Information Sciences 480 (2019) 273–286

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

E²SAM: Evolutionary ensemble of sentiment analysis methods for domain adaptation

Miguel López, Ana Valdivia, Eugenio Martínez-Cámara*, M. Victoria Luzón, Francisco Herrera

Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Spain

ARTICLE INFO

Article history:
Received 29 March 2018
Revised 12 December 2018
Accepted 20 December 2018
Available online 21 December 2018

Keywords:
Sentiment analysis
Ensembles classifiers
Genetic algorithms

ABSTRACT

Currently, a plethora of industrial and academic sentiment analysis methods for classifying the opinion polarity of a text are available and ready to use. However, each of those methods have their strengths and weaknesses, due mainly to the approach followed in their design (supervised/unsupervised) or the domain of text used in their development. The weaknesses are usually related to the capacity of generalisation of machine learning algorithms, and the lexical coverage of linguistic resources. Those issues are two of the main causes of one of the challenges of Sentiment Analysis, namely the domain adaptation problem. We argue that the right ensemble of a set of heterogeneous Sentiment Analysis Methods will lessen the domain adaptation problem. Thus, we propose a new methodology for optimising the contribution of a set of off-the-shelf Sentiment Analysis Methods in an ensemble classifier depending on the domain of the input text. The results clearly show that our claim holds.

1. Introduction

Sentiment Analysis (SA) is the field of Natural Language Processing (NLP) whose aim is to analyse automatically subjective information. People often express their opinions or sentiments towards events, topics, proposals, companies or products [23]. This area of study has grown over last few years due to the large amount of text stored in the Web 2.0 such as social networks, blogs and discussion platforms. Consequently, the interest in developing Sentiment Analysis Methods (SAMs) capable of detecting the polarity of a text has also risen [35], and nowadays there is a great variety of different tools trained for extracting and classifying opinions. The task of polarity classification can be defined as a binary or multi-class classification problem. In this paper, we consider polarity detection as a three class (`Positive`, `Neutral` and `Negative`) classification problem.

The performance of a SAM strongly depends on the learning approach followed [20]. Supervised based SAMs are mainly determined by the domain (financial [5], restaurant [22] or health [3]) and the genre (news, microblogging or reviews) of the data used in their training, whereas the unsupervised models depend on the language coverage of the linguistic resources used in their development. Consequently, there are two problems: (1) the generalisation capacity of machine learning (ML) algorithms, and (2) the lexical coverage of linguistic resources. These problems are widely known in the literature of SA as

* Corresponding author.
E-mail addresses: miguelberja@correo.ugr.es (M. López), avaldivia@ugr.es (A. Valdivia), emcamara@decsai.ugr.es (E. Martínez-Cámara), luzon@ugr.es (M.V. Luzón), herrera@decsai.ugr.es (F. Herrera).

https://doi.org/10.1016/j.ins.2018.12.038
0020-0255/

the domain adaptation problem for polarity detection and classification. For instance, the word *unpredictable* is negative in the domain of car reviews, "*my car has an unpredictable steering*". However, *unpredictable* is positive in the domain of film reviews, "*the plot of the last film that I watched is unpredictable*" [23].

Due to the high industrial demand of SA methods, several off-the-shelf SAMs have been released in the last few years. Each one of those SAMs has its own characteristics, i.e. training data, learning approach, features used for representing the input text and so on. In summary, each of them have their advantages and drawbacks. Specifically, their main drawback is related to the domain adaptation problem, which means that they only perform well when they classify domain text which is similar to that of the training set.

In this paper, we argue that the domain adaptation problem can be diminished by the right combination of a set of off-the-shelf SAMs. Accordingly, we propose a new methodology called Evolutionary Ensemble of SAMs (E^2SAM) for learning the most suitable combination of a set of off-the-shelf SAMs depending on the domain of the input text. E^2SAM is built upon an evolutionary algorithm (EA), which is able to optimise the contribution of each base SAM [12]. Since E^2SAM is based on a EA, we assessed the performance of three EAs in our specific scenario, specifically the implementation of a Memetic Algorithm [30], and the algorithms L-SHADE [42] and jSO [8], which reached strong results in CEC competitions.[1]

We select 7 off-the-shelf SAMs from the state-of-the-art, and we evaluate E^2SAM on 13 corpora of reviews from different domains and text genre. We compare our proposal with two baselines ensemble methods described in [45] in order to demonstrate its effectiveness. The results show that E^2SAM substantially outperforms the best SAM and the two baselines in 11 of the 13 corpora, which confirms the validity of E^2SAM.

The rest of this work is organized as follows: Section 2 describes some related studies; Section 3 presents our proposal; Section 4 shows and analyses the results, and Section 5 details the conclusions and future work.

2. Sentiment analysis and related works

We briefly describe the SA task in Section 2.1, the use of evolutionary algorithms in SA in Section 2.2, the use of ensemble methods in SA in Section 2.3 and the challenge of domain adaptation in Section 2.4.

2.1. Sentiment analysis

SA is defined as the computational treatment of opinions, sentiments and subjectivity in text [33], but a definition that better matches the current state of SA is the set of computational techniques for extracting, classifying, understanding, and assessing the opinions expressed in various on-line news sources, social media comments, and other user-generated contents [9].

SA is usually split up in two subtasks, namely subjectivity classification, the categorization of subjective and objective utterances [10]; and polarity classification, identification of the valence or polarity of a subjective utterance [23]. In this paper, we contribute to the polarity classification task through the optimisation of an ensemble meta-classifier. There is an ample research on polarity classification systems, so we recommend the reading of [6,23,33,41] to get a general idea of the task.

2.2. Evolutionary algorithms in SA

EAs [12] are optimisation algorithms with a wide variety of applications, and one of them is feature selection. SA systems, as other NLP systems, usually need a big set of hand-crafted features, the right selection of the most informative ones is crucial in order to reach an accurate classification. Thus, there are several studies that use genetic algorithms (GA) for selecting the best features in SA.

Abbasi et al. [1] developed an entropy weighted GA for the polarity classification at document level of reviews written in English and Arabic. The authors highlighted the suitability of GAs for the selection of features. Onan and Korukoğlu [31] used GAs for the ensemble of several traditional feature selection methods. The results of the ensemble of feature selection methods addressed by a GA outperformed all the results reached by each feature selection method. Das and Bandyopadhyay [11] also used GAs for feature selection in SA. Specifically, they studied a broad set of features and selected the most informative for the classification of reviews in English and Bengali.

Genetic programming have been also used in SA in micro-blogging platforms. Moctezuma et al. [29] proposed a genetic programming approach to combine the output of several supervised polarity classification systems of tweets written in Spanish. However, their method is agnostic with regard to the relation between the polarity labels and the base classifiers. In contrast, our EA needs to know the relation between the estimations and the base systems, because it assigns a different weight according to the performance of each base system.

Since the previous studies are mainly focused on the ensemble of machine learning methods and the use of EAs for the selection of features, our contribution is novel in the sense that we study the use of EAs to optimise the ensemble of a set of off-the-shelf polarity classification systems, and we evaluate our proposal in a wide and diverse number of sentiment corpora.

[1] http://www.ntu.edu.sg/home/epnsugan/index_files/cec-benchmarking.htm.

2.3. Ensemble methods in SA

The main use of ensemble methodology is to combine a set of classifiers in order to enhance the accuracy of the estimations that can be achieved by using a single classifier [36]. Further details about ensemble learning in [48].

Researchers have made use of ensemble strategies for improving the performance of polarity classification in one language and also in a multilingual environment. Kennedy and Inkpen [21] attempted to improve the performance of a supervised polarity classification system based on Support Vector Machines (SVM) by the development of ensemble methods with a lexicon-based polarity classification system. The authors compared a weighted voting system and a meta-classifier based on the use of the output of the two base classifiers as features of an SVM classifier. The results show that the meta-classifier grounded in SVM outperformed the weighted voting system. The lower results of the voting system might be caused by the lack of an optimisation of the weights of each base system. In contrast, we will show that the use of an optimisation method can enhance the performance of a voting system.

Appel et al. [2] also evaluated the performance of a voting system in the domain of movie reviews as in [21]. However, the proposal of Appel et al. [2] consisted in a majority voting system semantically close to the fuzzy linguistic quantifier "most of", which acquired good results. Prior to Appel et al. [2], Fersini et al. [14] proposed a voting system based on bayesian learning, roughly speaking, a model selection method based on the study of the contribution of each base classifier when is used with other classifiers.

Ensemble methods have also been used for enhancing the performance of SAMs in texts from microblogging platforms. For instance, da Silva et al. [39] proposed an average weighting system of several machine learning base classifiers. They evaluated their method in several corpora of tweets and they reached promising results. Martínez-Cámara et al. [27] proposed a majority voting system for the combination of three different classifiers for the polarity classification of tweets written in Spanish, specifically from the General Corpus of TASS.[2] Two of the base polarity systems used SVM as classification system but they used totally different feature sets. On the other hand, the third one was an unsupervised system grounded in the combination of several opinion lexicons. The ensemble method outperformed the three base classifiers.

Focused on polarity detection, Valdivia et al. [46] also proposed different ensemble models for detecting neutrality guided by fuzzy operators. They enhanced the performance of the system by removing those neutral reviews labelled by a consensus of SAMs. In our work, we implement two of their proposed aggregation systems.

One of the issues of NLP and SA is the lack of linguistic resources for some languages. Therefore, researchers usually develop ensemble methods combining several polarity classification systems for English language and almost one polarity classification system in the target language. Wan [47] evaluated several voting schemes for the combination of two unsupervised polarity classification systems, the first one classified reviews written in Chinese and the second one classified the translated into English version of the Chinese reviews. Martínez-Cámara et al. [28] described a stacking methodology for the combination of two unsupervised systems for the classification of reviews written in Spanish, and two unsupervised systems for the classification of reviews written in English. The results show that the incorporation of the information from the systems for English reviews in the ensemble method was critical to improving performance of the classification of the reviews written in Spanish.

2.4. Domain adaptation

The domain adaptation problem arises when there is a difference between the distribution of the training data and the distribution of the test data. Besides, the domain adaptation problem also arises when the training data comes from different domains or from different tasks. In these scenarios, multi-task and multi-view learning methods are useful to face the domain adaptation problem [24].

Jiang and Zhai [19] identified two kinds of domain adaptation: labelling adaptation and instance adaptation. When the same feature distribution follows a different labelling function, we face a labelling adaptation problem. On the other hand, the instance adaptation problem means that the feature distribution of the instances of different domains is dissimilar. In the context of SA, Blitzer et al. [7] proposed a method for resolving the instance adaptation problem by finding a match between features from source and target domains through modelling their correlations with pivot features. Pan et al. [32] proposed the use of spectral clustering to align domain-specific and domain-independent words into a set of feature-clusters. The results reached by Pan et al. [32] are higher than those obtained by Blitzer et al. [7]. In contrast to the two previous studies, which only face the instance labelling problem, Xia et al. [49] proposed the joint treatment of the instance and labelling adaptation problem, which obtained good results.

Our proposal works on the labelling adaptation problem and is able to improve the generalization capacity of the end classification system, because it optimises the ensemble of a set of SA base classifiers, which are trained on data from different domains and genre.

[2] Further details about the corpus at: http://www.sepln.org/workshops/tass/.

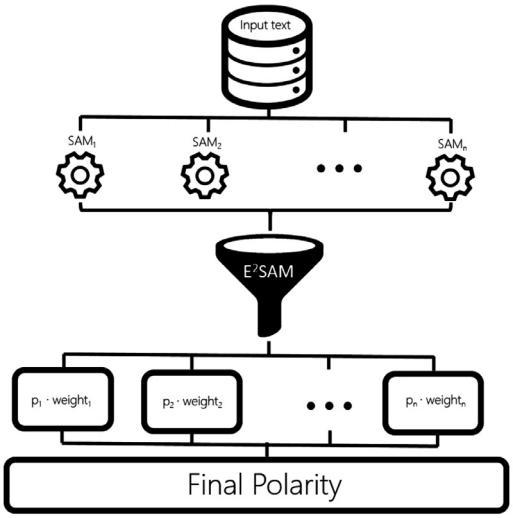

Fig. 1. Workflow of our proposal, which is composed of a set of base SAMs and an evolutionary method (ESAM) that calculates the right weight of each base SAM.

3. Evolutionary ensemble optimisation method

The performance of an off-the-shelf SAM depends on its learning approach and the relation between the domain and genre of the training and test data. We know that ensemble methods are able to overcome the results of individual classifiers, because they join the search space of the base systems and, consequently, they are able to find a better solution for specific expert domains. However, the challenge in the development of ensemble methods is how to learn the contribution of each base system in the joint solution of the ensemble.

We argue that EAs have the ability of finding out the right contribution of a set of base systems depending on the domain of the input data. Fig. 1 depicts our evaluation framework, which is composed of a set of base SAMs, and an ensemble method that calculates the contribution of each base SAM to the calculation of the final polarity value. Section 3.1 describes the base SAMs, Section 3.2 describes the developed SA baseline ensembles, and Section 3.3 presents the details of the three EAs analysed for the development of our proposal.

3.1. Base sentiment analysis methods

In recent years, several off-the-shelf SAMs have been released. Those SAMs follow one of the two main approaches in SA, namely semantic orientation (SO) methods [44] and ML methods [34].

SO strategies are methods which calculate the polarity by means of unsupervised methods driven by rules that use linguistic resources. Since SO methods are based on linguistic resources such as specific linguistic rules or lists of words, they show a poorer performance than ML methods, because the language coverage of these methods is limited by the coverage of the linguistic resources used in their development.

On the other hand, ML methods consist in the use of machine learning algorithms, such as probabilistic classifiers (Naïve Bayes, Bayesian Networks...), linear models (Support Vector Machines), decision trees and so on. ML methods need that the texts to be represented by a set of features, which reflect the linguistic properties of interest for the specific classification task. Some of these features are the frequency of the words, the pos-tags or the number of positive and negative words.

Since the success of an ensemble method depends on the heterogeneity of the base classifiers, we selected a set of SAMs trained with corpora from different domains and genre, which are built upon different approaches. Table 1 shows the SAMs included in our analysis, and we provide their name (Name), their main characteristics (Description), the approach followed in their development (Approach), the kind of output (Output), the domain (Domain) and the genre (Genre) of the data used in their development.[3]

3.2. Ensembles for enhancing SAMs performance

The possible ways of combining the outputs of a set of classifiers, in our case a set of SAMs (S), depends on the information returned by each of them. Each SAM of S produces a polarity value $p_i \in P$, where P is the set of possible polarity

[3] Further details about the Azure, MeaningCloud and Syuzhet SAMs may be respectively read in https://www.microsoft.com/cognitive-services/en-us/text-analytics-api, https://www.meaningcloud.com/products/sentiment-analysis and https://github.com/mjockers/syuzhet.

Table 1
Main characteristics of the SAMs used in our evaluation.

Name	Description	Approach	Output	Domain	Genre
Azure	Supervised ML method that works at document level. It was trained with a large corpus of reviews.	ML	$[0, 1] \in \mathbb{R}$	General	Reviews
Bing [17]	SO method that works at document level. It uses an opinion lexicon, and the polarity value of the document is the average polarity value of its sentences.	SO	$\{-1, 0, 1\}$	General	NA
CoreNLP [26]	Supervised ML method that works at document/sentence level. It is based on a Recursive Neural Tensor Network (RNTN), and classifies 5 intensity levels of opinion.	ML	$\{0, 1, 2, 3, 4\}$	Review	Movie reviews
MeaningCloud	SO method that works at document level. It is grounded in the pos-tags of the words and the use of an opinion lexicon, and it classifies 5 intensity levels of opinion.	SO	$\{0, 1, 2, 3, 4\}$	NA	NA
SentiStrength [43]	Supervised ML method grounded in features built upon opinion lexicons that work at document/sentence level. The output is the positive (Pos) and the negative (Neg) value of the input text.	SO & ML	Neg: $[-5, -1] \in \mathbb{Z}$; Pos: $[1, 5] \in \mathbb{Z}$	General	Social media
Syuzhet	SO method that works at document/sentence level. It uses an opinion lexicon. The document polarity value is the average of the polarity value of its sentences.	SO	$\{-1, 0, 1\}$	General	Novels
Vader [18]	SO method that works at document level and uses an opinion lexicon.	SO	$[-1, 1] \in \mathbb{R}$	General	Micro-blogging

values. Accordingly, for each given opinion o from a corpus of opinions O, the outputs of the S classifiers define a vector $\boldsymbol{p} = (p_1, \ldots, p_s)$. Therefore, given the vector $\boldsymbol{p}$, an ensemble classifier is formally defined in Eq. (1).

$$\begin{aligned} f_{ens} &: [0, 1]^{|S|} \to [0, 1] \\ (p_1, \ldots, p_s) &\mapsto f_{ens}(p_1, \ldots, p_s) \end{aligned} \tag{1}$$

According to Eq. (1), we propose a function f_{ens} built upon an evolutionary optimisation algorithm. In order to propose the most adequate f_{ens} function, we compared three EAs grounded in a different approach (see Section 3.3). The most suitable EA for the development of the f_{ens} is then compared with two baselines.

The function baselines are weighting functions, and they were presented in [45]. These models aim at assigning weights to the different SAMs in order to do a linear combination, whose output is the final polarity value. The two models are described as follows.

Average Based Model (AVG). It is a weighted aggregation model, in which all the SAMs contribute with the same weight to the final polarity value. Eq. (2) redefines the function of Eq. (1) as f_{avg}.

$$\begin{aligned} f_{avg}(p_1, \ldots, p_s) &= \sum_{i=1}^{|S|} w_i p_i \\ w_i &= \frac{1}{|S|}, \ \forall i \in \{1, \ldots, s\} \end{aligned} \tag{2}$$

Neutral Penalty Based Model (NEUTY). It gives less importance to those SAMs that estimate more neutral polarities. Eq. (3) redefines the function of Eq. (1) as f_{nuety}.

$$\begin{aligned} f_{neuty}(p_1, \ldots, p_s) &= \textstyle\sum_{i=1}^{|S|} w_i p_i \\ w_i &= \frac{|p_i - 0.5|}{\sum_{j=1}^{|S|} |p_j - 0.5|}, \ \forall i \in \{1, \ldots, s\} \end{aligned} \tag{3}$$

3.3. Evolutionary ensemble of SAMs (E^2SAM)

The previous two baselines (AVG and NEUTY) define an assignation of weights to each base SAM. In contrast, we claim that it is feasible to learn the right weight to assign to each SAM according to the domain of the input text. Since the assignation of weights may be addressed as an optimisation task, we propose an EA for optimising the contribution of each base SAM. Formally, Eq. (4) redefines the function of Eq. (1) as f_{e^2sam}.

$$f_{e^2sam}(p_1, \ldots, p_s) = \sum_{i=1}^{|S|} w_i p_i, \tag{4}$$

where the weights (w_i) are learnt using an EA.

We compare the performance of three EAs for the automatic assignation of weights to each SAM, i.e. three different EAs for the development of the function f_{e^2sam}.

EAs are optimisation algorithms that model natural evolution processes. These methods work on a set or population of possible solutions, and they are mainly composed of two processes: (1) A method that changes the set of solutions; and (2) a method that select the solutions to be kept and those ones to be removed from the pool of feasible solutions. According to the iterative nature of these algorithms, we have to conduct a number of iterations or chromosome evaluations. We empirically found out that 100,000 evaluations of the fitness function allow us to reach a good convergence for our study.

We detail the three EAs in the following subsections, specifically the MA implementation in Section 3.3.1, the two EA based on Differential Evolution, L-SHADE and jSO, in Section 3.3.2.

3.3.1. Memetic algorithms

Memetic algorithms (MA) are population-based metaheuristics composed of an EA and a set of local search algorithms [30]. The combination of global exploration (EA) and the local search allows MAs to achieve strong results while avoiding premature convergence [16]. Although MAs prevent reaching a local optimum, we need to make the solution search space larger in order to increase the likelihood of finding out the best solution, in other words, we need to make more diverse the search space. In this work, we use GAs to increase the diversity of the search space or population.

GAs are theoretically and empirically proven algorithms that provide a robust search in complex spaces. GAs model sexual reproduction, which is featured by recombining two parent strings or solutions into an offspring. The recombination operation is called crossover, which is the recombination of the selected solutions in the hope of producing a child with better fitness levels than its parents. For instance, the improvement of the fitness function is crucial in parameter optimisation problems with real coding [4]. Further details about GAs in [37].

A GA is defined by its components, hence we describe the components and configuration of our MA in the following lines.

Fitness function. It is the function to optimise (f_{opt}). Specifically, we optimise F_1 (see Eq. (8) in Section 4.2).

Chromosome. Representation of the solutions that aim at optimising the f_{opt} function. In our scenario, a solution is the set of weights for each base SAM. For example, given seven base SAMs, the size of the chromosomes c^1 and c^2 is seven, and their values will be $c^1 = (0.15, 0.15, 0.2, 0.05, 0.4, 0.02, 0.03)$ and $c^2 = (0.5, 0.0, 0.0, 0.2, 0.1, 0.1, 0.1)$.

Crossover Operator. It is the operator that performs the sexual reproduction or recombination of the chromosomes. Specifically, we use a Blend crossover operator (BLX-α) [13]. BLX-α generates the consecutive offspring as follows:

1. It randomly chooses two parents c^1 and c^2 from the population of chromosomes.
2. A value of each element c_i^n of the offspring vector c^n is randomly chosen from the interval $[C_i^1; C_i^2]$ following the uniform distribution:

$$\begin{aligned} C_i^1 &= min(c_i^1, c_i^2) - \alpha d_i \\ C_i^2 &= max(c_i^1, c_i^2) + \alpha d_i \\ d_i &= |c_i^1 - c_i^2| \end{aligned} \tag{5}$$

where c_i^1 and c_i^2 are the ith elements of c^1 and c^2 respectively, and α is a positive number to proportionally extend the interval of the parameter domain, d_i. In our proposal, we use the default value of α 0.1 [13].

Tournament selection scheme. It selects the chromosomes which will participate in the next generation. The selection is carried out after n binary tournaments. The number of tournaments is the same as the size of the population, which in our case is 30. The winning chromosome in each tournament is the one with a higher f_{opt} value.

Mutuation operator. It adds exploration capacity to the algorithm, because it randomly mutates some components of the solutions in order to explore the domain spaces of different solutions. The operator is run with a probability of 0.001, and it consists in adding to a gene a z random value that follows a Normal Distribution with mean 0 and standard deviation 0.3. For example, if the 3rd gene of our chromosome c^1, c_3^1, is randomly chosen to be mutated, and the random number z is 0.05, then the chromosome c^1 would result in $c^1 = (0.15, 0.15, 0.25, 0.05, 0.4, 0.02, 0.03)$. However, since we have to keep the constraint that weights addition must be 1, then the genes would be recalculated as $c_i^1 = \frac{c_i^1}{\sum_{j=1}^{j=n} c_j^1}$, where n is the number of genes. Consequently, c^1 would result in $c^1 = (0.143, 0.143, 0.238, 0.048, 0.38, 0.019, 0.029)$.

Replacement scheme. It defines how a new population will replace the old one for the next generation. Our replacement scheme replaces the entire old population with the new one and follows an elitist replacement approach, which consists in replacing the worst chromosome of the current population with the best one of the previous population.

Local search. As we mentioned before, MA runs a local search over some chromosomes of the population of each certain number of generations. We use the widely know Hill Climbing algorithm [38] as local search strategy. The local search is run every 10 generations on each of the chromosomes that there are in the population. This local search explores surrounding chromosomes, generating a maximum number of neighbours five times the size of the chromosome.

3.3.2. Differential evolution

Differential Evolution (DE) [40] is a robust evolutionary optimisation approach, which consists in a population of real parameters vectors $x_i, i \in \{0, \ldots, \text{NP-1}\}$, of size NP and dimension D, which are randomly initialised and are part of each generation G. A generic DE method is composed of the following components:

Mutation. The mutation operator generates a mutation of the solution vector x_i, adding a vector x_{r1} and a difference of two vectors (x_{r2} and x_{r3}) weighted by a constant parameter F (see Eq. (6)).

$$v_i = x_{r1} + F(x_{r2} - x_{r3}) \tag{6}$$

where $r1$, $r2$ and $r3$ are random indexes $\in \{0, \ldots, NP-1\}$.

Crossover. It generates an offspring vector u_i according to Eq. (7).

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} \text{ if}(randb(j) \leq CR) \text{ or } j = rnbr(i) \\ x_{ji,G} \quad \text{if } (randb(j) > CR) \text{ and } j \neq rnbr(i) \end{cases} \tag{7}$$

Where $j = 0, 1, \ldots, D\text{-}1$, CR is a control parameter $\in [0, 1]$, $randb(j)$ is a random number $\in [0, 1]$, $rnrb(i)$ is a random integer $\in \{0, \ldots, D\text{-}1\}$ and then G+1 refers to the next generation.

Selection. It selects the vector u_i which will be part of the next generation G+1. It consists in a comparison between the fitness function value of u_i and its target vector x_i. Vector u_i replaces x_i if the fitness value of x_i is improved.

In this work, we use two well-known EAs according to the CEC competitions,[4] namely L-SHADE [42] and jSO [8]. L-SHADE reached the first position in CEC 2014[5] competition and jSO reached the second position in CEC 2017.

L-SHADE does not require a pre-defined value of the parameters F and CR, because it automatically calculates the most suitable ones. L-SHADE also adds some modifications to the default mutation operator and a linear reduction to the population. Further details can be found in [42]. jSO is based on L-SHADE, but it has a different mutation operator. It also incorporates a set of rules for increasing the solution exploration capacity of the algorithm. Further details can be found in [8].

4. Experimental setup

In this section we describe the set up of our evaluation. First, we depict the corpora employed (Section 4.1), subsequently we define the evaluation framework (Section 4.2), then how the output of each of the base off-the-shelf SAMs is homogenised (Section 4.3), and the results reached by the baselines and our proposal and their analysis are in Section 4.5.

4.1. Datasets

We have used a subset of all the corpora utilised in [35] for our evaluation. Although, all the details of the corpora are in [35], we are going to succinctly describe the selected ones.

Since we want to show that the performance of a SAM is strongly dependent on the nature of their training corpus, we employ 13 datasets of multiple domains and genres. Therefore, the domain and genre of some of the corpora used in the evaluation is similar to the corpora used for training some of the base SAMs. For instance, movie reviews (`pang_movie`, `vader_movie`) or text from micro-blogging platforms (`debate`, `vader_twitter`, `english_dailabor`, `tweet_semevaltest`, `sentistrength_twitter`).

We also employ datasets whose domains and genres are not used to train the selected SAMs. These datasets are comments of technological or scientific forums or websites (`sentistrength_digg`), comments on websites (`sentistrength_youtube`, `sentistrength_bbc`, `sentistrength_myspace` and `vader_nyt`) and product reviews (`vader_amazon`).

Table 2 shows some statistics of the corpora used, specifically the size of each corpus (Size), the number of `Positive` (Pos.), `Negative` (Neg.) and `Neutral` (Neu.) reviews, the average number of sentences (Avg. Sent.), the average number of words (Avg. Words) and the category of the text (Genre).

As Table 2 shows, the selected corpora is very heterogeneous. `Pang_movie` and `vader_movie` contain 50% more messages than the others. `Sentistrength_bbc`, `sentistrength_digg` and `sentistrength_myspace` are those datasets with less instances. Some of the datasets are unbalanced, i.e., one of the category class is overrepresented. For example, `debate` corpus contains more negative and neutral reviews than positive, but `sentistrength_myspace` corpus contains more positives than negatives or neutrals. Finally, `pang_movie` corpus does not have any neutral reviews. The average sentence remains constant, except `sentistrength_bbc` or `sentistrength_digg` corpora that obtain higher values because comments do not have any text restrictions like tweets.

[4] http://www.ntu.edu.sg/home/epnsugan/index_files/cec-benchmarking.htm.

[5] http://www.ntu.edu.sg/home/epnsugan/index_files/cec2014/cec2014.htm.

Table 2
Summary of datasets used for our study.

Dataset	Size	Pos.	Neg.	Neu.	Avg. sent.	Avg. words	Genre
debate	3238	730	1249	1259	1.86	14.86	Micro-Blogging
english_dailabor	3771	739	488	2536	1.54	14.32	Micro-Blogging
pang_movie	10,662	5331	5331	–	1.15	18.99	Movie Reviews
sentistrength_bbc	1000	99	653	248	3.9	64.39	Forum Comments
sentistrength_digg	1077	210	572	295	2.50	33.97	Forum Comments
sentistrength_myspace	1041	702	132	207	2.22	21.12	Social Media Comments
sentistrength_twitter	4242	1340	949	1953	1.77	15.81	Micro-Blogging
sentistrength_youtube	3407	1665	767	975	1.78	17.68	Forum Comments
tweet_semevaltest	6087	2223	837	3027	1.86	20.05	Micro-Blogging
vader_amazon	3708	2128	1482	98	1.03	16.59	Product Reviews
vader_movie	10,605	5242	5326	37	1.12	19.33	Movie Reviews
vader_nyt	5190	2204	2742	274	1.01	17.76	Forum Comments
vader_twitter	4200	2897	1299	4	1.87	14.10	Micro-Blogging

4.2. Evaluation

The evaluation of our proposal entails two evaluations: (1) an evaluation of the best SAM in each dataset and the evaluation of the ensemble methods; and (2) an evaluation of the three EAs in order to choose the most suitable for developing E^{2}SAM. Since the base SAM are off-the-self classification systems, we carried out a hold-out validation approach for the first evaluation, hence we randomly split each of the 13 datasets into two subsets, 80% for the training set and 20% for the test set, keeping the proportion of classes. The base SAMs were only evaluated with the test set (20%).

Regarding the second evaluation, we followed a 5-fold cross-validation approach to evaluate the EAs. The data used for the 5-fold cross-validation was the training set (80%). However, the performance of each EA is not the average of the result of the 5-fold cross-validation, but the results reached with the best fold on the test subset (20%). For the sake of clarity, the output of each EA is a vector of weights for the function f_{e^2sam}, the final performance of each EA corresponds to the vector of weights that reached the higher results on a fold of the 5-fold cross-validation, which was used to classify the test subset.

We use three widely known evaluation measures in information retrieval and text classification [25], namely Precision, Recall and F_1 (see Eq. (8)).

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$$
$$Precision = \frac{tp}{tp + fp} \qquad Recall = \frac{tp}{tp + fn} \tag{8}$$

where *tp* is the number of instances in which the system estimation and the expert annotator agree, *fn* is the number of relevant instances for the expert annotator but not relevant for the system, and *fp* is the number of instances considered relevant by the system but not for the expert annotator.

Precision, Recall and F_1 are evaluation measures for information retrieval, so they are defined for classification problems of one class, as in information retrieval.[6] However, SA, and specifically polarity classification, may be a binary classification problem (positive/negative) or a classification task of several levels of opinion intensity. In our evaluation, we define the task of polarity classification as a three class classification problem, so each input text is classified as `Positive`, `Neutral` and `Negative`. Accordingly, we adapt F_1 to a three class classification problem, so we calculate the macro-precision and macro-recall of the precision and recall for each class and then we calculate the F_1 as in Eq. (8).

4.3. Base off-the-shelf SAMs

As we mentioned before, polarity classification may be a binary classification task {`Positive`, `Negative`} or can be defined as a classification of several levels of opinion intensity. For our evaluation, we defined the task as a three class {`Positive`, `Neutral`, `Negative`} classification problem, and formally as a regularization problem in which a polarity classification system returns a polarity value (p) that is a real value in the range [0, 1] ($p_i \in [0, 1]$). We also defined three thresholds in order to assign the final polarity class, namely `negative` $\in [0, 0.33]$, `neutral` $\in (0.33, 0.66]$ and `positive` $\in (0.66, 1]$.

In order to homogenise the output of SAMs, we defined the polarity value (p_i) as a real value from 0 to 1, $p_i \in [0, 1]$. So, we set the output of each SAM within this range as we explain in the following lines.

[6] In information retrieval, a document can be relevant or not relevant to a specific query.

Table 3
The average of F_1 value on the 30 iterations reached by each EA algorithm on the 13 datasets.

Dataset	MA	L-Shade	jSO
debate	0.443	**0.449**	0.448
english_dailabor	**0.725**	0.724	0.724
pang_movie	**0.637**	**0.637**	**0.637**
sentistrength_bbc	0.489	**0.496**	**0.496**
sentistrength_digg	0.552	0.554	**0.555**
sentistrength_myspace	**0.621**	0.610	0.608
sentistrength_twitter	**0.643**	0.637	0.640
sentistrength_youtube	0.627	**0.629**	**0.629**
tweet_semevaltest	0.643	**0.647**	0.645
vader_amazon	0.592	**0.597**	**0.597**
vader_movie	**0.565**	0.556	0.562
vader_nyt	0.543	**0.546**	0.545
vader_twitter	**0.754**	0.753	0.753
Average	**0.603**	**0.603**	**0.603**

Azure. We used the package `mscstexta4r`[7] of R to run this method. Since its output is in the range [0, 1], we did not normalize it.

CoreNLP. We used the package CoreNLP[8] of R to run this method. The output is discrete, so we had to assign to each discrete category a numerical value. Since CoreNLP returns 5 classes of polarity, from strong positive to strong negative, we transform each CoreNLP polarity class in a number value: strong negative is 0, negative is 0.25, neutral is 0.5, positive is 0.75 and strong positive is 1. Because of CoreNLP gives a sentence level output, the polarity of each input document was calculated as the average polarity of all its sentences.

MeaningCloud. It returns a discrete output, so we followed a similar approach than with CoreNLP. However, it works at document level, so we did not have to calculate the average polarity of its sentences.

SentiStrength.[9] Since it returns the positive and negative value of an input text, we calculated the final polarity value as the aggregation of both polarity values. So, an input text is negative if its polarity value is in the range [0, 0.33], and it is positive if the polarity value is in (0.66, 1].

Bing and Syuzhet. We used the `syuzhet` package[10] of R to run this method. Its output is an aggregation of the polarity of the different words of a sentence. Since it works at sentence level, the polarity of the input document is the average polarity of all its sentences, and we use the same negative and positive threshold as with SentiStrength.

Vader. We used the vader Python script[11] released by Ribeiro et al. [35] to run this method. We did a min-max normalization of its output.

4.4. Evolutionary ensembles

In order to select the most suitable EA for the development of the function f_{e^2sam}, we evaluate and compare the performance of the three EA described in Section 3.3. As we detailed in Section 4.2, (1) we performed a 5-fold cross-validation, (2) we used the weights returned by the best fold, and (3) we evaluated those weights on the test set. Since EA are probabilistic algorithms, we performed 30 iterations of the previous three steps, roughly speaking, 30 iterations of the evaluation of the EAs, therefore the results are the average of the results of the 30 iterations [15]. Table 3 displays the average F_1 of each EA proposed.

The results reached by each EA in each dataset are very similar, which is also evident in the average result of each EA. Accordingly, we used the Wilcoxon Text in order to study whether there exist significant differences between each pair of the three EAs. The test returned that there is not significant differences between the EAs with a p-value of 0.05. Consequently, we took into account the values of the signed ranks (R+ and R-) of the Wilcoxon test in order to decide what EA use. According to Table 4, the sum of the signed ranking values of jSO outperforms the signed rankings of the other two EA, therefore we selected jSO for the implementation of the function f_{e^2sam}, and thus for our proposal E^{2}SAM.

4.5. Results and analysis

Table 5 displays the results achieved by each SAM for each corpus. The results show that the SAMs trained in a similar corpus that the test data overcome the other SAMs. For instance `CoreNLP` stands out with `pang_movie` corpus and

[7] https://cran.r-project.org/web/packages/mscstexta4r/index.html.
[8] https://cran.r-project.org/web/packages/coreNLP/index.html.
[9] http://sentistrength.wlv.ac.uk/.
[10] https://cran.r-project.org/web/packages/syuzhet/vignettes/syuzhet-vignette.html.
[11] https://bitbucket.org/matheusaraujo/sentimental-analysis-methods.

Table 4
The sum of the signed rank values of the Wilcoxon test on each pair of the EAs.

	R+	R−
L-Shade vs. MA	43.5	34.5
jSO vs. L-Shade	15	13
jSO vs. MA	49.5	28.5

Table 5
F_1 results of each SAM in each corpus.

Dataset	Azure	Bing	CoreNLP	MCloud	SentiStr	Syuzhet	Vader
debate	0.436	0.458	0.400	**0.462**	0.405	0.429	0.437
english_dailabor	0.656	0.624	0.428	0.659	0.648	0.581	**0.677**
pang_movie	0.516	0.481	**0.635**	0.489	0.366	0.503	0.439
sentistrength_bbc	0.393	0.471	0.407	0.407	**0.557**	0.429	0.399
sentistrength_digg	0.464	0.490	0.446	0.501	0.518	0.494	**0.521**
sentistrength_myspace	0.512	0.449	0.410	0.536	**0.598**	0.543	0.560
sentistrength_twitter	**0.590**	0.552	0.404	0.584	0.559	0.549	0.588
sentistrength_youtube	0.564	0.540	0.493	0.584	**0.591**	0.521	0.580
tweet_semevaltest	0.521	0.565	0.416	0.594	0.575	0.542	**0.612**
vader_amazon	0.557	**0.586**	0.516	0.564	0.543	0.501	0.571
vader_movie	0.439	0.458	**0.550**	0.453	0.438	0.451	0.445
vader_nyt	0.527	0.522	0.489	**0.543**	0.481	0.526	0.533
vader_twitter	0.547	0.675	0.326	0.686	0.669	0.694	**0.748**

Table 6
F_1 reached by the baselines (AVG and NEUTY) and E^{2}SAM. For the sake of comparison, the best result achieved by an individual SAM is also showed (B. SAM). The † symbol means that the result is significant better than B. SAM according to the McNemar test (p-value < 0.01).

Dataset	AVG	NEUTY	E^{2}SAM	B. SAM	B. SAM name
debate	0.450	0.468	0.448	**0.462**	MCloud
english_dailabor	0.707	0.597	**0.724**†	0.677	Vader
pang_movie	0.462	0.527	**0.637**$^{\dagger(12)}$	0.635	CoreNLP
sentistrength_bbc	0.468	0.476	0.496	**0.557**	SentiStr
sentistrength_digg	0.536	0.550	**0.555**	0.521	Vader
sentistrength_myspace	0.529	0.568	**0.608**†	0.598	SentiStr
sentistrength_twitter	0.633	0.593	**0.640**†	0.590	Azure
sentistrength_youtube	0.627	0.586	**0.629**	0.591	SentiStr
tweet_semevaltest	0.635	0.576	**0.645**	0.612	Vader
vader_amazon	0.554	0.557	**0.597**†	0.586	Bing
vader_movie	0.527	0.497	**0.562**†	0.550	CoreNLP
vader_nyt	0.537	0.506	**0.545**†	0.543	MCloud
vader_twitter	0.699	0.589	**0.753**	0.748	Vader

`vader_movie` corpus, and `SentiStrength` with `sentistrength` corpora. If we then study the results of these SAMs on other corpora, we observe that their performance decreases. We thus conclude that SAMs' performance strongly depends on the genre and domains these SAMs have been trained with.

Table 6 shows the results of the two baselines ensemble models (AVG, NEUTY) and E^{2}SAM. For the sake of comparison, we also show the best results reached by each base SAM in each corpus extracted from Table 5 (B. SAM). First, we observe that ensemble AVG only overcomes the corresponding Best SAM in 5 corpora, and ensemble NEUTY only does in three corpora, which it was not expected because ensemble methods usually improve or match the performance of the best base system. Nevertheless, the performance of ensemble AVG on `sentistrength_digg` corpus stands out because it improves 8.16% the best SAM score (MeaningCloud).

The few cases in which the two baseline ensemble methods overcome the best SAM may mean that an ensemble approach to optimise the contribution of each SAM is needed, which is the basis of our claim. The results in Table 6 show that E^{2}SAM overcomes the best SAM and the two baselines in 11 of the 13 corpora, hence the optimisation of the contribution of a set of base SAMs in an ensemble classifier allows to improve the results with respect to the base SAMs and to the two ensemble baselines (AVG and NEUTY).

Significance analysis. We use the McNemar statistical test over the 30 iterations of each dataset against the best SAM for studying the significance difference between the results of E^{2}SAM and B. SAM. Considering a p-value < 0.01, our proposal (E^{2}SAM) significantly outperforms the best SAM in 6 datasets in more than 15 over the 30 iterations. In `pang_movie`, E^{2}SAM significantly outperforms the best SAM in 12 of the 30 iterations. Therefore, we conclude that

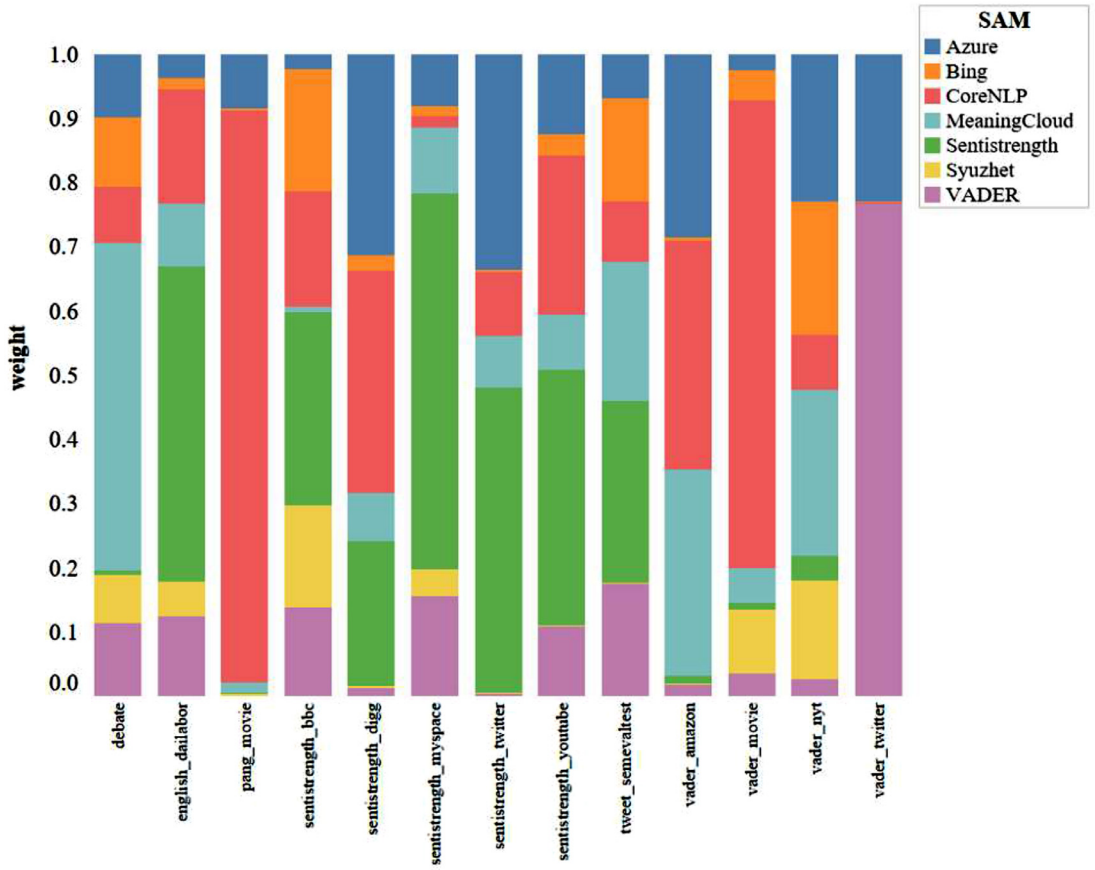

Fig. 2. Distribution of weights given by L-SHADE for each SAM in each dataset.

our proposal for optimising an ensemble method improves the performance of the base off-the-shelf SAMs, and other ensemble methods, therefore our claim holds.

Weight assignment analysis. The contribution of using an optimisation method as a EA is the automatic calculation of the relevance or weight of a base system in a ensemble method. Fig. 2 shows how E^{2}SAM assigns a weight value to each base SAM and that it is able to detect the domain of the input text because it usually gives more weight to those SAMs that were trained with a similar data than the input data. For instance, E^{2}SAM gives to CoreNLP more than 80% of weight when the evaluation data were of movie reviews domain (`pang_movie` and `vader_movie`), which is the domain used in the training of CoreNLP. The same behaviour is shown when the genre of the training and the test data matches, for instance E^{2}SAM gave more importance to the SAM Vader when the test data came from `vader_twitter` corpus. In other corpora like `sentistrength_digg`, `sentistrength_twitter` and `tweet_semevaltest`, we see that E^{2}SAM equally distributes the weights to all the base SAMs.

5. Conclusions

It is well-known that SA algorithms lack of versatility, i.e., their performance is soundly inefficient in domains which differ from their training domain. In order to address this problem, we propose to ensemble different base SAMs. We present a method built upon an evolutionary ensemble approach, E^{2}SAM, which learns the right combination of base SAMs according to the domain of the input data. We compare E^{2}SAM with the base SAMs and two ensembles as baselines.

The polarity detection results of each SAMs vary greatly, depending on the dataset used. We thus show that SAMs have a clear dependency on the domain or genre where they have been trained. So, the approach using an ensemble model is an efficient way to address this problem because it returns a concerted response.

However, the task of assigning weights is not trivial. The two baseline approaches (AVG and NEUTY) outperform the best individual SAM score in only 5 of 13 datasets. Therefore, we propose a more robust model based on a evolutionary weights optimisation method built upon jSO. The key of this ensemble is that it sets SAMs weights optimising the function with which we evaluate classification results, F_1. The aggregation guided by this model obtains the best score in 11 of the 13 datasets.

These results highlight several possible ideas for future research. Some of the datasets of this work are unbalanced, i.e., there is a polarity class that is overrepresented. We propose developing an analysis comparing the performance of the proposed ensembles with undersampling or oversampling techniques. Moreover, since the use of off-the-self SAMs implies the integration of very dissimilar systems from different perspectives, and also the use of divergent corpora, we will study the integration of other evaluation measures in the set of fitness functions of our optimisation method by using a multi-objective optimisation algorithm.

Acknowledgements

We want to take this opportunity to thank Matheus Araújo for providing us all their labelled datasets. We also thank Mike Thelwall for sharing the code of SentiStrength with us.

This research work is partially supported by the Spanish Government project TIN2017-89517-P, and a grant from the Fondo Europeo de Desarrollo Regional (FEDER). Eugenio Martínez Cámara was supported by the Spanish Government Programme Juan de la Cierva Formación (FJCI-2016-28353).

References

[1] A. Abbasi, H. Chen, A. Salem, Sentiment analysis in multiple languages: feature selection for opinion classification in web forums, ACM Trans. Inf. Syst. 26 (3) (2008) 12:1–12:34, doi:10.1145/1361684.1361685.
[2] O. Appel, F. Chiclana, J. Carter, H. Fujita, A consensus approach to the sentiment analysis problem driven by support-based iowa majority, Int. J. Intell. Syst. 32 (9) (2017) 947–965, doi:10.1002/int.21878.
[3] F.M. Plaza del Arco, M.T. Martín Valdivia, S.M. Jiménez Zafra, M.D. Molina González, E. Martínez Cámara, COPOS: corpus of patient opinions in spanish. application of sentiment analysis techniques, Procesamiento del Lenguaje Natural 57 (2016) 83–90.
[4] O.A. Arqub, Z. Abo-Hammour, Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm, Inf. Sci. 279 (2014) 396–415, doi:10.1016/j.ins.2014.03.128.
[5] M. Atzeni, A. Dridi, D. Reforgiato Recupero, Using frame-based resources for sentiment analysis within the financial domain, Progr. Artif. Intell. 7 (4) (2018) 273–294, doi:10.1007/s13748-018-0162-8.
[6] J.A. Balazs, J.D. Velásquez, Opinion mining and information fusion: a survey, Inf. Fusion 27 (2016) 95–110, doi:10.1016/j.inffus.2015.06.002.
[7] J. Blitzer, M. Dredze, F. Pereira, Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics, 2007, pp. 440–447.
[8] J. Brest, M.S. Mauec, B. Bokovi, Single objective real-parameter optimization: algorithm jso, in: 2017 IEEE Congress on Evolutionary Computation (CEC), 2017, pp. 1311–1318, doi:10.1109/CEC.2017.7969456.
[9] E. Cambria, A. Hussain, Sentic Computing, Springer Briefs in Cognitive Computation, 2, Springer Netherlands, 2012, doi:10.1007/978-94-007-5070-8.
[10] I. Chaturvedi, E. Cambria, R.E. Welsch, F. Herrera, Distinguishing between facts and opinions for sentiment analysis: Survey and challenges, Inf. Fusion 44 (2018) 65–77, doi:10.1016/j.inffus.2017.12.006.
[11] A. Das, S. Bandyopadhyay, Subjectivity detection using genetic algorithm, in: Proceedings of the 1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, 2010, pp. 14–21.
[12] A.E. Eiben, J.A. Smith, Introduction to Evolutionary Computing, Second, Springer-Verlag Berlin Heidelberg, 2015, doi:10.1007/978-3-662-44874-8.
[13] L.J. Eshelman, J.D. Schaffer, Real-coded genetic algorithms and interval-schemata, in: L.D. Whitley (Ed.), Foundations of Genetic Algorithms, Foundations of Genetic Algorithms, 2, Elsevier, 1993, pp. 187–202, doi:10.1016/B978-0-08-094832-4.50018-0.
[14] E. Fersini, E. Messina, F. Pozzi, Sentiment analysis: bayesian ensemble learning, Decis. Support Syst. 68 (2014) 26–38, doi:10.1016/j.dss.2014.10.004.
[15] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization, J. Heurist. 15 (6) (2008) 617, doi:10.1007/s10732-008-9080-4.
[16] M. Garza-Fabre, S.M. Kandathil, J. Handl, J. Knowles, S.C. Lovell, Generating, maintaining, and exploiting diversity in a memetic algorithm for protein structure prediction, Evol. Comput. 24 (4) (2016) 577–607, doi:10.1162/EVCO_a_00176.
[17] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2004, pp. 168–177.
[18] C.J. Hutto, E. Gilbert, Vader: a parsimonious rule-based model for sentiment analysis of social media text, in: Eighth International Conference on Weblogs and Social Media (ICWSM-14), 2014.
[19] J. Jiang, C. Zhai, Instance weighting for domain adaptation in nlp, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics, 2007, pp. 264–271.
[20] R. Jongeling, P. Sarkar, S. Datta, A. Serebrenik, On negative results when using sentiment analysis tools for software engineering research, Empir. Software Eng. 22 (5) (2017) 2543–2584.
[21] A. Kennedy, D. Inkpen, Sentiment classification of movie reviews using contextual valence shifters, Comput. Intell. 22 (2) (2006) 110–125, doi:10.1111/j.1467-8640.2006.00277.x.
[22] S. de Kok, L. Punt, R. van den Puttelaar, K. Ranta, K. Schouten, F. Frasincar, Review-aggregated aspect-based sentiment analysis with ontology features, Progr. Artif. Intell. 7 (4) (2018) 295–306, doi:10.1007/s13748-018-0163-7.
[23] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Cambridge University Press, 2015, doi:10.1017/CBO9781139084789.
[24] Y. Liu, Y. Zheng, Y. Liang, S. Liu, D.S. Rosenblum, Urban water quality prediction based on multi-task multi-view learning, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, in: IJCAI'16, AAAI Press, 2016, pp. 2576–2582.
[25] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008, doi:10.1017/CBO9780511809071.
[26] C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S.J. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60.
[27] E. Martínez-Cámara, Y. Gutiérrez-Vázquez, J. Fernández, A. Montejo-Ráez, R. Muñoz Guillena, Ensemble classifier for twitter sentiment analysis, in: R. Izquierdo (Ed.), Proceedings of the Workshop on NLP Applications: Completing the Puzzle co-located with the 20th International Conference on Applications of Natural Language to Information Systems (NLDB 2015), 2015, pp. 1–10.
[28] E. Martínez-Cámara, M.T. Martín-Valdivia, M.D. Molina-González, J.M. Perea-Ortega, Integrating Spanish lexical resources by meta-classifiers for polarity classification, J. Inf. Sci. 40 (4) (2014) 538–554, doi:10.1177/0165551514535710.
[29] D. Moctezuma, J. Ortiz-Bejar, E.S. Tellez, S. Miranda-Jiménez, M. Graff, INGEOTEC solution for task 1 in TASS'18 competition, in: Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN co-located with 34th SEPLN Conference (SEPLN 2018), 2018, pp. 45–49.
[30] F. Neri, C. Cotta, P. Moscato, Handbook of memetic algorithms, 379, Springer, 2012.
[31] A. Onan, S. Korukoğlu, A feature selection model based on genetic rank aggregation for text sentiment classification, J. Inf. Sci. 43 (1) (2017) 25–38, doi:10.1177/0165551515613226.
[32] S.J. Pan, X. Ni, J.-T. Sun, Q. Yang, Z. Chen, Cross-domain sentiment classification via spectral feature alignment, in: Proceedings of the 19th International Conference on World Wide Web, in: WWW '10, ACM, New York, NY, USA, 2010, pp. 751–760, doi:10.1145/1772690.1772767.

[33] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inf. Retrieval 2 (1–2) (2008) 1–135, doi:10.1561/1500000011.
[34] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: Sentiment classification using machine learning techniques, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, in: EMNLP '02, Association for Computational Linguistics, 2002, pp. 79–86, doi:10.3115/1118693.1118704.
[35] F.N. Ribeiro, M. Araújo, P. Gonçalves, M.A. Gonçalves, F. Benevenuto, Sentibench - a benchmark comparison of state-of-the-practice sentiment analysis methods, EPJ Data Sci. 5 (1) (2016) 1–29.
[36] L. Rokach, Ensemble methods for classifiers, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, Springer US, Boston, MA, 2005, pp. 957–980, doi:10.1007/0-387-25465-X_45.
[37] J.E. Rowe, Genetic algorithms, in: J. Kacprzyk, W. Pedrycz (Eds.), Springer Handbook of Computational Intelligence, Springer Berlin Heidelberg, 2015, pp. 825–844, doi:10.1007/978-3-662-43505-2_42.
[38] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 3rd, Prentice Hall Press, 2009.
[39] N.F. da Silva, E.R. Hruschka, E.R. Hruschka, Tweet sentiment analysis with classifier ensembles, Decis. Support Syst. 66 (2014) 170–179, doi:10.1016/j.dss.2014.07.003.
[40] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.
[41] S. Sun, C. Luo, J. Chen, A review of natural language processing techniques for opinion mining systems, Inf. Fusion 36 (2017) 10–25, doi:10.1016/j.inffus.2016.10.004.
[42] R. Tanabe, A.S. Fukunaga, Improving the search performance of shade using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1658–1665, doi:10.1109/CEC.2014.6900380.
[43] M. Thelwall, K. Buckley, G. Paltoglou, Sentiment strength detection for the social web, J. Am. Soc. Inf. Sci. Technol. 63 (1) (2012) 163–173, doi:10.1002/asi.21662.
[44] P.D. Turney, 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, in: ACL '02, 2002, pp. 417–424, doi:10.3115/1073083.1073153.
[45] A. Valdivia, M.V. Luzón, E. Cambria, F. Herrera, Consensus vote models for detecting and filtering neutrality in sentiment analysis, Inf. Fusion 44 (2018) 126–135, doi:10.1016/j.inffus.2018.03.007.
[46] A. Valdivia, M.V. Luzón, F. Herrera, Neutrality in the sentiment analysis problem based on fuzzy majority, in: 2017 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2017, Naples, Italy, July 9–12, 2017, IEEE, 2017, pp. 1–6, doi:10.1109/FUZZ-IEEE.2017.8015751.
[47] X. Wan, Using bilingual knowledge and ensemble techniques for unsupervised chinese sentiment analysis, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, in: EMNLP '08, 2008, pp. 553–561.
[48] M. Woźniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fusion 16 (2014) 3–17, doi:10.1016/j.inffus.2013.04.006. Special Issue on Information Fusion in Hybrid Intelligent Fusion Systems.
[49] R. Xia, C. Zong, X. Hu, E. Cambria, Feature ensemble plus sample selection: domain adaptation for sentiment classification, IEEE Intell. Syst. 28 (3) (2013) 10–18, doi:10.1109/MIS.2013.27.

Miguel López studied Data Science's M.Sc. at University of Granada and received a B.Sc. degree in Computer Science from the same university. Currently, he works as researcher at the University of Granada and he is specially interested on Natural Language Processing and Deep Learning.

Ana Valdivia obtained her B.Sc. in Mathematics from Polytechnic University of Catalonia (UPC) in 2014. She then worked as a research assistant at IESE Business School of Barcelona in 2015. She received her M.Sc. in Data Science and Computer Engineering from University of Granada (UGR) in 2016. Now, she is currently pursuing the Ph.D. degree in Computer Science and Artificial Intelligence in UGR. Her research is focused on Natural Language Processing, Sentiment Analysis and Deep Learning. She is also a strong advocate for Data Science for Social Good.

Eugenio Martínez-Cámara is a postdoctoral researcher at University of Granada, Spain. He received a B.Sc. degree in Computer Science and Management and M.Sc. degree in Computer Science from the University of Jaén, Spain, in 2008 and 2010, respectively. He received his Ph.D. in Computer Science in 2015 at the University of Jaén. Dr. Martínez-Cámara also worked as postdoctoral researcher at Technische Universität Darmstadt, Germany. His current research interest are sentiment analysis, information extraction and the use of deep learning in natural language processing.

M. Victoria Luzón is an associate professor in the Software Engineering Department at University of Granada. Her research interests include sentiment analysis, artificial intelligence, computer graphics and cultural heritage. Luzón has a Ph.D. in Industrial Engineering from the University of Vigo, Spain. Contact her at luzon@ugr.es.

Francisco Herrera received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991 (University of Granada, Spain). He is currently a professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. Prof. Herrera has supervised 42 Ph.D. thesis and published more than 380 journal papers (h-index = 121, Scholar Google). He currently acts as Editor in Chief of the international journals "Information Fusion" (Elsevier) and "Progress in Artificial Intelligence" (Springer). He has been selected as a Highly Cited Researcher http://highlycited.com/ (fields of Computer Science and Engineering, respectively, 2014 to present, Clarivate Analytics). His current research interests include soft computing (fuzzy modeling, evolutionary algorithms and deep learning), computing with words, information fusion, and data science (data preprocessing, prediction and big data).