





ARTICLE



<https://doi.org/10.1057/s41599-023-01711-0>

OPEN

Topic detection with recursive consensus clustering and semantic enrichment

Vincenzo De Leo ^{1,2}✉, Michelangelo Puliga^{1,3}, Marco Bardazzi^{4,6}, Filippo Capriotti^{4,7}, Andrea Filetti⁴ & Alessandro Chessa ^{1,5}

Extracting meaningful information from short texts like tweets has proved to be a challenging task. Literature on topic detection focuses mostly on methods that try to guess the plausible words that describe topics whose number has been decided in advance. Topics change according to the initial setup of the algorithms and show a consistent instability with words moving from one topic to another one. In this paper we propose an iterative procedure for topic detection that searches for the most stable solutions in terms of words describing a topic. We use an iterative procedure based on clustering on the consensus matrix, and traditional topic detection, to find both a stable set of words and an optimal number of topics. We observe however that in several cases the procedure does not converge to a unique value but oscillates. We further enhance the methodology using semantic enrichment via Word Embedding with the aim of reducing noise and improving topic separation. We foresee the application of this set of techniques in an automatic topic discovery in noisy channels such as Twitter or social media.

¹Linkalab, CoMPLeX SySTeMS CoMPuTaTioNaL LaBoRaToRy, Viale Elmas, 142, 09122 Cagliari, Italy. ²Department of Mathematics and Computer Science, University of Cagliari, 09121 Cagliari, Italy. ³DLT Science Foundation, London, UK. ⁴Eni S.p.A, Piazzale Enrico Mattei, 1, 00144 Rome, Italy. ⁵Data Lab, LUISS University, Viale Romania, 32, 00197 Rome, Italy. ⁶Present address: Bea Media Company S.r.l., Rome, Italy. ⁷Present address: Chora Media, a Be Content S.r.l. brand, Milan, Italy. ✉email: vincenzo.deleo@linkalab.it

Introduction

The Twitter micro-blogging platform introduced a new communication standard where a 'tweet' is the atomic unit that, in 140 characters (now 280), conveys a world of possibilities, with mentions, hashtags and links to external media that create user engagement. The study of Twitter gained recently a great attention in the scientific community, given the increasing importance that social networks have in everyday life (Lazer et al., 2009). Indeed, more and more people get informed through social networks and uses twitter as a political arena (Tumasjan et al., 2010) and to interact with companies and regulators (O'Connor et al., 2010). The facility of use of this kind of media was also responsible for the explosion of the phenomenon of fake news (Cadwalladr et al., 2017; Lazer et al., 2018). Detection of meaningful information in tweets amongst fake users (bots) (Caldarelli et al., 2020) and fake content (Agarwal et al., 2011) became then an important scientific question addressed by looking at the topology of connections as well as the semantic of the content.

Building a reliable system for topic detection (TD) on stream media like Twitter is an open research question. As the language associated with tweets tends to be concise, sometimes demanding to the hashtags the entire meaning of a sentence, the task of an automatic understanding is a major challenge along with automatically following a discussion (*tracking* a topic (Mahmud et al., 2018)). The huge amount of information created, counting millions of tweets per hour, makes these task highly computationally demanding. A partial mitigation of the tracking challenge relies on considering the "trending topics" by looking only to the hashtags that are easier to follow for automatic systems and still carry their strong semantic meaning. Unfortunately, however, hashtags can mutate along their trajectory in time: slightly different versions of the same word appear, or the same hashtag can be used in non-conventional ways by adding irony or sarcasm making this aspect another technical problem in Big data streams (Bharti et al., 2016).

When dealing with tweets, practitioners have to focus on several tasks at once: (a) tracking discussions grouping tweets into live clusters (b) filtering unwanted content and noise from the tweets (c) making an accurate TD deciding about the optimal number of topics (d) improve the topic purity and expand semantically the extracted topics for a better understanding of the subjects as tweets can be too short. All these goals are, in part, conflicting each other: for instance aggressive noise filtering—usually done removing low-frequency words—can change the topic representation, its purity and even how many topics were recovered in the corpus. Vice versa with a low level of noise filtering tracking topics, that rely on text similarity, can be difficult as unwanted terms in the short tweets can make two unrelated tweets appearing similar. These problems are reflected in the literature on TD that rarely addresses more than a single issue at once, papers tend to divide into TD methods (Likhitha et al., 2019), or tracking systems (Xu et al., 2019), or unsupervised clustering techniques (Haribhakta et al., 2012).

In natural language processing (NLP) topic detection is traditionally performed by using two different approaches: supervised and unsupervised. In the first case we know the topics in advance and we want to classify the documents assigning one or more topic per document, in the second case we are trying to automatically extract them from the documents using a collection of words that are the most representative of the content. The terms that describe each topic are somehow imprecise and only suggestive of the overall content. Another typical problem in this field is identifying the optimal number of topics in a totally automatic way. A low number of topics will create a situation where the words of each topic describe mixed concepts, vice versa a high number of topics will make several topics poorly defined

and plenty of ambiguities. The procedure to extract topics from a corpus involves the creation of a term-document matrix, often in the form of a large and sparse Tf-Idf matrix (term frequency inverse document frequency). This matrix with $N \times M$, N documents per M words must be reduced to a more tractable and smaller, and denser matrix of $l \times w$, l topics per $w \leq M$ words.

The *optimal* number of topics l is traditionally decided in advance (see Krasnov et al. (2019), and Arun et al. (2010)), in fact methods such as LDA (Latent Dirichlet Allocation) or NMF (Non-Negative Matrix Factorization) work with this property as input and allocate the best word partition. Other more modern techniques such as HDP (Hierarchical Dirichlet process, introduced in Teh et al. (2005)) can avoid selecting the initial number of topics trying to optimize the distribution of words among different groups (the topics). However, a quick look at these techniques show that they are bounded to hyperparameter tuning. With respect to HDP we have to set: α the concentration coefficient of Dirichlet Process for document-table, η the hyperparameter of the Dirichlet distribution for topic-word, and γ the concentration coefficient of Dirichlet Process for table-topic. Finally being the method a probabilistic one every run it produces slightly different results, meaning that a word can move from a topic to another one, and more importantly, the estimated number of topics can vary a bit around an average value.

All these methods lack of *stability*: no topic has always a consistent and stable set of words that move from a topic to another one, and the number of topics changes accordingly with some degree of randomness. In theory a good topic partition must have both (a) an optimal number of topics (b) a core of words that stay on the same topic for the majority of the simulation runs, and for different starting parameters. Within this framework only a robust (i.e., consistent over hyperparameter tuning) and *stable* partition of words (Greene et al., 2014) that describes each topic must be considered the optimal partition.

In literature a stability analysis for different values of the l initial topics has been introduced by Levine et al. (2001) with the goal of estimating the best partition. Among the possible measures of agreement, in this work we utilize the *consensus matrix* (see Strehl et al. (2003)) that measures the tendency of the words of a given topic to remain in the same clusters in each TD estimation. When the initial data is perturbed, or the parameters of the TD methodology were slightly modified, a consensus matrix can be used to measure the stability of topics. The most stable solution of the consensus matrix partition will be the optimal solution.

Starting from the existing literature, our contribution to the actual research can be divided in two parts. In the first one we propose a technique of optimal topic number detection, manipulating the term-document matrix as a starting point for TD, and the consensus matrix and its clustering in order to find the best topic-term partition (the most stable). We introduced a recursive procedure that searches for a stable value for number of clusters in the consensus matrix and at the same time in the TD factorization. In the second part of the paper we make use of the *word2vec* tool to semantically enrich the tweets with synonyms and related words. We demonstrate that this operation is able to improve the quality of the TD. Intuitively, expanding each tweet with more related words will make the documents longer and potentially more meaningful, with our matrix manipulation better suited for TD. We believe that these techniques are an initial step toward a better topic detection in a complete unsupervised fashion.

Methods

The study of topic detection and topic purity we performed in this work involved several steps: (a) the creation of the Twitter

dataset with filtering methods from NLP (b) the definition of a recursive procedure for TD that searches for the most stable partition of the consensus matrix obtained by the TD methodology (c) the demonstration with tests and examples that the semantic enrichment via word2vec will acts on the topic-term matrix H by improving the topics' purity. In short terms we applied a combination of clustering and topic detection to get the optimal topic number, with a semantic enrichment as a final step to deal with the purity of each topic.

Matrix rank reduction. Several methods for matrix rank reduction exist, among them we cite the LDA (Latent Dirichlet Allocation) (Blei et al., 2003), and pLSA (Hofmann, 1999) that uncovers the composition of the topics with probabilistic models (the LDA uses the prior information of a Dirichlet distribution to model the term-document matrix). These topic model techniques work well for corpora with long documents. However, experiments on short texts, and tweets in particular showed that the performance are poor (see Hong et al. (2010)). Intuitively when the number of words of each document is small (like in the tweets) the original term-document matrix representation $W(m \times n)$ (M documents by N words) becomes extremely sparse and the learning procedure performed by LDA or pLSA tends to be inefficient in capturing the topics; this is especially true if the number of topics K is much larger of the size of each document.

An alternative technique to the pLSA or LDA methods is the Non-Negative Matrix Factorization NMF that solves a rank reduction problem using an optimization technique on the decomposing the original term-document matrix $W \simeq V \cdot H$ into a document-topic matrix V and a topic-term matrix H using a norm (usually the Frobenius Norm) and a minimization function on it.

While the experiments show that NMF is fast and robust for short text, the sparseness problem can still reduce the ability of getting high-quality topics (Yan et al., 2013). Indeed while the size of a corpus of tweets can be large, the number of unique words on it grows much slower in a logarithmic fashion. The authors proposed to change the document-term matrix into a term-term matrix and on top of it building a NMF factorization. The term-term matrix, built from co-occurrence of words, has the advantage of being more dense, compact in size, and more stable in topic reconstruction.

Clustering techniques. Studying the topics can be done also using ordinary clustering techniques like K -means, creating cluster of documents (by similarity) that can further be described by the most frequent words or by another operation of topic detection. While this method is fast and easy to perform the K -Means suffers for the curse of dimensionality: for large matrices the Euclidean metrics is worse in retrieving the clusters (Steinbach et al., 2004). A more modern method for clustering is the HDBSCAN (hierarchical, density-based clustering with noise) procedure (Campello et al., 2013; McInnes et al., 2017) that is based on defining clusters according to their local density, while introducing also the concept of noise: not all points belong to a cluster, several ones will be outside and will be classified as noise. The HDBSCAN is different from the DBSCAN methods as the former defines a typical scale for the clustering density: in the latter the only requirement is specifying a minimum number of points n in each cluster.

Although clustering is an attractive way of grouping documents by similarity, a methodology to extract the concept out of clusters is still needed, and it can be a TD like NMF or LDA or even a simple *most frequent word* approach. It is worth also to remember that topics tend to appear in mixtures, and if each document has more than a topic, we can need a fuzzy clustering

(see the classical survey of Yang (1993)) approach if we need to group documents according to topics and not simply by similarity. Interestingly for short documents, and tweets in particular, the topic mixture is a minor issue as it is unlikely that each tweet carries more than a topic on it.

Semantic expansion. Any further step to improve the TD procedure must take into account semantic insights on each document. The typical representation of a topic with document-words lacks of precision. For instance a sentence like "The dog and the owner play on the field" can naturally suggest concepts like "promenade, funny moments, play and catch game" that cannot be derived from TD. Only a semantic expansion of the text, with the help of external knowledge, can complement the TD with more information. A semantic expansion of the NMF methodology has been proposed by Shi et al. (2018). The authors add the context of each word via word embedding directly in the Non-Negative Matrix Factorization. Their proposed algorithm "Semantic aware Non-Negative Matrix Factorization" (SeaNMF) is suggested as a technique to successfully deal with short texts. However, the construction of a good embedding dataset, context-aware for all possible short texts (tweets are a good example) is still an open question, and the authors do not suggest how to create this sample. One tool for semantic expansion is Word2vec: this tool (Mikolov et al., 2013) uses a large corpus, like Wikipedia, to create semantic relationships among words that take the form of a dense vector space. Each word is mapped to a relatively short vector (100:300 components), and semantically similar words are represented by close points in this multidimensional space. The tool can be used to complement each word with synonyms and similar words with a *most similar* query.

Dataset preparation. Creating a corpora on Twitter is a relatively simple task, a user needs only to select one or more words and use them as filters to search and extract relevant tweets. The choice of the search keywords is fundamental, popular and frequent words tend to create mixed topics, specialized words often refer to pure topics.

In this work we started by selecting 20 company names (see Table 1), chosen to be popular in their sectors (automotive,

Table 1 The list of companies with sector and number of tweets that we used as corpora source.

Name	Sector	ntweets
FIAT	automotive	9558
DEUTSCHEBANK	banking and finance	709
VODAFONEUK	telecommunication	126556
JPMORGAN	banking and finance	22434
MONSANTOCO	chemical	20094
ENI	oil and gas	54030
BP_PLG	oil and gas	2945
MELEGATTI	food	8987
KELLOGGSUS	food	20671
MONCLER	clothing	36879
BANCA_MPS	banking and finance	1093
VW	automotive	27909
TIM_OFFICIAL	telecommunication	77837
NOKIA	telecommunication systems	12163
BAYER	chemical	4633
BARILLA	food	42737
ROCHE	pharma	2445
RBS	banking and finance	215
SHELL	oil and gas	23513
MCDONALDS	food	73779
SAMSUNG	telecommunication systems	9433

technology, oil and gas, research, telecommunication). The advantages of using a database like this one is that each corpus will be coherent with the narrative of a company and the relative topic will be pure to some degree.

Tweets mentioning the companies are in the interval 2009-2019 offering enough data to populate many topics but not too many as the result of following the global Twitter stream for years. Depending on the popularity of the companies the number of the tweets per company can vary from few hundreds to more than 30k: the corpus have different length and the tweets were processed with tools from Natural Language Processing (NLP) software packages (in particular the NLTK Python package). The processing involves the following steps:

- **Language identification.** Each tweet has been classified by its language using of a pre-trained machine learning python library, named LangID, that is able to guess the correct language with a high accuracy. For the sake of simplicity in this work only tweets in English were considered.
- **Lower-casing and tokenization** The package TwitterTokenizer from the nltk library has been created to tokenize the tweets while preserving special objects such as emoticons. It is also able to recognize, and take care of several mistakes (for instance it adds an extra space before `http` to remove words containing the term “http” in it)
- **Mentions and hashtags, links removal.** Those special objects were removed to use only the information of plain English words.
- **Stopwords removal.** The most frequent and ordinary words of the English language were removed via stopwords removal.
- **Punctuation removal.** It is generally safe for short Tweets to perform punctuation removal as it is unlikely that tweets are made from more than a sentence. Punctuation removal is simplified by the Twitter tokenizer of NLTK.

Further possible extra steps in the cleaning procedure are available:

- **Stemming and Lemmatization.** This aggressive text transformation transforms each word to the original root lemma. For instance *interesting*, and *interested* become *interest*. We skipped this step as, from experimentation the short text of tweets will be affected too much by the lemmatization.
- **Very short text/word removal.** In the case of tweets a simple noise filtering procedure can be done by filtering tweets that have less than 2 words and/or removing the words that are shorter of than 3 characters and are not stopwords. The importance of short tweets, containing only words like “go” for topic detection is negligible.
- **Retweet and duplication removal.** In this case to avoid duplicates we should remove the “rt” (acronym for retweet) leaving only the original content if available.

Non-Negative Matrix Factorization and the consensus clustering. A popular technique for topic detection TD of the Twitter corpora is the Non-Negative Matrix Factorization (NMF or NNMF) (Sra et al., 2006) that is a special decomposition of a matrix representing words and documents, in components with lower rank. Let be W a matrix of $n \times m$ that, in the domain of text analysis, represents a tf-idf matrix where n is the number of documents and m is the corresponding size of the dictionary (the set of all distinct words present in the document). Usually W is a sparse and large matrix, to be decomposed in two matrix

V and H , with the former being the document-topic matrix and the latter representing the topic-words matrix.

We can write:

$$W_{n \times m} = V_{n \times l} \cdot H_{l \times m} \quad (1)$$

where l , the number of topics can be much lower than m (the size of the dictionary). For each topic in H are associated the frequencies of the m words in the dictionary.

To solve the equation (1) a recursive technique based on the following Frobenius norm:

$$\begin{aligned} \|W - V \cdot H\|_F^2 &= 0 \\ V &\geq 0 \\ H &\geq 0 \end{aligned} \quad (2)$$

usually the norm takes the form of a squared error function or a Kullback-Leibler divergence (Kullback et al., 1951) and the solution can require some sort of regularization to fight numerical instability. In fact the NMF algorithm can be tuned with the choice of two parameters $0 < l_1 < 1$ and α where the first is a regularization parameter that controls the metric used in the numeric solver (it can be L1 or L2, absolute value of square penalization); α is a multiplicative constant.

Finally from the recovered topic-words matrix H we can select per each of the l rows (the topics) a subset of words $m^* < m$ according to their weights, and use it to represent the m^* most important words of each topic.

While the NMF methodology is well established in literature, and it is considered a good method for short texts, the dimension l (number of topics) remains a parameter that must be estimated with care. There is no clear indication of the optimal number of topics, and the typical approach relies in direct explorations with several values of l . An interesting criteria for finding the optimal solution is a check of the stability of the m^* most important words selected per topic by the algorithm: the optimal choice of the parameter l arises when the most important words per topic tend to remain stable when varying the regularization parameters.

To assess the stability of the words representing each topic we highlighted here the importance of a widely used statistical technique: the *consensus clustering* (also known as consensus matrix) (Strehl et al., 2003) that introduces a way of studying the persistence of elements in clusters when external regularization parameters vary.

The idea behind the consensus clustering is that, repeating the clustering operation many times with varying NMF regularization parameters, the words that will stay most of the time in the same cluster are likely to be the correct cluster members.

Let be l the number of topics for n tweets (our documents) with usually $l < n$. During each of the k NMF decompositions with varying parameters α and l_1 let be $h_{ij}, j = 1, \dots, k$ the words defining a recovered topic i . We compute the consensus matrix $C(l)$ of l topics and k runs in the following way:

$$C(l) = \begin{cases} K_{ij}(l) & \text{if } i \neq j \\ k & \text{if } i = j \end{cases} \quad (3)$$

where $K_{ij}(l) \leq k$ is the number of co-occurrences of the pair formed by the word $h_s(k)$ and the word $h_t(k)$ in all the k runs of the NMF decompositions when the regularization parameters are varied for fixed l . In other words the consensus matrix counts how many times a pair of words is present in the same topic when the regularization parameters of the NMF decomposition are varied. We call the entries of the C matrix with the name *hits*.

The consensus matrix will show higher entries for those words that appear more frequently together in the same topic. Clustering the consensus matrix can be used as a filtering tool for topics, saving only the most stable words that stay in a topic.

As the number of l topics is fixed in advance, a clustering study on top of the matrix C is useful to recover a candidate number of clusters l' . Comparing l and l' to find a value l^* to which both numbers converge can indicate, in our hypothesis, a stability region and a final optimal number of topics.

In other words we make use of the consensus matrix to find the most stable clusters and discover the optimal number of stable clusters; if the separation among topics is sufficiently stable then the value l^* for which $l' = l$ is the optimal (i.e., stable) number of topics.

To avoid selecting by hand the optimal number of clusters in the K matrix we use the HDBSCAN algorithm (Campello et al., 2013; McInnes et al., 2017), a powerful clustering method that do not ask for number of clusters (like K -means) and it requires, as input parameter, the minimum number of members r in each cluster. To be conservative we fix $r = 2$. Moreover the feature of HDBSCAN of marking several points as noise is interesting too given the noisy nature of the tweets.

To estimate the optimal number of topics that remain stable in the consensus matrix we will use an iterative process described by the following steps: (a) fix number of topics l and compute, using NMF, the average consensus matrix K for a range of regularization parameters $\alpha = 0, .0.1, \dots, 1$. and $l_1 = 0, .0.1, \dots, 1$. (b) with HDBSCAN recover a number of clusters l' (c) insert back the number l' in the NMF computation and check again the clustering of K with HDBSCAN.

When the value of number of topics/number of clusters converges then $l = l' = l^*$ we are in the most stable region of TD and the number l^* can be considered the optimal number.

Word Embedding as a topic enhancement method. Once we found the best candidate for the number of topics we need a technique to filter out the noise of the words that belong to the discovered topics. One possible solution, especially useful with short text, is improving the content of the original documents by adding words (proper nouns, verbs, and adjectives) that have a meaning similar to the original terms. For instance, we can expand the word "cat" with "feline" or the word "dog" with "animal". If we are able to add semantically similar terms to the short tweets we can increase the overall text quality, and hopefully, make the TD more precise.

A technique to enrich the corpora with semantically similar content refers to the Word2Vec algorithm that creates a *word embedding* transformation of the documents (Mikolov et al., 2013). The algorithm will create numeric vectors associated with each term; similar words will be mapped by closer vectors. In the word2vec representation the semantic similarity among words like "cat" and "feline" is represented by a pair of adjacent vectors.

The idea we exploit in this paper is adding the most similar words to proper nouns, verbs and adjectives by randomly choosing a subset of tweets in each corpora and random words in tweets. The enriched tweets will be represented by longer sentences with more related words than the original ones.

To understand how the topics have been changed by semantic enrichment we studied the distribution of term weights in the H (topic-term) matrix when the new terms are added (we call the addition of words "impurity"). The intuition is that, with more semantic similar words, the relative frequency of the most important, and stable, words will increase, making the topic detection more precise. We will show later how this transformation will act on the most fundamental part of the TD procedure: the mapping of terms and documents into the term-document matrix (the $tf-idf$).

To prove our hypothesis on semantic enrichment we need first to average the distributions of weights for the words' topics

resulting from the NMF decomposition and iterative clustering procedure.

While the matrix $H(l, n)$ of the TD procedure is a topic-term matrix, the consensus matrix is by construction a term-term matrix. Let be $C(i) = K_j(i)$, with $j = 1, 2, \dots, n$ being the i -th row of the consensus matrix. If we average by row the C matrix we obtain the average *hits* of the consensus matrix:

$$p(\bar{C}_m(s)) = \frac{1}{m} \sum_{j=1}^m C_j(s), \quad j = 1, \dots, m \quad (4)$$

re-ordering the $\bar{C}_m(s)$ terms we get the hits distribution $p(\bar{K}_m(s))$. The $p(\bar{C}_m(s))$ distribution is controlling, with its shape, how much a topic is well described by its words. Intuitively the most frequent is a pair of words in a cluster the larger will be the corresponding entry of the K matrix.

Computing the distribution of frequency of each word in a cluster/topic we can study the importance of each word. If the first words by frequency prevail over the others the topic will be more pure: peaked distribution refer to purer topics. In practical terms when a topic is well defined the words associated with it will tend to have a larger weight in the matrix K . The primary effect is that clusters on the consensus matrix are better separated.

In principle, we do not know the effect of semantic enrichment but if this procedure will change the shape of the distribution $p(\bar{C}_m(s))$ toward a more peaked distribution we can affirm that the process of semantic enrichment increases the topic purity and the overall quality of TD.

A further, and more intuitive explanation of the effects of the semantic enhancement can be derived from the "tf-idf" observation. The $tf-idf$ formula is:

$$tf - idf = tf(t, d) \times idf(t) \quad (5)$$

where $tf(t, d)$ is the frequency of the term t in the document d , and

$$idf(t) = \log \frac{1 + n}{1 + df(t)} + 1$$

with $df(t)$ defined as the number of documents containing the word t . The idf term normalizes the importance of a word in a document by its relevance in the entire corpus. Interestingly if we set a minimum value for the inverse document frequency, considering only the words t_1, t_2, \dots, t_k that appear in at least $m \geq 1$ documents, we are filtering out those words that are irrelevant to the entire corpus. Imposing a threshold on the idf is a way to select a subset of words T that describe the document.

On the contrary if we extend the texts with more related words we increase the size of T (the *dictionary* representing all unique words of the corpus), we change the corresponding $tf-idf$, and by consequence we also modify the distribution of rows/columns in the consensus matrix $C(i, j)$. If we compare the two cases with different dictionaries $T < T'$ before and after semantic enrichment we will get different term-topic matrices. Since the original topic-term H matrix varies according to the TD regularization parameters of the NMF procedure we end considering the consensus matrices K before and after enrichment.

To better explain the impact of a richer dictionary $T' > T$ on the topic distribution we use the following procedure. From the main corpus of tweets we extract n subcorpora c_1, c_2, \dots, c_n each one with short or long dictionaries.

- From a set of corpora c_1, c_2, \dots, c_n each one with different dictionaries compute the $tf - idf$ with a given frequency threshold d , $idf(t) > d$ for the inverse document frequency matrix. Each corpus is now described by a matrix $tf - idf(c_k)$ of shape $N \times M'$ where N is the number of

documents in the corpus, and $M' < M$ the number of terms after the reduction of the *idf*.

- Change T and associate to each corpus the number of features (unique terms) recovered from the $tf - idf$.
- Compute the consensus matrix $K(l)$ for the NMF for different number of topics $l = 5, 10, 15, 20, 25, 30, 35, 40, 45, 50$.
- Explore how the size of T changes the distribution of the $K(i, j; l)$ entries normalizing each time by the given number of topics l . If the shape of the distribution is getting more peaked less words will describe each topic, and they will appear more frequently together.

we will study the (4) average distribution of words per different levels of semantic enrichment and per different number of topics l . If this distribution is getting more peaked when T is larger this will prove that effect of the enrichment on the topics is a general increase tool for topic quality (purity). Together with the stability of the topics, their purity is the goal of this research.

In summary we explore the impact of semantic enrichment on the consensus matrix using the distribution of its elements normalized by the number of topics that were used to build this matrix. If the distribution, after the semantic enrichment is getting more peaked, we can interpret this result as a prove that the enrichment is increasing the quality of the topics.

Results

A TD technique known as Non-Negative Matrix Factorization NMF works on the document matrix by selecting a subset of words that all together form the topics' mixture attributed to each document. As the initial number of topics is unknown, a methodology called consensus clustering (Xanthopoulos, 2014) will check for the most stable elements in each detected topic when a perturbation of the NMF parameters creates many realizations of the NMF factorization and therefore slightly different output topics. This set of factor matrices, one per each NMF decomposition, can be averaged to find the most stable configuration (the *consensus matrix*), and this object is investigated by using ordinary clustering techniques to recover the most stable sets of words defining a topic. We found during the creation of the consensus matrix (first part of the algorithm) and the clustering procedure (second part of the algorithm) a convergent and sometimes oscillatory behavior of the two operations. Depending from the starting number of topics l the consensus matrix will generate, via clustering, a different number l' of stable clusters, putting back l' into the topic detection phase we will get after few iterations a number l'' that can eventually coincide with the initial l , making this operation circular. In other cases the procedure converges to an optimal number l'' that is the most stable and in this sense the *optimal* number of topics. This finding is illustrated in the next paragraphs.

The topic detection phase is also influenced by the semantics of the documents. The richer the corpus are in terms of semantic the more likely the topics will be optimally defined, in this case the consensus matrix will be more stable and the clusters of words (the topics) less overlapping. To change the stability of the clusters and topics one can perform a semantic enrichment of the corpus via injection of synonyms and close matching words using the word embedding technique (word2vec algorithm).

Consensus clustering and iterative optimal cluster detection.

While there is no indication on the minimum number of topics, the maximum number is clearly limited by the size of the corpus and by its semantic richness. With the help of the iterative algorithm described in the methods we plot how, for various starting values of l number of topics (represented in the x axis),

we get a final value l' (in the y axis). The pairs (l, l') will form for all the corpus we analyzed (40 collections of tweets extracted randomly from the corpus) a plot like the one in Fig. 1 (left). In general larger corpora will have more topics (as expected), and when the number of initial topics l is too large, the algorithm will reduce that number going back to a lower value of l' . It is worth to notice that the larger is a corpus the clearer is the pattern: the behavior of the algorithm that selects the optimal region of stability for the TD indicated by an almost flat maximum in each curve: the region where the l, l' remain similar.

In several cases there is an oscillatory behavior where l an l' if used as starting number of topics will oscillate: the initial value of number of topics will give a number of clusters l' that, put back in the topic detection phase will again produce l clusters in the clustering procedure. Special values of l leading to oscillation are shown in black triangles in Fig. 1 (left). Figure 2 illustrates the two cases of convergence and oscillation.

We can speculate why we obtain this oscillatory solution considering that solutions in nature does not have to be unique. The consensus clustering with HDBSCAN on the matrix K has a stability island at the value l' and another at the value l'' where there are well defined partitions of the words belonging to each topic. In this situation several clusters split or join together, maintaining a high degree of coherence. In depth the explanation is as follows: the HDBSCAN procedure produces an l'' number of clusters, that used as input for the NMF factorization create a new consensus matrix where the HDSCAN is giving back l' (the other stable solution). This new value used as the basis of a new NMF factorization will generate a matrix whose best solution for HDBSCAN is again l'' creating a cycle. Words can be then safely arranged into two relatively stable configurations. It is important to notice that in case we select, as starting point a different number $l''' \neq l'' \neq l'$ of topics for the NMF factorization the procedure will converge to a unique, stable solution. As expressed by Fig. 1 (right) exploring the initial number of topics will tend to for a curve that for a sufficient large corpus saturates to a stable value of optimal number of topics before dropping to a situation where the number of unique words are not sufficient to cover the extremely large number of selected initial topics.

If by chance the NMF procedure with a l' number of clusters as input produces a consensus matrix where the HDBSCAN clustering is giving back l'' , and again this l'' as NMF input will produce l' in the clustering phase, the two most table partitions of the consensus matrix are indefinitely explored by the algorithm. The stability of the two solutions is strong enough to deny the convergence to a single optimal value. In theory if the consensus matrix has more than 2 stable clustering configurations the oscillations can involve more than two solutions, this will be subject to a more in deep research work.

Corpus features distribution. A general result about the topics' word composition can be obtained by checking the distribution of $C(i, j)$, the consensus matrix entries (the "hits"), for corpora represented by different feature sets T (corpus dictionaries, i.e., unique words).

The results of the topic detection have been studied with the help of the consensus matrix K checking the distribution of its values (we call "hits" the elements of the matrix). Each entry of the matrix represents the number of times a pair of words co-occurred in the same topic. This analysis allows us to understand that the richer is a corpus (larger set of features T) the more peaked is the distribution of hits (4) where the hit defines the occurrence of a word in a given topic during the consensus matrix building. In simple words if a corpus is rich of words (it has a rich semantic space) the average number of frequent and important

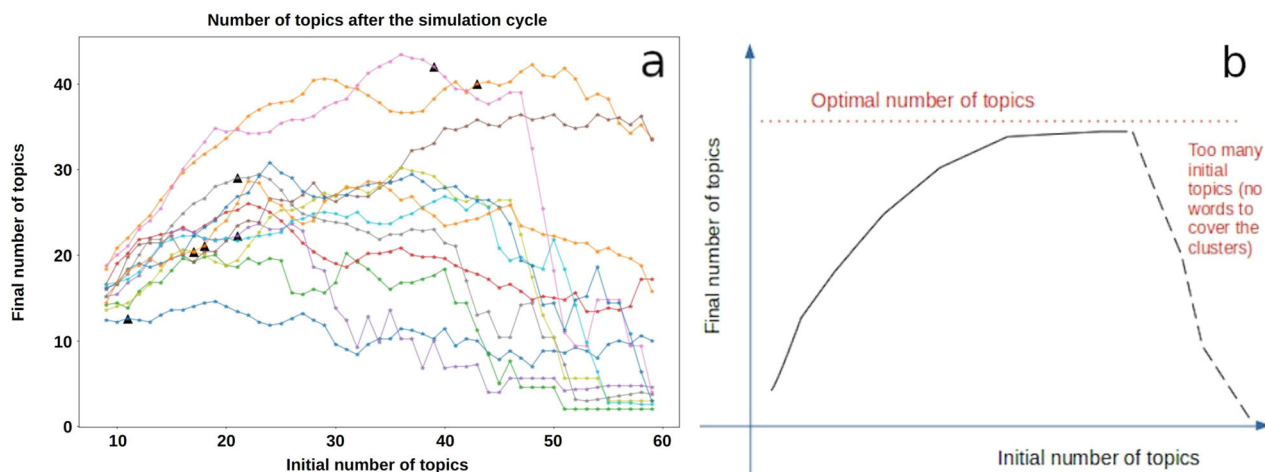


Fig. 1 Number of topics (initial and final) in the iterative procedure of topic detection and clustering of the consensus matrix C . Each line represent a different corpus, the larger is a corpus the higher usually is the number of topics. The black triangles illustrate the special values of the initial topics that oscillate without converging to a fixed point. (right) The schema of the initial, final topic plot: the number of topics grows till a given level then drops for lack of words to fill the individual topics. The highest level is the stability area of the topic partitioning. Each point of the curve in figure “1b” is the average of the points corresponding to the same value of the “Initial number of topics” in figure “1a”.

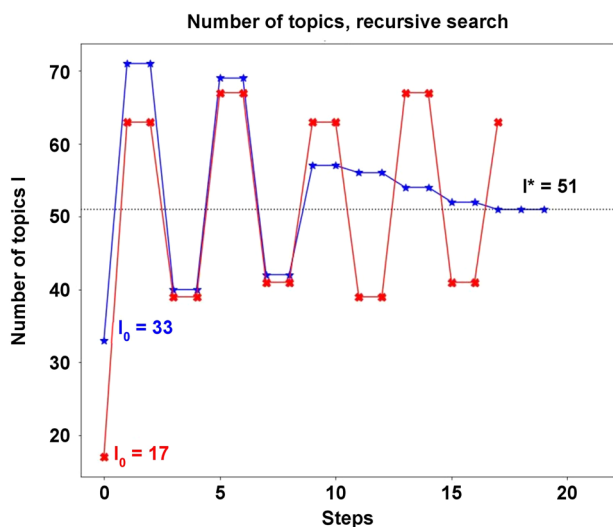


Fig. 2 Recursive search of optimal topic number l^* with the iterative procedure from NMF topic detection and clustering in the consensus matrix $C(i, j)$. In the normal cases the procedure starts from an initial value and converges to a fixed value l^* (blue line), in others (red line) the number of topics and clusters in the consensus matrix oscillates indefinitely.

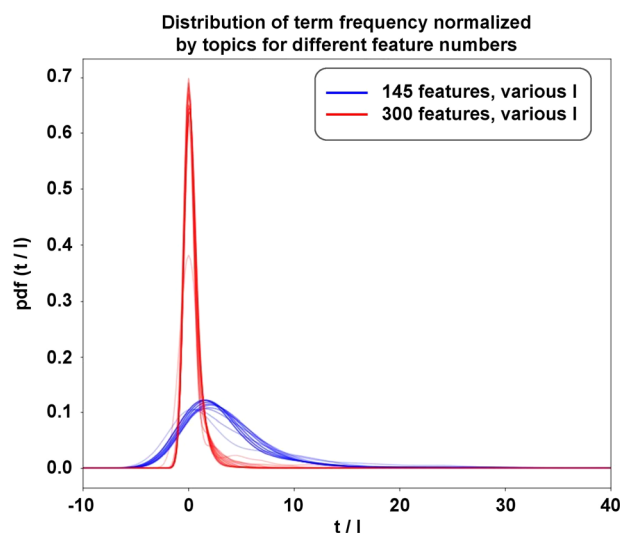


Fig. 3 Distribution of the consensus matrix hits for two cases: a semantic rich corpus represented by 350 features (red), and a poor corpus represented by only 145 features (blue). The distribution of the hits has been rescaled by the number of topics l in the interval 5-50 with steps of 5, individual curves with fixed l appear as tiny lines in each distribution. The effect of the semantic richness of a corpus is evident in the more peaked distribution: the more a corpus is rich the less is the fraction of words needed to describe a topic.

terms that are needed to describe each topic is smaller, and the words describing a topic are less overlapping with other topics.

We show the distribution of hits (normalized by the number of topics of the original NMF decomposition) in Fig. 3. This normalization will allow comparing the result of TD on the hits distribution for a given corpus when we change the linguistic variety (dictionary T). As expected, when the size of T is larger, the distribution of hits $p(t; l)$ on the same corpus but with two different number of features (145 and 350 in this case) is different: the more are features the more peaked is the hits distribution.

We can expand this first result by computing the average value of each hits distribution as in Fig. 3 with the dictionary size T varying continuously in an interval. This time in Fig. 4 we show that, when we have more features, we obtain more peaked hits

distribution. The interpretation of this phenomena is straightforward: the richer is a corpus the better the topics are defined, and in the consensus matrix, the clusters of words appear to be less overlapping. We can also say that, when a corpus is linguistically more complete, the topics are easier to identify and become more pure. This is in turn a clear indication of how the semantic enrichment will act on the TD procedure.

Using Word Embedding for semantic enrichment. The procedure we followed with word2vec for semantic enrichment is described below. For a given number of topics l on the corpus:

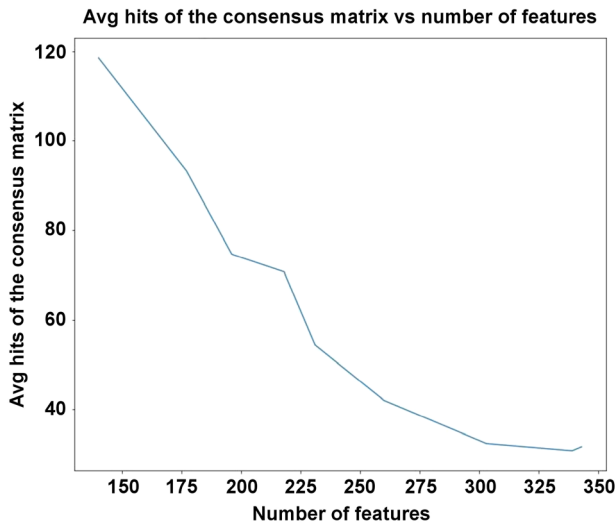


Fig. 4 Average number of hits in the consensus matrix in function of the size of the tf-idf corpora features (number of unique words). Richer corpora shows smaller average hits indicating a more peaked distribution for each topic.

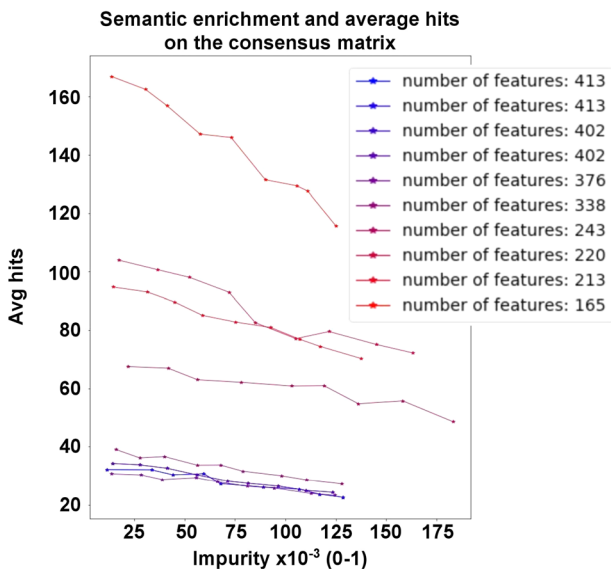


Fig. 5 Semantic enrichment (via word2vec) and average hits on the consensus matrix. Topic on larger and richer corpora are represented by words that have a higher value in the consensus matrix. In all cases the semantic enrichment is sufficient to enhance the quality of the words describing a topic.

- Extract random tweets from the corpus. Clean them from hashtags, links and user mentions
- From each tweet extract random words w only from proper nouns, verbs or adjectives
- Compute the most similar words W_1, \dots, W_k for each extracted word w using a pre-trained word2vec model from the English Wikipedia. We call *impurity* the fraction of enriched words we added over the total words.
- Compute the average number of hits (average value of each entry of the consensus matrix $C(i, j; l)$ see (3)) given the number of topics l
- Repeat this operation for N times changing the level of impurity in the corpus.

Table 2 Example of words extracted from the consensus matrix clustering at the beginning, and after 20 steps.

	Words_run1	Words_run20
Cluster		
-1	go, experience, bought, look, design ...	bus, much, even, cool, tweet, cars ...
0	volkswagen, force, game, super, show	volkswagen, force, game, super, show, tv, bowl
1	chattanooga, ads, new, brand, plant	chattanooga, ads, new, brand, plant, 2015
2	way, wow, favorite	favorite, brand
3	diesel, pretty, car	diesel, pretty, car, jetta
4	hybrid, test, coming, say, diesel	hybrid, test, coming
5	find, take, 2015	find, take, 2015, new, car
6	tomorrow, made, fuel, nice, dealer, ...	tomorrow, made, fuel, dealer, mpg, ...
7	makes, well, oh, always	makes, well, oh, always, vw
8	chance, work	chance, work
9	congrats, 5, 2011	congrats, 5, 2011, 'win,' car

The result of this procedure is shown in Fig. 5: each corpus is identified by its number of features (dictionary) T and its values are represented by lines in the picture. In the x axis we put the impurity defined as the fraction of added words from the word2vec semantic enrichment. In the y axis we report the average number of hits that is simply the average of the entries of the consensus matrix $C(i, j; l)$.

We clearly see two related patterns: (a) increasing the impurity we get a decreasing number of avg hits indicating a more peaked hits distribution. The semantic enrichment has the effect of making the topics more precise, with less words needed to describe a topic (b) we already know that linguistically richer corpora need less words to describe a topic (see also Fig. 4) as they specialize more the subjects.

The effect of semantic enrichment is then a transformation of the corpus toward a situation where the topics are better defined by fewer words or, as a result of this distribution shrinkage, the topics are formed by words that have a smaller overlap (i.e., the clusters are more separated).

We present here in Table 2 a sample of the topics extracted when specifically dealing with the “Volkswagen” corpus with the words from the first run of clustering in the consensus matrix, and after 20 runs. Note the excellent word stability of the topics.

Conclusions

In this work, we explored several aspects behind the Topic detection methodology. Getting the correct number and composition of topics out of a text has proved to be a difficult task, especially for unsupervised techniques on short texts. In fact topics recovered from short text such as tweets tends to be noisy while the optimal number of topics, that summarize the documents, remains undefined.

The TD procedure must deal with the search of this optimal number of topics, or at least define a reasonable interval for this number. Using clustering on the consensus matrix and the NMF factorization with an iterative procedure we were able to find an optimal number of topics that must be seen as the most *stable partition* of the consensus matrix or, in simpler words the steady set of terms that remains in each topic.

Interestingly, in the iterative procedure we found a resonance phenomena: for several values of l^* (the initial number of topics) the algorithm leads to a number of clusters l of the K matrix

number that, once inserted back in the NMF factorization, it will give back the initial value l^* in an endless cycle. This phenomena is due to the presence of two stable solutions in the clustering partitioning of the consensus matrix and it needs to be investigated in a more detailed study as it is present both in large and small corpora.

Once the optimal number of topics has been determined, we can improve how each topic is described by its components words. The optimal topics situation is when each group of words describing a topic has little overlap with the others topics, and its words belongs to specific arguments. One efficient way to improve the purity of topics is by semantic enrichment using word2vec or similar tools. The richer is a corpus the more pure and non-overlapping will be the topics.

The research on unsupervised topic detection is relatively young and more has to be done to understand how to improve the quality of the recovered subjects. One step in this direction is the semantic expansion of the text, however, to which limit extend a text without altering too much the composition of topics till a level that the original meaning will be lost is an open question worth of investigating. Again the complexity of the techniques of TD, the many factors related to the initial NLP procedures, and the vagueness of the concept of “noise” in the domain of text analysis make these studies a challenging field.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Received: 25 June 2021; Accepted: 17 April 2023;

Published online: 04 May 2023

References

- Agarwal A, Xie B, Vovsha I, Rambow O, Passonneau R (2011) Sentiment analysis of Twitter data. In: Proceedings of the Workshop on Language in Social Media. pp. 30–38
- Arun R, Suresh V, Veni Madhavan CE, Narasimha Murthy MN (2010) On finding the natural number of topics with latent dirichlet allocation: some observations. In: Zaki MJ, Yu JX, Ravindran B, Pudi V (eds). Advances in knowledge discovery and data mining. pp. 391–402
- Bharti SK, Vachha B, Pradhan RK, Babu KS, Jena SK (2016) Sarcastic sentiment detection in tweets streamed in real time: a big data approach. Digit Commun Netw 2:108–121. <https://doi.org/10.1016/j.dcan.2016.06.002>
- Blei D, Ng A, Jordan M (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022. <https://doi.org/10.1162/jmlr.2003.3.4-5.993>
- Cadwalladr C (2017) The great British Brexit robbery: how our democracy was hijacked, The Guardian
- Caldarelli G, De Nicola R, Del Vigna F, Petrocchi M, Saracco F (2020) The role of bot squads in the political propaganda on Twitter. Commun Phys 3:81
- Campello RJGB, Moulavi D, Sander J (2013) Density-based clustering based on hierarchical density estimates. In: Advances in knowledge discovery and data mining. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 130–172
- Greene D, O’Callaghan D, Cunningham P (2014) How many topics? stability analysis for topic models. In: Calders T, Esposito F, Hüllermeier E, Meo R (eds). Machine learning and knowledge discovery in databases. pp. 498–513, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44848-9_32
- Haribhakta Y, Malgaonkar A, Kulkarni P (2012) Unsupervised topic detection model and its application in text categorization. In: Proceedings of the CUBE International Information Technology Conference, CUBE ’12. Association for Computing Machinery, New York, NY, USA. pp. 314–319
- Hofmann T (1999) Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’99. Association for Computing Machinery, New York, NY, USA. pp. 50–57
- Hong L, Davison BD (2010) Empirical study of topic modeling in twitter. In: Proceedings of the first workshop on social media analytics. pp. 80–88, Association for Computing Machinery
- Krasnov F, Sen A (2019) The number of topics optimization: clustering approach. Mach Learn Knowl Extract 1:416–426
- Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22:79–86
- Lazer D, Pentland A, Adamic L, Aral S, Barabasi AL, Brewer D, Christakis N, Contractor N, Fowler J, Gutmann M, Jebara T, King G, Macy M, Roy D, Van Alstyne M (2009) Social science. Computational social science. Science (New York, N.Y.) 323:721–3. <https://doi.org/10.1126/science.1167742>
- Lazer DM et al. (2018) The science of fake news: addressing fake news requires a multidisciplinary effort. Science 359:1094–1096. <https://doi.org/10.1126/science.aao2998>
- Levine E, Domany E (2001) Resampling method for unsupervised estimation of cluster validity. Neur Comput 13:2573–2593. <https://doi.org/10.1162/089976601753196030>
- Likhitha S, Harish BS, Keerthi Kumar HM (2019) A detailed survey on topic modeling for document and short text data. Int J Comput Appl 178:975–8887. <https://doi.org/10.5120/ijca2019919265>
- Mahmud H, Orgun M, Schwitter R (2018) A survey on real-time event detection from the twitter data stream. J Inform Sci 44:443–463. <https://doi.org/10.1177/0165551517698564>
- McInnes L, Healy J, Astels S (2017) hdbscan: hierarchical density based clustering. J Open Source Softw 2:205
- Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds). Advances in Neural Information Processing Systems 26. Curran Associates, Inc. pp. 3111–3119
- O’Connor B, Balasubramanian R, Routledge BR, Smith NA (2010) From tweets to polls: linking text sentiment to public opinion time series. In: From tweets to polls: Linking text sentiment to public opinion time series, May, 122–129 (2010)
- Shi T, Kang K, Choo J, Reddy CK (2018) Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In: Proceedings of the 2018 World Wide Web Conference, WWW ’18. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE. pp. 1105–1114
- Sra S, Dhillon IS (2006) Generalized nonnegative matrix approximations with bregman divergences. In: Weiss Y, Schölkopf B, Platt JC (eds). Advances in Neural Information Processing Systems 18. MIT Press. pp. 283–290
- Steinbach M, Ertöz L, Kumar V (2004) The challenges of clustering high dimensional data. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 273–309
- Strehl A, Ghosh J (2003) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617. <https://doi.org/10.1162/153244303321897735>
- Teh Y, Jordan M, Beal M, Blei D (2005) Sharing clusters among related groups: hierarchical dirichlet processes. Advances In Neural Information Processing Systems 17. NeurIPS Proceedings, Massachusetts Institute of Technology Press
- Tumasjan A, Sprenger T, Sandner P, Welpe I (2010) Predicting elections with Twitter: what 140 characters reveal about political sentiment. Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA 10:178–185. <https://doi.org/10.1074/jbc.M501708200>
- Xanthopoulos P (2014) A review on consensus clustering methods. 553–566 Springer New York. pp. 553–566
- Xu G, Meng Y, Chen Z, Qiu X, Wang C, Yao H (2019) Research on topic detection and tracking for online news texts. IEEE Access 7:58407–58418. <https://doi.org/10.1109/ACCESS.2019.2914097>
- Yan X, Guo J, Liu S, Cheng X, Wang Y (2013) Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In: Proceedings of the 2013 SIAM International Conference on Data Mining (SDM). pp. 749–757
- Yang M-S (1993) A survey of fuzzy clustering. Math Comput Model 18:1–16. [https://doi.org/10.1016/0895-7177\(93\)90202-A](https://doi.org/10.1016/0895-7177(93)90202-A)

Author contributions

MP, AC, and VDL performed the analysis, MB, FC, and AF provided support for elaboration, discussion, and theoretical framework in the business processes.

Competing interests

The authors declare no competing interests.

Ethical approval

This article does not contain any studies with human participants performed by any of the authors.

Informed consent

This article does not contain any studies with human participants performed by any of the authors.

Additional information

Correspondence and requests for materials should be addressed to Vincenzo De Leo.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023