Simplifying Text Mining Activities: Scalable and Self-Tuning Methodology for Topic Detection and Characterization

Article

Simplifying text mining activities: scalable and self-tuning methodology for topic detection and characterization

Evelina Di Corso [1], Stefano Proto [1], Bartolomeo Vacchetti [1], Paolo Bethaz [1,*], Tania Cerquitelli [1]

[1] Affiliation 1; firstname.lastname@polito.it
* Correspondence: paolo.bethaz@polito.it

Abstract: In recent years, the number and heterogeneity of large scientific datasets have been growing steadily. Moreover, the analysis of these data collections is not a trivial task. There are many algorithms capable of analysing large datasets, but parameters need to be set for each of them. Moreover, larger data sets also mean greater complexity. All this leads to the need to develop innovative, scalable and parameter-free solutions. The goal of this research activity is to design and develop an automated data analysis engine that effectively and efficiently analyses large collections of text data with minimal user intervention. Both parameter-free algorithms and self-assessment strategies have been proposed to suggest algorithms and specific parameter values for each step that characterises the analysis pipeline. The proposed solutions have been tailored to text corpora characterised by variable term distributions and different document lengths. In particular, a new engine called ESCAPE (Enhanced Self-tuning Characterisation of document collections After Parameter Evaluation) has been designed and developed. ESCAPE integrates two different solutions for document clustering and topic modelling: the joint approach and the probabilistic approach. Both methods include ad-hoc self-optimization strategies to configure the specific algorithm parameters. Moreover, novel visualisation techniques and quality metrics have been integrated to analyse the performances of both approaches and help domain experts to interpret the discovered knowledge. Both approaches are able to correctly identify meaningful partitions of a given document corpus by grouping them according to topics.

Keywords: Textual data; unsupervised learning; self-tuning algorithms

Citation: Di Corso, E.; Proto, S.; Vacchetti, B., Bethaz, P., Cerquitelli, T. Simplifying text mining activities: scalable and self-tuning methodology for topic detection and characterization. *Journal Not Specified* **2022**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Nowadays, modern applications, from social networks like Facebook and Twitter, to digital libraries like Wikipedia, collect more and more textual data. Science is in a data-intensive age in which the creation and sharing of large scientific datasets is unheard of. Indeed, the pace of data analysis has been surpassed by the pace of data generation.

The text mining field focuses on the study and development of algorithms capable of finding meaningful, unknown and hidden information from the growing collections of textual documents. Text mining tools include: (i) grouping documents with similar properties or similar content [1,2], (ii) topic modelling [3,4], (iii) classification models [5], (iv) document summarization [6] and text stream analysis [7].

Each data analytics activity on textual data is challenging, as it is a process with multiple steps in which the analytics pipeline must be configured in order to discover and exploit interesting knowledge from the textual data.

There is no single pipeline to analyse textual data. In the literature, there are several algorithms that can solve a particular data mining task, but in most cases, no algorithm is universally superior. Various aspects affect the performance of the algorithms, such as the cardinality of the input data, its distribution, and the type of knowledge extracted (i.e., the type of analysis to be performed). However, some steps are common to the different pipelines, such as the collection of textual data (i.e., a set of documents of interest). Once the documents are collected, appropriate preprocessing is performed. The latter involves many steps and is an important and critical task that affects the quality of the text mining results.

To perform a particular phase of data analysis, there are a considerable number of algorithms, but for each one, the specific parameters need to be manually set and the results validated by a domain expert [8]. Moreover, real textual datasets are also characterised by an inherent sparseness and variable distributions, and their complexity increases with data volume. In the analytics process tailored to sparse data collections, it is necessary to transform the data appropriately in order to extract hidden insights from them and to reduce the sparseness of the problem. Furthermore, different weighting schemes (e.g., TF-IDF, LogTF-Entropy) can be used to emphasise the relevance of the terms in the collection. Nevertheless, there are several methods and the choice depends on the experience of the domain expert.

At the end, it is not trivial to obtain the best solution that, at the same time, has a reasonable execution time and proper quality results. It is necessary to devise parameter-free solutions that require less expertise in order to lighten the process of analysis of large textual data.

This paper presents ESCAPE (Enhanced Self-tuning Characterisation of document collections After Parameter Evaluation), a new data analytics engine based on self-tuning strategies that aims to replace the end-user in the selection of proper algorithm parameters for the whole analytics process on textual data collections. ESCAPE includes two different solutions to address document clustering and topic modelling. In each of the proposed solutions, ad hoc self-tuning strategies have been integrated to automatically configure the specific algorithm parameters, as well as the inclusion of novel visualisation techniques and quality metrics to analyse the performance of the methods and help domain experts easily interpret the discovered knowledge. Specifically, ESCAPE exploits a data reduction phase computed through the Latent Semantic Analysis, before the exploitation of the partitional K-Means algorithm (named *joint-approach*) and the probabilistic Latent Dirichlet Allocation (named *probabilistic approach*). The former exploits the dimensionality reduction of the document-term matrix representing each corpus, while the latter is based on learning a generative model of term distributions over topics. Both the joint-approach and the probabilistic model permit to find a lower dimensional representation for a set of documents compared to the simple document term matrix. Moreover, the outputs of the two methodologies are disjoint groups of documents with similar contents. In order to compare the results, ESCAPE provides different visualisation techniques to help the analyst in the interpretation of the ESCAPE results. The proposed engine has been tested through different real textual datasets characterised by a variable document length and a different lexical richness. The experiments performed by ESCAPE underline its capability to autonomously spot groups of documents on the same subject, avoiding the user having to set the parameters of the various algorithms and the selection of the most appropriate weighting scheme. This paper introduces a novel self-tuning methodology tailored to textual data collection to democratize the data science on corpora. The main objective is masking the complexity of data-driven methodology by allowing non-expert users to easily exploit complex algorithms in the proper way without knowing the technical details. The innovative aspects of the proposed approach are the following:

1. introduction of an automated data analytics pipeline that compares different algorithms and solutions tailored to textual data collection without requiring technical knowledge;
2. automation of the discovery of unsupervised and relevant topics process together with their characterization in a given corpus of documents;
3. integration of innovative and tailored self-tuning techniques drive the automatic choice of optimal parameters for each algorithm;
4. a novel self-assessment approach of the obtained results seeks the best weighting schema;

5. the implementation of different human-readable visualization techniques intended to facilitate the understanding of the results even for non-expert users;

This paper is organised as follows. Section 2 discusses the state-of-the-art methodologies. Section 3 presents the ESCAPE engine, while Sections 4 and 5 show in detail its main building components and the self-tuning algorithms used. Section 6 thoroughly display the experiments performed on six real text corpora, and also includes the comparison with state-of-the-art methods. Considerations about the obtained results are presented in Section 7. Finally, Section 8 draws conclusions and presents future developments of this work.

2. Literature review

Nowadays, several modern applications, such as e-learning platforms, social networks or digital libraries, are able to collect more and more textual data [1]. However, the exploitation of this data is rather limited. In particular, there are few approaches that are able to perform the analysis automatically and without user involvement. Text mining has been adopted in various sectors over the years, as illustrated in [9]. It is based on algorithms capable of deriving high-quality information from a large collection of documents. Its activities include: (i) grouping documents with similar properties or similar content [1,10] [11], (ii) topic modelling [3,12] [13–17], [18] and detection [19] [20], [21] , (iii) classification models [22,23] [24], (iv) opinion mining and sentiment analysis [25,26], and (vi) document querying [27].

Computational cost is a non-negligible issue when applying the above techniques to a large data collection. To address this issue, there have been several research efforts focused on developing innovative algorithms and methods to support large-scale analytics based on MapReduce [28]. Another improvement has been achieved with Apache Spark [29], which surpassed Hadoop performance due to its distributed memory abstraction, a primary aspect for data analytics algorithms.

In the scientific research, several approaches and solutions have been presented in order to represent, mine and retrieve information [30] from the text sources. Depending on the modelling of the text data and the used techniques, different models have been proposed in the scientific literature: set-theoretic [31] (such as the Boolean models, representing documents as sets of words or phrases), algebraic [1,32,33] (representing documents as vectors or matrices, such as the Vector Space models, the Latent Semantic Analysis, the Principal Components Analysis (PCA) [34] or the Sparse Latent Analysis [35]) and probabilistic [36,37] (such as the Latent Dirichlet Allocation, which represents documents as probabilities of words, or the Probabilistic Latent Semantic Analysis).

Figure 1 provides an overview of the state of the art in topic modeling and recognition methods. Based on the proposed methodology, the studies can be divided into unsupervised and (semi)supervised approaches. The work proposed in [16,17,20,24,38] belongs to the (semi)supervised methods. In [20] the authors propose a framework to improve topic detection based on text and image information. After applying image understanding through deep learning techniques they integrate the results with short textual information. Instead, [24] shows a semi-supervised approach. They present two frameworks: The first models short texts, while the second embeds the first for short text classification. In [16] the authors address topic detection on tweets related to Covid-19 in English and Portuguese. Also in [38] the authors uses as data Covid-19 tweets but they rely on a Naive Bayes classifier and logistic regression. In [17] the authors combine Heterogeneous Attention Network with a DBSCAN algorithm and Pairwise Popularity Graph Convolutional Network in order to detect streaming social event detection and study how they evolve in time.

Another research trend that has emerged in recent years is the integration of word embedding and clustering techniques, as seen in [14,15]. The main idea is to extract word embeddings from models such as BERT and apply clustering techniques to them. A variant of this strategy is proposed in [18]. Here, the authors modify the creation of the word embedding by constraints and then apply a Deep K-means algorithm. In [13], they combine traditional topic models, such as LDA with word embeddings. Other authors instead rely on more traditional approaches and focus their research efforts on other aspects. For example, in [11] the authors focus on the weighting schemes used, while in [12] the focus is on more readable visualization techniques or the implementation of self-optimization algorithms[1]. There are also those that implement topic detection techniques and breaking news detection. For example, in [21], the authors use document pivot and feature pivot techniques in combination with online clustering to understand what happens during a soccer match based on tweets.

Since text mining is a multi-step process that requires specific configurations and parameters for each algorithm involved in the analysis, in most of the work cited above the presence of experts and analysts is required to manage the retrieval process. To overcome this problem, innovative solutions are needed to make the analysis of large data scalable and not supervised by human analysts and data experts more effectively treatable. While ESCAPE exploits some of the techniques seen so far, the features that most of the methodologies mentioned are unable to address are the following: the automatic choice of parameters for the algorithms used, the comparison between different techniques through quality indexes and the graphical visualization of the obtained results. Some preliminary results of ESCAPE have been presented in [1,12,32]. While a preliminary cluster analysis on a collection of documents has been discussed in [32], a step toward a self-tuning joint-approach has been presented in [1], and a preliminary version of the self-tuning probabilistic approach has been proposed in [12] to analyze a large set of documents. However, the study presented here significantly improves our previous works, proposing a complete pipeline including different weighting schemes, different reduction strategies, and topic detection algorithms tailored to textual data collections capable of automatically grouping documents addressing similar topics. Moreover, these results can be displayed graphically using different visualization techniques, allowing the expert to easily characterize and compare each topic.

3. Framework

ESCAPE is a distributed self-tuning engine with the purpose of automatically extracting groups of correlated documents from a collection of textual documents, integrating document clustering and topic modelling approaches. Discovered topics hidden in the collection are shown to the end-users in a human-readable fashion to effectively support their easy exploration.

ESCAPE relies on automatic strategies with the purpose to select proper values for the overall textual data analytics process without the user intervention. The ESCAPE architecture, reported in Figure 2, includes four main components: (i) *Data processing and characterisation*, (ii) *Data transformation*, (iii) *Self-Tuning Exploratory Data Analytics*, and (iv) *Knowledge validation and visualisation*. Below each component is described in detail.

3.1. Data processing and characterisation.

In order to deal with the textual data analysis problem in a more efficient way, ESCAPE includes two steps to transform and characterise the textual corpora: (i) *document processing* and (ii) *statistics definition and computation*. These steps are performed automatically without any user intervention.

Document processing. In this block, five steps are performed sequentially as interrelated tasks:

1. *document splitting*: documents can be split into sentences, sections, or analysed in their entire content, according to the next analytic task. While short documents, such as emails or tweets, are represented with a single vector, longer documents can be decomposed into paragraphs or sentences, hence multiple vectors are required. Choosing the best procedure depends on the goals of the analysis: for the clustering task (as the scope of this paper), the entire document is analysed in its entire content; for sentimental analysis, document summarisation, or information retrieval, smaller units of text like paragraphs might be more appropriate;

	Paper	Data preprocessing	Model Used	Validation	Real Dataset	Special Features	Limitations
Semi/Supervised	Zhang et al. (2019)	word stemming, special character removal	LSTM and LDA	F1-measure	Tweets and image collected by twitter	Considering short text and image information for topic detection	No self tuning strategies implemented, no comparison
	Garcia et Berton (2021)	Stopword removal, special character removal	Sentence BERT, LDA, Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture (GSDMM)	Precision, f1-score	COVID-19 tweets from 17/04/2020 to 08/07/2020, in English and Portugese	(GSDMM), text mining on portugese texts	no self tuning strategies implementend, no comparison of different solutions through quality indexes
	Peng et al. (2021)	noise removal, duplicates removal	Heterogeneous Information Network, DBSCAN, Graph Convolutional Networks	Accuracy, f1-score	social messages from platforms: Sina Weibo and the Twitter of China and Chinese Social Media	a new framework for event indetification with state of the art results	no automatic parameter tuning, no visual representation of the results
	Samuel (2020)	Stopword removal, tokenization, part of speech tagging, parsing, stemming, lemmatization	Naive Bayes, logistic regression	accuracy	COVID-19 tweets from 02/2020 to 03/2020.	Use of exploratory and descriptive textual analytics and of textual datavisualization methods	no self tuning strategies implementend, no multiple approach integrated.
	Linmei et al. (2019)	remove non-English characters, the stop words, and low-frequency words appearing less than 5 times	Heterogeneous information network (HIN) for modeling texts. Heterogeneous Graph ATtention networks (HGAT) to embed the HIN for short text classification	Accuracy	AGNews, Snippets, Ohsumed, TagMyNews, MR and Twitter	it makes full use of both limited labeled data and large unlabeled data	Parameter setting is done manually
Unsupervised	Thompson & Mimno (2020)	documents are tokenized with the spaCy NLP toolkit. Remove frequent words and rare words	Bert to generate embeddings. Then K-means	Word entropy, Coherence, Exclusivity	Wikipedia - SCOTUS - Amaz Reviews	Popular contextualized language models are used	A graphic representation of the identified topics is not provided
	Sia et al. (2020)	each word type is converted to its embedding representation	Word2Vec, ELMo, GloVe, Fasttext, Spherical, Bert for obtaining embedding. K-means, k-medoids, VMFM, GMM to identify topics	NPMI (normalized pointwise mutual information)	20 Newsgroup dataset	First job presenting clustering word embeddings	no self tuning strategies implementend, no graphical representation of the obtained results
	Dieng et al. (2020)	filtering stop words, words with document frequency above 70%, and tokenizing.	ETM (embedded topic model): LDA with word embeddings	Topic coherence, Topic diversity	20 newsgroups corpus; New York Times corpus	Traditional topic models are enriched with word embeddings	parameter setting have to be done manually by expert user
	Abualigah et al. (2018)	tokenization, removing stopwords, stemming	B-hill climbing technique for text clustering	Accuracy, precision, recall, f1-measure	Eight text dataset taken from LABIC	A new weighting scheme is proposed	no self tuning strategies implemented, no comparison of different solutions through quality indexes, no visual representation of the results
	Di Corso et al. (2017)	data weighting strategy and LSI	K-Means	Silhouette - Rand Index - Fmeasure	Wikipedia textual data collections	Self-tuning configuration	no graphical representation of the obtained results,no multiple approach integrated
	Proto et al. (2018)	tokenization, stopwords removal, stemming, data weighting strategy	LDA	Perplexity - Silhouette - Entropy	Wikipedia - Routers	Visualization approach included	it has only one topic modeling methodology integrated
	Fard et al. (2020)	word stemming, stopwords removal	SD2C-Doc, SD2C-Rep, Deep K-means	Accuracy-ARI	20Newsgroup - Reuters 21578 - Yahoo Answer Dataset - DBPedia - AG News	A new framework(SD2C-) for word embedding	no self tuning strategies, no visual representation of the results, no comparison of different solution through quality indexes
	Mamo et al. (2021)	Stopwords removal, Tokenization, Vectorization, Weighting scheme	Online clustering algorithm	Precision - Recall - F1score	Six datasets of football match related tweets	A new real time system ELD, based on on-line clusering	no automatic parameter tuning, no graphical representation of the obtained results, no multiple approach integrated

Figure 1. Overview of related works

2. *tokenization*: it is the process of segmenting a text or texts into tokens (i.e., words) by the white space or punctuation marks within the same split;
3. *case normalisation*: capitalisation is very useful to humans in the reading phase. However, in many analytics tasks, a capital word at the beginning of a sentence should not

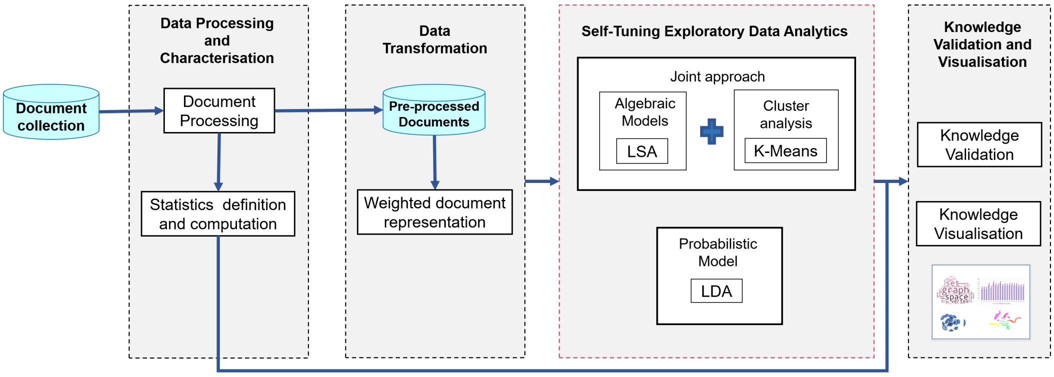


Figure 2. The ESCAPE System Architecture

be treated differently from the same lower case word that appears elsewhere in the document. For this reason, this step converts each token to completely upper-case or lower-case characters;

4. *stemming*: each token is mapped into its own root form. It includes the identification and removal of prefixes, suffixes, and pluralisation;
5. *stopwords removal*: stopwords are the grammatical words which are irrelevant to text contents (e.g. articles, pronouns, prepositions), so they need to be removed for more efficiency. These common words can be discarded before the feature generation process.

The document's main themes are depicted with the Bag-Of-Word (BOW) representation, which shows the most meaningful frequent terms in terms of multiplicity without caring about grammar rules and word order.

Information about the frequency of each word in a document can be useful to reduce the size of the dictionary. For example, the most frequently occurring words in a document are often stop words and should be deleted. Terms that are very rare should also be deleted, as they are often typos. The remaining most common words are the most important and significant. In general, the smaller the dictionary, the greater the intelligence to capture the most important words [39]. Tokenization and stemming are two steps that help us to reduce the size of the dictionary. After defining the set of words, the next step is to convert the document collection into a matrix structure format.

Let $D = \{d_1, d_1, \ldots, d_{|D|}\}$ be a corpus of documents, and $V = \{t_1, t_2, \ldots, t_{|V|}\}$ the set of distinct terms used at least once in the textual collection. The corpus D is represented as a matrix X, named *document-term* matrix, in which each row corresponds to a document in the collection and each column, one for each $t_j \in V$, corresponds to a term in the vocabulary.

Statistics definition and computation. ESCAPE includes the computation of several statistical indices [1,32,40] to characterise the document collection data distribution:

- *# categories*: the number of topics/clusters in the textual collection under analysis (if known a-priori);
- *Avg frequency terms*: the average frequency of token occurrence in the corpus;
- *Max frequency terms*: the maximum frequency of token occurrence in the corpus;
- *Min frequency terms*: the minimum frequency of token occurrence in the corpus;
- *# documents*: the number of textual documents in the corpus (i.e., total number of splits defined by the analyst);
- *# terms*: number of terms in the corpus, with repetitions (i.e., all words of a textual collection);
- *Avg document length*: the average length of documents in the corpus;
- *Dictionary*: the number of different terms in the corpus, without repetition (i.e., all words that are different from each other in a textual collection);

- *TTR*: the ratio between the dictionary variety (*Dictionary*) and the total number of tokens in a textual collection (# *terms*),in other words it represents the lexical diversity in a corpus.
- *Hapax* %: the percentage of Hapax, which is computed as the ratio between the number terms with one occurrence in the whole corpus (Hapax) and the cardinality of the Dictionary;
- *Guiraud Index*: the ratio between the cardinality of the Dictionary and the square root of the number of tokens (# *terms*). It highlights the *lexical richness* of a textual collection.

The joint analysis of these statistical features is able to describe and characterise the data distribution of each collection under analysis. ESCAPE includes also a Boolean feature, named *remove-hapax*, which, if set to *True*, removes the Hapax words for the subsequent analyses, otherwise these words are included in the analysis. This step could lead to different results for the different strategies included in ESCAPE . Indeed, algebraic models are less influenced by the presence of Hapax, as in the decomposition their affection is overridden by the most frequent terms. Probabilistic models, on the other hand, are influenced in a more negative way, as they introduce noise within the creation of the model.

3.2. Data transformation.

This component deals with the representation of weighted documents to emphasise the relevance of specific within the document collection. The weight of each word represents its importance degree. Depending on the weighting scheme adopted, the knowledge acquired from the collection might vary. Specifically, based on the document statistical features and the desired granularity of the outcomes, one of the weighting schemes might outperform the others.

To measure the relevance of the various terms in the document, each cell in the matrix X contains a *weight* x_{ij}, that is a positive real number indicating the importance of the term t_j appearing in the document d_i. [41] propose different weighting functions, combining a local term weight with a global term weight. By applying a weighting function to a collection D, we obtain its weighted matrix X. In particular, each element x_{ij} in the matrix represents the weight of the term t_j in the document d_i and is calculated as the product of a local term weight (l_{ij}) and a global term weight (g_j) (x_{ij}=$l_{ij} \times g_j$). A local weight l_{ij} refers to the relative frequency of a specific term j in a particular document i, while the global weight g_j represents the relative frequency of the specific term t_j within the whole corpus D.

Three local term weights and three global term weights are included in ESCAPE . The local weights are *Term-Frequency* (TF), *Logarithmic term frequency* (Log) and *Boolean*; while the global ones are *Inverse Document Frequency* (IDF), *Entropy* (Entropy) and *Term-Frequency* (TF$_{glob}$). Their definition is reported in Table 1. The TF weight (L1 in Table 1), defined as tf_{ij}, represents the frequency of term j in document i. A similar measure is also reported by Log weight, which, however, evaluates the frequency of the term on a base-2 logarithmic scale. Lastly, the Boolean weight function is equal to 1 if the frequency was non-zero and 0, otherwise. Intuitively, L1 and L2 give increasing importance to more frequent words, but L2 gives progressively smaller additional emphasis to larger frequencies, while L3 is sensitive only to whether the word is in the document.

After establishing the frequency of the different terms in the document the resulting count has to be altered accordingly to the perceived importance of that term by integrating the global importance of each word.

To this aim, the global weighting schemes reduce the weight of those terms that have a high frequency in a single document or appear in many documents, which involves interesting variations concerning the relative importance of document frequency, local frequency and global frequency. In particular, the global weight IDF (G1) measures how rare a term is within the corpora ($|D|$). This weight is calculated as the logarithm of the ratio between the total documents in ($|D|$) and the number of documents df_j containing the term j. The more frequent a term is in the various documents, the lower its IDF will be.

Entropy (G3) represents the real entropy of the conditional distribution given that the term i appeared. In documents, high normalised entropy is considered good and low normalised entropy is considered bad. Entropy as a weighting scheme is the most sophisticated one and it is built on information theoretic ideas. If a term has the same distribution over different documents it gets the minimum weight(i.e. where $p_{ij} = 1/ndocs$), while if a term is concentrated in a few documents it gets the maximum weight. In other words Entropy considers the distribution of terms over documents. Lastly, G3 represents the number of times in which the corresponding word j appears in the entire textual corpus D. It extends L1 considering the whole corpus.

ESCAPE integrates six different term weighting schemes to measure term relevance. We have obtained six of these schemes by combining one of the three local weights (TF, LogTF and Boolean) with either IDF or Entropy, while the last one is the combination between the local Boolean weight and the global TF$_{glob}$ weight. These weighting schemes are the most used in the state-of-the-art [41].

All these combinations are analysed to show how the different schemes are able to characterise the same dataset at a different granularity levels.

Weight	WId	**Definition**
Local	L1	TF = tf_{ij}
	L2	LogTF = $\log_2(\mathrm{tf}_{ij} + 1)$
	L3	Boolean = $\begin{cases} 0 \text{ if } \mathrm{tf}_{ij} = 0 \\ 1 \text{ otherwise} \end{cases}$
Global	G1	IDF = $\log \frac{\lvert D \rvert}{\mathrm{df}_j}$
	G2	Entropy = $1 + \sum_i \frac{p_{ij} \log p_{ij}}{\log \lvert D \rvert}$
	G3	TF$_{glob}$ = gf_j

Table 1. Local and Global weight functions exploited in ESCAPE

4. Self-Tuning Exploratory Data Analytics

Topic modelling and document clustering are closely related and they can mutually benefit one from another [42]. As a matter of fact, topic modelling projects documents into a topic space in order to try to facilitate an effective document clustering. On the other hand, after document clustering, the discovered cluster labels can be incorporated into topic models. In this way specific topics within each cluster and global topics shared by all clusters can be extracted.

Two well-known approaches for document clustering and topic modelling have been integrated in ESCAPE. For each strategy, a brief description is reported, together with ad-hoc self-tuning strategies to automatically configure each algorithm.

4.1. Joint-Approach

The joint-approach includes (i) a data reduction phase computed through the Latent Semantic Analysis [33] based on the Singular Value Decomposition, and (ii) the partitional K-Means algorithm [43]. Below, a brief description of the two algorithms is reported, including their main drawbacks. Lastly, the Subsection ends with the two proposed self-tuning algorithms to automatically set input parameters, respectively.

4.1.1. Latent Semantic analysis

To make the cluster analysis problem more effectively tractable, ESCAPE includes a natural language process named LSA (Latent Semantic analysis) [33]. LSA allows a reduction in the dimensionality of the document-term matrix X which captures the latent semantic structure. Choosing the right dimensionality reduction, while avoiding to lose significant information, is an open research issue and a very complex task. If there are not enough dimensions after the LSA process the data representation will be poor, while if there are too many dimensions it will lead to more noisy data. LSA maps both words and documents in a concept-space where is able to find the relationships between them. To find the hidden concepts, LSA applies the Singular Value Decomposition (SVD). SVD is a matrix factorisation method that decomposes the original matrix (document-term matrix) X into three matrices (U; S; V^T). To find the principal dimensions (K_{LSA}) in X, ESCAPE includes an innovative algorithm named ST-DaRe. Given K_{LSA}, ESCAPE uses only the highest singular K_{LSA} values in S, setting the others to zero. The approximated matrix of X, denoted $X_{K_{LSA}} = U_{K_{LSA}} S_{K_{LSA}} V^T_{K_{LSA}}$ is obtained through the reduction of all three decomposed matrices (U, S, V^T) to rank K_{LSA}. In general, the low-rank approximation of X by $X_{K_{LSA}}$ can be viewed as a constrained optimisation problem with respect to the constraint that $X_{K_{LSA}}$ have rank at most K_{LSA}. When the terms-documents matrix is tighten down to a k-dimensional space, terms with alike co-occurences should be brought together by the SVD. This insight indicates that the dimensionality reduction could improve the results.

Self-Tuning Data Reduction algorithm. The goal of the ST-DaRe (Self-Tuning Data Reduction) algorithm in ESCAPE is to pick out a proper number of dimensions to take into account in the successive analytics steps, while avoiding to lose relevant information, by identifying three reasonable values for the LSA parameter. The correct choice of the number of dimensions to be considered is an open research issue [41]. Selecting the maximum decrease point inside the singular value curve is an easy approach, but if a local minimum is hit the resulting choice would be inaccurate.

The original ST-DaRe algorithm [1] needs three parameters that have been experimentally set. These parameters are the singular value step and two thresholds. In this case, the singular values are plotted in descending order and, from the obtained curve, the singular values are analysed in pairs, using the singular value step set as parameter. For each pair, the marginal decrease of the curve is calculated. If this decrease is comparable to one of the two parameters chosen as thresholds, or to their average, then the smallest singular value of the analysed pair is chosen as one of three values.

Different from this original approach, in ESCAPE we propose a new strategy based on a single parameter T indicating the number of singular values to consider. In particular, after having ordered the singular values in descending order, for our analysis we consider only the first T of them. We calculate the average and the standard deviation for each of these singular values and we define a confidence interval. Then, the three values to choose representing the number of dimensions to be considered are selected in this way: (i) the first is the singular value in correspondence of the mean position, (ii) the second is the singular value in correspondence of the mean plus the standard deviation position, and (iii) the third is the singular value in correspondence of the mean position of the previous ones. Through this method the problem of the local optimality choice is overcome. A pseudo code that shows how the enhanced version of ST-DaRe works, is given in Algorithm 1.

Algorithm 1: The Enhanced ST-DaRe pseudo-code

Input : X, T
Output: $K_{LSA}[3]$

$N = 0$;
// compute the SVD decomposition of the truncated matrix X;
$[U, S, V] \leftarrow X.computeSvd(T)$;
$s \leftarrow normSingularValues(S)$;
// compute the mean of singular values;
$mean = s.mean()$;
// compute the standard deviation of singular values;
$stand\_deviation = s.std()$;
// compute the three values;
$val1 = s[mean]$;
$val2 = s[mean + stand\_deviation]$;
$val3 = s[(val1 + val2)/2]$;
$K_{LSA}.push(val1, val2, val3])$

T at most will be equal to the rank of the document-term matrix. Since the number of documents for all the textual corpora analysed is much smaller than the vocabulary used in each collection, the value T is set by ESCAPE to the 20% of the number of documents.

4.1.2. K-Means Algorithm

In the joint-approach, the singular value decomposition is applied to data to cut down the dimensions of the data prior to the learning process. Since the different document-concept vectors can be clustered, the learning process implements the K-Means algorithm. The difference between clustering and LSA is that clustering algorithms assign each document to a specific cluster, while LSA assigns a set of topics to each document. Still, a K-Means algorithm applied after the singular value decomposition improves the results, as shown in [1,32]. We have decided to implement the K-means clustering because it is an easy algorithm to implement that has good performance and which converges quickly, while providing good results [44], [45]. Moreover, the performance of the algorithm is still being researched in order to obtain better and better results [46], which would allow us easy adaptability in the case of new and better performing techniques.

ESCAPE manages to discover groups of documents that share a similar topic by self-assessing the quality of the found clusters. It uses an algorithm to automatically configure the cluster analysis activity through the analysis of different quality metrics to evaluate the obtained partitions. To this aim, several configurations have been tested by ESCAPE, modifying the specific-algorithm parameter (i.e., number of desired clusters).

Self-Tuning Clustering Evaluation.

After the formation of the K clusters from the collection of textual documents, it is necessary to corroborate the clustering results with three indicators obtained from the computation of the silhouette [47]. The silhouette index gauges from a qualitative point of view the similarity of an element with respect to its own cluster (cohesion) compared to other clusters (separation). The silhouette varies from -1 to +1. If the silhouette has a high value it means that the object is cohesive to its own cluster and well separated from the neighbouring cluster. In order to estimate the cohesion and separation of each cluster set, the solutions found are compared through the calculation of different Silhouette-based indices to measure t. Then the best three configurations, which identify a proper division of the original collection, are chosen. ESCAPE exploits three versions of the standard Silhouette index to assess the quality of the discovered cluster set: (i) the weighted distribution of the silhouette index (WS) [1], (ii) the average silhouette index (ASI) [48] and (iii) the global silhouette index (GSI) [48]. Specifically, WS index indicates the amount of documents in each positive bin properly weighted with an integer value $w \in [1; 10]$ (the highest weight is given to the first bin [1-0.9], and so on) and normalised within the sum of all the weights. It is better to have distributions with a positive asymmetry (i.e., more documents have silhouette values belonging to the higher bins) instead of those with a majority of lower silhouette values (negative skewness). ASI gives an overview of the average silhouette of the entire cluster set, while GSI is able to take into account the possible imbalance number of elements in each cluster. If these indicators have higher values it means that there is a better clustering validity. A detailed description of all the computation of these metrics is reported in Section 5. We apply a rank function for each quality index to estimate the cohesion and separation of each cluster set. The rank assigned to each quality index may vary from 2 (assigned to the solution with the highest Silhouette index) to Kmax (assigned to the solution with the lowest Silhouette index). Then, a global score function is defined as follow:

$$Score = (1 - rank_{GSI}/K_{max}) + (1 - rank_{ASI}/K_{max}) + (1 - rank_{WS}/K_{max})),$$

where K_{max} is the maximum value of clusters, while $rank_{GSI}$, $rank_{ASI}$ and $rank_{WS}$ are the ranks of the Average Silhouette Index, Global Silhouette Index and Weighted Silhouette, respectively. The score lies in the range $[0, (3 - \frac{6}{K_{max}})]$. ESCAPE selects the best value for each experiment. In ESCAPE, the analyst can choose how to set the value of the number of clusters through the setting of a parameter. Nevertheless, our framework proposes as the maximum value for analysis (a default configuration), the average document length for each corpus. In fact, we hypothesize that every word in the document belongs at most to a different topic. In this way, we set an upper-bound for the value of the number of clusters. Still, if the average document length is greater than the number of documents in the corpus under analysis, then the value is set to the average frequency of the term. However, these choices can be changed by each analyst, since the framework is distributed it is able to analyse several solutions in parallel.
Therefore, if the user does not manually specify any parameters at the beginning of the analysis, Kmax is set automatically on the basis of the average document length. Otherwise, the user can set the Kmax parameter according to his needs. In both cases, all solutions in the considered range are explored, in order to choose the three best ones.

4.2. Probabilistic-Approach

ESCAPE includes also the probabilistic topics modelling approach. This technique represents textual documents as probabilities of words and aims to discover and annotate large archives of texts with thematic information. In ESCAPE the Latent Dirichlet Allocation (LDA) is implemented. The intuition behind LDA is that documents are mixtures of multiple topics [3]. Topics are defined to be distributions over a fixed vocabulary. Documents, instead, are seen as a distribution over the set of different topics, thus showing multiple topics in different proportions. LDA requires the number of topics to be set apriori which is a open research issue [12].

4.2.1. Latent Dirichlet Allocation

The Latent Dirichlet Allocation (LDA) is a generative probabilistic model for collections of discrete data such as text corpora [36].

Using Bayesian inference (posterior inference), LDA infers the hidden structure to discover topics inside the collection under analysis. Documents are treated as mixtures of topics and topics as mixtures of words. For each document in the collection, words are generated through a two-stage process:

1. Firstly, a distribution over a topic is randomly chosen.
2. Then for each word in the document:
    a) a *topic* is randomly chosen from the distribution defined at the previous step (Step 1).
    b) a *word* is randomly chosen from the corresponding distribution over the dictionary.

Each document shows topics in different proportions (step 1); then, each word in each document is drawn from one of the topics (step 2b), where the selected topic is chosen from the per-document distribution over topics (step 2a).

In order to generate each document in the corpus, two steps are performed [3]:

1. The choice of the number of terms from a Poisson distribution;
2. After that, for each of the document's words:
   - The choice of a topic z_n from Multinomial($\boldsymbol{\theta}$), where $\boldsymbol{\theta}$ is a Dirichlet($\boldsymbol{\alpha}$), representing the document-topics distribution;
   - The choice of a word w_n from Multinomial($\boldsymbol{\phi}_{z_n}$), where $\boldsymbol{\phi}$ represents the topic-words distribution ($\boldsymbol{\phi} \sim$ Dirichlet($\boldsymbol{\beta}$)), conditioned on the previously chosen topic z_n.

So, if we consider a collection of K topics $\mathbf{z}$, a collection of N terms $\mathbf{w}$ and a document-topics distribution $\boldsymbol{\theta}$, the joint multivariate distribution can be defined as:

$$p(\mathcal{D}|\boldsymbol{\alpha},\boldsymbol{\beta}) = \prod_{d=1}^{K} \int p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_n|\boldsymbol{\theta}_d) p(w_{dn}|z_{dn},\boldsymbol{\beta}) \right) d\boldsymbol{\theta}_{\mathrm{d}},$$

where

- $\boldsymbol{\alpha}$ describes the concentration for the prior placed on documents' distributions over topics (θ). Low α values will create documents that likely contain a mixture of only few topics.
- $\boldsymbol{\beta}$ represents the concentration for the prior placed on topics' distributions over terms. Low β values will likely produce topics that are well described just by few words.

Generally, it is unfeasible to compute these distributions, and thus this posterior Bayesian inferential problem cannot be solved exactly. In order to bypass such an issue it is possible to exploit different approximate inference algorithms: the Online Variational Bayes algorithms [49] is the one that ESCAPE uses, while $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are set to maximise the log likelihood of the data under analysis.

4.2.2. Self-tuning LDA

In literature, different solutions have been explored and proposed in order to find the most suitable K.

Our proposed approach is still iterative, as all the approaches known so far in literature [50]. However, a trade-off between the computational costs and the goodness of the results will be considered, even when applied to large data volumes.

The newly proposed approach, called ToPIC-Similarity [12], is described in detail in the following paragraph.

4.2.3. ToPIC-Similarity

To find the appropriate number of topics into which to divide documents, ESCAPE proposes an automatic methodology called Topic Similarity, whose steps are described by pseudo code in the algorithm 2. After setting a minimum threshold (K_{min}) and a maximum threshold (K_{max}), a new LDA model is generated for each K within the range defined by the thresholds. Each of these models is then evaluated through two main steps:

- *topic characterisation*, to find the n most important words for each of the K topics identified;
- *similarity computation*, to assess the similarity between the various topics found, expressed through an index;

Finally, a third step called *K Identification* allows us to select the best configuration of the K parameter to use in analyses.

Topic characterisation. In this step, each topic identified is summarised with a list of its most significant n words. In order to automatically find the value of n, ESCAPE considers the number of words that appear most frequently, and then filters this number by dividing it by the average frequency of terms within the topic. In particular, the quantity of the most significant words, named Q, is defined as: $Q := \frac{|V| \cdot TTR}{AvgFreq}$, where |V| is the variety of the corpus dictionary and TTR is the Type-Token Ratio (total number of unique words divided by the total number of words). Given Q, the number n is then set as follows:

$$n = \begin{cases} \frac{Q}{K}, & if\ Q \geq K \cdot AvgFreq \\ AvgFreq, & if\ Q < K \cdot AvgFreq \end{cases} \tag{1}$$

When the average frequency of terms in the corpus is higher than the amount of words taken into account, the number n of words is set equal to the average frequency of terms in the corpus. Finally, for each word in each topic, the word is associated with the probability that the term has to be taken up in the topic (0 if it is not included in the list of n words).

Similarity computation. Here all possible pairs of topics are considered and, for each of them, their similarity is calculated. Cosine similarity is used to determine the similarity between two topics. Considering two topics t' and t'' belonging to the same partitioning K, the similarity between the topics is computed as follows: $similarity(t', t'') = \frac{\mathbf{N}_{t'} \cdot \mathbf{N}_{t''}}{\|\mathbf{N}_{t'}\|_2 \|\mathbf{N}_{t''}\|_2}$, where $\mathbf{N}_{t'}$ is the set of the representative words of topic t' and $\mathbf{N}_{t''}$ is the set of of the representative words of topic t''.
At the end of this step a symmetric matrix of dimension K is obtained. The generic cell (i,j) contains the index of similarity between the topic of row i and the topic of column j. The Topic Similarity index for the considered model is obtained by calculating the Frobenius norm of the whole similarity matrix, and dividing the result by K. Finally, since the Topic Similarity is a percentage, the index obtained is multiplied by 100.

K identification. Having calculated the Topic Similarity for each LDA model obtained with a different K, this step illustrates the methodology adopted to identify the best configuration of K. As the value of Topic Similarity decreases when the number of topics increases, two conditions have been set to find the best K:

- the chosen K must be a local minimum of the curve: *Topic Similarity*(K_i) < *Topic Similarity*$(K_{i+1}$;
- the selected value must belong to a decreasing segment of the curve (the second derivative must be positive)

ESCAPE considers the first three values that satisfy these requirements as the best K values to consider. The search ends when three values have been found, or when the considered K is larger than the K_{max} set at the beginning. For each experiment, three well-known statistical quality metrics are reported to characterise the found partitions. In ESCAPE, we have integrated three different measures to assess the quality of the probabilistic model: (i) *Perplexity*, (ii) *Entropy*, and (iii) *Silhouette*. The perplexity [3] indicates how well the probabilistic model represents a sample. A lower perplexity value represents a better model for the analysed collection. The entropy [51] is defined as the amount of information in a transmitted message. Hence a message with high uncertainty indicates a large amount of entropy. Lastly, the silhouette [47] takes into account both the cohesion and the separation of a document. The cohesion represents how similar a document is with respect to its own clusters, while the separation represents how different a document is from documents belonging to other clusters. The Silhouette Index can assume values between [-1, 1], where a value close to 1 indicates that the document is correctly located in the proper cluster.

Algorithm 2: ToPIC-similarity pseudo-code

```
Data: X, K_min, K_max
Result: kSol
// variable inisialisation
topicS = [ ],    NTerms = [ ];
for K ← K_min to K_max do
    // build the LDA model;
    LDAModel ← lda.fit(X);
    Q ← (|V| · TTR) / AvgFreq;
    // set the number of terms per topic;
    if Q ≥ K · AvgFreq then
        n ← Q/K);
    else
        n ← AvgFreq;
    end
    // collect together the terms of each topic;
    for t ← 0 to (K-1) do
        NTerms.append(LDAModel.describeTopics()[t].sort().take(n));
    end
    N ← NTerms.size();
    topicsDescr = zeros(K, N);
    simMatrix = zeros(K, K);
    for t ← 0 to (K-1) do
        for word ← 0 to N do
            // take the probability that the term has to be drawn
            // from the topic, given the LDAModel
            topicsDescr[t][word] ← LDAModel.describeTopics()[t, NTerms[word]];
        end
    end
    for t ← 0 to (K-1) do
        for s ← 0 to (K-1) do
            simMatrix[t][s] ← cosine(topicsDescr[t], topicsDescr[s]);
        end
    end
    topicS.append(Frobenius-norm(simMatrix)*100/K);
    if topicS[K] ≥ topicS[K-1] AND secondDerivative(topicS[K-1]) > 0 then
        kSol.append(topicS[K-1]);
        if kSol.size() > 3 then
            return kSol.take(3);
        end
    end
end
```

5. Knowledge validation and visualisation

Evaluating data models using unlabelled data is a complex and time-consuming task. ESCAPE includes both quantitative indices and visualisation techniques.

Quantitative metrics include for the joint-approach the silhouette-based indices, while for the probabilistic model (i) the perplexity and (ii) the entropy.

The silhouette-based indices could be summarised as follow:

- the weighted-Silhouette (WS) [1] is an index that can take values between 0 and 1 and represents the percentage of documents in each positive bin, suitably weighted with an integer value w between 1 and 10 (the highest weight is associated with the first bin [1-0.9] and so on) and normalised within the sum of all the weights. The higher the Silhouette index, the better the identified partition is.
- The average silhouette index (ASI) [48] is expressed as

$$ASI = \frac{1}{N} \sum_{k=1}^{K} \sum_{i \in C_k} s_i,$$

- The global silhouette index (GSI) [48] is expressed as

$$GSI = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i \in C_k} s_i.$$

On the other hand, for the probabilistic model ESCAPE integrates (i) the perplexity and (ii) the entropy.

- The perplexity is a measure of the quality of probabilistic models, that describes how well a model predicts a sample (i.e. how much it is perplexed by a sample from the observed data). Perplexity is monotonic decreasing in the likelihood of the data and is equivalent to the inverse of the per-word likelihood. It is defined as:

$$Perplexity(D) = \exp\left\{ - \frac{\sum_{d=1}^{D} \log p(w_d)}{\sum_{d=1}^{D} N_d} \right\}$$

  Here D is the number of documents (the corpus under analysis), w_d represents the words in document d, N_d the number of words in document d. Given a calculated model, the lower the general perplexity, the better the model performance and the probability estimate of the corpus [52].
- The entropy, when applied to the modelling context, measures how uncertain the model is: the lower the entropy of the model, the more certain it is that the model is describing the corpus under analysis. Specifically, for each d document in the corpus D we have calculated that entropy must belong to one of K' s topics and it is calculated as follows:

$$H(d) = -\sum_{k=1}^{K} p(d=k)\log(p(d=k))$$

  where $p(d=k)$ is the probability that the considered document will be assigned to the topic k. To compute the entropy of the whole clustering model, we averaged the entropy of each document on the whole corpus: $H(model) = \frac{\sum_{d=1}^{D} H(d)}{D}$.

To compare the different solutions found by ESCAPE, the Adjusted Rand Index (ARI) metric has been integrated in ESCAPE. The ARI is the corrected-for-chance version of the Rand index [53], [54] and [55]. The Rand Index can assume values between 0 and 1. When there is a perfect agreement between two partitions, the Rand index reaches the value 1 (its maximum). A limitation of the Rand index is that its expected value in the comparison of two randomly formed classifications is not always the same, as it should be. This problem is solved using the Adjusted Rand index [54], that assumes the generalised hyper-geometric distribution as the model of randomness. The Adjusted Rand index is ensured to have a value close to 0 in the case of random labeling and, differently from the Rand Index, it can assume negative values if the index is less than the expected index. Even if the partitions don't have the same number of clusters it is recommendable to use the Adjusted Rand.

To this aim, ESCAPE reported the ARI between solutions using the same strategy (i.e., Joint-Approach or Probabilistic- Approach) in order to compare the different weighting scheme impact. Such choice also enables us to analyse which are the main differences between the two approaches.

Besides displaying only statistical values or technical diagrams, which are often difficult to interpret, ESCAPE proposes several plots to explore and visualise the knowledge extracted from textual corpora. Specifically, ESCAPE enriches the cluster set, discovered through both approaches, to provide information that is more human-readable and therefore more understandable: (i) *document-topic distribution* and (ii) *topic-term distribution*.

Document-topic distribution characterises the distribution of the various topics identified within the document. It exploits the (i) *topic cohesion/separation* and the (ii) *coarse-grained vs fine-grained* groups, analysing how different weighting schemes can impact on the result. In particular, (i) is based on the *t-Distributed Stochastic Neighbour Embedding (t-SNE) [56], for the characterisation of the document distribution*. t-SNE allows representing high-dimensional data into lower dimensional maps through a non-linear transformation, suitable for human observation. Points assigned to different topics (i.e. clusters) are coloured differently. (ii) carries out the analysis of the weight impact in terms of coarse vs fine grained groups. To this aim, ESCAPE analyses the correlation matrices to analyse the possible correlation between different topics. At first documents are selected by topic, and then the dot products between all document pairs are computed. Thus, within the same macro category documents will be more similar to one another compared to those belonging to different categories.

Topic-term distribution characterises the distribution of the words within each latent topic. Specifically, ESCAPE includes the characterisation of (i) *topic-term distribution*, identifying the most relevant k words in terms of probabilities and frequency, and (ii) the *topic cohesion/separation* in terms of relevant words. Task (i) extracts the most probable top-k terms for each topic and represents them graphically using word-clouds [57], which is a popular visualisation of words typically associated with textual data. Lastly, for task (ii), we propose to use the graph representation to analyse the topic-term distribution. We have introduced two types of nodes: topic nodes and term nodes. The former, in green, represent the distinct topics, while the latter, in pink, represent the distinct terms within the collection under analysis. A topic is then connected through an edge with all the terms that are linked to it. To avoid links with low probability, ESCAPE extracts only the top-k most relevant (i.e., with the high probabilities) words for each topic. This parameter could be set by the analyst, however the default value is 20. If a word is connected with more than one topic, then the corresponding node is coloured in red. By doing so, we are able to compute the connectivity of the graph to analyse the results of the topic modelling. If there is any topic that is only connected with words that are not connected with any other topic, then this topic is separated from the rest of the graph. This means that the number of clusters selected by ESCAPE is able to separate the different topics. As a matter of fact, if all the words are connected to each other, all the terms have the same probability of belonging to each cluster.

6. Experimental Results

The experimental results performed to assess effectiveness and performance of ESCAPE are discussed in this section. We tested ESCAPE through different real datasets (dataset descriptions are reported in Subsection 6.1). The experimental setting is described in Section 6.2.

Experiments have been designed to address three main issues: (i) the ability of ESCAPE into performing all the textual analytics pipeline supporting the analyst into the setting parameters, (ii) the effectiveness of ESCAPE in discovering good document partitions, and (iii) the comparison with a state-of-the-art techniques.

6.1. Experiment datasets

The proposed framework has been tested over several datasets, belonging to different domains ranging from social networks and digital libraries (e.g. Twitter, Wikipedia) to scientific papers (e.g. PubMed collection). Corpora have been chosen to have different characteristics, from the number of documents to the length of each individual document, from lexical richness to the average frequency of terms. Moreover in the same corpus, the documents should be characterised by homogeneous lengths and heterogeneous subjects, as well as being produced by different authors. In this way these features allow results to be comparable and generic, avoiding overfitting of data sets. We have grouped the datasets based on their source and typology. In particular, datasets from D1 to D3 are collected from English documents from the Wikipedia collection[1] which is the largest knowledge-base ever known. The categories of each dataset have been chosen to be sufficiently separate and therefore detectable by the clustering algorithms. For each category, *top-k* articles are extracted, which will form our corpus. From these categories, different datasets have been generated, divergent by the number of documents extracted for each topic. To construct the first data set (i.e., D1), 200 articles were taken from the following five categories: *cooking*, *literature*, *mathematics*, *music* and *sport*. Instead, the following ten categories were chosen to build datasets D2 and D3: *astronomy*, *cooking*, *geography*, *history*, *literature*, *mathematics*, *music*, *politics*, *religion* and *sports*. D2 and D3 consist of 2500 and 5000 documents respectively,

[1] https://en.wikipedia.org/wiki/Wikipedia

chosen from these ten categories. Table 2 shows the *statistical features* of the three Wikipedia data sets used to test ESCAPE.

On the other hand, dataset D4 includes short messages extracted from Twitter. Twitter can be crawled to extract subsets of tweets related to a specific topic. We corroborated ESCAPE with experiments on a crisis tweet collection [58] that has 60,005 tweets with 16,345 distinguished words. Tweets were gathered from 6 large events in 2012 and 2013[2]. Hence, the dataset contains 10,000 tweets for each natural disaster and each tweet is labelled with relatedness (i.e., *on-topic* or *off-topic*). In our analysis, we remove the a-priori knowledge of each label, in order to understand if ESCAPE is able to eliminate the noise present in the collection. Dataset D5 involves 1000 papers extracted from the PubMed collection, which is an interface to MEDLINE[3], the largest biomedical literature database in the world. The number of expected categories is not a-priori known. Lastly, dataset D6 comprehends documents extracted from the Reuters collection[4] which is a widely used test collection for research purposes. The subset used for this study is the whole *Apte' Split 90 categories*, created merging together the test and the training part, for a total of 15.437 documents. The statistical features are reported in Table 3.

Features	**Wikipedia**					
Dataset ID	D1		D2		D3	
# categories	5		10		10	
# documents	990		2,469		4,939	
Max frequency	5,394		13,344		19,546	
Features	**WH**	**WoH**	**WH**	**WoH**	**WH**	**WoH**
Min frequency	1.0	2.0	1.0	2.0	1.0	2.0
Avg frequency	25	45	36	69	39	78
Avg document length	852	836	970	957	705	697
# terms	843,967	828,372	2,395,721	2,363,958	3,486,016	3,442,508
Dictionary $\mid V \mid$	33,635	18,040	65,629	33,866	87,419	43,911
TTR	0.04	0.03	0.02	0.01	0.03	0.01
Hapax %	46.3	0.0	48.2	0.0	49.1	0.0
Guiraud Index	36.61	19.82	42.40	22.02	46.82	23.66

Table 2. Statistical features for the Wikipedia collections

Through the analysis of the proposed statistical features, we are able to categorise the datasets into few groups according to their statistical indices. In fact, we can observe that the datasets have different characteristics. The Wikipedia documents together with the category PubMed articles are characterised by a greater length and a higher lexical richness than the others, in fact the Guiraud Index is higher for these datasets, reaching the maximum value with the PubMed articles. The dictionary, even after Hapax removal, is extremely high and reflects the complexity of the datasets chosen to test ESCAPE. Moreover, the PubMed collection presents a further complexity, i.e., the expected number of topics is not known a-priori.

On the other side, we have also included a dataset represented by smaller lexical richness, i.e., the Twitter collection. The average document length decreases considerably, as does the average frequency. Nevertheless, the Hapax rate is comparable with the other datasets, and the dictionary after the Hapax removal is smaller with respect to the other datasets. Among the datasets we have also included the Reuters collection, as it presents differences in data distributions with respect to the other datasets. The Reuters are characterised by a medium length and a lexical index not too high, since the average

[2] 2012 Sandy Hurricane, 2013 Boston Bombings, 2013 Oklahoma Tornado, 2013 West Texas Explosion, 2013 Alberta Floods and 2013 Queensland Floods
[3] https://www.ncbi.nlm.nih.gov/pubmed/
[4] http://www.daviddlewis.com/resources/testcollections/reuters21578

Features	**Twitter**		**PubMed**		**Reuters**	
Dataset ID	D4		D5		D6	
# categories	6		-		90	
# documents	60,005		1,000		15,437	
Max frequency	6,936		775		42,886	
Features	**WH**	**WoH**	**WH**	**WoH**	**WH**	**WoH**
Min frequency	1.0	2.0	1.0	2.0	1.0	2.0
Avg frequency	19	36	15	18	55	76
Avg document length	5	5	3600	3469	87	85
# terms	312,718	304,666	3,600,153	3,469,305	1,337,225	1,316,988
Dictionary |V|	16,345	12,136	227,210	96,362	24,239	17,153
TTR	0.05	0.03	0.06	0.05	0,02	0.01
Hapax %	49.26	0.0	57.02	0	29.2	0.0
Guiraud Index	29.23	15.02	119.75	51.73	20.96	14.95

Table 3. Statistical features for datasets D4, D5 and D6

frequency of the terms is the highest (i.e., the documents are characterised by a medium length with terms repeated several times). For this reason, the lexical richness is the lowest of all corpora.

6.2. Experimental setting

The ESCAPE framework has been developed to be distributed and has been implemented in Python. All the experiments have been performed on the BigData@PoliTO cluster[5] running Apache Spark 2.3.0. The virtual nodes deployed for this research, the driver and the executors, have a 7GB main memory and a quad-core processor each. Below we reported the default configuration for the Joint-Approach and the default configuration for the Probabilistic-Approach.

Joint-Approach configuration setting. For the joint-approach ESCAPE requires two parameters, i.e., the number of dimensions to be considered during the data reduction phase (SVD) and the number of clusters (topics) in which to divide the collection under analysis. During the singular value decomposition reduction phase, the reduction parameter analyses the trend of singular values in terms of their significance. Important dimensions are characterised by a large magnitude of the corresponding singular values, while those associated with a low singular value should be ignored in the subsequent phases. For this reason, we have decided to consider only the first T singular values for the analysis. T at most will be equal to the rank of the document-term matrix. This parameter should be set by the analysis, however, since the number of documents for all the textual corpora analysed is much smaller than the vocabulary used in each collection, the value T is set by ESCAPE to the 20% of the number of documents. Nevertheless, the analyst can decide to change the proposed configuration, setting other values for T. The second parameter that should be set is the number of topics. We have proposed a new self-tuning algorithm to automatically configure the best configuration. In ESCAPE, the default configuration for the maximum number of clusters is set to the average document length for each corpus. In fact, we have hypothesised that every word in the document belongs to at most a different topic. In this way, we set an upper-bound for the value of the number of clusters. Still, if the average document length is greater than the number of documents in the corpus under analysis, then the value is set to the average frequency of the term. Even so, these choices can be changed by every analyst, since the framework architecture is distributed it is also able to analyse several solutions in parallel.

Probabilistic model configuration setting. We recall that for the LDA probabilistic algorithm, five parameters should be set, which are the maximum number of iterations, the

[5] https://bigdata.polito.it/content/bigdata-cluster

Optimiser, the document concentration (α), the topic concentration (β) and the number of topics (clusters) in which each corpora should be divided. Except for the last parameter, for which we have integrated a self-configuring algorithm, the other four parameters have to be set by the analyst. In ESCAPE the maximum number of iterations within the model has to converge has been set to be equal to 100, the Optimiser (or inference algorithm used to estimate the LDA model) has been set to be Online Variational Bayes. Furthermore, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are set to maximise the log likelihood of the data under analysis. Since we have selected the `Online` optimiser, the $\boldsymbol{\alpha}$ value and the $\boldsymbol{\beta}$ value should be greater than or equal to 0. For this study, the default value for this parameter is $\boldsymbol{\alpha} = 50/K$, as proposed in the literature by different articles [59], [60], [61], and the value set for $\boldsymbol{\beta}$ is $\boldsymbol{\beta} = 0.1$, as proposed in [59].

ESCAPE offers an automatic methodology able to select the proper number of clusters, without involving the user in this decision. ESCAPE proposes a novel strategy to assess how semantically different the topics are and choose proper values for the configurations of the probabilistic modelling. As for the joint-approach, in ESCAPE, the default parameter for the maximum number of topics is set to the average document length for each textual collection. Indeed, each word in the document belongs to at most a different topic in our hypothesis. Thus, the upper-bound for the number of topic parameters is set to the average length. However, if the average document length is greater than the number of documents in the corpus under analysis, then the value is set to the average frequency of the term. As always these choices can be changed by the analyst.

6.3. ESCAPE Performances

Here we reported a summary of the experiments conducted on the six datasets using the Joint-Approach and the Probabilistic-Approach. ESCAPE has been run several times, once for each weighting strategy and dataset. Dataset D1 has been chosen as the running example for a detailed comparison.

Joint-Approach. Table 4 reports the experimental results obtained for D1 and includes the metrics computed for evaluating document partitions identified by our framework. For each weighting strategy, the top-3 solutions (i.e., configurations) are reported to the analyst. The best solution is reported in bold. We observe that ESCAPE tends to select a partition with a low-medium number of dimensions as the optimal partition. The variability of the data distribution and the complexity of the cluster activity are directly proportional to the $K - LSA$ value. So, Silhouette indices usually decrease when considering a large number of terms with each document (columns of the dataset).

For the weighting scheme TF-IDF, the three reduction factors for the SVD decomposition (K_{LSA}) are 26, 41 and 67. For each dimensionality reduction parameter, ESCAPE selects the best value for the clustering phase. Given these numbers of dimensions, ESCAPE selects $K_{Clustering=10}$ as the optimal partition. Since the silhouette-based metrics are quite stable, ESCAPE selects only the most relevant terms in the building of the model, ignoring the less relevant terms (dimensions).

The TF local weight tends to differentiate the weighted terms, thus obtaining a larger number of clusters than that discovered by LogTF (because now several clusters are associated with different topics of the same category). This is also confirmed by the weight definition. Indeed, the logarithmic function tends to decrease the very high frequency values. In fact, the more the frequency of the term increases, the more the function approaches the asymptote of the logarithm. This means that from a certain frequency, the value of local weight tends to flatten and the relevance of the most frequent terms is reduced. With respect to the global weight instead, we can observe that the Entropy tends to find in average a large number of clusters.

The TF-IDF and the TF-Entropy find a large number of topics with respect to the other solutions. The other weights instead are able to select the expected value of the category. Moreover, the weights TF-IDF and TF-Entropy not only find the original major category but are able to find also the sub-topic related to the major categories. In this way, if the analyst is interested in analysing the dataset at a minor level of detail, he could use these

Weight	K_{LSA}	$K_{Clustering}$	GSI	ASI	Weighted Silhouette	Execution Time
TF-IDF	26	7	0.383	0.358	0.408	22m, 20s
	41	**10**	**0.419**	**0.339**	**0.391**	
	67	10	0.361	0.297	0.352	
TF-Entropy	29	11	0.334	0.350	0.401	26m, 18s
	42	**10**	**0.368**	**0.331**	**0.382**	
	62	8	0.364	0.274	0.326	
LogTF-IDF	**19**	**5**	**0.437**	**0.431**	**0.480**	25m, 23s
	22	5	0.350	0.343	0.393	
	67	4	0.225	0.201	0.251	
LogTF-Entropy	**10**	**6**	**0.440**	**0.453**	**0.500**	27m, 12s
	24	5	0.323	0.318	0.367	
	67	7	0.268	0.218	0.267	
Bool-IDF	**8**	**5**	**0.445**	**0.444**	**0.494**	25m, 33s
	22	6	0.293	0.312	0.365	
	65	6	0.226	0.233	0.286	
Bool-Entropy	**9**	**5**	**0.447**	**0.444**	**0.495**	28m, 38s
	23	5	0.354	0.348	0.400	
	65	4	0.280	0.234	0.285	

Table 4. Experimental results for D1 through the joint-approach.

weights, and leave the others for a grain analysis. ESCAPE is able to analyse the same dataset at different granularity levels.

Probabilistic-Approach. Table 5 shows the results obtained using the Probabilistic-Approach for dataset D1. As for the joint-approach, each dataset is evaluated for every single weighting scheme considered in ESCAPE, showing the top-3 configurations. For each dataset under analysis, we will sum up the considerations about the effectiveness of ESCAPE in discovering good partitions, as the different weighting schemes vary.

The main results obtained by ESCAPE for each textual corpus and weighting strategies, are reported from Table 5 to Table 10. Specifically, Tables from 5 to 7 are related to the Wikipedia datasets, Table 8 with the Tweeter crisis collection. The PubMed results are explored in Tables 9. Lastly, the Reuters collection is shown in Table 10.

Weight	K	Perplexity	Silhouette	Entropy	Execution Time
TF-IDF	3	8.812	0.772	0.256	40m, 24s
	6	8.597	0.693	0.363	
	10	**8.482**	**0.682**	**0.395**	
TF-Entropy	**5**	**9.072**	**0.762**	**0.282**	30m, 32s
	8	9.248	0.632	0.338	
	9	9.267	0.631	0.339	
LogTF-IDF	**8**	**9.187**	**0.675**	**0.320**	
	17	9.126	0.637	0.362	40m, 17s
LogTF-Entropy	5	9.912	0.891	0.100	30m, 54
	7	**9.884**	**0.846**	**0.174**	
	11	9.979	0.951	0.108	
Boolean-TF	4	6.492	0.697	0.421	44m, 43s
	5	**6.464**	**0.661**	**0.483**	
	17	6.420	0.381	1.090	

Table 5. Experimental results for D1 through the probabilistic approach.

Joint-Approach						
Dataset	Weight	K-LSA	K-clus	GSI	ASI	Weig-Sil
D2	TF-IDF	57	13	0.280	0.236	0.288
	TF-Entropy	63	13	0.271	0.209	0.265
	LogTF-IDF	25	9	0.236	0.224	0.028
	LogTF-Entropy	26	7	0.270	0.233	0.281
	Bool-IDF	25	9	0.221	0.213	0.263
	Bool-Entropy	26	9	0.238	0.227	0.278

LDA					
Dataset	Weight	K	Perp	Silh	Entropy
D2	TF-IDF	10	8.943	0.553	0.611
	TF-Entropy	7	9.455	0.700	0.355
	LogTF-IDF	11	9.410	0.601	0.489
	LogTF-Entropy	7	10.203	0.875	0.125
	Bool-TF	18	6.569	0.320	1.326

Table 6. Experimental results for D2

Since the considered weighting schemes highlight the importance of terms within the documents, it could be interesting for the analyst to understand how different weights affect the probabilistic model generated by the LDA. Specifically, for each result table, ESCAPE includes a row for each K obtained through the ToPIC-Similarity curve together with the three well-known state-of-the-art quality indices used to explore the goodness of the statistical model generated.

Different trends can be pointed out and detected from the analysis of these tables. Firstly, we can highlight a reverse linear trend between entropy and silhouette metrics, since better clustering partitions are characterised by a high silhouette value and a small entropy one. Moreover, through the ToPIC-Similarity testing, the TF local weight usually finds in average a smaller number of clusters, independently of the global weight used. On the other hand, the LogTF local weight finds a large number of topics which allows the same dataset to be analysed in detail, since this weight can also find some interesting subtopics within the macro-topic. From the exploitation of the global weights, several comments can be made. In fact, the Global IDF results show a better value for the perplexity index (e.g. at least 0.1 greater) than those obtained using global Entropy, even though the other quality metrics are not in line.

Analysing all the corpora using the Boolean-TF instead, lead to a comparison of very different solutions. This weighting scheme is able to find, using our ToPIC-Similarity curve, three numbers of topics with different values. Moreover, the first two datasets lead to very high values of silhouette scores, while these values tend to decrease in the other datasets. In fact, the complexity of the PubMed collections or the Reuters one, imply smaller values of our quality metrics. However, with this methodology, the analyst is able to analyse the same dataset at different granularity levels. For the four datasets for which we know the number of categories (i.e., D1, D2,D3 and D4) the global weight Entropy underestimate the number of topics, finding at least as upper bound the expected number of categories, while the IDF weight tends to overestimate the number of topics. Moreover, the Wikipedia datasets represent the experiments in which the performance found are the highest ones. This behaviour is also confirmed for the other datasets for which we do not know the number of categories.

Nevertheless, analysing the goodness of the partitions found only through quantitative metrics is not sufficient, as we limit the analysis to measure the distances (Euclidean and probabilistic) between the groups of documents.

In order to effectively validate the probabilistic model, a deep and detailed knowledge of human common-sense should be provided to interpret the main argument of each cluster. Furthermore, since ToPIC-Similarity proposes a maximum of three good values for the topic analysis, the analyst can choose, among the various solutions proposed, the one that best reflects the required granularity of the arguments (i.e., topics). With respect to LSA (the joint-approach), the analysis of only quality metrics is not sufficient to analyse the partitions. A more detailed analysis should be included to help the analyst in interpreting the results. Also, the analysis of how each weighting strategy acts on the LDA model should be analysed to highlight interesting considerations.

6.4. Knowledge exploration and visualisation

The complete set of results obtained for the representative dataset D1 will be presented. Here we reported two types of human readable results able to provide to the analysts

Joint-Approach						
Dataset	Weight	K-LSA	K-clus	GSI	ASI	Weig-Sil
D3	TF-IDF	51	9	0.233	0.221	0.274
	TF-Entropy	51	11	0.246	0.221	0.272
	LogTF-IDF	26	9	0.220	0.205	0.255
	LogTF-Entropy	26	10	0.246	0.221	0.272
	Bool-IDF	22	7	0.225	0.191	0.241
	Bool-Entropy	23	6	0.257	0.196	0.247

LDA					
Dataset	Weight	K	Perp	Silh	Entropy
D3	TF-IDF	10	8.708	0.339	2.456
	TF-Entropy	7	9.050	0.214	1.852
	LogTF-IDF	16	8.917	0.198	1.819
	LogTF-Entropy	5	9.444	0.096	2.293
	Bool-TF	11	6.309	0.220	1.902

Table 7. Experimental results for D3

Joint-Approach						
Dataset	Weight	K-LSA	K-clus	GSI	ASI	Weig-Sil
D4	Bool-IDF	6	6	0.465	0.422	0.737
	Bool-Entropy	13	7	0.342	0.320	0.532

LDA					
Dataset	Weight	K	Perp	Silh	Entropy
D4	Bool-TF	6	2.808	0.546	0.613

Table 8. Experimental results for D4

interesting information at different granularity levels. Specifically, we reported extracted knowledge analysing the statistical quality metrics used to analyse the different partitions obtained running ESCAPE for each approach. However, analysing a corpus considering only quantitative measures is not sufficient. For this purpose, we have proposed several graphs useful for exploring the space of the results with innovative and useful visualisation techniques. By this way, the analysts could analyse the different representations integrated in ESCAPE.

Knowledge Validation Here we have displayed the main visualisations techniques integrated in ESCAPE. At first we want to focus the reader's attention on a deeper comparison between the two methodologies. In Tables 4 and 5 we have reported the results obtained for the dataset D1. Specifically, Table 4 reports the results obtained for the join-approach, while Table 5 reports the results obtained for the probabilistic approach, as discussed in detail in the previous subsection.

Instead, in Table 11 are reported the cardinalities of the different cluster-sets found by ESCAPE for dataset D1. We have compared the weighting schemes TF-IDF and LogTF-Entropy for the two different methodologies.

Knowledge exploration. Since the results obtained in the previous sections are described only using quantitative metrics, other graphical representations should be presented to exploit the hidden knowledge.

To graphically represent the effect of both weighting functions for the **joint-approach**, ESCAPE analyses the correlation matrix maps reported in Figure 3 for D_1. Five different colours were defined, based on the correlation range: black colour represents the highest range 0.87-1.00, dark gray the range 0.75-0.87, gray is used for the range 0.62-0.75, light gray is associated with the range 0.5-0.62, and white represents the lowest range 0.0-0.5. Documents are sorted according to their category and then the dot products between all document pairs are calculated. Figure 3 (Left) shows how the different weighting functions TF-IDF and LogTF-Entropy impact on the document collection. In both functions, the 5 macro-categories are depicted as five dark squares of similar size showing the highest similarity between documents. So, considering two documents belonging to the same macro category, they will tend to be more similar to each other than those belonging to other macro categories; LogTF-Entropy (Figure 3) (Left on the bottom) allows modelling

Joint-Approach						
Dataset	Weight	K-LSA	K-clus	GSI	ASI	Weig-Sil
D5	TF-IDF	56	10	0.098	0.087	0.136
	TF-Entropy	59	9	0.106	0.092	0.142
	LogTF-IDF	33	5	0.100	0.092	0.144
	LogTF-Entropy	35	5	0.098	0.090	0.140
	Bool-IDF	24	8	0.127	0.112	0.163
	Bool-Entropy	26	15	0.120	0.117	0.167

LDA					
Dataset	Weight	K	Perp	Silh	Entropy
D5	TF-IDF	14	7.662	0.085	1.902
	TF-Entropy	4	8.556	0.081	1.782
	LogTF-IDF	14	7.776	0.094	1.754
	LogTF-Entropy	4	8.622	0.080	1.743
	Bool-TF	10	5.220	0.101	1.318

Table 9. Experimental results for D5

Dataset	Weight	Joint-Approach K-LSA	K-clus	GSI	ASI	Weig-Sil
D6	TF-IDF	15	10	0.246	0.257	0.159
	TF-Entropy	16	14	0.254	0.256	0.157
	LogTF-IDF	16	13	0.232	0.236	0.146
	LogTF-Entropy	16	10	0.229	0.238	0.150
	Bool-IDF	13	9	0.229	0.235	0.147
	Bool-Entropy	13	10	0.220	0.223	0.143

Dataset	Weight	LDA K	Perp	Silh	Entropy
D6	TF-IDF	9	7.438	0.596	0.558
	TF-Entropy	9	8.710	-0.081	2.169
	LogTF-IDF	13	7.561	0.598	0.639
	LogTF-Entropy	5	8.788	0.077	1.609
	Bool-TF	16	3.730	0.301	1.311

Table 10. Experimental results for D6

	Weight	Cluster ID Cluster0	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6	Cluster7	Cluster8	Cluster9	Total
LSA	TF-IDF	215	176	159	139	99	93	49	25	19	15	989
	TF-Entropy	228	167	166	135	106	75	54	27	16	15	
	LogTF-IDF	225	212	191	183	178						
	LogTF-Entropy	223	191	184	183	105	103					
	Bool-IDF	236	223	191	181	158						
	Bool-Entropy	230	223	192	177	167						
LDA	TF-IDF	205	193	187	180	144	21	19	14	13	13	989
	TF-Entropy	464	406	91	8	7	5	5	3			
	LogTF-IDF	428	236	197	113	15						
	LogTF-Entropy	827	160	1	1	0						
	Bool-TF	230	215	194	188	162						

Table 11. Cardinality of each cluster set found for dataset D1 for the probabilistic approach

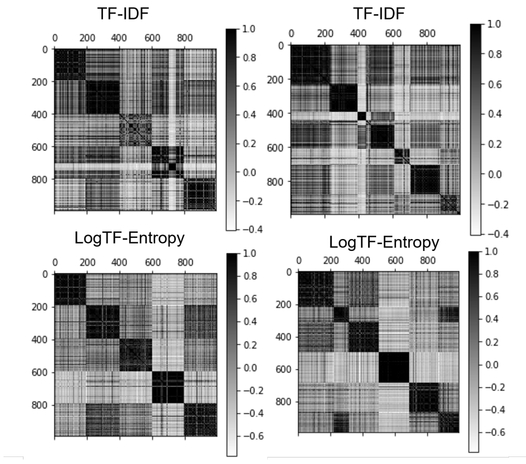


Figure 3. Correlation matrix maps for dataset D1 for analysing: the weighting impact (Left) and the best partitions (Right)

the 5 macro categories better than TF-IDF (Figure 3) (Left on the top) and also characterises some topics; whereas TF-IDF shows possible correlations between the different categories.

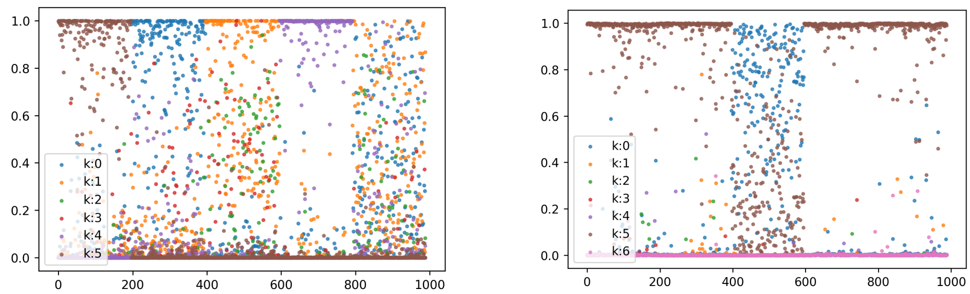


Figure 4. Document probability distributions in each topic for weighting TF-IDF (Top) and LogTF-Entropy

Figure 3 (on the Right) shows the correlation matrix maps for the best partitions identified by ESCAPE; LogTF-Entropy (Figure 3 Right on the bottom) correctly finds the dataset categories whereas TF-IDF (Figure 3 Right on the top) also highlights some relevant subtopics in the same category.

The importance of words within documents is determined by the weights; therefore, it is important to assess how the model is affected by different weighting schemes. For the representative dataset D1, ESCAPE computes the histogram of the TF-IDF and LogTF-Entropy weights. The LogTF-Entropy values are almost uniformly distributed in the range [0,1] (Kurtosis index > 0 and standard deviation 0.5). A different scenario is instead obtained with the IDF, where there is an asymmetrical bell distribution in which the average values are in the range [2,5] (Kurtosis index > 0 and standard deviation 12.7). Moreover, in this case the maximum value of the distribution is 8, while in the LogTF-Entropy case it is 1161. For the probabilistic approach, the IDF weight scheme better differentiates the weights within the corpus, and for this reason is able to produce a more performant probabilistic model. Figure 4 shows that providing relevance to words in all datasets, the Entropy global weight performs wrongly. This figure shows, for the LDA models, the probability distribution that each document in the D1 corpus has of belonging to the K selected topics. K is equal to 6 for TF-IDF (on the left) and is equal to 7 for LogTF-Entropy (on the right). For TF-IDF we used the second best solution due to the limited number of clusters. Analysing the results found in more detail, we can see that with the IDF weighted documents are more uniformly distributed among the various topics. On the other hand, as far as the Entropy weight is concerned, about 90% of the documents are assigned to the same cluster (topic) and this is the consequence of the fact that the entropy weight fails to isolate the most significant terms within the collection of documents.

We can conclude that some weighting strategies are useful for a particular analysis with respect to the others. As a matter of fact, from the analysis of the histograms, and also from the results analysed previously, we can assess that the IDF weight scheme performs better the function of differentiating weights within the corpus.

When we are in the situation where unbalanced clusters are present, the usual evaluation metrics are not sufficient to guarantee good performance. A high Silhouette index does not guarantee a good quality of the obtained clusters, because it is as if 90% of the documents were all classified with the same label, generating many false negatives. To overcome this situation, if the class label is available, we can use indices such as precision and recall, trying to identify incorrect assignments. Otherwise, if we don't have labels, methods that consider semantics must be presented.

On the other hand, the joint approach leads to better results from the point of view of the partitions. In fact, the weights in this case analyse the same dataset at different levels of detail, without creating unbalanced clusters. In fact, the K-Means algorithm is benefiting from the previous LSA reduction, in this way its performances are far superior.

6.5. Dealing with large dataset

In this section we show the results of the proposed approach when used with large datasets. As a case study, we tested ESCAPE with some datasets containing revisions of Amazon users. Data are retrieved from the Amazon Customer Reviews Database and reviews have been collected between 1995 and 2015. Reviews that refer to different categories, belong to different datasets. In particular, we have now focused on the following data, described in Table 12:

- D7: Digital Music (349933 documents);
- D8: Luggage (325588 documents);
- D9: Video Games (409551 documents).

The following subsections include results obtained for the joint approach and the probabilistic approach. Since the datasets are characterized by a very sparse data distribution, we didn't consider global weight Entropy in these experiments. For the probabilistic

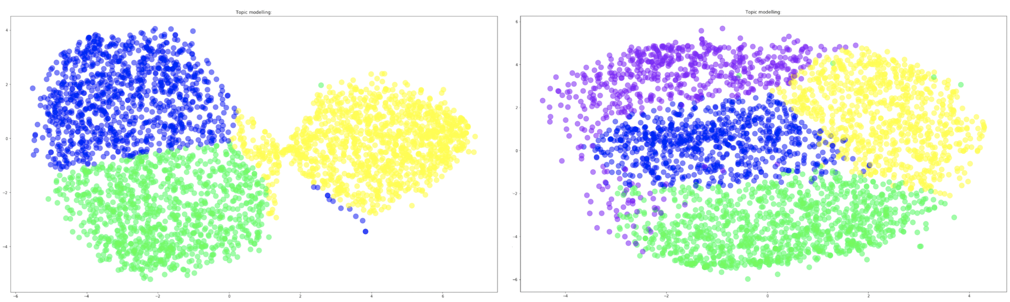
Features	Digital Music	Luggage	Video Games
Dataset ID	D7	D8	D9
# documents	349933	325588	409551
Max frequency	129584	112280	287780
Min frequency	2	2	2
Avg frequency	119	330	278
Avg document length	9.68	18.26	16.67
# terms	3386835	5946360	6828539
Dictionary V	28300	17999	24510
TTR	0.008	0.003	0.006
Hapax %	0	0	0
Guiraud Index	15.37	7.38	9.38

Table 12. Statistical characterization of datasets under analysis

approach, we only consider D8 and D9, where documents have the highest average length. For visualization results, we focus only on dataset D8, both for joint ad probabilistic approach.

6.5.1. Joint approach

The three different weighting schemas (Boolean-IDF, TF-IDF, LogTF-IDF) are tested with ESCAPE and the obtained results are shown in Table 13. In general, the Average and Global silhouette values corresponding to the selected best configurations are, for all the data-sets, in the range between 0.2 and 0.5, suggesting that the partitions are good. From the results we find that TF-IDF finds, in general, a larger number of topics (number of clusters) meaning that it is able to detect not only the original categories but also subtopics.



(a) Dataset D9. t-SNE representation. B-IDF weighting schema K=3

(b) Dataset D9. t-SNE representation. LogTF-IDF weighting schema K=4

Figure 5. Boolean-IDF and LogTF-IDF weighting schemas results for the Luggage dataset.

Figure 5 shows how the reviews of the Luggage dataset are distributed between clusters. It is possible to notice a difference between the two weighting schemas used in these graphs, in fact the shape of the Boolean-IDF clusters seems to be more defined with respect to LogTF-IDF.

6.5.2. Probabilistic approach

As mentioned earlier, in this section we conducted experiments only for datasets D8 and D9, which are those with highest average length. The performance of the statistical model has been explored thanks to the quality index of Perplexity computed within ESCAPE. These results are shown in table 14, where low perplexity values indicate better results.

Regardind dataset D8 on Luggage reviews, LogTF-IDF weighing strategy differs from the others since it provides a more detailed analysis discovering also subtopics, in addition

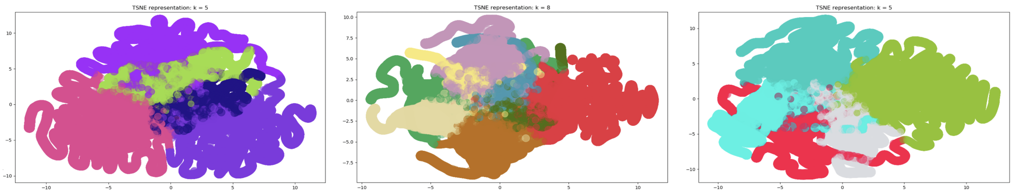
	Weight	**K_{LSA}**	**$K_{Clustering}$**	**GSI**	**ASI**	**Weighted Silhouette**
D7	BooL-IDF	4	4	0.371	0.364	0.009
		12	18	0.182	0.175	0.005
		31	15	0.221	0.248	0.007
	LogTF-IDF	5	3	0.310	0.325	0.008
		11	8	0.248	0.248	0.007
		28	19	0.191	0.192	0.006
	TF-IDF	6	2	0.474	0.532	0.013
		10	3	0.351	0.546	0.014
		22	2	0.394	0.389	0.010
D8	BooL-IDF	3	3	0.406	0.409	0.011
		7	6	0.170	0.172	0.005
		28	2	0.062	0.055	0.003
	LogTF-IDF	4	4	0.286	0.294	0.008
		9	8	0.170	0.170	0.005
		28	20	0.107	0.106	0.004
	TF-IDF	5	5	0.289	0.298	0.009
		13	18	0.206	0.189	0.006
		30	20	0.154	0.135	0.004
D9	BooL-IDF	3	3	0.390	0.396	0.009
		6	4	0.248	0.246	0.006
		25	15	0.163	0.163	0.004
	LogTF-IDF	3	3	0.399	0.406	0.009
		6	3	0.232	0.232	0.006
		25	17	0.174	0.184	0.004
	TF-IDF	4	2	0.358	0.355	0.008
		9	2	0.256	0.249	0.006
		26	13	0.189	0.172	0.004

Table 13. Experimental results through the joint-approach.

Dataset	**Weight**	**K_{CI}**	**Perplexity**
D8	BooL-IDF	5	7.273681
	LogTF-IDF	3	7.352020098
		5	7.263175195
		8	**7.190609656**
	TF-IDF	5	7.270052194
D9	BooL-IDF	2	7.588552184
	LogTF-IDF	2	7.581219438
	TF-IDF	2	7.583352794

Table 14. Experimental results for dataset D8 and D9 for the probabilistic approach.

to the five main topics already discovered also by the other schemas. Instead, this different level of results granularities is not present for the Video product category dataset (D9). The graphical visualization of the results obtained with D8 is then shown in Figure 6. In figures 6A and 6C we can see similar shapes and distribution of the documents between the clusters. In Fig 6B it is possible to recognize an imbalance of the colouring of the points: the main five topics containing a major number of documents and three smaller subtopics.



(a) Dataset D8. t-SNE representation. Bool-IDF weighting schema K=5 **(b)** Dataset D8. t-SNE representation. LogTF-IDF weighting schema K=8 **(c)** Dataset D8. t-SNE representation. TF-IDF weighting schema K=5

Figure 6. Best partitioning t-sne maps for all the weighting strategies for the Luggage dataset are displayed above

6.6. Comparison with respect to the state-of-the-art

Here follows a comparison between ESCAPE and the main state-of-the-art techniques.

Joint-Approach. In order to assess how effectively ESCAPE is able to select the proper number of clusters, we compared the results obtained with those proposed by a state-of-the-art methodology designed for the same purpose. This method is known as the *Elbow graph* or *Knee* approach [62]. In the following we will refer to this method as k_{SSE}. This method involves evaluating the evolution of the SSE (Sum of Squared Errors) value as the value k_{cls} increases. The k_{cls} value identified as optimal is the one immediately preceding a negligible change in the SSE value (there is no great performance advantage in adding another centroid). In the following we will refer to the dataset D_1 as representative, but similar trends have also occurred in other datasets.

In order to compare the methods fairly, both ESCAPE and the k_{SSE} method, receive as input the reduced matrix X_{K-LSI}. This matrix is obtained by analysing the trend of the singular values extracted by the decomposition of the original document-term matrix. In our proposed methodology, ESCAPE selects the possible good values at the points: 10, 24 and 67. These three points are able to characterise the singular value plot, analysing different values which subsequently include a large number of dimensions in the reduction phase.

However, the k_{SSE} method usually selects a lower number of optimal clusters than the one selected in ESCAPE.For example, in D_1 the k_{SSE} method selects 5 clusters by exploiting TF-IDF and 3 with LogTF-Entropy, against the 10 clusters selected by ESCAPE using TF-IDF and 6 clusters with LogTF-Entropy.

To evaluate the best configuration between those identified by the two approaches, we evaluated the Silhouette index for each clustered document, in both methods. As shown in Figure 7, more than 83% of the documents obtain a higher index in the approach proposed by us than in that based on the analysis of the SSE curve. Thus, this result tells us that ESCAPE is able to discover a cluster set better than the Knee approach.

Probabilistic Approach. Here, we offer a comparison between the results obtained by ESCAPE and those obtained with known state-of-the-art techniques such as *RPC* and *En-LDA*. RPC [50] is an heuristic algorithm that, in order to choose the proper number of topics, evaluates the average perplexity variation of the LDA models. Instead EnLDA [63] chooses as the optimal K value the one that best reduces the total amount of entropy of the topic modelling. These two approaches will be discussed in more detail below.

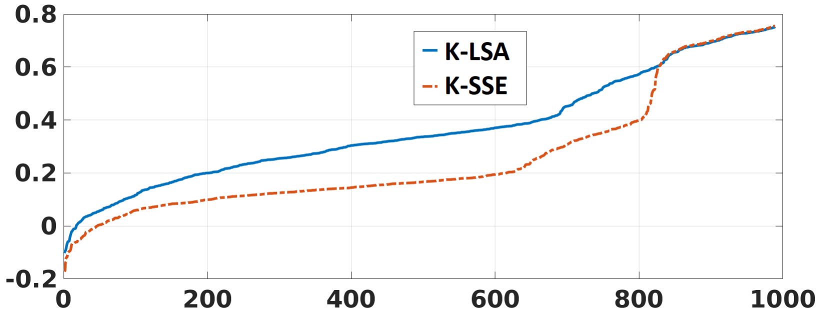


Figure 7. Silhouette index for $D1$ weighted via LogTF-Entropy for the joint approach

	Weights	Method	K	Perpl	Silh	Entr
D1	**TF-IDF**	RPC	3	8.812	0.772	0.256
		En-LDA	19	8.427	0.621	0.534
		ESCAPE	10	8.482	0.682	0.395
	TF-Entr	RPC	5	9.072	0.762	0.282
		En-LDA	5	9.072	0.762	0.282
		ESCAPE	5	9.072	0.762	0.282
	LogTF-IDF	RPC	7	9.183	0.693	0.319
		En-LDA	16	9.189	0.553	0.443
		ESCAPE	8	9.187	0.675	0.320
	LogTF-Entr	RPC	3	9.777	0.852	0.144
		En-LDA	3	9.777	0.852	0.144
		ESCAPE	7	9.884	0.846	0.174
	Boolean-TF	RPC	4	6.492	0.697	0.421
		En-LDA	20	6.412	0.661	1.255
		ESCAPE	5	6.464	0.661	0.483

Table 15. Comparison between ESCAPE 's performance and that of other state-of-the-art methods

Table 15 shows a comparison between the results obtained by ESCAPE and those obtained by the RPC and en-LDA methods, for the various weights considered. We can see that using TF-IDF, these two approaches produce as K values 3 and 19 (with RPC and En-LDA respectively). These values depict two different scenarios.
The RPC proposes 3 as the optimal number of clusters. This is the same value proposed by the first solution of the ESCAPE framework. As described above, the clustering result is not bad, but some of the original topics are mixed together (*music* and *literature*, *sports* and *mathematics*). In this sense, ESCAPE outperforms RPC giving more options with different granularity levels to the analyst.

With the En-LDA approach, which proposes 19 as the optimal number of clusters, good partitions are identified (the t-SNE representation of the clustering result is reported in Figure 10d). As a matter of fact, all the original categories of the dataset can be recovered in topics. Furthermore, the model identifies very specific topics, that describe only a few documents, and it often divides the main categories in subtopics, which deal with more specific arguments compared to main ones. For instance the En-LDA approach identifies the *opera* and the *instruments* topics, which both belong to the *music* main category. The modelling is overall good, but having more topics that the ones actually required not necessarily means having a better result. Indeed, too many topics may not be useful for the analysis since then the analysts have a more complex result set to consider in their work.

Figure 9 offers an intuitive graphical representation of the topics identified using TF-IDF as weighting scheme and K=10. The word clouds depicted represent the main categories present in the original dataset and effectively show which are the most significant

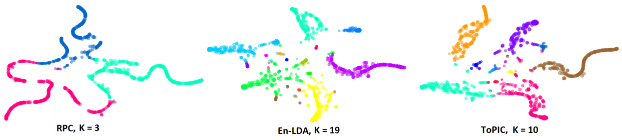


Figure 8. Comparison of t-SNE representations for dataset D1

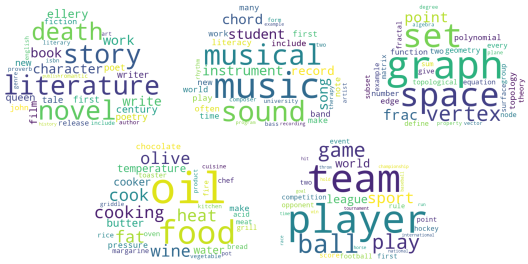
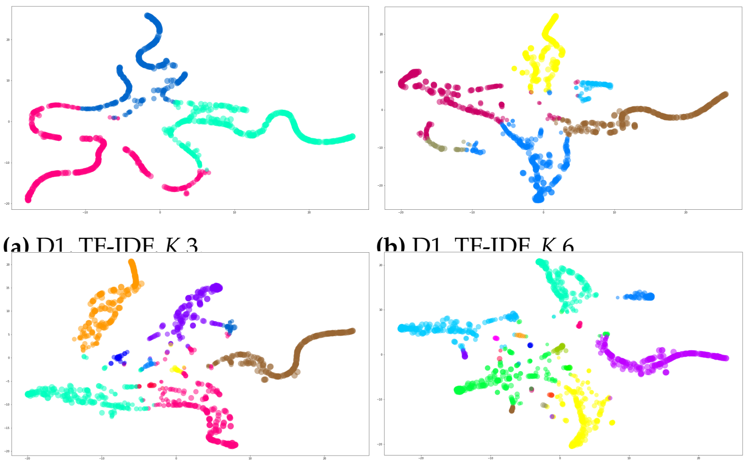


Figure 9. Word cloud representation of a subset of topics, dataset D1, TF-IDF weighting scheme, K = 10



(a) D1, TF-IDF, K 3. **(b)** D1, TF-IDF, K 6.

(c) D1, TF-IDF, K 10. **(d)** D1, TF-IDF, K 19.

Figure 10. D1 t-SNE representation, TF-IDF weighting scheme, K 3, 6, 10 and 19 respectively.

terms summarising the identified topics. The five missing clusters that do not appear in the representation are those that include terms referring to more detailed subtopics, and therefore have not been included in the figure.

Another appropriate comparison between ESCAPE and other state-of-the-art methods should be made from the point of view of computational cost and time. Compared to En-LDA, the proposed methodology is much faster; in fact, the number of iterations to be performed in En-LDA increases substantially with the growing vocabulary of documents. Furthermore, the search for the minimum entropy value among all possible solutions with a different K means that the methodology must be calculated for all the topics in the given set. RPC performance, on the other hand, from a computational cost perspective, can be compared to the one required by ESCAPE in the worst case. Moreover, with respect to the state-of-the-art techniques, ESCAPE considers the semantic descriptions of the topics to assess the level of separation of the clusters. This is not considered in the state-of-the-art approaches, that only evaluate the goodness of the results by means of probabilistic metrics. In ESCAPE the quantitative indices of confidence could be used instead to deeper analyse the proposed results.

6.7. Comparison between joint-approach and LDA

An analyst can be interested in analysing the difference between the two types of partitions obtained using the two strategies. To this aim, ESCAPE compares the best solutions found by the two different methodologies computing the ARI index, which give us a quick comparison of the obtained partitions.

	Weighting scheme				
Dataset	TF-IDF	LogTF-IDF	TF-Entropy	LogTF-Entropy	Boolean
D1	0.554	0.321	0.320	0.100	0.790

Table 16. Adjuster Rand Index for Dataset D1

The ARI index between the best partitions of the two methodologies is reported in Table 16. We can observe that the results are quite different and analysing only the previous table is not sufficient to draw conclusions on the two methodologies. Since the Boolean-IDF and Boolean-Entropy are very similar in terms of partitions for the joint-approach, we only consider the weight Boolean-Entropy for the comparison with respect to the Boolean-TF weight.

We recall also that the ARI index penalises the partitions with different numbers of clusters more than the Rand Index; however, especially for the weighting LogTF-Entropy, the comparison value is really poor.

To analyse in a major detail the partitions obtained, ESCAPE includes several graphical representations that are self-explained. These proposed graphical representations are exploited to simplify and synthesise the extracted knowledge patterns in a compact, human-readable, detailed and exhaustive representation.

For each experiment, ESCAPE reports the proposed visualisation techniques, allowing different stakeholders to easily capture the high-level overview of topic detection in each corpus.

We recall that the two highest similarity weighting schemes are the TF-IDF and the Boolean for both the topic modelling approaches. The partitions are not the same because the ARI index tends to 0.554 and 0.790, respectively. Still, analysing only the values is not sufficient to quantify the similarity between the topics. Below, we have reported the analysis of these two weighting strategies to highlight the main differences between the two approaches.

6.7.1. TF-IDF weight

Here, we have analysed the impact of the TF-IDF weighting function on both the methodologies integrated in ESCAPE. To this aim, we have reported the word-cloud comparison for the weighting scheme TF-IDF for both the methodologies. Specifically, in Figure 11 are reported the 10 word-clouds related to the joint-approach, while in Figure 12 are reported the 10 ones related to the LDA modelling. By analysing the most probable words for each topic, we can extract the following considerations.

In both the partitions found, we have 10 clusters. However, the partitions should not be the same, since the value of the ARI index is not 1. Moreover, we recall that the 5 a-priori known categories are: *cooking*, *literature*, *mathematics*, *music* and *sport*. We expect to find these themes in the 10 partitions.

Firstly, we reported a summary of the found topic in Table 17. Although the partitions are equivalent in number (10 topics), the meaning of the topics found are different. In fact, the five macro categories are correctly identified by both approaches, but the algebraic method finds subdivisions for the mathematics and sport categories, while the probabilistic method for literature and sports. Both the results are satisfactory.

We have also included the correlation analysis of the discovered partitions. For the joint approach we have reported the correlation matrix in terms of hot-cold topic. In this way, the colors help the analyst to read the possible correlation between topics. We have used the red color to highlight correlation between partitions (see Figure 13). Meanwhile,

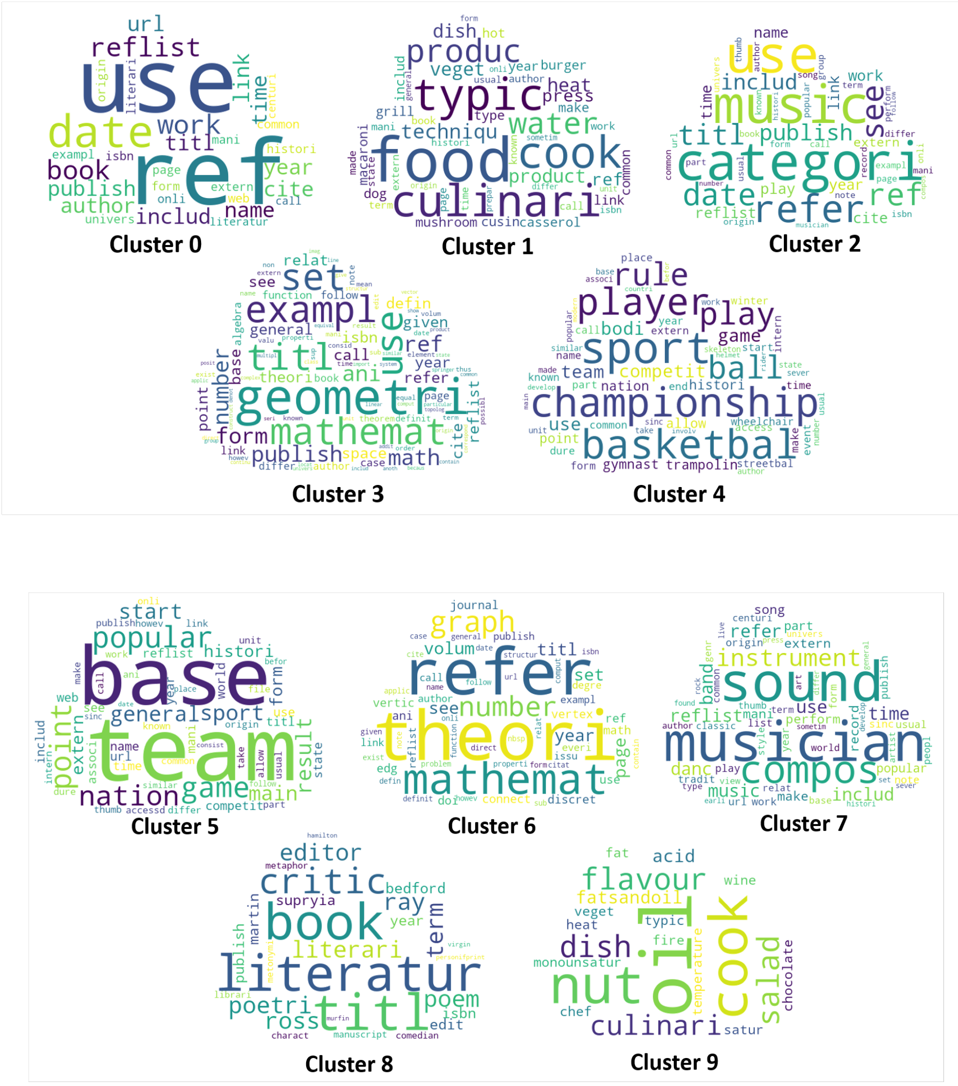


Figure 11. D1, word-cloud representation, TF-IDF weighting scheme for joint-approach.

in the probabilistic approach we have reported the graph representation, which is able to help the end-user to analyse the possible intersection between words in the different topics (see Figure 14). To compute the correlation matrix, ESCAPE first sorts the clusters based on their cardinality, then calculates the correlation between all the pairs of documents.

From Figure 13, we can notice a high correlation between cluster 4 and 5, which analysing Table 17, (column Topic Joint-Approach) are both related to sports. Moreover, there is another correlation between 3 and 6, which looking always at Table 17 or also the previously presented word-clouds, are both related to maths topics. Specifically, cluster 3 is related to several maths topics, while cluster 6 is inherent mainly to graph theory.

Instead, Figure 14 reports the graph representation for the probabilistic LDA modelling. The most relevant words for each topic, (i.e., the words which are most likely to belong to a particular topic) are well-separated, as can be deduced from the graph analysis. Considering both the top-20 (see Figure 14 (Left)) and the top-40 (see Figure 14 (Right)) words, the graph is still very disconnected, indicating that the analysed partitions are well separated.

Another way to compare the found partitions wrt the two approaches is the analysis of the t-SNE representations, which give the analyst the possibility to plot into a lower space (i.e., 2D in our framework) the high dimensional data under analysis. This representation

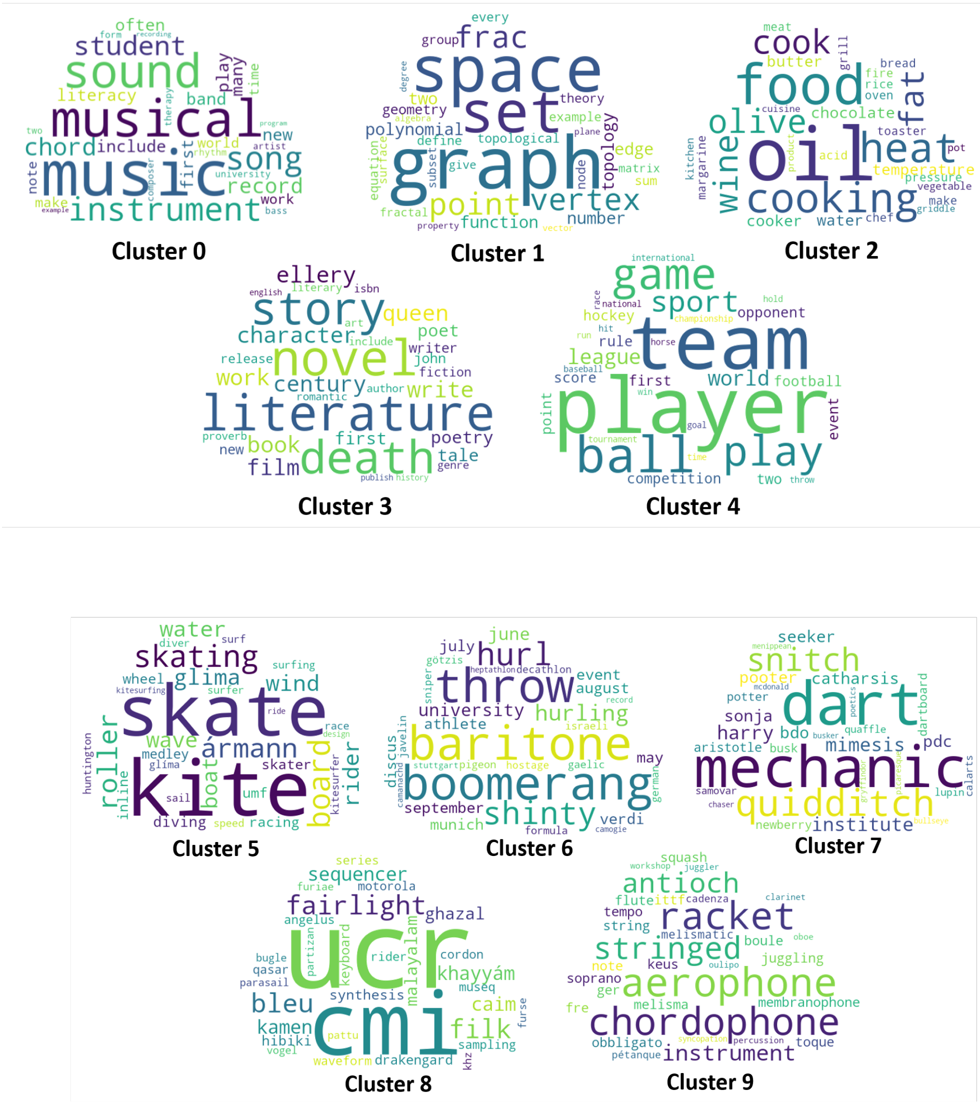


Figure 12. D1, word-cloud representation, TF-IDF weighting scheme for LDA modelling.

ClusterID	Topic Joint-Approach	Topic probabilistic Modelling
Cluster0	Literature	Music
Cluster1	Food	Maths
Cluster2	Music	Oil Food
Cluster3	Maths	Literature
Cluster4	Sport	Sport
Cluster5	Sport	Dynamic sport
Cluster6	Graph Theory	Music
Cluster7	Music	Quiddich - Literature
Cluster8	Literature	Literature
Cluster9	Oil	Musical Instruments

Table 17. Topic description for dataset D1 for both the approaches.

is reported in Figure 15. We recall that the T-distributed Stochastic Neighbor Embedding (t-SNE) is a machine learning algorithm for visualisation, which is based on a non-linear

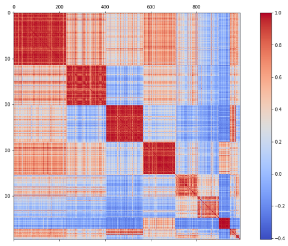
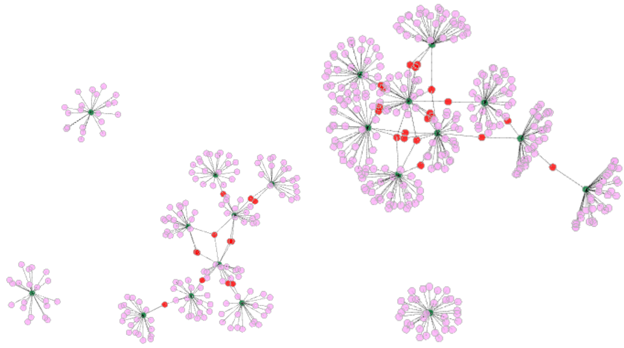


Figure 13. D1 Hot-topic correlation matrix representation, TF-IDF weighting scheme, K 10, joint approach.



(a) D1, TF-IDF, K 3. **(b)** D1, TF-IDF, K 6.

Figure 14. D1 graph representation, TF-IDF weighting scheme, K 10, Probabilistic approach, considering the top-20 (left) and the top-40 (right) words.

dimensionality reduction technique well-suited for embedding high-dimensional data for visualisation in a low-dimensional space. It is based on the concept of probability distribution, indeed it constructs a probability distribution over pairs of high-dimensional objects in such a way that similar objects have a high probability of being picked, whilst dissimilar points have an extremely small probability of being picked.

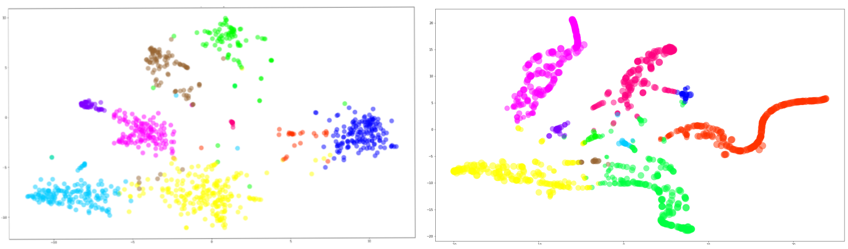


Figure 15. D1 t-SNE representation, TF-IDF weighting scheme, K 10, Joint-approach (Top) and Probabilistic approach (Bottom).

A key feature aspect of t-SNE is a tunable parameter, *perplexity*, which we have presented as a quality metric to evaluate the goodness of the probabilistic LDA modelling. This parameter says how to balance attention between local and global aspects of the data under analysis. The parameter is related to the concept of the number of close neighbours each point has. The perplexity value has a complex effect on the resulting pictures, in fact, since the algebraic model is not born to measure the perplexity in probabilistic terms, the good value to be set for its plot could be complex to infer. In Figure 15 we have reported the representations of the t-SNE visualisation for the joint approach (Top) and for the probabilistic approach (Bottom). The shape is quite similar, however the plot using the LDA model converges better in the presented figures. Probably, it is bad news that to see global geometry shape it is necessary a fine-tuning perplexity parameter. Moreover, since real data are characterised by multiple clusters with different cardinality (i.e., number of documents), it could happen that using only one single perplexity value is not enough to capture distances across all clusters. Indeed, the perplexity metric is a global parameter defined for the entire model. Thus, an interesting area for future researches could be the fixing of this problem.

6.7.2. Boolean weight

While analysing the ARI between the two approaches for dataset D1, the highest value is computed for the Boolean weighting strategy. It highlights a great similarity between the two partitions. Moreover, the number of documents in each cluster is comparable. In the joint-approach we have integrated two weighting strategies wrt the local weight Boolean, which are Boolean-IDF and Boolean-Entropy. However, since the two partitions were really similar, we only consider the Boolean-IDF as comparison wrt the Boolean-TF used for the LDA modelling.

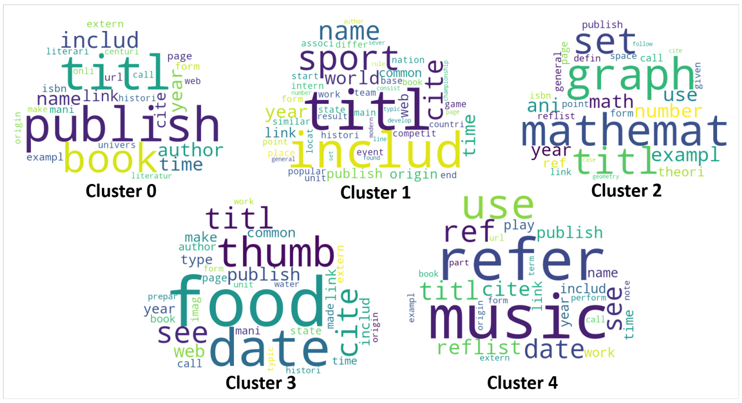


Figure 16. D1 word-cloud representation, Boolean-IDF weighting scheme, joint approach.

We have reported in Figure 16 and Figure 17 the word-clouds of the two approaches, respectively. Specifically, Figure 16 is related to the five-topic found using the algebraic approach, while Figure 17 is related to the probabilistic model. In detail, analysing Figure 16, we can observe that wrt to the TF-IDF local weight, the analysis is less precise. We can extract the main topic from each word-cloud; however, the partitions present more common words used for more topics.

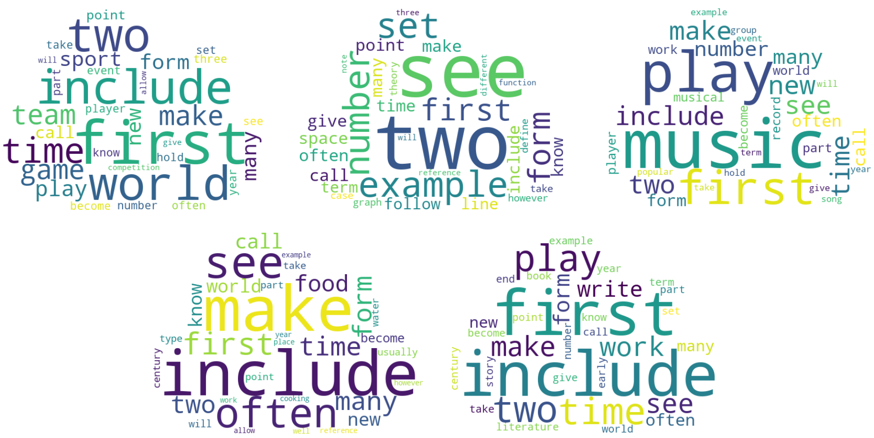


Figure 17. D1 word-cloud representation, Boolean-TF weighting scheme, *K* 5, probabilistic modelling.

For the probabilistic model, we can observe that when we consider the clustering obtained with K equal to 5 and its topic descriptions, when looking at the word clouds in Figure 17, many terms (such as *include* or *first*) appear to be in all the groups of the most significant words for each cluster. This happens because the Boolean-TF weighting scheme gave more relevance to words which appear most in the whole corpus, without penalising them. However, it could mean that these words do not belong to any specific topic, or they just do not bring any additional information useful for the topic modelling description phase. To this aim, we have included a post-processing phase for this particular weighting scheme.

K	**Topic description**
1	game, team, sport, player, event, competition, ball, rule, international, must, country, united, man, national, run
2	space, theory, case, graph, define, function, note, every, write, order, result, element, must, system, general
3	music, musical, player, record, song, event, write, release, instrument, note, sound, international, style, piece, back
4	food, water, cooking, united, sometimes, produce, result, high, oil, modern, large, require, must, list, process
5	write, book, literature, story, character, art, university, music, novel, modern, english, word, note, study, later

Table 18. D1 topic-terms representation, Boolean-TF weighting scheme, K 5, probabilistic modelling.

In order to not consider these terms and bring up the words characterising the topics identified by the LDA modelling process, we have decided to apply a further post-processing step to evaluate the results. Once the models have been created and the K values selected, we took into consideration more words to describe the topics, and then we removed from them all the words appearing at least in four topic representations.

The results obtained by this post-processing operation are reported in Table 18. In this way, the most common words that do not carry any specific information have been excluded from the descriptions, and the terms relevant for the meaning of the categories are visible to the analysts. As a matter of fact, the assigned labels to the clusters generated by the LDA model cover the following main topics: *sport*, *mathematics*, *music*, *cooking*, *literature*. Using this post-processing approach, we are able to describe perfectly the macro-categories of this data set.

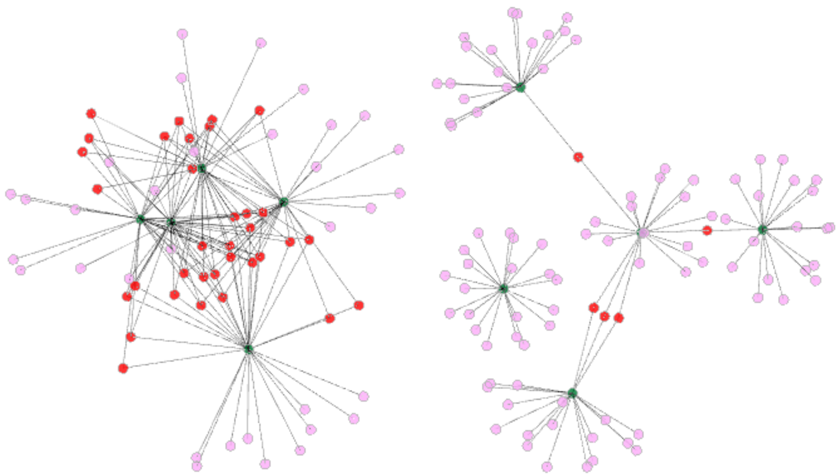


Figure 18. D1 t-SNE representation, Boolean-TF weighting scheme, K 5, without post-processing (Left) and with post-processing (Right).

To better show the impact of removing words that appear at least in four topics, we reported the graph representation before and after this improvement. Figure 18 shows the graph representation analysing the top-20 words for each topic. Specifically, on the left, is reported the case without the post-processing, while on the right, we reported the case with the proposed post-processing. The first graph is more connected wrt to the second one; moreover, from the analysis of the graph after the post-processing, we can see the improvement of this phase, since the new graph is not connected at all. This means that the words that describe each topic are well-separated from cluster to cluster.

7. Discussion

From the analysis of the obtained experimental results, we can assess that ESCAPE performs well in describing the six corpora under analysis, clustering the documents based on their main content. The proposed framework is generally able to group the documents into well separated topics.

We have observed that the joint approach, which is based on a dimensionality algebraic phase before the application of the partitional K-Means algorithms, is able to find homogeneous partitions in terms of documents for each cluster. In other words, this approach creates more balanced clusters. Moreover, changing the weighting strategy, the end-user is able to clusterise the same dataset, at different granularity levels. Specifically, we have seen that the global weight IDF is able to create more clusters able to find also sub-topics related to the major category. so, this weighting scheme is able to characterise each dataset in a more precise way. On the other hand, the Entropy is able to find larger clusters, finding only the main relevant topic associated with each partition. Indeed, both the clusterizations are able to split the corpora into well separated groups.

For the probabilistic approach, considering the semantic similarity among the produced topics, it turned out that outperforms the current used approach to find the proper number of clusters. As a matter of fact, the proposed algorithm is able to capture the effective cohesion level of the clusters, and then properly identify the optimal number of topics. The results obtained from all the datasets considered in the thesis confirm the clusters to be well separated, especially for certain weighting schemes such as TF-IDF. Nevertheless, wrt the joint-approach, some weighting schemes lead to very poor results, such as the Entropy-based scheme. In general, the probabilistic model tends to find more inhomogeneous clusters; however, despite these schemes, the other results are also satisfactory.

ESCAPE turns out to be really helpful for the analysts during the analytic tasks. Indeed, the analyst can choose to assign to the words in the documents different relevance by means of different weights and compare the solutions obtained using the two approaches, analysing the different granularity levels. The best partitions can also be compared using innovative visualisation techniques, which are able to help the analyst during the validation step. Moreover, the two proposed approaches are able to characterise different aspects in which the analyst may be interested, including also the possibility of comparing the proposed approaches wrt the other state-of-the-art techniques.

8. Conclusion and Future Work

This paper has presented the ESCAPE framework (**E**nhanced **S**elf-tuning **C**haracterisation of document collections **A**fter **P**arameter **E**valuation), which is able to support the user during all the phases of the analysis process tailored to textual data. ESCAPE includes three main building blocks to streamline the analytics process and to derive high-quality information in terms of well-separated and well-cohesive groups of documents characterising the main topics in a given corpus.

Firstly, the data distribution of each corpus is characterised by several statistical indices (e.g. Guiraud Index, TTR). The joint analysis of these statistical features is able to describe the lexical richness and characterise the data distribution of each collection under analysis. Then, a pre-processing phase is applied to prepare the textual content of documents for the next phases. These activities, which are done subsequently, represent each document via the Bag-Of-Word (BOW) representation. Using this model, a text (e.g. a sentence or a document) is represented as the bag (multi-set) of its words, disregarding grammar and even word order but keeping multiplicity. To measure the relevance of these multiplicities, ESCAPE includes several weighting strategies, which are able to measure term relevance in the same dataset by exploiting a local weighting scheme (e.g. TF, LogTF) together with a global weighting scheme (e.g. Entropy, IDF). ESCAPE automatically exploits all the possible combinations of local and global weighting schemes to suggest to the user the ones that well model the term relevance in the collection under analysis. Since we are interested in finding out the number of topics contained in a given collection of documents, in ESCAPE we have integrated two strategies because no strategy is universally superior.

Specifically, we have integrated:

- an algebraic model based on SVD decomposition together with the K-Means clustering algorithm (i.e., the joint-approach);
- a probabilistic model, based on the analysis of latent variables through the LDA (i.e., the probabilistic method).

Each strategy has been enriched with a self-tuning methodology to automatically set the specific-input parameters required by each involved algorithm. This frees the end user from the correct configuration of the input parameters, which is usually a time consuming activity. Lastly, several user-friendly and exhaustive informative dashboards have been embedded to help the end-user to effectively and efficiently explore the results. To evaluate the quality of corpora partitions automatically discovered by ESCAPE, a variety of quality indices have been integrated into the proposed framework.

Possible future extensions concern the *integration* in ESCAPE of:

1. *New data analytics algorithms* to exploit other interesting models. Specifically, we are currently including:
    - other *algebraic data reduction algorithms* (e.g. Principal Component Analysis (PCA)) for the joint-approach together with the exploitation of other clustering methods (e.g. hierarchical algorithm) and other *probabilistic topic modelling methods* (e.g. Probabilistic Latent Semantic Analysis (pLSA));
    - *autoencoder-based data reduction algorithms* to compress the information of the input variables into a reduced dimensional space and then recreate the input data set;
    - more *weighting functions* (e.g. aug-norm) to underline the relevance of specific terms in the collection;
    - more *statistical indices* to characterise the corpora distribution (e.g., [64]), and innovative strategies to extend the ability of ESCAPE to be more domain-adaptive ([65]).
    - *Deep Learning models* to deal with a large set of corpora characterized by a variable data distribution. These models can be used either to improve the preprocessing phase or to facilitate the modeling task by shifting the current methods to the supervised ones.
2. A *semantic component*: (e.g. WordNet [66]) able to support the analyst in a double phase. Such component would be useful both during the pre-processing phase, to eliminate semantically bound words, in this way we are able to reduce the dictionary and also the complexity of the algorithms, also during the post-processing phase. In this way, it would be possible to analyse through the most relevant words for each topic, those that are related to each other, helping the analyst in understanding the outputs. Specifically, each topic can be characterised by words which are semantically related, and so could represent subtopic of the same macro category. Moreover, thanks to the ontological base, the analyst could also add a hierarchy level for each word of the dictionary to support other analytics tasks (e.g. generalised association rules discovery).

3. A *Knowledge-Base*: to store all the results of the experiments, including the data characterisation and the top-k selected results, for each methodology and weighting scheme to efficiently support self-tuning methodologies.
4. A *self-learning methodology*: based on a classification algorithm trained on the knowledge base content to forecast the best methods for future analyses. So, when a new collection needs to be analysed, ESCAPE should compute the data distribution characterisation through statistical features and suggest possible good configurations without performing all the analytics tasks.

Author Contributions: Conceptualization, E.D.C. and S.P.; methodology, E.D.C., S.P. and T.C.; software, E.D.C., S.P., B.V. and P.B.; validation, E.D.C., S.P., B.V. and P.B.; investigation, E.D.C., S.P., B.V. and P.B.; data curation, E.D.C., S.P., B.V.. and P.B.; writing—original draft preparation, E.D.C, S.P. and T.C.; writing—review and editing, E.D.C., S.P., B.V., P.B. and T.C.; visualization, E.D.C., S.P., B.V. and P.B.; supervision, T.C.; funding acquisition, T.C.. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

Di Corso, E.; Cerquitelli, T.; Ventura, F. Self-tuning techniques for large scale cluster analysis on textual data collections. Proceedings of the Symposium on Applied Computing. ACM, 2017, pp. 771–776.

Di Corso, E.; Ventura, F.; Cerquitelli, T. All in a twitter: Self-tuning strategies for a deeper understanding of a crisis tweet collection. In *IEEE BigData 2017, Boston, MA, USA* [2], pp. 3722–3726. doi:10.1109/BigData.2017.8258369.

Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *Journal of machine Learning research* **2003**, *3*, 993–1022.

Li, C.; Duan, Y.; Wang, H.; Zhang, Z.; Sun, A.; Ma, Z. Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Transactions on Information Systems (TOIS)* **2017**, *36*, 11.

Li, C.; Chen, S.; Xing, J.; Sun, A.; Ma, Z. Seed-Guided Topic Model for Document Filtering and Classification. *ACM Transactions on Information Systems (TOIS)* **2018**, *37*, 9.

Maña-López, M.J.; De Buenaga, M.; Gómez-Hidalgo, J.M. Multidocument summarization: An added value to clustering in interactive retrieval. *ACM Transactions on Information Systems (TOIS)* **2004**, *22*, 215–241.

Liang, S.; Yilmaz, E.; Shen, H.; Rijke, M.D.; Croft, W.B. Search result diversification in short text streams. *ACM Transactions on Information Systems (TOIS)* **2017**, *36*, 8.

Fadda, E.; Perboli, G.; Tadei, R. Customized multi-period stochastic assignment problem for social engagement and opportunistic IoT. *Computers & Operations Research* **2018**, *93*, 41–50.

Jung, H.; Lee, B.G. Research trends in text mining: Semantic network and main path analysis of selected journals. *Expert Systems with Applications* **2020**, *162*, 113851. doi:https://doi.org/10.1016/j.eswa.2020.113851.

Saxena, G.; Santurkar, S. An Iterative MapReduce Framework for Sports-based Tweet Clustering. Proceedings of the Sixth International Conference on Computer and Communication Technology 2015; ACM: New York, NY, USA, 2015; ICCCT '15, pp. 9–14.

Abualigah, L.; Khader, A.T.; Hanandeh, E., A Novel Weighting Scheme Applied to Improve the Text Document Clustering Techniques; 2018; pp. 305–320. doi:10.1007/978-3-319-66984-7_18.

Proto, S.; Di Corso, E.; Ventura, F.; Cerquitelli, T. Useful ToPIC: Self-Tuning Strategies to Enhance Latent Dirichlet Allocation. 2018 IEEE International Congress on Big Data (BigData Congress). IEEE, 2018, pp. 33–40.

Dieng, A.B.; Ruiz, F.J.R.; Blei, D.M. Topic Modeling in Embedding Spaces. *Transactions of the Association for Computational Linguistics* **2020**, *8*, 439–453. doi:10.1162/tacl_a_00325.

Sia, S.; Dalmia, A.; Mielke, S.J. Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too! Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); Association for Computational Linguistics: Online, 2020; pp. 1728–1736. doi:10.18653/v1/2020.emnlp-main.135.

Thompson, L.; Mimno, D. Topic Modeling with Contextualized Word Representation Clusters. *ArXiv* **2020**, *abs/2010.12626*.

Garcia, K.; Berton, L. Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA. *Applied Soft Computing* **2021**, *101*, 107057. doi:10.1016/j.asoc.2020.107057.

Peng, H.; Li, J.; Song, Y.; Yang, R.; Ranjan, R.; Yu, P.S.; He, L. Streaming Social Event Detection and Evolution Discovery in Heterogeneous Information Networks. *ACM Trans. Knowl. Discov. Data* **2021**, *15*. doi:10.1145/3447585.

Fard, M.M.; Thonet, T.; Gaussier, E. Seed-Guided Deep Document Clustering. Advances in Information Retrieval; Jose, J.M.; Yilmaz, E.; Magalhães, J.; Castells, P.; Ferro, N.; Silva, M.J.; Martins, F., Eds.; Springer International Publishing: Cham, 2020; pp. 3–16.

Allan, J.; Carbonell, J.; Doddington, G.; Yamron, J.; Yang, Y.; et al. Topic detection and tracking pilot study: Final report. Proceedings of the DARPA broadcast news transcription and understanding workshop. Citeseer, 1998, Vol. 1998, pp. 194–218.

Zhang, C.; Lu, S.; Zhang, C.; Xiao, X.; Wang, Q.; Chen, G. A Novel Hot Topic Detection Framework With Integration of Image and Short Text Information From Twitter. *IEEE Access* **2019**, *7*, 9225–9231. doi:10.1109/ACCESS.2018.2886366.

Mamo, N.; Azzopardi, J.; Layfield, C. Fine-grained Topic Detection and Tracking on Twitter. 2021, pp. 79–86. doi:10.5220/0010639600003064.

Bouaziz, A.; da Costa Pereira, C.; Pallez, C.D.; Precioso, F. Interactive Generic Learning Method (IGLM): A New Approach to Interactive Short Text Classification. Proceedings of the 31st Annual ACM Symposium on Applied Computing; ACM: New York, NY, USA, 2016; SAC '16, pp. 847–852. doi:10.1145/2851613.2851646.

Duchrow, T.; Shtatland, T.; Guettler, D.; Pivovarov, M.; Kramer, S.; Weissleder, R. Enhancing navigation in biomedical databases by community voting and database-driven text classification. *BMC Bioinformatics* **2009**, *10*, 317.

Linmei, H.; Yang, T.; Shi, C.; Ji, H.; Li, X. Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); Association for Computational Linguistics: Hong Kong, China, 2019; pp. 4821–4830. doi:10.18653/v1/D19-1488.

Pang, B.; Lee, L. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.* **2008**, *2*, 1–135.

Vashishtha, S.; Susan, S. Highlighting keyphrases using senti-scoring and fuzzy entropy for unsupervised sentiment analysis. *Expert Systems with Applications* **2021**, *169*, 114323. doi:https://doi.org/10.1016/j.eswa.2020.114323.

Jamil, H.M.; Jagadish, H.V. A Structured Query Model for the Deep Relational Web. Proceedings of the 24th ACM International on Conference on Information and Knowledge Management; ACM: New York, NY, USA, 2015; CIKM '15, pp. 1679–1682.

Dean, J.; Ghemawat, S. MapReduce: simplified data processing on large clusters. OSDI'04, 2004, pp. 10–10.

Zaharia, M.; Chowdhury, M.; Das, T.; Dave, A.; Ma, J.; McCauley, M.; Franklin, M.J.; Shenker, S.; Stoica, I. Resilient Distributed Datasets: A Fault-tolerant Abstraction for In-memory Cluster Computing. NSDI'12, 2012.

Alper, P.; Belhajjame, K.; Goble, C.A.; Karagoz, P. Enhancing and abstracting scientific workflow provenance for data publishing. Proceedings of the Joint EDBT/ICDT 2013 Workshops. ACM, 2013, pp. 313–318.

Singh, J.N.; Dwivedi, S.K. A comparative study on approaches of vector space model in information retrieval. International Conference of Reliability, Infocom Technologies and Optimization, 2013.

Cerquitelli, T.; Di Corso, E.; Ventura, F.; Chiusano, S. Data miners' little helper: data transformation activity cues for cluster analysis on document collections. Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics. ACM, 2017, p. 27.

Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **1990**, *41*, 391–407.

KPFRS, L. On Lines and Planes of Closest Fit to Systems of Points in Space. Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (SIGMOD), 1901.

Calafiore, G.C.; Ghaoui, L.E.; Preziosi, A.; Russo, L. Topic analysis in news via sparse learning: a case study on the 2016 US presidential elections. *IFAC-PapersOnLine* **2017**, *50*, 13593 – 13598. doi:https://doi.org/10.1016/j.ifacol.2017.08.2380.

Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. Advances in neural information processing systems, 2002, pp. 601–608.

Hofmann, T. Probabilistic latent semantic analysis. Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.

Samuel, J.; Ali, G.G.M.N.; Rahman, M.M.; Esawi, E.; Samuel, Y. COVID-19 Public Sentiment Insights and Machine Learning for Tweets Classification. *Information (Switzerland)* **2020**, *11*, 1–22. doi:10.3390/info11060314.

Weiss, S.M.; Indurkhya, N.; Zhang, T. *Fundamentals of predictive text mining*; Springer, 2015.

Cerquitelli, T.; Corso, E.D.; Ventura, F.; Chiusano, S. Prompting the data transformation activities for cluster analysis on collections of documents. Proceedings of the 25th Italian Symposium on Advanced Database Systems, Squillace Lido (Catanzaro), Italy, June 25-29, 2017., 2017, p. 226.

Nakov, P.; Popova, A.; Mateev, P. Weight functions impact on LSA performance. EuroConference RANLP'2001 (Recent Advances in NLP, 2001, pp. 187–193.

Xie, P.; Xing, E.P. Integrating document clustering and topic modeling. *arXiv preprint arXiv:1309.6874* **2013**.

Juang, B.H.; Rabiner, L. The segmental K-means algorithm for estimating parameters of hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing* **1990**, *38*, 1639–1641.

Singh, N.S.D. Performance Evaluation of K-Means and Heirarichal Clustering in Terms of Accuracy and Running Time.

Chakraborty, S.; Nagwani, N.; Dey, L. Performance Comparison of Incremental K-means and Incremental DBSCAN Algorithms. *International Journal of Computer Applications (0975 – 8887)* **2011**, *27*, 975–8887.

Fränti, P.; Sieranoja, S. How much can k-means be improved by using better initialization and repeats? *Pattern Recognition* **2019**, *93*, 95–112. doi:https://doi.org/10.1016/j.patcog.2019.04.014.

Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* **1987**, *20*, 53 – 65.

Cerquitelli, T.; Chicco, G.; Di Corso, E.; Ventura, F.; Montesano, G.; Armiento, M.; González, A.M.; Santiago, A.V. Clustering-Based Assessment of Residential Consumers from Hourly-Metered Data. 2018 International Conference on Smart Energy Systems and Technologies (SEST). IEEE, 2018, pp. 1–6.

Hoffman, M.; Bach, F.R.; Blei, D.M. Online Learning for Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 23*; Lafferty, J.D.; Williams, C.K.I.; Shawe-Taylor, J.; Zemel, R.S.; Culotta, A., Eds.; Curran Associates, Inc., 2010; pp. 856–864.

Zhao, W.; Chen, J.J.; Perkins, R.; Liu, Z.; Ge, W.; Ding, Y.; Zou, W. A heuristic approach to determine an appropriate number of topics in topic modeling. *BMC bioinformatics* **2015**, *16*, S8.

Quinlan, J.R. Induction of decision trees. *Machine Learning* **1986**, *1*, 81–106. doi:10.1007/BF00116251.

Hörster, E.; Lienhart, R.; Slaney, M. Image retrieval on large-scale image databases. Proceedings of the 6th ACM international conference on Image and video retrieval. ACM, 2007, pp. 17–24.

Rand, W. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* **1971**, *66*, 846–850.

Hubert, L.; Arabie, P. Comparing partitions. *Journal of classification* **1985**, *2*, 193–218.

Vinh, N.X.; Epps, J.; Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research* **2010**, *11*, 2837–2854.

van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **2008**, *9*, 2579–2605.

Heimerl, F.; Lohmann, S.; Lange, S.; Ertl, T. Word cloud explorer: Text analytics based on word clouds. System Sciences (HICSS), 2014 47th Hawaii International Conference on. IEEE, 2014, pp. 1833–1842.

Olteanu, A.; Castillo, C.; Diaz, F.; Vieweg, S. CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises. International AAAI Conference on Web and Social Media, 2014.

Griffiths, T.L.; Steyvers, M. Finding scientific topics. *Proceedings of the National academy of Sciences* **2004**, *101*, 5228–5235.

Saleh, I.; El-Tazi, N. Automatic Organization of Semantically Related Tags Using Topic Modelling. Advances in Databases and Information Systems. Springer, 2017, pp. 235–245.

Wood, J.; Tan, P.; Wang, W.; Arnold, C. Source-LDA: Enhancing probabilistic topic models using prior knowledge sources. Data Engineering (ICDE), 2017 IEEE 33rd International Conference on. IEEE, 2017, pp. 411–422.

Pang-Ning T. and Steinbach M. and Kumar V.. *Introduction to Data Mining*; Addison-Wesley, 2006.

Zhang, W.; Cui, Y.; Yoshida, T. En-LDA: An Novel Approach to Automatic Bug Report Assignment with Entropy Optimized Latent Dirichlet Allocation. *Entropy* **2017**, *19*, 173.

Wang, Z.; Du, B.; Tu, W.; Zhang, L.; Tao, D. Incorporating Distribution Matching into Uncertainty for Multiple Kernel Active Learning. *IEEE Transactions on Knowledge and Data Engineering* **2019**, *PP*, 1–1. doi:10.1109/TKDE.2019.2923211.

Wang, Z.; Du, B.; Guo, Y. Domain Adaptation With Neural Embedding Matching. *IEEE Transactions on Neural Networks and Learning Systems* **2019**, *PP*, 1–11. doi:10.1109/TNNLS.2019.2935608.

Miller, G. *WordNet: An electronic lexical database*; MIT press, 1998.