT by different confidence levels added to training in classification

| | Confidence level | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Data set | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
| english_dailabor | 67.80 | 66.82 | 67.28 | 67.50 | 67.65 | 67.63 | 67.57 | 67.64 |
| aisopos_ntua | 64.53 | 64.19 | 63.81 | 64.95 | 66.21 | 60.89 | 57.57 | 57.36 |
| tweet_semevaltest | 62.56 | 62.88 | 62.75 | 62.65 | 62.82 | 63.33 | 63.21 | 63.02 |
| sentistrength_twitter | 58.11 | 58.08 | 58.47 | 59.24 | 58.14 | 56.75 | 57.71 | 56.12 |
| sentistrength_youtube | 56.64 | 56.50 | 55.59 | 55.55 | 56.28 | 56.93 | 56.86 | 56.04 |
| sanders | 55.22 | 55.88 | 54.83 | 54.62 | 55.65 | 54.70 | 53.81 | 53.78 |
| sentistrength_myspace | 51.86 | 51.77 | 52.89 | 52.46 | 55.22 | 52.75 | 54.51 | 53.03 |
| sentistrength_digg | 51.75 | 50.98 | 53.15 | 51.93 | 52.59 | 51.11 | 50.86 | 50.85 |
| sentistrength_rw | 46.46 | 45.60 | 50.24 | 49.61 | 50.84 | 50.59 | 45.77 | 47.30 |
| sentistrength_bbc | 45.67 | 45.75 | 46.16 | 45.17 | 47.19 | 48.00 | 46.01 | 48.24 |
| debate | 45.77 | 45.95 | 46.10 | 46.53 | 45.48 | 45.26 | 43.94 | 43.93 |
| nikolaos_ted | 45.80 | 44.70 | 43.05 | 46.42 | 44.76 | 46.00 | 45.79 | 43.48 |
| vader_nyt | 38.53 | 38.33 | 38.49 | 38.65 | 37.69 | 37.27 | 36.94 | 36.10 |

TABLE 3. Results of 10SENT using different set of features for Random Forest

| Data set | Bag of words | BaseMethods | BoW + BaseMethods |
|---|---|---|---|
| english_dailabor | 68.4 | 67.1 | **72.4** |
| aisopos_ntua | **72.3** | 62.0 | 69.9 |
| tweet_semevaltest | 58.3 | 62.8 | **65.2** |
| sentistrength_twitter | 58.8 | 59.1 | **61.2** |
| sentistrength_youtube | 56.6 | 56.1 | **58.7** |
| sanders | **61.5** | 54.1 | 56.4 |
| sentistrength_myspace | 50.2 | **52.3** | 52.2 |
| sentistrength_digg | 45.4 | 50.1 | **50.6** |
| nikolaos_ted | **51.3** | 45.9 | 49.0 |
| debate | **57.1** | 45.9 | 47.1 |
| sentistrength_rw | 48.3 | **48.5** | 45.5 |
| sentistrength_bbc | 34.8 | **45.5** | 43.8 |
| vader_nyt | 28.0 | 38.9 | **39.2** |

indeed provide trustful information about the polarity of message. Accordingly, in these experiments, we used the "rules of thumb" suggested in (Gonçalves et al., 2013) to translate emoticons into polarities.

As one might expect, the fraction of messages containing emoticons is very low compared to the total number of messages. As we can see in Table 4 emoticons appeared just in a very small amount of instances (observed in the coverage column). In spite of that, the accuracy of emoticons is often very precise to distinguish polarity of sentiment, reaching more than 90% in "nikolaos_ted" data set. This is also in agreement with previous efforts (Gonçalves et al., 2013).

Our ultimate goal here is to extract some information about the text of those messages to our classification step. For this, we incorporate into the training data these instances labeled with emoticons extracted from the respective data sets.

To compare the effect of transfer learning from emoticons, we separated it in three different experiments: first with our traditional 10SENT; next we used just emoticon labels to create the training, without our majority voting predictions; then we combined these two to check the impact of emoticons in our method. Results of this experiment can be seen in Table 5. We can see that improvements of up to 6% (e.g., in case of the sentistrength_myspace data set) can be obtained in terms of Macro F1, with no significant losses in most data sets and with no extra (labeling) cost. Thus, this approach represents an interesting opportunity to provide to the user some help in terms of labeling effort.

## Comparative Results

### *Baseline Comparison*

Results for all baseline methods are shown in Table 6. As we can note, Majority Vote performed better than the others baselines in most cases (10 wins out of 12, to be exact).

Considering the merged lexicon approaches, we can note that the merge with average of co-occurrences presented better results than the majority merged approach. However, both, as well as the Vader implementation using a merged lexicon, showed results worse than the Majority Voting baseline in terms of F1-score. This may be due to the fact that each dictionary was built with distinctive purposes and independent metrics.

Regarding the results of the expanded lexicon, they are usually better than the merged strategies, with two overall


DOI: 10.1002/asi

TABLE 4. Accuracy and coverage of emoticons in training experiments for all data sets

| | Accuracy | Coverage |
|---|---|---|
| nikolaos_ted | 0.919 | 0.014 |
| sentistrength_myspace | 0.800 | 0.091 |
| aisopos_ntua | 0.787 | 0.526 |
| tweet_semevaltest | 0.693 | 0.071 |
| english_dailabor | 0.687 | 0.064 |
| sentistrength_youtube | 0.686 | 0.085 |
| sentistrength_twitter | 0.627 | 0.097 |
| sentistrength_rw | 0.619 | 0.148 |
| sentistrength_digg | 0.600 | 0.028 |
| sanders | 0.359 | 0.045 |
| debate | 0.339 | 0.015 |
| sentistrength_bbc | 0.173 | 0.006 |
| vader_nyt | - | - |

TABLE 5. Macro-F1 results for experiments on 10SENT using Transfer Learning

| | 10Sent | Emoticons | 10Sent + Emoticons |
|---|---|---|---|
| aisopos_ntua | 69.91 | 35.48 | **73.61** |
| english_dailabor | 70.62 | 25.57 | **72.02** |
| tweet_semevaltest | 64.78 | 18.06 | **65.13** |
| sentistrength_twitter | 62.17 | 22.93 | **62.87** |
| sentistrength_youtube | 57.06 | - | **59.36** |
| sanders | 56.19 | 11.83 | **56.78** |
| sentistrength_digg | 51.91 | - | **52.22** |
| sentistrength_myspace | 50.22 | - | **53.20** |
| nikolaos_ted | 47.97 | - | **48.97** |
| debate | 47.18 | - | **47.37** |
| sentistrength_rw | **47.15** | - | 45.25 |
| sentistrength_bbc | **43.76** | - | 43.18 |
| vader_nyt | **39.81** | - | 39.01 |

wins in case of HSSWE Sum (sentistregth_myspace and sentistregth_rw), indicating that new terms found by expansion had some value for the classification task.

We can also see in Table 6 that the deep learning approach (i.e., DeeMoji) achieved the best results in one data set (aisopos_ntua) despite adopting a pre-trained model. This shows the potential of this type of solution, but also evince that the need of a lot of training data in the right context and adaptation, given the not "so-strong" results in the other data sets.

To summarize, Majority Voting, despite simple, was the strongest baseline when compared to merged, expanded and a pretrained deep learning baselines. From now on, we will focus on this baseline.

TABLE 6. Results for all baselines methods ("*" indicates values that the difference compared to second best method was not statistically significant and "$^{\Delta}$" are the values statistically superior to the second best result)

| | Voting | Deep learning | Expanded lexicon | | Merged lexicon | | |
|---|---|---|---|---|---|---|---|
| Data set | Majority Vote | Pre-trained DeepMoji | HSSWE + Vader | HSSWE Sum | Merged +Vader | Merged Majority | Merged Mean |
| english_dailabor | **68.16**$^{\Delta}$ | 54.70 | 51.15 | 51.33 | 48.68 | 40.38 | 47.61 |
| tweet_semevaltest | **62.64**$^{\Delta}$ | 51.76 | 46.95 | 48.49 | 41.59 | 27.48 | 39.90 |
| aisopos_ntua | 59.45 | **61.40**$^{\Delta}$ | 53.55 | 56.09 | 54.51 | 42.40 | 43.71 |
| sentistrength_twitter | **58.89**$^{\Delta}$ | 50.77 | 50.53 | 51.81 | 47.37 | 41.58 | 45.17 |
| sanders | **54.75**$^{\Delta}$ | 46.27 | 45.69 | 44.96 | 47.61 | 38.48 | 47.06 |
| sentistrength_youtube | **54.60**$^{\Delta}$ | 49.65 | 51.21 | 50.95 | 46.81 | 43.76 | 41.54 |
| sentistrength_myspace | 51.56 | 50.67 | 53.38 | **53.92***| 42.76 | 36.57 | 40.47 |
| sentistrength_digg | **51.50**$^{\Delta}$ | 44.58 | 45.19 | 47.71 | 46.31 | 44.75 | 42.62 |
| sentistrength_rw | 48.34 | 42.93 | 50.25 | **52.81*** | 32.21 | 24.84 | 28.72 |
| nikolaos_ted | **47.17**$^{\Delta}$ | 36.92 | 38.94 | 37.68 | 39.49 | 41.62 | 39.94 |
| sentistrength_bbc | **45.19**$^{\Delta}$ | 38.64 | 34.10 | 34.51 | 40.99 | 36.29 | 39.78 |
| debate | **43.99**$^{\Delta}$ | 37.63 | 39.98 | 39.77 | 39.91 | 38.21 | 40.80 |
| vader_nyt | **37.19*** | 33.33 | 32.74 | 33.20 | 29.97 | 32.87 | 30.65 |


DOI: 10.1002/asi

*Comparison with Base Methods*

We now turn our attention to the comparison between 10SENT, the "strongest" baseline (Majority voting) and the base methods. We should point out that in these comparisons, a mention to 10SENT corresponds to the results obtained with the best unsupervised configuration found in the previous analyses, in other words, the original 10SENT representation (methods' decisions) along with the Bag Of Words and the transfer learning.

We can observe in Figure 1 that our method has a higher Macro-F1, above the baselines, in most data sets. In fact, 10SENT is the best method in 7 out of 13 data sets and it is close to the top of the rank in several others. This is also reflected in the Mean Rank, shown in Table 7, confirming that 10SENT is the overall winner across all tested data sets.

In fact, 10SENT can be considered as the most *stable method* as it produces the best (or close to the best) results in most data sets in different domains and applications. In other words, by using our proposed method, one can almost always guarantee top-notch results, at no extra cost, and without the need to discover the best method for a given context/dataset/domain.

*UpperBound Comparison*

For analysis purposes, we also perform a comparison of 10SENT with some "uppperbound" baselines which use some type of privileged information, most notably the real label of the instances in the training set, an information not available to us. The idea here it to understand how far our proposed unsupervised approach is from the ones that exploit such information as well as to understand the limits to what we can achieve with an unsupervised solution.

The first "upperbound" baseline is a fully supervised approach which uses all the labeled information available in the training data. As normally done in fully supervised approaches, the parameters of the RF algorithm are determined using a validation set.

The second baseline is an *Exhaustive Weighted Majority Voting* method that uses the real labels of messages of the data sets to find the best possible linear combination of weights for each base method. Differently from the Majority Voting baseline, in which all methods have the same weight, in this approach, each individual base method has a different weight, so that the influence of each one in the final classification is different.

The weights for each method are found by means of an exhaustive search in each (training portion of the) data set. That is, for each data set we found the "close-to-ideal" weights that would lead to the best possible result when combining the exploited base methods. Then, for each method, a weight was associated with its output and, finally, the class with the highest weight was marked as the resulting label of each instance. This search was performed in exhaustive mode, that is, we evaluate every possible combination, seeking to maximize the Macro-F1 in each data set. During the experiments, we limited the search to five different weights in the range $[0 - 1] : W = \{0, 0.25, 0.5, 0, 75, 1\})$ to estimate "close-to-best" results, while maintaining feasible computational costs.

Table 8 shows the average weights and corresponding standard deviation for each method in some data sets, but for all the results are similar. We can see that most methods have different behaviors in different data sets

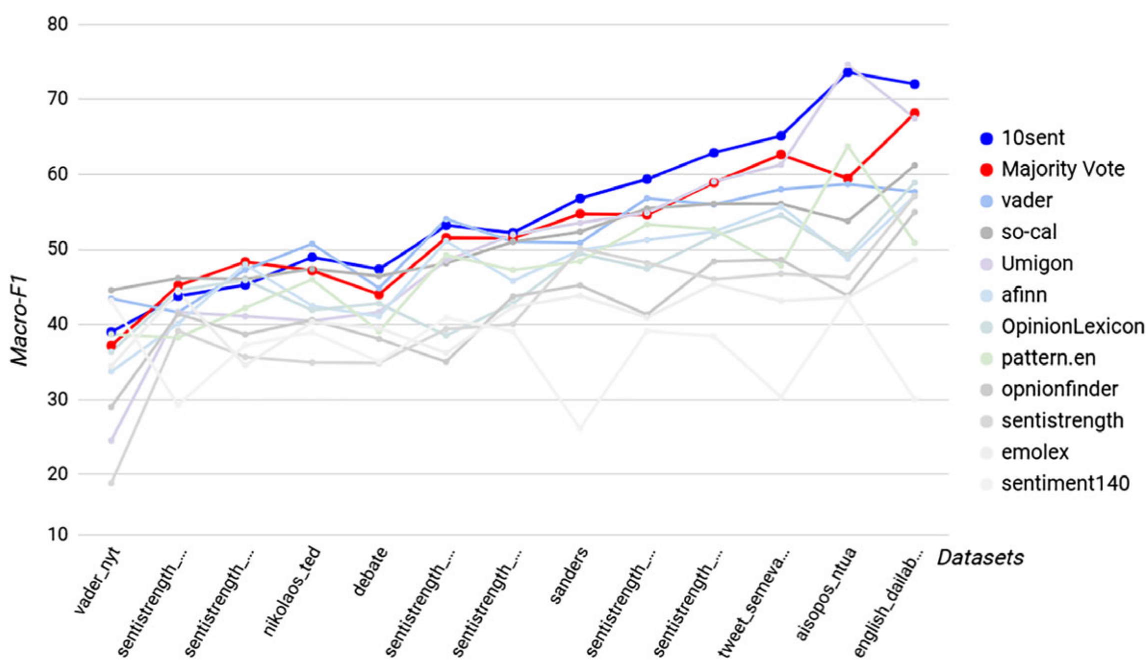


FIG. 1. Macro-F1 results of 10SENT compared with each individual base method for all data sets. [Color figure can be viewed at wileyonlinelibrary.com]


DOI: 10.1002/asi

TABLE 7. Mean Rank of methods for all data sets

| Method | Mean rank | POS | Deviation |
|---|---|---|---|
| 10SENT | 2.154 | 1 | 1.457 |
| Majority Voting | 3.154 | 2 | 1.350 |
| Vader | 3.692 | 3 | 1.814 |
| SO-CAL | 3.769 | 4 | 1.717 |
| Umigon | 4.923 | 5 | 2.921 |
| Afinn | 6.615 | 6 | 1.820 |
| OpinionLexicon | 6.923 | 7 | 1.900 |
| pattern.en | 7.000 | 8 | 2.287 |
| OpinionFinder | 9.308 | 9 | 1.136 |
| Sentistrength | 9.846 | 10 | 1.747 |
| Emolex | 9.923 | 11 | 2.055 |
| Sentiment140 Lexicon | 10.692 | 12 | 2.493 |

(implied by the large deviations). In other words, the same method may have a huge variance in effectiveness in different data sets, which precludes the use of a single unique method for all cases. Despite this, we can observe that some methods have clearly a higher average than others even with this high deviation.

Finally, the third "upperbound" baseline is the best single base method in each data set. Since the base methods are unsupervised "off-the-shelf" ones, we determine the best method for each data set also using the labels in the training sets. It is also an "upperbound" because the best method cannot be determined, in advance, without supervision, that is, a training set.

The results of those upperbounds are shown in Table 9. For comparative purposes we also included in this table the results of the unsupervised Majority Voting. As before, all results correspond to the average performance in the 5 test sets of the folded cross-validation procedure using 10SENT with its best configuration including Bag of Words and Transfer Learning.

Values marked with "*" in this table indicate that the difference was not statistically significant when compared to 10SENT in a paired-test with 95% confidence. Results reported with "$^{\Delta}$" are those statistically better than those of 10SENT. On the other hand, our method demonstrated to be statistically superior to the ones whose values are marked with "$^{\nabla}$".

As highlighted before, 10SENT is tied or better than the traditional majority voting in most data sets, being statistically superior in 7 out of 12 cases, tying in other 5 and losing only in one data set (sentistrength_rw). Gains can achieve up to 23.8% against this baseline. When compared to the best individual method in each data set, 10SENT wins (4 cases) or ties (7 cases) in 11 out of 13 cases, a strong result. This shows that 10SENT is a good and consistent choice among all available options, independently of which data set is used.

When compared to supervised Exhaustive Weighted Majority Voting, a first observation is that, as expected, it is always superior to the simple Majority Voting. Although we cannot beat this "upperbound" baseline, we tie with it in four data sets (sentistrenth_youtube, sentistrength_twitter, tweet_semevaltest, english_dailabor) and get close results in others such as aisopos_ntua, sanders and debate. This with no cost at all in terms of labeling effort.

Regarding the strongest upperbound baseline—Fully Supervised—an interesting observation to make is that in some data sets its results get very close to those of the Exhaustive Weighted Majority Voting, even losing to it in two (sentistrength_bbc, vader_nyt). This is a surprising result, meaning that the combination of both strategies is also an interesting venue to pursue in the future. When comparing this baseline to 10SENT, as expected, we can also not beat it, but can tie with it in two data sets and get close results in others, mainly in those cases in which our method was a good competitor against Exhaustive Weighted Majority Voting. We consider these very strong results.

For a deeper understanding of the results, Table 10 shows the set size of 10SENT used to train the classifier before and after the bootstrapping step (lines 9-11 of Algorithm 1). As we can see, the majority voting heuristics selects a relative large amount of training data from the original data sets. This may explain some of the good results obtained in our experiments, since the classifiers have a reasonable amount of data to be trained with.

However, this is only part of the story. One question that remains to be answered is: "What is the **quality** of the automatically labeled training set". We can answer this question by looking at the columns "Accuracy" in the

TABLE 8. Average and deviation for weights found during Exhaustive Weighted Vote step

| | Weights | | | | |
|---|---|---|---|---|---|
| | pattern.en | sentiment140 | emolex | opinionfinder | sentistrength |
| Avg. Weight | 0.28 | 0.37 | 0.26 | 0.40 | 0.85 |
| Std. Deviation | 0.25 | 0.28 | 0.29 | 0.31 | 0.28 |

TABLE 9. Results in terms of Macro-F1 comparing 10SENT with all other evaluation methods ("*" indicates values that the difference was not statistically significant compared to the 10SENT; "$^\nabla$" are values that 10SENT wins and "$^\Delta$" are the values statistically superior to the 10SENT result)

| | Fully supervised | Exhaustive weighted majority voting | Best individual | Majority voting | 10SENT |
|---|---|---|---|---|---|
| aisopos_ntua | **76.64**$^\Delta$ | 75.8$^\Delta$ | 74.58* | 59.45$^\nabla$ | 73.61 |
| english_dailabor | **75.63**$^\Delta$ | 71.9* | 67.47$^\nabla$ | 68.16$^\nabla$ | 72.02 |
| tweet_semevaltest | **66.77*** | 65.5* | 61.27$^\nabla$ | 62.64$^\nabla$ | 65.13 |
| sentistrength_twitter | **66.14**$^\Delta$ | 62.9* | 59.05$^\nabla$ | 58.89$^\nabla$ | 62.87 |
| sentistrength_youtube | **61.77**$^\Delta$ | 60.6* | 56.81$^\nabla$ | 54.60$^\nabla$ | 59.36 |
| sanders | **62.76**$^\Delta$ | 58.0$^\Delta$ | 53.52* | 54.75* | 56.78 |
| sentistrength_myspace | 57.47$^\Delta$ | **57.8**$^\Delta$ | 54.05* | 51.56* | 53.20 |
| sentistrength_digg | **59.52**$^\Delta$ | 57.3$^\Delta$ | 51.98* | 51.50* | 52.22 |
| nikolaos_ted | **57.43**$^\Delta$ | 56.1$^\Delta$ | 50.76$^\Delta$ | 47.17* | 48.97 |
| debate | **58.75**$^\Delta$ | 49.1$^\Delta$ | 46.45* | 43.99$^\nabla$ | 47.37 |
| sentistrength_rw | **53.52**$^\Delta$ | 52.2$^\Delta$ | 47.97* | 48.34$^\Delta$ | 45.25 |
| sentistrength_bbc | 44.00* | **51.8**$^\Delta$ | 46.17* | 45.19* | 43.18 |
| vader_nyt | 46.87$^\Delta$ | **51.9**$^\Delta$ | 44.56$^\Delta$ | 37.19$^\nabla$ | 39.01 |

table. This metric calculates the proportion of correctly assigned labels in the training sets when compared to the "real labels". For a considerable number of data sets, the accuracy is relatively high, between 0.6-0.8. In fact, the cases in which 10SENT gets closer to the fully supervised method correspond to those in which the accuracy in the training is higher. We can also see that after the bootstrapping, in general the accuracy in the training drops a bit, which is natural since the heuristics is not perfect, but this is compensated by the increase in training size, resulting in a learned model that generalizes better.

Finally, we can see that the absolute results of the best overall method in each data set are still not very high (maximum of 76%), which shows the difficulty of the sentiment analysis task and that there is a lot of room for improvements.

## Conclusions and Future Work

We presented a novel unsupervised approach for sentiment analysis on sentence-level derived from the combination of several existing "off-the-shelf" sentiment analysis methods. Our solution was thoroughly tested in a wide and diversified environment. We cover a vast amount of methods and labeled data sets from different domains. The key advantage of 10SENT is that it is a first step towards fixing one major issue in this field—the large variability of the methods across domains and data sets. Our experimental results show that our self-learning approach has the lowest prediction performance variability because of its ability to slightly adapt to different contexts. This is a crucial issue in an area in which researchers are mostly interested in using an "off-the-shelf" method to different contexts. Our approach is also easily expandable to include any new developed unsupervised method. Our experimental results also show that 10SENT achieves good effectiveness compared to our baselines. 10SENT was superior to all existing individual methods and also obtained better results than the traditional majority voting, with gains of up to 17.5%. In an upperbound comparison, we saw that 10SENT can get close to the best supervised results. Finally, our analysis on transfer learning shows us the possibility of adapting

TABLE 10. Set size, "noise"(indicated by accuracy) and Macro-F1 values to 10SENT training sets without bootstrapping and including bootstrapping step

| | 10SENT | | | | | |
|---|---|---|---|---|---|---|
| | Majority voting | | | Bootstrapping | | |
| | Set size | Accuracy | F1 | Set size | Accuracy | F1 |
| english_dailabor | 1,999 | 0.858 | 70.61 | 2165 | 0.826 | 70.62 |
| aisopos_ntua | 215 | 0.762 | 71.67 | 238 | 0.734 | 69.91 |
| tweet_semevaltest | 2,781 | 0.796 | 65.04 | 3139 | 0.757 | 64.78 |
| sentistrength_twitter | 2,042 | 0.706 | 63.06 | 2238 | 0.665 | 62.17 |
| sentistrength_youtube | 1,688 | 0.660 | 58.37 | 1837 | 0.645 | 57.06 |
| sanders | 1,760 | 0.762 | 55.94 | 1929 | 0.758 | 56.19 |
| sentistrength_digg | 474 | 0.630 | 49.32 | 519 | 0.615 | 51.91 |
| sentistrength_myspace | 488 | 0.641 | 49.34 | 535 | 0.645 | 50.22 |
| debate | 1,422 | 0.508 | 47.30 | 1620 | 0.530 | 47.97 |
| nikolaos_ted | 329 | 0.606 | 47.11 | 370 | 0.556 | 47.18 |
| sentistrength_rw | 417 | 0.618 | 43.12 | 471 | 0.601 | 47.15 |
| sentistrength_bbc | 376 | 0.687 | 37.91 | 418 | 0.661 | 43.76 |
| vader_nyt | 2,222 | 0.363 | 40.44 | 2563 | 0.366 | 39.81 |


DOI: 10.1002/asi

the method to include more strategies that can lead to better results. We also release our codes and data sets to the research community as part of known sentiment analysis benchmark systems (Araújo et al., 2016).

As future work, we intend to better explore weighting strategies as well choosing other setups for different scenarios. Particularly, we plan to investigate whether machine learning models trained with specific data sets can be used as input for the ensemble step as well as which trained data sets are better suited for our method. Additionally, we will explore other syntactic and semantic aspects of the text of the messages to improve results.


## Acknowledgments

This research is partially funded by Google, the Brazilian National Institute of Science and Technology for Web Research (MCT/CNPq/INCT Web Grant Number 573871/2008-6), MASWeb (grant FAPEMIG/PRONEX APQ-01400-14), and by the authors individual grants from CNPq and FAPEMIG.


## References


Araújo, M., Diniz, J.P., Bastos, L., Soares, E., Júnior, M., Ferreira, M., Ribeiro, F., and Benevenuto, F. (2016). iFeel 2.0: A multilingual benchmarking system for sentence-level sentiment analysis. In Proceedings of 10th International AAAI Conference on Web and Social Media (ICWSM'16). Cologne, Germany.

Bollen, J., Mao, H., & Zeng, X.-J. (2011). Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1–8.

Bravo-Marquez, F., Frank, E., and Pfahringer, B. (2015, June). Positive, negative, or neutral: Learning an expanded opinion lexicon from emoticon-annotated tweets. In 24th International Joint Conference on Artificial Intelligence (IJCAI) (pp. 1229–1235). Palo Alto, California.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3), 27.

Dang, Y., Zhang, Y., & Chen, H. (2010). A lexicon-enhanced method for sentiment classification: An experiment on online product reviews. IEEE Intelligent Systems, 25(4), 46–53.

Deriu, J., Lucchi, A., De Luca, V., Severyn, A., Müller, S., Cieliebak, M., Hofmann, T., and Jaggi, M. (2017). Leveraging large amounts of weakly supervised data for multi-language sentiment classification. In Proceedings of the 26th International Conference on World Wide Web (WWW'17) (pp. 1045–1052). Perth, Australia: International World Wide Web Conferences Steering Committee.

dos Santos, C., and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers (COLING 2014) (pp. 69–78). Dublin, Ireland.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In Conference on Empirical Methods in Natural Language Processing (EMNLP). Copenhagen, Denmark.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In Proceedings of the 28th International Conference on Machine Learning (ICML-11) (pp. 513–520). Bellevue, Washington.

Gonçalves, P., Araújo, M., Benevenuto, F., and Cha, M. (2013). Comparing and combining sentiment analysis methods. In Proceedings of the First ACM Conference on Online Social Networks (COSN) (pp. 27–38). Boston: ACM.

Gonçalves, P., Dalip, D.H., Costa, H., Gonçalves, M.A., and Benevenuto, F. (2016). On the combination of "Off-The-Shelf" sentiment analysis methods. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC'16) (pp. 1158–1165). Pisa, Italy.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04) (pp. 168–177). Seattle, Washington.

Hutto, C. and Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of Eighth International AAAI Conference on Weblogs and Social Media (ICWSM'14). Ann Arbor, Michigan.

Levallois, C. (2013). Umigon: Sentiment analysis for tweets based on terms lists and heuristics. In Second Joint Conference on Lexical and Computational Semantics (* SEM), Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013) (Vol. 2, pp. 414–417). Atlanta, Georgia.

Mohammad, S., & Turney, P.D. (2013). Crowdsourcing a word-emotion association lexicon. Computational Intelligence, 29(3), 436–465.

Mohammad, S.M., Kiritchenko, S., and Zhu, X. (2013). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In Second Joint Conference on Lexical and Computational Semantics (* SEM), Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013) (Vol. 2, pp. 321–327). Atlanta, Georgia.

Moraes, F., Vasconcelos, M.A., Prado, P., Dalip, D.H., Almeida, J.M., and Gonalves, M.A. (2013). Polarity detection of foursquare tips. In Proceedings of International Conference on Social Informatics (SOCINFO) (pp. 153–162). Kyoto, Japan: Springer.

Mudinas, A., Zhang, D., and Levene, M. (2012). Combining lexicon and learning based approaches for concept-level sentiment analysis. In Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining WISDOM'12. Article No. 5 (p. 5). Beijing, China: ACM.

Mukherjee, S. and Bhattacharyya, P. (2012). WikiSent: Weakly supervised sentiment analysis through extractive summarization with Wikipedia. In Joint European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD'12) (pp. 774–793). Berlin, Heidelberg: Springer.

Nielsen, F.Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big Things Come in Small Packages (pp. 93–98). Heraklion, Greece.

Oliveira, N., Cortez, P., and Areal, N. (2013). On the predictability of stock market behavior using StockTwits sentiment and posting volume. In Proceedings of 16th Portuguese Conference on Artificial Intelligence (EPIA'13) (pp. 355–365). Angra do Heroísmo, Açores, Portugal: Springer.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (ACL'02) (Vol. 10, pp. 79–86). Stroudsburg, Pennsylvania: Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research (JMLR), 12, 2825–2830.

Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. Journal of Informetrics, 3(2), 143–157.

Ribeiro, F.N., Araújo, M., Gonçalves, P., Gonçalves, M.A., & Benevenuto, F. (2016). SentiBench - A benchmark comparison of state-of-the-practice sentiment analysis methods. EPJ Data Science, 5(1), 1–29.

Siddiqua, U.A., Ahsan, T., and Chy, A.N. (2016). Combining a rule-based classifier with weakly supervised learning for twitter sentiment analysis. In Innovations in Science, Engineering and Technology (ICISET), International Conference (pp. 1–4). Kumira, Chittagong, Bangladesh: IEEE.




DOI: 10.1002/asi

Smedt, T.D., & Daelemans, W. (2012). Pattern for Python. Journal of Machine Learning Research, 13(1), 2063–2067.
Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. Computational Linguistics, 37(2), 267–307.
Thelwall, M. (2013). Heart and soul: Sentiment strength detection in the social web with SentiStrength. In J. Holyst (Ed.), Cyberemotions: Collective Emotions in Cyberspace (CyberEmotions'13) (pp. 1–14). Berlin: Springer.
Wang, L., and Xia, R. (2017). Sentiment lexicon construction with representation learning based on hierarchical sentiment supervision. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 502–510). Copenhagen, Denmark.
Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. (2005). OpinionFinder: A system for subjectivity analysis. In Proceedings of HLT/EMNLP 2005 Interactive Demonstrations (pp. 34–35). British Columbia, Canada: Association for Computational Linguistics.
Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., and Liu, B. (2011). *Combining lexicon-based and learning-based methods for Twitter sentiment analysis.* HP Laboratories, Technical Report HPL-2011, 89. Retrieved from http://www.hpl.hp.com/techreports/2011/HPL-2011-89.html
Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1253.


DOI: 10.1002/asi