Artificial Intelligence Review (2021) 54:1887–1936
https://doi.org/10.1007/s10462-020-09895-6

On the evaluation and combination of state-of-the-art features in Twitter sentiment analysis

Jonnathan Carvalho[1] · **Alexandre Plastino**[2]

Published online: 27 August 2020

Abstract
Sentiment analysis of short informal texts, such as tweets, remains a challenging task due to their particular characteristics. Much effort has been made in the literature of Twitter sentiment analysis to achieve an effective and efficient representation of tweets. In this context, distinct types of features have been proposed and employed, from the simple *n*-gram representation to meta-features to word embeddings. Hence, in this work, using a relevant set of twenty-two datasets of tweets, we present a thorough evaluation of features by means of different supervised learning algorithms. We evaluate not only a rich set of meta-features examined in state-of-the-art studies, but also a significant collection of pre-trained word embedding models. Also, we evaluate and analyze the effect of combining those distinct types of features in order to detect which combination may provide core information in the polarity detection task in Twitter sentiment analysis. For this purpose, we exploit different strategies for combination, such as feature concatenation and ensemble learning techniques, and show that the sentiment detection of tweets benefits from combining different types of features proposed in the literature.

Keywords Sentiment analysis · Meta-features · Word embeddings · Ensemble learning · Twitter

1 Introduction

In recent years, much attention has been given to the content generated by Internet users. Since people can express their opinions and emotions about any target, such as products, services, and events around the globe, many consumers and companies can make decisions

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10462-020-09895-6) contains supplementary material, which is available to authorized users.

✉ Jonnathan Carvalho
joncarv@iff.edu.br

Alexandre Plastino
plastino@ic.uff.br

[1] Instituto Federal Fluminense (Campus Itaperuna), Itaperuna, Brazil

[2] Universidade Federal Fluminense, Niterói, Brazil

based on this ever-growing opinionated content. However, as a huge amount of opinions is published every day, manually seeking for and identifying them as conveying a positive or negative sentiment may be impractical. In this context, Sentiment Analysis, or Opinion Mining, is the field of study that analyzes people's opinions, sentiments, appraisals, attitudes, and emotions toward entities and their attributes expressed in written text (Liu 2015).

One of the key challenges in this field is regarding the automatic identification of opinions and emotions expressed in short informal texts, such as tweets. Tweets, which are short texts published on Twitter,[1] make the task of sentiment analysis very complex due to their inherent characteristics, such as their informal linguistic style, the presence of misspelled words, and the careless use of grammar (Martínez-Cámara et al. 2014). Although sentiment analysis has recently been recognized as a suitcase research problem (Cambria et al. 2017; Chaturvedi et al. 2018), which involves various Natural Language Processing (NLP) tasks, including sarcasm detection, aspect extraction, and subjectivity detection, our focus is on the polarity detection task. Regarding supervised machine learning strategies, which are also the focus of this work, much effort has been made in the literature of Twitter sentiment analysis to achieve an effective representation of tweets. In this context, distinct types of features have already been proposed, from the simple *n*-gram-based representation to meta-level features to word embeddings.

N-grams are the most basic feature representation when dealing with text classification problems, having motivated early works on Twitter sentiment analysis (Go et al. 2009; Pak and Paroubek 2010; Pang et al. 2002). In this scenario, raw sequences of *n* words extracted from tweets constitute a sparse and high-dimensional feature space for the classification task. Later, in an attempt to deviate from the sparsity issue, several state-of-the-art studies have proposed different sets of features by developing an abstract representation of tweets, comprising meta-information extracted from their textual content (Barbosa and Feng 2010). Those features, also called meta-level features, can capture new, insightful information from tweets, taking into account their peculiarities. More recently, distributed representations of words generated from deep learning approaches, namely word embeddings, have emerged as an efficient feature representation for text documents. They are currently the main focus of most works on sentiment detection in tweets. Word embeddings encode linguistic patterns of words from a vast corpus of textual data and can represent the textual content of tweets in low-dimensional feature vectors.

As far as we know, despite the efforts on designing effective and efficient feature representation in the literature of Twitter sentiment analysis, there is a gap regarding the effect of combining such distinct types of features proposed in state-of-the-art works. In this study, we recognize three main groups of features considering their structural properties and how they are engineered, such as the n-gram language model, meta-level features, and word embedding-based features. Each of these groups encloses a rich disjoint set of features which may boost classification effectiveness if appropriately combined.

Moreover, as for meta-level features, we have observed that only a small and different fraction of features are employed on each work in the literature. For that, we propose to fill another gap by aggregating meta-level features designed in different works. We believe that combining them into a unique set might benefit sentiment detection in tweets, as we shall see later. Also, we categorize this aggregated set of meta-level features, putting together

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

features that share similar aspects, so as to examine whether the sentiment classification of tweets can benefit from different categories of meta-level features.

In this work, our main goal is to improve classification performance in Twitter sentiment analysis. In this context, this study is conducted in order to provide a response to the following three main research questions:

RQ1. Which group of features is the most effective in Twitter sentiment analysis? Given the large number of features of distinct types designed and employed in the literature, such as n-grams, meta-level features, and word embedding-based features, we propose to perform a comparative evaluation of their predictive performances, by means of a large collection of datasets of tweets. Our goal is to detect the most powerful feature set in the sentiment classification of tweets from various domains.

However, we believe that an improper choice of a learning algorithm to be used with a specific feature set may hinder classification performance. As a result, it might prevent the classifier from learning how to assign a sentiment label to tweets accurately. With that said, in order to take maximum advantage of the features from each feature set, we leverage the best classifiers constructed for each feature set, instead of comparing them by merely relying on the same learning algorithm. More clearly, we answer the intermediate question "*Which classification strategies are the most suitable for each group of features?*" by evaluating distinct supervised learning algorithms for each feature set. After identifying the best classifiers under the individual evaluation of each feature set, we then carry out a fair comparative assessment of their predictive potential.

As a result of the comparative study among the best classifiers for each feature set, as we shall see, a classifier made up of a concise —yet rich—set of meta-level features from well-referenced works (Agarwal et al. 2011; Barbosa and Feng 2010; Bravo-Marquez et al. 2014; Buscaldi and Hernandez-Farias 2015; da Silva et al. 2014; Davidov et al. 2010; Go et al. 2009; Hagen et al. 2015; Jiang et al. 2011; Khuc et al. 2012; Kouloumpis et al. 2011; Mohammad et al. 2013; Park et al. 2018; Vo and Zhang 2016; Zhang et al. 2011) achieves improved results, which may be a piece of evidence that such feature set plays an essential role in this task. Going further, we propose to categorize this rich set of meta-level features, this being an extension of our previous study (Carvalho and Plastino 2016). In this work, the categories proposed in Carvalho and Plastino (2016) are revisited, and we include some new meta-level features. In addition to this categorization, we investigate whether the classification of tweets from different domains can benefit from these distinct categories of meta-level features. For this purpose, we evaluate the predictive power of those categories in order to give a more general understanding of the relevance of the most common meta-level features proposed in the literature.

Lastly, regarding the word embedding-based features, we also present an underlying evaluation of a significant collection of generic and affective pre-trained embedding models that we have identified in the literature, in order to acknowledge the most effective one for the polarity classification of tweets. Pre-trained models are publicly available embedded representations of words, trained with different deep learning methods. While generic pre-trained models comprise word vectors trained for general purpose, the affective ones are specifically trained for the sentiment and emotion detection tasks.

RQ2. Can the concatenation of different types of features proposed in the literature boost classification performance in Twitter sentiment analysis? We propose to evaluate distinct combinations of the feature sets investigated in this work, (i.e., *n*-grams, meta-level features, and word embedding-based features), considering that features from different groups might complement one another, leading to an improvement in detecting the polarity of tweets. Our goal is to determine which combinations of distinct feature sets may provide

the core information in Twitter sentiment analysis. To this end, we adopt a simple feature concatenation approach that aims at combining features from distinct groups into a unique feature vector. In this work, we investigate whether the concatenation of all feature sets, as well as pairs of distinct feature sets, can improve sentiment classification effectiveness.

Furthermore, despite the acknowledged use of SVM due to its robustness on large feature spaces (Carvalho and Plastino 2016; Hagen et al. 2015; Jabreel and Moreno 2017; Mohammad et al. 2013), to the best of our knowledge, no study in the literature evaluates the effectiveness of different learning methods in the presence of the different types of features studied in this work. We believe that some learning algorithms may be more effective than others when features from distinct natures are put together, depending on their intrinsic properties and how the learning algorithms can harness them. In this scenario, we also conduct experiments to identify which classification strategies are the most suitable when combining features of different types.

RQ3. Can the sentiment classification of tweets benefit from the use of ensemble classification strategies having the best classifiers for each type of feature as base learners? Another approach to combine the discriminative power of different sets of features is through ensemble classification methods. Ensemble methods are learning algorithms that create a set of classifiers, also called base classifiers or base learners, which are used to classify new instances by taking a vote of their predictions (Dietterich 2000).

According to Zhang and Duin (2011), in practice, there exist two main kinds of ensemble strategies. In the first, the predictions of homogeneous classifiers are combined according to some rule. The second is marked by the use of heterogeneous classifiers. While homogeneous classifiers use the same learning algorithm with different representations of the feature space, the heterogeneous ones apply different classification algorithms to the same input features. In this work, we exploit a hybrid approach to ensemble learning.

Specifically, given the varied nature of features studied in this work, we use different learning algorithms as base classifiers, each one provided with a specific feature representation for the same dataset of tweets (i.e., n-grams, meta-level features, or embedding-based features). For most situations, we show that those classifiers can complement one another in the sentiment detection of tweets, properly dealing with the peculiarities of the data that might be uncovered by some of them. In addition, we provide an in-depth analysis of the correlation among the base classifiers, showing that there is sufficient diversity among them, which is an imperative condition for ensemble strategies to succeed (Dietterich 2000)

In summation, the main contributions of this study are: (i) a literature review and analysis of the most common feature representations of tweets for supervised sentiment classification, including n-grams, meta-level features, and word embedding-based features; (ii) the categorization of a rich set of meta-level features developed in state-of-the-art works and the evaluation of each proposed category; (iii) a comparative study of a significant collection of publicly available pre-trained word embedding models in the sentiment classification of tweets; (iv) an assessment of the combination effectiveness of the different sets of features studied in this work, by feature concatenation and ensemble learning; and (v) the use of twenty-two datasets of tweets in all experiments performed in this work. To the best of our knowledge, this is the first study that evaluates different types of features and classifiers for a significant number of tweet datasets.

This article is organized as follows. In Sect. 2, we present the related work, offering a description of the distinct types of feature representation, as well as how they have been combined in the literature to increase the predictive performance of Twitter sentiment analysis. Sections 3, 4, and 5 present a literature review of the features exploited in this work,

such as *n*-grams, meta-level features, and word embedding-based features, respectively. The computational experiments conducted in this work to answer the research questions introduced in this section are described in Sect. 6. Finally, in Sect. 7, we present the conclusions of this work and directions for future research.

2 Related work

Sentiment analysis. Over the years, sentiment analysis has been broadly used to summarize people's opinions and sentiments about products, services, organizations, individuals, and events (Liu 2012). In the pioneer works in sentiment analysis, Pang et al. (2002) and Turney (2002) applied distinct machine learning methods in the domain of product reviews.

In Pang et al. (2002), Pang et al. applied three supervised machine learning algorithms to determine the polarity of movie reviews. Conversely, using an unsupervised approach, Turney (2002) presented a simple strategy for classifying reviews of automobiles, banks, movies, and travel destinations as recommended or not recommended, i.e., whether the reviews convey a positive or a negative opinion. Since then, sentiment analysis has been applied in various domains to solve distinct types of problems (Cambria et al. 2010; Tumasjan et al. 2010; Valdivia et al. 2017; Wang et al. 2012a; Yoo et al. 2018).

In past years, sentiment analysis has been used to generate real time insights during political debates (Tumasjan et al. 2010; Wang et al. 2012a), detect real-time events (Yoo et al. 2018), and in health (Cambria et al. 2010) and tourism applications (Valdivia et al. 2017). Also, as social media interactions grow, companies can collect customers feedback and influence their decisions by designing intelligent marketing systems, as well as using public mood to predict the stock market (Bollen et al. 2011). In this scenario, applications of sentiment analysis on social media marketing and financial forecasting have received attention from the research community in recent years (Li et al. 2020; Xing et al. 2018, 2019).

Xing et al. (2018) addressed the problem of incorporating public mood to the asset allocation problem, which is an investment strategy that aims at balancing the trade-off between asset returns and the risk taken by investors. In Xing et al. (2018), they developed an ensemble of an evolving clustering method and long short-term memory (LSTM) neural network to formalize sentiment information in market views. To this end, they proposed to compute sentiment time series from social media by using the sentic computing framework Cambria and Hussain (2015), arguing that it enables sentiment analysis not only at document or sentence level but also at concept level.

Recently, Li et al. (2020) studied how to combine technical indicators from stock prices and news sentiments from textual news articles, which is considered as an open research topic in financial market. To this end, they used different sentiment dictionaries to model news sentiment and constructed a two-layer LSTM network to make stock predictions. They showed that the LSTM incorporating both technical indicators and news sentiments outperformed the baseline models that use only one of these information sources at a time.

Although much effort in the literature of sentiment analysis has been on exploiting only English content, Lo et al. (2017) claim that it is no longer sufficient, considering that Asia now has the most Internet users (52.2%), followed by Europe (15.1%).[2] Thus, dealing with

[2] https://www.internetworldstats.com/stats.htm.

multilingual language content represents one of the major challenges in sentiment analysis (Araújo et al. 2020; Dashtipour et al. 2016; Lo et al. 2017). For example, Araújo et al. (2020) investigated how a simple translation strategy can address the problem of sentiment analysis in multiple languages. In Araújo et al. (2020), they showed that machine translation systems such as Google Translate, Microsoft Translator Text API, and Yandex Translate, are mature enough to produce reliable translations to English that can be used for sentence-level sentiment analysis.

At present, with the explosion of social media networks, semi-supervised strategies have also been emerging in the literature of sentiment analysis taking advantage of the massive amount of unlabeled data available (Fu et al. 2019; Hussain and Cambria 2018). Hussain and Cambria (2018) describe semi-supervised learning as a supervised learning problem biased by an unsupervised reference solution. In Hussain and Cambria (2018), they proposed a novel semi-supervised learning model for the task of emotion recognition based on the combined use of random projections and support vector machines. Fu et al. (2019) built a novel model to perform aspect-level sentiment classification, called AL-SSVAE (Semi-supervised Aspect Level Sentiment Classification Model based on Variational Autoencoder), based on the variational autoencoder framework (Kingma and Welling 2013). The proposed model introduces a given aspect of text into the encoder and decoder, and adds an aspect level sentiment classifier for semi-supervised learning in the aspect level sentiment classification.

Feature representation. One of the most significant challenges when dealing with text classification problems is related to feature engineering, especially in short texts such as tweets. Among the broad set of features that have emerged in the literature of Twitter sentiment analysis, the *n*-gram features have been widely employed because of their simplicity in representing tweets (Agarwal et al. 2011; Araque et al. 2017; Arif et al. 2018; Barbosa and Feng 2010; Bermingham and Smeaton 2010; Bifet and Frank 2010; Chikersal et al. 2015; Cozza and Petrocchi 2016; da Silva et al. 2016, 2014; Davidov et al. 2010; Emadi and Rahgozar 2019; Go et al. 2009; Hagen et al. 2015; Hamdan 2016; Hamdan et al. 2015; Jabreel and Moreno 2017; Jiang et al. 2011; Kouloumpis et al. 2011; Lin and Kolcz 2012; Lochter et al. 2016; Miranda-Jiménez et al. 2017; Mohammad et al. 2013; Narr et al. 2012; Pak and Paroubek 2010; Saif et al. 2012; Siddiqua et al. 2016; Speriosu et al. 2011; Wang et al. 2012b; Zhang et al. 2011).

N-gram features are contiguous sequences of n words from a text. Despite their simplicity, it has already been acknowledged that this type of feature may negatively impact the predictive performance of the classification because of the large number of uncommon words in Twitter (Saif 2015), and because people tend to use much less characters of the 140-character limit for tweets (da Silva et al. 2016). Indeed, analyzing a corpus of 1.6M tweets, Go et al. (2009) have reported that the average length of a tweet is 14 words, or 78 characters. Further, in Saif et al. (2012), it was brought to attention that 93% of the words in a corpus of 60,000 tweets are highly infrequent, occurring less than ten times. These drawbacks make the data very sparse due to the curse of dimensionality, which can sometimes prevent the classifier from correctly learning how to assign a sentiment label to unseen tweets.

Beyond the sparsity issue, another factor that makes the sentiment classification even harder is related to the challenging nature of tweets, such as their informal linguistic style and the careless use of grammar (Martínez-Cámara et al. 2014), resulting in a new form of written text, termed microtext (Cambria et al. 2017). In this context, while some studies propose methods for normalizing tweets to plain English hence improving classification accuracy (Satapathy et al. 2017), other state-of-the-art works have explored feature

engineering by designing hand-crafted features or meta-level features. Meta-level features are usually extracted from other features and are able to capture insightful new information about the data (Canuto et al. 2016). These features include summations and counts of: part-of-speech of words (Agarwal et al. 2011; Barbosa and Feng 2010; Bravo-Marquez et al. 2014; Go et al. 2009; Kouloumpis et al. 2011; Mohammad et al. 2013), punctuation marks (Agarwal et al. 2011; Barbosa and Feng 2010; Davidov et al. 2010; Hagen et al. 2015; Jiang et al. 2011; Mohammad et al. 2013), specific characteristics of Twitter and short messages, such as hashtags, user mentions, retweets (RT), abbreviations, etc. (Agarwal et al. 2011; Barbosa and Feng 2010; Hagen et al. 2015; Jiang et al. 2011; Kouloumpis et al. 2011; Mohammad et al. 2013; Zhang et al. 2011), emoticons (Agarwal et al. 2011; da Silva et al. 2014; Hagen et al. 2015; Mohammad et al. 2013), and lexicon features (Agarwal et al. 2011; Bravo-Marquez et al. 2014; da Silva et al. 2014; Hagen et al. 2015; Jiang et al. 2011; Khuc et al. 2012; Kouloumpis et al. 2011; Mohammad et al. 2013; Vo and Zhang 2016), which use the prior sentiment information of words annotated in existing lexicon resources. For example, Mohammad et al. (2013) have implemented a large set of meta-features (referred to as NRC-features), while also emphasizing the importance of a set of lexicon-based features. In Mohammad et al. (2013), authors have designed lexicon-based features such as the total number of positive and negative tokens from a tweet, the overall and the maximal score of a tweet, and the score of the last token of a tweet. All those features were extracted for each of the five different sentiment lexicons. The results of the experiments have shown that the most influential features for the two assessed datasets of tweets were the lexicon-based ones, which led to an improvement of 8.5% in terms of the macro-averaged F-score of the positive, negative, and neutral classes.

With the revival and success of deep learning techniques in traditional machine learning applications, distributed representations of words have emerged as a solution to the curse of dimensionality issue (Bengio et al. 2003; Collobert et al. 2011; Mikolov et al. 2013a, b; Pennington et al. 2014). In this context, neural networks based on dense vector representations have been producing superior results in many NLP tasks (Young et al. 2018). Bengio et al. (2003) have discussed two main characteristics of the n-gram model that can lead to misclassification problems: the context and the similarity between words are not taken into consideration. Although some context can be caught by using higher-order n-grams, such as 5-grams, it does not consider contexts farther than n words. Besides that, it makes the dimensionality even higher. Collobert et al. (2011) introduce a method to overcome these limitations, which relies on largely unlabeled data and uses a multilayer neural network architecture to learn word representations, namely word embeddings. Word embeddings are dense, low-dimensional, and real-valued vectors, each one representing a word in the vocabulary, and encode linguistic patterns that can capture context from a massive corpus of textual data. This method has been successfully applied in many NLP tasks such as part-of-speech tagging, named entity recognition and semantic role labeling (Collobert et al. 2011).

In the context of sentiment analysis, some works have effectively designed sentiment and emotion-specific embedding learning methods (Agarwal et al. 2018; Felbo et al. 2017; Tang et al. 2014; Xu et al. 2018). For example, Tang et al. (2014) have observed that traditional methods for learning word embeddings ignore the sentiment information of text, which may become a problem since words that appear in similar contexts but carrying opposite polarities are mapped into close vectors (for example, *good* and *bad*). In Tang et al. (2014), this issue is addressed by extending the method proposed in Collobert et al. (2011). Specifically, Tang et al. have developed a sentiment-specific word embedding (SSWE) neural network that incorporates the sentiment information of texts into the

embedding learning process, using a corpus of 10M tweets with emoticons as a noisy, distant-supervised training data. In the experiments conducted to evaluate their approach, Tang et al. have shown that the results achieved by the SSWE learning method are competitive with those achieved by the state-of-the-art meta-level features proposed in Mohammad et al. (2013) (84.98% and 84.73% in macro-F1, respectively).

Recently, deep learning methods have also been successfully applied to aspect-based sentiment analysis (Chen et al. 2017; Ma et al. 2018; Wang et al. 2017), which aims at identifying the polarity of specific aspects rather than the document itself in its entirety (Poria et al. 2016). For example, Ma et al. (2018) have proposed a long short-term memory (LSTM) neural architecture that incorporates the attention mechanism. LSTM is a recurrent neural network (RNN) that can handle sequences of data. The attention mechanism takes an external memory and representations of a sequence as input and produces a probability distribution related to each position of the sequence. In Ma et al. (2018), authors have modeled attention as a two-step model: target-level attention and sentence-level attention, and they have shown that the proposed attention architecture can outperform state-of-the-art methods in aspect-based sentiment analysis.

Combination strategies. Arguing that the combination of classifiers has not been properly explored in the literature of Twitter sentiment analysis, da Silva et al. (2014) show that a classifier ensemble formed by Multinomial Naive Bayes (MNB), Support Vector Machines (SVM), Random Forest (RF), and Logistic Regression (LR) can improve the classification accuracy on four sentiment datasets used in the investigation, when combined in a majority voting strategy. The diversity in the classifier ensemble is addressed by varying only the base learners, all of which using the same bag-of-words feature representation. Prusa et al. (2015) have evaluated seven base classifiers combined with either bagging or boosting ensemble strategies on the sentiment classification of tweets, using only unigrams as features. In bagging, different training partitions are sampled from the original training dataset (with replacement), and a single base learner is trained on each partition. Boosting, on the other hand, iteratively creates the base classifiers, where in each iteration a classifier is trained based on the misclassified instances from previous iterations. At the end of the process, both ensemble techniques aggregate the resulting classifiers by averaging the posterior probabilities of each model in the ensemble. In Prusa et al. (2015), they show that using ensemble strategies such as bagging and boosting can benefit the sentiment classification of tweets, particularly on high dimensional datasets.

In Fersini et al. (2014), Fersini et al. propose a Bayesian Ensemble Learning approach based on Bayesian Model Averaging (BMA), which uses a greedy backward elimination strategy to select the optimal set of base classifiers. The base candidate classifiers that integrate the search space are a dictionary-based approach (DIC), NB, SVM, Maximum Entropy (ME), and Conditional Random Fields (CRF). The feature space used for learning is the bag-of-words model, except for the DIC approach, which relies on the polarities of words in sentiment lexicons. Interestingly, although the dictionary approach presents the lower individual performance on the datasets used in the experimental evaluation, the optimal ensemble provided by BMA always includes DIC as one of the base classifiers for all datasets.

Recently, Fersini et al. (2016) pointed out that not only words are key features in detecting the sentiment polarity of tweets, but also some strong signals can help to discriminate the positive messages from the negative ones. In this context, in Fersini et al. (2016), the combination of the bag-of-words representation of tweets with adjectives, pragmatic particles (emoticons, initialisms for emphatic expressions, and onomatopoeic expressions), and expressive lengthening are investigated independently and as part of an ensemble learning

strategy. More precisely, the bag-of-words vectors representing each tweet are expanded with five new features: the number of positive and negative adjectives, the number of positive and negative pragmatic particles and the expressive lengthening of a tweet. In the experimental investigation, they show that using the bag-of-words model expanded with all those expressive signals on an ensemble learning framework (BMA Fersini et al. 2014) can lead to a significant improvement in terms of accuracy.

The combination of distinct preprocessing techniques with well-established classification algorithms has been investigated by Lochter et al. (2016). In Lochter et al. (2016), they propose an ensemble system that performs a grid search to select the best combination between text processing techniques and different classification methods, such as Naive Bayes (NB), SVM, LR, *k*-Nearest Neighbors (*k*-NN), and Decision Trees (DT). In Lochter et al. (2016), they evaluate the predictive power of the ensemble system on nine datasets of tweets. As their goal is to detect the best combination of text preprocessing techniques and classifiers, they have used a small fixed set of features for each learning method assessed, such as unigrams and the count of positive and negative terms in each tweet. Emadi and Rahgozar (2019) have recently proposed a classifier ensemble approach which combines supervised and unsupervised methods in Twitter sentiment classification. To this end, three supervised machine learning algorithms, such as SVM, NB, and ME are used as base classifiers, each of them supplied with unigrams, bigrams, and a combination of both. In addition to those classifiers, an unsupervised NLP-based method is used. The classifiers are chosen based on diversity measures in order to select methods that complement one another. Once the diverse set of classifiers is identified, i.e., classifiers with sufficient diversity, a learning fusion method is applied to assign a polarity orientation for each tweet. In Emadi and Rahgozar (2019), the Choquet Fuzzy Integral (CFI) method is used as a meta-learning strategy, which combines the decision of each classifier. Araque et al. (2017) have investigated different combinations of features via ensemble learning and through feature concatenation. They evaluate and compare the predictive performance of these combinations against a supervised baseline model fed with word embeddings trained on a corpus of 1.28M tweets. For the ensemble model, they use as base classifiers six different sentiment methods, each one trained with various, though rather simple, features (e.g., *n*-grams, POS features, and polarity values for each word), in addition to classifiers trained with generic and affective word embeddings, i.e., word vectors trained for general purpose and for the sentiment analysis task, respectively.

Different from ensemble learning methods, which combine the strength of classifiers and features at prediction time, feature concatenation consists in combining different sets of features into a unified set as a preprocessing step prior to the classification process. Aiming at evaluating the combination of several types of features, Araque et al. (2017) have proposed three feature concatenation models. The first one, denoted M_{SG}, combines a small set of meta-level features and generic word embedding vectors. The second type, M_{GA}, combines generic and affective word vectors. Finally, the third, M_{SGA}, consists in the combination of the features included in the first and second models, i.e., meta-level features, generic and affective word vectors. In the experimental evaluation, both the ensemble model and the feature concatenation model M_{SG} achieved the best results, with no significant statistical difference between them. Agarwal et al. (2011) have proposed a rich set of meta-level features, termed Senti-features, which were divided into three categories: $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{B}$. Features from category $\mathbb{N}$ are those whose value is a positive integer (e.g., #hashtags, #positive words, etc.). Features from category $\mathbb{R}$ are those whose value is a real number (e.g., polarity score of words in some lexicon). Lastly, features whose value is a boolean (e.g., presence of capitalized text) make up category $\mathbb{B}$. Besides, they have adopted unigrams

as a baseline. In the experimental evaluation of the proposed set of features, features were added incrementally to the baseline unigram model and they show that the best result is achieved by using all meta-level features in combination with the unigrams, through feature concatenation. Mansour et al. (2015) examine a large set of features introduced in the literature of Twitter sentiment analysis. In Mansour et al. (2015), authors have performed an exhaustive combination of features aiming at identifying a compact feature subset that can reduce the computational complexity without harming classification accuracy. To this end, in addition to unigrams and bigrams, they have also investigated Senti-features (Agarwal et al. 2011), NRC-features (Mohammad et al. 2013), and SSWE embeddings (Tang et al. 2014). In their experimental evaluation, they employed two datasets of tweets and the Maximum Entropy classification algorithm. For polarity detection, they identified that the best results were achieved by concatenating unigrams, bigrams, NRC-features, and SSWE embeddings with macro-F1 of 89%.

In Tang et al. (2014), Tang et al. explore the combination of the SSWE embeddings and the state-of-the-art NRC-features (Mohammad et al. 2013) through feature concatenation, which improved prediction performance from 84.98% to 86.58%. In order to obtain rich sources of information, Vo and Zhang (2015) employed, as features, a combination of word vectors trained with two different embedding learning approaches, namely Google's word2vec (Mikolov et al. 2013b) and SSWE (Tang et al. 2014). To this end, they have trained the embeddings with a large-scale corpus of 5M unlabeled tweets and show that the combination of generic and affective word vectors are beneficial to the sentiment classification of tweets. Xu et al. (2018) investigate the performance of the proposed affective embedding learning system, Emo2vec, by combining the word vectors obtained with both their approach and Stanford's GloVe vectors (Pennington et al. 2014), in an attempt to render feature representation more accurate, since Emo2vec is weak on capturing syntactic and semantic meaning. Table 1 presents a summary of the combination methods discussed in this section.

Discussion. Diversity is a key point in designing ensemble approaches (Brown et al. 2005). Despite the application of combination methods in Twitter sentiment classification, most works use the same feature representation varying only the classification algorithms, as shown in Table 1, with the exception of Araque et al. (2017), whose base learners are state-of-the-art classifiers from the literature. We believe that different classification strategies may benefit from the use of an appropriate set of features. For example, an SVM classifier fed with n-grams may be successful in this task, but this might not stand true if we employ the same feature representation with another inducer, such as Random Forest. We investigate this hypothesis by evaluating the predictive power of features of varied types, feeding them to different state-of-the-art learning algorithms. Furthermore, we combine the best classification strategies for each group of features through feature concatenation, as well as using them as base classifiers for ensemble strategies. The features used in this investigation are n-grams, meta-level features, and word embeddings, which have been widely adopted in the literature as of late.

Though some other relevant studies present the assessment of different Twitter sentiment analysis methods (Maynard and Bontcheva 2016; Zimbra et al. 2018), their primary focus is on the evaluation of existing systems, treating them as black boxes. Our assessment, on the other hand, is more fine-grained in the sense that we analyze the performance of distinct classification strategies at feature-level, i.e., we evaluate the effect of different kinds of features on the polarity detection task. Also, even though some other works sought to examine the combination of different kinds of features, such as the one presented in Mansour et al. (2015), we fulfill a more robust evaluation.

Table 1 Summary of combination strategies on Twitter sentiment classification, separated by classifier ensemble and feature concatenation approaches

References	Strategy	Feature representation
Classifier ensemble approaches		
Araque et al. (2017)	Majority voting and stacking	
	Sentiment140 (Go et al. 2009)	Unigrams + bigrams + POS
	Stanford CoreNLP (Manning et al. 2014)	Word embeddings
	Sentiment WSD (Kathuria 2019)	Polarity of words in SentiWordNet
	Vivekn (Narayanan et al. 2013)	*n*-grams
	pattern.en (De Smedt and Daelemans 2012)	POS + polarity and subjectivity scores of words + WordNet vocabulary information
	TextBlob (Loria 2016)	Unigrams
	M_G (LR)	Generic word embeddings
	M_{GA} (LR)	Generic + affective embeddings
	M_{SG} (LR)	Generic embeddings + #(+/−) words + #neutral words + #exclamation marks + #question marks + #words in all caps + #elongated words + #hashtags
da Silva et al. (2014)	Majority voting (MNB, SVM, RF, LR)	Bag-of-words
Emadi and Rahgozar (2019)	Choquet fuzzy integral (SVM, NB, ME, NLP-based approach)	Unigrams + bigrams
Fersini et al. (2014)	BMA (dictionary-based approach, NB, SVM, ME, CRF)	Bag-of-words
Fersini et al. (2016)	Majority voting and BMA (MNB, SVM, DT, Bayesian networks)	Bag-of-words + #(+/−) adjectives + #(+/−) pragmatic particles + expressive lengthening
Lochter et al. (2016)	Weighted majority voting (NB, SVM, LR, *k*-NN, DT)	Unigrams + #(+/−) terms
Prusa et al. (2015)	Bagging and boosting (*k*-NN, DT, SVM, LR multilayer perceptron, RBF)	Bag-of-words
Feature concatenation approaches		
Agarwal et al. (2011)	Support vector machines	Unigrams + senti-features (Agarwal et al. 2011)
Araque et al. (2017)	M_{GA} (LR)	Features from M_{GA}
	M_{SG} (LR)	Features from M_{SG}
	M_{SGA} (LR)	Features from M_{GA} and M_{SG}

Table 1 (continued)

References	Strategy	Feature representation
Mansour et al. (2015)	Maximum entropy	Unigrams + bigrams + SSWE embeddings (Tang et al. 2014) + senti-features (Agarwal et al. 2011) + NRC-features (Mohammad et al. 2013)
Tang et al. (2014)	Logistic regression	Affective embeddings + NRC-features (Mohammad et al. 2013)
Vo and Zhang (2015)	Logistic regression	Generic + affective embeddings
Xu et al. (2018)	Logistic regression	Generic + affective embeddings

Specifically, in Mansour et al. (2015), Mansour et al. have only employed a subset of the features we have examined in our experiments. For instance, regarding the representation provided by pre-trained embeddings, Mansour et al. (2015) have only investigated the SSWE embeddings (Tang et al. 2014), whereas we present an extensive analysis of ten different pre-trained embeddings from the literature. Lastly, in Mansour et al. (2015), while authors have only adopted two datasets of tweets and one classification algorithm, we present a more robust assessment on 22 datasets and making use of three distinct classification algorithms.

Another point we have observed is that works in sentiment classification of tweets do not fully exploit the predictive power of meta-level features, especially those in which combination strategies are proposed. Over the years, many researchers have experimented with meta-level features in sentiment classification, only a small representation of which having been adopted by each work. Hence, in this work, we show that aggregating meta-level features from different works into a unique feature set can achieve higher accuracies when compared to *n*-grams and word embeddings. Also, we show that combining meta-level features with either *n*-grams or word embedding vectors can significantly improve the predictive performance of supervised Twitter sentiment analysis. In the following sections, we present an overview of each of these feature representations.

3 N-gram features

Different types of features have been engineered and used in Twitter sentiment analysis, from the most common representation, such as *n*-grams, to meta-level features and word embeddings. N-grams are contiguous sequences of n tokens from a text. The most common representation of textual data is the bag-of-words or unigram model ($n = 1$), in which each word of a tweet is considered as a feature. In general, the feature space is represented by a binary feature vector indicating whether each word of the vocabulary occurs in the tweet or not. In that case, the values 0 and 1 represent the absence and presence of each word in the tweet, respectively (Pak and Paroubek 2010).

In the task of sentiment analysis, Pang et al. (2002) are the pioneer authors using *n*-grams as features to detect the polarity of movie reviews. In the sentiment classification of tweets, Go et al. (2009) have used the same approach as in Pang et al. (2002) to classify the sentiment expressed in tweets using a distant supervision method. This method relies on positive and negative emoticons as noisy labels in a training dataset of 1.6M tweets. Since then, *n*-grams have been one of the most adopted features in supervised learning strategies due to their simplicity in representing tweets (Agarwal et al. 2011; Araque et al. 2017; Arif et al. 2018; Barbosa and Feng 2010; Bermingham and Smeaton 2010; Bifet and Frank 2010; Chikersal et al. 2015; Cozza and Petrocchi 2016; da Silva et al. 2016, 2014; Davidov et al. 2010; Emadi and Rahgozar 2019; Go et al. 2009; Hagen et al. 2015; Hamdan 2016; Hamdan et al. 2015; Jabreel and Moreno 2017; Jiang et al. 2011; Kouloumpis et al. 2011; Lin and Kolcz 2012; Lochter et al. 2016; Miranda-Jiménez et al. 2017; Mohammad et al. 2013; Narr et al. 2012; Pak and Paroubek 2010; Saif et al. 2012; Siddiqua et al. 2016; Speriosu et al. 2011; Wang et al. 2012b; Zhang et al. 2011).

Table 2 presents an overview of the *n*-gram features in the literature of Twitter sentiment analysis. As shown in Table 2, most studies in the literature discourage the use of higher-order *n*-grams, such as 4- and 5-grams, trying to minimize the sparsity problem.

Table 2 Overview of the n-grams features used in the literature of Twitter sentiment classification ordered by publication year

Year	References	$n = 1$ bag-of-words	$n = 2$	$n = 3$	$n = 4$	$n = 5$
2009	Go et al. (2009)	✓	✓			
2010	Barbosa and Feng (2010)	✓				
	Bermingham and Smeaton (2010)	✓	✓	✓		
	Bifet and Frank (2010)	✓				
	Davidov et al. (2010)	✓	✓	✓	✓	✓
	Pak and Paroubek (2010)	✓	✓	✓		
2011	Agarwal et al. (2011)	✓				
	Jiang et al. (2011)	✓	✓			
	Kouloumpis et al. (2011)	✓	✓			
	Speriosu et al. (2011)	✓	✓			
2012	Lin and Kolcz (2012)				✓	
	Narr et al. (2012)	✓	✓			
	Saif et al. (2012)	✓				
	Wang et al. (2012b)	✓				
2013	Mohammad et al. (2013)	✓	✓	✓	✓	
2014	da Silva et al. (2014)	✓				
2015	Chikersal et al. (2015)	✓	✓	✓		
	Hagen et al. (2015)	✓	✓	✓	✓	
	Hamdan et al. (2015)	✓	✓			
	Mansour et al. (2015)	✓	✓			
	Zhang et al. (2011)	✓				
2016	Cozza and Petrocchi (2016)		✓			
	da Silva et al. (2016)	✓	✓	✓		
	Hamdan (2016)	✓	✓	✓		
	Lochter et al. (2016)	✓				
	Siddiqua et al. (2016)	✓				
2017	Araque et al. (2017)	✓	✓			
	Jabreel and Moreno (2017)	✓	✓	✓	✓	
	Miranda-Jiménez et al. (2017)	✓	✓	✓		
2018	Arif et al. (2018)	✓	✓			
2019	Emadi and Rahgozar (2019)	✓	✓			

4 Meta-level features

Meta-level features, also called hand-crafted features, are usually extracted from other features and can capture insightful new information about the data (Canuto et al. 2016), exploring the content of tweets more efficiently than merely relying on raw sequences of words. In this study, we consider as meta-level features those referred to counts and summations, which are, in general, secondary information extracted from tweets. Meta-level features are referred to hereafter as meta-features.

Table 3 Overview of the meta-level features proposed in the literature of Twitter sentiment classification

Category	Features
Microblog (*10 features*)	*Wheter the tweet has*: retweet, hashtag, user mention, URL, repeated letters, abbreviation, internet slang
	Number of: repeated letters, abbreviation, internet slang
Part-of-Speech (*25 features*)	*Number of*: common noun, proper noun, personal pronoun, common noun + possessive, common noun + verb, proper noun + possessive, proper noun + verb, verb, adjective, adverb, interjection, punctuation, determiner, pre or post-position, conjunction, verb particle, predeterminer, predeterminer + verb, hashtag, user mention, discourse marker ("RT" and ":" in retweet), URL or email address, numeral, symbol
Surface (*15 features*)	*Whether the tweet has*: question mark, exclamation mark
	Whether last token contains: question mark, exclamation mark
	Number of: words, capitalized words, words with all letters capitalized, capital letters, punctuation, question mark, exclamation mark, sequence of question marks, sequence of exclamation marks, sequence of both question and exclamation marks
	Average of: char length of words
Emoticon (*10 features*)	*Wheter the tweet has*: emoticon, positive emoticon, negative emoticon
	Whether the last token is: positive emoticon, negative emoticon
	Number of: emoticons, positive emoticons, negative emoticons, extremely positive emoticons, extremely negative emoticons
Lexicon-based (*70 features*)	*Number of*: positive adjective, negative adjective, positive noun, negative noun, positive adverb, negative adverb, positive verb, negative verb, negated contexts, negation words, intensifier words, counter factuality words, temporal compression words
	Sum of the scores of the adjectives, adverbs, verbs, and nouns
	For each sentiment lexicon (AFINN Nielsen 2011, Bing Liu's lexicon Liu 2012, NRC-emotion Mohammad and Turney 2013, NRC-hashtag Mohammad et al. 2013, OpinionFinder Wilson et al. 2005 Sentiment140 lexicon Mohammad et al. 2013, and SentiWordNet Baccianella et al. 2010):
	– *Number of*: positive words, negative words
	– *Total score of*: positive words, negative words
	– *Maximal score of*: positive words, negative words
	– *Balance score of the tweet*
	– *Score of the last token*

In this section, we present and categorize the most common types of meta-features we have examined in a set of well-referenced works in supervised sentiment classification of tweets (Agarwal et al. 2011; Barbosa and Feng 2010; Bravo-Marquez et al. 2014; Buscaldi and Hernandez-Farias 2015; da Silva et al. 2014; Davidov et al. 2010; Go et al. 2009; Hagen et al. 2015; Jiang et al. 2011; Khuc et al. 2012; Kouloumpis et al. 2011; Mohammad et al. 2013; Park et al. 2018; Vo and Zhang 2016; Zhang et al. 2011). This categorization is an extension of the study presented in Carvalho and Plastino (2016). In this work, the categories proposed in Carvalho and Plastino (2016) are revisited. For this purpose, considering that features sharing structural aspects should fall into the same group, we have categorized them into five categories, namely: Microblog, Part-of-Speech, Surface, Emoticon, and Lexicon-based features. An overview of the meta-features and their respective categories are presented in Table 3. The number in parentheses right below the name of

each category corresponds to the total number of features in that category. In the following, we describe each category of meta-features.

Microblog features. The Microblog category refers to those features that leverage the syntax and the vocabulary used in tweets and microblog messages, as used in Agarwal et al. (2011), Barbosa and Feng (2010), Hagen et al. (2015), Jiang et al. (2011), Kouloumpis et al. (2011), Mohammad et al. (2013), Zhang et al. (2011). More specifically, some characteristics of how microblog posts are written may be good indicators of sentiment, such as the use of repeated letters and internet slang present in the vocabulary of this type of text. Furthermore, Twitter-specific tokens, such as user mentions (followed by the special character @), retweets (indicated by RT), URLs, and hashtags (followed by the special character #) have also been explored in the literature.

Part-of-Speech features. Although some studies have already acknowledged that part-of-speech (POS) features are not useful for sentiment classification (Go et al. 2009; Pang et al. 2002), this category of features is still used to determine the sentiment of tweets, in combination with other features (Agarwal et al. 2011; Barbosa and Feng 2010; Bravo-Marquez et al. 2014; Go et al. 2009; Kouloumpis et al. 2011; Mohammad et al. 2013). For example, assuming that some adjectives and verbs are good indicators of positive and negative sentiment, Barbosa and Feng (2010) map each word in a tweet to its POS, being able to identify nouns, verbs, adjectives, adverbs, interjections, and others. Similarly, Agarwal et al. (2011) consider the number of adjectives, adverbs, verbs, and nouns as features. In order to capture the informal aspects of tweets, some works (Bravo-Marquez et al. 2014; Mohammad et al. 2013) use a POS tagset, presented in Gimpel et al. (2011), to identify some special characteristics of short and noisy texts, such as misspelling words.

Surface features. Surface features capture superficial stylistic content of the tweet, such as the number of words, capitalized words, words with all caps, capital letters, and punctuation (Agarwal et al. 2011; Barbosa and Feng 2010; Davidov et al. 2010; Hagen et al. 2015; Jiang et al. 2011; Kouloumpis et al. 2011; Mohammad et al. 2013; Park et al. 2018). Punctuation may also play an important role in sentiment detection of microblog messages. Thus, punctuation features have also been explored in the literature (Agarwal et al. 2011; Barbosa and Feng 2010; Davidov et al. 2010; Hagen et al. 2015; Jiang et al. 2011; Mohammad et al. 2013; Park et al. 2018). The most usual meta-features in this category are the number of exclamation and question marks, as appearing in (Agarwal et al. 2011; Barbosa and Feng 2010; Davidov et al. 2010; Hagen et al. 2015; Jiang et al. 2011; Park et al. 2018). Some works have already proposed more sophisticated meta-features, such as the number of contiguous sequences of exclamation and question marks (Hagen et al. 2015; Mohammad et al. 2013), regarding their use in microblog messages to convey intonation.

Emoticon features. The polarity of emoticons may also be another relevant characteristic for Twitter sentiment analysis. Since emoticons are used by microblog users to summarize the sentiment they intend to communicate, some works have also extracted meta-features from emoticons, such as the number of positive and negative emoticons in a tweet, as employed in Agarwal et al. (2011), da Silva et al. (2014), Hagen et al. (2015), Mohammad et al. (2013), Park et al. (2018).

Lexicon-based features. A different manner of exploring the content of tweets in order to determine the sentiment expressed in them is from using existing sentiment lexical resources or dictionaries in the literature. These lexicons consist of lists of words with positive and negative terms, such as Bing Liu's opinion lexicon (Liu 2012), NRC-emotion (Mohammad and Turney 2013), and OpinionFinder lexicon (Wilson et al. 2005), as well as lexical resources containing words and phrases that are scored on a range of real values, such as AFINN (Nielsen 2011), SentiWordNet (SWN) (Baccianella et al. 2010),

NRC-hashtag (Mohammad et al. 2013), and Sentiment140 lexicon (Sent140) (Mohammad et al. 2013). Meta-features of this category have been widely explored in sentiment classification of tweets (Agarwal et al. 2011; Bravo-Marquez et al. 2014; Buscaldi and Hernandez-Farias 2015; da Silva et al. 2014; Hagen et al. 2015; Jiang et al. 2011; Khuc et al. 2012; Kouloumpis et al. 2011; Mohammad et al. 2013; Vo and Zhang 2016), especially the total count of positive and negative words.

It has already been acknowledged that negation can affect the polarity of an expression (Wiegand et al. 2010). Indeed, the expression *not good* is the opposite of *good*. In this context, an interesting meta-feature proposed in the literature to handle negation is the number of negated contexts (Mohammad et al. 2013). Mohammad et al. (2013) have defined a negated context as a segment of a tweet that starts with a negation word, such as *shouldn't*, and ends on the first punctuation mark after the negation word.

Regarding irony, Reyes et al. (2013) argue that it represents a meaningful obstacle for determining the polarity of texts accurately. For example, in domains like politics, health campaigns, and natural disasters, Twitter users post ironic messages criticizing and blaming the government, and most sentiment analysis models cannot deal properly with them. To this end, in Reyes et al. (2013), they have proposed features to help capture irony in text, such as the number of counter-factuality words (e.g., *nonetheless*, *nevertheless*) and temporal compression words (e.g., *suddenly*, *now)*, which have been used in Twitter sentiment analysis (Buscaldi and Hernandez-Farias 2015). As described in Reyes et al. (2013), while counter-factuality words are discursive terms that hint at contradiction in a text, temporal compression words are focused on identifying elements related to the opposition in time, i.e., words that indicate an abrupt change in a narrative.

5 Word embedding-based features

Although the well-known bag-of-words and *n*-gram representations have been extensively used regarding their simplicity, they make the feature space highly dimensional leading to the curse of dimensionality, as discussed in Sect. 2. Also, hand-crafted features are time-consuming, often incomplete, and requires significant human effort (Pouyanfar et al. 2018; Young et al. 2018). On the other hand, deep learning approaches perform feature extraction in an automated way, which allows researchers to extract discriminative features with minimal domain knowledge (Pouyanfar et al. 2018). In recent research, with the increasing interest in deep learning approaches for NLP applications, distributed representations of words in a vector space, or word embeddings, have received much attention due to their ability to achieve high performance in many text classification tasks.

Word embeddings can capture the semantic and syntactic relations between words from a large amount of unlabeled text data, representing them in dense real-valued vectors that can be used as features in supervised machine learning frameworks. As described in Bengio et al. (2003), the feature vectors associated with each word are learned from large corpora, and each value represents a different aspect, or dimension, of the word. The main idea is that words that frequently occur together in the same contexts are mapped to similar regions of the vector space (Agarwal et al. 2018).

In Mikolov et al. (2013a), Mikolov et al. have designed the word2vec tool (w2v), comprising the CBOW and the Skip-gram models, which are neural architectures to train word embeddings. More specifically, given a massive text corpus, these architectures learn vector representation of words based on its vocabulary. As described in Mikolov et al. (2013a), the

CBOW method predicts the source word based on its context, while the Skip-gram predicts nearby words given a source word. Later, in Mikolov et al. (2013b), they have improved the Skip-gram model, making it much more computationally efficient. In Mikolov et al. (2013b), they have used an internal dataset of news articles from Google with one billion words to train the model, generating a 300-dimensional word vector. Pennington et al. (2014) argue that the statistics of the words in a given training corpus are sub utilized by the Skip-gram model (Mikolov et al. 2013b) since it does not take into account global co-occurrence counts of words. For that reason, they propose a weighted least squares model, namely GloVe (Global Vectors), that leverages global word-word co-occurrence counts in the word embedding training phase. They have trained a 300-dimensional word vector and evaluated the proposed model on the word analogy, word similarity, and named entity recognition tasks, proving that GloVe outperforms the w2v models (CBOW and Skip-gram) by a significant margin.

Most techniques to train word vectors ignore the internal structure of words, making it difficult to learn good representations for morphologically rich languages, which have many different inflected forms for the same word. Thus, Bojanowski et al. (2016) have proposed the fastText model, which learn representations for character *n*-grams as an extension of the Skip-gram model (Mikolov et al. 2013b). Later, Mikolov et al. (2018) have combined some pre-processing strategies rarely used together to improve the standard fastText model and achieved state-of-the-art results on several tasks.

Arguing the inefficiency of traditional approaches to train word embeddings for sentiment analysis, some authors have designed solutions to train word vectors specifically for the sentiment analysis task (Agarwal et al. 2018; Felbo et al. 2017; Tang et al. 2014; Xu et al. 2018). Tang et al. (2014) developed a neural network to learn sentiment-specific word embeddings (SSWE) on a massive corpus of tweets. They used the SSWE word vectors as features in a supervised machine learning strategy and reported comparable results with those achieved by applying the meta-level features proposed in Mohammad et al. (2013). Felbo et al. (2017) took advantage of the vast amount of emoji occurrences on tweets to train models with rich emotional representations by using a transfer learning approach, namely DeepMoji. They have evaluated the DeepMoji model on eight benchmark datasets for the emotion, sarcasm, and sentiment classification tasks and their results outperformed state-of-the-art results for all assessed datasets, including the results achieved with the SSWE (Tang et al. 2014) method.

In Xu et al. (2018), Xu et al. proposed Emo2Vec, which is a multi-task training framework that incorporates six different emotion-related tasks in the training process, such as sentiment analysis, emotion classification, sarcasm detection, abusive language classification, stress detection, insult classification, and personality recognition. They argue that including the affective information from all those domains may benefit the learning process, thus enabling the creation of a more general embedding emotional space. Compared with the SSWE and DeepMoji models, the Emo2Vec word vectors achieved competitive results. Also, claiming that Emo2Vec is weak on capturing the syntactic and semantic meaning of words, they concatenated Emo2Vec with the pre-trained GloVe (Pennington et al. 2014) vectors for comparison with state-of-the-art results on 14 datasets from distinct domains. In the experimental evaluation, the combination of Emo2Vec with GloVe vectors fed to an LR classifier achieved comparable performance for some datasets.

Discussing the challenges of the emotion classification problem, Agarwal et al. (2018) address some limitations of this task by leveraging noisy training data with a large range of emotions to learn emotion-enriched word representations, namely Emotion Word Embeddings (EWE). Instead of tweets, they have explored product reviews, as this type of text

may generalize better for other domains. They have evaluated the predictive performance of EWE against state-of-the-art pre-trained word vectors (Felbo et al. 2017; Mikolov et al. 2013b; Pennington et al. 2014; Tang et al. 2014) on four datasets from various domains, such as fairy tales, blogs, experiences, and tweets. To this end, they have used LR and SVM as the learning strategies showing that the proposed method outperforms all the other methods with a statistically significant difference.

Recent advances in language modeling using neural networks have made it viable to model language as distributions over characters (Akbik et al. 2018). Akbik et al. (2018) have proposed a method that passes sentences as sequences of characters into a character-level language model to form word-level embeddings. The character-level language model can capture syntactic-semantic word features and disambiguate words in context, resulting in state-of-the-art performance in NLP sequence labeling tasks. The method proposed in Akbik et al. (2018) can produce different embeddings for the same word depending on its context. Later, in Akbik et al. (2019), Akbik et al. have developed FLAIR,[3] an NLP framework which facilitates the training and distribution of state-of-the-art language models, thus reducing architectural complexity.

Recently, Peters et al. (2018) introduced ELMo (Embeddings from Language Models), a deep contextualized word representation that models not only complex characteristics of word usage, such as syntax and semantics, but also how these uses vary across linguistic contexts. ELMo is a feature-based approach for applying pre-trained language representations to downstream tasks, which extracts and concatenates independently context-sensitive features from a left-to-right and a right-to-left language model.

In contrast to ELMo (Peters et al. 2018), which is not deeply bidirectional since it simply concatenates left-to-right and right-to-left representations, Devlin et al. (2019) presented a strategy called BERT (Bidirectional Encoder Representations from Transformers), which is a fine-tuning approach that alleviates the unidirectionality constraint by applying masked language models to enable pre-trained deep bidirectional representations. This model randomly masks some percentage of the input tokens, and the objective is to predict those masked tokens based on their context (Devlin et al. 2019). In addition, they also use a next sentence prediction (NSP) task for predicting whether two text segments follow each other in the original text. In Devlin et al. (2019), Devlin et al. showed that BERT outperforms many task-specific architectures and achieved state-of-the-art performance for eleven NLP tasks.

Later, in Liu et al. (2019), Liu et al. proposed modifications for training BERT models, which they called RoBERTa (Robustly optimized BERT approach). For example, regarding the masked language model, while BERT performs masking once during data preprocessing, resulting in a single static mask, RoBERTa applies dynamic masking where the masking pattern is generated every time a sequence is fed to the model. Also, Liu et al. questioned the necessity of the NSP loss and trained RoBERTa without it. In Liu et al. (2019), Liu et al. combined these improvements and evaluated their combined impact on downstream tasks using three benchmarks datasets, achieving state-of-the-art results.

Another point recently investigated in the literature is the application of capsule networks in NLP tasks (Yang et al. 2018; Zhao et al. 2019). Capsule networks (Sabour et al. 2017) are neural network architectures proposed in the domain of image classification which tackle some problems with convolutional neural networks (CNN). For

[3] https://github.com/zalandoresearch/flair.

Table 4 Characteristics of the pre-trained word embeddings separated by type and ordered by the number of dimensions ($|D|$ column)

Type	Embedding	$\|D\|$	$\|V\|$	Corpus
Generic	GloVe-TWT (Pennington et al. 2014)	200	1.2 M	Twitter (27B tokens)
	GloVe-WP (Pennington et al. 2014)	300	400 K	Wikipedia/Gigaword (6B tokens)
	fastText (Mikolov et al. 2018)	300	1 M	Wikipedia/web pages/news (16B tokens)
	w2v-GN (Mikolov et al. 2013b)	300	3 M	Google news (100B tokens)
	w2v-Edin (Bravo-Marquez et al. 2016)	400	259 K	Twitter (10M tweets)
	w2v-Araque (Araque et al. 2017)	500	57 K	Twitter (1.28M tweets)
Affective	SSWE (Tang et al. 2014)	50	137 K	Twitter (10M tweets)
	Emo2Vec (Xu et al. 2018)	100	1.2 M	Twitter (1.9M tweets)
	DeepMoji (Felbo et al. 2017)	256	50 K	Twitter (1B tweets)
	EWE (Agarwal et al. 2018)	300	183 K	Amazon reviews (200K reviews)

example, Zhao et al. (2019) argue that pooling operations in CNNs wrongly discard positional information and do not consider hierarchical relationships between local features, thus requiring massive amounts of training samples for generalization. On the other hand, capsule networks have the ability to learn hierarchical relationships between consecutive layers by using routing processes (Zhao et al. 2019). In Zhao et al. (2019), Zhao et al. extended existing capsule networks regarding NLP tasks, and achieved state-of-the-art results on question answering and multi-label text classification.

It has already been acknowledged that achieving suitable and sufficient representations of words depends on the volume of data used to train the word embedding models. Much effort in recent research is mainly focused on scalability issues of existing methods. For that reason, many researchers make the word vectors trained with their architectures available for public use. Those publicly available word vectors are referred to as pre-trained word embeddings.

Table 4 presents the characteristics of the pre-trained word embeddings generated by some methods discussed in this section. The $|D|$ and $|V|$ columns refer to the dimension and vocabulary size of each pre-trained embedding, respectively. The Type column separates the word embeddings trained for general purpose (generic) from those specially trained for the sentiment analysis and emotion detection tasks (affective). Additionally, under the Corpus column, we present information about the textual corpora used to train the embeddings.

As described in Table 4, the GloVe-TWT and GloVe-WP word vectors (Pennington et al. 2014) were trained on massive text corpora from Twitter and Wikipedia+Gigaword, respectively. The fastText vectors (Mikolov et al. 2018) were trained on rich and vast sources of data, including Wikipedia, news from statmt.org, and the UMBC text corpus.

Regarding the word vectors trained with the word2vec tool, w2v-GN is the former one whose construction is detailed in Mikolov et al. (2013b). Bravo-Marquez et al. (2016) have used the Skip-gram method implemented in the word2vec tool to train word vectors on a vast corpus of ten million tweets from the Edinburgh Twitter corpus (Petrović et al. 2010). In Bravo-Marquez et al. (2016), they have optimized the parameters for classifying words into emotions and made the pre-trained vectors publicly available (w2v-Edin). More recently, Araque et al. (2017) developed a supervised learning system using word vectors as features. The w2v-Araque vectors were trained on a corpus of 1,280,000 tweets with the word2vec tool, and the system was used as a baseline to compare it to other approaches.

Regarding the affective pre-trained vectors, which leverage the sentiment or emotion information during the training phase, the SSWE (Tang et al. 2014), Emo2Vec (Xu et al. 2018), and DeepMoji (Felbo et al. 2017) word vectors were trained on tweets, while the EWE (Agarwal et al. 2018) representations were trained on product reviews from Amazon. All of them were generated using specific methods for creating word representations to incorporate the sentiment information of texts during the training process.

6 Experimental evaluation

This section presents the computational results obtained by evaluating the different feature representations presented in Sects. 3, 4, and 5, namely *n*-grams, meta-level features, and word embeddings, respectively. Next, we present the results achieved by combining those distinct types of features through combination strategies, such as feature concatenation and classifier ensembles, as discussed in Sect. 2.

We begin by describing the experimental protocol we followed in Sect. 6.1. Then, in Sects. 6.2 and 6.3, we report and discuss the results of a set of experiments so as to arrive at answers to research questions RQ1 through RQ3, introduced in Sect. 1.

6.1 Experimental setting

Attempting to answer the research questions presented in Sect. 1, we adopted Weka's (Hall et al. 2009) implementation of machine learning algorithms Support Vector Machines (SVM), L2-regularized Logistic Regression (LR), and Random Forests (RF). For SVM and LR, we used the LIBSVM[4] (Chang and Lin 2011) and LIBLINEAR[5] (Fan et al. 2008) implementations, respectively. Also, we set the regularization parameter to its default value (C = 1.0), and we employed the linear kernel for LIBSVM. Table 5 shows a summary of the classification algorithms adopted in this work, remarking their advantages and disadvantages.

We used a set of twenty-two datasets for the computational experiments reported in this section. These datasets have been extensively used in the literature of Twitter sentiment analysis. To the best of our knowledge, this is the first study using a significant number of Twitter datasets in the evaluation of different types of features employed in the literature over the years. The datasets are: irony (Gonçalves et al. 2015), sarcasm (Gonçalves et al. 2015), aisopos,[6] SemEval-Fig (Ghosh et al. 2015), sentiment140 (Go et al. 2009), person (Chen et al. 2012), hobbit (Lochter et al. 2016), iphone6 (Lochter et al. 2016), movie (Chen et al. 2012), sanders,[7] Narr (Narr et al. 2012), archeage (Lochter et al. 2016), SemEval18 (Mohammad et al. 2018), OMD (Diakopoulos and Shamma 2010), HCR (Speriosu et al. 2011), STS-Gold (Saif et al. 2013), SentiStrength (Thelwall et al. 2012), Target-dependent (Dong et al. 2014), Vader (Hutto and Gilbert 2014), SemEval13 (Nakov et al. 2013), SemEval16 (Nakov et al. 2016), and SemEval17

[4] Available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[5] Available at http://www.csie.ntu.edu.tw/~cjlin/liblinear.
[6] http://grid.ece.ntua.gr.
[7] http://www.sananalytics.com/lab/twitter-sentiment.

Table 5 Summary of the classification algorithms adopted in this work

Algorithm	Advantages	Disadvantages
SVM	SVM performs relatively well in high dimensional spaces	SVM may underperform in cases where number of features exceeds the number of training instances
	SVM works well when there is clear margin of separation between classes	Choosing a good kernel function is not easy
LR	LR is highly interpretable	LR does not solve non-linear problems
	LR is very efficient to train	LR does not perform well with independent variables that are not correlated to the target variable
RF	RF is robust to outliers and non-linear data	RF models are difficult to interpret
	RF can handle unbalanced data	For very large datasets, RF can be very computationally expensive

Table 6 Characteristics of the Twitter sentiment datasets ordered by size (*#tweets* column)

Dataset	#tweets	#positive	#negative
Irony (Gonçalves et al. 2015)	65	22	43
Sarcasm (Gonçalves et al. 2015)	71	33	38
Aisopos	278	159	119
SemEval-Fig (Ghosh et al. 2015)	321	47	274
Sentiment140 (Go et al. 2009)	359	182	177
Person (Chen et al. 2012)	439	312	127
Hobbit (Lochter et al. 2016)	522	354	168
Iphone6 (Lochter et al. 2016)	532	371	161
Movie (Chen et al. 2012)	561	460	101
Sanders	1224	570	654
Narr (Narr et al. 2012)	1227	739	488
Archeage (Lochter et al. 2016)	1718	724	994
SemEval18 (Mohammad et al. 2018)	1859	865	994
OMD (Diakopoulos and Shamma 2010)	1906	710	1196
HCR (Speriosu et al. 2011)	1908	539	1369
STS-gold (Saif et al. 2013)	2034	632	1402
SentiStrength (Thelwall et al. 2012)	2289	1340	949
Target-dependent (Dong et al. 2014)	3467	1734	1733
Vader (Hutto and Gilbert 2014)	4196	2897	1299
SemEval13 (Nakov et al. 2013)	4378	3183	1195
SemEval17 (Rosenthal et al. 2017)	6347	2375	3972
SemEval16 (Nakov et al. 2016)	12,216	8893	3323

(Rosenthal et al. 2017). Some characteristics of these datasets are presented in Table 6, namely their total number of tweets, positive, and negative tweets.

It is worth mentioning that the original sentiment140 dataset, as described in Go et al. (2009), contains 1.6 million training tweets, which were annotated using a distant supervision approach, meaning that emoticons from such tweets were used as noisy labels. However, in our experiments, we have only used the test partition, which contains 177 negative and 182 positive tweets manually labeled by a set of human annotators. Although the test partition is relatively small, it has been widely used in Twitter sentiment analysis (Bakliwal et al. 2012; Bravo-Marquez et al. 2013; Saif et al. 2013, 2012; Speriosu et al. 2011).

In the experimental evaluation, the predictive performance of the sentiment classification is measured in terms of classification accuracy. For each evaluated dataset, the accuracy of the classification was computed as the ratio between the number of correctly classified tweets and the total number of tweets, following a stratified tenfold cross-validation.

Moreover, as suggested by Demšar (2006), we ran the Friedman test followed by the Nemenyi post-hoc test to determine whether the differences among the accuracies are statistically significant at a 0.05 significance level. Whenever applicable, we present the results of the statistical tests immediately below each results table. For this purpose, we use the symbol $\succ$ to show that some classifier x is significantly better than some classifier y, so that $\{x\} \succ \{y\}$.

6.2 Answering research question RQ1

The experiments conducted in this section aim at answering research question RQ1, as follows:

RQ1. Which group of features is the most effective in Twitter sentiment analysis?

We answer this question throughout Sects. 6.2.1, 6.2.2, and 6.2.3, by assessing the distinct groups of features we have identified in the literature. Those features include *n*-grams, meta-features, and word embeddings. Then, after determining the best classifiers for each group of features, we perform a comparison between them to determine the most representative one for Twitter sentiment analysis. The discussion on this comparison is presented in Sect. 6.2.4.

Besides the comparative evaluation of the feature sets, we present an analysis of the categories of the meta-features introduced in Sect. 4, as well as an assessment study of a significant collection of pre-trained embedding models in Sects. 6.2.2 and 6.2.3, respectively.

6.2.1 Effectiveness of *n*-gram features

The *n*-gram features used in the computational experiments reported in this section are unigrams, bigrams, and trigrams. We do not explore higher-order *n*-grams so as to try and minimize the negative effect of high dimensionality. Besides, unigrams, bigrams, and trigrams are the most adopted *n*-gram features in the literature of sentiment detection in tweets, as previously seen in Table 2.

As a preprocessing step, we used the same strategy as done in Mohammad et al. (2013). Each tweet was tokenized and labeled according to their part-of-speech tag, using the Twitter-specific part-of-speech tag set tool (Gimpel et al. 2011). This tag set consists of twenty-five POS tags, specifically designed for tweets, that takes into account the different aspects that tweets have as compared to regular text. Then, for each tweet in a given dataset, we replaced URLs by the token "http://someurl" and user mentions by the token "@someuser". Regarding stopwords removal, we discarded stopwords only as unigrams, since it has been acknowledged that stopwords can affect the polarity of some expressions in higher-order *n*-grams (Speriosu et al. 2011). Finally, considering that negation words[8] ("shouldn't", for example) can affect the *n*-gram-based features, we handle negation by employing the same approach used by Mohammad et al. (2013). In Mohammad et al. (2013), negated contexts change *n*-gram-based features. Specifically, they add the tag _NEG on each token within a negated context. More precisely, in a negated context, Mohammad et al. concatenate the tag _NEG to every token between the negation word and the first punctuation mark after it. For example, in the sentence "He isn't a great book writer, but I read his books.", the unigrams "great", "book", and "writer" become "great_NEG", "book_NEG", and "writer_NEG", respectively.

After preprocessing all tweets and extracting the *n*-gram features, the feature space is represented by a binary feature vector indicating whether each *n*-gram existing in the vocabulary occurs in the tweet or not. In that case, the values 0 and 1 represent the absence and presence of each *n*-gram in the tweet, respectively.

Table 7 shows the results of the evaluation of the *n*-gram features in terms of classification accuracy (%), as well as the number of features extracted for each dataset (#features

[8] We used the negation words available at http://sentiment.christopherpotts.net/lingstruc.html#negation.

Table 7 Accuracies (%) achieved by evaluating the n-gram features using SVM, LR, and RF classifiers, respectively

Dataset	#features	SVM	LR	RF
Irony	1.8K	66.2	66.2	66.2
Sarcasm	1.8K	50.7	**52.1**	46.5
Aisopos	6.5K	**87.8**	87.4	72.7
SemEval-Fig	8.8K	**91.0**	90.0	85.4
Sentiment140	7.6K	84.1	**84.4**	83.0
Person	10.0K	79.0	**79.5**	71.8
Hobbit	8.5K	92.9	**93.3**	87.5
Iphone6	9.4K	77.6	**78.0**	77.4
Movie	10.2K	**84.1**	83.2	82.0
Sanders	23.6K	**83.0**	81.6	73.1
Narr	24.2K	**83.7**	82.6	73.7
Archeage	28.2K	**86.3**	85.9	82.8
SemEval18	42.0K	**80.2**	79.2	71.9
OMD	32.1K	81.2	**82.4**	77.5
HCR	40.5K	79.1	**79.5**	76.7
STS-gold	37.4K	**84.0**	83.6	74.6
SentiStrength	49.4K	**73.2**	72.4	64.1
Target-dependent	66.6K	81.4	**82.0**	78.8
Vader	68.4K	**84.8**	83.3	75.5
SemEval13	105.0K	**81.0**	79.9	74.1
SemEval17	127.6K	86.9	**87.1**	84.5
SemEval16	252.1K	**85.8**	85.0	74.0
#wins		**12**	9	0
Rank sums		**31.5**	34.5	64.5

$\{\text{SVM}, \text{LR}\} \succ \{\text{RF}\}$

column). The boldfaced values indicate the best results, and the total number of wins for each classifier is presented in the #wins row. Also, we compute a ranking to make a fair comparison between the results. Precisely, for each dataset, we assign scores from 1.0 to 3.0 for each tested situation (each column), in ascending order of accuracy, where the score 1.0 is assigned to the situation with the highest accuracy. Thus, low score values indicate better results. Finally, we sum up the assigned scores for each classifier, as shown in the rank sums row.

As we can observe in Table 7, the best results were achieved by SVM in 12 out of the 22 datasets. Indeed, SVM has proven its robustness on large feature spaces in Twitter sentiment analysis (Mohammad et al. 2013). The LR classifier achieved comparable performance to SVM. Conversely, the worse performance was achieved by the RF classifier. The poor performance of RF may be due to the sparse nature of the data, in which most feature values are zero, increasing the risk of selecting a subset of irrelevant or noisy features when splitting the data at an internal node in the tree.

Another point worth highlighting is that the *n*-gram model does not seem to be a good choice for representing tweets from datasets irony and sarcasm. This can be justified by the rather small numbers of tweets these datasets contain, that is, 65 and 71, respectively. It appears that the *n*-gram-based features may not be representative enough in the sentiment classification of the tweets from these datasets, since classification is performed based on

Table 8 Accuracies (%) achieved by evaluating the meta-features using SVM, LR, and RF classifiers, respectively

Dataset	SVM	LR	RF
Irony	76.9	78.5	**81.5**
Sarcasm	71.8	69.0	**80.3**
Aisopos	**94.2**	93.5	92.8
SemEval-Fig	88.5	90.0	**90.3**
Sentiment140	85.2	**85.5**	85.0
Person	82.2	82.5	**83.6**
Hobbit	88.9	89.5	**91.6**
Iphone6	80.5	81.4	**82.5**
Movie	85.6	86.5	**87.0**
Sanders	81.0	80.9	**84.8**
Narr	89.6	89.5	**90.3**
Archeage	84.6	**85.4**	**85.4**
SemEval18	85.6	85.2	**86.0**
OMD	78.1	78.2	**79.8**
HCR	75.8	76.0	**77.5**
STS-gold	92.2	91.8	**93.1**
SentiStrength	83.2	**83.6**	83.3
Target-dependent	**83.3**	82.9	83.1
Vader	**93.3**	93.2	93.0
SemEval13	86.4	86.7	**86.9**
SemEval17	86.4	86.3	**86.5**
SemEval16	**85.6**	85.3	85.4
#wins	4	3	**16**
Rank sums	51.0	49.5	**31.5**

$\{RF\} \succ \{SVM, LR\}$

the vocabulary extracted from the training set, that is, the *n*-grams themselves. Finally, the Friedman test followed by the Nemenyi post-hoc test detected that both SVM and LR are significantly better than RF for this particular type of feature, but there is no significant difference between them.

6.2.2 Effectiveness of meta-level features

In this section, we present an assessment study of the meta-features in two parts. First, we show and compare the predictive performance of SVM, LR, and RF by using the full set of meta-features, in order to isolate the most appropriate one when this type of feature is exploited. Then, we evaluate each category of meta-features to identify the most effective one in the sentiment classification of tweets.

The meta-features evaluated in this section are those described and categorized in Sect. 4 (see Table 3 for details). To determine the polarity of adjectives, nouns, adverbs, and verbs, we used the SentiWordNet sentiment lexicon (Baccianella et al. 2010). In addition, we made use of the internet slang and emoticon lists introduced in Agarwal et al. (2011) in order to identify such language in the tweets. For abbreviations, we adopted the Internet Lingo Dictionary (Wasden 2010), as employed in Kouloumpis et al. (2011).

Table 9 Accuracies (%) achieved by evaluating each category of meta-features using a RF classifier

Dataset	MICRO 10	POS 25	SUR 15	EMO 10	LEX 70	ALL 130
Irony	64.6	61.5	60.0	66.2	76.9	**81.5**
Sarcasm	59.2	54.9	47.9	53.5	**81.7**	80.3
Aisopos	61.2	68.7	54.0	91.4	82.4	**92.8**
SemEval-Fig	87.5	87.2	83.5	85.0	87.5	**90.3**
Sentiment140	59.1	49.9	53.5	54.9	84.1	**85.0**
Person	66.3	69.2	67.4	71.1	83.1	**83.6**
Hobbit	66.5	74.7	67.4	70.1	**91.8**	91.6
Iphone6	65.4	77.3	73.1	69.5	82.3	**82.5**
Movie	80.9	81.6	79.3	82.2	86.6	**87.0**
Sanders	60.2	68.2	66.3	57.0	83.6	**84.8**
Narr	62.3	65.5	62.9	62.8	90.0	**90.3**
Archeage	71.1	75.7	72.0	65.6	84.1	**85.4**
SemEval18	54.8	59.7	56.4	57.2	85.8	**86.0**
OMD	62.1	65.2	64.8	63.0	77.6	**79.8**
HCR	69.9	73.2	69.3	71.9	76.2	**77.5**
STS-gold	68.0	69.7	66.8	69.1	**93.5**	93.1
SentiStrength	60.6	60.9	59.0	60.2	82.6	**83.3**
Target-dependent	52.1	59.4	56.2	51.1	83.0	**83.1**
Vader	68.7	71.4	66.7	71.0	92.1	**93.0**
SemEval13	71.7	73.5	70.5	73.8	86.1	**86.9**
SemEval17	66.0	70.0	67.5	64.4	86.3	**86.5**
SemEval16	72.6	72.9	70.4	73.1	85.3	**85.4**
#wins (categories)	1	0	0	1	**21**	–
Rank sums (categories)	85.5	56.0	92.0	73.0	**23.5**	–
#wins (overall)	0	0	0	0	3	**19**

$\{\text{LEX}\} \succ \{\text{MICRO}, \text{POS}, \text{SUR}, \text{EMO}\}$

$\{\text{POS}\} \succ \{\text{MICRO}, \text{SUR}\}$

The results of the first experiment are reported in Table 8. The RF classifier performed significantly better than SVM and LR, achieving the highest accuracies in 16 out of the 22 datasets. Although RF may not be a good choice on sparse feature spaces, it is robust to outliers, noise, and can handle class imbalance (Breiman 2001). Those characteristics may have allowed for an improvement in classification accuracy, as compared to SVM and LR. In general, SVM and LR achieved comparable performances. However, in spite of LR performing slightly better than SVM, as shown in the rank sums row, there is no significant difference between the results achieved.

Categories of meta-level features. The second part of the experiments reported in this section consists of determining the most predictive categories of meta-features, following the categorization proposed in Sect. 4.

Table 9 presents the accuracies achieved by assessing each category of meta-features using an RF classifier (MICRO, POS, SUR, EMO, and LEX columns), and their comparison with the results achieved by using the set of all meta-features (ALL column). We used an RF classifier since it achieved significantly better results than SVM and LR in the previous experiment. The best overall results are in boldface type, and the best results among

the categories are underlined. The values immediately below each category name refer to the number of features in that particular category.

As we can see in Table 9, the category Lexicon-based (LEX column) achieved the best results among all the categories, with the highest number of wins in 21 out of the 22 datasets. In the overall evaluation, this category outperformed the set of all meta-features in three datasets (sarcasm, hobbit, and STS-gold). None of the other categories, namely Microblog (MICRO column), Part-of-speech (POS column), Surface (SUR column), and Emoticon (EMO column) achieved meaningful results. The Friedman and the Nemenyi tests detected that category Lexicon-based is significantly better than all other categories, while category Part-of-speech is only better than the categories Microblog and Surface.

Although category Emoticon does not seem to be useful in the sentiment classification of tweets, it is worth pointing out that it achieved the best accuracy for the dataset aisopos. Analyzing the tweets from this dataset, we note that about 80% of them contain emoticons. Since the polarity of emoticons are taken into account in this category's features, the sentiment detection of tweets may benefit from this information. As a matter of fact, analyzing the most informative meta-features for this dataset by ranking all of them with the Information Gain (IG) relevance measure, four out of the top five most relevant features are *whether the tweet has negative emoticon*, *number of negative emoticons*, *whether the last token is negative emoticon*, and *whether the tweet has positive emoticon*, all of them from category Emoticon.

Still, regarding the rank generated by the IG measure, the meta-features belonging to category Surface appear at the bottom of the rank for most datasets. This is reasonable, taking into account that this category achieved the worst performance across all categories assessed, as shown in Table 9. Interestingly, for datasets irony and OMD, surface features referring to punctuation, such as *whether the tweet has exclamation mark*, *number of exclamation mark*, *whether the tweet has question mark*, and *number of question mark* are ranked among the top 25 most significant features. This fact is in agreement with recent findings on the irony detection task, which acknowledged that punctuation marks are useful to identify irony, especially in tweets (Farias and Rosso 2017). Also, it is worth mentioning that dataset OMD, whose tweets are related to a political debate, may contain ironic content due to its nature.

In addition to the individual assessment of each category, we also investigate the reverse situation. We analyze how each category of meta-features contributes to the set of all features. Table 10 shows the results of this investigation. The Loss column shows the loss (or gain) in accuracy when one category is removed, as compared to the set of all meta-features (ALL column). As we can see in the #gains and #losses rows, removing one category at a time from the full set of meta-features is not beneficial, especially considering the categories Emoticon and Lexicon-based (losses in 20 and 22 datasets, respectively). In general, all gains achieved by removing the meta-features from some category do not seem to be significant, except for dataset aisopos, whose accuracy increased by up to 1.1% by removing the meta-features from the category Part-of-Speech (ALL−POS column).

6.2.3 Effectiveness of word embedding-based features

In this section, we present the evaluation of the word embedding-based features. We used the ten different pre-trained embedding models summarized in Table 4, aiming at determining the most discriminative one in distinguishing the sentiment expressed in tweets.

Table 10 Accuracies (%) achieved by evaluating different subsets of meta-features

Dataset	ALL	ALL-MICRO		ALL- POS		ALL- SUR		ALL- EMO		ALL- LEX	
		Acc.	Loss	Acc.	Loss	Acc.	Loss	Acc.	Loss	Acc.	Loss
Irony	**81.5**	75.4	− 6.1	76.9	− 4.6	80.0	− 1.5	76.9	− 4.6	66.2	− 15.3
Sarcasm	**80.3**	73.2	− 7.1	76.1	− 4.2	78.9	− 1.4	74.6	− 5.7	53.5	− 26.8
Aisopos	92.8	93.5	+ 0.7	**93.9**	+ 1.1	93.2	+ 0.4	81.7	− 11.1	91.0	− 1.8
SemEval-Fig	**90.3**	89.7	− 0.6	89.4	− 0.9	90.0	− 0.3	90.0	− 0.3	87.9	− 2.4
Sentiment140	**85.0**	84.7	− 0.3	83.0	− 2.0	84.4	− 0.6	84.4	− 0.6	63.8	− 21.2
Person	**83.6**	**83.6**	−	83.1	− 0.5	82.9	− 0.7	83.4	− 0.2	72.4	− 11.2
Hobbit	91.6	91.8	+ 0.2	91.8	+ 0.2	**92.0**	+ 0.4	91.0	− 0.6	75.3	− 16.3
Iphone6	82.5	**83.3**	+ 0.8	83.1	+ 0.6	82.1	− 0.4	81.8	− 0.7	79.1	− 3.4
Movie	**87.0**	86.6	− 0.4	86.5	− 0.5	**87.0**	−	86.6	− 0.4	81.3	− 5.7
Sanders	84.8	**84.9**	+ 0.1	84.2	− 0.6	84.1	− 0.7	83.3	− 1.5	71.5	− 13.3
Narr	90.3	**90.4**	+ 0.1	90.1	− 0.2	90.1	− 0.2	90.1	− 0.2	74.2	− 16.1
Archeage	**85.4**	84.5	− 0.9	84.9	− 0.5	84.5	− 0.9	84.8	− 0.6	80.0	− 5.4
SemEval18	**86.0**	85.6	− 0.4	85.7	− 0.3	85.4	− 0.6	85.4	− 0.6	64.2	− 21.8
OMD	**79.8**	79.2	− 0.6	79.3	− 0.5	78.4	− 1.4	79.3	− 0.5	68.2	− 11.6
HCR	77.5	75.9	− 1.6	**77.7**	+ 0.2	77.1	− 0.4	77.4	− 0.1	73.5	− 4.0
STS-gold	93.1	92.7	− 0.4	92.8	− 0.3	**93.2**	+ 0.1	92.8	− 0.3	71.5	− 21.6
SentiStrength	**83.3**	82.7	− 0.6	82.9	− 0.4	83.0	− 0.3	82.4	− 0.9	66.1	− 17.2
Target-dependent	**83.1**	**83.1**	−	82.8	− 0.3	82.9	− 0.2	82.9	− 0.2	61.6	− 21.5
Vader	93.0	92.7	− 0.3	93.1	+ 0.1	**93.2**	+ 0.2	92.1	− 0.9	74.5	− 18.5
SemEval13	**86.9**	**86.9**	−	**86.9**	−	86.6	− 0.3	86.6	− 0.3	75.1	− 11.8
SemEval17	86.5	86.5	−	86.5	−	86.5	−	**86.6**	+ 0.1	72.8	− 13.7
SemEval16	85.4	85.4	−	85.5	+ 0.1	85.3	− 0.1	**85.7**	+ 0.3	74.3	− 11.1
#gains	−	5		6		4		2		0	
#losses	−	12		14		16		20		22	

We adopted Weka's AffectiveTweets package (Bravo-Marquez et al. 2019) for calculating the features from the pre-trained word embeddings. More precisely, for each dataset, we applied the default configuration of the *TweetToEmbeddingFeatureVector* filter to create a representation for each tweet by aggregating the embedding values of the words. In the default configuration of the filter, the aggregation is done by averaging the word vectors. Also, as a preprocessing step, we replaced URLs with the token "someurl", user mentions with the token "someuser", and removed stopwords.

We evaluate the word embedding representations in two steps. Initially, to determine which classification strategy is the most suitable for this type of feature, we evaluate the predictive performance of SVM, LR, and RF by using the features extracted from each of the ten pre-trained word vectors, one at a time, and for each algorithm. For space reasons, we only report a summary of the results (refer to Online Resource 1 for the detailed evaluation). Then, after determining the best classification strategy, we compare and analyze the predictive power of the features extracted from each pre-trained word vector, to identify the most appropriate one for Twitter sentiment analysis.

Table 11 shows a summary of the results achieved by evaluating each strategy (SVM, LR, and RF columns) on the 22 datasets, and by using as features those calculated from

Table 11 Overview of the results achieved by evaluating SVM, RF, and LR classifiers on the 22 datasets of tweets, and by using as features those calculated from each pre-trained word embedding model

Embedding	SVM #wins	LR #wins	RF #wins	Friedman statistical test	Nemenyi post-hoc test
w2v-GN	5 (41.5)	19 (**26.5**)	1 (64.0)	✓	$\{SVM, LR\} \succ \{RF\}$
GloVe-WP	3 (46.0)	18 (**26.0**)	1 (60.0)	✓	$\{LR\} \succ \{SVM, RF\}$
fastText	0 (48.0)	20 (**24.0**)	2 (60.0)	✓	$\{LR\} \succ \{SVM, RF\}$
EWE	4 (46.0)	17 (**28.0**)	1 (58.0)	✓	$\{LR\} \succ \{SVM, RF\}$
GloVe-TWT	4 (44.5)	20 (**25.5**)	1 (62.0)	✓	$\{SVM\} \succ \{RF\}$ $\{LR\} \succ \{SVM, RF\}$
w2v-Araque	2 (49.0)	18 (**26.5**)	3 (56.5)	✓	$\{SVM\} \succ \{RF\}$ $\{LR\} \succ \{SVM, RF\}$
w2v-Edin	5 (42.5)	19 (**26.5**)	1 (63.0)	✓	$\{SVM\} \succ \{RF\}$ $\{LR\} \succ \{SVM, RF\}$
SSWE	9 (**40.5**)	5 (45.5)	8 (46.0)	✗	Not applicable
Emo2Vec	11 (39.5)	8 (**38.0**)	5 (54.5)	✓	$\{SVM, LR\} \succ \{RF\}$
DeepMoji	5 (42.0)	16 (**28.0**)	1 (62.0)	✓	$\{SVM\} \succ \{RF\}$ $\{LR\} \succ \{SVM, RF\}$

one embedding model at a time (Embedding column). For each scenario assessed, we present the number of wins achieved by each classifier as well as the rank sums, in parenthesis. We also show whether the differences between the results are statistically significant (Friedman and Nemenyi post-hoc test columns).

From Table 11, we can observe that LR presented the best results in nine out of the ten pre-trained models tested, while SVM performed slightly better merely by using SSWE embeddings. Moreover, LR outperformed SVM with a statistical difference between them in seven out of the nine wins. Conversely, RF did not achieve meaningful results, with the lowest number of wins. Based on this evaluation, we chose LR as the most suitable classifier for the embedding-based features.

Next, we analyze the performance of an LR classifier fed with the embedding-based features from each pre-trained model, so as to identify which one is better-suited for the tweet polarity detection. The results are presented in Table 12. The number of dimensions immediately below each embedding name refers to the number of features calculated from each pre-trained model.

As we can see in Table 12, the w2v-Edin model achieved the best performance in eight out of the 22 datasets and was ranked first in the overall evaluation (rank position row). Although this model did not leverage any sentiment information during its construction, as enlightened by its authors in Bravo-Marquez et al. (2016), its training parameters were optimized for the emotion detection task on tweets, which may have benefited the sentiment classification of tweets. The fastText model had the second best results, followed by GloVe-TWT, DeepMoji, and w2v-GN, respectively.

Among the affective embeddings, the SSWE model featured the worse performance, which is in agreement with other works (Agarwal et al. 2018; Felbo et al. 2017; Xu et al. 2018). Surprisingly, the generic embeddings w2v-Edin, fastText, and GloVe-TWT outperformed all the affective embeddings (DeepMoji, Emo2Vec, SSWE, and EWE). Although unexpected, Agarwal et al. (2018) have reported similar results. One possible reason is the number of words embedded in the models, i.e., the vocabulary size of each pre-trained

Table 12 Comparison among the results achieved with each pre-trained embedding model by using an LR classifier

Dataset	w2v-GN 300d	GloVe-WP 300d	fastText 300d	EWE 300d	GloVe-TWT 200d
Irony	70.8	**76.9**	73.8	69.2	66.2
Sarcasm	67.6	63.4	64.8	69.0	69.0
Aisopos	90.6	86.3	76.6	73.7	76.6
SemEval-Fig	88.2	86.3	88.2	86.9	87.2
Sentiment140	84.1	85.5	82.5	83.8	83.6
Person	81.3	80.4	83.1	**84.1**	83.1
Hobbit	91.0	90.4	91.0	92.5	90.0
Iphone6	78.8	77.8	81.2	78.4	**82.1**
Movie	87.9	87.3	88.2	87.2	86.6
Sanders	80.6	77.4	80.1	79.0	80.6
Narr	88.0	84.9	86.1	84.5	88.6
Archeage	83.1	82.5	83.9	83.5	85.2
SemEval18	79.0	79.2	81.2	79.5	81.4
OMD	81.2	81.9	80.1	78.6	77.2
HCR	78.8	76.0	77.3	76.9	**79.0**
STS-gold	84.6	83.3	85.3	85.1	85.3
SentiStrength	77.0	75.9	78.2	77.9	78.0
Target-dependent	81.9	80.5	82.5	82.6	**83.1**
Vader	87.7	87.1	88.5	88.0	87.4
SemEval13	83.1	81.8	83.2	82.5	83.1
SemEval17	86.4	86.6	**88.5**	87.2	87.7
SemEval16	84.8	85.2	86.2	85.6	**86.4**
#wins	0	1	1	1	4
Rank sums	122.0	152.5	97.0	129.5	105.5
Rank position	5	9	2	7	3
Dataset	w2v-Araque 500d	w2v-Edin 400d	SSWE 50d	Emo2Vec 100d	DeepMoji 256d
Irony	69.2	75.4	73.8	**76.9**	73.8
Sarcasm	70.4	56.3	**73.2**	62.0	59.2
Aisopos	74.8	92.8	91.7	79.9	**94.6**
SemEval-Fig	86.0	89.1	87.5	87.5	**89.4**
Sentiment140	80.2	**87.7**	84.1	84.7	80.8
Person	78.6	81.3	78.6	79.3	80.4
Hobbit	92.3	92.5	83.1	88.7	**92.7**
Iphone6	78.4	81.6	74.8	78.8	79.7
Movie	87.0	88.6	88.4	**89.3**	86.5
Sanders	77.9	**82.9**	77.8	79.1	82.1
Narr	85.3	**89.6**	89.5	88.6	89.1
Archeage	83.2	**87.0**	79.5	81.9	83.5
SemEval18	75.3	**82.8**	80.8	80.4	80.0
OMD	77.1	**83.3**	77.2	76.4	75.9
HCR	74.1	78.5	73.6	75.3	75.4

Table 12 (continued)

Dataset	w2v-Araque 500d	w2v-Edin 400d	SSWE 50d	Emo2Vec 100d	DeepMoji 256d
STS-gold	86.2	87.5	**87.8**	85.8	**87.8**
SentiStrength	76.8	81.2	79.2	**85.4**	79.9
Target-dependent	81.5	82.5	77.5	81.3	82.0
Vader	86.7	**89.3**	87.7	87.1	88.6
SemEval13	81.0	83.6	83.2	**88.7**	83.4
SemEval17	83.1	87.6	80.8	85.3	84.9
SemEval16	82.7	**86.4**	81.5	84.5	84.3
#wins	0	**8**	2	4	4
Rank sums	175.5	**53.0**	138.0	126.5	109.0
Rank position	10	**1**	8	6	4

{fastText, GloVe-TWT, Deepmoji} $\succ$ {w2v-Araque}

{w2v-Edin} $\succ$ {w2v-GN, GloVe-WP, EWE, w2v-Araque, SSWE, Emo2Vec}

Table 13 Coverage analysis (%) of the pre-trained word vectors vocabulary for the five best ranked embeddings

Dataset	w2v-GN $\lvert V\rvert = 3$M	fastText $\lvert V\rvert = 1$M	GloVe-TWT $\lvert V\rvert = 1.2$M	w2v-Edin $\lvert V\rvert = 259$ K	DeepMoji $\lvert V\rvert = 50$ K
Irony	71.33 (5.0)	78.05 (3.0)	78.23 (2.0)	**82.48** (1.0)	75.40 (4.0)
Sarcasm	72.27 (5.0)	76.76 (2.0)	75.98 (3.0)	**81.64** (1.0)	74.22 (4.0)
Aisopos	71.12 (5.0)	76.31 (3.0)	77.81 (2.0)	**82.67** (1.0)	75.02 (4.0)
SemEval-Fig	70.31 (5.0)	74.24 (4.0)	74.40 (2.0)	**81.17** (1.0)	74.34 (3.0)
Sentiment140	75.69 (5.0)	80.80 (2.0)	80.45 (3.0)	**86.49** (1.0)	76.92 (4.0)
Person	74.81 (5.0)	81.53 (2.0)	80.66 (3.0)	**86.65** (1.0)	77.51 (4.0)
Hobbit	67.33 (5.0)	74.29 (2.0)	73.29 (3.0)	**77.27** (1.0)	69.25 (4.0)
Iphone6	63.88 (5.0)	66.29 (3.0)	67.04 (2.0)	**73.27** (1.0)	65.16 (4.0)
Movie	80.52 (5.0)	84.25 (3.0)	85.26 (2.0)	**91.59** (1.0)	82.36 (4.0)
Sanders	61.77 (5.0)	66.01 (2.0)	65.99 (3.0)	**75.04** (1.0)	62.18 (4.0)
Narr	72.76 (5.0)	79.60 (3.0)	81.76 (2.0)	**88.43** (1.0)	78.73 (4.0)
Archeage	61.71 (5.0)	70.45 (2.0)	69.12 (3.0)	**74.51** (1.0)	63.41 (4.0)
SemEval18	51.99 (5.0)	60.76 (3.0)	61.74 (2.0)	**68.15** (1.0)	59.24 (4.0)
OMD	72.15 (5.0)	85.04 (2.0)	82.84 (3.0)	**86.95** (1.0)	75.94 (4.0)
HCR	52.13 (5.0)	63.82 (2.0)	62.19 (3.0)	**70.26** (1.0)	55.47 (4.0)
STS-gold	63.64 (5.0)	73.36 (3.0)	73.82 (2.0)	**79.53** (1.0)	69.30 (4.0)
SentiStrength	54.31 (5.0)	64.04 (3.0)	66.01 (2.0)	**71.81** (1.0)	60.50 (4.0)
Target-dependent	65.57 (5.0)	79.81 (3.0)	82.98 (2.0)	**84.75** (1.0)	73.85 (4.0)
Vader	66.79 (5.0)	82.07 (3.0)	83.26 (2.0)	**88.93** (1.0)	75.32 (4.0)
SemEval13	**80.60** (1.0)	62.01 (4.0)	65.58 (3.0)	70.81 (2.0)	57.70 (5.0)
SemEval17	38.13 (5.0)	50.23 (2.0)	49.80 (3.0)	**54.19** (1.0)	42.85 (4.0)
SemEval16	38.67 (5.0)	51.92 (3.0)	53.23 (2.0)	**57.40** (1.0)	45.52 (4.0)
Rank sums	106.0	59.0	54.0	**23.0**	88.0

word vector, as shown in Table 4 ($|V|$ column). Indeed, the vocabulary sizes of the fastText and GloVe-TWT embeddings (1M and 1.2M, respectively) are much larger than the DeepMoji, EWE, and SSWE ones (50K, 183K, and 137K, respectively). Although the number of words embedded in the Emo2Vec model is as large as in the GloVe-TWT one (1.2M), Emo2Vec may have performed poorly considering that it is weak on capturing the syntactic and semantic meaning of words, as reported in Agarwal et al. (2018).

Table 13 presents a coverage analysis of the pre-trained models for the five best-ranked embeddings (w2v-Edin, fastText, GloVe-TWT, DeepMoji, and w2v-GN, respectively). More specifically, for each dataset, we show the fraction of words in the dataset that appear in a given pre-trained model. The information below each model name refers to their vocabulary size. We also show, in parenthesis, the rank assigned for each model. We can observe that the w2v-Edin model, which achieved the best overall accuracies, has the highest coverage for all datasets, except for Semeval13. Also, fastText and GloVe-TWT, whose vocabulary sizes are much larger than DeepMoji's, have the second and third highest coverage, followed by DeepMoji. The w2v-GN model presents the lowest coverage, even though it has the largest vocabulary size (3M). Since this model was trained on a corpus of Google news articles, its predictive power might not have generalized well to short, noisy texts, as tweets.

Lastly, the Friedman test detected a significant difference between the results. The Nemenyi test showed that the accuracies achieved by the w2v-Edin embeddings are significantly better than those by w2v-GN, GloVe-WP, EWE, w2v-Araque, SSWE, and Emo2Vec. Furthermore, fastText, GloVe-TWT, and DeepMoji results are significantly better than those of w2v-Araque's, which presented the worst overall performance.

6.2.4 Overall analysis of features

In previous Sects. 6.2.1, 6.2.2, and 6.2.3, we have identified the best classifiers for each group of features proposed in state-of-the-art research on Twitter sentiment analysis, i.e., *n*-grams, meta-features, and word embedding-based features. Here, we present an overall analysis of these different feature sets. More specifically, we aim at effectively responding to research question RQ1 ("*Which group of features is the most effective in Twitter sentiment analysis?*"), by performing a comparison between the following classifiers: RF with meta-features, SVM with *n*-grams, and LR with embedding-based features from the w2v-Edin model.

Table 14 presents the comparison made between the aforementioned classifiers (meta-features, *n*-grams, and w2v-Edin columns). We can see that the RF classifier fed with meta-features achieved the highest accuracies in 13 out of the 22 datasets, followed by the embedding-based representation provided by w2v-Edin word vectors. In general, the *n*-gram features achieved worse predictive performance. The Friedman test followed by the Nemenyi post-hoc test detected a significant difference between the results achieved by the meta-features and n-grams classifiers. The Nemenyi test evidenced that the accuracies achieved by the meta-features classifier are significantly better than those presented by the *n*-grams approach.

Note that the *n*-gram features outperformed meta-features and w2v-Edin only for datasets SemEval-Fig, hobbit, and HCR. The tweets from SemEval-Fig and HCR are considered to belong to challenging domains, namely metaphorical language and health campaigns, respectively. For that reason, the *n*-grams may have succeeded in capturing more context from the specific language used in these datasets. Indeed, by analyzing the most

Table 14 Comparison among the best classifiers for each group of features

Dataset	Meta-features RF	*n*-grams SVM	w2v-Edin LR
Irony	**81.5**	66.2	75.4
Sarcasm	**80.3**	50.7	56.3
Aisopos	**92.8**	87.8	**92.8**
SemEval-Fig	90.3	**91.0**	89.1
Sentiment140	85.0	84.1	**87.7**
Person	**83.6**	79.0	81.3
Hobbit	91.6	**92.9**	92.5
Iphone6	**82.5**	77.6	81.6
Movie	87.0	84.1	**88.6**
Sanders	**84.8**	83.0	82.9
Narr	**90.3**	83.7	89.6
Archeage	85.4	86.3	**87.0**
SemEval18	**86.0**	80.2	82.8
OMD	79.8	81.2	**83.3**
HCR	77.5	**79.1**	78.5
STS-gold	**93.1**	84.0	87.5
SentiStrength	**83.3**	73.2	81.2
Target-dependent	**83.1**	81.4	82.5
Vader	**93.0**	84.8	89.3
SemEval13	**86.9**	81.0	83.6
SemEval17	86.5	86.9	**87.6**
SemEval16	85.4	85.8	**86.4**
#wins	**13**	3	7
Rank sums	**37.5**	55.0	39.5

$\{\text{Meta-features}\} \succ \{n\text{-grams}\}$

relevant amongst all features for dataset HCR, we observed that the unigram "#tcot", which means "*top conservatives on Twitter*", appears at the top of the ranking as the most important feature. Since this term is very context-sensitive, the *n*-gram classifier may have benefited from this particular information.

Regarding meta-features, we can observe that applying them on tweets from datasets irony and sarcasm led to a significant gain in accuracy as compared to *n*-grams and embedding-based features. As previously mentioned, ironic and sarcastic tweets usually contain signals, such as punctuation marks, that may help determine the sentiment expressed through them.

It is worth mentioning that the number of meta-features is much smaller than the number of *n*-grams. As shown in Table 7 (#features column), the number of *n*-grams varies from 1.8 to 252.1 K (datasets irony and SemEval16, respectively), while an increased predictive performance was achieved by using only a small set of 130 meta-features. Similarly, the number of meta-features is smaller than the number of the features extracted from the w2v-Edin pre-trained model, i.e., 400 features, as shown in Table 4 ($|D|$ column).

Another advantage of meta-features over word embedding representations is the fact that meta-features can be easily interpreted. For example, by applying relevance measures, such as IG, to determine the most predictive meta-features, we are able to determine the

type of information that may be useful in distinguishing the positive tweets from the negative ones, given some specific domain. On the other hand, the features calculated from pre-trained embedding models, i.e., real values corresponding to distinct aspects of words, are complex to explain.

Also, unlike meta-features, pre-trained embedding models are language-dependent. In general, the word vectors are trained on huge text corpora containing documents from the same language. Otherwise, it is not possible to capture semantic and syntactic relationships between words. Meta-features, in turn, can be employed regardless of language limitations, with the exception of lexicon-based meta-features, which rely on sentiment lexicons from specific languages. Nevertheless, it is possible to use lexicons generated for any language, if available. With that being said, it is not necessarily a limitation of meta-features. In fact, Sousa et al. (2018) successfully used a subset of meta-features, which we have examined and categorized in our previous work (Carvalho and Plastino 2016), to identify relevant tweets in preventing mosquito-borne diseases, such as the Zika virus, in tweets in Portuguese. Therefore, meta-features are not only language-independent but can also be easily employed in cross-domain problems.

6.3 Answering research questions RQ2 and RQ3

In this section, we explore the combinations of the feature sets evaluated in the previous sections. Specifically, we address research questions RQ2 and RQ3, as follows:

RQ2. Can the concatenation of different types of features proposed in the literature boost classification performance in Twitter sentiment analysis?

After evaluating the individual performance of each feature set (Sect. 6.2), we examine how they complement one another in the polarity detection task on Twitter by using a simple feature concatenation approach. We address this question in Sect. 6.3.1.

RQ3. Can the sentiment classification of tweets benefit from the use of ensemble classification strategies having the best classifiers for each type of feature as base learners?

In Sect. 6.3.2, we use and evaluate the best individual classifiers as base learners of two distinct ensemble learning strategies, one of which being majority vote, doing so by averaging the probability distributions of base learners, and also via stacking (Wolpert 1992), which is a meta-learning technique that uses the probability distributions of base learners as meta-features for a new learning problem.

6.3.1 Combining features through feature concatenation (RQ2)

Here, we present the results achieved by combining *n*-grams, meta-features, and embedding-based features through feature concatenation, i.e., by concatenating each feature set into a unique feature vector. We evaluated each possible combination of feature sets (*n*-grams, meta-features, and word embeddings) with one classification algorithm at a time (SVM, RF, and LR). For space constraints, we only show the best results for each combined feature vector, as shown in Table 15 (feature concatenation column).

The best results when combining *n*-grams with any other feature set, that is, meta-features or embedding-based features, were achieved by using SVM (fifth and seventh columns). This may be due to the higher number of *n*-grams, since SVM performed better under the individual evaluation of the *n*-gram features, as seen in Table 7. Regarding the combination of meta-features with word embedding features (sixth column), LR outperformed SVM and RF. Finally, the combination of all feature sets into one unique feature

Table 15 Accuracies (%) achieved by combining different feature sets through feature concatenation

Dataset	Meta-features	*n*-grams	w2v-Edin	Feature concatenation			
	■	□	●	■ + □	■ + ●	□ + ●	■ + □ + ●
	RF	SVM	LR	SVM	LR	SVM	LR
Irony	**81.5**	66.2	75.4	80.0	**81.5**	73.8	75.4
Sarcasm	**80.3**	50.7	56.3	76.1	73.2	56.3	70.4
Aisopos	92.8	87.8	92.8	92.8	93.5	93.5	**94.6**
SemEval-Fig	90.3	91.0	89.1	91.9	90.7	91.3	**92.2**
Sentiment140	85.0	84.1	87.7	**90.0**	89.1	88.0	89.7
Person	83.6	79.0	81.3	85.0	85.4	82.7	**86.1**
Hobbit	91.6	92.9	92.5	**93.3**	90.8	92.9	93.1
Iphone6	82.5	77.6	81.6	81.8	82.5	81.6	**83.5**
Movie	87.0	84.1	88.6	88.2	**89.1**	87.7	88.2
Sanders	84.8	83.0	82.9	87.3	85.0	86.3	**88.1**
Narr	90.3	83.7	89.6	90.1	90.5	88.5	**90.6**
Archeage	85.4	86.3	87.0	89.2	87.8	89.1	**90.3**
SemEval18	86.0	80.2	82.8	**86.7**	86.5	84.8	86.0
OMD	79.8	81.2	83.3	85.0	84.1	84.3	**86.0**
HCR	77.5	79.1	78.5	80.9	79.0	80.5	**81.7**
STS-gold	**93.1**	84.0	87.5	92.3	91.7	89.0	92.2
SentiStrength	83.3	73.2	81.2	83.2	83.1	81.7	**84.2**
Target-dependent	83.1	81.4	82.5	84.9	84.8	83.4	**85.5**
Vader	93.0	84.8	89.3	93.2	93.6	89.8	**93.9**
SemEval13	86.9	81.0	83.6	88.4	87.3	85.2	**88.6**
SemEval17	86.5	86.9	87.6	90.0	89.9	89.8	**90.8**

Table 15 continued

Dataset	Meta-features	*n*-grams	w2v-Edin	Feature concatenation			
	■	□	●	■ + □	■ + ●	□ + ●	■ + □ + ●
	RF	SVM	LR	SVM	LR	SVM	LR
SemEval16	85.4	85.8	86.4	88.6	87.5	88.3	**88.9**
#wins	3	0	0	3	2	0	**15**
Rank sums	100.0	139.5	119.0	53.0	71.5	94.0	**37.5**

$\{\text{meta-features} + n\text{-grams} + \text{w2v-Edin}\} \succ \{\text{meta-features}, n\text{-grams}, \text{w2v-Edin}, n\text{-grams} + \text{w2v-Edin}\}$

$\{\text{meta-features} + n\text{-grams}\} \succ \{\text{meta-features}, n\text{-grams}, \text{w2v-Edin}\}$

$\{\text{meta-features} + \text{w2v-Edin}\} \succ \{n\text{-grams}, \text{w2v-Edin}\}$

$\{n\text{-grams} + \text{w2v-Edin}\} \succ \{n\text{-grams}\}$

vector was most benefited by using LR (last column). Indeed, the LR algorithm achieved comparable performance to SVM for the *n*-gram features (Table 7) and the second best results in the meta-features evaluation (Table 8). In both assessments, LR was ranked as the second best classifier. Therefore, combining all feature vectors may have caused LR to excel on SVM and RF.

We can observe that the sentiment classification of tweets benefits from the concatenation of all feature sets, i.e., meta-features + *n*-grams + w2v-Edin (last column), achieving the best overall results in 15 out of the 22 datasets. The second best results were achieved by meta-features + *n*-grams (fifth column), followed by meta-features + w2v-Edin (sixth column). The least accurate results were yielded by *n*-grams + w2v-Edin (seventh column).

Interestingly, concerning the combinations of pairs of feature sets (fifth, sixth and seventh columns), only the concatenation provided by meta-features + *n*-grams performed statistically better than all individual classifiers (MF, *n*-grams, and w2v-Edin columns). This may be evidence that combining meta-features with *n*-grams is beneficial in detecting the sentiment expressed in tweets.

It is also noteworthy that the combination of all feature sets is significantly better than all individual classifiers and also *n*-grams + w2v-Edin. On the other hand, the results achieved by concatenating all feature sets are not statistically more significant than the meta-features + *n*-grams and the meta-features + w2v-Edin classifiers. Moreover, the meta-features classifier (MF column) achieved an overall performance comparable to that of *n*-grams + w2v-Edin, as we can see in the rank sums (overall) row (100.0 and 94.0, respectively). These results emphasize the predictive power of meta-features and their importance in the context of Twitter sentiment analysis.

6.3.2 Combining features via ensemble learning (RQ3)

In this section, we present the predictive performance achieved by combining all individual classifiers as base learners of two distinct ensemble strategies. More precisely, we use the best individual classifiers for each type of feature, i.e., RF with meta-features, SVM with *n*-grams, and LR with embedding-based features from the w2v-Edin model, as base learners of two ensemble classification strategies, namely majority vote and stacking (Wolpert 1992).

An ensemble of classifiers combines the decisions of a set of classifiers according to some rule. In the majority voting ensemble, we used the average of probabilities combination rule, which averages the posterior probability distributions for each class value. Thus, the class value with the highest averaged probability is chosen as the final prediction. Stacking, or stacked generalization (Wolpert 1992), is an ensemble technique that uses the predictions made by the base learners as inputs for a meta-learning task. First, the base classifiers, also referred to as level-0 models, are trained on the original feature space, or level-0 data, and their predictions are used as new data (level-1 data) for another learning problem. Next, in the second stage, a meta-learning algorithm, or level-1 generalizer, is trained on the level-1 data to solve this new learning problem (Ting and Witten 1999). In this work, we used LR as the level-1 generalizer.

Table 16 summarizes the results. As we can see, both ensemble strategies (ensemble column) effectively outperformed all individual classifiers, except for datasets irony and sarcasm. This may be due to the poor performance of the *n*-gram features on both datasets (66.2% and 50.7%, respectively). For dataset sarcasm, not only the *n*-grams performed poorly, but also the embedding-based features (56.3%). In general, the stacking technique

(stacking column) achieved the best results in 13 out of the 22 datasets. Notwithstanding, the ensemble by the average of probabilities rule (avg. prob. column) achieved a performance comparable to that of stacking. Regarding the Friedman and Nemenyi tests, both ensemble strategies are statistically better than all individual classifiers, though there is no significant difference between them.

As stated by Dietterich (2000), for the predictive performance of an ensemble of classifiers to be better than its base learners, they must be accurate and diverse, meaning that they should make good but different decisions. In this context, we present an analysis of the correlation between the predictions made by each classifier comprising the ensembles. Precisely, we computed the Pearson correlation coefficient between the outputs (predictions) of each pair of classifiers. The Pearson coefficient ranges from -1 to $+1$, where a value less (greater) than zero indicates a negative (positive) association between the outputs. In that case, for any pair of classifiers, the closer to zero the Pearson coefficient, the more different the decisions made by them. Table 17 shows the Pearson correlation matrices for selected datasets considering the predictions made by each classifier.

We can note that, in general, the predictions made by the base classifiers are sufficiently uncorrelated, leading to improved predictive performance of the ensemble strategies for most datasets. For example, analyzing the correlation matrix for dataset HCR, we see that the correlations between the predictions of each pair of classifiers are sufficiently low. Besides, as shown in Table 16, each classifier has achieved competitive accuracies for this dataset (77.5, 79.1, and 78.5%). As a result, the predictive performance achieved by ensembling them (i.e., 81.5%) effectively surpassed the best individual classifier by up to 2.4%. Similarly, for dataset OMD, we can observe that the low correlations between the predictions made by the base learners, along with their fair accuracies (79.8, 81.2, and 83.3%), may lead to improved ensemble performance, i.e., 86.5% for the average probability ensemble (avg. prob.), and 85.9% for stacking. As compared to the best base model (83.3%), this represents a gain in accuracy of up to 3.2 and 2.6%, respectively. We can see an analogous effect on datasets person, iphone6, SemEval18, and SemEval13.

Interestingly, for dataset STS-gold, it is evident that although the correlation coefficients between meta-features and *n*-grams, and between *n*-grams and w2v-Edin base classifiers are moderately low (0.6193 and 0.5638, respectively), the *n*-gram classifier does not seem to be as accurate as the meta-features one. More specifically, while the meta-features classifier achieved a classification accuracy of 93.1%, the *n*-gram classifier achieved 84.0% only. It is possible that, for this reason, the ensemble strategies did not achieve meaningful results for dataset STS-gold. As can be observed in Table 16, the accuracy achieved by the best ensemble classifier is 93.2% (stacking), which represents a gain of only 0.1% over the best base classifier (meta-features).

For dataset hobbit, even though all individual classifiers have achieved very high and competitive accuracies (91.6, 92.9, and 92.5%), the correlation coefficients between any pair of classifiers are greater than 0.8, which means that their predictions are very similar to one another. Hence, there is no sufficient diversity among the base classifiers in the ensembles. This may have caused the ensemble strategies to achieve rather comparable performances to the best individual base learner, i.e., 93.1% (avg. prob.) and 92.7% (stacking), against 92.9% (*n*-gram classifier), respectively.

In order to illustrate how diversity is relevant when choosing the base learners of an ensemble model, we show that the predictive performance of the ensemble can be improved if we select different base classifiers by leveraging the Pearson coefficients between their predictions. For example, regarding dataset hobbit, Table 18 shows that the predictive performance of the ensemble is improved by up to 1.0%, by switching from the

w2v-Edin classifier to the fastText one. Indeed, as shown in Table 19, analyzing the correlation coefficients among the base classifiers of this new ensemble model (0.8580, 0.7464, and 0.7661), we note that they are lower than the coefficients of the original base models, i.e., the meta-features, *n*-grams, and w2v-Edin classifiers (0.8580, 0.8127, and 0.8796). However, though the fastText classifier is less accurate than the w2v-Edin one, it is still a good choice as compared to a simple classifier that classifies each instance into the majority class (majority class column).

The result of the previous experiment gives us evidence that selecting the most accurate classifiers as members of an ensemble model does not ensure higher predictive performances. To verify this hypothesis, we performed an experiment to test whether the predictive performance of an ensemble improves when replacing the least accurate base classifier for a more accurate counterpart. The result is presented in Table 20.

As we can observe in Table 20, regarding dataset SemEval18, switching the least accurate classifier for this dataset (the *n*-gram classifier) to the fastText one, which is the second best embedding-based classifier, the classification accuracy of the stacking ensemble drops from 87.3 to 86.6%. Analyzing the correlation coefficients across the base classifiers of this new ensemble, as shown in Table 21, we can see that their predictions are much more correlated than the predictions of the base classifiers from the original ensemble (i.e., 0.5725, 0.6847, 0.5371). Lastly, although the *n*-gram classifier is less accurate than the fastText one, it can still be regarded as a good and accurate classifier as compared to the majority class one (majority class column).

6.3.3 Comparing combination methods

In this section, we perform a comparison between the combination methods exploited in this study, such as feature concatenation and ensemble of classifiers.

Table 22 presents the comparison between the feature concatenation and stacking ensemble methods, both of which performed better than the avg. prob. ensemble. We can see that the ensemble learning approach outperformed the concatenation of feature vectors with an LR classifier in 12 out of the 22 datasets. Nevertheless, the feature concatenation technique achieved a comparable performance to the stacking ensemble.

Disregarding datasets irony and sarcasm, for which the best performances were achieved by using the RF with meta-features classifier (classification accuracies of 81.5 and 80.3%, respectively), it appears as though smaller datasets, such as aisopos, SemEval-Fig, sentiment140, person, and hobbit, have benefited from the feature concatenation approach. On the other hand, larger datasets, such as STS-gold, Target-dependent, Vader, SemEval13, SemEval17, and SemEval16, achieved higher predictive performances by using the ensemble learning technique.

With regards to the differences between the results achieved by the combination methods, the Friedman test did not attest any significant statistical difference.

7 Conclusions and future work

In this article, we presented a thoughtful evaluation of the distinct types of features employed in state-of-the-art research on Twitter sentiment analysis. The rich feature space exploited in this work includes features extracted from the basic *n*-gram language model to more sophisticated features such as meta-features and word embeddings. Besides the

Table 16 Accuracies (%) achieved by combining different feature sets as base classifiers of an ensemble strategy

Dataset	Meta-features	*n*-grams	w2v-Edin	Ensemble	
				Avg. prob.	Stacking
	RF	SVM	LR		LR
Irony	**81.5**	66.2	75.4	75.4	76.9
Sarcasm	**80.3**	50.7	56.3	73.2	78.9
Aisopos	92.8	87.8	92.8	**93.5**	93.2
SemEval-Fig	90.3	91.0	89.1	**91.9**	**91.9**
Sentiment140	85.0	84.1	87.7	**90.8**	89.4
Person	83.6	79.0	81.3	84.3	**85.4**
Hobbit	91.6	92.9	92.5	**93.1**	92.7
Iphone6	82.5	77.6	81.6	83.5	**86.1**
Movie	87.0	84.1	88.6	87.5	**89.8**
Sanders	84.8	83.0	82.9	86.8	**87.2**
Narr	90.3	83.7	89.6	90.5	**91.0**
Archeage	85.4	86.3	87.0	**90.0**	89.7
SemEval18	86.0	80.2	82.8	**87.4**	87.3
OMD	79.8	81.2	83.3	**86.5**	85.9
HCR	77.5	79.1	78.5	**81.5**	**81.5**
STS-gold	93.1	84.0	87.5	91.9	**93.2**
SentiStrength	83.3	73.2	81.2	83.5	**84.2**
Target-dependent	83.1	81.4	82.5	**85.7**	**85.7**
Vader	93.0	84.8	89.3	93.2	**94.2**
SemEval13	86.9	81.0	83.6	87.7	**88.7**
SemEval17	86.5	86.9	87.6	**91.1**	91.0
SemEval16	85.4	85.8	86.4	88.5	**89.1**
#wins	2	0	0	10	**13**
Rank sums	76.5	98.0	82.0	40.0	**33.5**

$\{\text{ensemble} - \text{avg. prob.}\} \succ \{\text{meta-features}, n\text{-grams}, \text{w2v-Edin}\}$

$\{\text{ensemble} - \text{stacking}\} \succ \{\text{meta-features}, n\text{-grams}, \text{w2v-Edin}\}$

individual evaluation of each feature set, we also investigated the effect of combining them through feature concatenation and via ensemble learning strategies, considering that features from different sets can complement one another.

The meta-features examined in this work were collected from a wide range of studies in the literature of Twitter sentiment analysis. Although these studies have proposed different meta-features, we filled the existing gap of aggregating and evaluating the predictive power of those meta-features proposed in the literature over the years. Further, as an extension of our previous study (Carvalho and Plastino 2016), we categorized this rich set of meta-features to examine the effectiveness of different types of meta-features in the discernment of positive tweets from negative ones. Moreover, regarding the vast number of publicly available pre-trained word embeddings, we conducted experiments to identify the most suitable ones for detecting the sentiment expressed in tweets.

Based on the experimental results of this study, we can draw the following conclusions:

Table 17 Pearson correlation matrices regarding the predictions made on distinct datasets by using the meta-features (RF), *n*-grams (SVM), and w2v-Edin (LR) classifiers

	Person		Hobbit		iphone6	
	n-grams	w2v-Edin	*n*-grams	w2v-Edin	*n*-grams	w2v-Edin
Meta-features	0.3817	0.5815	0.8580	0.8127	0.4577	0.5926
n-grams	–	0.5233	–	0.8796	–	0.4690
	Sanders		SemEval18		OMD	
	n-grams	w2v-Edin	*n*-grams	w2v-Edin	*n*-grams	w2v-Edin
Meta-features	0.5967	0.6614	0.5725	0.6847	0.4479	0.5506
n-grams	–	0.6118	–	0.5371	–	0.6204
	HCR		STS-gold		SemEval13	
	n-grams	w2v-Edin	*n*-grams	w2v-Edin	*n*-grams	w2v-Edin
Meta-features	0.4235	0.4065	0.6193	0.7404	0.4360	0.5951
n-grams	–	0.4650	–	0.5638	–	0.3978

- Regarding the categories of meta-features, we could observe that the features from the Lexicon-based category presented themselves to be the most relevant ones. In this work, we employed seven different lexicons and word lists. Since each lexicon comprises different words, we believe that they can effectively complement one another in representing the tweets. Nevertheless, we encourage the use of the set of all meta-features, as they are able to achieve even further improved results.
- For each feature set studied in this work, we could see that an appropriate choice of a supervised learning algorithm can boost classification effectiveness on a large collection of 22 datasets of tweets. Specifically, for most situations, we showed that *n*-grams, meta-features, and embedding-based features could achieve significantly better results when fed to SVM, RF, and LR, respectively.
- We showed that the rich set of meta-features exploited in this study outperformed *n*-grams and word embedding-based features. Nevertheless, the sentiment classification of tweets benefits from the combination of all feature sets through feature concatenation. Despite that, we could see that only the combination provided by meta-features + *n*-grams performed statistically better than all individual classifiers. For that reason, we believe that meta-features and *n*-grams can effectively complement each other in the sentiment classification of tweets.
- Finally, we showed that the combination of *n*-grams, meta-features, and word embedding-based features via an ensemble technique can achieve overall best performances than a simple feature concatenation approach. Furthermore, we could see that the classification effectiveness of an ensemble of classifiers can be improved provided that the diversity among them is leveraged.

For future work, we plan to investigate more specific types of embedding models, such as Tweet2Vec (Vosoughi et al. 2016), which is a method for generating a general-purpose representation of tweets, using a character-level neural architecture. We also intend to reproduce the experiments conducted in this work on different types of text data, such as Facebook and YouTube comments, in order to attest whether the results would carry over to those domains.

Table 18 Accuracies (%) achieved by combining different feature sets as base classifiers of an ensemble strategy

Majority class	Meta-features	*n*-grams	w2v-Edin	fastText	Stacking	
	■	□	●	○	■□●	■□○
	RF	SVM	LR	LR	LR	LR
67.8	91.6	92.9	92.5	91.0	92.7	**93.7**

Table 19 Pearson correlation matrix regarding the predictions made on the dataset hobbit by using the meta-features (RF), *n*-grams (SVM), and fastText (LR) classifiers

	Meta-features	*n*-grams	fastText
Meta-features	–	0.8580	0.7464
n-grams	–	–	0.7661
fastText	–	–	–

Table 20 Accuracies (%) achieved by combining different feature sets as base classifiers of an ensemble strategy

Majority class	Meta-features	*n*-grams	w2v-Edin	FastText	Stacking	
	■	□	●	○	■□●	■○
	RF	SVM	LR	LR	LR	LR
53.5	86.0	80.2	82.8	81.2	**87.3**	86.6

Table 21 Pearson Correlation matrix regarding the predictions made on the dataset SemEval18 by using the meta-features (RF), w2v-Edin (LR), and fastText (LR) classifiers

	Meta-features	w2v-Edin	fastText
Meta-features	–	0.6847	0.6397
w2v-Edin	–	–	0.6796
fastText	–	–	–

Table 22 Comparison among the results achieved by evaluating distinct methods of feature combination (feature concatenation and ensemble learning)

Dataset	Feature concatenation	Ensemble learning
	Meta-features + *n*-grams + w2v-Edin	Meta-features (RF) *n*-grams (SVM) w2v-Edin (LR)
	LR	LR (stacking)
irony	75.4	**76.9**
sarcasm	70.4	**78.9**
aisopos	**94.6**	93.2
SemEval-Fig	**92.2**	91.9
sentiment140	**89.7**	89.4
person	**86.1**	85.4
hobbit	**93.1**	92.7
iphone6	83.5	**86.1**
movie	88.2	**89.8**
sanders	**88.1**	87.2
Narr	90.6	**91.0**
archeage	**90.3**	89.7
SemEval18	86.0	**87.3**
OMD	**86.0**	85.9
HCR	**81.7**	81.5
STS-gold	92.2	**93.2**
SentiStrength	84.2	84.2
Target-dependent	85.5	**85.7**
Vader	93.9	**94.2**
SemEval13	88.6	**88.7**
SemEval17	90.8	**91.0**
SemEval16	88.9	**89.1**
#wins	9	**12**

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Agarwal A, Xie B, Vovsha I, Rambow O, Passonneau R (2011) Sentiment analysis of Twitter data. In: Proceedings of the workshop on languages in social media. Association for Computational Linguistics, pp 30–38

Agrawal A, An A, Papagelis M (2018) Learning emotion-enriched word representations. In: Proceedings of the 27th international conference on computational linguistics, pp 950–961

Akbik A, Bergmann T, Blythe D, Rasul K, Schweter S, Vollgraf R (2019) FLAIR: an easy-to-use framework for state-of-the-art NLP. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations). Association for Computational Linguistics, Minneapolis, Minnesota, pp 54–59. https://doi.org/10.18653/v1/N19-4010

Akbik A, Blythe D, Vollgraf R (2018) Contextual string embeddings for sequence labeling. In: Proceedings of the 27th international conference on computational linguistics. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pp 1638–1649

Araque O, Corcuera-Platas I, Sanchez-Rada JF, Iglesias CA (2017) Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Exp Syst Appl 77:236–246

Araújo M, Pereira A, Benevenuto F (2020) A comparative study of machine translation for multilingual sentence-level sentiment analysis. Inf Sci 512:1078–1102

Arif MH, Li J, Iqbal M, Liu K (2018) Sentiment analysis and spam detection in short informal text using learning classifier systems. Soft Comput 22(21):7281–7291

Baccianella S, Esuli A, Sebastiani F (2010) SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the 7th international conference on language resources and evaluation, pp 2200–2204

Bakliwal A, Arora P, Madhappan S, Kapre N, Singh M, Varma V (2012) Mining sentiments from tweets. In: Proceedings of the 3rd workshop in computational approaches to subjectivity and sentiment analysis. Association for Computational Linguistics, Jeju, Korea, pp 11–18

Barbosa L, Feng J (2010) Robust sentiment detection on Twitter from biased and noisy data. In: Proceedings of the 23rd international conference on computational linguistics: posters. Association for Computational Linguistics, pp 36–44

Bengio Y, Ducharme R, Vincent P, Janvin C (2003) A neural probabilistic language model. J Mach Learn Res 3:1137–1155

Bermingham A, Smeaton A (2010) Classifying sentiment in microblogs: is brevity an advantage?. In: Proceedings of the 19th ACM international conference on information and knowledge management. Association for Computational Linguistics, pp 1833–1836

Bifet A, Frank E (2010) Sentiment knowledge discovery in Twitter streaming data. In: Proceedings of the 13th international conference on discovery science. Springer, pp 1–15

Bojanowski P, Grave E, Joulin A, Mikolov T (2016) Enriching word vectors with subword information. CoRR **abs/1607.04606**

Bollen J, Mao H, Zeng X (2011) Twitter mood predicts the stock market. J Comput Sci 2(1):1–8

Bravo-Marquez F, Frank E, Mohammad SM, Pfahringer B (2016) Determining word-emotion associations from tweets by multi-label classification. In: 2016 IEEE/WIC/ACM international conference on web intelligence (WI), pp 536–539

Bravo-Marquez F, Frank E, Pfahringer B, Mohammad SM (2019) Affectivetweets: a weka package for analyzing affect in tweets. J Mach Learn Res 20(92):1–6

Bravo-Marquez F, Mendoza M, Poblete B (2013) Combining strengths, emotions and polarities for boosting twitter sentiment analysis. In: Proceedings of the 2nd international workshop on issues of sentiment discovery and opinion mining, WISDOM '13. Association for Computational Linguistics, New York, NY, USA. https://doi.org/10.1145/2502069.2502071

Bravo-Marquez F, Mendoza M, Poblete B (2014) Meta-level sentiment models for big social data analysis. Knowl Based Syst 69:86–99

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Brown G, Wyatt J, Harris R, Yao X (2005) Diversity creation methods: a survey and categorisation. Inf Fusion 6(1):5–20

Buscaldi D, Hernandez-Farias I (2015) Sentiment analysis on microblogs for natural disasters management: a study on the 2014 genoa floodings. In: Proceedings of the 24th international conference on world wide web, pp 1185–1188

Cambria E, Hussain A (2015) Sentic computing: a common-sense-based framework for concept-level sentiment analysis, 1st edn. Springer, Berlin

Cambria E, Hussain A, Durrani T, Havasi C, Eckl C, Munro J (2010) Sentic computing for patient centered applications. In: Proceedings of the 10th IEEE international conference on signal processing, pp 1279–1282

Cambria E, Poria S, Gelbukh A, Thelwall M (2017) Sentiment analysis is a big suitcase. IEEE Intell Syst 32(6):74–80. https://doi.org/10.1109/MIS.2017.4531228

Canuto S, Gonçalves M, Benevenuto F (2016) Exploiting new sentiment-based meta-level features for effective sentiment analysis. In: Proceedings of the 9th ACM international conference on web search and data mining. Association for Computational Linguistics, pp 53–62

Carvalho J, Plastino A (2016) An assessment study of feature and meta-level features in twitter sentiment analysis. In: Proceedings of the 22nd European conference on artificial intelligence. IOS Press, pp 769–777

Chang C, Lin C (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2:1–27

Chaturvedi I, Cambria E, Welsch RE, Herrera F (2018) Distinguishing between facts and opinions for sentiment analysis: survey and challenges. Inf Fusion 44:65–77. https://doi.org/10.1016/j.inffus.2017.12.006

Chen L, Wang W, Nagarajan M, Wang S, Sheth A (2012) Extracting diverse sentiment expressions with target-dependent polarity from Twitter. In: Proceedings of the 6th international AAAI conference on weblogs and social media, pp 50–57

Chen P, Sun Z, Bing L, Yang W (2017) Recurrent attention network on memory for aspect sentiment analysis. In: Proceedings of the 2017 conference on empirical methods in natural language processing. Association for Computational Linguistics, Copenhagen, Denmark, pp 452–461. https://doi.org/10.18653/v1/D17-1047

Chikersal P, Poria S, Cambria E, Gelbukh A, Siong CE (2015) Modelling public sentiment in twitter: Using linguistic patterns to enhance supervised learning. In: Gelbukh A (ed) Proceedings of the 16th international conference on intelligent text processing and computational linguistics. Springer International Publishing, Cairo, Egypt, pp 49–65

Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537

Cozza V, Petrocchi M (2016) mib at semeval-2016 task 4a: exploiting lexicon based features for sentiment analysis in twitter. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp 133–138

da Silva N, Colleta L, Hruschka E, Hruschka E Jr (2016) Using unsupervised information to improve semi-supervised tweet sentiment classification. Inf Sci 355:348–365

da Silva N, Hruschka E, Hruschka E Jr (2014) Tweet sentiment analysis with classifier ensembles. Decis Support Syst 66:170–179

Dashtipour K, Poria S, Hussain A, Cambria E, Hawalah AY, Gelbukh A, Zhou Q (2016) Multilingual sentiment analysis: state of the art and independent comparison of techniques. Cognit Comput 8(4):757–771

Davidov D, Tsur O, Rappoport A (2010) Enhanced sentiment learning using Twitter hashtags and smileys. In: Proceedings of the 23rd international conference on computational linguistics: posters. Association for Computational Linguistics, pp 241–249

De Smedt T, Daelemans W (2012) Pattern for python. J Mach Learn Res 13:2063–2067

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp 4171–4186. https://doi.org/10.18653/v1/N19-1423. https://www.aclweb.org/anthology/N19-1423

Diakopoulos N, Shamma D (2010) Characterizing debate performance via aggregated Twitter sentiment. In: Proceedings of the SIGCHI conference on human factors in computing systems. Association for Computing Machinery, pp 1195–1198

Dietterich TG (2000) Ensemble methods in machine learning. In: Multiple classifier systems. Springer, Berlin, pp 1–15

Dong L, Wei F, Tan C, Tang D, Zhou M, Xu K (2014) Adaptive recursive neural network for target-dependent Twitter sentiment classification. In: Proceedings of the 52nd annual meeting of the association for computational linguistics: short papers. Association for Computational Linguistics, pp 49–54

Emadi M, Rahgozar M (2019) Twitter sentiment analysis using fuzzy integral classifier fusion. J Inf Sci 46:1–17

Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) Liblinear: a library for large linear classification. J Mach Learn Res 9:1871–1874

Farias DH, Rosso P (2017) Chapter 7 - irony, sarcasm, and sentiment analysis. In: Pozzi FA, Fersini E, Messina E, Liu B (eds) Sentiment analysis in social networks. Morgan Kaufmann, Boston, pp 113–128. https://doi.org/10.1016/B978-0-12-804412-4.00007-3. http://www.sciencedirect.com/science/article/pii/B9780128044124000073

Felbo B, Mislove A, Søgaard A, Rahwan I, Lehmann S (2017) Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. arXiv preprint arXiv:1708.00524

Fersini E, Messina E, Pozzi F (2014) Sentiment analysis: Bayesian ensemble learning. Decis Support Syst 68:26–38

Fersini E, Messina E, Pozzi F (2016) Expressive signals in social media languages to improve polarity detection. Inf Proc Manag 52(1):20–35

Fu X, Wei Y, Xu F, Wang T, Lu Y, Li J, Huang JZ (2019) Semi-supervised aspect-level sentiment classification model based on variational autoencoder. Knowl Based Syst 171:81–92

Ghosh A, Li G, Veale T, Rosso P, Shutova E, Barnden J, Reyes A (2015) SemEval-2015 task 11: Sentiment analysis of figurative language in twitter. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015). Association for Computational Linguistics, Denver, Colorado, pp 470–478. https://doi.org/10.18653/v1/S15-2080

Gimpel K, Schneider N, O'Connor B, Das D, Mills D, Eisenstein J, Heilman M, Yogatama D, Flanigan J, Smith N (2011) Part-of-speech tagging for Twitter: annotation, features, and experiments. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies: short papers. Association for Computational Linguistics, pp 42–47

Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. Technical report CS224N, Stanford

Gonçalves P, Dalip D, Reis J, Messias J, Ribeiro F, Melo P, Gonçalves M, Benevenuto F (2015) Caracterizando e detectando sarcasmo e ironia no Twitter. In: Proceedings of the Brazilian workshop on social network analysis and mining

Hagen M, Potthast M, Büchner M, Stein B (2015) Twitter sentiment detection via ensemble classification using averaged confidence scores. In: Proceedings of the 37th European conference on IR research. Springer, pp 741–754

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten I (2009) The weka data mining software: an update. SIGKDD Explor Newsl 11(1):10–18

Hamdan H (2016) Sentisys at semeval-2016 task 4: feature-based system for sentiment analysis in twitter. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp 190–197

Hamdan H, Bellot P, Bechet F (2015) Lsislif: Crf and logistic regression for opinion target extraction and sentiment polarity analysis. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), pp 753–758

Hussain A, Cambria E (2018) Semi-supervised learning for big social data analysis. Neurocomputing 275:1662–1673

Hutto C, Gilbert E (2014) Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the 8th international AAAI conference on weblogs and social media

Jabreel M, Moreno A (2017) Sitaka at semeval-2017 task 4: sentiment analysis in twitter based on a rich set of features. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pp 694–699

Jiang L, Yu M, Zhou M, Liu X, Zhao T (2011) Target-dependent Twitter sentiment classification. In: Proceedings of the 49th annual meeting of the ACL: human language technologies. Association for Computational Linguistics, pp 151–160

Kathuria P (2019) Sentiment classification using WSD, maximum entropy and Naive Bayes classifiers. https://github.com/kevincobain2000/sentiment_classifier. Accessed 30 08 2019

Khuc V, Shivade C, Ramnath R, Ramanathan J (2012) Towards building large-scale distributed systems for Twitter sentiment analysis. In: Proceedings of the 27th annual ACM symposium on applied computing. Association for Computing Machinery, pp 459–464

Kingma DP, Welling M (2013) Auto-encoding variational Bayes

Kouloumpis E, Wilson T, Moore J (2011) Twitter sentiment analysis: the good the bad and the omg! In: Proceedings of the 5th international AAAI conference on web and social media, pp 538–541

Li X, Wu P, Wang W (2020) Incorporating stock prices and news sentiments for stock market prediction: a case of Hong Kong. Inf Process Manag 57(5):102212. https://doi.org/10.1016/j.ipm.2020.102212

Lin J, Kolcz A (2012) Large-scale machine learning at Twitter. In: Proceedings of the 2012 ACM SIGMOD international conference on management of data. Association for Computing Machinery, pp 793–804

Liu B (2012) Sentiment analysis and opinion mining. Synth Lect Hum Lang Technol 5(1):1–167

Liu B (2015) Sentiment analysis: mining opinions, sentiments, and emotions. Cambridge University Press, Cambridge

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) RoBERTa: a robustly optimized BERT pretraining approach

Lo SL, Cambria E, Chiong R, Cornforth D (2017) Multilingual sentiment analysis: from formal to informal and scarce resource languages. Artif Intell Rev 48(4):499–527

Lochter JV, Zanetti RF, Reller D, Almeida TA (2016) Short text opinion detection using ensemble of classifiers and semantic indexing. Expert Syst Appl 62:243–249

Loria S (2016) Textblob: simplified text processing. https://textblob.readthedocs.io/en/dev/index.html. Accessed 08 30 2019

Ma Y, Peng H, Cambria E (2018) Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In: Proceedings of 32nd AAAI conference on artificial intelligence. New Orleans, Louisiana, pp 5876–5883

Manning C, Surdeanu M, Bauer J, Finkel J, Bethard S, McClosky D (2014) The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. Association for Computational Linguistics, Baltimore, Maryland, pp 55–60

Mansour R, Hady MFA, Hosam E, Amr H, Ashour A (2015) Feature selection for twitter sentiment analysis: An experimental study. In: Gelbukh A (ed) Proceedings of the 16th international conference on intelligent text processing and computational linguistics. Springer International Publishing, Cairo, Egypt, pp 92–103

Martínez-Cámara E, Martín-Valdivia M, Ureña-López L, Montejo-Ráez A (2014) Sentiment analysis in twitter. Nat Lang Eng 20(1):1–28

Maynard D, Bontcheva K (2016) Challenges of evaluating sentiment analysis tools on social media. In: Proceedings of the 10th international conference on language resources and evaluation (LREC'16). European Language Resources Association (ELRA), Portorož, Slovenia, pp 1142–1148

Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781

Mikolov T, Grave E, Bojanowski P, Puhrsch C, Joulin A (2018) Advances in pre-training distributed word representations. In: Proceedings of the international conference on language resources and evaluation (LREC 2018)

Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th international conference on neural information processing systems, vol 2, NIPS'13, pp 3111–3119

Miranda-Jiménez S, Graff M, Tellez ES, Moctezuma D (2017) Ingeotec at semeval 2017 task 4: a b4msa ensemble based on genetic programming for twitter sentiment analysis. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pp 771–776

Mohammad S, Kiritchenko S, Zhu X (2013) Nrc-canada: building the state-of-the-art in sentiment analysis of tweets. In: Proceedings of the 7th international workshop on semantic evaluation exercises. Atlanta, Georgia, USA

Mohammad S, Turney P (2013) Crowdsourcing a word-emotion association lexicon. Comput Intell 29(3):436–465

Mohammad SM, Bravo-Marquez F, Salameh M, Kiritchenko S (2018) Semeval-2018 task 1: affect in tweets. In: Proceedings of 12th international workshop on semantic evaluation (SemEval 2018). Association for Computational Linguistics, New Orleans, LA, USA

Nakov P, Ritter A, Rosenthal S, Stoyanov V, Sebastiani F (2016) SemEval-2016 task 4: sentiment analysis in Twitter. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval 2016), SemEval'16. Association for Computational Linguistics, San Diego, California

Nakov P, Rosenthal S, Kozareva Z, Stoyanov V, Ritter A, Wilson T (2013) SemEval-2013 task 2: sentiment analysis in twitter. In: Proceedings of the 7th international workshop on semantic evaluation (SemEval 2013). Association for Computational Linguistics, Atlanta, Georgia, pp 312–320

Narayanan V, Arora I, Bhatia A (2013) Fast and accurate sentiment classification using an enhanced naive Bayes model. In: Intelligent data engineering and automated learning—IDEAL 2013. Springer, Berlin, pp 194–201

Narr S, Hulfenhaus M, Albayrak S (2012) Language-independent Twitter sentiment analysis. In: Proceedings of the workshop on knowledge discovery, data mining and machine learning

Nielsen FÅ (2011) A new ANEW: evaluation of a word list for sentiment analysis in microblogs. CoRR **abs/1103.2903**. http://arxiv.org/abs/1103.2903

Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of the 7th international conference on language resources and evaluation, pp 1320–1326

Pang B, Lee L, 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. Association for Computational Linguistics, pp 79–86

Park JH, Xu P, Fung P (2018) Plusemo2vec at semeval-2018 task 1: exploiting emotion knowledge from emoji and# hashtags. In: Proceedings of the 12th international workshop on semantic evaluation (SemEval-2018), pp 264–272

Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Empirical methods in natural language processing (EMNLP), pp 1532–1543. http://www.aclweb.org/anthology/D14-1162

Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (long papers). Association for Computational Linguistics, New Orleans, Louisiana, pp 2227–2237. https://doi.org/10.18653/v1/N18-1202. https://www.aclweb.org/anthology/N18-1202

Petrović S, Osborne M, Lavrenko V (2010) The Edinburgh twitter corpus. In: Proceedings of the NAACL HLT 2010 workshop on computational linguistics in a world of social media. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 25–26

Poria S, Cambria E, Gelbukh A (2016) Aspect extraction for opinion mining with a deep convolutional neural network. Knowl Based Syst 108:42–49. https://doi.org/10.1016/j.knosys.2016.06.009 New Avenues in Knowledge Bases for Natural Language Processing

Pouyanfar S, Sadiq S, Yan Y, Tian H, Tao Y, Reyes MP, Shyu ML, Chen SC, Iyengar SS (2018) A survey on deep learning: algorithms, techniques, and applications. ACM Comput Surv 51(5):1–36

Prusa J, Khoshgoftaar TM, Dittman DJ (2015) Using ensemble learners to improve classifier performance on tweet sentiment data. In: 2015 IEEE international conference on information reuse and integration, pp 252–257

Reyes A, Rosso P, Veale T (2013) A multidimensional approach for detecting irony in twitter. Lang Resour Eval 47(1):239–268

Rosenthal S, Farra N, Nakov P (2017) SemEval-2017 task 4: sentiment analysis in Twitter. In: Proceedings of the 11th international workshop on semantic evaluation (SemEval 2017), SemEval'17. Association for Computational Linguistics, Vancouver, Canada

Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. In: Proceedings of the 31st international conference on neural information processing systems, NIPS'17. Curran Associates Inc., Red Hook, NY, USA, pp 3859–3869

Saif H (2015) Semantic sentiment analysis of microblogs. Ph.D. thesis, The Open University. http://oro.open.ac.uk/44063/

Saif H, Fernandez M, He Y, Alani H (2013) Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold. In: Proceedings of the 1st workshop on emotion and sentiment in social and expressive media

Saif H, He Y, Alani H (2012) Alleviating data sparsity for Twitter sentiment analysis. In: Proceedings of the 2nd workshop on making sense of microposts. CEUR-WS, pp 2–9

Satapathy R, Guerreiro C, Chaturvedi I, Cambria E (2017) Phonetic-based microtext normalization for twitter sentiment analysis. In: 2017 IEEE international conference on data mining workshops (ICDMW), pp 407–413. https://doi.org/10.1109/ICDMW.2017.59

Siddiqua UA, Ahsan T, Chy AN (2016) Combining a rule-based classifier with ensemble of feature sets and machine learning techniques for sentiment analysis on microblog. In: 2016 19th international conference on computer and information technology (ICCIT), pp 304–309

Sousa L, de Mello R, Cedrim D, Garcia A, Missier P, Uchôa A, Oliveira A, Romanovsky A (2018) Vazadengue: an information system for preventing and combating mosquito-borne diseases with social networks. Inf Syst 75:26–42. https://doi.org/10.1016/j.is.2018.02.003

Speriosu M, Sudan N, Upadhyay S, Baldridge J (2011) Twitter polarity classification with label propagation over lexical links and the follower graph. In: Proceedings of the 1st workshop on unsupervised learning in NLP. Association for Computational Linguistics, pp 53–63

Tang D, Wei F, Yang N, Zhou M, Liu T, Qin B (2014) Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (vol 1: long papers). Association for Computational Linguistics, Baltimore, Maryland, pp 1555–1565. https://doi.org/10.3115/v1/P14-1146. https://www.aclweb.org/anthology/P14-1146

Thelwall M, Buckley K, Paltoglou G (2012) Sentiment strength detection for the social web. J Am Soc Inform Sci Technol 63(1):163–173

Ting KM, Witten IH (1999) Issues in stacked generalization. J Artif Intell Res 10:271–289

Tumasjan A, Sprenger TO, Sandner PG, Welpe IM (2010) Predicting elections with twitter: what 140 characters reveal about political sentiment. In: Fourth international AAAI conference on weblogs and social media

Turney PD (2002) Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting on association for computational linguistics, ACL'02. Association for Computational Linguistics, USA, pp 417–424

Valdivia A, Luzón MV, Herrera F (2017) Sentiment analysis in tripadvisor. IEEE Intell Syst 32(4):72–77

Vo D, Zhang Y (2016) Don't count, predict! an automatic approach to learning sentiment lexicons for short text. In: Proceedings of the 54th annual meeting of the association for computational linguistics. Association for Computing Machinery

Vo DT, Zhang Y (2015) Target-dependent twitter sentiment classification with rich automatic features. In: Proceedings of the 24th international conference on artificial intelligence, IJCAI'15. AAAI Press, pp 1347–1353

Vosoughi S, Vijayaraghavan P, Roy D (2016) Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In: Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval, SIGIR'16. ACM, New York, NY, USA, pp 1041–1044

Wang B, Liakata M, Zubiaga A, Procter R (2017) TDParse: multi-target-specific sentiment recognition on twitter. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics: volume 1, long papers. Association for Computational Linguistics, Valencia, Spain, pp 483–493

Wang H, Can D, Kazemzadeh A, Bar F, Narayanan S (2012) A system for real-time twitter sentiment analysis of 2012 U.S. presidential election cycle. In: Proceedings of the ACL 2012 system demonstrations, ACL'12. Association for Computational Linguistics, USA, pp 115–120

Wang H, Can D, Kazemzadeh A, Bar F, Narayanan S (2012) A system for real-time Twitter sentiment analysis of 2012 US presidential election cycle. In: Proceedings of the ACL 2012 system demonstrations. Association for Computational Linguistics, pp 115–120

Wasden L (2010) Internet lingo dictionary: a parents' guide to codes used in chat rooms, instant messaging, text messaging, and blogs. Technical report, Office of the Attorney General

Wiegand M, Balahur A, Roth B, Klakow D, Montoyo A (2010) A survey on the role of negation in sentiment analysis. In: Proceedings of the workshop on negation and speculation in natural language processing. Association for Computational Linguistics, pp 60–68

Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing. Association for Computational Linguistics, pp 347–354

Wolpert DH (1992) Stacked generalization. Neural Netw 5(2):241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

Xing FZ, Cambria E, Welsch RE (2018) Intelligent asset allocation via market sentiment views. IEEE Comput Intell Mag 13(4):25–34

Xing FZ, Cambria E, Zhang Y (2019) Sentiment-aware volatility forecasting. Knowl Based Syst 176:68–76

Xu P, Madotto A, Wu C, Park JH, Fung P (2018) Emo2vec: learning generalized emotion representation by multi-task training. In: Proceedings of the EMNLP WASSA workshop

Yang M, Zhao W, Ye J, Lei Z, Zhao Z, Zhang S (2018) Investigating capsule networks with dynamic routing for text classification. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Association for Computational Linguistics, Brussels, Belgium, pp 3110–3119

Yoo S, Song J, Jeong O (2018) Social media contents based sentiment analysis and prediction system. Expert Syst Appl 105:102–111

Young T, Hazarika D, Poria S, Cambria E (2018) Recent trends in deep learning based natural language processing [review article]. IEEE Comput Intell Mag 13(3):55–75. https://doi.org/10.1109/mci.2018.2840738

Zhang CX, Duin RP (2011) An experimental study of one- and two-level classifier fusion for different sample sizes. Pattern Recognit Lett 32(14):1756–1767. https://doi.org/10.1016/j.patrec.2011.07.009

Zhang L, Ghosh R, Dekhil M, Hsu M, Liu B (2011) Combining lexicon-based and learning-based methods for Twitter sentiment analysis. Technical report HPL-2011-89, HP Laboratories

Zhao W, Peng H, Eger S, Cambria E, Yang M (2019) Towards scalable and reliable capsule networks for challenging NLP applications. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, Italy, pp 1549–1559

Zimbra D, Abbasi A, Zeng D, Chen H (2018) The state-of-the-art in twitter sentiment analysis: a review and benchmark evaluation. ACM Trans Manag Inf Syst. https://doi.org/10.1145/3185045