Statistical Methods & Applications
https://doi.org/10.1007/s10260-023-00691-4

ORIGINAL PAPER

Semi-supervised sentiment clustering on natural language texts

Luca Frigau[1] · Maurizio Romano[1] · Marco Ortu[1] · Giulia Contu[1]

Accepted: 9 March 2023

Abstract
In this paper, we propose a semi-supervised method to cluster unstructured textual data called semi-supervised sentiment clustering on natural language texts. The aim is to identify clusters homogeneous with respect to the overall sentiment of the texts analyzed. The method combines different techniques and methodologies: Sentiment Analysis, Threshold-based Naïve Bayes classifier, and Network-based Semi-supervised Clustering. It involves different steps. In the first step, the unstructured text is transformed into structured text, and it is categorized into positive or negative classes using a sentiment analysis algorithm. In the second step, the Threshold-based Naïve Bayes classifier is applied to identify the overall sentiment of the texts and to define a specific sentiment value for the topics. In the last step, Network-based Semi-supervised Clustering is applied to partition the instances into disjoint groups. The proposed algorithm is tested on a collection of reviews written by customers on *Booking.com*. The results have highlighted the capacity of the proposed algorithm to identify clusters that are distinct, non-overlapped, and homogeneous with respect to the overall sentiment. Results are also easily interpretable thanks to the network representation of the instances that helps to understand the relationship between them.

Keywords Tb-NB · NeSSC · Reviews · Tourism data · Booking.com

✉ Maurizio Romano
romano.maurizio@unica.it

Luca Frigau
frigau@unica.it

Marco Ortu
marco.ortu@unica.it

Giulia Contu
giulia.contu@unica.it

[1] Department of Economics and Business Sciences, University of Cagliari, Viale Sant'Ignazio 17, Cagliari 09123, Italy

Published online: 03 April 2023

1 Introduction

The World Wide Web constantly generates a high number of contents related to any topic (Gowda et al 2016). Different sources such as social network feed, emails, blogs, online forums, survey responses, corporate documents, news, and call center logs produce a huge amount of information. This information is contained in online text assuming the form of either structured or unstructured data. The former presents a tabular format and are machine-readable. The latter does not fit in a pre-defined model and, for this reason, it is necessary to transform it into a machine-readable format (Baek et al 2021). Generally, online textual data are investigated with the aim of converting large volumes of text into information able to support the decision-making process (Gandomi and Haider 2015).

Different approaches have been used to operate on the unstructured data and to identify a structure inside the data. For instance, *Natural Language Processing* (NLP) refers to methods and algorithms that take as input or produce as output unstructured, natural language data (Goldberg 2017). Text mining (TM) gathers different techniques crucial to discover non-trivial patterns and knowledge from unstructured text data by combining data science and knowledge management techniques (Gaikwad et al 2014). TM techniques are information extraction, summarization, categorization, clustering, and information visualization. Each technique has a specific aim. The information extraction aims to analyze unstructured text by identifying key phrases and relationships within the text. The categorization has the goal of assigning one or more categories to free-text documents. The visualization focuses on the representation of individual documents or groups of documents with the aim to support the identification of information inside the data. The text summarization aims to show the most important points and the general meaning of a text. Finally, the clustering can be used to find groups of documents with similar content (Gaikwad et al 2014). Generally, it is an unsupervised method that identifies interrelationships among data to make an assessment (Jain et al 1999; Vallejo-Huanga et al 2017). Over time, researchers have proposed new approaches to improve the results of the clustering method. They have focused their attention on the identification of a new class of models called *semi-supervised methods*. Semi-supervised clustering is a variant of the traditional clustering paradigms. It aims to obtain a better partitioning of data considering and incorporating background knowledge (Gao et al 2006; Hang et al 2017).

In this paper, we propose a semi-supervised method for clustering unstructured textual data according to their sentiment called Semi-Supervised Sentiment Clustering on natural language Texts (S3CT). The method is built by combining different techniques and methodologies. In the first stage, natural language texts are structured, and a Sentiment Analysis algorithm (SA) is used to categorize them into positive or negative classes. In the second stage, the Threshold-based Naïve Bayes classifier (Tb-NB) is applied with the aim of both refining the output of SA and providing the information needed for features texts extraction. Specifically, Tb-NB defines a refined overall sentiment of the texts and allows to define the specific sentiment values of the topics discussed in them. In the last stage, Network-based Semisupervised Clustering (NeSSC) is applied to partition the instances into K disjoint groups. NeSSC defines the pairwise affinity of the instances through a classifier and uses this information to organize them into a network. Finally, a community detection algorithm partitions the instances into communities (i.e., clusters) by minimizing the overlapping of the overall sentiment as much as possible.

Four sections, besides the introduction, complete this study. In Sect. 2 we describe the basic ingredients of S3CT, these are Sentiment Analysis (Sect. 2.1), and Semi-Supervised Clustering (Sect. 2.2). Section 3 illustrates the methodological framework of S3CT. Section 4 reports an example of S3CT algorithm performance on real data of online reviews published on *Booking.com*, specifically data collection and cleaning process (Sect. 4.1), S3CT setting and feature extraction (Sect. 4.2), results and discussion (Sect. 4.3). Section 5 ends the paper with some concluding remarks.

2 Background

2.1 Sentiment analysis

Natural human language conveys two kinds of information: objective information about facts and subjective information driven by human emotional states (Nanli et al 2012). Sentiment analysis, also known as opinion mining, investigates, analyzes, and extracts subjective humans' opinions, preferences, and sentiment to quantify the affective states and subjective information expressed by humans in textual form. Broadly speaking, sentiment analysis can be divided into three sub-levels: i) Document level, ii) Sentence level, and iii) Aspect level (Nanli et al 2012).

The purpose of sentiment analysis is the detection of *Subjectivity/Objectivity*, *Polarity* and *Discrete Emotions* in textual data. Subjectivity/Objectivity detection concerns the primary identification of subjective versus objective text, since only subjective text contains sentiment information. Polarity detection aims to assign a qualitative (positive/negative) or quantitative (a number in a given range) sentiment score to a given text. Discrete Emotion detection is intended as a finer grain analysis to extract emotions such as *joy*, *love* etc. from human language.

The information extracted with the sentiment analysis is essential in activities such as *decision-making support*, *business applications* and *predictions and trend analysis* (Pang and Lee 2008).

Sentiment analysis is a multidisciplinary task that mainly involves: natural language processing, text analysis, and computational linguistics. Sentiment analysis techniques can be categorized into two main categories (Kaur et al 2017; Liu 2012): i) Lexicon based approaches ii) Machine Learning based approaches. Lexicon based approaches (Taboada et al 2011) are further distinguished in *Dictionary based* and *Corpus based* approaches (Darwich et al 2019). The former involves the use of a dictionary of terms built by linguists experts, who assign a score relative to the sentiment of every single term. The latter approach relies on co-occurrence statistics or syntactic patterns in text corpora and a set of predefined positive and negative seed words.

Machine learning approaches (Agarwal and Mittal 2016) are divided into three main categories (Madhoushi et al 2015): i) supervised, ii) unsupervised, and iii) semi-supervised approaches. *Supervised learning* is a robust and effective solution in traditional document classification, and it is adopted for sentiment analysis with good results (Sodanil 2016). Examples of supervised sentiment analysis algorithms are: Naïve Bayes, a generative classifier that assumes independence between the features; Support Vector Machine (SVM), a discriminative classifier that makes no prior assumptions based on training data; and deep learning methods based on attention mechanism (Vaswani et al 2017) with general higher classification performance but less interpretation than other algorithms. The most prevalent and interpretable supervised learning methods in sentiment analysis are Naïve Bayes (Sun et al 2012; Nigam et al 2000) and SVM (Madhoushi et al 2015).

Unsupervised learning methods in sentiment analysis do not need prior information in the training data to detect sentiment polarity; examples of such methods are the *rule-based* classifiers (Vashishtha and Susan 2019; Hu et al 2013).

In contrast with supervised learning, which learns from labeled data only, *semi-supervised learning* learns from both labeled and unlabeled data (Liu 2012). While unlabeled data does not give information about classes, it does provide information on joint distribution over classification features, which is the basic idea driving semi-supervised learning. In cases where the labeled data is limited in the target data domain, semi-supervised learning using unlabeled data can outperform supervised learning. Common examples of semi-supervised learning methods include (Silva et al 2016): self-training (He and Zhou 2011), generative models (Eguchi and Lavrenko 2006), co-training (Wan 2009), multi-view learning (Sadr et al 2020), and graph-based methods (Wang et al 2011; Goldberg and Zhu 2006).

2.2 Semi-supervised clustering

Semi-supervised clustering models have been developed to overcome issues both of unsupervised clustering methods and of supervised classification methods. Regarding the former, Yu et al (2016) highlighted experts do not take advantage of their prior knowledge on the dataset in the clustering process as well as results obtained handling high dimensional data are not relevant. Instead, Bair (2013) demonstrated that traditional supervised classification methods are not useful when it is necessary to analyze a subset of labeled data. Consequently, several mixed approaches of the unsupervised and supervised methods have been proposed based on semi-supervised clustering models. They can be classified in different ways according to their characteristics. A first classification divides them in three general categories of methods: *constraint-based* , *distance based* and *hybrid* (Xiang and Min 2010; Yi et al 2015; Yoshida 2014). The aim of constraint-based methods is to handle the clustering process with pairwise instance constraints or initialize cluster centroids by labeled instances. They use a specific constraint to address the algorithm to obtain the appropriate data partitioning (Basu et al 2004) by modifying the objective function in order to respect the provided constraints (Nogueira et al 2017). Instead, the distance-based approach uses the constraints to learn a new distance metric and group instances. Specifically, it is characterized by the definition of a clustering distortion measure able to define a good partition. Lastly, the hybrid approach combines the two above-mentioned approaches under a probabilistic framework.

In addition, semi-supervised methods can be distinguished with respect to the similarity measure and the pairwise constraints. Specifically, Grira et al (2004) have classified the methods in *similarity-adapting methods*, where the similarity measures are adapted with respect to the available constraints, and in *search-based methods*, where the clustering algorithm is modified to consider the constraints and the labels to better perform clustering. The use of constraints can modify the results of clustering algorithms.

Another interesting classification of semi-supervised clustering was proposed by Bair (2013). They have identified three main semi-supervised clustering approaches: *partially labeled data*, *cluster with constraints*, and *cluster associated with an outcome variable*. The former is based on label information. The cluster assignment for a subset of the data is known previously. Consequently, the cluster analysis involves classifying the remaining unlabeled observations considering the known cluster assignment for labeled data. In this approach, labeled and unlabeled data are used in the algorithm (Bilenko et al 2004; Sun et al 2012). The goal is to understand both how combining labeled and unlabeled data can change the learning behavior and how to design algorithms to take advantage of such a combination (Zhu and Goldberg 2009). The second class of algorithms is based on the presence of complex relationships among the observations. Two kinds of constraints have been defined in literature: *Must-Link Constraints* and *Cannot-Link Constraints*. The two constraints identify if two points belong to the same cluster or different clusters (Kestler et al 2006). Before applying the cluster method, researchers know if a specific relationship among the observations exists, and the cluster model involves taking into account this information. Finally, in *cluster associated with an outcome variable*, the clusters are realized considering a given outcome variable and defined considering previous information of the outcome variable.

Text Clustering is the task to automatically group textual documents (e.g. PDFs, social media content, reviews) into clusters which contain documents with similar content (e.g. document regarding a specific topic should appear in the same cluster). In literature, different researchers have used the first approach of semi-supervised clustering to classify texts and documents. For instance, Sun et al (2012) have proposed a *Semi-Supervised Cluster tree method* (SSC). It is a tree-like semi-supervised classifier built taking into account both labeled and unlabeled data. It is composed of different steps. In the first phase, the clustering is carried out. In the second phase, the information of cluster is used to define a batch-model query strategy to select the most informative data, with the aim to distribute in all clusters the selected unlabeled examples. Zhang et al (2015) have theorized the *TExt classification using Semi-supervised Clustering* (TESC) in order to improve text classification. They have used semi-supervised clustering to identify text components and, later, to use text components to predict labels of unlabeled documents. Gowda et al (2016) have proposed a semi-supervised learning algorithm for the classification of text documents. It is defined using K-means algorithm for partitioning both labeled and unlabeled data collection. K-means algorithm is applied recursively until each partition contains labeled documents of a single class. Finally, cluster centroids are used for classifying an unknown text document.

3 Methodology

Textual data consists of a collection of texts, and each one refers to a single observation. Texts are made up of one or more sentences and deal with many topics. The method we propose aims to partition into K disjoint clusters a set of observations characterized by textual data. Initially, their features expressing topics are identified and characterized by sentiment scores. Successively, a semi-supervised clustering method which follows the class of algorithms called *cluster associated with an outcome variable* is applied using a feature measuring the overall sentiment of the observations as noisy surrogate of the unobserved clusters.

3.1 Sentiment scores definition

Let us consider a collection of texts $\mathcal{T} = \{t_1, \dots, t_T\}$ describing the observations included in $\mathcal{O} = \{1, \dots, n\}$, such that they have a surjective relation $f : \mathcal{T} \rightarrow \mathcal{O}$. We consider $f(\cdot)$ as known; consequently, it is possible to assign the texts to the corresponding observation. Specifically, each t is considered as a collection of n unordered sentences, that is $t = \{s_1, s_2, \dots\}$, and a sentence j as a set of words, that is $s_j = \{w_1, w_2, \dots\}$. The collection of all sentences is represented by the set $\mathcal{S} = \bigcup_{i=1}^{T} \{\forall s \in t_i\}$.

The first step of the algorithm consists in features extraction and sentiment score assignment. At the beginning, the sentences are mapped to the set $\Omega = \{\mathcal{P}, \mathcal{N}\}$ such that $g : \mathcal{S} \rightarrow \Omega$, where $g(\cdot)$ is a function that identifies the sentiment of a sentence, which can be either positive ($\mathcal{P}$) or negative ($\mathcal{N}$). The elements of $\mathcal{S}$ can be grouped according to their sentiment into a set of positive sentences $\mathcal{S}^+ = \{\forall s \in \mathcal{S} : g(s) = \mathcal{P}\}$ and into a set of negative ones $\mathcal{S}^- = \{\forall s \in \mathcal{S} : g(s) = \mathcal{N}\}$ such that $\mathcal{S}^+ \cup \mathcal{S}^- = \mathcal{S}$ and $\mathcal{S}^+ \cap \mathcal{S}^- = \emptyset$.

In order to refine the output of $g(\cdot)$ with a precise score and provide information needed for features extraction, Tb-NB is applied on the results obtained from $g(\cdot)$. Tb-NB is a new version of Naïve Bayes classifier that utilizes a data-driven decision rule to assign to a sentence $s_j \in \mathcal{S}$ its probabilities to belongs to $\mathcal{S}^+$ and to $\mathcal{S}^-$. Let us define $\mathcal{W}$ as the collection of all words used in all sentences, that is the Bag-of-Words (BoW) of $\mathcal{S}$. Considering a probability function $\pi(\cdot)$, Tb-NB builds on the Bayes' rule and computes a scoring function $\Lambda(\cdot)$ for all the words $w_k \in \mathcal{W}$ as for predicting if they are included in sentences with a positive or negative sentiment. Specifically, $\Lambda(w_k)$ corresponds to the log-odds ratio of the probability that a sentence s is positive given that it includes a certain word w_k, i.e. $\pi(s \in \mathcal{S}^+ | w_k)$, over the probability that s is negative given that it includes w_k, i.e. $\pi(s \in \mathcal{S}^- | w_k)$. Notionally, the log-odds ratio for a word w_k included in a sentence s is the following

$$
\begin{aligned}
\Lambda\big(w_k|(\mathcal{S}^+,\mathcal{S}^-)\big) &= \log\left[\frac{\pi(s\in\mathcal{S}^+|w_k)}{\pi(s\in\mathcal{S}^-|w_k)}\right] \\
&= \log\left[\frac{\pi(w_k|s\in\mathcal{S}^+)}{\pi(w_k|s\in\mathcal{S}^-)}\cdot\frac{\pi(\bar{w}_k|s\in\mathcal{S}^+)}{\pi(\bar{w}_k|s\in\mathcal{S}^-)}\cdot\frac{\pi(s\in\mathcal{S}^+)}{\pi(s\in\mathcal{S}^-)}\right] \\
&= \underbrace{\Big[\log\pi(w_k|s\in\mathcal{S}^+)-\log\pi(w_k|s\in\mathcal{S}^-)\Big]}_{\mathcal{L}(w_k)} \\
&\quad+\underbrace{\Big[\log\pi(\bar{w}_k|s\in\mathcal{S}^+)-\log\pi(\bar{w}_k|s\in\mathcal{S}^-)\Big]}_{\mathcal{L}(\bar{w}_k)} \\
&\quad+\underbrace{\Big[\log\pi(s\in\mathcal{S}^+)-\log\pi(s\in\mathcal{S}^-)\Big]}_{\varphi} \\
&= \mathcal{L}(w_k)+\mathcal{L}(\bar{w}_k)+\varphi.
\end{aligned}
\tag{1}
$$

Thus, $\Lambda(w_k)$ derives from the sum of three components: the functions $\mathcal{L}(w_k)$ and $\mathcal{L}(\bar{w}_k)$, expressing the log-likelihood ratio of the events $(w_k \in s)$ and $(w_k \notin s)$, that measure how much the presence and the absence of a word w_k in a sentence influence its positive sentiment; the third component is the term $\varphi = \Big[\log\pi(s\in\mathcal{S}^+)-\log\pi(s\in\mathcal{S}^-)\Big] = \Big[\log\Big(\frac{|\mathcal{S}^+|}{n}\Big)-\log\Big(\frac{|\mathcal{S}^-|}{n}\Big)\Big]$, which consists in the difference between the logarithmic of the proportion of the positive sentences and that of the negative ones, expressing how much a sentence is characterized by a positive sentiment regardless of the words that compose it; φ does not depend on w_k and consequently assumes a constant value for all the words, not providing information on their ability to influence the sentiment of the sentence.

Overall sentiment scores $\alpha^\star$ for the sentences are built considering the two components of $\Lambda(w_k)$ that depend on w_k, that is $\mathcal{L}(w_k)$ and $\mathcal{L}(\bar{w}_k)$, by summing the values for all the words of the BoW according to the words that are present $(w_k \in s_j)$ and those that are absent $(w_k \notin s_j)$.

Specifically,

$$
\alpha^\star(s_j) = \sum_{w_k\in\mathcal{W}}\big[\mathcal{L}(w_k)I(w_k\in s_j)+\mathcal{L}(\bar{w}_k)I(w_k\notin s_j)\big],
\tag{2}
$$

where $I(\cdot)$ is an indicator function.

3.2 Feature extraction

Concerning the features extraction, the BoW set is partitioned into M disjoint subsets each one dealing with a specific topic, such that $\mathcal{W}=\mathcal{A}_1\cup\mathcal{A}_2\cup\cdots\cup\mathcal{A}_M$. Then, a sentiment score for each topic is measured as

$$\alpha^m(s_j) = \sum_{w_k \in \mathcal{A}_m} \left[\mathcal{L}(w_k) I(w_k \in s_j) + \mathcal{L}(\bar{w}_k) I(w_k \notin s_j) \right]. \tag{3}$$

The features of the observations are computed by averaging the sentiment scores of the sentences by the observations to which the sentences are referred. We indicate the noise surrogate feature with $\mathbf{y}$ and with $\mathbf{x}_m$ the feature concerns the topic m. More in details, for each observation h the value of the feature m is

$$x_{hm} = \frac{\sum_{s \in \mathcal{O}_h} \alpha^m(s)}{|\mathcal{O}_h|} \tag{4}$$

where $\mathcal{O}_h = \{\forall s \in \mathcal{S} : f(s) = h\}$ is the set that identifies the sentences referred to the observation h, and $|\mathcal{O}_h|$ is the cardinality of the set $\mathcal{O}_h$.

The noise surrogate feature, instead, is computed in the same way but using the overall sentiment scores $\alpha^\star$.

3.3 Computation of proximity matrix

Once the features $\mathbf{x}_1, \dots, \mathbf{x}_M$ and $\mathbf{y}$ are computed, NeSSC algorithm (Frigau et al 2021) is performed on them. NeSSC aims to find a partition of observations into K disjoint clusters taking into account a noisy surrogate of the unobserved clusters. NeSSC carries out three steps: initialization, training, and agglomeration. The initialization and training steps deal with the estimation of the affinities between observations by using an iterative process. Specifically, the goal of these two steps is to estimate an affinity matrix $\mathbf{\Pi}$, where the value of its generic element in the r-th row and the c-th column, $\mathbf{\Pi}_{rc}$, is strictly proportional to the affinity between the instances r and c, that is their propensity to behave in the same manner with respect to $\mathbf{y}$. A generic element of $\mathbf{\Pi}$ after B iterations is:

$$\mathbf{\Pi}_{rc}^{(B)} = \frac{1}{B} \left[\sum_{b=1}^{B} I\left(\left[\hat{\mathcal{F}}\left(\mathbf{x}^{(b)}, \mathbf{y}, \mathbf{u}^{(b)}, \boldsymbol{\theta}^{(b)}\right)\right]_r = \left[\hat{\mathcal{F}}\left(\mathbf{x}^{(b)}, \mathbf{y}, \mathbf{u}^{(b)}, \boldsymbol{\theta}^{(b)}\right)\right]_c \right) \right]. \tag{5}$$

$\mathcal{F}$ is a supervised statistical classifier, that uses one feature $\mathbf{x}$ to evaluate the aptitude of r and c to behave similarly with respect to the outcome $\mathbf{y}$: that is reached by training the model on data using $\mathbf{y}$ as the dependent variable and $\mathbf{x}$ as the an independent one, and checking if the trained model predicts the observations r and c with the same value. At each iteration, the feature $\mathbf{x}$ to use in $\mathcal{F}$ is sampled among the M features according to the features' weights $\mathbf{u}^{(b)}$, which considers the average predictive ability of each feature in the previous $b-1$ iterations. The vector $\boldsymbol{\theta}^{(b)}$, instead, concerns the weights associated with observations in the iteration, which are inversely proportional to the average the affinity of each observation computed after $b-1$ iterations. The estimation of the affinity matrix concludes after B^* iterations, that is the number of iterations needed to the distribution of $\mathbf{\Pi}^{(b)}$ to stabilize and to be similar to $\mathbf{\Pi}^*$. The latter corresponds to the affinity matrix expressing the *true* propensity of observations to behave in the same manner with respect to $\mathbf{y}$. Formally,

$\mathbf{\Pi}^{(b)}$ converges in distribution to $\mathbf{\Pi}^{*}$, thus $\lim_{b\to\infty}\mathbf{\Pi}^{(b)} = \mathbf{\Pi}^{*}$. The optimal number of iterations B^{*} leading to $\mathbf{\Pi}^{(B^{*})} \approx \mathbf{\Pi}^{*}$ is reached when the average differences of the elements of $\mathbf{\Pi}^{(B^{*})}$ and those of the affinity matrices obtained in the previous γ iterations are smaller than a threshold value ϵ. Specifically,

$$B^{*} = \min\,(b \in \mathcal{B}) \;:\; \sum_{h=1}^{\gamma}\left[I\left(\mathbf{1}^{\top}\left|\mathbf{\Pi}^{(b)} - \mathbf{\Pi}^{(b-h)}\right|\mathbf{1}\right) < n(n-1)\cdot\epsilon\right] = \gamma. \tag{6}$$

3.4 Data partitioning

In the third and last step of NeSSC, i.e. agglomeration, *Overlapping criterion* is performed. It consists in transforming the affinity matrix $\mathbf{\Pi}^{(B^{*})}$ into a set of subnetworks, where the observations correspond to nodes and $\mathbf{\Pi}^{(B^{*})}_{rc}$ defines the strength of the edge between the two observations (i.e. nodes) r and c. More precisely, each subnetwork included in the set is defined on the basis of a threshold value τ that indicates if, for a generic $\mathbf{\Pi}^{(B^{*})}_{r,c}$, an edge exists between observations r and c or not. In practice, this criterion implies that:

$$\begin{cases} \mathbf{\Pi}^{(B^{*})}_{r,c} \geqslant \tau \rightarrow \text{edge between } \text{ r and } \text{ c exists} \\ \mathbf{\Pi}^{(B^{*})}_{r,c} < \tau \rightarrow \text{edge between } \text{ r and } \text{ c does not exist} \end{cases}$$

The threshold values τ are detected by taking into account the structural changes of the empirical fluctuation process of the frequency distribution of the values of $\mathbf{\Pi}^{(B^{*})}$.

In order to identify the threshold values in which the trend of frequencies of the values of $\mathbf{\Pi}^{(B^{*})}$ changes significantly, an empirical fluctuation process is defined by the corresponding recursive residuals. The threshold values correspond to the breakpoints in which the empirical fluctuation process is different from a linear model with null slope ($p < 0.05$). The empirical process is assumed to follow a MOSUM process and a OLS-based MOSUM test is performed (Zeileis et al 2003; Kleiber 2002). Next, a community detection algorithm is trained on the nodes of the networks in order to find, for each network, a certain number of homogeneous communities corresponding to the groups defining a proper partition of the original data. Finally, among all the partitions, the optimal one is that presenting the lowest minimum penalized (to avoid overestimation of the number of clusters) average overlapping, the latter computed as the weighted mean of the pairwise group overlapping index.

4 Motivating example

The proliferation of online customers' reviews (e-WOM) has been reported to being as one of the most important information sources in the industry, thus it has gained considerable attention (Schuckert et al 2015). Moreover, since the online reviews include their peripheral cues, such as user-supplied photos and the reviewer's personal information, they are intended as means of persuasive communication in order to build credibility and influence user behavior (Sparks et al 2013). Consequently, we have decided to present the basic features of S3CT by clustering hotel structures according to their customers' reviews, with the aim to identify groups of hotels similar with respect to the overall sentiments of the reviews. We have chosen *Booking.com* as a reference platform because the reviews there available come from customers who effectively stayed in a hotel. We focus the analysis on data related to the hotel located in Sardinia, an Italian island, lived by tourists.

4.1 Data processing

By using a web-scraping extractor developed in Python, all the 66,237 reviews from January 3rd, 2015 to May 27th, 2018 (1,240 days) are then gathered from the platform *Booking.com*. They concern 619 hotels located all around Sardinia. In addition some features on the hotels were gathered as well: the hotel rating in five-star system (*stars*); if the reason of the trip is for business rather than pleasure (*business*); if the district is close to the coast (*coast*) and if it has more than 35,000 inhabitants (*over 35k*), and if the length of stay is within three days (*short stay*).

One of the most important phases in the process of analyzing textual data is data cleaning. In fact, usually, the raw data (especially if web-scraped) come with problems that confuse the algorithm, such as: unnecessary words, stop words, acronyms. Furthermore, emojis or emoticons needs to be properly converted into suitable textual information. Dealing with those kinds of problems allows improving the quality of the analysis. For that, as a preprocessing phase, those subsequent steps describe the detailed procedure applied to every observation contained in the dataset.

1. *basic filtration*: firstly, all the useless information is removed. For instance, links (especially truncated ones), unknown acronyms, and other recurrent or meaningless keywords like RT (re-tweets), @ username, nonsense #hashtags, alphanumeric characters, etc.;
2. *emoticons and emoji replacement*: emoticons like:-) or:-(and emojis like ☺ or ☹ are properly converted in text with the corresponding meaning. With this transformation, it is possible to treat them as normal text hence accessing their useful information;
3. *advanced filtration and normalization*: content is divided into separate words without punctuation (tokens). Cases of all alphabets are normalized to lowercase. Moreover, stop words like "an", "I", "oh", "or" are removed from data as they do not provide any sentiment information. However, negative words that might alter the sentence meaning (like "not") are kept;
4. *merging and assembling*: as the last step, the tokens are reduced to their root or base form (a process also known as stemming). For example, "consult", "consultant", "consulting" are all reduced to "consult" and replaced by it. In that way,

most of the words related to the same topic are merged. Moreover, all the tokens are assembled back to recreate the sentence.

4.2 S3CT setting and features extraction

A peculiar characteristic to the platform *Booking.com* is that the customers are asked to separate the positive sentences from the negative ones when leaving a review (hence called also text). In the application of S3CT it means to have the sentences already mapped in Ω. In other words, it is like we have fitted the function $g(\cdot)$ perfectly, because the sentences are correctly divided into positive and negative. Considering all the 66,237 texts, we collected 106,800 sentences of which 62,291 are positive whilst 44,509 are negative.

In order to extract the features from the sentences $M = 13$ reference categories of words included in the BoW $\mathcal{W}$ are defined. They concern 12 specific topics (Table 1) and one residual category. Due to the indistinctness of the latter category and its consequent impossibility to provide useful interpreting elements, that was not employed in the clustering process.

In the semi-supervised clustering stage, the affinity matrix $\mathbf{\Pi}$ was fitted by setting $\epsilon = 0.001$ and $\gamma = 30$, whilst the community detection algorithm adopted was Label Propagation (Raghavan et al 2007).

4.3 Results

Table 2 reports the characterization of the clusters obtained by carring out S3CT algorithm to *Booking.com* dataset. In Table 2, the values of the overall sentiment and those of the twelve features were standardized by subtracting them to the overall mean and dividing by the overall standard deviation.

Table 1 Reference categories of words included in the BoW used for defining the features employed in the clustering process of S3CT

Feature	Description
Bar	Bar service, minibar or affiliated bars around the structure
Cleaning	Cleaning of the room and of the hotel structure
Comfort	Comfort perceived by guests during their stay
Food	Food quality served in the hotel structure or affiliated restaurant
Hotel	Quality of physical structure
Position	Position quality of the hotel structure
Quality-price-rate	Quality-price rate
Room	Room quality referred to the type of room
Services	Quality of services provided by the hotel
Sleep-quality	Quality of sleep such as bed, pillows and noises
Staff	Kindness, courtesy, professionalism of the staff
Wifi	Wi-Fi connection quality, such as general coverage and lags

Table 2 Characterization of the five clusters obtained partitioning *Booking.com* dataset by the method proposed

Feature	Cluster 1		Cluster 2		Cluster 3		Cluster 4		Cluster 5	
Overall sentiment	**1.285**	***	**0.676**	***	**0.354**	**	**– 0.128**	*	**– 0.801**	***
Bar	0.113		0.139		0.158		– 0.081		– 0.094	
Cleaning	0.645	***	0.249	**	0.139		0.003		– 0.403	***
Comfort	0.836	***	0.355	***	0.043		– 0.188	**	– 0.324	***
Food	0.286	**	0.270	**	0.271	*	– 0.023		– 0.308	***
Hotel	– 0.349	**	0.129		– 0.009		0.099		– 0.015	
Position	0.162		0.038		0.030		0.016		– 0.101	
Quality-price-rate	0.298	**	0.398	***	0.187		– 0.086		– 0.288	***
Room	0.137		0.093		0.180		0.103		– 0.247	***
Services	0.843	***	0.169		0.184		– 0.136	*	– 0.335	***
Sleep-quality	– 0.058		0.085		0.150		– 0.044		– 0.036	
Staff	1.787	***	0.738	***	0.330	**	– 0.174	**	– 0.964	***
Wifi	0.168		0.183		– 0.064		– 0.049		– 0.081	
Stars	3.023		3.125		3.355		3.473		3.573	
Over 35k	7.0%		13.9%		8.1%		16.0%		39.3%	
Coast	93.0%		94.4%		80.6%		91.3%		89.9%	
Business	3.5%		3.5%		5.1%		6.1%		16.2%	
Short stay	65.4%		67.5%		68.4%		68.3%		77.6%	

The first column reports the names of the features. From the second to the sixth columns the values referred to clusters. The first 13 rows concern the average values of the clusters scaled by subtracting them to the overall mean and dividing by the overall standard deviation. The results of the outcome variable *overall sentiment* are reported in bold. Furthermore, the statistical significance of the difference between the cluster means and the overall one (performed before scaling) are indicated as follows: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$. The last five rows concern structural features describing the clusters that have not taken part in the partitioning process

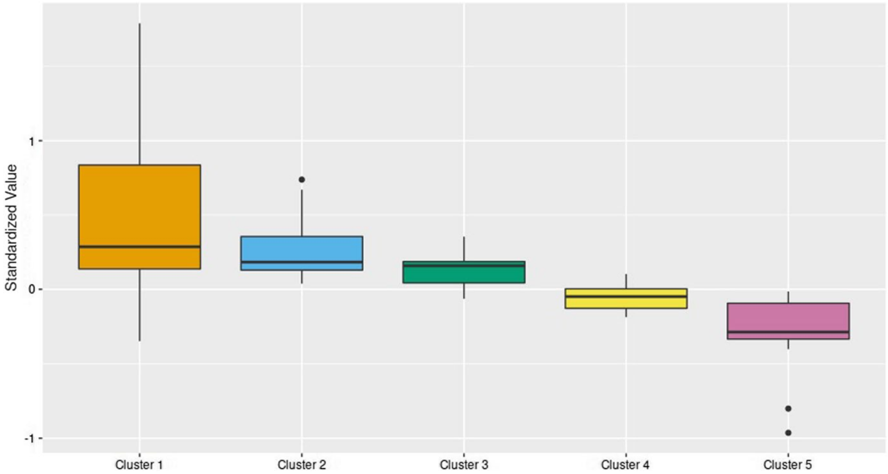

Fig. 1 Distributions of the features standardized values of Table 2 by clusters: each boxplot is built using the thirteen standardized values of the features that characterize each cluster

The results show the presence of five clusters significantly characterized by a different level of the overall sentiment. Moreover, Fig. 1 highlights how the clusters are characterized by standardized values of the features very close to each other.

The first cluster presents an overall sentiment greater than the mean, aspects as cleaning, food, comfort, price-quality rate, services, and staff influence it positively. Only the quality of the physical structure of the hotel negatively influences the sentiment. The cluster groups hotels characterized by a low average of stars (3.023), located in a district close to the coast (93.3% of the hotels) and in a small district with a reduced population (only the 7.0% of the hotels is located in districts with more than 35,000 inhabitants). Moreover, the hotels record the number of short-stay equal to 65.4%, the lowest value among the clusters, and the 3.5% of tourists is a businessman.

The clusters two and three present in a similar way an overall sentiment greater than the mean; however, some differences can be recorded. Specifically, the sentiment of cluster two is influenced positively by cleaning, food, comfort, and staff feature. The hotels grouped in this cluster are located close to the sea (the 94.4%, which is the highest value among the clusters). They are characterized by a higher percentage of short stays than cluster one. Instead, the percentage of businessmen is equal to the first cluster. The sentiment of cluster three is conditioned only by the food and the staff. The hotels grouped in this cluster present an average of stars higher than cluster one. The tourism structures are located in small districts, and the percentage of the hotels close to the coast is the lowest (80.6%). The percentage of businessmen who choose these hotels is higher than clusters one and two, as the percentage of the short stay.

The last two clusters are characterized by an overall sentiment lower than the mean. In cluster four, only comfort, services, and staff condition negatively the sentiment. This cluster groups the hotels characterized to an average value of stars equal to 3.473, a percentage of districts with more of 35, 000 inhabitants equal to 16%. Finally, cluster five seems to be totally opposite to cluster one since it is characterized by sentiments lower than the mean. Aspects as cleaning, food, comfort, price-quality rate, room services, and staff negatively impact the sentiment. The hotels in this cluster present the highest average value of stars (3.573); they are located for the 39.3% in districts with an elevated number of inhabitants. Businessmen are the 16.2% of the tourists hosted in hotels, and the 77.6% of the tourists choose to spend a reduced number of days in the hotels.

Figure 2 shows a representation of the cluster communities in the network obtained using the Label Propagation algorithm; edges are presented in grey, and nodes are colored according to their clusters. This representation highlights the relative connection on the network communities, which represent the clusters obtained in the clustering step. To describe the relationship of the observations by a network allows enhancing their interpretation by using the statistical tools typical of graphs. For instance, it could be interesting to use a measure of centrality like the normalized degree of vertices, which corresponds to the number of connections of the nodes divided by the number of the possible connections, to assess the communities' similarity. Considering the whole network, the average of all degrees of vertices is equal to 0.15. Instead, within the communities, it

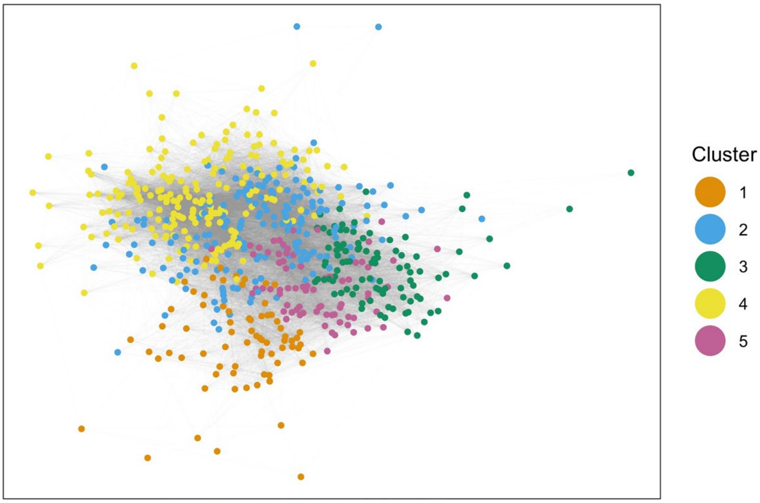

Fig. 2 Network and communities detected of the Sardinian hotels of *Booking.com* dataset. Each node of the graph corresponds to a hotel. The nodes assume different colors according to their own community of belonging. The grey lines identify the edges between the nodes

ranges from 0.37 to 0.66, and between them, from 0.03 to 0.09, demonstrating a strong inner connection and weak connection with other communities. Furthermore, the network structure allows studying analytically the single observations (i.e., the hotels), because the presence of an edge means similarity between the two observations linked through it exists.

Figure 3 illustrates the clusters in a three-dimensional space by projecting the instances using the coordinates of the first three principal components obtained from a Principal Components Analysis, which preserves 96.2% of the total variance. The quality of the visualization is enhanced by T-distributed Stochastic Neighbor Embedding (T-SNE) (Van der Maaten and Hinton 2008; Krijthe and Van der Maaten 2015), which preserves the distances in the reduced space, namely points that are close/far to each other in the original space are close/far in the reduced space. We can note that there is a general tendency of proximity for points belonging to the same cluster, emphasizing the feasibility and goodness of the clustering technique proposed. PCA and T-SNE are often used in combination, as the T-SNE for dense or sparse data requires another dimensionality reduction method to improve performance and to suppress some noise and speed up the computation of pairwise distances between samples (Van Der Maaten 2010).

Instead, by analyzing Fig. 4, we can observe that the hotels grouped in the different clusters are equally distributed in the Sardinian territory. This means that the sentiment score lower or greater than the mean is not associated with the location, but with the aspects related to the different services, staff, and price.

In terms of managerial implication, the relevance of these aspects should take into account by hotel managers in order to improve the overall sentiment. For instance, the feature staff is one of the most relevant in terms of overall sentiment.

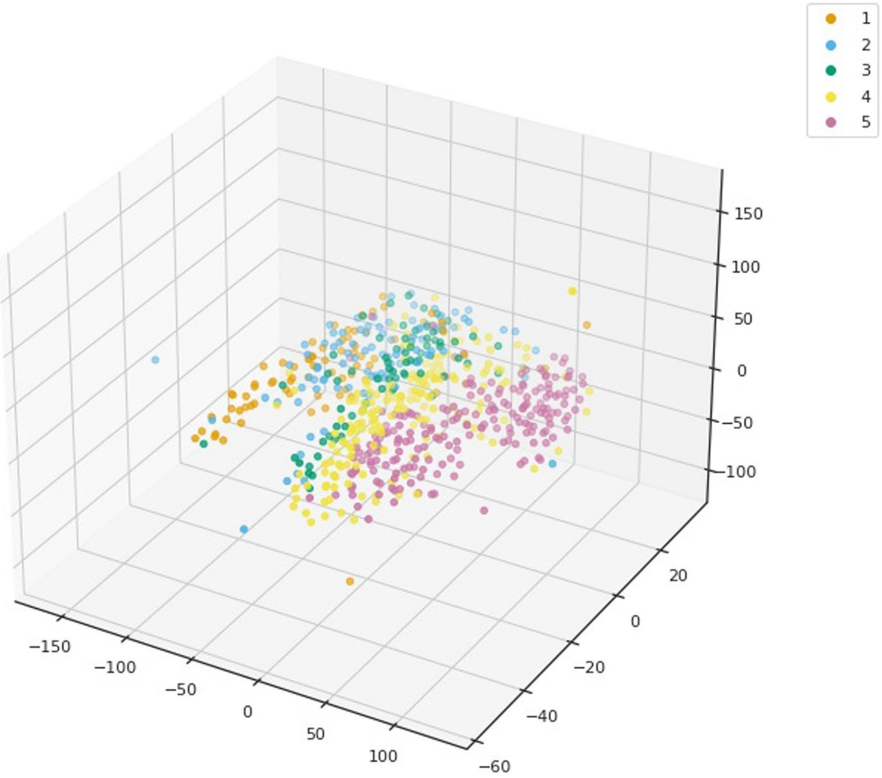

Fig. 3 Three dimensional space representation of the clusters using the Principal Component Analysis and the T-distributed Stochastic Embedding. The three principal components explain the 96.2% of the total variance of the samples. The points assume different colours according to their own cluster of belonging

The results suggest the need to employ inside the hotel a staff collaborative, supportive, and able to help the tourist during the hotel's stays. The presence of the staff represents a key to overall sentiment success. Moreover, particular attention should be focused on a specific typology of clients as businessmen. The lowest overall sentiment is recorded in the cluster where the percentage of businessmen is the highest. This aspect underlines how specific typologies of tourists have more necessity and expectation than others, and their reviews are lower than the average.

5 Conclusions

This paper proposed Semi-Supervised Sentiment Clustering on natural language Texts framework for clustering textual data according to the overall sentiment expressed in their texts. The method is built by combining different statistical and natural language processing techniques and methodologies. Firstly, a SA algorithm

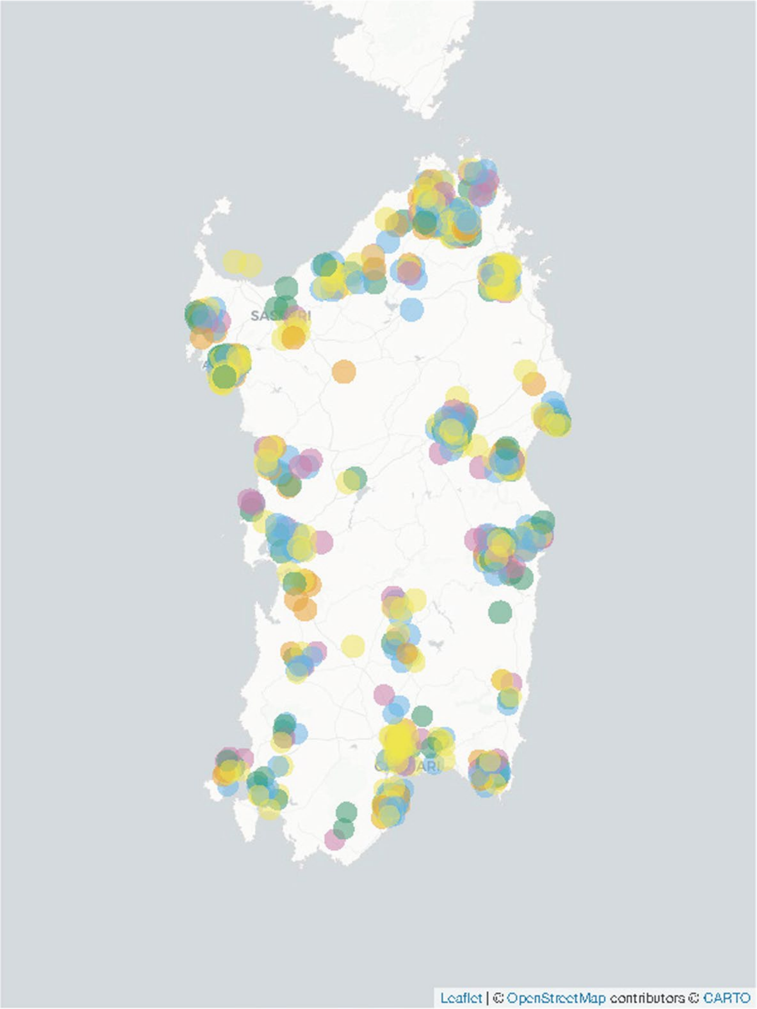

Fig. 4 Map of the Sardinian hotels of *Booking.com* dataset. Each point corresponds to a hotel and assumes a different color according to its own cluster of belonging

is used to categorize natural language texts into positive/negative classes. Secondly, Tb-NB is applied to refine SA's output and to provide a numerical output. Specifically, Tb-NB defines a refined general sentiment of the texts and allows defining the specific sentiment values of the topics discussed in them. Thirdly, NeSSC is carried out to partition the instances into *K* disjoint groups. It defines the pairwise affinity of the instances through a classifier and uses this information to organize them into a complex network. Finally, a community detection algorithm partitions the instances into communities (i.e., clusters) by minimizing the overlapping of the overall sentiment as much as possible.

The framework is then applied to a real case motivating example, using *Booking.com* dataset concerning hotel structures of Sardinia island (Italy). The results highlight the feasibility and interpretability of the framework with real data: we were able to find well-defined, non-overlapping, and interpretable clusters among the hotel structures. The implication of the results from the managerial point of view highlighted actionable insights spendable in decision-making support.

Author contributions All authors have contributed equally to the work.

Funding Open access funding provided by Università degli Studi di Cagliari within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Agarwal B, Mittal N (2016) Machine learning approach for sentiment analysis. Springer, Cham, pp 21–45

Baek S, Jung W, Han SH (2021) A critical review of text-based research in construction: data source, analysis method, and implications. Autom Constr 132(103):915

Bair E (2013) Semi-supervised clustering methods. Wiley Interdiscipl Rev Comput Stat 5(5):349–361

Basu S, Bilenko M, Mooney RJ (2004) A probabilistic framework for semi-supervised clustering. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, pp 59–68

Bilenko M, Basu S, Mooney RJ (2004) Integrating constraints and metric learning in semi-supervised clustering. In: Proceedings of the 21st international conference on Machine learning, p 11

Darwich M, Mohd SA, Omar N et al (2019) Corpus-based techniques for sentiment lexicon generation: a review. J Digit Inf Manag 17(5):296

Eguchi K, Lavrenko V (2006) Sentiment retrieval using generative models. In: Proceedings of the 2006 conference on empirical methods in natural language processing, pp 345–354

Frigau L, Contu G, Mola F et al (2021) Network-based semisupervised clustering. Appl Stoch Model Bus Ind 37(2):182–202

Gaikwad SV, Chaugule A, Patil P (2014) Text mining methods and techniques. Int J Comput Appl 85(17)

Gandomi A, Haider M (2015) Beyond the hype: big data concepts, methods, and analytics. Int J Inf Manag 35(2):137–144

Gao J, Tan PN, Cheng H (2006) Semi-supervised clustering with partial background information. In: Proceedings of the 2006 SIAM international conference on data mining, SIAM, pp 489–493

Goldberg AB, Zhu X (2006) Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In: Proceedings of TextGraphs: the first workshop on graph based methods for natural language processing, pp 45–52

Goldberg Y (2017) Neural network methods for natural language processing. Synth Lect Hum Lang Technol 10(1):1–309

Gowda HS, Suhil M, Guru D, et al (2016) Semi-supervised text categorization using recursive k-means clustering. In: International conference on recent trends in image processing and pattern recognition. Springer, pp 217–227

Grira N, Crucianu M, Boujemaa N (2004) Unsupervised and semi-supervised clustering: a brief survey. A review of machine learning techniques for processing multimedia content 1:9–16

Hang W, Choi KS, Wang S et al (2017) Semi-supervised learning using hidden feature augmentation. Appl Soft Comput 59:448–461

He Y, Zhou D (2011) Self-training from labeled features for sentiment analysis. Inf Process Manag 47(4):606–616

Hu X, Tang J, Gao H, et al (2013) Unsupervised sentiment analysis with emotional signals. In: Proceedings of the 22nd international conference on World Wide Web, pp 607–618

Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv (CSUR) 31(3):264–323

Kaur H, Mangat V et al (2017) A survey of sentiment analysis techniques. In: 2017 International conference on I-SMAC in social mobile, analytics and cloud (IoT) (I-SMAC), IEEE, pp 921–925

Kestler HA, Kraus JM, Palm G, et al (2006) On the effects of constraints in semi-supervised hierarchical clustering. In: IAPR workshop on artificial neural networks in pattern recognition, Springer, pp 57–66

Kleiber A (2002) An {R} package for testing for structural change in linear regression models. An {R} Package for Testing for Structural 7(2)

Krijthe JH, Van der Maaten L (2015) Rtsne: T-distributed stochastic neighbor embedding using barnes-hut implementation. R package version 013. https://github.com/jkrijthe/Rtsne

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

Madhoushi Z, Hamdan AR, Zainudin S (2015) Sentiment analysis techniques in recent works. In: 2015 science and information conference (SAI), IEEE, pp 288–291

Nanli Z, Ping Z, Weiguo L, et al (2012) Sentiment analysis: a literature review. In: 2012 International symposium on management of technology (ISMOT), IEEE, pp 572–576

Nigam K, McCallum AK, Thrun S et al (2000) Text classification from labeled and unlabeled documents using EM. Mach Learn 39(2):103–134

Nogueira BM, Tomas YKB, Marcacini RM (2017) Integrating distance metric learning and cluster-level constraints in semi-supervised clustering. In: 2017 International joint conference on neural networks (IJCNN), IEEE, pp 4118–4125

Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends® Inf Retrieval 2:1-2(1-2):1–135

Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76(3):036106

Sadr H, Pedram MM, Teshnehlab M (2020) Multi-view deep network: a deep model based on learning features from heterogeneous neural networks for sentiment analysis. IEEE Access 8:86984–86997

Schuckert M, Liu X, Law R (2015) A segmentation of online reviews by language groups: how English and non-English speakers rate hotels differently. Int J Hosp Manag 48:143–149

Silva NFFD, Coletta LF, Hruschka ER (2016) A survey and comparative study of tweet sentiment analysis via semi-supervised learning. ACM Comput Surv (CSUR) 49(1):1–26

Sodanil M (2016) Multi-language sentiment analysis for hotel reviews. In: MATEC web of conferences, EDP Sciences, p 03002

Sparks BA, Perkins HE, Buckley R (2013) Online travel reviews as persuasive communication: the effects of content type, source, and certification logos on consumer behavior. Tour Manag 39:1–9

Sun Z, Ye Y, Zhang X, et al (2012) Batch-mode active learning with semi-supervised cluster tree for text classification. In: 2012 IEEE/WIC/ACM international conferences on web intelligence and intelligent agent technology, IEEE, pp 388–395

Taboada M, Brooke J, Tofiloski M et al (2011) Lexicon-based methods for sentiment analysis. Comput Linguist 37(2):267–307

Vallejo-Huanga D, Morillo P, Ferri C (2017) Semi-supervised clustering algorithms for grouping scientific articles. Procedia Comput Sci 108:325–334

Van Der Maaten L (2010) Fast optimization for t-sne. In: In 2010 Workshop on challenges in data visualization neural information processing systems (NIPS)

Van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9(11)

Vashishtha S, Susan S (2019) Fuzzy rule based unsupervised sentiment analysis from social media posts. Expert Syst Appl 138(112):834

Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008

Wan X (2009) Co-training for cross-lingual sentiment classification. In: Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP, pp 235–243

Wang X, Wei F, Liu X, et al (2011) Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: Proceedings of the 20th ACM international conference on information and knowledge management, pp 1031–1040

Xiang G, Min W (2010) Applying semi-supervised cluster algorithm for anomaly detection. In: 2010 3rd International symposium on information processing, IEEE, pp 43–45

Yi J, Zhang L, Yang T, et al (2015) An efficient semi-supervised clustering algorithm with sequential constraints. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1405–1414

Yoshida T (2014) A graph-based approach for semisupervised clustering. Comput Intell 30(2):263–284

Yu Z, Luo P, You J et al (2016) Incremental semi-supervised clustering ensemble for high dimensional data clustering. IEEE Trans Knowl Data Eng 28(3):701–714

Zeileis A, Kleiber C, Krämer W et al (2003) Testing and dating of structural changes in practice. Comput Stat Data Anal 44(1–2):109–123

Zhang W, Tang X, Yoshida T (2015) Tesc: an approach to text classification using semi-supervised clustering. Knowl-Based Syst 75:152–160

Zhu X, Goldberg AB (2009) Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning 3(1):1–130