Information Fusion 44 (2018) 126–135

Contents lists available at ScienceDirect

Information Fusion

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

Consensus vote models for detecting and filtering neutrality in sentiment analysis

Ana Valdivia[*,a], M. Victoria Luzón[b], Erik Cambria[c], Francisco Herrera[a]

[a] Department of Computer Science & Artificial Intelligence, Universidad de Granada, Spain
[b] Department of Software Engineering, Universidad de Granada, Spain
[c] School of Computer Science and Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Keywords:
Sentiment analysis
Neutrality
Filtering
Consensus
Aggregation

ABSTRACT

Recently, interest in sentiment analysis has grown exponentially. Many studies have developed a wide variety of algorithms capable of classifying texts according to the sentiment conveyed in them. Such sentiment is usually expressed as positive, neutral or negative. However, neutral reviews are often ignored in many sentiment analysis problems because of their ambiguity and lack of information. In this paper, we propose to empower neutrality by characterizing the boundary between positive and negative reviews, with the goal of improving the model's performance. We apply different sentiment analysis methods to different corpora extracting their sentiment and, hence, detecting neutral reviews by consensus to filter them, i.e., taking into account different models based on weighted aggregation. We finally compare classification performance on single and aggregated models. The results clearly show that aggregation methods outperform single models in most cases, which led us to conclude that neutrality is key for distinguishing between positive and negative and, then, for improving sentiment classification.

1. Introduction

Over the past few decades, the amount of social media data (e.g., reviews, opinions or posts) stored in the Web 2.0 has grown exponentially. This type of website consists of social media platforms (e.g., Blogger and TripAdvisor), social networks (e.g., Facebook and Twitter) and photo, audio or video portal hosting (e.g., Instagram and YouTube). The essence of such tools is the possibility to interact with other users or provide content that enriches the browsing experience.

Sentiment analysis has emerged as a new tool for analyzing Web 2.0 information [1–5]. It is a branch of affective computing research [6] that aims to classify text (but sometimes also audio and video [7]) as either positive or negative.

The main aim of sentiment analysis is to systematically analyze people's opinion on a product, organization, or event [2]. Hence, its most important goal is Sentiment Analysis Classification (SAC), i.e., determining whether an opinion, sentence, or aspect expresses a *positive, neutral*, or *negative* sentiment orientation. Because of the many possible applications and domains of sentiment analysis, different sentiment analysis methods (SAMs) have been developed to address SAC [8,9].

Usually, in many SAC models, neutral reviews are not considered [10,11]. There are two main reasons for this: 1) most SAC models focus on binary classification, i.e., in the identification of positive versus negative opinions; 2) neutral reviews lack information due to their ambiguity. However, we consider that the neutral class is key for improving sentiment classification performance [10]. Since neutrality is considered somewhere between positivity and negativity, the idea is to deal with it as potential noise, i.e., from a noise filtering classification point of view [12]. It is understood that detecting and removing noise can improve a model's performance.

In this paper, we assume that neutral opinions must be detected and filtered to improve binary polarity classification. We claim that there is a lack of agreement among SAMs for detecting neutral opinions. So, there is a need to develop a consensus model for improving the identification of neutrality.

Our proposal is to detect neutrality guided by consensus voting among SAMs, and to filter it before the opinion classification step. We first present a neutrality proximity function that assigns weights to polarities according to its proximity to the neutral point. We then propose two polarity aggregation models based on a Weighting Average using the proximity function and on Induced Ordered Weighted Averaging (IOWA) guided by linguistic quantifiers to represent the majority concept, respectively. The main idea is to obtain polarities

[*] Corresponding author.
E-mail address: avaldivia@ugr.es (A. Valdivia).

https://doi.org/10.1016/j.inffus.2018.03.007
Received 30 October 2017; Received in revised form 24 March 2018; Accepted 31 March 2018
Available online 31 March 2018

from several SAMs and aggregate them based on those aggregation models designed using the neutrality proximity function.

We consider an experimental framework with 9 different context datasets and 6 off-the-shelf SAMS. We compute the aggregation polarities and filter out neutral reviews. After that, we develop a SAC task with positive and negative polarities, extracting unigram features and applying two machine learning algorithms. We finally compare and analyze the results and conclude that the polarity consensus voting models, together with neutrality filtering, outperform SAC results.

The paper is structured as follows: we first describe the sentiment analysis problem in Section 2; we define the model for detecting neutrality based on consensus voting in Section 3; we present the experiment setup and the results in Section 4; we draw conclusions and conclude the paper in Section 5.

2. Sentiment analysis

In this section, we describe the main concepts of sentiment analysis (Section 2.1) and the SAMs that we apply (Section 2.2).

2.1. The sentiment analysis problem

Due to the increasing number of online reviews, sentiment analysis has emerged as a new field for analyzing this amount of data [13]. It aims is to analyze sentiments in written text [2]. The number of possible applications is very broad [14]: business intelligence (analyze customer's reviews towards a product) [15], politics (predict election results mining social opinions) [16–18], tourism [19,20], personality recognition [21] or social studies (evaluate the level of sexist messages in social networks or detect cyberbullying) [22–24]. While most studies approach it as a simple categorization problem, sentiment analysis is actually a 'suitcase' research problem that requires tackling many NLP sub-tasks such as:

Sentiment Analysis Classification: This is the most popular task. The aim of SAC is to develop models capable of detecting sentiment in texts. The first step is to collect text or reviews to set our analysis. After that, the sentiment is detected. It can be computed by the reviewer or computed with SAMs. Then, features are selected to train the classification model. In this step, text mining techniques are commonly used to extract the most significant features.

Subjectivity Detection: This task is related to SAC in the sense that the objective is to classify subjective and objective opinions. The purpose is to filter subjective sentences because they are more opinionated and, hence, can improve classification models.

Opinion Summarization: Also known as aspect-based summary or feature-based summary. It consists of developing techniques to sum up large amounts of reviews written by people. The summarization should focus on entities or aspects and their sentiment and should be quantitative [25,26].

Opinion Retrieval: This is a retrieval process, which requires documents to be retrieved and ranked according to their relevance.

Sarcasm and Irony: This task aims to detect opinions with sarcastic or ironic expressions. As in subjectivity detection, the target is to delete these opinions from the sentiment analysis process [27,28].

Others: Due to the fact that sentiment analysis is a growing research branch, over recent years many new tasks have emerged, e.g, temporal tagging [29], word polarity disambiguation [30].

As we have previously stated, SAC is a very important task in sentiment analysis. These models classify texts according to their sentiment. This sentiment can be identified in different ways: label polarity ({positive, neutral, negative}), numerical rating ({0, 1, ... , 4} or [0,1]) or emotions {anger, disgust, fear, happiness, sadness, surprise}.

There are three different levels of analysis in this problem:

- The *document level* extracts the sentiment of the whole opinion. This is considered to be the simplest task.
- The *sentence level* extracts sentiments in each sentence of the text. This level is highly related to classifying subjective and objective sentences.
- The *aspect level* is the fine-grained level. This is the most challenging analysis because it extracts sentiments with respect to each opinion target.

2.2. Sentiment analysis methods

The main task of sentiment analysis is to detect polarity within a text. Therefore, multiple SAMs have been developed to automatically address this challenge. These methods are considered to be varied due to the different properties of online reviews (short texts like tweets, long reviews of microblogs, texts with emoticons, etc.). There are different studies that analyze and compare a large variety of these tools [8,9]. These SAMs are mainly classified into three groups [31]:

- **Lexicon-Dictionary Based Method (LD):** This method relies on a sentiment dictionary which contains words denoting a sentiment. This dictionary is built from seed words (contained in the corpus or not) and it is then extended with synonyms and antonyms from those seed words [32].
- **Machine Learning Based Method (ML):** The main idea is to develop classification models to evaluate new opinions. The classifier algorithm is trained and validated with labeled opinions [33].
- **Hybrid Based Method (LD & ML):** The hybrid method consists of a mixture of both methods, LD and ML [34].

Table 1shows an overview of the main characteristics of those used in this study and they are introduced in the following subsections.

2.2.1. Bing

This method is considered one of the first LD methods. It was developed by Hu and Liu [26]. They took a number of seed adjectives and then developed this dictionary with WordNet [40]. It contains around 6800 words with its orientation. This method scores sentences with -1 (*negative*), 0 (*neutral*) or 1 (*positive*).

2.2.2. CoreNLP

This method was developed by the Stanford NLP group. They introduce in [41] a deep learning method, Recursive Neural Tensor Network (RNTN), trained with 215,154 labeled sentences. One of the main contributions of this study is the introduction of a Sentiment Treebank capable of detecting the compositional effects of sentiments in language, such as negations. CoreNLP outperforms sentiment sentence classification improving by 80.7 This algorithm scores sentence sentiments with a discrete scale from 0 (*very negative*) to 4 (*very positive*).

2.2.3. MeaningCloud

It is a ML method that performs a detailed multilingual sentiment analysis of texts from different sources [37]. The text provided is analyzed to determine if it expresses a positive, negative or neutral

Table 1
Summary of 5 popular SAMs.

SAM	Type	Output	Reference
Bing	LD	$\{-1, 0, 1\}$	[26]
VADER	LD	$[-1, 1] \in \mathbb{R}$	[35]
CoreNLP	ML	$\{0, 1, 2, 3, 4\}$	[36]
MeaningCloud	ML	$[0, 1] \in \mathbb{R}$	[37]
Microsoft Azure	LD & ML	$[0, 1] \in \mathbb{R}$	[38]
SentiStrength	LD & ML	$\{-1, 0, 1\}$	[39]

sentiment. To this end, the local polarity of the different sentences in the text is identified and the relationship between them evaluated, resulting in a global polarity value for the whole text. Besides polarity at sentence and document level, MeaningCloud uses advanced NLP techniques to detect the polarity attributed to entities and concepts from the text. It provides a reference in the relevant sentence and a list of elements detected with the aggregated polarity derived from all their appearances, also taking into account the grammatical structures in which they are contained.

2.2.4. *Microsoft Azure*

It is a NLP web service developed by Microsoft Corporation and integrated into Azure Machine Learning toolkit [38]. This API analyzes unstructured text for many NLP tasks. The sentiment analysis task was built as a mix of LD and ML and it was trained for sentiment classification using Sentiment140 data [42]. It scores close to 0 indicating negative sentiment and close to 1 indicating positive sentiment.

2.2.5. *SentiStrength*

It estimates the strength of positive and negative sentiment in short texts, even for informal language. It has human-level accuracy for short social web texts in English, except political texts [39,43]. It builds a lexicon dictionary annotated by humans and improved with the use of machine learning. SentiStrength reports two sentiment strengths, −1 (not negative) to −5 (extremely negative) and 1 (not positive) to 5 (extremely positive). It uses two set scores because psychological research has revealed that humans process simultaneously positive and negative sentiments.

2.2.6. *VADER*

It is a human-validated SAM developed for twitter and social media contexts. VADER was created from a generalizable, balanced-based, human-curated gold standard sentiment lexicon, [35]. It combines a lexicon and the processing of the sentence characteristics to determine a sentence polarity. VADER's author identified five heuristics based on grammatical and syntactical cues to convey changes to sentiment intensity that go beyond the bag-of-words model.

3. Neutrality detection based on consensus vote

SAMs are methods trained from different texts. But there are many types of texts: short, long, expressing opinions, objectives, etc. This makes the behavior of SAMs very diverse, and there is a lack of consensus when it comes to detecting polarities, in particular neutrality, as we will demonstrate later in this section. To address this problem, we propose ensembling the different polarities in order to reach a consensus.

Firstly, we explain the main facts that led us to propose an aggregation system for detecting neutrality based on, together with the polarities, aggregation and neutrality filtering for SAC (Section 3.1). We then present a proximity function for neutrality detection (Section 3.2). We describe the two main consensus voting models, the first considers on weighted aggregation based on a proximity function to the neutrality and the second one uses IOWA operators with weights based on a fuzzy majority guided by linguistic quantifiers (Section 3.3).

3.1. *Motivation: the global process of SAMs aggregation for SAC*

There is a relation between *subjectivity* and *neutrality* which is not clear in the sentiment analysis literature. A *subjective* sentence is defined as the absence of factual material which implies a certain amount of opinion or sentiment that comes from the issuer [44]. Liu argues that there are *subjective* sentences that may express objective information [2].

Neutrality means the absence of sentiment or no sentiment [2]. However, we think that opinions expressing mixed or conflicting sentiment may also be considered as *neutral. Neutral* reviews show an ambiguous weight of sentiment, i.e., contain an equitable burden of positive and negative polarity.

Due to this fact, this class has been considered as noisy and is broadly excluded in many sentiment models [45–47]. However, some researchers have tackled the problem of classifying it: the authors in [10] propose taking into account neutral reviews in order to improve classification results or authors in [48] propose a multi-sentiment scale, 1–5 stars, to solve the problem of a wider range of sentiment representation, including neutral reviews. In this direction, we propose detecting neutrality by obtaining the votes of several SAMs. We then filter these polarities to improve classification results. The SAC models can improve their polarity classification.

In this study, we consider that opinions can be labeled as *positive, negative* or *neutral. Neutral* opinions define a threshold between *positive* and *negative*.

We present the notation used in this paper:

- $M = \{m_1, \cdots, m_T\}$ the set of SAMs.
- $O = \{o_1, \cdots, o_N\}$ the set of opinions.
- p_{ik} is the normalized value of the ith SAM on the kth opinion, i.e., $p_{ik} \in [0, 1]$.
- $I = (e, 1-e)$ where $e \in [0, 0.5]$ as the *Neutral Interval*.

We thus define the *Sentiment Scale* (Fig. 1) which is a numeric scale, from 0 to 1, divided into three chunks: the *negative*, the *neutral* and the *positive*.

Following these considerations, we claim that there exists a very low agreement on detecting neutral opinions. To discuss this assumption, we apply 6 different SAMs (introduced in Section 2.2) on 9 datasets from different context and domain. Then, we count the total number of neutral opinions detected by each SAM on each corpus.

As we can observe in Table 2, there are only 0.33% of reviews where all SAMs agree on detecting neutrality in all the datasets. However, in a 78.64% of reviews, at least one SAM obtains neutral polarities. This fact has led us to conclude that our claim holds. There is a need for developing consensus models to detect neutrality, filter them and enhance sentiment classification. To summarize our proposal graphically, the Fig. 2 shows a flowchart that presents the global process of SAMs aggregation for SAC. The first step is to collect opinions, then we apply a total number of T SAMs to these opinions. After that, we extract the consensual polarity applying our models. Filtering out neutral reviews and applying text mining techniques for aspect extraction are the next steps. Finally, we classify polarity labels.

3.2. *Neutral proximity function*

In order to measure the proximity to the neutral point, we propose the Neutral Proximity Function (NPF).

Definition 1 (*Neutral Proximity Function*). The NPF is a function that measures the proximity of polarity p_{ik} to the neutrality, rising its absolute maximum when $p_{ik} = 0.5$.

We propose to use the following parametric function of NPF with $\alpha \in (0, 2]$:

$$\begin{aligned} NPF_\alpha \colon [0,1] &\to [0,1] \\ p_{ik} &\mapsto 1-\alpha|p_{ik}-0.5|,\ \alpha \in (0,2]. \end{aligned}$$

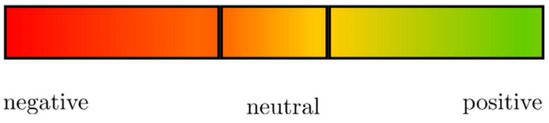

Fig. 1. The Sentiment Scale. *Negative* opinions score from 0 to *e*, *neutral* from *e* to 1 − *e* and *positive* from 1 − *e* to 1, *e* ∈ [0, 0.5].

Table 2
Number of Neutrality Consensus per Corpus.

Corpus (500 reviews per Corpus)	AllAgree	AtLeastOneAgree
Amazon	3	404
ClintonTrump	9	433
Food	0	422
Cinema	3	377
Movies	0	357
RW	0	441
Ted	0	388
TA-Sagrada Familia	0	348
TA-Alhambra	0	369

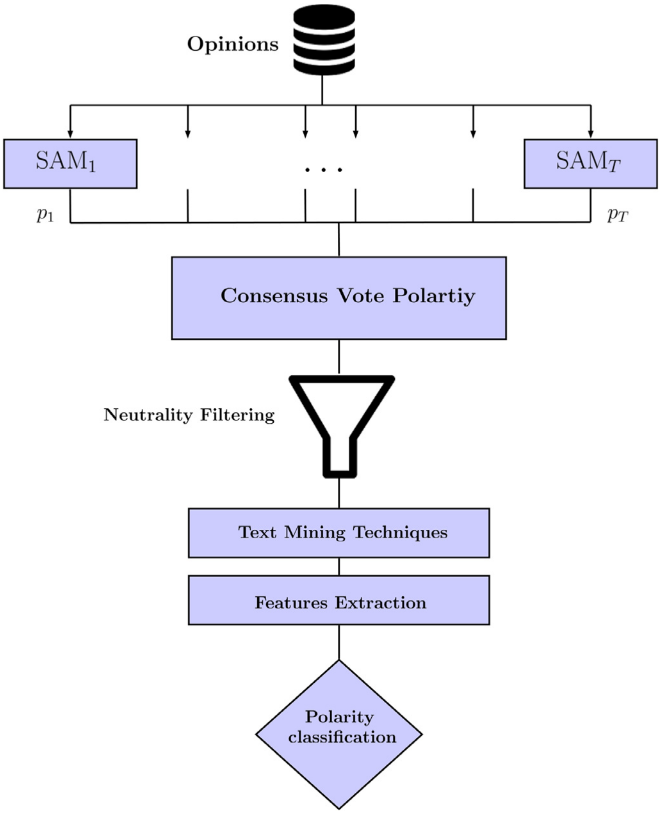

Fig. 2. Flowchart of the global process of SAMs aggregation for SAC.

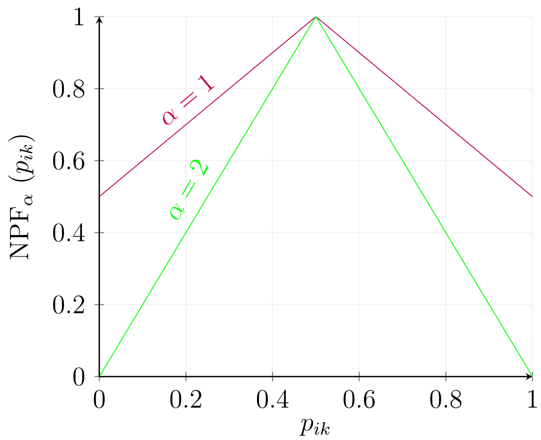

Fig. 3. Representation of NPF_1 and NPF_2. If a polarity is very negative or very positive, the function gets the minimum value. If a polarity p_{ik} is close to the *Neutral Interval*, the function rises the maximum value, which is always 1 ($\mathrm{NPF}_\alpha(0.5) = \max(\mathrm{NPF}_\alpha(p_{ik})) = 1$).

The value α is used in NPF_α to scale the proximity values. Fig. 3 shows two cases of NPF for $\alpha = 1$ and $\alpha = 2$. As we observe, if a polarity is very negative or very positive ($p_{ik} \approx 0$ or $p_{ik} \approx 1$, respectively) both functions obtain values close to 0. Otherwise, if a polarity is neutral ($p_{ik} \approx 0.5$), they get values close to 1. NPF_α always reaches the absolute minimum when $p_{ik} = 0$ or $p_{ik} = 1$ and the absolute maximum when $p_{ik} = 0.5$. So, it clearly models the proximity of polarities to the *Neutral Interval*, the closer the polarity is, the more weight it gets.

3.3. Neutrality detection weighting aggregation

In this section we propose different aggregation models based on weights. The first ones are guided by the NPF (defined before) and the second ones by ordered weights averaging.

3.3.1. Weighting aggregation based on a proximity function

We propose two average weighting models based on the proximity function (NPF_α) to the neutral point, for detecting neutral reviews by consensus. The aggregated polarities are guided by this function. Thus, the aggregated polarity shows consensus on detecting neutrality if its value belongs to the neutrality interval.

Definition 2 (*Pro-Neutrality Weight Based Model (ProN)*). The *ProN*, Φ_{ProN}, is an aggregation model for sentiment polarities which defines the weighting vector W guided by the $\mathrm{NPF}_{\alpha=1}$, i.e., $w_{ik}= \mathrm{NPF}_{\alpha=1}(p_{ik}) = 1 - |p_{ik} - 0.5|$ and it is expressed such that:

$$\Phi_{ProN}\colon [0,1]^T \to [0,1]$$
$$(p_{1k},\cdots,p_{Tk}) \mapsto \sum_{i=1}^{T} \frac{w_{ik}}{\sum_{i=1}^{T} w_{ik}} p_{ik}.$$

Therefore, the ensembled polarity of an opinion o_k is expressed by:

$$\begin{aligned}\Phi_{ProN}((p_{1k},\cdots,p_{Tk})) &= \sum_{i=1}^{T} \frac{w_{ik}}{\sum_{h=1}^{T} w_{ik}} p_{ik}\\ &= \sum_{i=1}^{T} \frac{NPF_{\alpha=1}(p_{ik})}{\sum_{i=1}^{T} NPF_{\alpha=1}(p_{ik})} p_{ik}\\ &= \sum_{i=1}^{T} \frac{1 - |p_{ik} - 0.5|}{\sum_{i=1}^{T} 1 - |p_{ik} - 0.5|} p_{ik}.\end{aligned}$$

Definition 3 (*Pro-Neutrality Extreme Weight Based Model (ProNE)*). The *ProNE*, Φ_{ProNE}, is an aggregation model for sentiment polarities which defines the weighting vector W guided by the $\mathrm{NPF}_{\alpha=2}$, i.e., $w_{ik}= \mathrm{NPF}_{\alpha=2}(p_{ik}) = 1 - 2|p_{ik} - 0.5|$ and it is expressed such that:

$$\Phi_{ProNE}\colon [0,1]^k \to [0,1]$$
$$(p_{1k},\cdots,p_{Tk}) \mapsto \sum_{h=1}^{T} \frac{w_{ik}}{\sum_{h=1}^{T} w_{ik}} p_{ik}.$$

Therefore, the ensembled polarity of an opinion o_k is expressed by:

$$\begin{aligned}\Phi_{ProNE}((p_{1k},\cdots,p_{Tk})) &= \sum_{h=1}^{T} \frac{w_{ik}}{\sum_{h=1}^{T} w_{ik}} p_{ik}\\ &= \sum_{h=1}^{T} \frac{NPF_{\alpha=2}(p_{ik})}{\sum_{h=1}^{T} NPF_{\alpha=2}(p_{ik})} p_{ik}\\ &= \sum_{h=1}^{T} \frac{1 - 2|p_{ik} - 0.5|}{\sum_{h=1}^{T} 1 - 2|p_{ik} - 0.5|} p_{ik}.\end{aligned}$$

As reference for experimental analysis, we consider the basic model which averages polarities and give them an equal weight.

Definition 4 (*Average Based Model (AVG)*). The *AVG* is an aggregation model for sentiment polarities which defines the weighting vector by $W = \frac{1}{T}$ and it is expressed such that:

$$\begin{aligned}\Phi_{AVG}\colon [0,1]^T &\to [0,1]\\ (p_{1k},\cdots,p_{Tk}) &\mapsto \frac{1}{T}\sum_{h=1}^{T} p_{ik}.\end{aligned}$$

Note that this model is equivalent to the *arithmetic mean* over the k polarities.

3.3.2. Aggregation based on majority vote guided by linguistic quantifiers

In many decision-making problems, the opinion of the majority of agents is the relevant output [49]. Yager proposed the *Ordered Weighted Averaging (OWA) operator* modelling the *fuzzy majority*, i.e., the idea that a decision will be made if most of the agents agree [50,51]. Soon after, the same author proposed an OWA operator but induced the order of the argument variable via an order-induced vector, the *Induced Ordered Weighted Averaging (IOWA) operator*. The IOWA operator is considered a generalization of OWA operators with a specific semantic in the aggregation process [50–52]. Recently, IOWA operators have been used for sentiment classification using the vote of majority for classifiers aggregation [53].

Definition 5 (*OWA [50,54]*). An *OWA operator* of dimension n is a mapping $\phi\colon \mathbb{R}^n \to \mathbb{R}$ that has an associated weighting vector W such that $w_i \in [0,1]$, $\sum_{i=1}^{n} w_i = 1$, and is defined to aggregate a list of values $\{p_1, \ldots, p_n\}$ following this expression:

$$\phi(p_1,\ldots,p_n) = \sum_{i=1}^{n} w_i p_{\sigma(i)},$$

being $\sigma\colon \{1,\ldots,n\} \to \{1,\ldots,n\}$ a permutation such that $p_{\sigma(i)} \geq p_{\sigma(i+1)}$, $\forall\, i = 1,\ldots,n-1$.

Definition 6 (*IOWA [52,54]*). An *IOWA operator* of dimension n is a mapping $\Psi\colon (\mathbb{R}\times\mathbb{R})^n \to \mathbb{R}$ that has an associated weighting vector W such that $w_i \in [0,1]$, $\sum_{i=1}^{n} w_i = 1$, and it is defined to aggregate the set of second arguments of a list of n 2-tuples:

$$\Psi(\langle u_1, p_1\rangle,\ldots,\langle u_n, p_n\rangle) = \sum_{i=1}^{n} w_i p_{\sigma(i)},$$

being $\sigma\colon \{1,\ldots,n\} \to \{1,\ldots,n\}$ a permutation such that $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall\, i = 1,\ldots,n-1$.The vector of values $U = (u_1,\ldots,u_n)$ is defined as the *order-inducing* vector and $(p_1,\ldots,p_n)$ as the *values of the argument variable*. In this way, the *order-inducing* reorders the *values of the argument variable* based on its magnitude.

Linguistic quantifiers are widely used for modeling the concept of quantification to represent the fuzzy majority [55]. *At least half, Most of* and *Many as possible* are some examples of these quantifiers (see Fig. 4), which can be modeled explicitly as a fuzzy set by the following function proposed by Yager in [51]. We propose to use these operators because they are aligned to the idea that we are considering for aggregating polarities [53].

$$Q_{(a,b)}(x) = \begin{cases} 0 & \text{if } 0 \leq x < a,\\ fracx-ab-a & \text{if } a \leq x \leq b,\\ 1 & \text{if } b \leq x \leq 1\end{cases}$$

The values that are used for the pair (a, b) are [56]:

$$Q_{Atleasthalf}(x) = Q_{(0,0.5)}(x)$$

$$Q_{Mostof}(x) = Q_{(0.3,0.8)}(x)$$

$$Q_{Manyaspossible}(x) = Q_{(0.5,1)}(x)$$

Then, the weights are calculated as follows:

$$w_i^{(a,b)} = Q_{(a,b)}\left(\frac{i}{T}\right) - Q_{(a,b)}\left(\frac{i-1}{T}\right)$$

We define the following IOWA based models taking into account linguistic quantifiers and the NPF$_\alpha$ to tackle the consensus voting among SAMs based on majority:

Definition 7 (*IOWA At Least Half Pro-Neutrality System Based (ALH-ProN)*). The *IOWA ALH-ProN operator* of dimension T is a mapping $\Psi_{ALH-ProN}\colon ([0,1]\times[0,1])^T \to [0,1]$ that has an associated weighting vector W such that $w_i^{(0,0.5)}$ and it is defined to aggregate the set of second arguments of a list of T 2-tuples:

$$\Psi_{ALH-ProN}(\langle u_1, p_{1k}\rangle,\ldots,\langle u_T, p_{Tk}\rangle) = \sum_{i=1}^{T} w_i^{(0,0.5)} p_{\sigma(i)k},$$

being $\sigma\colon \{1,\ldots,T\} \to \{1,\ldots,T\}$ a permutation such that $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall\, i = 1,\ldots,T-1$, and $u_i = NPF_{\alpha=1}(p_{ik}) = 1 - |p_{ik} - 0.5|$.

Definition 8 (*IOWA Most Of Pro-Neutrality System Based (MO-ProN)*). The *IOWA MO-ProN operator* of dimension T is a mapping $\Psi_{MO-ProN}\colon ([0,1]\times[0,1])^T \to [0,1]$ that has an associated weighting vector W such that $w_i^{(0.3,0.8)}$ and it is defined to aggregate the set of second arguments of a list of T 2-tuples:

$$\Psi_{MO-ProN}(\langle u_1, p_{1k}\rangle,\ldots,\langle u_T, p_{Tk}\rangle) = \sum_{i=1}^{T} w_i^{(0.3,0.8)} p_{\sigma(i)k},$$

being $\sigma\colon \{1,\ldots,T\} \to \{1,\ldots,T\}$ a permutation such that $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall\, i = 1,\ldots,T-1$, and $u_i = NPF_{\alpha=1}(p_{ik}) = 1 - |p_{ik} - 0.5|$.

Definition 9 (*IOWA Many As Possible Pro-Neutrality System Based (MAP-ProN)*). The *IOWA MAP-ProN operator* of dimension T is a mapping $\Psi_{MAP-ProN}\colon ([0,1]\times[0,1])^T \to [0,1]$ that has an associated weighting vector W such that $w_i^{(0.5,1)}$ and it is defined to aggregate the set of second arguments of a list of T 2-tuples:

$$\Psi_{MAP-ProN}(\langle u_1, p_{1k}\rangle,\ldots,\langle u_T, p_{Tk}\rangle) = \sum_{i=1}^{T} w_i^{(0.5,1)} p_{\sigma(i)k},$$

being $\sigma\colon \{1,\ldots,T\} \to \{1,\ldots,T\}$ a permutation such that $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall\, i = 1,\ldots,T-1$, and $u_i = NPF_{\alpha=1}(p_{ik}) = 1 - |p_{ik} - 0.5|$.

Note that in these operators, the neutrality proximity function (see

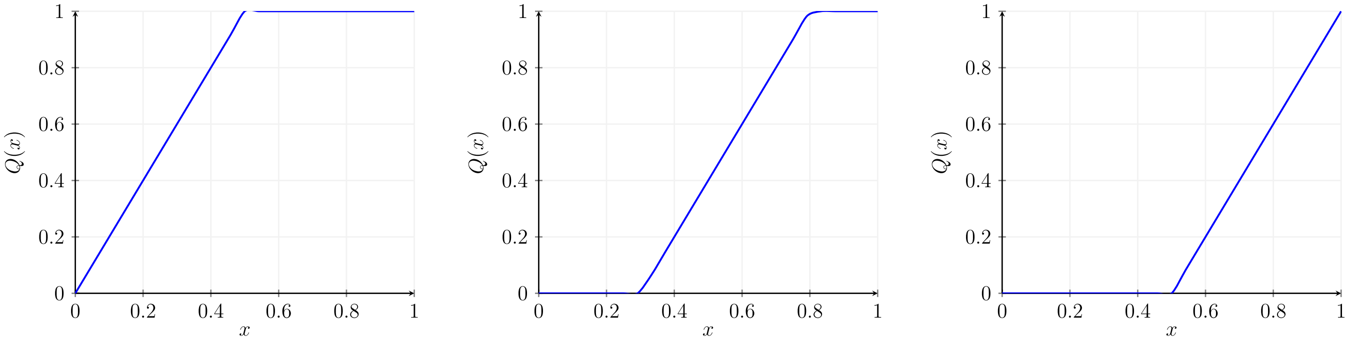

Fig. 4. Linguistic Quantifiers Represented as Fuzzy Sets: *At least half, Most of* and *Many as possible*, respectively.

Table 3
Summary of Quantitative Text Analysis of Datasets (Words and Sentences).

Corpus	NumWords	AVGNumWords	NumSentences	AVGNumSentences
Amazon	7787	15.57	500	1.00
ClintonTrump	8674	17.35	881	1.76
Food	40,775	81.55	2512	5.02
Cinema	10,433	20.87	564	1.13
Movies	9623	19.25	523	1.05
RW	34,871	69.74	2337	4.67
Ted	8971	17.94	502	1.00
TA-Sagrada Familia	30,520	61.04	2033	4.07
TA-Alhambra	45,665	91.33	2800	5.60

Figure 3) sorts polarities and linguistic quantifiers (see Fig. 4) provide weights.

Finally, we introduce two particular cases of IOWA. They induce weights taking into account the minimum and maximum extreme polarity. More precisely:

Definition 10 (*IOWA Minimum Neutrality (MinN)*). The *IOWA MinN* of dimension T is a mapping Ψ_{MinN}: $([0, 1] \times [0, 1])^T \rightarrow [0, 1]$ that has an associated weighting vector W and it is defined to aggregate the set of second arguments of a list of T 2-tuples:

$$\Psi_{MinN}(\langle u_1, p_{1k}\rangle, \ldots, \langle u_T, p_{Tk}\rangle) = \sum_{i=1}^{T} w_i p_{\sigma(i)k},$$

being σ: $\{1, \ldots, T\} \rightarrow \{1, \ldots, T\}$ a permutation such that $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall\ h = 1, \ldots, T-1$, and $u_i = NPF_{\alpha=1}(p_{ik}) = 1 - |p_{ik} - 0.5|$ with $w_T = 1$ and $w_i = 0$ for $\forall\ i = 1, \ldots, T-1$.

Definition 11 (*IOWA Maximum Neutrality (MaxN)*). The *IOWA MaxN* of dimension T is a mapping Ψ_{MaxN}: $([0, 1] \times [0, 1])^T \rightarrow [0, 1]$ that has an associated weighting vector W and it is defined to aggregate the set of second arguments of a list of T 2-tuples:

$$\Psi_{MaxN}(\langle u_1, p_{1k}\rangle, \ldots, \langle u_T, p_{Tk}\rangle) = \sum_{h=1}^{T} w_i p_{\sigma(i)k},$$

being σ: $\{1, \ldots, T\} \rightarrow \{1, \ldots, T\}$ a permutation such that $u_{\sigma(i)} \geq u_{\sigma(i+1)}$, $\forall\ h = 1, \ldots, T-1$, and $u_i = NPF_{\alpha=1}(p_{ik}) = 1 - |p_{ik} - 0.5|$ with $w_1 = 1$ and $w_i = 0$ for $\forall\ h = 2, \ldots, T$.

The *IOWA MinN* operator (Ψ_{MinN}) simply selects the polarity with the more extreme value (very positive or negative polarities) and the *IOWA MaxN* (Ψ_{MaxN}) with the more central value (neutral polarities).

A very interesting property of *IOWA MinN* operator (Ψ_{MinN}) is that it can detect whether all SAM agree on detecting neutrality. Note that if all SAM polarities are close to the neutral point, the aggregated polarity of this operator is also close to this point (Maximum Consensus on Neutrality). On the other hand, *IOWA MaxN* (Ψ_{MaxN}) detects when at least one SAM detects a neutral review. Table 2 shows their associated neutralities for our cases of study, where *All agree* refers to *MinN* and *AtLeastOneAgree* to *MaxN*.

4. Experimented study

In this section, we present an experimented analysis to validate the consensus vote for neutrality detection. We describe the datasets that we use for our study (Section 4.1). After we explain the process of our experiment (Section 4.2) and finally show the results (Section 4.3).

4.1. Datasets

This study is based on nine datasets. We have collected text data from different sources. In order to develop robust analysis, we get data with different properties (short texts like tweets, long reviews like Trip Advisor data) and from different domains (politics, tourism, movies...).

- Amazon[4]: Sentiment ratings from a minimum of 20 independent human raters (all pre-screened, trained, and quality checked for optimal inter-rater reliability).
- ClintonTrump[1]: Tweets from the major party candidates for the 2016 US Presidential Election.
- Food[2]: Food reviews from Amazon [57].
- Cinema Reviews[4]: It includes 10,605 sentence-level snippets. The snippets were derived from an original set of 2000 movie reviews (1,000 positive and 1000 negative).
- Movies[3]: Single sentences extracted from movie reviews [48].
- Runner's World (RW)[4]: Comments from Runner's World Forum.
- TED Talks[4]: Influential videos from expert speakers on education, business, science, tech and creativity, with subtitles in more than 100 languages.
- TA-Sagrada Familia: TripAdvisor reviews from the most popular monument in Barcelona, the Sagrada Familia.
- TA-Alhambra: TripAdvisor reviews from the most popular monument in Granada, the Alhambra.

Table 3 shows the number of words and sentences and its average by corpus. From these numbers, we can infer that Amazon, ClintonTrump, Cinema, Movies and Ted are short reviews. This is because these corpora are texts from Twitter (140 character limit) or single sentences. Food, RW, and TA are corpora with larger reviews.

From each dataset, we randomly select 500 reviews which sum up a total of 4500 opinions.

4.2. Experimental setup

The main target of our experiments is to study the behaviour of polarity classification algorithms in different scenarios. The idea is to evaluate if these algorithms considering neutral reviews as class noise can improve their performance. The experiment setup is described as follows (considering the flowchart of Fig. 2).

We apply the six described SAMs on the datasets. For the Bing and CoreNLP methods, we split the text into sentences and extract sentiment for each. The overall sentiment is defined by majority vote. For MeaningCloud, Microsoft Azure, SentiStrength and VADER the whole text is evaluated.

Once we obtain the sentiment for each review, we normalize the polarities taking into account the SAM and the corpus at the [0, 1] interval.

We then compute the proposed aggregation approaches over the 6 SAM normalized outputs. Finally, we label the polarities as follows: [0,

[1] https://www.kaggle.com/benhamner/clinton-trump-tweets
[2] https://www.kaggle.com/snap/amazon-fine-food-reviews
[3] https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data
[4] https://bitbucket.org/matheusaraujo/ifeel-benchmarking-datasets/src

0.4] are negative reviews, (0.4, 0.6) are neutral reviews and [0.6, 1] are positive reviews (taking notation of Section 3.1, $e = 0.4$ and $I = (0.4, 0.6)$).

We preprocess the text removing stop words, punctuation and numbers. We stem all words and extract the 10 more relevant features in positive and negative reviews with the *tf-idf* metric. We then build the document-term matrix dummy which element $a_{ij} = 1$ if in the *i*-document/review the *j*-word is present.

The models are validated with a 5-fold cross validation (datasets are split in 80% for training and 20% for testing). The classification algorithms selected for this study are: SVM and XGBOOST. We select SVM algorithm because it has been broadly used in the sentiment analysis literature [58]. On the other hand, XGBOOST has been widely deployed in many data science competitions [59]. The parameters of these algorithms are tuned by the `train function` of the `caret` package of R Studio. Finally, we analyze the AUC measure in the test set.

4.3. Results

In this section we first evaluate the consensus among SAMs in detecting neutral opinions (Section 4.3.1). Afterwards, we present the classification results of the proposed models (Section 4.3.2).

4.3.1. Model analysis: neutrality consensus among SAMs

We first study the consensus rate when it comes to detect neutrality. For this, we present Tables 4 and 5 which show the number of neutral instances by the six individual SAMs and the aggregation models, respectively.

As we observe in Table 4 of individual SAMs, there are significant differences between the number of neutral reviews for each dataset. For instance, we observe that VADER detects 16 neutral reviews in TA-Alhambra, while CoreNLP detects 239 instances. But this same SAM only detects 34 neutral reviews in the Cinema's dataset, while Bing detects 283. This fact confirms that our claim holds, there is a need for a consensus voting model to detect neutral reviews.

In Table 5 we present the total number of neutral reviews detected for each proposed aggregation model. MinN obtains a very low number of neutral reviews per corpus but, on the other hand, MaxN obtains a high number. As we have explained before, that means that low agreement exists when detecting neutrality in reviews. There are significant differences in the consensus voting guided by the proximity function, averaging and linguistic quantifiers.

4.3.2. Model analysis: classification performance

We study the classification performance after filtering neutral polarities. We discuss the results of the individual SAMs and the consensus models. We present Tables 6 and 7 which contain the test AUC scores of the SVM and XGBOOST models. Polarities are obtained by each SAM. We then introduce Tables 8 and 9 which show the test AUC scores of the two classifiers. In this case, the polarities correspond to the consensus vote models.

We discuss the attained results summarized in the following items:

- **SVM and XGBOOST for Individual SAMs** (Tables 6 and 7). As we observe in Table 6, there is no method that stands out from the rest. The classification results with the SVM algorithm varies widely on each column, which means that SAMs strongly depends on the corpus where they are evaluated. The results of the XGBOOST classifier presented in Table 7 shows a very similar behaviour. We also detect overfitted models due to the fact that the test AUC is much lower than train (CoreNLP and Ted dataset, SentiStrength and Ted dataset, etc.)
- **SVM and XGBOOST for Aggregation Models** (Tables 8 and 9). We observe that the Aggregation Models also present widely results over the datasets. The results for the SVM and XGBOOST classifiers are similar. We also detect some overfitted models, but to a lesser extent. Note that MaxN is not reported because of the low number of positive and negative instances.
- **Weighting Aggregation vs. Linguistic Quantifiers** (Tables 8 and 9). Studying the average of the polarity classification results of both

Table 4
Number of Neutral Instances of Individual SAMs, I = (0.4, 0.6).

Corpus	Bing	CoreNLP	MC	Microsoft	SentiStr	VADER
Amazon	115	105	186	144	251	193
ClintonTrump	277	221	115	93	228	130
Food	239	209	61	40	102	22
Cinema	283	34	92	52	152	110
Movies	124	42	124	86	174	150
RW	236	211	114	50	156	87
Ted	147	55	107	98	194	155
TA-Sagrada Familia	80	222	66	36	99	34
TA-Alhambra	99	239	54	22	84	16
Average	177.778	148.667	102.111	69.000	160.000	99.667

Table 5
Number of Neutral Instances of Aggregation Models, I = (0.4, 0.6).

Corpus	MinN	MaxN	AvgN	ProN	ProNE	MAP-ProN	ALH-ProN	MO-ProN
Amazon	3	404	221	245	287	183	182	174
ClintonTrump	9	433	207	228	298	163	225	149
Food	0	422	115	151	329	60	258	77
Cinema	3	377	143	167	239	103	156	106
Movies	0	357	199	220	289	175	168	157
RW	0	441	175	214	333	123	270	128
Ted	0	388	144	172	247	119	134	98
TA-Sagrada Familia	0	348	119	158	331	61	260	76
TA-Alhambra	0	369	96	132	307	41	281	61
Average	1.667	393.222	157.667	187.444	295.556	114.222	214.889	114.000

Table 6
Test AUC for SVM models, Individual SAMs.

Corpus	Bing	CoreNLP	MC	MSAzure	SentiStr	VADER	Average
Amazon	0.407	0.800	0.576	0.663	**0.851**	0.734	0.672
ClintonTrump	**0.878**	0.721	0.816	0.694	0.723	0.755	0.764
Food	**0.763**	0.647	0.500	0.613	0.469	0.413	0.567
Cinema	0.387	**0.628**	0.507	0.343	0.368	0.381	0.436
Movies	0.360	0.431	0.398	**0.546**	0.362	0.509	0.434
RW	0.697	**0.751**	0.505	0.559	0.682	0.596	0.632
Ted	**0.804**	0.260	0.263	0.341	0.215	0.346	0.371
TA-Sagrada Familia	**0.850**	0.656	0.714	0.651	0.690	0.724	0.714
TA-Alhambra	0.647	**0.895**	0.763	0.751	0.592	0.565	0.702
Average	**0.644**	0.643	0.560	0.573	0.550	0.558	0.588

Table 7
Test AUC for XGBOOST models, **Individual SAMs**.

Corpus	Bing	CoreNLP	MC	MSAzure	SentiStr	VADER	Average
Amazon	0.400	0.265	0.434	0.613	**0.853**	0.690	0.542
ClintonTrump	**0.795**	0.735	0.767	0.719	0.711	0.731	0.743
Food	0.507	0.648	**0.692**	0.564	0.566	0.629	0.601
Cinema	0.485	0.387	0.518	0.367	0.438	**0.595**	0.465
Movies	0.529	0.471	0.437	**0.574**	0.415	0.409	0.472
RW	0.291	**0.710**	0.579	0.498	0.498	0.418	0.499
Ted	**0.366**	0.318	0.302	0.296	0.252	0.358	0.315
TA-Sagrada Familia	0.529	0.643	0.439	0.622	0.653	**0.746**	0.605
TA-Alhambra	0.678	0.870	**0.916**	0.592	0.501	0.541	0.683
Average	0.509	0.561	0.565	0.538	0.543	**0.569**	0.547

Table 8
Test AUC for SVM models, **Aggregation Models**.

Corpus	MinN	AvgN	ProN	ProNE	MAP-ProN	ALH-ProN	MO-ProN
Amazon	0.302	0.795	**0.853**	0.752	0.633	0.730	0.757
ClintonTrump	0.461	0.733	**0.870**	0.620	0.683	0.790	0.824
Food	0.456	0.407	0.302	0.500	**0.613**	0.609	0.609
Cinema	0.440	0.473	0.218	**0.740**	0.612	0.719	0.558
Movies	0.638	0.396	0.429	0.645	**0.645**	0.532	0.524
RW	0.349	0.375	0.631	0.652	0.713	**0.737**	0.530
Ted	0.305	0.215	0.230	0.286	0.713	0.708	**0.757**
TA-Sagrada Familia	0.535	**0.891**	0.693	0.693	0.875	0.776	0.629
TA-Alhambra	0.819	0.824	**0.941**	0.941	0.507	0.845	0.767
Average	0.478	0.568	0.574	0.648	0.666	**0.716**	0.662

aggregation models, we observe that Linguistic Quantifiers shows a better performance except for the MinN.

- **ALH-ProN, the best aggregation model** (Tables 8 and 9). This model obtains the best average classification results (0.716 and 0.669 for SVM and XGBOOST algorithms). The main idea behind this Linguistic Quantifier is to obtain *at least half* of the consensus among the different SAMs. If we analyze the number of detected neutral instances of this model (see Table 5), we observe that ALH-ProN obtains an average level of neutrality detection. The weights obtained by this model are: (0.3, 0.3, 0.3, 0, 0, 0) which led us to conclude that it does not take into account the 3 SAMs with extreme maximum polarities and gives more weight to those with more conservative behaviour. Therefore, ALH-ProN is a conservative model in terms of neutrality.
- **ALH-ProN vs. Single Models** (Tables 6–9). Finally, we compare the performance of ALH-ProN and the SAMs. Analyzing the results for SVM (see Tables 6 and 8), we observe that ALH-ProN obtains better results on average (ALH-ProN gets 0.072 more points than Bing).

Table 9
Test AUC for XGBOOST models, **Aggregation Models**.

Corpus	MinN	AvgN	ProN	ProNE	MAP-ProN	ALH-ProN	MO-ProN
Amazon	0.336	0.776	**0.819**	0.730	0.600	0.767	0.737
ClintonTrump	0.372	0.671	**0.827**	0.415	0.648	0.724	0.656
Food	0.671	0.234	0.344	0.324	**0.687**	0.637	0.538
Cinema	0.444	0.425	0.371	0.500	**0.657**	0.448	0.514
Movies	0.435	**0.598**	0.471	0.427	0.568	0.434	0.440
RW	0.315	0.375	0.640	0.708	0.691	0.704	**0.717**
Ted	0.346	0.215	0.264	0.323	0.701	**0.810**	0.686
TA-Sagrada Familia	0.499	**0.859**	0.667	0.667	0.721	0.756	0.551
TA-Alhambra	0.768	0.792	**0.932**	0.932	0.606	0.743	0.652
Average	0.465	0.549	0.593	0.559	0.653	**0.669**	0.610

Analyzing the results for XGBOOST (see Tables 7 and 9), we observe that ALH-ProN also obtains better results on average (ALH-ProN gets 0.1 more points than VADER). Therefore, ALH-ProN outperforms single models.

5. Concluding remarks

In this study we have shown that there is a low consensus among SAMs in detecting neutrality. This may be due to different reasons, such as that some tools are trained for one type of text, making it difficult for them to find the polarity in another. As we know, humans write in very different ways and even more so if we have space constraints, as in the case of Twitter. Therefore, a tool trained with tweets will not behave well when analyzing opinions in TripAdvisor, where the text is longer and emoticons are not usually used.

This led us to propose two models of consensus via polarity aggregation. The idea is to detect neutrality based on these consensus models and then filter it out. Then, we study their performance on positive and negative polarities. The results obtained in this study have shown that detecting neutrality based on a consensus improves classification precision. In fact, the ALH-ProN model gets the best results on average. It weighs the polarity of the 3 out of 6 less extreme SAMs.

In fact, there is a wide analysis of classification aggregation [60,61]. There are studies showing that ensembles have proven to outperform single models for SAC [47,53]. The aggregation is also positive for neutrality detection via polarity aggregation.

For future work, we will consider studying different methods for feature or aspects extraction in order to evaluate the robustness of the models [25,62]. We propose to compare the aggregated polarities with the ground truth. In this sense, we propose to label opinions by different experts and then to build aggregation models taking into account experts' sentiment and learning how to aggregate SAM based polarities.

Acknowledgments

We thank our anonymous reviewers for their valuable feedback, which helped to improve the paper. This research has been supported by FEDER and by the Spanish National Research Project TIN2014-57251-P.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi: 10.1016/j.inffus.2018.03.007.

References

[1] E. Cambria, D. Das, S. Bandyopadhyay, A. Feraco, A Practical Guide to Sentiment Analysis, Springer, 2017.

[2] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Cambridge University Press, 2015.

[3] J.A. Balazs, J.D. Velásquez, Opinion mining and information fusion. a survey, Inf. Fusion 27 (2016) 95–110.

[4] S. Sun, C. Luo, J. Chen, A review of natural language processing techniques for opinion mining systems, Inf. Fusion 36 (2017) 10–25.

[5] G. Bello-Orgaz, J.J. Jung, D. Camacho, Social big data: recent achievements and new challenges, Inf. Fusion 28 (2016) 45–59.

[6] S. Poria, E. Cambria, R. Bajpai, A. Hussain, A review of affective computing: from unimodal analysis to multimodal fusion, Inf. Fusion 37 (2017) 98–125.

[7] S. Poria, E. Cambria, D. Hazarika, N. Majumder, A. Zadeh, L.-P. Morency, Context-dependent sentiment analysis in user-generated videos, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 1 (2017), pp. 873–883.

[8] 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.

[9] J. Serrano-Guerrero, J.A. Olivas, F.P. Romero, E. Herrera-Viedma, Sentiment analysis: a review and comparative analysis of web services, Inf. Sci. 311 (2015) 18–38.

[10] M. Koppel, J. Schler, The importance of neutral examples for learning sentiment, Comput. Intell. 22 (2) (2006) 100–109.

[11] M. Koppel, J. Schler, Using neutral examples for learning polarity, International Joint Conference on Artificial Intelligence, 19 Morgan Kaufmann Publishers Inc., 2005, pp. 1616–1617.

[12] J.A. Sáez, M. Galar, J. Luengo, F. Herrera, INFFC: an iterative class noise filter based on the fusion of classifiers with noise sensitivity control, Inf. Fusion 27 (2016) 19–32.

[13] E. Cambria, Affective computing and sentiment analysis, IEEE Intell. Syst. 31 (2) (2016) 102–107.

[14] B. Pang, L. Lee, et al., Opinion mining and sentiment analysis, Found. Trends Inf. Retrieval 2 (1–2) (2008) 1–135.

[15] G. Mishne, N.S. Glance, et al., Predicting movie sales from blogger sentiment. AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, (2006), pp. 155–158.

[16] H. Wang, D. Can, A. Kazemzadeh, F. Bar, S. Narayanan, A system for real-time twitter sentiment analysis of 2012 us presidential election cycle, Proceedings of the ACL 2012 System Demonstrations, (2012), pp. 115–120.

[17] A. Bermingham, A. Smeaton, On using twitter to monitor political sentiment and predict election results, Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology), (2011), pp. 2–10.

[18] A. Ceron, L. Curini, S.M. Iacus, G. Porro, Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizensâ;; political preferences with an application to italy and france, New Media Soc. 16 (2) (2014) 340–358.

[19] E. Marrese-Taylor, J.D. Velásquez, F. Bravo-Marquez, A novel deterministic approach for aspect-based opinion mining in tourism products reviews, Expert Syst. Appl. 41 (17) (2014) 7764–7775.

[20] A. Valdivia, M.V. Luzón, F. Herrera, Sentiment analysis in tripadvisor, IEEE Intell. Syst. 32 (4) (2017) 72–77.

[21] N. Majumder, S. Poria, A. Gelbukh, E. Cambria, Deep learning-based document modeling for personality detection from text, IEEE Intell. Syst. 32 (2) (2017) 74–79.

[22] A. Jha, R. Mamidi, When does a compliment become sexist? Analysis and classification of ambivalent sexism using twitter data, Proceedings of the Second Workshop on NLP and Computational Social Science, (2017), pp. 7–16.

[23] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, A. Vakali, Mean birds: detecting aggression and bullying on twitter, arXiv preprint arXiv:1702.06877, 2017.

[24] J.-M. Xu, X. Zhu, A. Bellmore, Fast learning for sentiment analysis on bullying, Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining, ACM, 2012, p. 10.

[25] S. Poria, E. Cambria, A. Gelbukh, Aspect extraction for opinion mining with a deep convolutional neural network, Knowl. Based Syst. 108 (2016) 42–49.

[26] M. Hu, B. Liu, Mining and summarizing customer reviews, Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2004, pp. 168–177.

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

[28] A. Reyes, P. Rosso, D. Buscaldi, From humor recognition to irony detection: the figurative language of social media, Data Knowl. Eng. 74 (2012) 1–12.

[29] X. Zhong, A. Sun, E. Cambria, Time expression analysis and recognition using syntactic token types and general heuristic rules, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, (2017), pp. 420–429.

[30] Y. Xia, E. Cambria, A. Hussain, H. Zhao, Word polarity disambiguation using bayesian model and opinion-level features, Cognit. Comput. 7 (3) (2015) 369–380.

[31] W. Medhat, A. Hassan, H. Korashy, Sentiment analysis algorithms and applications: a survey, Ain Shams Eng. J. 5 (4) (2014) 1093–1113.

[32] E. Cambria, S. Poria, R. Bajpai, B. Schuller, SenticNet 4: a semantic resource for sentiment analysis based on conceptual primitives, International Conference on Computational Linguistics, (2016), pp. 2666–2677.

[33] L. Oneto, F. Bisio, E. Cambria, D. Anguita, Statistical learning theory and ELM for big social data analysis, IEEE Comput. Intell. Mag. 11 (3) (2016) 45–55.

[34] E. Cambria, A. Hussain, Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis, Springer, 2015.

[35] C.J. Hutto, E. Gilbert, Vader: a parsimonious rule-based model for sentiment analysis of social media text, Eighth International AAAI Conference on Weblogs and Social Media, (2014).

[36] C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S.J. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, (2014), pp. 55–60.

[37] Meaningcloud opinion mining api, 2016a, (https://www.meaningcloud.com/products/sentiment-analysis). Online; accessed Jan 2017.

[38] Microsoft cognitive services text analytics api, 2016b, (https://www.microsoft.com/cognitive-services/en-us/text-analytics-api). Online; accessed Jul 2016.

[39] M. Thelwall, The heart and soul of the web? sentiment strength detection in the social web with sentistrength, Cyberemotions: Collective Emotions in Cyberspace, Springer International Publishing, 2017, pp. 119–134.

[40] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K.J. Miller, Introduction to wordnet: an on-line lexical database, Int. J. Lexicograph. 3 (4) (1990) 235–244.

[41] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, et al., Recursive deep models for semantic compositionality over a sentiment treebank, Proceedings of the conference on empirical methods in natural language processing (EMNLP), (2013), pp. 1631–1642.

[42] A. Go, R. Bhayani, L. Huang, Twitter Sentiment Classification Using Distant Supervision, CS224N Project Report, Stanford, 2009.

[43] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, A. Kappas, Sentiment strength detection in short informal text, J. Am. Soc. Inf. Sci. Technol. 61 (12) (2010) 2544–2558.

[44] J.M. Wiebe, R.F. Bruce, T.P. O'Hara, Development and use of a gold-standard data set for subjectivity classifications, Proceedings of the 37th annual meeting of the Association for Computational Linguistics, (1999), pp. 246–253.

[45] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: Sentiment classification using machine learning techniques, Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, Association for Computational Linguistics, 2002, pp. 79–86.

[46] S.V. Wawre, S.N. Deshmukh, Sentiment classification using machine learning techniques, Int. J. Sci. Res 5 (4) (2016) 1–3.

[47] N.F. Da Silva, E.R. Hruschka, E.R. Hruschka, Tweet sentiment analysis with classifier ensembles, Decis. Support Syst. 66 (2014) 170–179.

[48] B. Pang, L. Lee, Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2005, pp. 115–124.

[49] J.J. Jung, G.-S. Jo, Consensus-based evaluation framework for cooperative information retrieval systems, KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, Springer, 2007, pp. 169–178.

[50] R.R. Yager, On ordered weighted averaging aggregation operators in multicriteria decisionmaking, IEEE Trans. Syst. Man Cybern. 18 (1) (1988) 183–190.

[51] R.R. Yager, Quantifier guided aggregation using owa operators, Int. J. Intell. Syst. 11 (1) (1996) 49–73.

[52] R.R. Yager, D.P. Filev, Induced ordered weighted averaging operators, IEEE Trans. Syst., Man, Cybern., Part B (Cybern.) 29 (2) (1999) 141–150.

[53] 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.

[54] F. Chiclana, E. Herrera-Viedma, F. Herrera, S. Alonso, Some induced ordered weighted averaging operators and their use for solving group decision-making problems based on fuzzy preference relations, Eur. J. Oper. Res. 182 (1) (2007) 383–399.

[55] G. Pasi, R.R. Yager, Modeling the concept of majority opinion in group decision making, Inf. Sci. 176 (4) (2006) 390–414.

[56] J. Kacprzyk, Group decision making with a fuzzy linguistic majority, Fuzzy Sets Syst. 18 (2) (1986) 105–118.

[57] J.J. McAuley, J. Leskovec, From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews, Proceedings of the 22nd international conference on World Wide Web, ACM, 2013, pp. 897–908.

[58] H. Poursepanj, J. Weissbock, D. Inkpen, uottawa: system description for semeval 2013 task 2 sentiment analysis in twitter, SemEval@ NAACL-HLT, (2013), pp. 380–383.

[59] T. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining, ACM, 2016, pp. 785–794.

[60] L.I. Kuncheva, J.J. Rodríguez, A weighted voting framework for classifiers ensembles, Knowl. Inf. Syst. 38 (2) (2014) 259–275.

[61] L. Rokach, Decision forest: twenty years of research, Inf. Fusion 27 (2016) 111–125.

[62] K. Schouten, F. Frasincar, Survey on aspect-level sentiment analysis, IEEE Trans. Knowl. Data Eng. 28 (3) (2016) 813–830.