POLITECNICO DI TORINO

Management Engineering
Master's Degree

Master's Thesis

Tailoring the Knowledge Data Discovery process to e-commerce reviews

Supervisors
Prof. Tania Cerquitelli
Dott.Evelina Di Corso

Candidate
Francesca Gubbiotti

Academic Year 2018/2019

To someone I never met and I will keep forever in my heart

Abstract

This thesis mainly concerns data driven technologies and the related potentiality to bring out, from textual data, previously unknown knowledge. The goal, for the current work case, is to extract potentially useful information from Amazon products reviews. To this aim, Knowledge Data Discovery (KDD) process, applied on textual data collection, has been tailored on Amazon reviews dataset.

First of all, a broad description of text mining with its benefits and pitfalls has been introduced along with existing algorithm and methodologies; then, ESCAPE engine has been studied, tailored and proposed as a not-time consuming and low-computational cost solution. The proposed tool approaches and integrates all the building blocks of KDD processes such as data processing and characterization, self-Tuning exploratory data analytics, and knowledge validation and visualization. Two different approaches with self-tuning algorithms for the exploratory phase are also included.

Before running the experiments, Amazon Web Service platform (AWS) case and the importance of text data analysis for business choice have been discussed.

A large number of experiments, applying the two approaches, have been performed on almost twenty products reviews dataset, some of them specifically built during the development of the thesis according to the work needs. Experimental results have been finally analysed, validated and visualized with several techniques in order to show, from a technical point of view, the performances of both the two ESCAPE approaches and the strategies used within them, and, from a business analysis point of view, interesting features among the different products categories comments.

Contents

Introduction

This is the era of big data.

A report of IDC [6], published at the end of 2018, predicts that the collective sum of the world's data will grow from 33 zettabytes (at December 2018) to a 175ZB by 2025, for a compounded annual growth rate of 61%. E-commerce represents a large chunk of this digital universe since the constant development of hardware and software platforms for online shopping is enabling the rapid creation of huge repositories of several kinds of data which are growing even more day by day. By analysing these data, companies can understand customers' purchasing behaviour and gain a competitive advantage. In this context products reviews are a huge source of information from which useful knowledge can be derived. This is what text mining aims to do: analyse textual content and discover relationship and patter within them, but, to deal with the high volume and complexity of the data, effective methodologies have to be developed.

Starting from these considerations the project thesis has been developed with a dual purpose. The first one has a more academic nature and it is the study, tailoring and validation of ESCAPE engine, a data driven methodology firstly developed by Evelina Di Corso during her doctoral project. The second one is more related to business analysis aiming to discover relevant informations from Amazon products reviews which can be exploited for improving companies business choice.

The first chapter is dedicated to investigate the particular structure of textual content and to give a description of the steps involved in the knowledge discovery processes. An overview of the existing methodologies with a particular focus on clustering and topic modelling is also provided.

In the second chapter, ESCAPE engine is studied, tailored and proposed as a data-driven solution which address all the building blocks of the KDD processes.

In chapter 3 ESCAPE is applied on a large number of datasets. Actually this chapter consists of three parts. The first part is for introducing Amazon Web Service platform as the source of the data under analysis but also as a solution for many issues when working

with big data. The second part describes, analyses and visualize the experimental results obtained running ESCAPE with different strategies and approaches on the retrieved datasets. In the third part novel datasets are specifically built in order to stronger investigate and validate ESCAPE abilities.

The last chapter provides a summary of the obtained results along with technical consideration about ESCAPE performances and potential improvements. Also interesting features discovered within the datasets will be included.

1 Text Data Mining

Text data mining is the process of mining raw large-size textual data in order to discover and bring out previously hidden and valuable information. it includes grouping documents with similar properties or similar content, topic modelling, clustering web services and long text summarizing. These techniques allow to detect patterns, trends and behaviours that may become knowledge. Text mining uses natural language processing (NPL) to transform the free text in documents and databases into normalized, structured data suitable for analysis. If text mining deal with the text itself, NPL deals with the underlying metadata, it performs a special kind of linguistic analysis that essentially helps a machine “read” text. Many people treat data mining as a synonym for, knowledge discovery from data, or KDD, while others view data mining as merely an essential step in the process of knowledge discovery [2]. In this context data mining is defined as the fifth, out of seven sequential steps in the knowledge discovery process after data cleaning, integration, selection and transformation and before data evaluation and presentation. However, in many fields, such as industry, media, and research milieu, the term data mining is often used to refer to the entire knowledge discovery process so this broader view will be adopted in the current work. Basing on their structure, data can be classified as structured and unstructured data. The former are highly-organized and formatted, making them easily searchable while the latter are "everything else". It has to be clear that considering unstructured data, which have a heterogeneous nature and are unorganized, requires more work then structured data. Text data belong to this category and are, indeed, very variable, dirty, and different depending on the typology, the source, the target and the field of expertise. For this reason, several pre-processing steps are, almost always, needed to prepare data before running a program. Messy text has to be transformed into structured data, stored in rows and columns, and an acceptable number of concepts have to be identified. Then, standard data mining techniques (clustering, predictive modelling, classification) can be applied to discover potential relationship between concepts and hidden patterns.

Joining (structured) data mining and text mining can provide better insights than adopting

any one of the two because of the issue related with synonymy and polysemy which makes difficult to detect valid relationships between different parts of a text. Synonymy is when different words have the same meaning while polysemy is when a word has more than one meaning. To deal with this, text has to be transformed in structured data, then concept and category models have to be built in order to apply text mining algorithms and, only at end, standard data mining techniques can be used to discover link between concepts and bring out potentially high-quality information. It is common to use bag of words (BOW) representation for documents, accounting for the number of occurrences of each term but ignoring the order. This representation allows to balance computational efficiency with the need to retain the document content. Each dimension of the BOW vectors corresponds to a term in the documents so the dimensionality are very high. Dimension reduction methods can be then applied to find a lower-dimensional semantic space that preserve relationship and essential features.

After text pre-processing an intermediate step, assigning a weight to all of the Terms in the Document-BOW, can be included to streamline the data mining techniques application phase such as the clustering process and better derive the hidden intelligence.

1.1 Cluster analysis and topic modelling

The major and essential tasks involved in text data analysis concern clustering analysis and topic modelling.

The clustering problem is defined to be that of finding groups of similar objects in the data. [4] The cluster analysis aims to divides data into meaningful and useful groups named clusters. The similarity between the elements to divide is expressed by a proper similarity function and the usefulness of the clusters is defined by the goals of the data analysis. These two definitions are also the reasons why a notion of cluster cannot be

precisely identified; nevertheless, all the clustering algorithms have a common final aim that is grouping elements into well-separated groups.'Well-separated' means that a cluster is a set whose objects are closer (or more similar) to each other than objects assigned to different groups. Cluster analysis is sometimes referred to as unsupervised classification since there is no prior information about the group or cluster membership for any of the elements. Indeed these techniques, in contrast with (supervised) classification, derives class (cluster) labels only from data. The most known and studied distinction between clustering methods is if they are partitional or hierarchical. In the first case data are simply divided into non-overlapping sets (clusters) such that each data object belongs to only one set, instead in the hierarchical clustering, clusters are allowed to have sub-clusters and they are organized as a tree. Traditional methods for clustering have generally focussed on quantitative or categorical data but many of these algorithms can be extended to any kind of data including text. In this domain, objects to be clusters can be of different granularities such as documents, paragraphs, sentences or terms. The sparse and high dimensional representation of the text documents often become obstacles, this lead to the early use of dimensionality reduction algorithms to help and improve clustering process. Text clustering algorithms are divided into a wide variety of different types including agglomerative clustering algorithms, partitioning algorithms, and standard parametric modelling based methods.
There are two applications of the clustering methods that are specifically useful for this thesis case. The first one is corpus summarisation, which means providing cluster-outline or word-cluster to give a summary of the data collection and insights into the content of the underlying corpus.
The second, is building a Recommender system, a system that is capable of predicting the future preference of a set of items for a user, and recommend the top items. Nowadays, these systems are most commonly recognised as product recommender for E-Commerce websites like Amazon but also as a playlist generator for Video and Music service.

As already said text mining uses NPL. An important part of NPL is the Topic Detection and Tracking (TDT) problem; as reported in [6] , TDT programs aims to develop technologies

that search, organize and structure multilingual, news oriented textual materials from a variety of broadcast news media. These technologies operate in a dynamic way, on data which are continuously evolving and are collected in real time from a variety of sources.
The topic detection task evaluates technologies that detect previously unknown topics which are defined linking together stories or words that discuss the same topic. These systems must understand what constitute a topic independently from them since an a priori knowledge of topics is not given. The topic detection task in general is designed to be multilingual spanning languages in different clusters (groups of homogeneous elements within data) and detect clusters of stories or words that discuss the same topic. The multilingual characteristic is not going to be exploited in this thesis case since only one language (English) is used in the data. A story is a coherent set of information which involves one or more proposition about an event but what Fiscus and Doddington says about stories is true and can be applied also, on a lower level, to words.
The assessment of the performance is not trivial since stories frequently discuss multiple topics, as well as, words are, in many more cases, associated to several topics. This phenomenon means that clusters are dependent on previously processed words and the decomposition of performance into casual subsets is misleading. The multi-topic stories are considered unscorable even though the clustering is performed on all the test data. Thus, this kind of stories may have influence on the system but they do not affect the error measure. Performance assessment for topic detection uses topic-weighted measure. Topic detection can be addressed in two different ways: the retrospective topic detection and the online topic detection.
Using the first approach, each topic is detected into a stories corpus and is defined basing on the linked stories which can belong only to one cluster that represents the topic. With the second approach stories are elaborated one by one sequentially and the system decide if a story is about a novel topic or not, before the elaboration of the next story.

This second way is addressed trough topic modelling whose aim is to discover the latent themes (topics) assumed to have generated the documents of a corpus. Topic modelling

methods are based on distributional hypothesis, suggesting that similar words occur in similar contexts. Textual documents are represented as probability of words. There is a link between probabilistic topic models and dimensionality reduction methodologies. Indeed the former can provide an intuitive, probabilistic foundation for dimension reduction. They allow analysts to reason about the topics present in a document and expose the probability of seeing each word in any given topic.This makes it much easier to interpret what the topics mean.[4]

In this age, all the text mining processes are even more challenging because of the constantly increasing volume of documents: if, on one hand, all the information needed to solve almost any kind of problem is at fingertips, on the other one, people are not able to master the complexity, the dynamics, and the huge amount of available data. Text data can be any type of textual communication such as tweets, comments, reviews, emails, letters. These are technically called 'Documents', while words within them are called 'Terms'.

It is important to notice that there is a big distinction between data and information objects: the former are just raw symbols without any meaning while the latter are collections of data that carry out some semantics (knowledge) which is a pre-condition for correct interpretation. The point is, then, not just collect and process volumes of complex data but understand trends, uncover hidden patterns, detect anomalies, and so on.

Text data analysis is, though, a complex process and there is not a optimal way to do it, it depends on many factors (context, data type, data richness, computational requirements and many others). In the literature, there are a lot of algorithms to perform any phase, but for each one the specific parameters have to be manually set and validated; furthermore, a proper combination of these different analytics algorithms, should be defined in order to correctly model data from dataset with specific text characteristics. The experts are requested to make a lot of effort in order to correctly configure each algorithm, and this is even more tricky for people that don't have a deep knowledge. Strategies to automatically

select proper parameters have to be carefully assessed since they have a big impact on the analysis result, usually the assessment process require a lot of time and it is difficult to give a clear interpretation. More over data mining requires huge computational resources such as data warehouse or database and substantial processing power.
To streamline the analysis process and hide the underlying complexity, scalable and parameter-free solutions have to be explored. One of the main issue is, in fact, the parameter setting for each algorithm so auto-selection strategies to off-load the parameter tuning from end-user should be considered.

1.2 Text Mining Applications

One of the most relevant characteristic of textual data is that they offer real and full insights into phenomena more than quantitative data; for this reason, text mining has many application and helps in a lot of scenarios; its several tasks depend on the different fields where it is applied.
Text mining can be used to answer interesting, business questions and to optimise day-to-day operational efficiency; but also to improve long-term strategic decisions. Text miners, indeed, can provide a fundamental support to strategic decision-making process thanks to a faster and more efficient data analysis, extracting only the relevant information.
For this reason, it is one of the most important Business intelligence tools. Business intelligence is a set of organizational processes that allow companies turning raw data into useful and high-quality information [7] and there is a variety of fields where it can be applied: Sales, CRM (customer relationship management) analysis, HR, performance audit. Business intelligence embodies also other two, not mutually exclusive definition: the first in the one mentioned above; the second refers to the technology used to design and implement these processes and the third is referred to the knowledge obtained thanks to these processes. However, what is true is that organizations today rely on a set of automated tools for

knowledge discovery to gain business insight and intelligence. [7] Also middle-sized and small enterprises which lack the infrastructure and budget available for large enterprises, use mining tools. This is possible thanks to the new trend of Cloud Computing that helps in providing mining tools at relatively lower and acceptable costs.These companies can outsource and access through the web the actual data warehouse[8].

Thanks to the matching between the infrastructure economic efficiency and advanced software tools enterprises are adopting mining techniques as internal and essential business process. Indeed, they are becoming integral part of decision-making and provide the prevalent support system.

Text mining and natural language processing are also widely useful and used for customer care applications. Customers' relationship management CMR is improved mining real habits, patterns, and even customers churn. In fact, through surveys, problem tickets and other types of valuable information sources, text analysis techniques can provide better customer experience optimizing quality, effectiveness and speed in solving problems.

It is important that not only experts could understand and interpret the results of these analysis. Attention has to be but on the final aim of the analysis and representations given to the users.

1.3 The current state-of-the-art

In the literature many existing algorithm to perform the several phases of text mining can be found. Some of them are reported below.

One of the main feature of text data is that dimensionality of their representation is very large, but the underlying data is sparse, so, in order to perform a proper data dimensionality reduction several algorithms as been studied. PCA (principal component analysis) is a statistical procedure for reducing the dimension of a $n \times p$ data matrix X. It uses an orthogonal transformation to convert a set of observations of possibly correlated variables, into a few linearly uncorrelated variables called principal components. The first principal component direction of the data is that along which the observations vary the most and

they are used to project the original components into the reduced space with the associated principal values.
There is also a variant of this technique called principal components regression (PCR) which involves constructing the first M principal components, Z_1, . . ., Z_M, and then using these components as the predictors in a linear regression model that is fit using least squares.
The main reduction algorithm used in literature is the Singular Value Decomposition (SVD) also known as LSA (Latent Semantic analysis) when it is applied to the document-term matrix in the textual contest. LSA allows reducing the dimensionality of the document-term matrix while disregarding some irrelevant dimensions. It maps words and documents into a concept-space where comparison between terms is done. In this transformed space some insignificant dimensions can be easily identified through the matrix factorization technique which allows to express the relative importance of the dimensions. Dimensions expressed by low magnitude of singular values can be disregarded while the ones with higher values are maintained and represent the hidden concepts.

The difference between SVD and PCA is that they use a different coordinate system to map the position of the documents.
Neural network are also exploited as new ways to reduce the dimension of large data sets. Autoencoder is a type of artificial neural network used to learn efficient data patterns in an unsupervised manner. It includes an encoder and decoder and AIMS to compress the information of the input variables into a reduced dimensional space and then recreate the input data set.

For what concern cluster analysis (clustering) various algorithms exist and vary a lot basing on the cluster model employed and the relative notion of a cluster. Either way, the final common goal is grouping data objects into well separated clusters (groups) so that objects within each group are more similar to each other, while objects in different groups are more different from each other. Similarity and differences are measured by some distances.
K-Means is one of the most well known unsupervised learning algorithm. It is a simple

partitional strategy which seek to partition the data into a pre-specified number of clusters represented by their centroids (mean value of the objects in each cluster). Initially documents are assigned to the cluster whose centroid, randomly chosen at the begin, is the nearest to that document, then the mean of the documents in each cluster is computed to recalculate the new centroids. The process stops when the centroids do not change.
An alternative to this approach which does not require the number of clusters to be previously set is Hierarchical clustering. It can be a bottom up or top down process which end up with a tree-like visual representation of the data objects, called Dendogram, that allows to view at once the clusterings obtained for each possible number of clusters, from 1 to the number of data objects. In order to identify clusters on the basis of the dendrogram, a horizontal cut at a certain height of the dendogram is done.

Differently from the approaches explained above there exist algorithms based on statistical methods which analyse text and words and attempts to find the topics the documents talk about and the possible related documents. These techniques represent documents as probability of words, the two main models are Probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet allocation (LDA).
pLSA, also known as pLSI (Probabilistic Latent Semantic indexing),is based on a statistical latent class model where documents, words and (hidden) topics are variables linked together; more specifically, topics are associated with the observed pairs (document, term). The final aim of this approach is to explain documents as a mixture of topics which arise from a co-occurrence matrix. The idea behind latent semantic analysis is to derive low-dimensional representation of the observed variables in terms of their affinity to certain hidden variables. In pLSI, each document is represented as a list of numbers (the mixing proportions for topics), and there is no generative probabilistic model for these numbers. This leads to problem of over-fitting and lack of clarity. To go beyond this issue it is necessary to consider the assumption of ex-changeability for the words in a document that is a precondition for latent Dirichlet allocation (LDA) model.
LDA is a generative probabilist model which aims to automatically discover the topics

from a collection of textual data. The documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA uses Bayesian inference to infer the hidden structure inside the collection of documents under analysis and discover the topics. it involves three levels of representation in order to allow the association of a document to more than one topic. The two stages involved in the process, to generate words for each document, are:

1. Random choice of a distribution over topics
2. For each word in the document:
    (a) Random choice of a topic from the distribution defined at the previous step
    (b) random chioce of a word from the corresponding distribution over the dictionary.

Since no strategy is universally superior, in the proposed text mining engine, ESCAPE, two of the most papular strategies in the literature are integrated: the joint approach, an algebraic model based on SVD decomposition together with the K-Means clustering algorithm and the probabilistic method, a probabilistic model, based on the analysis of latent variables through the LDA.

2 Escape

Given an overview of the context and the guidelines for designing efficient and effective text mining algorithms it is possible to introduce the proposed algorithm ESCAPE. It has been tested only on a few datasets (seven in total), with different structures; in the current work it has been tailored on short text and low-level vocabulary richness datasets. A large number of real datasets has been included have a better understanding of the performance, identify major pitfalls and suggest some improvements. Often, it happens that models are beautiful but not suitable for data or data for testing process are not available.
The starting point has been analysing the various component of the algorithm in order to understand if it is suitable for the data under analysis. Also the data structure and type have been evaluated to make some preliminary hypothesis.

ESCAPE (Enhanced Self-tuning Characterisation of document collections After Parameter Evaluation) is an efficient and effective distributed self-tuning engine to cluster collections of textual data into correlated and well-separated groups of documents. It aims to automatically discover and properly present to the end-user interesting and latent topics hidden in a given corpus. The number of topics, which will be also defined as categories or clusters later on, are not previously known.
ESCAPE runs on Apache Sparke, which has a distributed memory and allows parallel computation, important characteristic for big data analysis algorithms.
ESCAPE includes also the computation of some statistical indices to characterise the document collection data distribution under analysis.

ESCAPE solve many of the significant issue in the text data analytic process. The main advantage of it is that it is a parameter-free solution. It has the ability to autonomously address all the steps of the analytics pipeline, properly enriched with self-tuning and self-assessment strategies. To relieve the end-user of the burden of selecting proper values for the overall process of cluster collections of textual data, automatic strategies are integrated.

This is the reason why this engine is effective: it does not need constant human supervision to mine data and retrieve information. The end users do not have to tune the parameters: finding and setting the optimal number of topics in which to cluster the documents is not their issue.
Another important characteristic is that ESCAPE supports large-scale analytics and does not require multiple algorithms, so it is not time-consuming, and has lower computational costs. To reduce the latter,in fact, distributed approaches have been exploited.

Two approaches are integrated to divide collection of textual corpora into groups of documents related to specific topics within ESCAPE: the Joint approach and the probabilistic topic modelling approach.
The joint approach consists of reducing the dimensionality of the dataset under analysis, though the application of algebraic models using an unsupervised algorithm on the weighted matrix to construct its low-rank approximation. Then an unsupervised (as well) clustering algorithm is applied.
The probability approach involves topic modelling methods, which are built on the distributional hypothesis, suggesting that similar words occur in similar contexts. The used algorithm is based on statistical methods that analyse text and discover the treated topic and the relationship between different documents.

2.1 The process

The pipeline for ESCAPE architecture is made by three main blocks which address all the building steps of KDD processes:

I. Data processing and characterisation

II. Self-Tuning Exploratory Data Analytics

III. Knowledge validation and visualisation

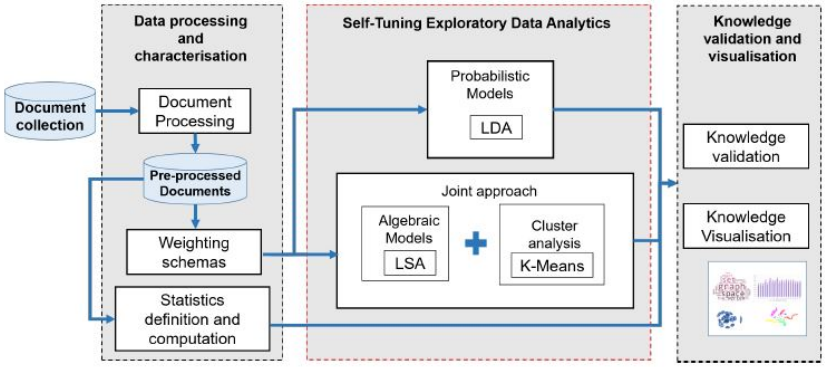


Figure 1: ESCAPE-architecture

Actually, the very first step in any data analysis process is to collect data, i.e. the set of documents of interest. Rarely, this is a quick and trivial phase because, most of the times, data are not immediately available in a data warehouses or database in the shape and "clean" structure needed for the analytic process. Moreover, it is important to pay attention on the source of the data and analyse if they are corrupted or distorted. For a trivial example, if we consider dataset of the fines in the USA it is possible that the data are not reliable because some policeman, especially in the past, tended to penalise more Afro-American people so the hypothetical results will not be valid for the entire American population.

2.1.1 Data processing and characterisation

Given the set of interest, the first block of the engine architecture, showed above, can be addressed. As in any natural language process, textual data have to be pre-elaborated. Pre-processing is a fundamental step since it affects the quality of the final results. It involves five components:

- *Document splitting.* Depending on the needs and the goal of the text mining process, documents can be split into sentences, paragraph or analysed just the way they are. Short documents (like reviews in this study case) are, usually, translated into a single vector for each message while longer documents can be analysed as the entire document or split into sections.

- *Tokenization.* This is the process of breaking up a stream of textual data into tokens (such as words) within the same sentence by the white space or punctuation marks.

- *Case Normalization.* This is a conversion of each token to upper-case or lower-case characters.

- *Stop words removal.* Common, irrelevant words such as articles and preposition are disregarded since they don't provide any useful information.

- *Stemming.* All the items are replaced by their stem and prefixes, suffixes, and pluralisation are removed. (e.g., connected, connecting, connection, . . . , become connect)

After these steps, the documents are represented in the bag of words (BOW) form, where, not the order, but only the frequency of the terms is relevant. The documents collection is then ready to be converted to matrix structure format. Usually, the smaller the dictionary, the greater the intelligence to capture the most and the best words to use in the following steps of the analysis.

To better identify topics and to help the clustering process ESCAPE includes a term relevance step: weights are assigned to each term in the corpus through several weighting function in order to highlight their degree of importance. Different weighting schemas are available since, in different scenarios, one can outperforms the others.

First, formally, some notations have to be defined:

$D = d_1, d_2, \ldots d_{(|D|)}$: collection of |D| documents, named Corpus (the textual dataset to put under analysis).

d_i : general document in the corpus.

$V = t_1, t_2, \ldots t_{|V|}$: the set of distinct terms in the textual collection, i.e. the set of all tokens used at least once in a document.

t_j: general token (term) in a document.

After the already mentioned pre-processing steps, D can be represented as a Document-term Matrix (X) where a 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.
Each cell in the matrix X is then associated to a weight w_{ij} to measure the relevance of term j appearing in the document i. The weight is computed as the product between a local (l_{ij}) and a global (g_{ij}) term weight. The local weight measures the relative frequency of a certain term in a particular document, while the global weight describes the relative frequency of the specific term within the whole corpus. Three local and three global term weights are included in ESCAPE. The local ones are:

- $TF = tf_{ij}$: Term Frequency weight which represents frequency of term j in document i;
- LogTF=$log_2(tf_{ij} + 1)$: Logarithmic Term Frequency weight which is used to diminish the large number frequencies;
- Boolean 0,1: Binary which is a weight function equal to 1 if the frequency was non-zero (i.e. if a term appeared at least once in the document) and 0, otherwise.

The first two functions give more relevance to more frequent words while the latter is

sensitive only if a words is in the documents or not.
Global weighting functions are useful in order to give less importance to the words that appear more frequently or in many documents and the proposed ones are:

- Global term frequency ($TF_{glob} = tf_j$)
- Inverse Document Frequency ($IDF = log(\frac{|D|}{df_j})$) where df_j represents the number of documents where the term appears, it defines the rareness of a term;
- Entropy ($1 + \sum_i \frac{p_{ij} * log(p_{ij}))}{log(n)}$, where $p_{ij} = \frac{1}{ndocs}$) which assigns minimum weight to terms that are equally distributed over documents and maximum weight to terms which are concentrated in a few documents.

Several schemas are involved in the analysis since different combinations are able to characterize datasets at different level of granularity.
The used combinations are TF-IDF (with this schema the value of the weight is high when a term appear frequently in a certain document but rarely in the whole collection) , LogTF-IDF (this schema penalizes frequent words more than the previous one), TF-Entropy, LogTF-Entropy, Binary-IDF, Binary-Entropy, Binary-TFglob.

For what concerns data characterization,in the beginning step, some statistical indices are computed within ESCAPE in order to give more insight about the lexical richness and other features of the input textual collection.
The proposed indices are the following:

- Categories number: number of topics in the collection, if a-priori known.
- N Doc: number of documents in the corpus;
- Max/Min/Avg freq: maximum, minimum and average frequency of a term's occurrence.
- Terms number: number of terms in the collection with repetitions.

- Dictionary: number of terms in the collection without repetition.
- TTR (Type-Token Ratio): the ratio between the dictionary variety (Dictionary) and the total number of tokens in the collection (term). It represents the data sparsity: a high value corresponds to a high degree of lessical variation within the data-set.
- Hapax rate: the ratio between the number of Hapax (absolute frequency of terms with one occurrence) and the cardinality of the dictionary. It has to highlighted that if the Boolean feature of ESCAPE, named remove-Hapax is set to TRUE then ESCAPE removes the Hapax words for subsequent analyses.
- Guiraud Index: the ratio between the cardinality of the dictionary and the square root of terms number. It highlights the lexical richness of a textual collection.

2.1.2 Self-Tuning Exploratory Data Analytics

The Self-Tuning Exploratory Data Analytics phase which involves document clustering and topic modelling is developed using two different strategies: the joint approach and the probabilistic model.

2.1.3 Joint approach

For what concern the Joint Approach, the first step is applying a dimension reduction algorithm on the previously built weighted matrix X using Latent Semantic Analysis (LSA) based on the Singular Value Decomposition (SVD), then the partitional K-Means algorithm is applied in order to perform the clustering phase.
This step aims to find hidden concepts and to delete some irrelevant dimensions without losing significant information. LSA allows to analyse relationships between group of documents and terms, mapping them in a concept space where a proper comparison is done. In this way it is possible to consider the meaning which lies behind the words and do not compare them just the way they are. The main difficulty which arise to find relevant

documents from search words is then overcome.

To streamline this process, documents are seen as a Bag Of Words where only the frequency, and not the order in which terms appear, is relevant while concepts are represented as a pattern of words that appear together in the collection.

SVD, the matrix factorisation method, takes as input, the weighted document-term matrix X, and decomposes it in the product of three matrices.

$$X = USV^t$$

where:

U(d,r)=document-concept similarity matrix.

S(d,d)=concept matrix.

V(r,t)=term-concept similarity matrix.

U and V are column-orthonormal matrix, while V is a diagonal matrix.

Each term in a specific document can then be seen as a linear combination of the term-concepts and the document-concept weights, and the significance of each dimension in the document collection is defined by the magnitude of the correspondent singular values in S. Values with low magnitude can be interpreted as noise in the data and can be disregarded. Consequently, the k relevant dimensions (corresponding to the largest singular values in S) can be identified and the dimension of each matrix can be, accordingly, reduced:

$$X_{K_{LSA}} = U_{K_{LSA}} S_{K_{LSA}} V^t_{K_{LSA}}$$

This is an optimal approximation of the original matrix.

In the common LSA process, the Frobenius norm is computed to select the k-rank matrices, among all the available ones, which will be analysed in the following steps.Only the ones for which the norm is minimised are retained.[10]

The innovative algorithm included in ESCAPE to automatically determine the main relevant dimensions (k_{lsa}) and perform dimensionality reduction of the matrix X is called ST-DARE. ESCAPE uses only the largest singular values of the given K_{LSA} in the matrix S and sets

the remaining ones to zero. Three good values for the number of dimensions (Term) are automatically identified without losing significant information. The approach of choosing only the maximum decreasing point of the singular value curve is avoided since it can lead to meet a local optimum which is not the correct solution.[13] For this reason ST-DARE is, instead, integrated in ESCAPE in an enhanced version with only one parameter as input to analyse the trend of the significance of the singular values. The significance of the dimension is expressed by the the magnitude of the relative singular values, the ones with low magnitude can represent noise so they can be disregarded. Thus, only for the first T (set equal to the 20% of the rank of the document-term matrix) singular values, the relative mean and the standard deviation values are computed and a confidence interval is established. The three good values are selected along this curve, they are in the correspondence of the mean position, the mean plus the standard deviation position and the mean of the previous one positions. In this way no local optimum is met.[11] The pseudo-code is reported in the figure below.

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

```
1  N = 0;
2  // compute the SVD decomposition of the truncated matrix X;
3  [U,S,V] ← X.computeSvd(T);
4  s ← normSingularValues(S);
5  // compute the mean of singular values;
6  mean = s.mean();
7  // compute the standard deviation of singular values;
8  stand_deviation = s.std();
9  // compute the three values;
10 val1 = s[mean];
11 val2 = s[mean + stand_deviation];
12 val3 = s[(val1 + val2)/2];
13 K_LSA.push(val1, val2, val3])
```

Figure 2: Enhanced ST-DaRe pseudo-code

The second step in the joint approach is the cluster analysis performed applying the k-means algorithm. The difference between LSA and K-means is that the former aims to assign a set of topic loadings to each document while the second assign each document in the collection to a specific group (cluster). This unsupervised learning algorithm uses

a partitional strategy aims to find a proper number (K) of clusters represented by their centroids computed as the mean of the objects in the group (the reviews in this thesis). K random centroids are selected at the begin and each document is associated to the most similar centroid in order to create the initial clusters, then the mean of the documents in the cluster is computed again to identify the new centroid. The process iterates until each centroid does not change any more. The similarity between two documents is measured by the Euclidean distance (between the words within the documents) . The cosine distance would have better expressed the human perceptions but if the vectors are normalised there is a connection between the two distance measures and Euclidean distance can be properly used.

K-means requires the user to previously know about the number of clusters, to overcome this issue ESCAPE integrate a self tuning clustering algorithm to automatically identify a good number of clusters, which represent the hidden topics within the data. Different documents partitions, obtained with different K-means configurations, are compared and ranked. The clustering validity assessment uses three indicators based on silhouette definition. The silhouette index is a quality measure of how well the clustering has worked. It measures how close (similar) an objects is to the neighbours in its cluster (cohesion) compared to the objects in the other clusters (separation). Higher values represents better quality. The top 3 K-configuration with the higher index values are selected.

Two variation of the standard silhouette index are integrated within ESCAPE: starting from the purified silhouette index (PS), the weighted purified distribution of silhouette index (WS), the average silhouette index (ASI), and the global silhouette index (GSI) are computed. ASI represents the average silhouette of the entire cluster while GSI, considering the possible imbalance number of elements in the clusters, penalises more the clusters with the large number of documents. A rank function is applied, first, for each index, then globally in order to report to the users only the best solution for the datasets. The global score function is defined as follows:

$$score = ((1 - \frac{rank_{GSI}}{k_{max}}) + (1 - \frac{rank_{ASI}}{k_{max}}) + (1 - \frac{rank_{WS}}{k_{max}}))$$

An upper-bound for the number of clusters in the analysis is set to the average document length for each corpus but this choice could be changed by any analyst.

2.1.4 Probabilistic approach

To address the self tuning exploratory data analytics phase ESCAPE integrates also a probabilistic topic model approach (LDA) which is totally different from the previous one. It is a generative statistical-based technique that describes topics and words as probabilistic distributions from which document terms will be drown. The idea behind this technique is that documents are a mixture of latent topics. Topics are defined as a distribution over a fixed, previously generated, vocabulary; while documents are defined as a distribution over the set of different latent topics. LDA uses Bayesian (posterior) inference in order to infer the hidden structure and discover the topics inside the collection under analysis.
This algorithm does not require an a priori knowledge of the dataset characteristics but requires the number of topics to be previously set.
The words of each document in the collection are generated in two steps: first, a distribution over topics is randomly chosen, then, for each word in the document,a topic is randomly chosen from the distribution defined at the previous step and a word is randomly chosen from the corresponding distribution over dictionary. The result is that the same set of topics is shared between all the documents but the proportions in which the topics appear within them is different.
More in detail, the two stages needed to generate a document in the corpus are: selection of the number of terms from a Poisson distribution and, for each document's word, selection of a topic and a word from a multinomial distribution where the parameters represent, respectively, the document-topic distribution and the topic-words distribution conditioned on the topic. As already said, the per-document distribution is drown using Dirchelet distribution ($p(D|\alpha, \beta)$) but, unfortunately, it is infeasible to compute it so ESCAPE exploits an on-line variational bayes algorithm and sets α, β to maximize the log likelihood

of data under analysis. α indicates the concentration for the prior placed on documents' distribution over topics (higher α value for documents with a few topics), β describes the concentration for the prior placed on topics distribution over terms. (lower β values for topics described by a few words). [9]

A novel iterative approach, called TOPIC SIMILARITY, is proposed within ESCAPE in order to identify a proper number of topics (k), fundamental for optimizing the clustering process results. Several LDA models with different k values, between a pre-specified lower and upper bound (K_{min}, K_{max}), are computed, then they are evaluated basing on the topic content and quality metrics in order to identify the best configuration. This strategy asses,in three steps for each LDA model, how topics are semantically diverse:[12]

the first step is the Topic characterization through its n most representative words. Basing on the TTR, only the richest part of the corpus is considered, then the remaining words are sampled by the average frequencies of the terms and this total quatity of considered term is denominated Q. Then, n is automatically set equal to $\frac{Q}{K}$ if the final number of considered words is major than the average term frequency of the corpus terms otherwise it is set equal to the average frequency of terms in the corpus. In this way all the topics are represented at least by a number of words equal to the average frequency. At the end, once repetitions of terms describing a topics are removed, if a word appears in a topic, the correspondent value represents the probability that the term has to be picked up in the topic.

The second step is the similarity computation between all the possible pair of topics within the same K partitioning using cosine similarity, really efficient measure to reflect human perception of similarity. Each value becomes a cell value of the K x K symmetric matrix representing the similarity between the topic in the row and the topic in the column, then the Frobenius norm of the whole matrix is computed and divided by K. The ToPIC-similarity index is obtained multiplying these values by 100.

These first two steps are repeated for all of the topics included in every K LDA model. The third and final step involves the identification of three K values for a good clustering configuration that satisfy two conditions. Considering the topic-similarity function obtained from the previous steps and the fact that, empirically, this curve is decreasing but not

always monotonic, the conditions are set to reach also a good trade-off between optimal results and computational costs:

- K has to be a local minima (i.e. topic similarity(k_i) < topic similarity(k_{i+1}))).
- K has to be the only point belonging to a decreasing segment of the curve (i.e points that have a positive second derivate).

The search stops when the first three values are found or when a K upper bound set by the analyst is reached.

2.1.5 Knowledge validation and visualization

The Knowledge validation and visualization phase is addressed in a quantitative and qualitative way, using different kind of representation and explanation to be as more intelligible as possible so that users with higher and lower level of knowledge can read and understand the results. The extracted information is provided at different levels of detail to allow high level overviews but also to find out domain specific information.[13]
The quantitative technique for the Joint approach includes the already mentioned silhouette based indices which measure the cohesion and the separation of each different cluster set. High values means that an element in well matched with the other elements in the cluster and poorly related with the other clusters. The different computation gives relevance to different aspects of the data structure: the PSI (purified silhouette index) disregards documents which appear in a singleton cluster (ESCAPE plots this ordered distribution to make a comparison of the different partition of the same cluster); the WSI (weighted silhouette index) represents the percentage of documents in each bin weighted with an integer value and normalized within the sum of all weights; the ASI (average silhouette index) gives an overview of the silhouette of the total cluster set and the GSI (global silhouette index) takes into account possible in balance of the number of documents in each cluster. These values are used to identify the best three clustering configuration ranking them from the higher to the lower values and are plotted in comparison with a benchmark,

a final rank function is computed to provide the best solution for the experimental sets. For what concern the probabilistic model, a quality measure to describe how well the model predicts a sample, the Perplexity, is computed. It is a monotonic decreasing function in the likelihood of the data so the lower the value the better the model performance and the probability estimate of the corpus. Also the Log-likelihood , which describes how "likely" things are, is computed and reported. The only real interpretation for log-likelihood is, "higher is better", but if it is taken into account to evaluate only one model for representing the data, value is absolutely meaningless. The differences in the log-likelihood when different model are compared are instead relevant.

Visualization techniques are the key to gain an immediate, clear and better insight into the data. The proposed techniques aim to show interesting correlation among the data at different level of granularities.[14]

t-SNE (distributed stochastic Neighbour Embedding) reduces the representation of high dimensional data into a two or three dimensional map without loosing significance. The similarity between two data points is computed converting the euclidean distance into conditional probability. Unlike the SNE, the t-SNE minimises the sum of differences in conditional probabilities with a symmetric kind of the SNE cost function, with simple gradients. Close points will have higher values while for far points the value will be almost zero. The probability is computed again in the reduced space in order to print high-dimensional data and, in the meanwhile, visualize both the original structure and the relationship between data by exploiting points colouring which reflects the assignment to a specific topic. This representation provides a better results then the, usually used, linear representation, when working with curved manifolds. It performs in fact different transformations on different regions but they can be misleading. t-SNE often fails to preserve the global geometry of the data, this means that the relative position of clusters on the t-SNE plot is almost arbitrary and depends on random initialisation more than on anything else. Attention has to be put on perplexity and iterations parameters in order to give a proper interpretation of the

results and avoid misleading. Perplexity values allows to balance the relevance of global and local aspects of the data under analysis giving a rough hypothesis about the number of close neighbours each point has. Common range is between 5 and 50 but many different values should be analysed to gain an optimal representation; a rule of thumb says that a good value corresponds to the 1% of the sample size as a large perplexity for any given data set. Also for successive runs, t-SNE does not produce the same output so different results for different number of iterations should be investigated. It has to be highlighted that t-SNE naturally expands dense clusters, and contracts sparse ones, evening out cluster sizes so they are not meaningful.

Word Clouds are one of the most immediate representation and allow to directly observe if the results of the clustering phase are good. This method uses informative images which gives the perception of the most representative words (selecting a maximum number) of a topic. The clouds, essentially, represent the topic term distribution, and the term with higher probability are emphasised with a larger fontsize. The words used are the ones which describe each cluster content obtained by clustering and topic modelling.

Graph representation exploits the widespread graph structure which, in the unstructured domains as this case, displays only the most relevant process. An undirect graph G(v,e) with V nodes and e unordered edges models the topic-term distribution. Topic and term are nodes and the edges are the link between them if the probability of a term belonging to a topic is major of a certain treeshold. If a term apperas in more topics the relative node is colored in red. An analysis of the results can be provided computing the connectivity of the graph and indentifying if a topic is characterized by words not used in other ones (the topic will be disconnected by the others).

Word Tables are, basically, list of the words, which describe the topic content, in descending order of probability. These tables take into account the arguments of cohesion and coherence through their content in order to evaluate the clustering process and help the analysis of the

previously mentioned graph. A threshold can be set to represent only the salient and most frequent words. A possible extension may association rules extraction to detect interesting correlation between words.

The last used representation method is the Correlation matrix map. This tool shows and allows to analyse possible correlation between topics using five different coloured correlation ranges. The dot product between all the documents, sorted by topic, is computed and basing on this value the cell is coloured. The correlation squared matrix with values between -1 and 1 can be also converted into an adjacency matrix to allow a graph representation useful when working with high dimensional data.

It has to be pointed out that ESCAPE includes also a FP-Growth algorithm to detect when a set of words, called itemset, appears in several documents and therefore it can be considered frequent. Interesting association rules can be derived from these itemsets. It is involved in the process of the creation of the word clouds in order to detect and display the most relevant and distinctive terms for a certain topic.
An adjusted version of the Rand index is then used to make a comparison between the two approaches used in the exploratory data analytics phase (LDA, lsa) but also between different weighting schemas results within an approach. The general Rand Index between two random partition has not a constant expected value for random clusters agreement while this is equal to 0 for the Adjusted index. It has a maximum value equal to one which means a total agreement between two partitions.

3 Amazon reviews case

3.1 Amazon Web Service and the S3 bucket

Since 1995, time of the first review, Amazon, the e-commerce giant, collected over a hundred million customer reviews where people express their opinions and describe their experiences regarding products on the Amazon.com website.

Interpretation of customer needs, behaviour and preferences are the basis to ensure a conscious and efficient decisions making process to make companies move promptly in the right strategic direction. This is the best way to gain users satisfaction which is fundamental for Amazon. The first of its four guiding principles is, in fact, "customer obsession rather than competitor focus" and in the 2018 letters to shareholders, Jeffrey P. Bezos (Amazon founder and CEO) highlighted again a lot the importance of listening to customers, saying: "The biggest needle movers will be things that customers don't know to ask for" [15]

In this situation, analysis of products reviews text can bring out useful and important implicit knowledge to exploit. Moreover, products reviews are a huge source of information not only for Amazon itself but also for academic researchers and other companies, which could feel constrained by their limited commercial database options.

A clear example of constraints can be found in all the companies which do not have direct channels of distribution and,consequentially, they do not have the opportunity to collect massive amounts of customer data essential for driving sales, understanding clients actual needs and delivering personalized experiences.

As a confirmation of the high level of utility and benefit which can derive from the knowledge hidden inside the products comments, Amazon built, within the Amazon web service platform (AWS), an Amazon Custom Reviews Dataset where over 130 million customer reviews, from 1995 to 2015, are collected and available.

Amazon Web Service is essentially the world's most comprehensive and broadly adopted cloud platform. It offers more than one hundred services through a powerful infrastructure with high level of reliability and scalability and lower costs. Hundreds of thousands of

businesses in 190 countries around the world, including many of the largest enterprises, leading government agencies and start-ups are the customers and are powered by AWS every day.

The common reasons of the widespread adoption of the cloud platform are in part linked to the operational and maintenance costs. Components cost of the infrastructure, flexibility costs (i.e. the possibility to choose between several solutions) and updating and day-by-day maintenance costs are totally translated to AWS allowing also major levels of the employee productivity by removing wasting time. The other reasons are more related to risk management: AWS allows companies to improve operational resilience and their ability of reaction to economic and environmental change.

Essentially, using AWS, businesses can take advantage of Amazon's expertise and economies of scale to access resources when their business needs them, delivering results faster and at a lower cost.

Specifically,in the amazon-reviews-pds S3 bucket in AWS US East Region, Amazon Custom Reviews Dataset are available. Amazon S3 is object storage built to store and retrieve any amount of data from anywhere on the Internet. It provides a web service interface that customers can use to store and retrieve any amount of data, at any time, from anywhere on the web. Amazon Custom Reviews Dataset has been constructed to represent a sample of customer evaluations and opinions, variation in the perception of a product across geographical regions, and promotional intent or bias in reviews. There are, actually, three parts within the database.

For this reason, as well as the collection of more than 130 Million US customer reviews mainly addressed to facilitate study into the properties (and the evolution) of customer reviews, there are other two main components.

The second component is a collection of more than 200.000 reviews about products in multiple languages from different Amazon marketplaces (across five different countries), intended to facilitate analysis of customers' perception of the same products and wider consumer preferences across languages and countries.

The third one is a collection of several thousand reviews that have been identified as

non-compliant with respect to Amazon policies. This is intended to provide a reference dataset for research on detecting promotional or biased reviews. This part of the dataset is distributed separately and is available upon request.

A registry of open data also exists on Amazon S3. It allows "common" people to discover and share datasets that are available via AWS resources. In this case, datasets are provided and maintained not by AWS, but by a variety of third parties (government organizations, researchers, businesses, and individuals) under a variety of licenses. People and researchers can, then, share and exploit them for several usage and fields of application: from sustainability to transport, weather, chemistry, food security and many others.

This kind of data perfectly embodies the "5V" characteristics of the Big Data[16]
Volume the size of this collection of data is huge, probably the biggest in the e-commerce field and too large to be managed with traditional approach.
Velocity: products sold an Amazon and consequently reviews accumulation increase with an incredible rate even more every day.
Variety Text Data such as reviews are unstructured, unorganized and cannot be stored in the form of rows and columns so they are the most complex type to be analysed.
Variability reviews could be not always available for all products and the information within them is highly variable since you can find every type of products and the related review, moreover, potentially each person can write a review so you can find any kind of language, dialect, way of speaking.
Value row data such as reviews themselves are useless, they have to be transformed in actionable knowledge.

Big issues when working with Big Data are the scalability to huge data volumes,the data storage space and the needed computational power. To deal with them a big amount of financial, human and technical resources are necessary. Distributed file systems, computing clusters, cloud computing, and data stores supporting data variety and agility are necessary

to provide the infrastructure for processing of big data. The choice of AWS as study case is also related to the fact that in this context Amazon proposes itself also as a solution for companies, researches and any kind of organization.
Moreover, The recent news regrading AWS is worthy of attention: on the 15th October 2019 Jeff Bezos announced that 75 petabytes of internal data stored in nearly 7,500 Oracle databases have been migrated to multiple AWS database services. After several years of work, with the collaboration of more then 100 teams in Amazon's Consumer business, the database migration effort is now complete and Amazon Customer business is totally independent from Oracle. Thanks to this migration, many benefits, both for Amazon itself and the AWS users have been obtained in terms of: Cost Reduction (database costs are reduced by over 60% on top of the heavily discounted rate Amazon negotiated based on its scale. Customers regularly report cost savings of 90% by switching from Oracle to AWS), Performance Improvements (Latency of Amazon consumer-facing applications was reduced by 40%). and Administrative Overhead (The switch to managed services reduced database administration overhead by 70%). [17]

Data Mining plays a key role to extract and exploit information from collections of data like the ones we are talking about since it provides an essential support. More specifically, text data analysis can help analytics businesses by extracting insights from free textual data written by (or about) customers, combining it with feedback data (if available), and identifying patterns and trends.
In order to extract potential high quality information ESCAPE is applied on reviews made on different categories of Amazon products. Running ESCAPE on these several datasets will allow an additional test of its potentiality in terms of effectiveness and efficiency and the detection of pitfalls or potential improvements. It will be possible to asses the differences between the approaches, which of the two will perform better and why and the impact of the different weighting schemas. Also the number of topic detected and the partitioning will be evaluated along with the different visualization technique to identify the best one

according to specific needs.

3.2 Data description

As said at the begin, it is really important to have knowledge about the source structure of the data of interest and potential corruption or distortion in order to allow a full and clear interpretation of the final results. For this reason a detailed explanation about the data origin and structure is provided.
Data are retrieved from the Amazon Customer Reviews Database, mentioned above, in TSV format files. Reviews have been collected between 1995 and 2015; each line in the data files corresponds to an individual review (tab delimited, with no quote and escape characters) and metadata are also included.
In each row, except the first one which contains the header, in addition to the review text (labelled as review_body), there are the following information:

- Marketplace: country code of the marketplace where the review was written
- customer_id: Random identifier that can be used to aggregate reviews written by a single author.
- review_id: The unique ID of the review.
- product_id: The unique Product ID the review pertains to. In the multilingual dataset the reviews for the same product in different countries can be grouped by the same product_id.
- product_parent: Random identifier that can be used to aggregate reviews for the same product.
- product_title: Title of the product.
- product_category: Broad product category that can be used to group reviews .

- star_rating: The 1-5 star rating of the review.
- helpful_votes: Number of helpful votes.
- total_votes: Number of total votes the review received.
- Vine: Review was written as part of the Vine program.
- verified_purchase: The review is on a verified purchase.
- review_headline: The title of the review.
- review_body
- review_date: The date the review was written.

It has to be highlighted that an user has to satisfy some eligibility requirements before being allowed to publish a product review or to contribute to other customer features such as customer answers or idea lists. He/She does not have to buy the product to write a review about but he/she must have spent at least 50 dollars on Amazon.com using a valid credit or debit card in the past 12 months.
These requirements are not valid for reading content posted by other contributors or post Customer Questions, or for creating or modifying Profile pages, Shopping Lists, Wish Lists or Registries.
Not all the reviews submitted by the users are automatically published and there is always a waiting time in which Amazon has a check. There are, in fact, some guidelines about the review content to ensure (as more as possible) helpful, relevant content to customers based on their own honest opinions and experience with respect for others. If a review does not comply with Amazon guidelines, it can be removed or rejected and the reviewer may not resubmit a review on the same product.
These constraints lower a bit the possibility of corruption and distortion within the data but some issue remain opened: reviews can, for example, contain URL (only linked to other Amazon products) and this make data even more complex to mine.

3.3 Data collection and preparation

The first step in any data mining process is data collection; for this analysis, data of interest are retrieved from the first main component of the Amazon Products Reviews datasets. Not all, but different-sizes reviews datasets regarding different products category have been downloaded and used to experimentally find previously unknown and potential useful information applying ESCAPE engine. Topic detection and clustering analysis are the main gaols.

The involved products categories, which will be the database for the analysis, are the following:

1. D1: Automotive
2. D2: Camera
3. D3: Digital music
4. D4: Digital Software
5. D5: Mobile Electronics
6. D6: Furniture
7. D7: Gift Card
8. D8: Luggage
9. D9: Major Appliances
10. D10: Office Products
11. D11: Personal Care
12. D12: Pet Products
13. D13: Shoes

14. D14: Video Games

15. D15: Video

A necessary data collection process has been carried out to obtain the document Corpora for the analysis.
To download the data from the Amazon platform the AWS Command Line Interface, has been downloaded from the Amazon website and installed; all the available data in the bucket has been listed per product category and then, the chosen ones have been downloaded. The basic idea for selecting the categories to download is having a heterogeneous and whole database representing more or less all the products which can be bought on Amazon.
From the command prompt the 'ls' command has been used to list all the data in the bucket:
'aws s3 ls s3://amazon-reviews-pds/tsv/'
Then, to copy the files in the own local directory, the 'cp' command has been used for each chosen category product reviews file (dataset):
'aws s3 cp s3://amazon-reviews-pds/tsv/amazon_reviews_us_nameOfChosenCategory_v1_00.tsv.gz'
Each of the downloaded TSV file have been processed using Python to extract only the Review column which has been then saved as text files in order to run ESCAPE on it.
The reviews have been collected between 1995 and 2015 and the most recent ones are, for all the dataset, dated 25/08/2015. The language used should be the American english since the "Marketplace" field in the original files specifies that the review were written in USA but different dialect are probably used.
Technically, each category of product represents a Corpus which is made by the collection of products reviews that are the Documents. These Documents (reviews)are, obviously made of Words named Tokens. All the documents in the collections are short text data.

3.4 Analysis and Results

The first experiments have been run on all the datasets, i.e. all the product categories reviews, applying the Joint approach with the different weighting schemas of local and global weights: Boolean-IDF, LogTF-IDF, TF-IDF. From the begin it has been clear that the probabilist topic modelling approach was not performing well on all the datasets, so the data characterization metrics has been taken into account to select the Corpus on which apply the probabilistic approach. However, statistics and indices to give an overview and characterize the datasets have been computed on each category product reviews dataset. They are showed in the table below.

	N-Doc	Dict.	fMax	fMin	TotWords	AvL	AvF	TTR	Gir.	%H	MaxVar
D1	3182095	60999	798558	2	47903747	15,05	785	0,001	8,81	0	159422921284
D0	1661190	41529	619386	2	36912478	22,22	888	0,001	6,84	0	95909134864
D2	1317819	42952	784215	2	29950860	22,73	697	0,001	7,85	0	153747507342
D3	349933	28300	129584	2	3386835	9,68	119	0,008	15,37	0	4197873681
D4	96226	11066	27660	2	2282029	23,72	206	0,005	7,32	0	191241241
D6	755040	27443	338851	2	14711552	19,48	536	0,002	7,15	0	28704661200
D7	121801	7953	105359	2	1253868	10,29	157	0,006	7,10	0	2775024362
D8	325588	17999	112280	2	5946360	18,26	330	0,003	7,38	0	3151587321
D9	91955	12240	35742	2	2812857	30,58	229	0,004	7,29	0	319336900
D5	100407	13386	37062	2	2222062	22,13	165	0,006	8,98	0	343360900
D10	2429464	57579	750511	2	45922558	18,90	797	0,001	8,49	0	140815939770
D11	82293	13877	25220	2	1807173	21,96	130	0,007	10,32	0	158986881
D12	2493661	55547	752228	2	47117462	18,89	848	0,001	8,09	0	141460988769
D13	755455	19835	290709	2	8392679	11,11	423	0,002	6,85	0	21127639962
D14	409551	24510	287780	2	6828539	16,67	278	0,006	9,38	0	20704044321
D15	362051	77017	295164	2	16957563	46,84	220	0,004	18,70	0	21780151561

Table 1: statistical characterization of all the datasets under analysis

The first thing that can be noticed is that all the collections are characterized by a large number of documents but the dimension varies a lot from the biggest one (Automotive), made of more then three millions of documents to the smallest one (Personal Care) made of about 80000 documents.[?] Such difference may reflect the fact that some categories of products are relative older and are available on Amazon from more time in comparison with

other or some products are more popular so people have given many opinions about them: surely office products are "older" and more used by common people then Digital software, so it's natural to have a bigger database with reviews about the first kind of products. [?]
Since the data are in the type of short text the lexical richness for all the dataset is low as shown by the Giraud index values (Recall that it is computed as the ratio between the cardinality of the dictionary and the square root of the number of tokens). The only exception is D15 (Video) which shows a higher value of the Giraud index (about 17) but it is still low in comparison with the average Giraud index value (54.89) of the Pubmed collections analysed in the doctoral thesis. This result makes sense and is predictable since the medical documents include many technical terms and use a high level language while reviews, most of the times, are not written by experts using a small vocabulary.
Moreover, between the dictionary values (number of terms in the whole corpus with repetition) and the number of terms (with repetitions) there is a big distance. This difference suggests that there are many words that are repeated several times within each corpus. The average frequency is indeed very high, always bigger than 100, also if stop-words, which are the most frequent ones, should have been removed during the pre-processing phase. There are also some extremely high values like 848 or 797 for, respectively, the Personal Care or Office products dataset.
The lexical complexity is also expressed by the TTR index (computed as the ratio between dictionary and the total number of token in the collection) which describes the data sparsity. In these fifteen datsets the index falls into the range [0.001,0.008]; collections of documents with lower values, as Pet products or Automotive, have not a high degree of lexical variation so they are denser then Corpus where TTR assumes higher values: digital music reviews, for example, consituite a very sparse dataset. In contrast with the giraud index which mostly assumes a common behavior among the datasets, the TTR varies much more suggesting that the lexical variation is not constant among the different databases.
All the Hapax rate is null for each dataset, this because the relative parameter has been set to zero. The experiments made during the doctoral thesis, in fact, have showed that there is no big difference between the results where words which appear only one time in

the collection of documents (i.e. Hapax) are removed and the ones where they are retained. Actually, the analysis performs better in the first case. Another consequence of this type of setting is that the minimum frequency is equal to 2 for each dataset.

Basing on the type of data under analysis, the joint approach has been applied first, before the probabilistic one, since it was predictable that it would have perform better. Reviews are indeed short texts and the used vocabulary is quite poor, so it is hard or even unfeasible to infer a hidden structure and identify a probability distribution within the datasets using the probabilistic approach. As reported in the tab 1, the average length of the reviews is quite low for all the datasets; the longest reviews (around forty words, on average) belong to the Video category but they are still short and the majority of the datasets has reviews with an average length between fifteen and twenty five words. Also, the Giraud index, which represents the richness of the used language, has very low values.
For this reason the probabilistic topic modelling approach has been applied only on the Video and Major Appliances datasets where documents have the highest average lenght. Despite this, the experiments took a long time to be exectued.
The global weight Entropy has not been used because of the bad performnces obtained when appling the probabilistic approach. For all the experiments that have been tested in the doctoral thesis, in fact, the cardinality of the resulting clusters was always unbalanced: there was one cluster with the majority of the data (80/90%).

3.4.1 Joint Approach

Before running the experiments the two parameters required by ESCAPE for the Joint Approach has been set.

- The T value which represents the number of the first singular values considered during the data reduction phase is set equal to 20'%' of the number of documents. This reduction parameter analyses the trend of the significance of the singular value and

suggests that dimensions represented by low magnitude singular values may represent noise and can be disregarded for the analysis. The ones considered for the following steps will be only the first T which, in the ESCAPE framework, is equal to 20 '%' of the rank of the document-term Matrix.

- The upper bound for the number of clusters used in the self tuning algorithm for the detection of the best configuration is set to the average document lenght for each corpus. If the average lenght is gratehr then the number of documents in the corpus under analysis the values is st to average frequency of the term. It has to be pinted out hat this is absolutely not this thesis case.

Both of these choices can be changed by the analyst.

3.4.2 Solutions

Using the different weighting schemas (Boolean-IDF, TF-IDF, LogTF-IDF), ESCAPE has been accordingly run. The obtained results are in the tables below.
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; similar values for the weighted silhouette can be obtained diving the index by the conversion factor (0.1818) .This latter index is computed as the ratio between the sum of the percentage of documents in each positive bin weighted with an integer value which has higher value when assigned to the first bin and then decreases, and the overall sum of the weights. Thus, the index weights more silhouette values in the top bin which in this case is not a very good partition, as the low value suggests. However, silhouette values are always grater then 0 so there are no wrong partitions.
It is possible to identify a trend for all the experimental results among all the dataset and the applied weighting schemas: the number of the identified clusters increases, or, in some rare case, remains the same by increasing the number of dimensions selected through LSA.

These differences mean that ESCAPE is able to analyse textual data at lower and higher levels of granularities.

The TF-IDF find, 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. Therefore this schema is useful for an analysis at high level of detail.

It has to be pointed out that, basing on the type of data under analysis, the most appropriate local weight is the Boolean, so the related results are the most reasonable. However, they are not so different in comparison with the ones obtained using LogTF or TF. The reason lies behind the fact that reviews are short text, so meaningful words appear a few times within a document and there are not very frequent terms which are penalized by the logarithm; moreover it can happen that a relevant word appears only one in a review so the local term frequency can have just two values (if it appears in the review or not) like the boolean.

Table 2: Experimental results for all the datasets for the Joint Approach

	Weights	K-Lsa	K-Cl	GSI	ASI	W-SIHL
D9	Bool-IDF	**3**	**3**	**0,422**	**0,430**	**0,042**
		7	6	0,215	0,218	0,023
		20	16	0,118	0,119	0,014
	LogTF-IDF	**5**	**5**	**0,316**	**0,311**	**0,031**
		9	8	0,228	0,217	0,023
		22	17	0,144	0,141	0,016
	TF-IDF	**6**	**6**	**0,334**	**0,317**	**0,031**
		13	14	0,222	0,196	0,021
		27	18	0,209	0,187	0,020
D2	Bool-IDF	**4**	**4**	**0,297**	**0,299**	**0,002**
		7	6	0,213	0,206	0,001
	LogTF-IDF	**4**	**4**	**0,324**	**0,325**	**0,002**
		8	8	0,225	0,228	0,002
	TF-IDF	6	6	0,309	0,283	0,002
D6	Bool-IDF	**3**	**3**	**0,381**	**0,398**	**0,004**
		7	7	0,210	0,201	0,003
	LogTF-IDF	**4**	**4**	**0,401**	**0,408**	**0,005**
		8	8	0,245	0,226	0,003
	TF-IDF	**5**	**4**	**0,413**	**0,371**	**0,004**
		10	6	0,303	0,255	0,003
D7	Bool-IDF	**5**	**5**	**0,264**	**0,267**	**0,021**
		13	14	0,167	0,166	0,014
		37	20	0,116	0,104	0,010
	LogTF-IDF	**6**	**6**	**0,236**	**0,237**	**0,018**
		14	12	0,167	0,170	0,014
		36	18	0,121	0,108	0,010
	TF-IDF	**5**	**5**	**0,296**	**0,296**	**0,022**
		12	11	0,168	0,171	0,014
		32	19	0,131	0,129	0,012
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
D5	Bool-IDF	**4**	**4**	**0,315**	**0,317**	**0,029**
		8	5	0,172	0,169	0,018
		25	14	0,095	0,089	0,011
	LogTF-IDF	**5**	**5**3	**0,309**	**0,314**	**0,029**
		11	7	0,192	0,178	0,018
		25	20	0,121	0,116	0,013
	TF-IDF	**5**	**5**	**0,296**	**0,296**	**0,022**
		12	11	0,168	0,171	0,014
		32	19	0,131	0,129	0,012

	Weights	K-Lsa	K-Cl	GSI	ASI	W-Sihl
D3	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
D11	Bool-IDF	**3**	**3**	**0,374**	**0,381**	**0,041**
		7	7	0,175	0,171	0,021
		27	16	0,097	0,092	0,014
	LogTF-IDF	**3**	**3**	**0,396**	**0,411**	**0,045**
		9	8	0,225	0,196	0,024
		26	19	0,125	0,105	0,015
	TF-IDF	7	4	0,308	0,280	0,032
		16	11	0,247	0,177	0,022
		30	19	0,172	0,137	0,019
D13	Bool-IDF	**5**	**5**	**0,260**	**0,256**	**0,003**
		10	9	0,207	0,204	0,003
	LogTF-IDF	**5**	**5**	**0,289**	**0,286**	**0,004**
		11	11	0,206	0,214	0,003
	TF-IDF	**7**	**7**	**0,296**	**0,277**	**0,003**
		13	13	0,210	0,215	0,003
D4	Bool-IDF	**4**	**4**	**0,328**	**0,325**	**0,031**
		8	6	0,207	0,205	0,021
		21	18	0,106	0,107	0,013
	LogTF-IDF	**4**	**4**	**0,400**	**0,392**	**0,036**
		9	6	0,198	0,194	0,020
		25	17	0,132	0,127	0,014
	TF-IDF	**9**	**3**	**0,338**	**0,275**	**0,028**
		14	6	0,223	0,216	0,022
		29	17	0,178	0,166	0,018
D15	Bool-IDF	**4**	**3**	**0,314**	**0,316**	**0,008**
		7	4	0,197	0,190	0,005
		18	20	0,096	0,085	0,003
	LogTF-IDF	**4**	**2**	**0,376**	**0,381**	**0,009**
		8	4	0,207	0,193	0,005
	TF-IDF	**6**	**3**	**0,385**	**0,382**	**0,010**
		11	4	0,260	0,254	0,007
		30	8	0,148	0,142	0,004
D14	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

The first step of the joint approach involves the document-term matrix factorization (SVD) and a reduction phase to select only the relevant dimensions for the subsequent steps of the analysis. The importance of the dimensions is evaluated through the magnitude of corresponding the singular values in the matrix S. For each weighting schema the three values used during the data reduction phase are reported and the best configuration is highlighted. A common trend among all the dataset suggests that only a few dimensions are needed to describe the whole corpus. The eigenvalues represent the percentage of how much a Term affects a Topic and they are selected basing on their significance. From the K-lsa values in the table it is possible to see that a number between three and six dimensions are enough to describe the whole database, this is valid for all the weighting schemas. When the selected number is three, for example, it means that each one of the three matrices, in which the original matrix X is decomposed, is reduced to a dimension equal to three: we are in a space where the database can be represented.

As shown in the eigenvalues graphics below, in fact, the elbow is always at the begin and then the curve become flat; the significance of the added eigenvalues (i.e. dimension) is even less relevant in order to describe the dataset content. This configuration reflects the fact that reviews already belong to a category product so it is not easy to identify subgroups,it is like a main topic already exists and the algorithm is looking for subtopics. At least we can expect two clusters in which review from satisfied clients are divided from the ones wrote by clients that are not happy with the bought products.

For each weighting schema the eigenvalues graphic is reported for the Major Appliances dataset. The selected K-lsa for the Boolean-IDF schema is equal to 3, for the LogTF-IDF schema is equal to 5 and for the LogTF-IDF schema is equal to 6. The graphs below demonstrate that these choices make sense.

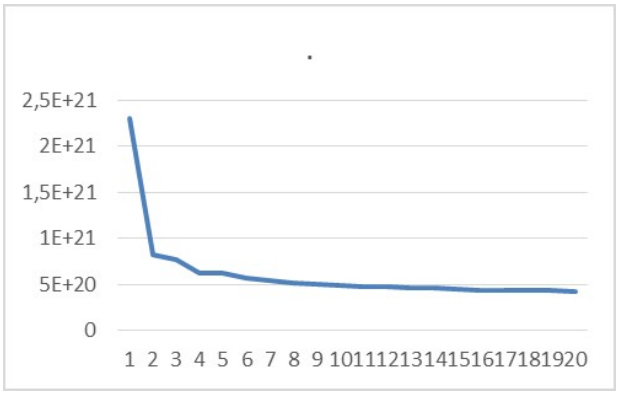


Figure 3: Top singular values for dataset D9 weighted via Bool-IDF

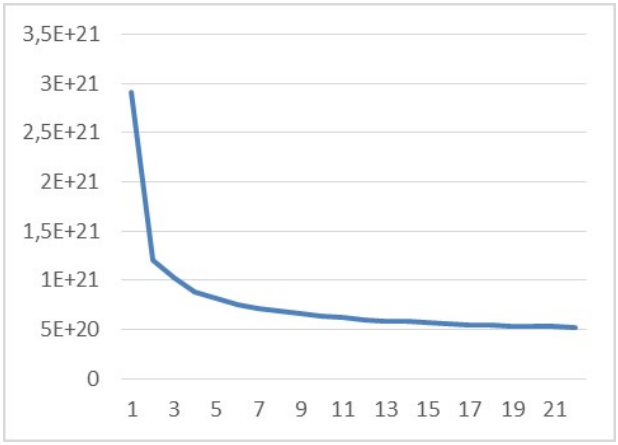


Figure 4: Top singular values for dataset D9 weighted via LogTF-IDF

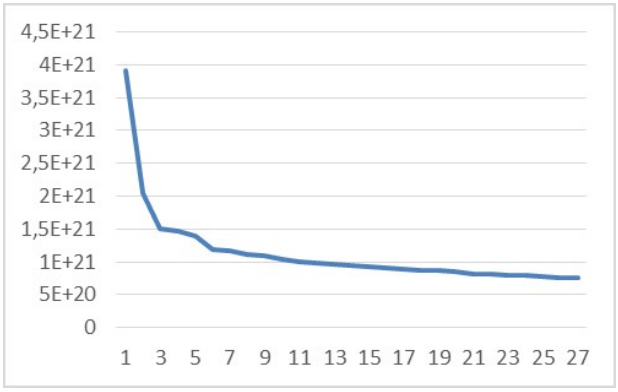


Figure 5: Top singular values for dataset D9 weighted via LogTF-IDF

Escape selects for the best configuration a low number of dimensions meaning that the data

distribution is not so variable, this was predictable since each data-set contains reviews referred to a specific category of products; nevertheless, there are some differences depending on the applied weighting schema.

In the Boolean-IDF weighting schema case (Figure 1) the optimal selected number of dimensions (K-lsa) is the lowest in comparison with the other schemas because the Boolean local weight does not distinguish between words which appear many or a few times, it takes into account only if a word appears or not so all the terms have the same local relevance. The number of dimensions increases using the log term frequency as local weight (Figure 2) but it remains lower then the number selected using the Term Frequency (Figure 3) because this latter is able to differentiate the terms at most (in term also of their frequency) while, in the first case, applying th logarithm, the importance of terms which appear frequently is diminished and the terms are less differentiated. From a certain frequency, in fact, the value of the logarithm function, i.e the local weight, tends to flatten, reducing the importance of the terms which appear very frequently.

Once the joint approach algorithm run, ESCAPE reports the best three configurations also for clustering. For each dimensionality reduction parameter (K_lsa), ESCAPE selects the best value for the clustering phase. To do this and report to the user only the best configuration a majority model which consider Silhouette-based quality indices is exploited. Silhouette measures the cohesion and the separation, so it indicates how well the clustering phase has performed on the various datasets. The weighted purified distribution of silhouette index (WS), the average silhouette index (ASI), and the global silhouette index (GSI) are computed. For each index, separatly, a rank from 2 to the maximum number of clusters is defined, then the global score function is computed and a final rank sort all the scores.

$$score = ((1 - \frac{rank_{GSI}}{k_{max}}) + (1 - \frac{rank_{ASI}}{k_{max}}) + (1 - \frac{rank_{WS}}{k_{max}}))$$

The score lies in the range [0, 2.842] since Kmax is set equal to 20 but, from the table above, it is possible to see that this threshold for the optimal number of clusters is rarely reached. In the figure below the silhouette-based ranking process for detecting the best

partitioning is shown. It is referred to the Major Appliances data-set for one of the three selected k-lsa (6) using the Boolean-IDF weighting schema.

k_Cl	GSI	rank	ASI	rank	Weight_sihl	rank	score_fun	final_rank
2	0,3626	3	0,3644	3	0,0369	3	2,55	3
3	**0,4225**	**1**	**0,4306**	**1**	**0,0420**	**1**	**2,85**	**1**
4	0,3733	2	0,3678	2	0,0369	2	2,7	2
5	0,3302	5	0,3257	6	0,0326	6	2,15	6
6	0,3488	4	0,3508	4	0,0348	4	2,4	4
7	0,3279	6	0,3268	5	0,0331	5	2,2	5
8	0,3213	7	0,3186	9	0,0316	14	1,5	9
9	0,3128	11	0,3130	15	0,0312	17	0,85	15
10	0,3067	17	0,3080	18	0,0307	19	0,3	18
11	0,3104	12	0,3139	14	0,0320	9	1,25	11
12	0,3184	9	0,3206	8	0,0322	8	1,75	8
13	0,3075	16	0,3108	17	0,0312	16	0,55	17
14	0,3194	8	0,3218	7	0,0324	7	1,9	7
15	0,3129	10	0,3173	10	0,0319	11	1,45	10
16	0,3096	15	0,3153	12	0,0319	10	1,15	12
17	0,3097	14	0,3140	13	0,0313	15	0,9	14
18	0,3098	13	0,3154	11	0,0318	13	1,15	12
19	0,3023	19	0,3053	19	0,0311	18	0,2	19
20	0,3048	18	0,3113	16	0,0319	12	0,7	16

Table 3: Rank function example for a dataset D9 weighted using Bool-IDF.

Looking at the graphic below, which represents the silhouette values it is clear why the best partitioning involves three clusters; this partition have the highest silhouette values which detect the best clustering process. Weighted silhouette values are misleading, they seems to be not relevant but actually they vary between 0 and 0.1818 (which is the conversation factor) so if coverted, higher values can be obtained. For example for the best configuration the converted value is equal to 0,23. . Global and average silhouette indices, in many of the cases like this, tend to be similar since reviews are very short texts. For the non-optimal numbers of cluster, instead, all the values are lower and stable giving more support to the

fact that three clusters are the best in terms of cohesion and separation.

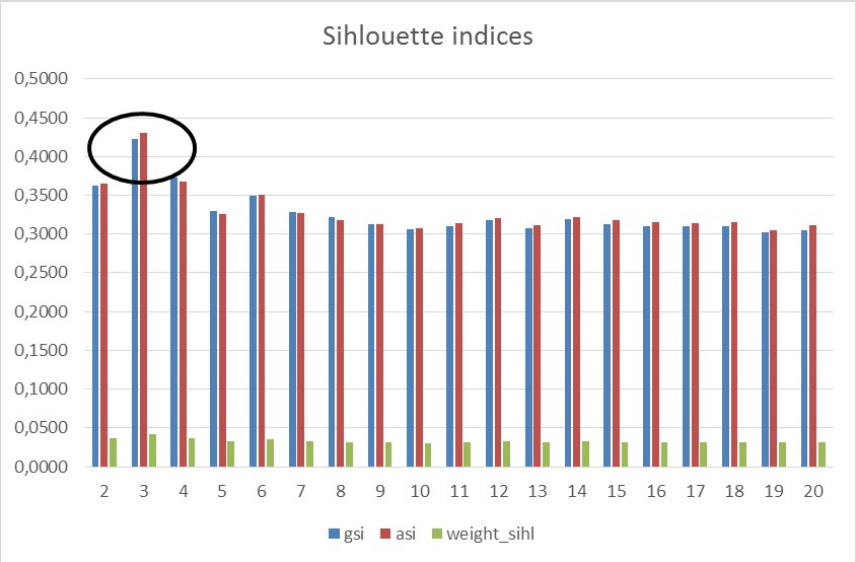


Figure 6: Plot of the silhouette-based indices.

For a deeper comparison between the weighting strategy the symmetrix matrix A with the ARI values for each couple of schemas (in the rows and columns) is computed. The Adjusted Rand Index (ARI), when is large, means the agreement between two partitions; the maximum is equal to one while the expected value is in case of random clusters equal to 0 (solving the issue of the Rand Index with no constant expected value for random clusters).the ARI index penalises more partitions with different number of cluster, but in some case it is low also for configuration with the same number of cluster mining that ESCAPE has identified two partitions of the same dataset. Boolean-IDF has, for many datasets, lower values with respect to the other weighting schemas.

D9	B-IDF	TF-IDF	LogTF-IDF
B-IDF		0.3670	0.774
TF-IDF			0.47359

Table 4: d9: ARI index for the joint approach.

M-D4	B-IDF	TF-IDF	LogTF-IDF
B-IDF		0.3073	0.411
TF-IDF			0.467

Table 5: D4: ARI index for joint approach.

The clusters cardinality has been also investigated for a deeper evaluation of the weight

impact on clustering process. As already said, on average, the boolean local weight tends to identify less clusters but there is also an inverse trend for three datasets where the number of detected clusters is the lowest using TF and it increases with LogTF and even more with Boolean. The partitions obtained by Boolean ad LogTF are, in general, more similar, especially for D2 (Camera), D4(Digital Software), D13(Shoes), D14 (Software). Actually for the Gift Card category dataset the partitions identified with all the different weighting schemas are almost equal. However, partitions are quite homogeneous for the majority of the dataset except D3 and D11 where TF and LogTF have identified the strong predominance of a cluster; this is valid also for D4 with TF.

D9	B-IDF	LogTF-IDF	TF-IDF
Cl1	31215	8118	13772
Cl2	42213	24252	21449
Cl3	49743	24363	8009
Cl4		12860	31114
Cl5		22362	11549
Cl6			6062

Table 6: clusters cardinality for dataset D9 for the Joint approach

D2	B-IDF	LogTF-IDF	TF-IDF
Cl1	299331	388364	48833
Cl2	320081	344434	261544
Cl3	411823	278163	266810
Cl4	286584	306858	149315
Cl5			320772
Cl6			270545

Table 7: clusters cardinality for dataset D2 for the Joint approach

D4	B-IDF	LogTF-IDF	TF-IDF
cl1	30266	26399	60669
cl2	19211	15623	22366
cl3	22489	33458	13191
cl4	24260	20746	

Table 8: clusters cardinality for dataset D4 for the Joint approach

D7	B-IDF	LogTF-IDF	TF-IDF
Cl1	21064	21874	30997
Cl2	32982	20887	30278
Cl3	26253	28020	23210
Cl4	14672	20666	22996
Cl5	26830	17020	14320
Cl6		13334	

Table 9: clusters cardinality for dataset D7 for the Joint approach

3.4.3 Visualization

As already said, visualization techniques are fundamental for a whole comprehension of the analysis and for allowing different kind of user to understand the results.

First, the correlation matrix maps, which graphically display the impact of the weighting functions, are analysed. The most interesting are also reported. For the main beans, five coloured correlation ranges, from white to black, have been used: from 0.0 to 0.5 white, from 0.5 to 0.62 light gray,from 0.62 to 0.75 gray, from 0.75 to 0.87 dark gray and from 0.87 to 1.00 black. For each corpus the dot product between all document pairs, sorted by category, is computed and basing on the range in which this value fall the map is coloured. Documents which belong to the same cluster are more similar than the ones which belong to different clusters so the proximity will be higher and this will be represented by dark areas within the map.

Looking at the Digital Software maps, reported below, it is possible to see that both Log-TF and Boolean are able to identify four dark rectangles representing the clusters; for Log-TF rectangles dimensions are quite homogeneous while in the Boolean case the dimensions, i.e. the number of elements in each cluster are more variable, as already shown during the cluster cardinality analysis above. TF detects instead three clusters, moreover, within the biggest one, four areas with strongest relationship can be identified suggesting that there are four relevant subtopics within the same category.

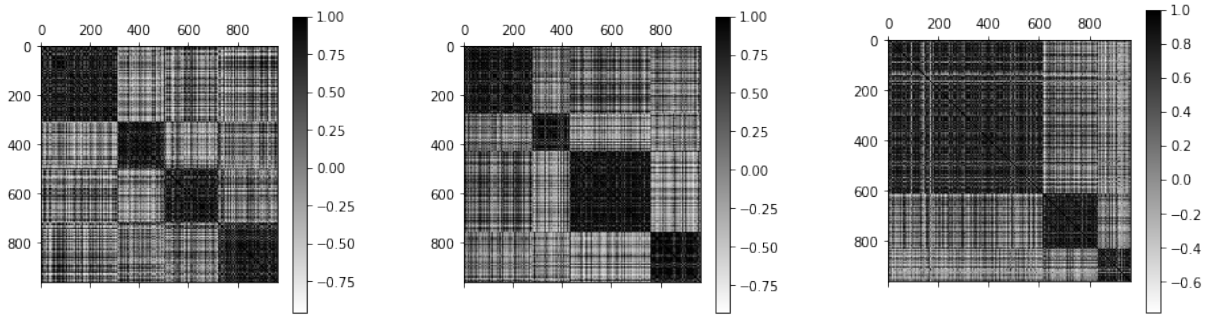


Figure 7: Dataset D4. Correlation matrix maps for the best configurations.

The correlation matrix maps are not so clear for the Gift Card data-set: there are 5 or 6 darker rectangles according to the number of clusters identified as the best configurations with the different weighting schemas but many sub categories are highlighted around the maps. Especially Boolean and Log-TF are not good in modelling the categories since two of them seem to constitute only one cluster.

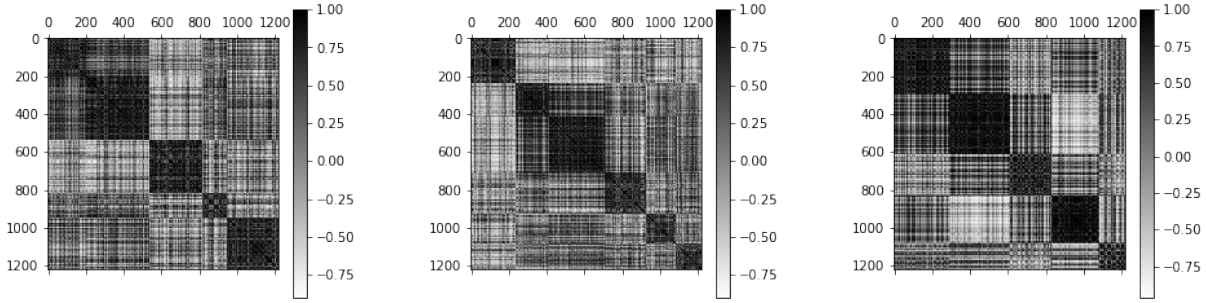


Figure 8: Dataset D3. Correlation matrix maps for the best configurations.

It is also interesting to look at the maps of the Personal Care and Digital Music products categories dataset. In both cases TF highlights a very strong relationship between the documents belonging to the biggest and darkest rectangular ,i.e. cluster, and their dimensions are highly heterogeneous. But there are also some remarkable subcategories well represented by dark areas. Actually these very strong links are in contrast with the expectation because when you have a few big clusters, the elements within them are expected to be more variable and not highly related to each other.

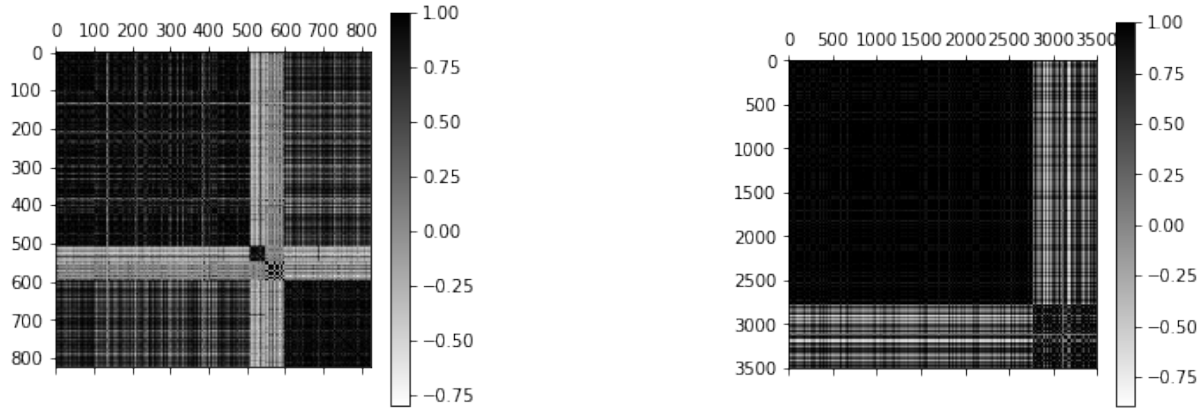


Figure 9: Dataset D11. Correlation matrix maps for the best configurations.

In order to explore the balancing between the resulting clusters, t-SNE plots have been exploited since they provide compelling and faithful two-dimensional "maps" from high dimensional data. In this way the dataset behaviour can be analysed in a smaller space. t-SNE is a very useful representation tool but only if proper interpreted. As previously said, in order to effectively use t-SNE representation and prevent some common misreadings, one should focus on perplexity and iteration parameters. Common perplexity values are in the range [5,50] but in general, to avoid unexpected behaviour, they have to be minor then the number of points. This is a very high number since the dataset under analysis have huge dimension, so, first, lower values in the common range has been used and then much higher ones which gives a better sense of the global geometry. T-SNE, indeed, excels at revealing local structure in high-dimensional data but sometimes tends to misrepresent the global geometry. For what concern iterations, it is important to reach a number which corresponds to a stable configuration. In order to immediately detect this value a big number of iterations has been first set, and the algorithm automatically says after how many runs it diverges.

Essentially, this representations provide information on how the documents are distributed between clusters (i.e topics), in any of the maps it is not possible to see well separated clusters that are identified by points of the same colour. In some case, increasing the perplexity values, it is possible to see a convergence to a particular configuration as shown below

for the Major appliances, Software and Personal Care product category with overlapping clusters.

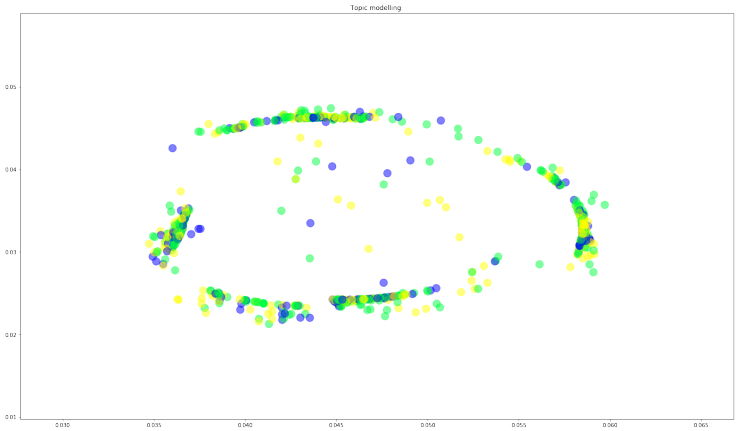


Figure 10: Dataset D9. t-SNE representation. B-IDF weighting schema K=3

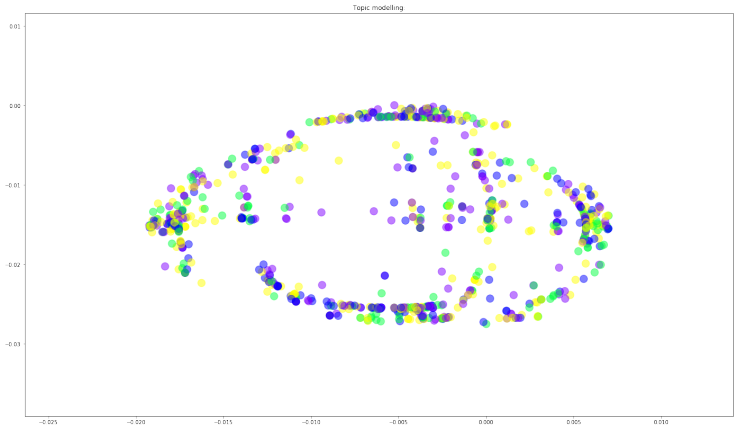


Figure 11: Dataset D4. t-SNE representation. B-IDF weighting schema K=4

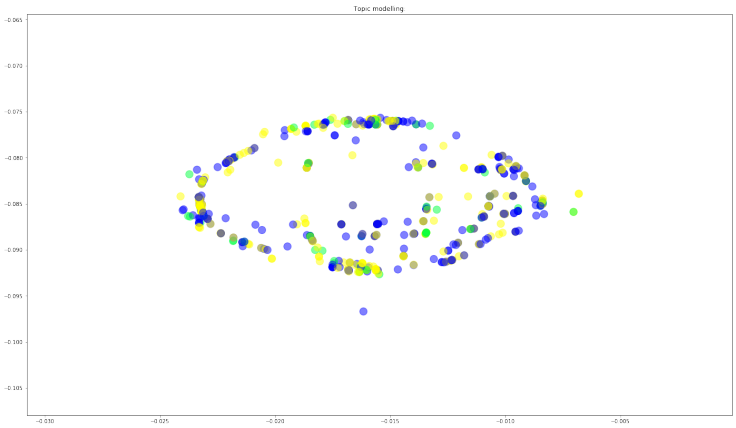


Figure 12: Dataset D11. t-SNE representation. LogTF-IDF weighting schema K=3

For most of the other dataset the t-SNE output are point clouds, this is a common behaviour among the different weighting schemas. Nevertheless it is possible to notice a difference between Boolean-IDF and LogTF-IDF weighting schemas results for Mobile Electronics (Fig.13, Fig.14) and Luggage (Fig. 15, Fig.16) product category dataset. First, for D5 one can better glimpse a configuration of the documents clusters than for D8, while for both datasets the shape of the Boolean-IDF clusters are more defined with respect to LogTF-IDF.

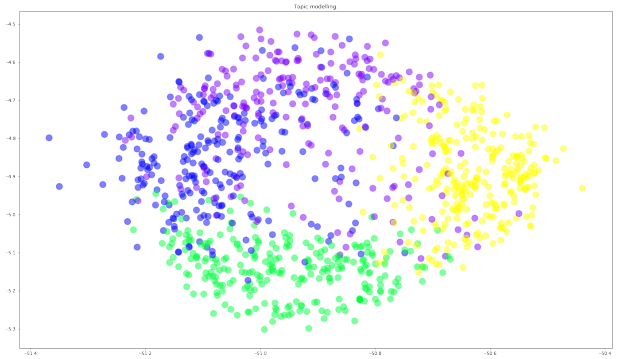


Figure 13: Dataset D5. t-SNE representation. B-IDF weighting schema K=4

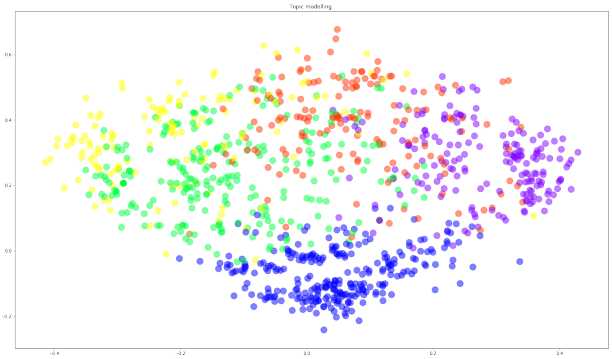


Figure 14: Dataset D5. t-SNE representation. LogTF-IDF weighting schema K=5

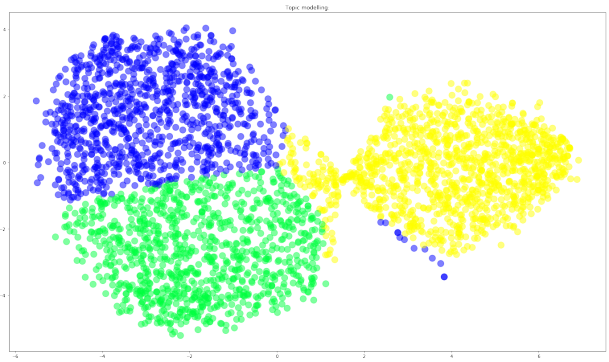


Figure 15: Dataset D8. t-SNE representation. B-IDF weighting schema K=3

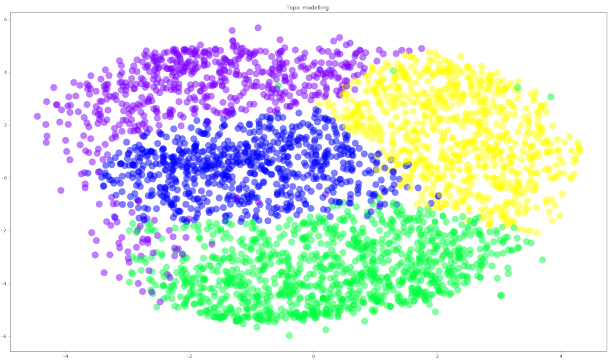


Figure 16: Dataset D8. t-SNE representation. LogTF-IDF weighting schema K=4

Also if the representation shapes vary between the results, in most of the cases, colours are well balanced except for Software (Fig.19), Personal Care (Fig.18) and Music (Fig.17) datasets when with TF-IDF weighting schema where the predomination of a colour is evident looking at the maps, meaning that documents are not equally distributed among the clusters (i.e topics).

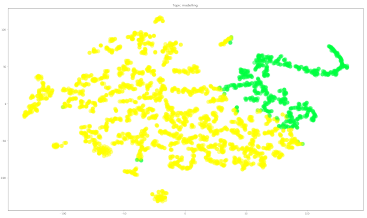


Figure 17: Dataset D3. t-SNE representation. TF-IDF weighting schema K=2

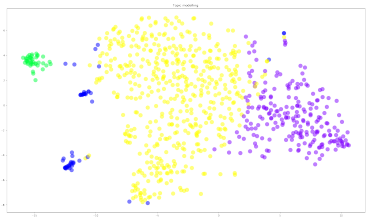


Figure 18: Dataset D11. t-SNE representation. TF-IDF weighting schema K=4

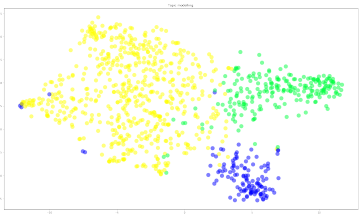


Figure 19: Dataset D4. t-SNE representation. TF-IDF weighting schema K=3

Until now the best identified partitions are anonymous, it is known how many groups are identified within each dataset and how documents are distributed between them but what they talk about is unknown so ESCAPE involves the words clouds representation to gain a better level of explain-ability. Through a FP-Growth algorithm, the set of words characterised by a frequency greater of a certain threshold (named support) is extracted and showed in a clear and simply way. This is done for each cluster of the identified best partitions in order to reveal the main hidden topics. Actually, the type of data under

analysis is not so compliant with this process: the reviews, also if about different products, have similar structure and people use a restricted vocabulary and often the same words. Therefore the top-k frequent items are, in most of the cases, the same for the different clusters within a dataset and they are not useful to identify the topic. Specific dictionary with the common most frequent words have been then built and removed before applying the FP-Growth algorithm. The focus has also been shifted on the different words to better detect which are the main distinctive features of each cluster, i.e topic. The first thing that is remarkable is that the most frequent words have always a positive like "Love", "great", "well", these are always included in the dictionary to remove ma still this mean that people tend more to make a review when they are satisfied about the product they have bought. For the dataset of Camera products reviews both Boolean and LogTF have identified a group of comments related to charging theme: "charger" "hours" are some of the distinctive and most used words , another cluster brings together comments concerning remote device since they are characterised by words such as "attach", "strap". Within the Personal care products reviews dataset all the different weighting schemas highlight a set of comments where the most frequent terms are related to olfactory sensation (es: perfumes, body spray etc.), relevant words in another cluster concern more body wellness and diet; finally a more unexpected set of document which go through the arguments of safety and security is detected.(See Fig.20). TF-ID,F which provide a more detailed analysis by identifying one more cluster than the other schemas, highlights a group of reviews about epilation where the name of a brand (Braun) is part of the most frequent item-set that describes the topic.(Fig.21)

Figure 20: D11:WordCloud representations

Figure 21: D11: Wordcloud representation of cluster 4 for TF-IDF

It is interesting to highlight that within the Mobile electronics products comment dataset, a cluster whose distinct features are referred to Garmin nuvi, can be identified. It is a GPS navigator which is very used for outdoor sports and has many accessories as can be realized from the words in the image below.

Figure 22: D5: Wordcloud representation for cluster 2 for the Bool-IDF weighting schema

3.4.4 Probabilistic approach

As already said the probabilistic topic modelling approach has been run only on the dataset containing the reviews with the highest average length. Five parameters have to be set before running the algorithm.

An important feature of this methodology is that it does not require a-priori knowledge of the structure of the data under analysis but it requires the number of topics i.e. clusters in which divide each corpus to be previously set. ESCAPE involves a novel iterative approach, TOPIC SIMILARITY, to automatically identify a proper number of clusters (K). In order to select a proper values for the configurations of the probabilistic modeling TOPIC SIMILARITY asses how topics are semantically diverse basing on their content (Words) and not on the internal LDA perplexity parameter or probabilistic quality metrics,

as the others state-of-the-art technique. The upper bound for K is set equal to the average length of the documents in the corpus, as done for the joint approach. In the event that the average length of the documents is grater then the number of documents in the analysed corpus the upper-bound becomes the average frequency of the terms. The hypothesis behind is indeed that each words in a document belongs at most to a topic.
For the other four parameters a self-configuring algorithm is not available so they have to be set by the analyst.

- The maximum number of iteration within the model has to converge is set to 100.

- The Dirchelet distribution used to draw the per-document distribution and estimate the LDA model is computed setting the optimizer to be the Online Variational Bayes, otherwise it would have been unfeasible to compute it.

- α and β, respectively, the document and topic concentration are set to maximize the log-likelihood of the data under analysis and they have to be equal or major then 0 because of the choice of the Online Optimizer. The default value for α is set equal to $\frac{50}{K}$, while for β it is set equal to 0,1 as proposed in the article [**?**]

3.4.5 Solutions

The first thing that has to be highlighted is that the experiments need much more time in comparison with the ones run using the LSA. This is a common behaviour but it is even more emphasized because of the structure of the data under analysis which makes hard for the LDA algorithm to infer a hidden structure within the documents.
For any of the experiments ESCAPE has not been able to identify three good clustering values. The Topic Similarity strategy can provide at least three proper values which have to satisfy two conditions considering the topic similarity function but the search stopped

before because the pre-set upper-bound was reached.

ESCAPE didn't identified any proper k value for the Major Appliances data-set which is made by documents with average length equal to 30 words. The points highlighted in the figures below are proper k-values selected not by the algorithm but by the analyst basing on the trend of the similarity index. Where the function is more stable it means that the k-value is a proper candidate to model the dataset. This is true for all the weighting schemas suggesting that the poor performance are independent from them. Actually different behaviour can be identified with a deeper analysis. It is interesting to investigate more the differences since weights highlight the importance of terms within the documents and thus, affect the probabilistic model generated by the LDA. Using the local logarithmic term frequency the similarity measures are more stable and it is possible to identify three good value for k (see Fig.23), this is instead not possible using the boolean weight since there is only one point where the ToPIC Similarity function seems to be more stable (Fig.24).

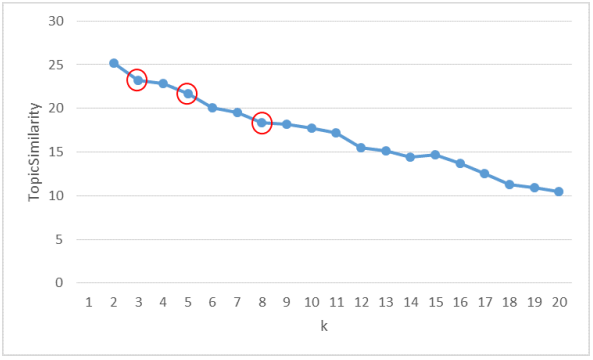


Figure 23: Dataset D9. Topic Similarity index. LogTF-IDF weighting schema

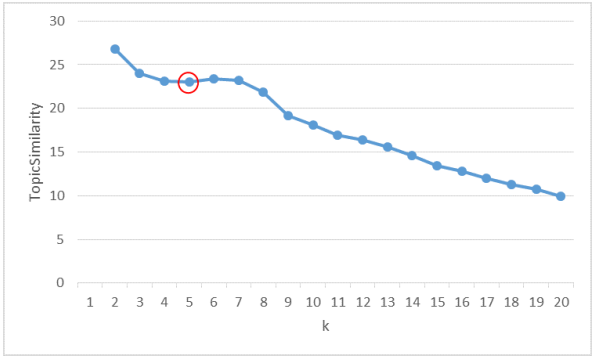


Figure 24: Dataset D9. Topic Similarity index. Bool-IDF weighting schema

The performance are slightly better for the Videodataset. It is the one composed by the longest reviews (46 words on average) making easier for the LDA algorithm to infer a structure within them. Nevertheless, only one proper k value which satisfies the two Topic Similarity function conditions has been find out by ESCAPE. The identified partitioning value is equal to 2 for all the weighting schemas and the function has a pick in correspondence of the value 5 but then the behaviour changes. When the boolean local weight is used

the function is more regularly decreasing in comparison with the other cases. A comparison between boolean and the Logarithmic frequecny is showed in the figures below.
In the fig.26, as well as the value 2 (detected by ESCAPE) also 12 could be considered as a proper clusters number since the function seems stabilizing.

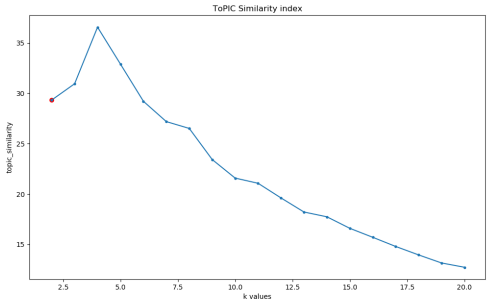


Figure 25: Dataset D15. Topic Similarity index. Bool-IDF weighting schema

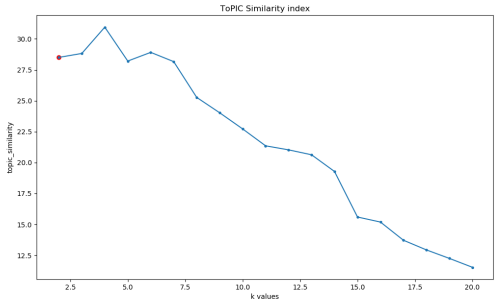


Figure 26: Dataset D15. Topic Similarity index. LogTF-IDF weighting schema

The goodness of the statistical model has been explored thanks to the well known quality indices of Perplexity and Entropy computed within ESCAPE. They are summarised in the table10 along with the best identified partitioning.

Dataset	Weights	K_Cl	Perplexity	Log-likelihood
D8	Bool-IDF	5	7,273681	-132217367
	LogTF-IDF	3	7,352020098	-133641368
		5	7,263175195	-132026390
		8	**7,190609656**	**-130707329**
	TF-IDF	5	7,270052194	-13122795
D14	Bool-IDF	2	7,588552184	-135192847
	LogTF-IDF	2	7,581219438	-131108665
	TF-IDF	2	7,583352794	-131532596

Table 10: Experimental results for dataset D8 and D14 for the probabilistic approach

Recall that higher Sihlouette values means better clustering partitioning but this is the opposite for the Entropy and Perplexity values. If perplexity values are high, it means

that the probability estimate of the corpus are not good. In both case the perplexity has values around 7 but they are lower for the Major Appliances dataset (D14) meaning that the model better fit the data. Analysing the partitioning results by using only these quantitative measures is not enough and other visualization techniques are necessary for a better interpretation. Moreover, Topic similarity process provide (theoretically) three k values between which the analyst can select the one that reflects the best required granularity of the clusters and ,consequentially, topics. For example, basing only on perplexity, one should select 8 has optimal number of clusters in D14 with LogTF-IDF schema since the perplexity is the smallest, but it is still not sure that this is the most proper configuration. The log-likelihood has higher values if the model well fit the data, it can be seen as a measure of "how likely things are", so since it is always negative, lower absolute values are better. The only real interpretation for log-likelihood is then higher is better. If one looks at only one model for the data under analysis, the number is absolutely meaningless while it is useful for a comparison between models. Basing on this considerations it is possible to confirm that eight clusters seem the optimal solution.

The obtained cardinality partition and the Adjusted Rand Index for both datasets are reported in the following tables. In this way it is possible to validate or confute what statistical indices have suggested and better investigate the performance and the impact of each weighting strategy on the same dataset.

D8	B-IDF	LogTF-IDF	TF-IDF
Cl1	12811	15936	17999
Cl2	17343	10178	18391
Cl3	27787	6823	27458
Cl4	22197	6882	15035
Cl5	11817	21611	13072
Cl6		10184	
Cl7		14347	
Cl8		5994	

Table 11: D9: Cardinality of each cluster set found for the probabilistic approach.

D14	B-IDF	LogTF-IDF	TF-IDF
Cl0	162584	264643	250961
Cl1	246967	144908	158590

Table 12: D14: Cardinality of each cluster set found for the probabilistic approach.

With respect to Major Appliances (D9) clusters cardinality, documents are quite well balanced between the different clusters also if there is always one cluster that can be well recognized as the biggest one. Cardinality does not have a linear trend for any of the weighting strategy but the cluster in the middle is always the one with the highest number of reviews. LogTF-IDF weighing strategy differs from the others since it provides a more detailed analysis discovering also subtopics, in addition to the five main topics already discovered also by the other schemas. This is different level of results granularities is not present for the Video product category dataset (D14). In this case between the two clusters identified with all the the weighting schemas. it is possible to clear recognize a cluster composed by a larger number of reviews. A good news is not to find empty cluster.

D8	B-IDF	TF-IDF	LogTF-IDF
B-IDF		0.371616	0.336831
TF-IDF			0.491105

Table 13: D9: ARI index for Probabilistic model.

D14	B-IDF	TF-IDF	LogTF-IDF
B-IDF		0.201689	0.218886
TF-IDF			0.228325

Table 14: D14: ARI index for Probabilistic model.

For D9 ARI index has lower value for the Boolean-IDF weighting strategy however all the values are not so different and and also not high (never more than 0.5). Since the ARI

index reflects the agreement between two partitions it was predictable to have bigger value for the Boolean and TF pair since they have the same number of clusters but the not so high value means that the topic detected are not the same for each cluster. For D15 the ARI index shows a weak agreement between the different weighting strategies also if two clusters, i.e topics, are detected by each schema.

3.4.6 Visualization

Interesting and explanatory output obtained thanks to the already mentioned visualization techniques are reported. Specifically this time, t-sne maps, word clouds for the main topics and graphs representations will be exploited.

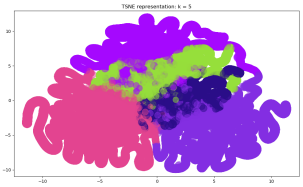


Figure 27: Dataset D9. t-SNE representation. Bool-IDF weighting schema K=5

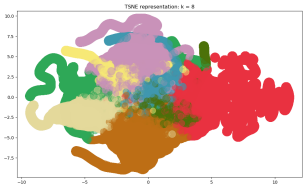


Figure 28: Dataset D9. t-SNE representation. LogTF-IDF weighting schema K=8

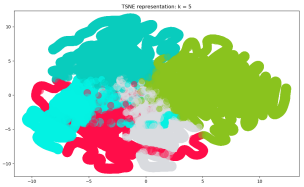


Figure 29: Dataset D9. t-SNE representation. TF-IDF weighting schema K=5

The best partitioning t-sne maps for all the weighting strategies for the Major Appliances product category dataset are displayed above (Fig.27, Fig.28, Fig.29) while for th Video product category dataset they are displayed below (Fig.30 Fig.31 Fig.32). In figures 27 and 29 one can see similar shapes and distribution of the documents between the clusters (i.e topics). In Fig 28 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. All the representation have a features in common: the maps are points clouds and no particular configuration can be identified. In figures 32 and 33 not a point cloud but a defined line which is divided into two parts (i.e clusters) is displayed, these are not well separated but the colours are quite well balanced also if, for all the weighting strategy a light predominance of a cluster can be noticed.

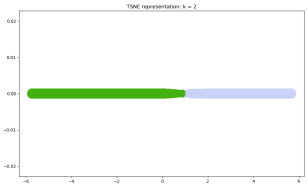


Figure 30: Dataset D14. t-SNE representation. Bool-IDF weighting schema K=2

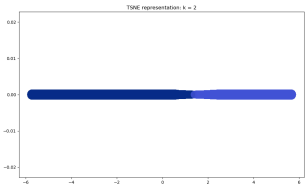


Figure 31: Dataset D14. t-SNE representation. LogTF -IDF weighting schema K=2

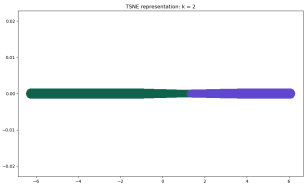


Figure 32: Dataset D14. t-SNE representation. TF-IDF weighting schema K=2

Graph representation are now exploited in order to show the strongest links (edges of the graph) between Topics and Terms which are represented by nodes. Recall that if a term belongs to more topics then the dot colour is pink and that an edge is drawn only if the probability of a term belonging to a topic is major than the trees-hold. In all the graphs there are many pink dots, these reflect the lexical structure of the reviews which are written using a common and poor language shared among all the reviews also if about different products.

Looking at the following graphs, it is clear that clusters are not well separated, also if the graph is not connected there are many words that appear in more than one cluster. Each topic, represented by the light blue node, is characterised by a set of distinctive terms and a set of shared terms with at least another one topic.

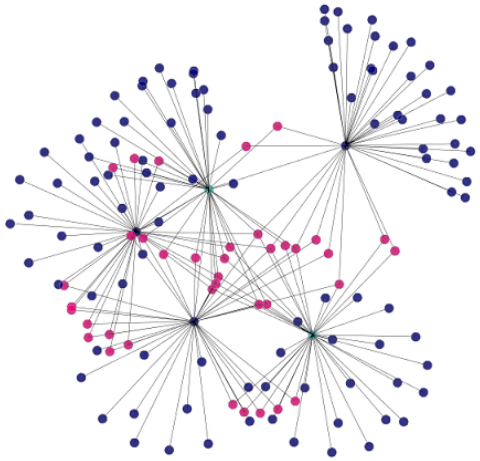


Figure 33: Dataset D9. Graph representation. Bool-IDF weighting schema K=5

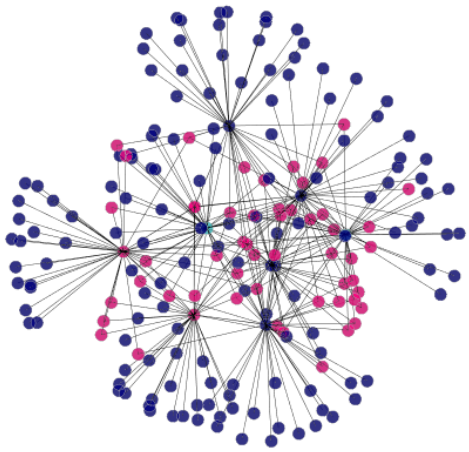


Figure 34: Dataset D9. Graph representation. LogTF -IDF weighting schema K=8

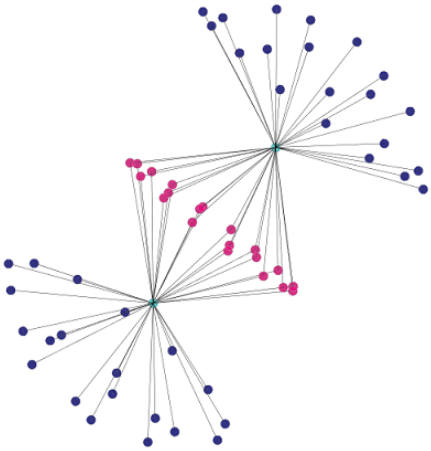


Figure 35: Dataset D15. Graph representation. Bool-IDF weighting schema K=2

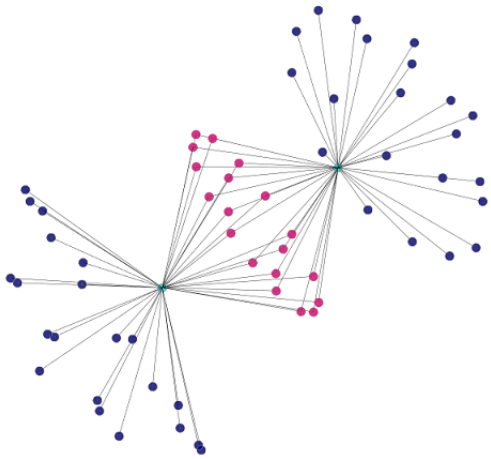


Figure 36: Dataset D15. Graph representation. LogTF -IDF weighting schema K=2

Although the two clusters showed by the graphs have some terms in common from the word clouds one can realised that one category is main related to opinions about the characters of a movie while the other one is characterized by comments with general movie impressions.

Figure 37: Dataset D15. WordClouds representation

3.5 Mix Dataset

Novel datasets have been created in order to test the ability of ESCAPE on topic detection. Different numbers of reviews from different product category collections have been brought together into novel datasets using python so it has been possible to compare the number of product category integrated in the created corpus and the number detected by ESCAPE having a previous knowledge of the number of categories. The a-priori knowledge of the label of each document is removed bringnig together random documents from different corpus in order to understand if ESCAPE is able to remove noise. Several types of novel corpus have been created: one with reviews about similar products (regarding the Electronics/digital topic) and three with reviews of different products including reviews from six to ten different product categories. Some novel collections have the same number of documents from the various categories, while the others have the same number of reviews from each product category. The idea is to have a proper degree of variety and investigate different level of similarity within data. The documents from each collection have been randomly selected in order to have as representative a sample as possible. A detailed description of the novel Mix Dataset is reported in the tables below.

Mix-Dataset1			
source datsets:	D9	D8	D7
number of reviews:	50k	20k	100k

Table 15: structure of the mix dataset 1

Mix-Dataset2						
source datsets:	D9	D8	D7	D11	D5	D4
number of reviews:	50k	20k	100k	10k	20k	30k

Table 16: structure of the mix dataset 2

Mix-Dataset3 (SIMILAR)						
source datsets:	D2	D3	D4	D5	D14	D15
number of reviews:	10k	10k	10k	10k	10k	10k

Table 17: structure of the mix dataset 3

Mix-Dataset4									
source datsets:	D1	D2	D3	D6	Baby	D10	D11	D12	D13
number of reviews:	10k	10k	10k	10k	10k	10k	10k	10k	10k

Table 18: structure of the mix dataset 4

3.5.1 Analysis and Results

The same process, as made for the original dataset, has been applied. At first the statistical feature are computed. Hapax-Rate values and Min-Freq values are not included in the table below because they are respectively equal to 0 and 2 for the same reasons of original dataset.

	N-Doc	Dict.	Max-F	Tot_words	Avg-l	Avg-F	TTR	Giraud	max-var
M-D1	147856	14389	71655	2644035	17,883	183	0,0054	8,8491	1283538102
M-D2	204885	19185	72501	3961198	19,334	206	0,0048	9,6394	1314026250
M-D3	106295	26435	27092	2608925	24,544	98	0,0101	16,3662	183467025
M-D4	182270	30065	53275	3850256	21,124	128	0,0078	15,3220	709503132

Table 19: Statistical characterization of mix datasets

First, it is possible to see that the number of documents does not corresponds to the number of reviews that have been brought together in the created dataset. This is because of the too-short reviews that have been deleted by ESCAPE during the pre-processing phase.
For what concern the lexical complexity, both the TTR and Giraud indexes have, on average, higher values in comparison with the original dataset. As it was predictable, the Giraud index has a higher value for the dataset made by reviews from ten different products category (M-D4) since it is the most heterogeneous but it has to be highlighted that the dataset made by joining the reviews about electronic related products(M-D3) has the highest values suggesting a lexical richness used for the opinions of the clients interested in this field. The data sparsity, expressed by TTR, have values in the range [0.005,0.01], all the dataset have, therefore, a high level of lexical variation, especially M-D3 which is the most sparse dataset. However, there are some words that are repeated many times, in fact the average frequency does not reach the extremely high values of the original datasets but has values between 100 and 200. This is confirmed also by the quite big difference between Tot_words and Dictionary values which are, respectively, the number of words in the collection of documents with and without repetitions

3.5.2 joint approach

Initialy, following the same process as the category datasets one, experiments have been run applying the joint approach. It has to be highlighted that in this case the expected number of categories is known a priori and is equal to three, six or ten since the dataset have been specifically built. The two parameters, the T value for the dimensionality reduction phase, and the upper-bound for the maximum number of clusters, have been respectively set equal

to 20% of the number of documents and the average document lenght.The experiments have been run using the different weighting schemas and the results are shown in the following tables. Only the best configuration, i.e the one with the best sihlouette values, within the three selected K_lsa, is reported for all the schemas, while for the Boolean-IDF strategy the three selected values during the reduction phase are reported as an example of some interesting features.

Mix-Dataset1					
Weights	K_Lsa	k_Cl	GSI	ASI	W_SIHL
Bool-IDF	12	2	0,381	0,359	0,0223
LogTF-IDF	13	2	0,354	0,298	0,0206
TF-IDF	13	4	0,417	0,377	0,0291

Table 20: Experimental results for the dataset mix 1 for the Joint approach

Mix-Dataset1					
Weights	K-Lsa	K-Cl	GSI	ASI	W-SIHL
Boolean-IDF	7	5	0,332	0,438	0,027
	12	**2**	**0,381**	**0,359**	**0,022**
	26	2	0,250	0,216	0,015

Table 21: Experimental results for the dataset mix 1 with Boolean-IDF strategy for the Joint approach

The results obtained after running ESCAPE on the dataset made by documents from three different categories of products show a higher number of dimensions needed to describe the collection of documents. This means that the data distribution is more variable and the clustering activity will be more complex. The selected k-lsa for the best configuration is, unlike the results of the original datasets, not the lowest of the 3 values identified by SVD in fact, for example, for the Boolean-IDF strategy it is equal to 12 (tab. 21). The significance of the added eigenvalues (i.e. dimensions) in not immediately irrelevant and the curve becomes flat after higher values of k-lsa as shown in the Fig.38

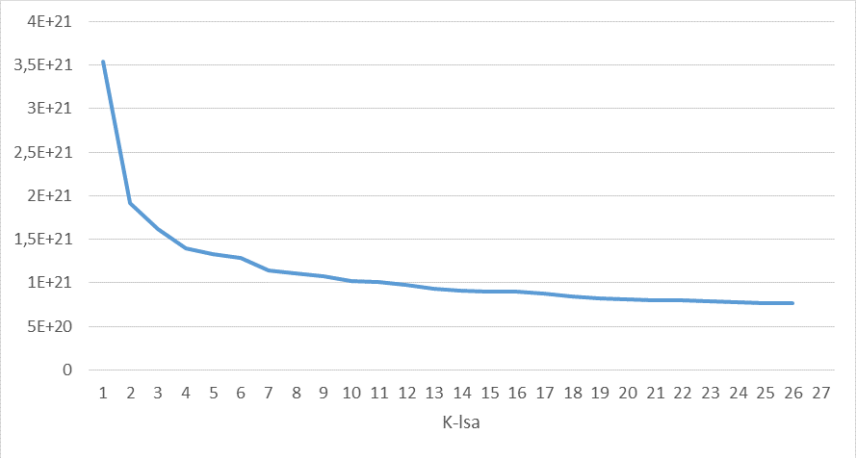


Figure 38: Top singular values for Dataset M-D1 weighted via Boolean-IDF.

The detected number of clusters for the optimal configuration does not reflect the structure of the dataset. It is equal to 2, in the Boolean-IDF and LogTF-IDF case, since the relative silhouette values are the highest and the score function makes this solution the first in the ranking. The reason may lie behind the fact that the Boolean local weight has not high differentiation capability, thus, it is not able to cluster the dataset at a detailed level of description. A finer analysis is made by TF-IDF which identifies 4 cluster but this still does not reflect the original dataset categorization. Except for the best solution, the global and average silhouette tend to differentiate more in comparison with one-category dataset results. Considering the conversion factor for the weighted silhouette, which make the relative values comparable with the other indices it can be affirmed that the validity of the clustering process is still supported by them. The trend of global silhouette is "regularly" decreasing with the increasing of the number of clusters while the average silhouette makes a big hope between five and six clusters (see Fig. 39) suggesting that the previous configurations are much better than the followers.

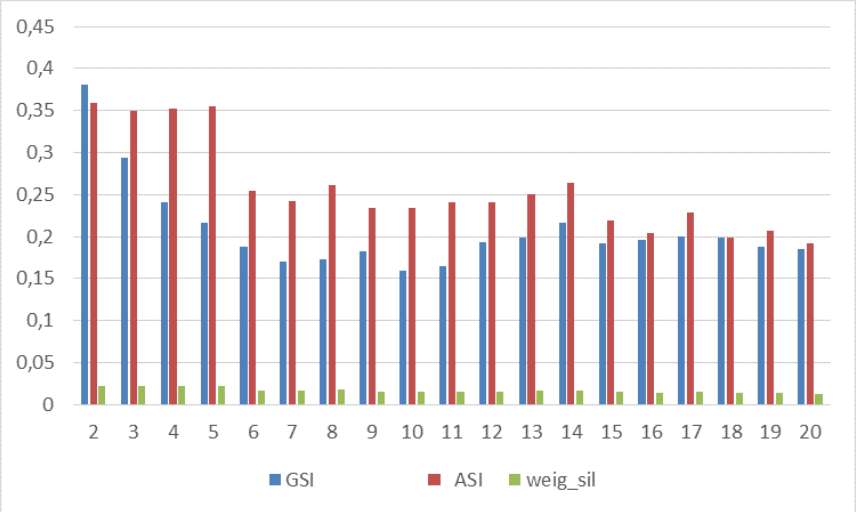


Figure 39: Plot of silhouette based indices for D-M1 weighted by Boolean_IDF

ESCAPE provides similar results for the dataset made by reviews from 6 different products category. The number of dimensions to explain the dataset, for the best configuration, is relative high as a consequence of the fact that the data are more variable but ,unlike M-D1, the identified number of clusters does reflect the structure of the dataset. It is in fact equal to 6 for both Boolan-IDF and LogTF-IDF. TF-IDF detects seven clusters (i.e topics) in contrast with the input dataset structure but this result has also the lowest silhouette values mining that it does not model the corpus in the best way.

Mix-Dataset2					
Weights	K_Lsa	k_Cl	GSI	ASI	W_SIHL
Bool-IDF	14	6	0,273342223	0,320018559	0,014701912
LogTF-IDF	14	6	0,28884	0,36481	0,020046
TF-IDF	13	7	0,2541	0, 3321	0.02071

Table 22: Experimental results for the dataset mix 2 for the Joint approach

Both the MD-1 and MD-2 optimal results are the same for the Boolean-IDF and LogTF-IDF weighting schemas but the results obtained with the boolean weight are characterised by a number of clusters decreasing with the increasing of the number of dimensions while the trend is inverse in the logarithmic and normal term frequency case.

Six original dataset has been included also in the third mix dataset but this time comments of similar products have been broguht together in a novel dataset to strongly test ESCAPE Topic detection and clustering ability. The resulting best configuration are reported in the Tab. 23.

Mix-Dataset3					
Weights	K_LSA	k_Cl	GSI	ASI	W_SIHL
Bool-IDF	7	4	0,3778	0,2984	0,03012
LogTF-IDF	8	5	0,3971	0,3478	0,02991
TF-IDF	8	5	0,3999	0,3446	0,02943

Table 23: Experimental results for the dataset mix 3 for the Joint approach

The number of dimensions identified for the best configuration is lower in comparison with M-D2 as a predictable consequence of the fact that the data should be less variable. The partition with the highest score-function value, i.e the optimal on, is made, in contrast with the original structure, of 5 or 4 clusters. For all the weighting strategy and especially for TF-IDF the behaviour of the global silhouette indices strongly highlights the best partitioning: it grows and then has a peak in correspondence of the optimal clustering value and then decreases while the average has a more levelled behaviour. (Fig.40)

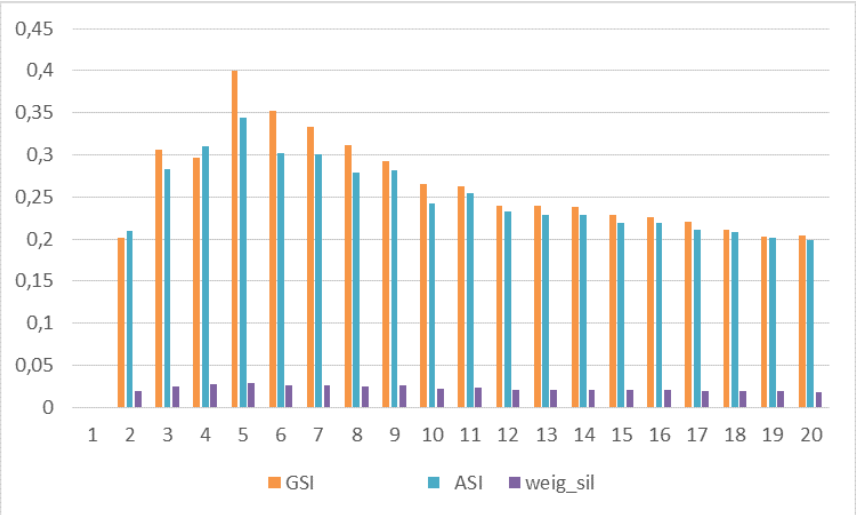


Figure 40: Sihlouette based indices for mix dataset 3 weighted by TF-IDF

Escape has shown low performance when used for topic detection and clustering on M-D4, the dataset made of reviews from ten product categories comments datasets. As shown in the table below, which reports also the three K-lsa, on average the number of dimensions to explain the dataset is higher then the original datasets results but the data are clustered, in the optimal solution, in a few clusters (at least four). The average and global silhouette have not low values suggesting that the corpus is well modelled but an a-priori knowledge of the number of clusters is available so ESCAPE has probably identified groups not basing on the cateogory of the products. What is missed in almost all the dataset is the ability of ESCAPE to provide an analysis at different level of granularities because the number of cluster detected is really similar among the weighting strategies.

DM-4					
Weights	K_LSA	k_Cl	GSI	ASI	W_SIHL
Boolean-IDF	**6**	**4**	**0,2981**	**0,3333**	**0,0166**
	15	9	0,2503	0,1868	0,0104
	29	20	0,2256	0,1744	0,0097
LogTF-IDF	**6**	**3**	**0,2797**	**0,3926**	**0,0199**
	15	9	0,2456	0,18367	0,0103
	29	13	0,2159	0,1569	0,0091
TF-IDF	**6**	**4**	**0,3034**	**0,3331**	**0,0173**
	15	9	0,2509	0,1855	0,0104
	29	20	0,2281	0,1777	0,0100

Table 24: Experimental results for the dataset mix 4 for the Joint approach

In order to better investigate the weight impact on the results and make a comparison between the found partitions, the ARI index for each cuople of weighting schemas and the clusters cardinality are computed for the best configurations.

M-D1	B-IDF	LogTF-IDF	TF-IDF
Cl1	79413	54906	43111
Cl2	68443	92950	70147
Cl3			34598
Cl4			13221

Table 25: M-D1: Cardinality of each cluster set found for the joint approach.

D14	B-IDF	LogTF-IDF	TF-IDF
Cl1	66545	65842	65732
Cl2	24463	26841	21932
Cl3	21423	20205	15022
Cl4	27214	21479	24400
Cl5	3658	10254	11562
Cl6	61582	60264	46841
Cl7			19396

Table 26: M-D2: Cardinality of each cluster set found for the joint approach.

M-D3	B-IDF	LogTF-IDF	TF-IDF
Cl1	9041	1020	27905
Cl2	44478	46589	19733
Cl3	26439	20201	45745
Cl4	19693	25847	12912
Cl5	6644	12638	

Table 27: M-D3: Cardinality of each cluster set found for the joint approach.

M-D4	B-IDF	LogTF-IDF	TF-IDF
Cl1	11077	27828	10458
Cl2	28741	123370	32014
Cl3	30848	31072	100254
Cl4	111604		39544

Table 28: M-D4: Cardinality of each cluster set found for the joint approach.

M-D1	B-IDF	TF-IDF	LogTF-IDF
B-IDF		0.6696	0.8918
TF-IDF			0.6045

Table 29: M-D1: ARI index for joint approach.

M-D2	B-IDF	TF-IDF	LogTF-IDF
B-IDF		0.7971	0.9816
TF-IDF			0.8103

Table 30: M-D2: ARI index for joint approach.

M-D3	B-IDF	TF-IDF	LogTF-IDF
B-IDF		0.8726	0.8012
TF-IDF			0.9472

Table 31: M-D3: ARI index for joint approach.

M-D4	B-IDF	TF-IDF	LogTF-IDF
B-IDF		0.7554	0.883
TF-IDF			0.7671

Table 32: M-D4: ARI index for joint approach.

The ARI indexes have in general very high values suggesting a strong agreement between all the partitions, also if they do not always have the same number of clusters. The lowest values can be found for D-M1 (composed by reviews from 3 product categories) where solutions obtained with TF-IDF are more different from the others. This could be predictable since

TF-IDF categorization has a different number of clusters but, also if two solution have the same name they can have lower ARI values meaning that the detected topics are different for the same dataset.
The high ARI values fo M-D1 are a confirmation of the fact that the two major cluster identified by ESCAPE with Bool-IDF and Log-IDF are recognizable also in the categorization obtained with TF-IDF. Also for D-M2 two main cluster are identified with all the weighting strategy. After will be investigate if they correspond to the categories with the major number of reviews included in the novel datasets. For M-D3 and M-D4 one major cluster can be always identified, in the first case it could correspond to two product categories jointed by ESCAPE.

3.5.3 Visualization

The weight impact can also be graphically and quick shown through correlation matrices. Recalling that dark areas represent higher level of correlation, it is interesting to see that, for M-D1 dataset partition with Bool-IDF, the relationship between documents in a cluster are much stronger than the documents in the other one (see Fig.41). The reason is that in one cluster ESCAPE have brought together comments from different categories which are less related to each other. However clusters are well identified; this is not valid for the 5 clusters detected by LogTF-IDF and TF-IDF for D-M3: one cluster is very small as already showed from the cardinality computation, the relatioship between the documents belonging to the same cluster are weaker and ther are some others dark areas, especially in TF case, which represented subtopics(see Fig. 42).

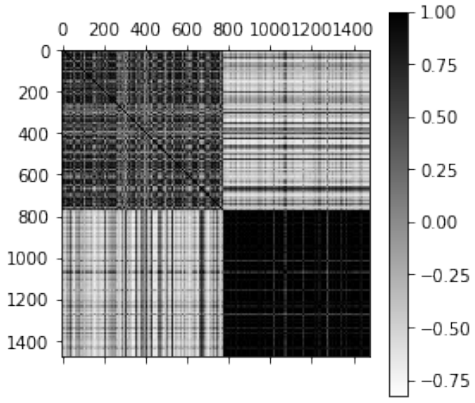


Figure 41: M-D1: Correlation map

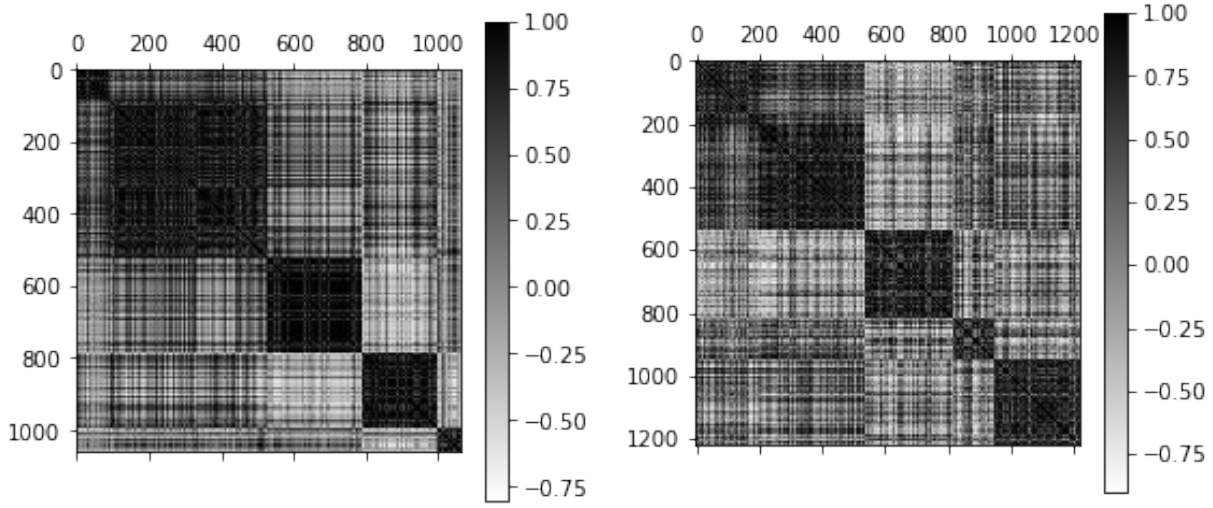


Figure 42: Dataset M-D3. correlation matrix LogTF-IDF(left) TF-IDF(right) K=5

T-sne maps are mostly points clouds, points colouring is not so balanced meaning that the distribution of the documents is not equal among the clusters: for example there is one little red area in the M-D3 tsn-representations. They provide also a lightly particular shape, similar for the different weighting schemas. The colouring unbalancing is instead attenuated for M-D1.

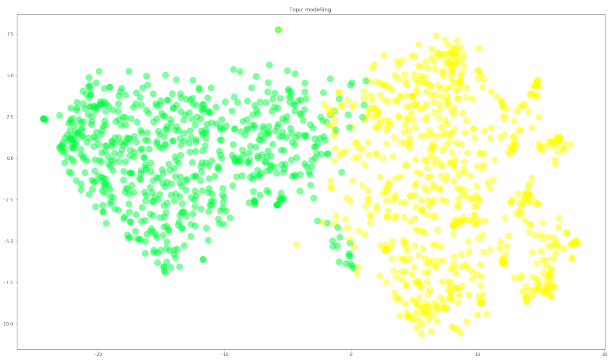


Figure 43: Dataset M-D3. t-sne map Bool-IDF K=2

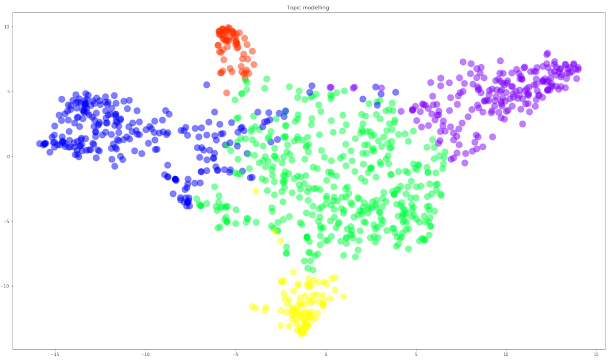


Figure 44: Dataset M-D3. t-sne map TF-IDF K=5

As already suggested, the Word clouds for M-D1 (Fig.45) shows that in one cluster are grouped comments about products belonging to the luggage and major appliances categories and in the other reviews of gift card, in this category the topic "holiday" is clearly visible and thus it includes also some term from the luggage dataset. The strong relationship showed in the correlation matrix between the documents within the cluster are confirmed since they are from the same original dataset.

Figure 45: Dataset M-D1. WordClouds

For M-D2 ESCAPE have perfect detected the original structure of the dataset, in fact the number of identified clusters corresponds to the number of products categories brought together in the novel dataset. Nevertheless it could be possible that ESCAPE have identified a different partition within the collection of documents. This issue is overcome looking at the the Word clouds representations (Fig.46, Fig.47): each most frequent item-set displayed in the clouds can be clearly link to a product category.

Figure 46: M-D2:WordCloud representations

Figure 47: M-D2:WordCloud representations

3.5.4 Probabilistic approach

The novel mix dataset does not have high values for the average length of the documents (i.e. reviews), the maximum is equal to 24 words on average for the dataset created with reviews about product of similar categories. As already said, this feature is important to make the LDA work proper, basing on it the probabilist approach has been applied to the different dataset in descending order of average documents length and the results are reported following this order in Tab. 33

The five parameters have been set in the same way of the previous experiments with the original dataset. The upper bound for the number of cluster, necessary for the Topic Similarity strategy, is set equal to the average length of the documents in the corpus or to the average frequency of the terms. The maximum number of iteration within the model has to converge is set to 100. The Dirchelet distribution used to draw the per-document distribution and estimate the LDA model is computed setting the optimizer to be the Online Variational Bayes. α and β, respectively, the document and topic concentration are set to maximize the log-likelihood of the data under analysis to be equal or major then 0. The default value for α is set equal to $\frac{50}{K}$, while for β it is set equal to 0,1.

spacing

Dataset	Weights	K_CL	Perplexity	Log-likelihood
M-D3	Bool-IDF	2	8,11051	-160905465
		4	7,89945	-156718273
		13	**7,71579**	**-153074464**
	LogTF-IDF	2	8,04629	-145412601
		4	7,97401	-144106248
		16	**7,76695**	**-140364339**
	TF-IDF	5	7,87256	-159413306
		13	7,73157	-153387607
		16	**7,69199**	**-152602400**
M-D4	Bool-IDF	2	8,0176	-237695780
		4	**7,9313**	**-235137146**
	LogTF-IDF	2	8,0016	-237746112
		4	**7,8896**	**-233805169**
	TF-IDF	2	8,0193	-237894496
		5	**7,8515**	-229993495
M-D1	Bool-IDF	2	7,3105	-136258417,2
		8	7,0599	-135882668,4
		13	**6,9635**	**-134599963,4**
	LogTF-IDF	2	7,4210	-135248988
	TF-IDF	2	7,3950	-132078412

Table 33: Experimental results for mix datasets for the probabilistic approach

For not all the datasets ESCAPE has detected the maximum possible number of proper K values for the clustering analysis, meaning that it has not found 3 values for satisfying the two conditions imposed considering the Topic-Similarity function.

Recall that M-D3 is made by 6 categories, M-D4 is made by 10 categories and M-D1 by 3 categories; for any of the datasets ESCAPE has not identified the value which reflects the original structure. Mostly, low values of k have been selected but, the lowest, thus the best, values for perplexity always correspond to the maximum identified k number. TF-IDF has selected the highest values for k for the dataset made of 6 similar categories (M-D3), this is true also for M-D4 but in a much less evident way. As predictable, ESCAPE has performed better on M-D3 with respect to the other dataset, providing 3 good values for the number of clusters fo all the weighting schema; M-D3 is indeed the dataset with the longest review (on average) however, it has to be highlighted that the perplexity has the lowest value for

the results of the dataset with the shortest reviews.

the ARI indices showed in the tables below demonstrate a weak agreement between the partitions found by the different weighting strategies. Also if, for M-D3, LogTF-IDF and TF-IDF have found the same number of clusters the index value is not high meaning that ESCAPE has detected different categorization for same dataset using the same number of clusters.This is true also for M-D4 with B-IDF and LogTF-IDF schemas. The only ARI value major than 0,5 is for the pair LogTF-IDF,TF-IDF for M-D1 which have detected the same number of clusters and also have similar perplexity values.

M-D3	B-IDF	TF-IDF	LogTF-IDF
B-IDF		0.3274	0.4068
TF-IDF			0.4068

Table 34: M-D3: ARI index for probabilistic approach.

M-D4	B-IDF	TF-IDF	LogTF-IDF
B-IDF		0.1594	0.2060
TF-IDF			0.377

Table 35: M-D4: ARI index for joint approach.

M-D1	B-IDF	TF-IDF	LogTF-IDF
B-IDF		0.211	0.1668
TF-IDF			0.674

Table 36: M-D4: ARI index for joint approach.

3.5.5 Visualization

From the t-sne maps it is possible to see the document distribution among the clusters thanks to the colouring of the points. For M-D3, through the t-sne maps, it is possible to clearly see six main clusters which also correspond to the number of categories included in the novel dataset, i.e the topics, and some other subtopics, some of them very confused.(Fig.48) For MD-4, the t-sne representation shows an unbalancing with one cluster to which belongs many more documents with respect to the others. (Fig.49)

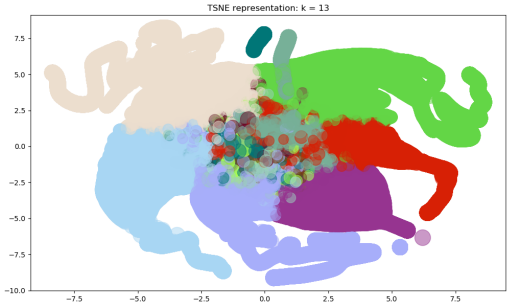


Figure 48: Dataset M-D3. t-sne map Bool-IDF K=13

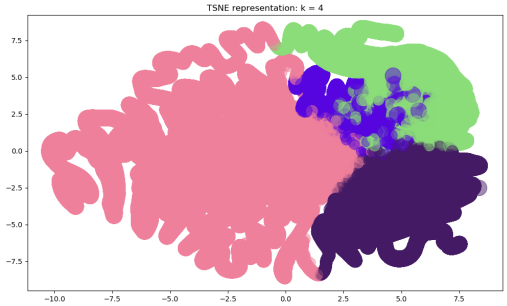


Figure 49: Dataset M-D4. tsne-map TF-IDF K=4

The relationship between the Topics and the relative representative words are represented by un-orientated graphs. Also if quite confused the graph for the best configuration for M-D3 with Boolean_IDF is reported to show that topics (clusters) can be individuated also if they share some words with other clusters. This means that if a set words are selected to describe a topics, some of them will be present also in other topics so attention has to be paid to selct a proper number of relevant words. Also for M-D4 solutions, characterized by a lower number of clusters, the graph clearly reveal the strongest link between a topic and the relative words but also the sharing of a word between more than one topic.

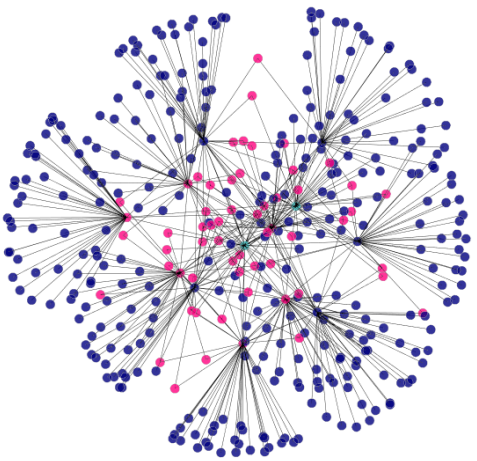


Figure 50: Dataset M-D3. Graph representation. Bool-IDF weighting schema K=13

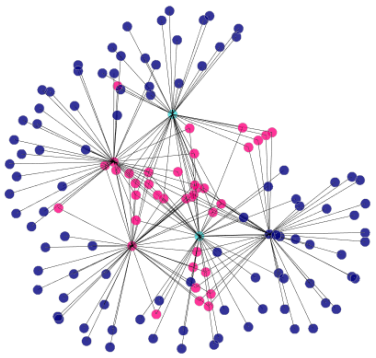
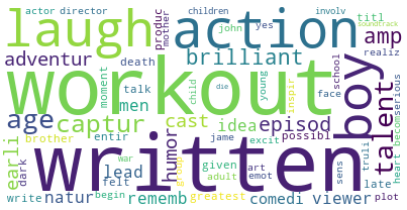


Figure 51: Dataset M-D4. Graph representation. TF -IDF weighting schema K=5

For M-D3 the word clouds representations show an interesting result: three out of four detected clusters with LogTF-IDF have the most frequent item set composed by terms which describe on topic of the original datasets while one cluster contain words which mainly refer to payment and site profile management.



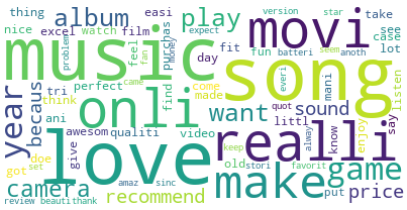
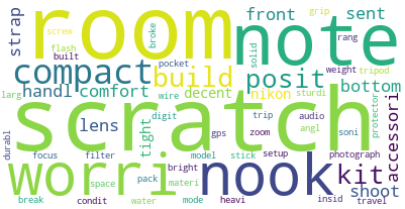


Figure 52: M-D3:WordCloud representations



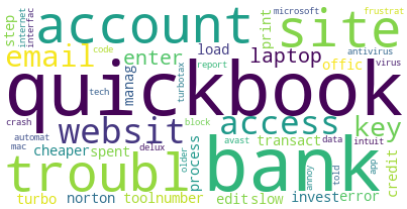


Figure 53: M-D3:WordCloud representations

4 Conclusions

The realized work have been developed in the context of textual data mining, it started from the doctoral project of Evelina di Corso with a dual purpose: on one hand studying, testing and tailoring the data driven technology ESCAPE on a particular type of textual data, specifically e-commerce reviews; on the other hand bring out from them potentially high-quality and previously unknown knowledge which can be useful for business analysis. The data for the analysis have been selected with the express purpose of having many real dataset which can be actually available and beneficial in some way for different kind of users, and which can be a proper set for validating the proposed methodology. The structure and source of the data, i.e AWS, have been deeply investigate since it is interesting how Amazon is proposing itself, through this platform, as a solution for many issue of data mining processes.

The proposed engine address all the blocks of the knowledge data discovery process and integrates two strategies for the exploratory phase. As hypothesized before running the experiments, it can be affirmed that the joint approach perform much more better on short text than the probabilistic approach since it is often unfeasible for the LDA model to infer a hidden structure in documents made by a few words. It has to be highlighted that the ability of ESCAPE to provide an analysis at different level of granularities has been a bit missed in the sense that the differences between the results obtained using the different weighting strategies are not so remarkable. Actually the local weight impact is higher and more clear for long documents where the language used is more rich and complex. Comments are instead written using a poor vocabulary: usually people does not use the same or a relevant word more than one time in a comment and this is reflected in the low values of TTR and Giraud Indices (which describe the language complexity and variety). The consequence is that local weighting factor LogTF is often equal to TF which can assume, in many case, only two values: 1 if a term is used 0 otherwise as for the boolean factor. Some novel methodologies to improve the process could be integrated, such as Sparse PCA. It is a reformulation of Principal component analysis to find a low-rank

approximation of a Salton matrix (by finding directions of greatest variance) alongside a penalty for non-zero model parameter values. In this way the variance explained using only the top few components is much higher then the normal PCA.

After the comparison of the joint and probabilistic approach for the original datasets, novel datasets with a knowledge of the labelling have been created to better validate ESCAPE abilities. Also this time the joint approach have performed better but it has not always been able to detect the expected number of categories. Only for the novel dataset with comments about 6 different products categories, the joint approach integrated in ESCAPE succeeds in identifying the exact number of cluster which reflects the corpus structure. The reason lays behind the fact that words which characterised the topics seem to be more separated. The low performance for the dataset composed by ten categories are, instead, a matter of topic coherence: also if the number of original label was equal to 10, the words used within the documents, do not allow to actually detect that number of topics.

The visualization techniques has been mined in order to well fit and display the obtained results. In particular for the word clouds a dictionary for the most common words used in the collection have been created and removed from each cluster in order to show the relevant and distinctive words. This is a step which could have been involved in the preprocessing phase in order to make more effective the follower phases of the analysis.

From a business and strategical point of you, some interesting features from the comments analysis have come up. From the first characterization of all the datasets, it has been clear that reviews about technological product are written with a more complex language, they are also the longer (on average) suggesting that this kind of consumer pay more attention on the proper and the others opinions when are buying something. It has not always been possible to understand the argument each cluster went through. Nevertheless, in some dataset, clusters mostly related to a specific product have been detected (for example the Garmin nuvi GPS), with also some relative "ancillary" or associated products which can be identified as complementary or substitute products. Exploiting these relationships the mix of products users buy may be alter and significant increase in purchase volume can be driven. Moreover it is interesting that there are some brands which appear as distinctive

words for the description of a topic,(Es. Bosh, Canon). By matching this information with the metadata that, in the current work, has been removed before applying the algorithm, the analysis can be further improved.

List of Figures

List of Tables

References

[1] Evelina Di corso, Text miner's little helper: scalable self-tuning methodologies for knowledge exploration. *Politecnico di Torino*, *2019-07-01*

[2] Han, Kamber. Data mining: concepts and techniques. Morgan Kaufmann, *2006*

[3] Tan, Steinbach, Kumar. Introduction to data mining. Pearson, *2006*

[4] Charu C. Aggarwal, ChengXiang Zhai. Mining Text Data. Springer.

[5] Jonathan G. Fiscus and George R. Doddington. Topic detection and tracking evaluation overview. In *Topic detection and tracking*, James Allan (Ed.). Kluwer Academic Publishers, Norwell, MA, USA 17-31.*2002.*

[6] David Reinsel, John Gantz, John Rydning. The Digitization of the World: from Edge to Core. In *IDC White Paper US44413318*, *November 2018*

[7] Abdul-Aziz Rashid Al-Azmi. Data, Text and web mining for business intelligence: a survey, *International Journal of Data Mining and Knowledge Management Process (IJDKP) Vol.3, No.2, March 2013*

[8] Charu Aggarwal and Chengxiang Zhai. A survey of text clustering algorithms. *Mining Text Data, 08 2012.*

[9] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, *2004*

[10] Evelina Di Corso, Tania Cerquitelli, and Francesco Ventura. Self-tuning techniques for large scale cluster analysis on textual data collections. In *Proceedings of the Symposium on Applied Computing* pages 771–776. ACM,*2017*

[11] Evelina Di Corso, Francesco Ventura, and Tania Cerquitelli. All in a twitter: Self-tuning strategies for a deeper understanding of a crisis tweet collection. In *IEEE BigData 2017Boston, MA, USA*[6], pages 3722–3726.

[12] Stefano Proto, Evelina Di Corso, Francesco Ventura, and Tania Cerquitelli. Useful topic: Self-tuning strategies to enhance latent dirichlet allocation. In 2018 *IEEE International Congress on Big Data (BigData Congress)* pages 33–40. IEEE, *2018*

[13] Tania Cerquitelli, Evelina Di Corso, Francesco Ventura, and Silvia Chiusano. Prompting the data transformation activities for cluster analysis on collections of documents. In *Proceedings of SEBD 2017* pages 226–234, *2017*

[14] Evelina Di Corso, Stefano Proto, Tania Cerquitelli, and Silvia Chiusano. Towards automated visualisation of scientific literature. In European Conference on Advances in *Databases and Information Systems*Springer, *2019*

[15] https://blog.aboutamazon.com/company-news/2018-letter-to-shareholders

[16] Samuel Fosso Wamba, Shahriar Akter, Andrew Edwards, Geoffrey Chopin, and Denis Gnanzou. How 'big data'can make big impact: Findings from a systematic review and a longitudinal case study *International Journal of Production Economics* 165:234–246, *2015*

[17] https://aws.amazon.com/it/blogs/aws

[18] *Python Documentation*

Acknowlegments

If I look back at the end of these intense years, the most beautiful thing I see are the people I met. Each of them have contributed in some way to the thesis project and have given me support during all this period. I wish to say thank you to my special mates for this journey, Adelia, Camilla and Chiara and to the Borsellino family (with a special mention for Gigi) for having made me feel at home, for giving me advice also when it was hard to hear and for lightning me up whenever I was down.

A special thank goes also to Evelina, my guardian angel during the development of the thesis project, for the patience and the kind help and to the supervisor, prof. Cerquitelli, for providing me with a challenging and stimulating work.

I cannot forget and thank people who have always been by my side. My sisters by blood, Vittoria and Ludovica and not by blood, Marghe and Alli because I can always feel their presence also if we are far away. Last, but definitely not least, I would like to say thank you to mum, my every day model of kindness and courage, and dad, my inspiration for never giving up.