View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by AIR Universita degli studi di Milano

Exploratory Analysis of Textual Data Streams

Silvana Castano[a], Alfio Ferrara[a,*], Stefano Montanelli[a]

[a]*Department of Computer Science, Università degli Studi di Milano, via Comelico 39, 20135 Milan, Italy*

Abstract

In this paper, we address exploratory analysis of textual data streams and we propose a bootstrapping process based on a combination of keyword similarity and clustering techniques to: i) classify documents into fine-grained similarity clusters, based on keyword commonalities; ii) aggregate similar clusters into larger document collections sharing a richer, more user-prominent keyword set that we call *topic*; iii) assimilate newly extracted topics of current bootstrapping cycle with existing topics resulting from previous bootstrapping cycles, by linking similar topics of different time periods, if any, to highlight topic trends and evolution. An analysis framework is also defined enabling the topic-based exploration of the underlying textual data stream according to a thematic perspective and a temporal perspective. The bootstrapping process is evaluated on a real data stream of about 330.000 newspaper articles about politics published by the New York Times from Jan 1st 1900 to Dec 31st 2015.

Keywords: Clustering, Textual Data Stream, Exploratory Analysis, Topic Evolution, Detection of Emergent Topics

1. Introduction

In many information repositories available over the Internet (e.g., web sites, newsgroups, blogs, social networks, forums), data are accessible as a continuous stream of textual information. The web is continuously updated with new pages, we continuously receive new emails, our Twitter timeline, Facebook profile, or blog is continuously updated with new posts. Users frequently consult their favorite information sources, either for seeking news from newspapers or for staying informed about friends postings on the social media. Available data are not only big and rich, but also dynamic, and the capability to deal with time becomes a crucial requirement for the analysis of this continuous flow of information [1, 2]. On the one side, users need to perform a thematic, exploratory analysis of the underlying document flow, driven by representative, significant

*Corresponding author
Email addresses: silvana.castano@unimi.it (Silvana Castano), alfio.ferrara@unimi.it (Alfio Ferrara), stefano.montanelli@unimi.it (Stefano Montanelli)
URL: http://islab.di.unimi.it/castano (Silvana Castano), http://islab.di.unimi.it/ferrara (Alfio Ferrara), http://islab.di.unimi.it/montanelli (Stefano Montanelli)

topics extracted from the information flow itself [3, 4]. On the other side, featuring topics must be correctly located in the timeline, to easily get fresh and emergent topics and to study and understand topic evolution along time [5, 6]. Classification and analysis techniques are thus required capable of working in an incremental fashion on textual data streams in order to discover new topics as long as they emerge from incoming documents, and to capture their degree of specificity and popularity with respect to the time period of the documents they refer to.

In this paper, we envisage the exploratory analysis of textual data streams as a continuous bootstrapping process, where each bootstrapping cycle works on an incoming document chunk of the stream related to a fixed-size time window. Each incoming document is acquired and indexed to extract a representative keyword-set from its textual content to be used for bootstrapping. The proposed bootstrapping process is based on a combination of keyword similarity and clustering techniques to: i) classify documents into fine-grained similarity clusters, based on keyword commonalities; ii) aggregate similar clusters into larger document collections sharing a richer, more user-prominent keyword set that we call *topic*; iii) assimilate newly extracted topics of current bootstrapping cycle with existing topics resulting from previous bootstrapping cycles, by linking similar topics of different time windows, if any, to highlight topic trends and evolution. An analysis framework is also defined enabling the topic-based exploration of the underlying textual data stream according to a thematic perspective and a temporal perspective. Given a topic of interest T produced in a time window W, the thematic analysis perspective focuses on W and exploits the topic keywords to perform an "in-depth analysis" of T and its related document collection, by highlighting the most prominent/ highly-correlated keywords of T and, possibly, of other topics T' in the same W. The temporal analysis perspective reconstructs the trend of T by showing how the keyword-set of T evolves in time, to capture the "variants" and "invariants" keyword portions as well as the level of popularity and specificity of T in the different time periods of the trend. Temporal analysis is also useful to discriminate between different kinds of topics, such as *persistent topics*, which are always present in the document flow, with some modifications of keyword-set across different periods of time to reflect the variations of the arguments/perspectives in the underlying documents, and *spot topics*, which are bound to a limited and well defined time period. The proposed bootstrapping process is evaluated on a real data stream of about 330.000 newspaper articles about politics published by the New York Times from Jan 1st 1900 to Dec 31st 2015.

The paper is organized as follows. In Section 2, we present the related work. In Section 3, we introduce the proposed approach for exploratory analysis of textual data streams. In Section 4, 5, and 6, we describe the document clustering task, the topic discovery task, and the topic assimilation task, respectively. In Section 7, we discuss the analysis framework and the operators for enabling thematic and temporal topic analysis and exploration as well as application issues of our approach. Experimental evaluation is presented in Section 8. Finally, Section 9 provides our concluding remarks.

2. Related work

Work related to the exploratory analysis of textual data streams falls in three main categories, namely *Text Stream Classification (TSC)*, *Topic Detection and Tracking (TDT)*, and *Topic Modeling (TM)*.

In the following, contributions of each category will be classified according to the following criteria: *Topic Discovery*, to denote the technique used for topic recognition; *Topic Assignment*, to denote whether a document is assigned to only one topic, i.e., hard assignment, or to more than one topic, i.e., soft assignment; *Document Modeling*, to denote whether the approach models single-topic documents, or multi-topic documents; *Stream Management*, to denote whether approaches exploit a time window for discretizing the textual data stream or whether they produce a unique topic set which is updated along time; *Experimentation Dataset*, to denote the main dataset used for evaluation purposes; *Topic Tracking*, to denote, when applicable, the kind of technique employed for topic tracking, by distinguishing between techniques based on the structure and contents of topics and techniques based on the terminology used in topic labels.

In Sections 2.1 to 2.3, we first survey the related work of each category by also providing the classification of our approach in the righthand side of each summary table; then we provide a comprehensive description of the original contribution of our work in Section 2.4.

2.1. Text stream classification

The aim of text stream classification is to study the stream clustering problem for evolving data sets, by examining the behavior of document clusters over different time horizons. A summary of contributions in this field is shown in Table 1. A common characteristic of these approaches is to extend hard clus-

Table 1: Summary of contributions on TSC

Reference	**Aggarwal et al. [1]**	**Ghosh et al. [7]**	**Liu et al. [8]**	**Zhong [9]**	**Our approach**
Topic Discovery	Clustering	Clustering	Clustering	Clustering	Clustering
Topic Assignment	Hard	Hard	Hard	Hard	Soft
Document Modeling	Single-topic	Multi-topic	Single-topic	Single-topic	Multi-topic
Stream Mgmt.	Discretized	Continuous	Continuous	Discretized	Discretized
Experim. Dataset	*Yahoo!*	*Facebook*	20ng-news	20ng-news	NYT
Topic Tracking	Cluster	Terminology	-	-	Terminology

tering algorithms to the problem of efficiently processing dynamic text data streams in order to classify documents into topics. Document models are usually single-topic because the aim of these approaches is to discover the most prominent topic within each document for classification purposes. An exception is [7], where the authors deal with Facebook conversations by modeling their multi-topic nature through a fuzzy model and subsequently classifying them through hard spectral clustering. Concerning the stream management, a fundamental distinction is between discretized approaches that extract different

sets of topics for different time periods [1, 9], usually exploiting a *time window*, and continuous approaches that produce a unique topic set that changes along time [7, 8]. Only [1, 7] address the problem of tracking the evolution of topics in time, although not providing a notion of topic trend. In [1] topic tracking is based on the analysis of cluster contents and size, while in [7] topic tracking is based on the analysis of term frequency.

2.2. Topic detection and tracking

The goal of topic detection and tracking (TDT) is to recognize events/stories in a stream of information usually based on broadcast news [10]. Representative TDT works are summarized in Table 2.

Table 2: Summary of contributions on TDT

Reference	**Nguyen et al. [11]**	**Gaul et al. [5]**	**Kaleel et al. [4]**	**AlSumait et al. [6]**	**Our approach**
Topic Discovery	-	Clustering	Clustering	Modeling	Clustering
Topic Assignment	-	Hard	Hard	Soft	Soft
Document Modeling	Multi-topic	Single-topic	Single-topic	Multi-topic	Multi-topic
Stream Mgmt.	Discretized	Discretized	Discretized	Continuous	Discretized
Experim. Dataset	Patents	*Der Spiegel*	*Twitter*	*Reuters*	NYT
Topic Tracking	Terminology	Cluster	Cluster	Terminology	Terminology

TDT approaches are mainly characterized by the use of hard clustering algorithms [4, 5] for topic detection. Sometimes, the use of soft-clustering algorithms is also proposed, especially when multi-topic documents are considered (e.g., newswire and multimedia contents), while in other cases [11] topics are tracked by working directly on document terminology rather than on document clusters (further examples are provided in [2, 12]). In [6], an online extension of LDA is presented based on the incremental update of an initial classification model, rather than on multiple window-based models. In TDT, topic tracking, also known as trend discovery, is enforced by calculating the degree of similarity among clusters emerged in different time instants/windows, so that trends can be tracked by observing size changes on similar clusters (i.e., cluster-based approaches) [4, 5, 13]. As an alternative solution, topic tracking can be enforced by observing changes on term frequencies over time (i.e., term-based approaches) [6, 11].

2.3. Topic modeling

Several contributions have been provided in the category of topic modeling, starting from Latent Semantic Analysis (LSA) [14] back in the '90s to more recent Latent Dirichlet Allocation (LDA) [15] in the 2000s. A summary of main topic modeling-based approaches for topic discovery in textual data streams is given in Table 3. We recall that topic modeling are based on the idea that topics are latent variables that explain the distribution of words in documents and provide statistical techniques for finding the most representative variables given the data. For this reason all the contributions provide soft assignment of documents to topics. In [16], a solution for efficiently inferring topic models

Table 3: Summary of contributions on topic modeling

Reference	**Yao et al. [16]**	**Hong et al. [17]**	**Hoffmann et al. [18]**	**Wang et al. [3]**	**Our approach**
Topic Discovery	Modeling	Modeling	Modeling	Modeling	Clustering
Topic Assignment	Soft	Soft	Soft	Soft	Soft
Document Modeling	Multi-topic	Single-topic	Multi-topic	Multi-topic	Multi-topic
Stream Mgmt.	Discretized	Discretized	Continuous	Continuous	Discretized
Experim. Dataset	*Pubmed*	*Twitter*	*Wikipedia*	NIPS Conf.	NYT
Topic Tracking	-	Terminology	-	-	Terminology

from document streams is proposed. The proposed method generates a unique topic model for the whole data stream, by updating the topics according to the new documents acquired in subsequent time windows. A similar work has been proposed also in [18], where an online learning algorithm for LDA is described. Also in this case, the algorithm works incrementally, but the topics are not forced to be specifically derived from a given time window. In [17], a model is proposed to extend topic models by allowing each text stream to have local topics and shared topics. It is also the only approach that addresses the goal of tracking topic evolution. A similar approach, called TOT (Topics Over Time), but not based on a discretization of time intervals, is presented in [3].

2.4. Contribution of our work

With respect to the related work previously discussed, our approach provides a single, comprehensive solution where the peculiar features of topic modeling techniques (i.e., soft-assignment of documents to topics and topic labeling) are combined with topic tracking and trend modeling functionalities that are typical of TDT approaches. The main contribution of our approach can be summarized as follows:

1. *Capability of modeling multi-topic documents in the clustering process.* We propose a bootstrapping process based on clustering techniques with specific extensions to enforce soft-assignment of documents to clusters in order to model multi-topic documents in textual data streams. This is motivated by the fact that, in our work, topic discovery is finalized to exploratory analysis rather than to document classification. This is a distinguishing feature with respect to related work on text classification and TDT which are generally based on hard clustering techniques and finalized to document classification.
2. *Capability of automatically labeling topics out of clustered documents.* We propose clustering techniques where topic labels are derived in an automated way from cluster contents, since labeling is intrinsically part of the clustering process itself. This is a further distinguishing feature of our work in that cluster labeling in related work approaches usually requires an additional processing step for analyzing the contents of the created clusters.
3. *Capability of deriving topic trends based on terminological similarity techniques for topic matching.* We propose techniques based on similarity

and specificity metrics for matching and linking similar topics of different bootstrapping cycles, to highlight topic trends and evolution. Topic tracking is poorly addressed by topic modeling and text stream classification approaches, while it is more studied in the topic detection and tracking field. In this respect, we observe that our approach can be classified as terminology-based, and it has been conceived to produce topic links for enabling trend exploration and topic evolution along time.

3. Bootstrapping process of textual data streams

We call **textual data stream** $S = \langle (d_0, t_0), (d_1, t_1), \ldots \rangle$ a continuous flow of documents coming from one or more considered documental datasources, where the pair (d_j, t_j) denotes that the incoming document d_j is acquired at time t_j. We call **time window** $\mathcal{W}_i$ a time interval of fixed size δ starting at time t_i.

The proposed approach on exploratory analysis of textual data streams is characterized by the execution of a new, independent bootstrapping cycle $\mathcal{B}_i$ for each time window $\mathcal{W}_i$. Each $\mathcal{B}_i$ is in charge of processing a **document chunk** $\mathcal{D}_i$, that is the portion of the data stream S containing the documents incoming in $\mathcal{W}_i$, namely $\mathcal{D}_i = \{d_j \mid j \in [i, i + \delta]\}$ (Figure 1).

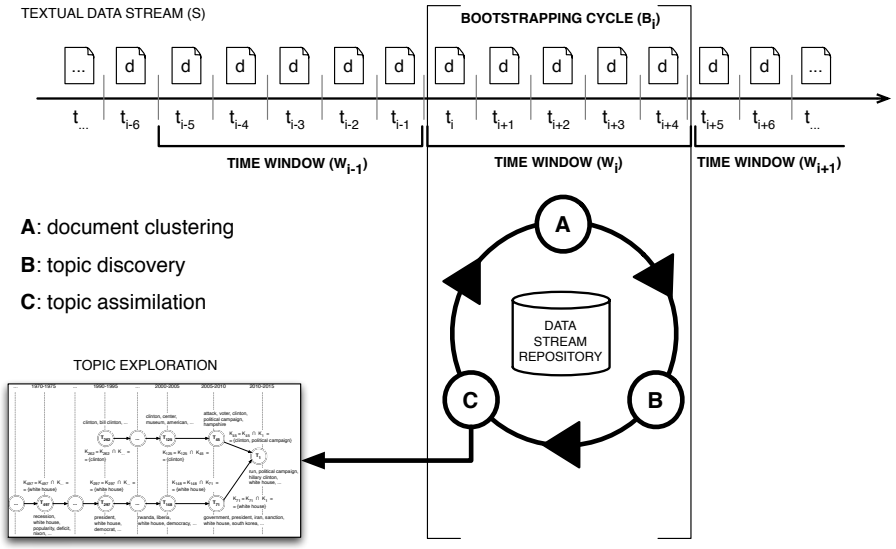


Figure 1: Bootstrapping of a textual data stream

For each document $d_j \in \mathcal{D}_i$, the acquired textual content (e.g., the text of a tweet, the title and the abstract of a PDF document, the content of a web page) is stored in a Data Stream Repository along with the corresponding timestamp t_j. Each document $d_j \in \mathcal{D}_i$ is then associated with a keyword set K_{d_j} extracted through the execution of a conventional text tokenization and normalization procedure[1].

[1]Conventional techniques are applied in the procedure, such as elision removal, lower case normalization, stop-word removal, and compound-term detection [19]).

A bootstrapping cycle $\mathcal{B}_i$ is triggered at the $\mathcal{W}_i$ times window and it consists in the sequential execution of three tasks called *document clustering*, *topic discovery*, and *topic assimilation*.

A. Document clustering. For the time window $\mathcal{W}_i$, the corresponding document chunk $\mathcal{D}_i$ is submitted to a clustering procedure aimed at categorizing the acquired documents into clusters based on the similarity of their corresponding keyword sets. The goal is to generate a set of clusters $\mathcal{C}_i$, where each document cluster $cl_k \in \mathcal{C}_i$ is usually small in size and densely-cohesive, meaning that a document contained in cl_k is pairwise similar to most of the other cluster elements and few keywords are enough to effectively describe the entire cluster contents since they are common to all the documents in the cluster (see Section 4).

B. Topic discovery. This task consists in applying a second round of clustering to the set of document clusters $\mathcal{C}_i$ to build a topic-based view of the underlying document chunk which is more prominent for the final user than $\mathcal{C}_i$. The goal is to generate a set of topics $\mathcal{T}_i$, where each topic $T_h \in \mathcal{T}_i$ is usually large in size and aggregates similar clusters into larger document collections sharing a richer, more effective keyword set (see Section 5).

C. Topic assimilation. Topic assimilation has the goal to correctly link *newly-emerged* topics (i.e., topics discovered in the current bootstrapping cycle $\mathcal{B}_i$) with topics discovered in the previous bootstrapping cycle $\mathcal{B}_{i-1}$. A new topic is linked with an existing one when it is recognized to be similar based on their keyword sets. Topic assimilation operations are defined on linked topics to recognize either a positive trend when the relevance of the newly-emerged topic increases, or a negative trend when its relevance decreases (see Section 6).

Example. As running and evaluation dataset, we consider a textual data stream called NYT composed of about 330.000 newspaper articles (i.e., documents) about politics published by the New York Times from Jan 1st 1900 to Dec 31st 2015[2]. In particular, we collected about 10 articles per day from 1900 to 2015 and we extracted keywords sets considering the headline and the abstract. A summary view of the NYT stream dataset is shown in Table 4. For each time window $\mathcal{W}_i$ of five years, we provide the number of acquired documents (i.e., document chunk size) and the corresponding number of extracted and distinct keywords, respectively.

3.1. Similarity evaluation

The overall bootstrapping cycle is based on the notion of similarity to be employed first for documents, then for clusters, and finally for topics. All the three tasks of the bootstrapping cycle work on indexed objects (i.e., documents, clusters, topics), namely objects associated with a keyword set providing an essential description of the object content. Given two objects x and y and the corresponding keyword sets K_x and K_y, we rely on keyword-based similarity

[2]Articles have been acquired through the New York Times Article Search API v2 (`http://developer.nytimes.com/docs/read/article_search_api_v2`) by executing the keyword search politics.

Table 4: The NYT stream dataset

Period	#Docs	#keywords	#Dist_keywords
1900-1905	8756	67668	3758
1905-1910	6331	65192	3858
1910-1915	8378	76311	3911
1915-1920	6528	85714	3909
1920-1925	9798	118967	4320
1925-1930	12865	189300	5036
1930-1935	17804	267567	5385
1935-1940	16510	213417	5197
1940-1945	11816	137708	4646
1945-1950	16290	203164	4905
1950-1955	17400	212803	5181
1955-1960	16905	218045	5180
1960-1965	15994	183195	5118
1965-1970	15722	201906	5440
1970-1975	13486	318664	6257
1975-1980	16757	346788	6550
1980-1985	16963	105820	4785
1985-1990	15967	51523	4042
1990-1995	16476	57993	4037
1995-2000	17186	281375	6434
2000-2005	17982	424841	7205
2005-2010	17615	153832	5745
2010-2015	17952	148572	5565
Total	331481	4130365	12964

measures. In particular, we use the Jaccard index [20] which returns a similarity coefficient $\sigma(K_x, K_y) \in [0, 1]$ proportional to the number of common keywords computed as follows:

$$\sigma(K_x, K_y) = \frac{\mid K_x \cap K_y \mid}{\mid K_x \cup K_y \mid}$$

Example. Consider the following keyword sets $K_{d_{295}}$ and $K_{d_{432}}$ associated with the corresponding documents d_{295} and d_{432} acquired in the time window $\mathcal{W}_{2010}$:

$K_{d_{295}}$ = {collins column, hillary, hillary clinton, history, presidential election, winning}
$K_{d_{432}}$ = {hillary, hillary clinton, politics, presidential election, talk, today}

According to the Jaccard coefficient $\sigma(K_{d_{295}}, K_{d_{432}}) = 3/9 = 0.33$.

4. Document clustering task

This task has the goal to move from a basic level where objects are independent documents towards an intermediate level where objects are clusters of similar documents. To this end, document clusters are generated by grouping the documents in the chunk $\mathcal{D}_i$ according to their Jaccard degree of similarity. Document clustering is performed using the HC^{f+} algorithm, a feature-based clustering algorithm which extends the hierarchical algorithm of agglomerative type [21]. The Jaccard similarity coefficient $\sigma_d(K_{d_x}, K_{d_y})$ is computed for each pair of documents $d_x, d_y \in D_i$ based on the corresponding keyword sets K_{d_x} and K_{d_y}, respectively. Results are stored in a *similarity matrix* σM_i,

where an entry $\sigma M_i[x, y]$ corresponds to the similarity coefficient $\sigma_d(K_{d_x}, K_{d_y})$ of K_{d_x} and K_{d_y}. Moreover, a *keyword matrix* κM_i is created, where an entry $\kappa M_i[x, y] = K_{d_x} \cap K_{d_y}$ represents the set of common keywords that concurred to determine the Jaccard similarity $\sigma_d(K_{d_x}, K_{d_y})$ of the documents d_x and d_y. As a difference with the classical hierarchical clustering algorithm, HC^{f+} is capable to support overlapping clusters (i.e., soft clustering). This means that a document d_x can be placed in two clusters cl_a and cl_b due to the fact that d_x can be similar to the documents of cl_a and cl_b for different common keywords.

In HC^{f+}, clusters are obtained through iterative merging operations over documents based on the similarity matrix σM_i and the keyword matrix κM_i. Initially, the two documents $d_x, d_y \in \mathcal{D}_i$ (with $x \neq y$) having the highest similarity coefficient in σM_i are selected and merged in a cluster cl_k with $K_{cl_k} = \kappa M_i[x, y]$ and $D_{cl_k} = \{d_x, d_y\}$. A new line and column of order k are inserted in both σM_i and κM_i for the cluster cl_k. The entry $\sigma M_i[z, k]$ and $\kappa M_i[z, k]$ are then computed to determine the similarity coefficient and the set of common keywords between the new cluster cl_k and each entry z of σM_i and κM_i, respectively. $\sigma M_i[z, k]$ is set to zero when $\kappa M_i[z, k] = \emptyset$. Two documents are candidate/considered for merging in HC^{f+} only if they have a non-empty set of common keywords in κM_i. A clearing stage over the σM_i matrix is performed to remove all those entries that are irrelevant for further merge operations. An entry $\sigma M_i[x, k]$ is irrelevant when the corresponding set of common keywords $\kappa M_i[x, k]$ is a subset of $\kappa M_i[x, y]$, meaning that the cluster cl_k originated by merging d_x, d_y already considers the keywords in $\kappa M_i[x, k]$. Analogously, the entry $\sigma M_i[k, y]$ can be irrelevant and thus cleared from the matrix σM_i. HC^{f+} terminates when the dimension of σM_i is 1. A detailed description of the HC^{f+} algorithm is provided in [21].

The result of the document clustering task is a cluster set $\mathcal{C}_i$ where each cluster $cl_k \in \mathcal{C}_i$, $cl_k = (K_{cl_k}, D_{cl_k}, \mathcal{W}_i)$ is constituted by a cluster keyword-set K_{cl_k}, a set of documents $D_{cl_k} \subset \mathcal{D}_i$, and the time window $\mathcal{W}_i$ where document clustering has been executed.

Example. Consider the following keyword sets $K_{d_{367}}$, $K_{d_{458}}$, and $K_{d_{552}}$ associated with the corresponding documents d_{367}, d_{458}, and d_{552} acquired in the time window $\mathcal{W}_{2010}$:

$$\begin{array}{lcl} K_{d_{367}} & = & \{\text{clinton, head, hillary clinton, israel}\} \\ K_{d_{458}} & = & \{\text{clinton, hillary clinton, political campaign}\} \\ K_{d_{552}} & = & \{\text{political campaign, senate}\} \end{array}$$

whose σM_i and κM_i matrix entries are as follows:

$$\begin{array}{lcllcl} \sigma M_i[367, 458] & = & 0.4 & \kappa M[367, 458] & = & \{\text{clinton, hillary clinton}\} \\ \sigma M_i[367, 552] & = & 0 & \kappa M[367, 552] & = & \{\emptyset\} \\ \sigma M_i[458, 552] & = & 0.25 & \kappa M[458, 552] & = & \{\text{political campaign}\} \end{array}$$

Documents d_{367} and d_{458} are merged into $cl_{367-458}$ since $\sigma M_i[367, 458]$ is the highest value in σM_i. The cluster $cl_{367-458}$ is associated with a keyword set $K_{cl_{367-458}} = \{\text{clinton, hillary clinton}\}$ that are the common keywords of $K_{d_{367}}$ and $K_{d_{458}}$. Analogously, documents d_{458} and d_{552} are merged into $cl_{458-552}$ since $\sigma M_i[458, 552]$ is the only remaining non-zero value. The cluster $cl_{458-552}$ is associated with a keyword set $\kappa cl_{458-552} = \{\text{political campaign}\}$. The resulting document clusters $cl_{367-458}$ and $cl_{458-552}$ are shown in Figure 2.

cluster	$cl_{367-458}$ (2 documents)
keywords	clinton, hillary clinton
documents	d_{367}: Hillary Clinton Heads to Israel 2012-11-20 13:46:54 d_{458}: 2016 Campaign Checklist: Hillary Clinton 2014-06-10 19:30:21
cluster	$cl_{458-552}$ (2 documents)
keywords	political campaign
documents	d_{458}: 2016 Campaign Checklist: Hillary Clinton 2014-06-10 19:30:21 d_{552}: One President, Two Campaign Styles on West Coast Swing 2010-10-23 00:00:00

Figure 2: Example of document clusters $cl_{367-458}$ and $cl_{458-552}$

5. Topic discovery task

This task has the goal to move from an intermediate level characterized by document clusters towards larger document collections (i.e., clusters of document clusters) characterized by thematic topics. Topics are extracted by aggregating document clusters of $\mathcal{C}_i$ through the execution of a second clustering stage based on the HC^{f+} algorithm. Similarity evaluation is performed over cluster keyword-sets, meaning that the Jaccard cluster-similarity coefficient $\sigma_{cl}(K_{cl_x}, K_{cl_y})$ is computed for each pair of clusters $cl_x, cl_y \in C_i$ based on their corresponding keyword sets K_{cl_x} and K_{cl_y}, respectively. Also in this case, similarity results are stored in a similarity matrix σM_i and in a corresponding keyword matrix κM_i. As a difference with the HC^{f+} execution enforced for document clustering, an entry $\kappa M_i[x, y] = K_{cl_x} \cup K_{cl_y}$ is generated as the union of the cluster keyword-sets of cl_x and cl_y. This choice is motivated by the need to provide a rich and exhaustive topic description of the underlying document collection for subsequent exploratory analysis purposes.

The result of topic discovery task is a set of topic clusters $\mathcal{T}_i$ where a topic cluster (from now on simply topic) $T_h \in \mathcal{T}_i$ is defined as follow:

Definition 1 (Topic). A *topic* $T_h = (K_{T_h}, D_{T_h}, \mathcal{W}_i)$ represents a synthetic view described through a keyword set K_{T_h} of a (usually-large) collection of similar documents D_{T_h} (i.e., documents dealing with the same argument) acquired in the time window $\mathcal{W}_i$.

A topic results from the execution of the HC^{f+} clustering algorithm and it is built by grouping the clusters $C_{T_h} \subseteq \mathcal{C}_i$ found to be similar. In particular, given $n = |C_{T_h}|$, the topic keyword-set $K_{T_h} = \bigcup_{j=1}^{j=n} K_{cl_j}$ corresponds to the union of all the keywords featuring the clusters in C_{T_h}, and $D_{T_h} = \bigcup_{j=1}^{j=m} D_{cl_j}$ is the union of the documents belonging to the clusters in C_{T_h}.

Example. Consider the six clusters shown in Figures 3 and 4 belonging to a cluster set $\mathcal{C}_{2010}$ produced as the output of the document clustering task on the documents of $\mathcal{W}_{2010}$.

By executing the HC^{f+} clustering algorithm, such clusters originate the topics T_1 and T_2, respectively (see Figure 5). In particular, clusters cl_{14}, cl_{35}, and cl_{61} are aggregated into the topic T_1 about the campaign of Hillary Clinton for the White House. The clusters cl_{01}, cl_{32}, and cl_{80} are aggregated into the topic T_2 about the Senate elections. The resulting topics T_1 and T_2 are shown in Figure 6 with corresponding documents (i.e., D_{T_1}, D_{T_2}) and topic keyword-sets (i.e., K_{T_1}, K_{T_2}).

cluster	cl_{61} (3 documents)
keywords	run, clinton
documents	d_{927}: Clinton Has Plenty of Reasons to Run for President. 2014-06-14 05:30:09 d_{534}: Chelsea Clinton: I Might Run for Office Someday. 2014-04-16 15:26:25 d_{550}: Clinton Hires Campaign Lawyer Ahead of Likely Run. 2015-03-04 19:13:07
cluster	cl_{14} (7 documents)
keywords	political campaign, hillary clinton
documents	d_{458}: 2016 Campaign Checklist: Hillary Clinton. 2014-06-10 19:30:21 d_{5405}: McConnell Campaigns With an Eye on Clinton. 2014-10-31 12:54:06 d_{5570}: Clinton Campaign Still Aimed at April Start, Supporters Say. 2015-03-11 12:13:05 ...
cluster	cl_{35} (4 documents)
keywords	white house, clinton
documents	d_{5279}: Clinton White House Lawyer Named Top Obama Counsel. 2014-04-21 15:24:46 d_{513}: The Obamas Hosted the Clintons at the White House. 2013-03-08 12:23:50 ...

Figure 3: Three clusters of $\mathcal{C}_{2010}$ resulting from the document clustering task on $\mathcal{W}_{2010}$

cluster	cl_{80} (3 documents)
keywords	democrat, kansas, senate, ballot
documents	d_{781}: Kansas Court: Remove Democrat From Senate Ballot. 2014-09-18 17:40:31 ...
cluster	cl_{01} (3 documents)
keywords	democrat, kansas, senate, court
documents	d_{754}: Court Says Kansas Democrats Don't Have to Run Senate Candidate. 2014-10-01 15:12:30 ...
cluster	cl_{32} (3 documents)
keywords	republican party, overflow, senate
documents	d_{4267}: Racial Politics Churn Miss. GOP Senate Runoff. 2014-06-23 17:10:20 ...

Figure 4: Three clusters of $\mathcal{C}_{2010}$ resulting from the document clustering task on $\mathcal{W}_{2010}$

6. Topic assimilation task

The assimilation task has the goal to recognize links and trends between the topics $\mathcal{T}_i$ discovered in the bootstrapping cycle $\mathcal{B}_i$ with respect to the topics $\mathcal{T}_{i-1}$ emerged in the previous bootstrapping cycle $\mathcal{B}_{i-1}$. A link is established between a topic $T_a \in \mathcal{T}_i$ and a topic $T_b \in \mathcal{T}_{i-1}$ to represent a similarity relationship between two topics emerged in contiguous time windows. A trend is established between a pair of liked topics $T_a \in \mathcal{T}_i$ and $T_b \in \mathcal{T}_{i-1}$ to represent that the argument described by the topic has gained/lost attention in contiguous time windows. Assimilation is performed through *topic similarity* and *topic specificity* measures. In particular, topic similarity identifies pairs of semantically-related topics across contiguous time windows based on their (possibly large) set of common keywords. Topic specificity evaluates how much a topic can be "featuring" for a time window by considering the frequency of all

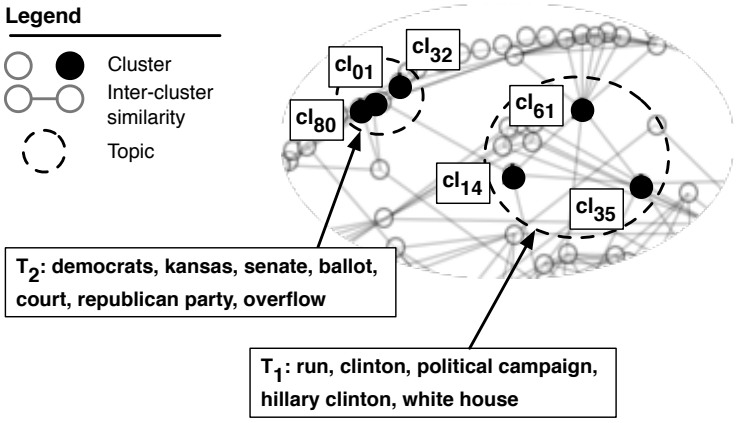


Figure 5: Example of discovered topics

topic	T_1 (14 documents)
keywords	run, clinton, political campaign, hillary clinton, white house
documents	d_{927}: Clinton Has Plenty of Reasons to Run for President. 2014-06-14 05:30:09 d_{458}: 2016 Campaign Checklist: Hillary Clinton. 2014-06-10 19:30:21 d_{5279}: Clinton White House Lawyer Named Top Obama Counsel. 2014-04-21 15:24:46 ...
topic	T_2 (9 documents)
keywords	democrats, kansas, senate, ballot, court, republican party, overflow
documents	d_{781}: Kansas Court: Remove Democrat From Senate Ballot. 2014-09-18 17:40:31 d_{754}: Court Says Kansas Democrats Don't Have to Run Senate Candidate. 2014-10-01 15:12:30 d_{4267}: Racial Politics Churn Miss. GOP Senate Runoff. 2014-06-23 17:10:20 ...

Figure 6: Example of topic T_1 and T_2 representing information about Clinton's campaign for the White House and Senate elections, respectively

the topic keywords over two contiguous time windows.

Definition 2 (Topic similarity). Given two topics $T_a \in \mathcal{T}_i$ and $T_b \in \mathcal{T}_{i-1}$, topic similarity measure $\sigma_T(K_{T_a}, K_{T_b})$ corresponds to the Jaccard similarity coefficient over the topic keyword-sets K_{T_a} and K_{T_b}.

Definition 3 (Topic specificity). Given a topic $T_a \in \mathcal{T}_i$, its specificity degree $\psi(T_a \mid \mathcal{T}_{i-1})$ with respect to the overall topics of $\mathcal{T}_{i-1} \cup \mathcal{T}_i$ is computed as follows:

$$\psi(T_a \mid \mathcal{T}_{i-1}) = \sum_{h=1}^{|K_{T_a}|} \frac{f(k_h) - f^*(k_h)}{\sqrt{f^*(k_h)}}$$

where $k_h \in K_{T_a}$ is a keyword in the set of topic keywords of K_{T_a}, $f(k_h)$ is the frequency of k_h in the set of keywords occurring in $\bigcup_{r=1}^{r=|\mathcal{T}_i|} K_{T_r}$ with $T_r \in \mathcal{T}_i$, and $f^*(k_h)$ is the frequency of k_h in the whole set of keywords occurring in $\bigcup_{r=1}^{r=|\mathcal{T}_i \cup \mathcal{T}_{i-1}|} K_{T_r}$ with $T_r \in \mathcal{T}_i \cup \mathcal{T}_{i-1}$.

For topic assimilation, topic specificity is computed for each topic $T_a \in \mathcal{T}_i$ and $T_b \in \mathcal{T}_{i-1}$. In particular, $\psi(T_a \mid \mathcal{T}_{i-1})$ is computed by considering the frequency of topic keywords of K_{T_a} with respect to topic keywords of $\mathcal{T}_i$ over topic keywords of $\mathcal{T}_i \cup \mathcal{T}_{i-1}$. Analogously, $\psi(T_b \mid \mathcal{T}_i)$ is computed by considering the frequency of topic keywords of K_{T_b} with respect to topic keywords of $\mathcal{T}_{i-1}$ over topic keywords of $\mathcal{T}_i \cup \mathcal{T}_{i-1}$.

A topic $T_a \in \mathcal{T}_i$ is linked to a topic $T_b \in \mathcal{T}_{i-1}$ according to appropriate assimilation operations based on the resulting topic similarity and topic specificity values as described in the following.

6.1. Assimilation of similar topics

Consider a pair of similar topics $T_a \in \mathcal{T}_i$ and $T_b \in \mathcal{T}_{i-1}$, $\sigma_T(T_a, T_b) > 0$. The following assimilation operations are defined based on the value of topic specificity (see the summary table of assimilation rules shown in Figure 7).

$\sigma_T(T_a, T_b) > 0$	$\psi(T_a \mid \mathcal{T}_{i-1}) > 0$	$\psi(T_a \mid \mathcal{T}_{i-1}) \leq 0$
$\psi(T_b \mid \mathcal{T}_i) > 0$	$(T_a \leftrightarrow T_b)^{\uparrow}$	$(T_b \rightarrow T_a)^{\searrow}$
$\psi(T_b \mid \mathcal{T}_i) \leq 0$	$(T_b \rightarrow T_a)^{\nearrow}$	T_a*, T_b*

Figure 7: Assimilation rules for similar topics

Topic merging. When both $\psi(T_a \mid \mathcal{T}_{i-1}) > 0$ and $\psi(T_b \mid \mathcal{T}_i) > 0$, the topics T_a and T_b are considered as "featuring" with respect to the topics discovered in their respective time windows. The two topics are merged (denoted as $T_a \leftrightarrow T_b$). The resulting topic contains the document of both T_a and T_b ($D_{T_a} \cup D_{T_b}$) and the topic keyword-set is defined as $K_{T_a} \cup K_{T_b}$. Moreover, the resulting topic is associated with a *high-specificity mark* (denoted as $^{\uparrow}$) in both the time windows $\mathcal{W}_{i-1}$ and $\mathcal{W}_i$. The specificity of the merged topic $T_a \leftrightarrow T_b$ corresponds to $\psi(T_a \mid \mathcal{T}_{i-1}) + \psi(T_b \mid \mathcal{T}_i)$.

Topic linking. When $\psi(T_a \mid \mathcal{T}_{i-1}) > 0$ and $\psi(T_b \mid \mathcal{T}_i) \leq 0$, the topic T_a is considered as "featuring" for the time window $\mathcal{W}_i$, while the topic T_b is considered as "non-featuring" for the time window $\mathcal{W}_{i-1}$. This is the case of a topic that gains attention moving from the time window $\mathcal{W}_{i-1}$ to time window $\mathcal{W}_i$. The two topics are connected through an uplink (denoted as $T_b \rightarrow T_a^{\nearrow}$). In the opposite situation (i.e., $\psi(T_a \mid \mathcal{T}_{i-1}) \leq 0$ and $\psi(T_b \mid \mathcal{T}_i) > 0$), we are dealing with a topic that loses attention moving from the time window $\mathcal{W}_{i-1}$ to time window $\mathcal{W}_i$. In this case, the two topics are connected through an downlink (denoted as $T_b \rightarrow T_a^{\searrow}$).

Topic pruning. When both $\psi(T_a \mid \mathcal{T}_{i-1}) \leq 0$ and $\psi(T_b \mid \mathcal{T}_i) \leq 0$, the topics T_a and T_b are considered as "non-featuring" with respect to the topics discovered in their respective time windows. The two topics are marked to be discarded (denoted as T_a*, T_b*). Marked topics marked are maintained in the set of available topics until the periodic execution of garbage collection definitively removes them (see Section 7).

topic	T_{45} (14 documents)
keywords	attack, voter, clinton, political campaign, hampshire
documents	d_{291}: Clinton Campaign Starts 5-Point Attack on Obama 2008-02-26 00:00:00
	d_{211}: Clinton Is Out $13 Million She Lent Campaign 2008-12-23 00:00:00
	d_{146}: Clintons Campaign 2007-10-13 00:00:00
	...

Figure 8: Example of topic T_{45} discovered in $\mathcal{W}_{2005}$

Example. Consider the topic T_1 shown in Figure 6 about Clinton's campaign that has been discovered in $\mathcal{W}_{2010}$ and a topic T_{45} shown in Figure 8 that has been discovered in $\mathcal{W}_{2005}$. The topics T_1 and T_{45} are similar ($\sigma_T(K_{T_1}, K_{T_{45}}) = 0.2 > 0$). Topic specificity is $\psi(T_1 \mid \mathcal{T}_{2005}) = -0.014$ and $\psi(T_{45} \mid \mathcal{T}_{2010}) = 0.007$. This means that T_1 is losing attention from $\mathcal{W}_{2005}$ to $\mathcal{W}_{2010}$. According to the assimilation rules, T_1 and T_{45} are connected through a downlink ($T_{45} \rightarrow T_1^{\searrow}$).

6.2. Assimilation of new topics

A new topic $T_a \in \mathcal{T}_i$ is a topic about an emerging argument for which a similar topic does not exists in previous bootstrapping cycles, $\nexists T_b \in \mathcal{T}_{i-1} \mid \sigma_T(T_a, T_b) > 0$. Two different operations are possible based on the value of topic specificity $\psi(T_a \mid \mathcal{T}_{i-1})$. If $\psi(T_a \mid \mathcal{T}_{i-1}) > 0$, the topic T_a is associated with a *high-specificity mark* $^{\uparrow}$ to denote that T_a can be considered as a "featuring" topic for the time window $\mathcal{W}_i$. On the opposite, if $\psi(T_a \mid \mathcal{T}_{i-1}) \leq 0$, the topic T_a is marked to be discarded (T_a*) to denote that T_a can be considered as a "non-featuring" topic of the time window $\mathcal{W}_i$.

7. Exploratory analysis framework

The goal of exploratory analysis is twofold. On one side we aim at supporting thematic analysis and topic expansion techniques by analyzing topic keywords. The focus is on getting information on topics within a given time window. On the other side, we aim at supporting temporal analysis and trend discovery for studying topic dynamics and evolution across different time windows. Topic(s) of interest for exploratory analysis can be retrieved through the topic filtering operator $\phi(K, s, e)$ defined as follows.

Definition 4 (Topic filtering). Given a time interval of interest, specified by the starting time s and the ending time e, and a set K of keywords of interest, topic filtering $\phi(K, [s, e])$ returns all the topics T_k with time window $\mathcal{W}_i$ such that $K \cap K_{T_k} \neq \emptyset$ and $s \leq i + \delta \vee e \geq i$.

In other terms, the filtering operator returns an ordered list of topics that are associated with at least one of the keywords in K whose time window has a non-empty overlap with the time interval $[s, e]$. Retrieved topics are listed by decreasing order of specificity value $\psi(T_k \mid \mathcal{T}_i)$ and grouped by time window. On each retrieved topic, the user can select the analysis to be performed, either thematic or temporal.

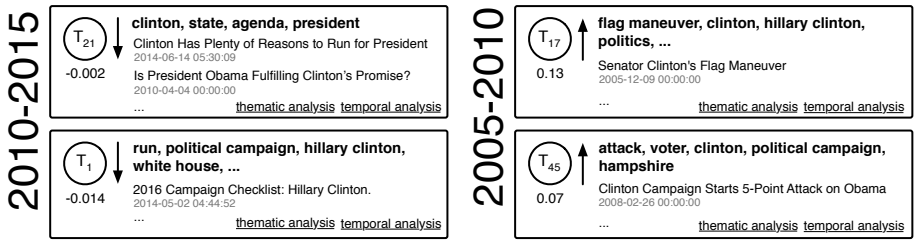


Figure 9: Example of topics retrieved by the filtering operation $\phi(\{clinton,\ white\ house\},\ [2005,\ 2015])$

Example. As an example of topic filtering, we execute the search $\phi(\{clinton,\ white\ house\},\ [2005,\ 2015])$ to retrieve topics referring to Clinton or the White House in the time interval [2005-2015]. A portion of the retrieved topics ordered by decreasing specificity and grouped by time windows is shown in Figure 9.

7.1. Thematic analysis and topic expansion

Thematic analysis and topic expansion focuses on analyzing keyword distributions and it is based on the keyword correlation measure defined as follows.

Definition 5 (Keyword correlation). Given a time window $\mathcal{W}_i$, the correlation of any pair of topic keywords (k_n, k_m) is calculated as follows:

$$\kappa(k_n, k_m) = \mid \{T_j \in \mathcal{T}_i : k_n \in K_{T_j} \wedge k_m \in K_{T_j}\} \mid,$$

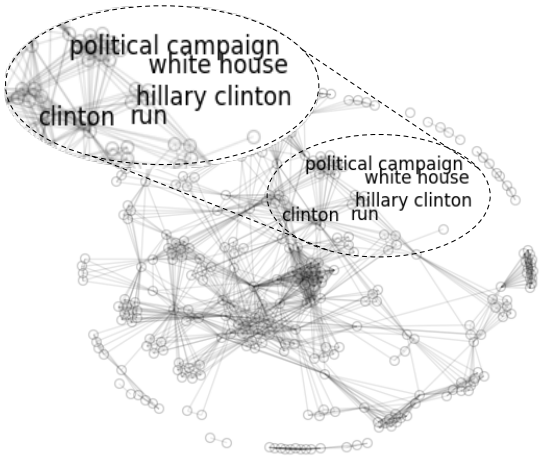
where $\mathcal{T}_i$ denote the set of topics discovered in the time window $\mathcal{W}_i$.

Given a retrieved topic T_k, keyword correlation provides analytical statistics about the correlation of keywords in all the topics $\mathcal{T}_i$ that have been discovered in the same time window $\mathcal{W}_i$ associated with T_k. We exploit keyword correlation in order to build a correlation graph $\mathcal{G}_i^+$ where nodes represent keywords and labeled edges represent correlations with the corresponding degree of correlation. On top of the correlation graph, we calculate also a ranking of keywords of the time window $\mathcal{W}_i$ according to their closeness centrality. Given n as the number of keywords (i.e., nodes) in $\mathcal{G}_i^+$, closeness centrality is defined as the reciprocal of the sum of the shortest path distances from a keyword k_j to all n-1 other nodes, normalized by the sum of minimum possible distances $n - 1$.

Example. As an example of thematic analysis of topic T_1 with $\mathcal{W}_{2010}$, we consider all the topics $\mathcal{T}_{2010}$ discovered in the time window $\mathcal{W}_{2010}$. In Figure 10, we show the resulting correlation graph where for the sake of readability we highlight only the keywords of the topic T_1 shown in Figure 6. Moreover, Figure 10 reports also a portion of the ranking of keywords of the time window $\mathcal{W}_{2010}$ according to their decreasing order of centrality, where T_1 keywords are in bold.

Keyword correlation graph can be exploited also to expand T_k in order to show potentially interesting relations of T_k with other topics in $\mathcal{T}_i$.

Definition 6 (Topic expansion). Given a topic T_k of $\mathcal{T}_i$, we define topic expansion T_k^+ as the set of topics having at least one keyword directly connected with at least one keyword of K_{T_k} in the keyword correlation graph $\mathcal{G}_i^+$, such that:



president	0.61
politics	0.585
...	...
political campaign	0.56
...	...
white house	0.56
run	0.513
...	...
clinton	0.437
...	...
election	0.337
...	...
hillary clinton	0.295
scandal	0.29
...	...

Figure 10: Thematic analysis of topic T_1

$$T_k^+ = \{T_j : \exists k_n \in K_{T_j}, k_n \in K_{T_k}^+\}.$$

The topic expansion T_k^+ is the set of topics that have at least one keyword directly correlated with at least one of the keywords of T_k.

Example. The topic expansion T_1^+ of topic T_1 is graphically shown in Figure 11.

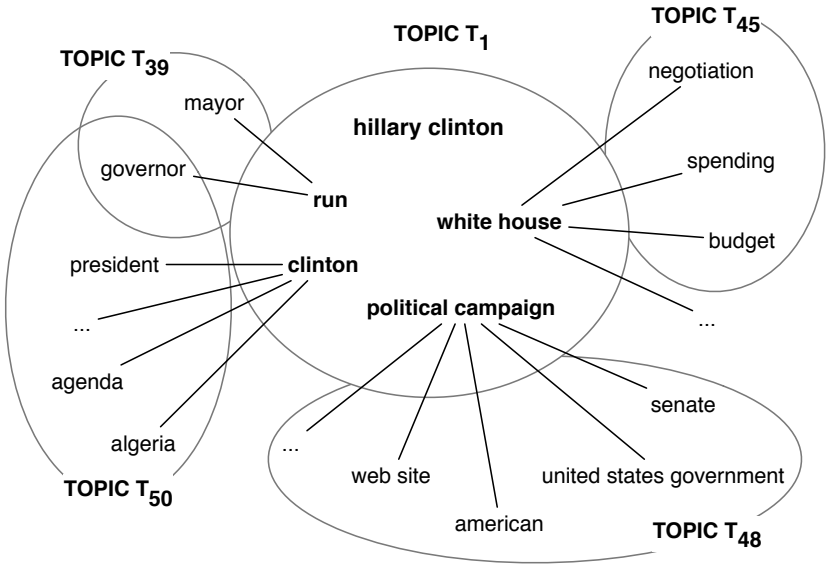


Figure 11: Graphical representation of topic expansion T_1^+

Figure 11 shows an example of how topic expansion can be used to improve the understanding of a topic of interest by analyzing its direct context.

7.2. Temporal analysis and trend discovery

Temporal analysis aims at supporting users in analyzing the evolution and trends of topics in time. To this end, we exploit the links between topics of different time windows resulting from the assimilation task (see Section 6) to build the directed topic link graph $\mathcal{G}^{\mathcal{L}}$ whose nodes denote topics and edges denote links between topics. Given a topic T_k of the time window $\mathcal{W}_i$, T_k can be linked to one or more topics of the previous window and to one or more topics of the subsequent window, depending on the topic similarity measure. Topic evolution is formalized through the notion of *topic trend*.

Definition 7 (Topic trend). Given a topic T_k, the topic trend $\overrightarrow{T_k}$ represents the evolution of T_k in previous time periods (backward path) and/or in subsequent time periods (forward path), according to topic links recognized during the assimilation task. Moreover, topic trend shows the degree of specificity of the topic in each time period.

In Figure 13, we show the trend of topic T_1 from $\mathcal{W}_{2010}$ back. In the trend graph: nodes correspond to the topics that was linked to T_1 in that time window; each topic is collocated in the graph according to its relative degree of specificity with respect to the previous and subsequent topics; the size of each topic is proportional to the number of documents therein contained. In analyzing the trend, the user can see the "invariants" and "variants" keyword portions of the topic over time, by looking at the keywords permanence and disappearance in the different periods. What emerges from the analysis of T_1 trend for instance is a cross-temporal topic related to the name of Clintons, which changes its meaning in different periods of time, due to the fact that it is associated with different events reported by the New York Time.

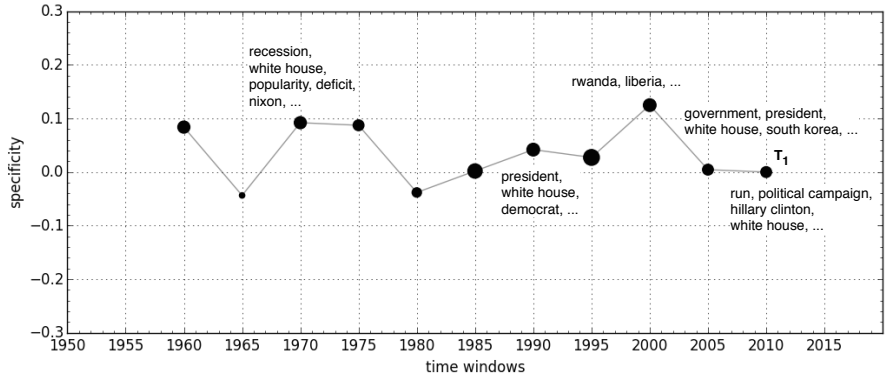


Figure 12: Example of trend for topic T_1

Given a topic T_k, $\overrightarrow{T_k}$ is constructed by recursively visiting the topic link graph $\mathcal{G}^{\mathcal{L}}$ in order to calculate the backward path $\overleftarrow{P}(T_k)$ and the forward path $\overrightarrow{P}(T_k)$ of a topic T_k.

Definition 8 (Backward path). The backward path $\overleftarrow{P}(T_k) = \langle T_0 \rightarrow T_1 \rightarrow \ldots \rightarrow T_n \rangle$ is defined as a path in $\mathcal{G}^{\mathcal{L}}$ such that:

$$W_j \leq W_{j+1}, \forall\, T_j, T_{j+1} \in \overleftarrow{P}(T_k)$$

$$\bigcap_{v=1}^{n} K_{T_v} : T_v \in \overleftarrow{P}(T_k) \neq \emptyset$$

Definition 9 (Forward path). The forward path $\overrightarrow{P}(T_k) = \langle T_0 \rightarrow T_1 \rightarrow \ldots \rightarrow T_n \rangle$ is defined as a path in $\mathcal{G}^{\mathcal{L}}$ such that:

$$W_j \geq W_{j+1}, \forall\ T_j, T_{j+1} \in \overrightarrow{P}(T_k)$$

$$\bigcap_{v=1}^{n} K_{T_v} : T_v \in \overrightarrow{P}(T_k) \neq \emptyset$$

In particular, the topic trend $\overrightarrow{T_k}$ of a topic T_k is constructed according to the following procedure. We first construct the backward path $\overleftarrow{P}(T_k)$ by taking into account all the neighbors T_j of T_k in $\mathcal{G}^{\mathcal{L}}$ such that $W_{T_j} < W_{T_k}$ and we calculate $K_{T_j} \cap K_{T_k}$. If $K_{T_j} \cap K_{T_k} \neq \emptyset$, we insert T_j in the backward path of T_k and we associate T_j with the new set of keywords $K_{T_j} \cap K_{T_k}$. The graph visit is recursively started from T_j until no more neighbor topics T_n are found or when $K_{T_n} \cap K_{T_{n-1}} = \emptyset$. The same procedure is then applied to construct the forward path $\overrightarrow{P}(T_k)$, by recursively visiting the neighbors T_j of T_k such that $\mathcal{W}_j > \mathcal{W}_k$. Finally, we sort $\overleftarrow{P}(T_k)$ and $\overrightarrow{P}(T_k)$ in increasing order of time and we join the two paths on T_k to compose the topic trend $\overrightarrow{T_k}$:

$$\overrightarrow{T_k} = \langle \overleftarrow{P}(T_k), T_k, \overrightarrow{P}(T_k) \rangle$$

Example. As an example of topic trend construction, we take into account the topic T_1 of Figure 6. Since T_1 has been discovered in the last window of the stream (i.e., $\mathcal{W}_{2010}$), the trend is composed only by the backward path. In Figure 13, we graphically show the procedure of trend construction.

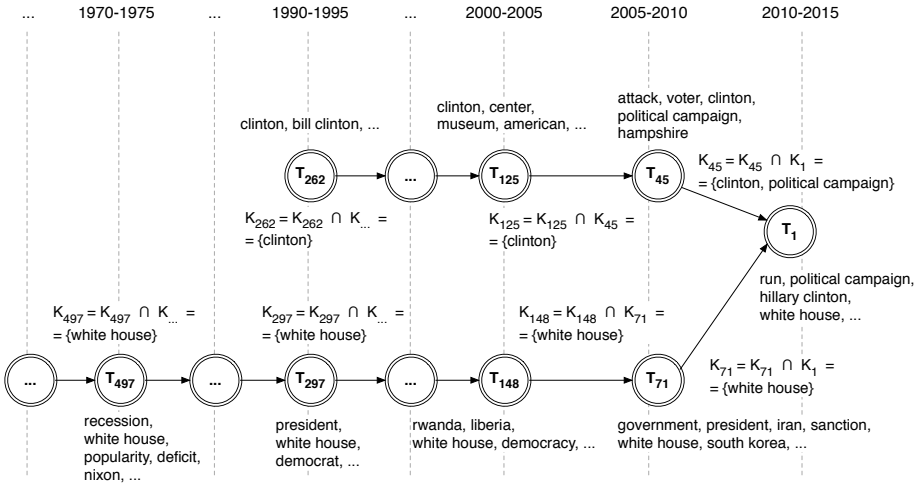


Figure 13: Graphical representation of the trend construction process

The backward path of T_1 is composed by two chains of topics corresponding to different "invariant" parts, one deriving from the keyword 'clinton' and the other from 'white house' (which is reported in Figure 12). As expected, the chain

deriving from 'white house' produces a more persistent and longer backward path, because the term 'white house' has a longer history than 'clinton' in the US politics.

7.3. Application scenarios and issues

The HC^{f+} approach on topic discovery in textual data streams can be applied to different types of textual data streams in different application scenarios. The main applicability issues are related to bootstrapping frequency, keyword management, and topic maintenance.

Bootstrapping frequency. A crucial parameter for supporting different processing needs for different kinds of textual data streams is the size of the time window triggering the bootstrapping cycle. Different window sizes should be adopted and the size of the window should be modified over time, for instance to model the fact that a certain size could have a different meaning in different time periods. We note that our bootstrapping cycles are completely independent because topics are compared only after the clustering phase. As a consequence, it is possible to set time windows of different sizes for different bootstrapping cycles. In particular, we envisage two main scenarios.

Historical analysis. This scenario is suitable when the user is interested in analyzing an entire document flow that has been collected during a (possibly long) period of time, like for instance a document flow acquired since a certain date from document archives, newspaper collections, online news databases. The goal is to explore the evolution of topics over time, such as for studying popularity trends, for mining opinions, or for studying the evolution of ideas in different historical periods. Time is the criterion for recognizing different kinds of topics, like for example "spot" topics, that are relevant only for a limited period of time, or "persistent" topics, that are recurring in different time periods. In the historical analysis scenario, it is typical to choose a large time window (e.g., one or more years) to collect enough data for a significative trend analysis.

Up-to-date analysis. This scenario is suitable for instantly/continuously monitoring one or more document flows, like for instance Twitter timelines, email Inboxes, newspaper web sites. The goal is to get the so-called "hot" topics, namely topics that are fresh in terms of novelty and well-known in terms of popularity. Here, the availability of analysis techniques dealing with time is crucial since the user is interested in discriminating new topics from those already discovered in the past. In the up-to-date analysis scenario, it is typical to choose a small time window (e.g., hours up to one day) to have a frequent update of analysis results.

Keyword management. Document keywords are the core element of the overall bootstrapping process. A design choice of the bootstrapping approach has been to rely only on automatically extracted keywords, from the initial document indexing stage until the final topic discovery and assimilation stages. For this reason, we rely on standard and consolidated text analysis techniques for indexing documents in the data stream. Such keywords are then used to label similarity clusters and topics according to a keyword-set formation procedure which is always automated (i.e., keyword intersection or keyword union operations). This choice of working only on keywords extracted from the textual

data stream is important to make the approach applicable to multiple kinds of textual data streams and to enable a fully automated topic discovery procedure. User-supplied keywords would be more expressive but would require an interactive bootstrapping which is not suitable for managing data streams, especially when frequent updates of topics are needed or large document chunks are involved.

Topic maintenance. In both the scenarios, we accumulate a new set of topics after each bootstrapping cycle. Besides topics that are linked with other topics of the previous and/or subsequent window originating topic trends, non-linked topics marked for elimination in the topic assimilation stage need periodic maintenance. It is important not to purge a marked topic immediately after its corresponding bootstrapping cycles $\mathcal{B}_i$ because it may be linked with other topics in subsequent cycle $\mathcal{B}_{i+\delta}$. For topic maintenance, we adopt a periodic "garbage collection" mechanism. Besides marked topics, garbage collection deletes also topics which are old and not very specific as well as those that are part of very short and not specific trends.

8. Experimental results

The evaluation of the proposed approach aims at demonstrating its effectiveness by focusing on scalability and performance evaluation of HC^{f+} and on the quality of generated clusters, topics, and trends with respect to a gold standard. In fact, we observe that the core algorithm influencing the effectiveness and complexity of proposed bootstrapping approach is the clustering algorithm HC^{f+}, since natural language processing stuff is a pre-processing step on documents performed with standard library routines. A formal description of HC^{f+} and related complexity issues has been discussed in [21]. In the following, we focus on the time and scalability analysis of our clustering techniques for topic discovery and we select as benchmark the Latent Dirichlet Allocation (LDA) and the Hierarchical Dirichlet Process (HDP), the two commonly employed topic modeling techniques. Motivations for this choice are related to the fact that our approach performs clustering with two capabilities that are typical of topic modeling techniques, namely, i) a mechanism for associating descriptive labels to topics and ii) a soft assignment of documents to topics in order to manage multi-topic documents.

8.1. Experimental set up

Our evaluation has been run on top of the NYT dataset presented in Table 4. To set the ground truth, we use the thematic tags directly provided by the New York Times for each article. In particular, given a time window $\mathcal{W}_i$ of size $\delta =$ 5 years, we collected the NYT articles that have been published in $\mathcal{W}_i$ and we created a reference ground truth $\mathcal{R}(\mathcal{W}_i)$ by grouping together all the documents that have at least one tag in common. In such a way, $\mathcal{R}(\mathcal{W}_i)$ contains groups of thematically similar documents (called from now on document *categories*). Document categories constitute the gold standard against which we compare resulting document clusters and topics produced by our bootstrapping process for quality evaluation purposes. The clustering and topic discovering techniques presented in Section 4 have been implemented with the Python programming

language and the evaluation has been run on a Workstation Intel(R) Xeon(R) 3.60GHz with 16Gb RAM running Linux. For the comparison against LDA and HDP, we exploited the Python gensim[3] library, which provides a standard implementation of the online inference for LDA [18] and HDP. This library has been chosen because it is based on the same programming language used for implementing our approach and also because it provides an incremental version of LDA suitable to deal with data streams (see Section 2 for further details). For evaluation, we set a minimum threshold of similarity equal to 0.4 for clustering documents to ensure highly homogeneous clusters. LDA has been configured for discovering 400 topics and running 1000 iterations (HDP does not require to set a predefined number of topics in the configuration phase).

8.2. Time and scalability evaluation

In order to measure the time of computation, we executed our bootstrapping process over the NYT dataset from 1900 to 2015 with time windows of five years. We measured the three main computations of the bootstrapping process, namely *matching*, in which we evaluate document similarity, *document clustering*, in which we execute the HC^{f+} clustering algorithm for document aggregation, and *topic discovery*, in which we aggregate clusters to derive topics. In Figure 14, we report the average time of computation required by each computation, compared with the number of documents processed in each time window.

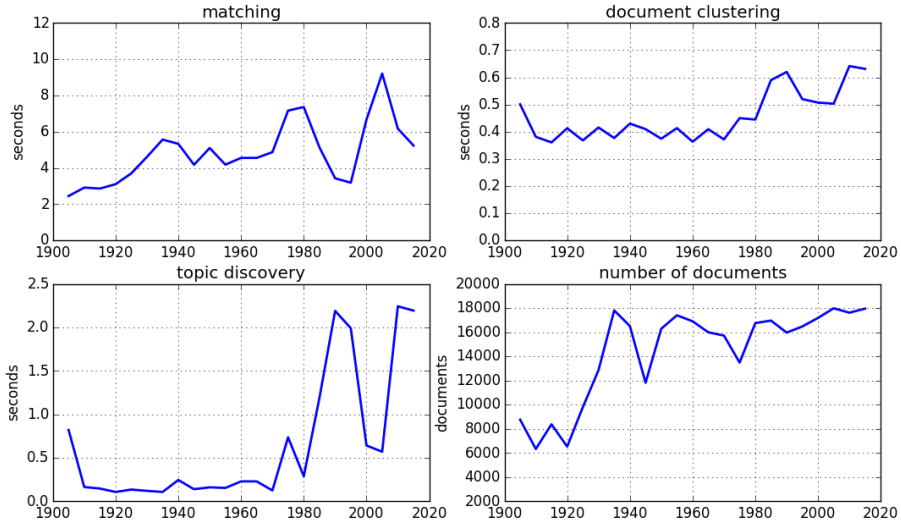


Figure 14: Time required by each bootstrapping computation

We note that matching is the most expensive computation, while document clustering and topic discovery are always faster than matching. As expected, all the three computations depend on the number of documents composing the document chunk in different time windows. In particular, document clustering

[3]https://radimrehurek.com/gensim/

and topic discovery require a total time varying from less than 1s to a maximum of 3s with about 18.000 documents. The average time is about 5s for the matching phase, about 0.4s for document clustering, and about 0.6s for topic clustering.

In Figure 15, we compare our HC^{f+} algorithm with the LDA and HDP algorithms. The incremental execution of LDA/HDP on the whole NYT dataset would produce a single topic model for the entire document collection, resulting in topics that mix documents of different time windows. In order to avoid this and to keep LDA/HDP comparable with our approach, we executed LDA/HDP separately for each time window of the example.

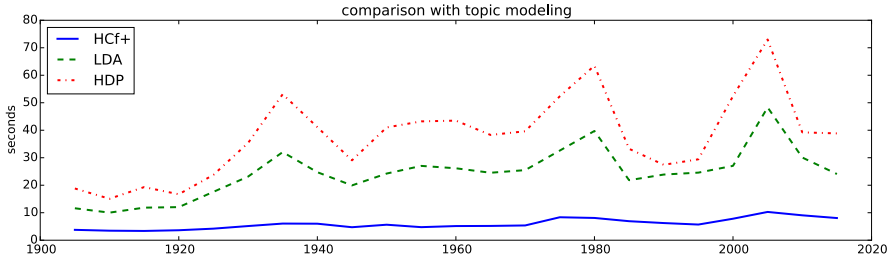


Figure 15: Comparison of our approach with LDA and HDP

We note that our approach outperforms LDA and HDP in all the cases. The average time of computation is about 6s for our approach versus about 24s for LDA and 36s for HDP. The better performance of our approach is mainly due to the fact that clustering operates only on the pairs of documents that have a similarity value higher than the threshold. Documents that are not similar and do not contain common keywords are never compared directly and can be clustered together only when two clusters are merged in the clustering aggregation phase. On the contrary, topic modeling requires to compute the complete distribution of documents and keywords over topics and only after that a threshold-based mechanism can be enforced in order to determine the assignment of documents and keywords to topics.

Scalability. In order to support the exploration of textual data streams acquired from multiple datasources and/or with a high frequency of new document production, it is crucial to scale well when the document chunk size (i.e., the number of documents per time window) becomes large. In order to measure the scalability of our approach, we executed the matching, document clustering and topic discovery computations on document chunks of variable size on the NYT dataset, by processing an increasing number of documents ranging from few thousands to over 180.000, as shown in Figure 16.

We note that document and topic clustering are not affected by the increasing number of documents, mainly because the number of pairs of similar documents that are candidate to be clustered does not increases as the number of document increases. On the opposite, the matching phase grows linearly with respect to the number of documents. However, the comparison with LDA and HDP shows that our approach has a remarkably slower growing rate than LDA/HDP. As in the previous experiment, in our approach the matching phase

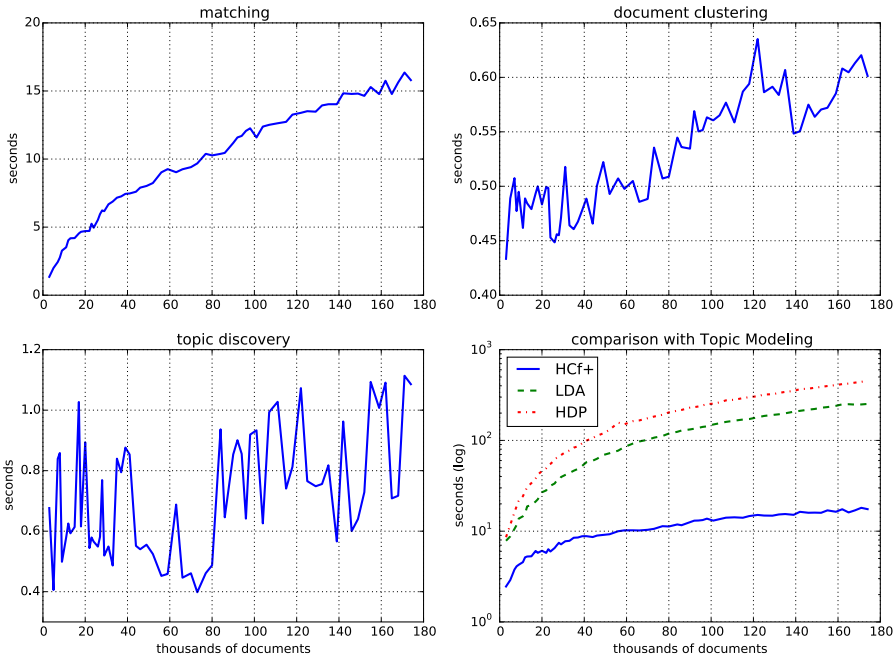


Figure 16: Scalability of our approach with respect to LDA and HDP

discriminates among documents and pre-selects the pairs of documents candidate for matching, while in LDA/HDP the distribution over topics has to be computed for all the documents in the chunk.

8.3. Quality of clusters and topics

For the evaluation of quality of clusters and topics, we compared the results of HC^{f+} against document categories of the ground truth. The comparison includes also the evaluation of quality of topics calculated with LDA/HDP. We aim at evaluating two crucial capabilities of HC^{f+}: i) clustering documents consistently with the ground truth, which means that documents classified in the same category of the ground truth should be grouped in the same HC^{f+} cluster and, conversely, documents that belong to different categories in the ground truth should belong to different HC^{f+} clusters; ii) labeling topics with right keywords, which means that the keywords associated with a topic should correspond to the NYT tags of the topic documents in the ground truth. In order to achieve these goals, we exploited two different measures of quality, namely the Rand coefficient [22] and the keyword quality coefficient.

Rand coefficient. Rand coefficient is based on the idea to analyze pairs of documents and their placement in clusters and categories. In particular, we calculate four statistics: i) TT, the number of document pairs appearing in the same category and HC^{f+} cluster; ii) TF, the number of document pairs appearing in the same category but in different HC^{f+} clusters; iii) FT, the number of document pairs appearing in the same HC^{f+} cluster but different categories; iv) FF, the number of document pairs appearing in different HC^{f+} clusters and

categories. In other words, $TT + FF$ is the number of documents that have been grouped consistently in the ground truth and in the HC^{f+} cluster set, while $TF + FT$ is the number of documents that have been grouped differently in the ground truth and the HC^{f+} cluster set. According to these statistics, the Rand coefficient is calculated as the fraction of consistent results over the total number of results:

$$rand(\mathcal{C}_i) = \frac{TT + FF}{TT + TF + FT + FF}$$

The average results of the evaluation of HC^{f+} and LDA/HDP clusters[4] with respect to the ground truth in all the time windows from 1900 to 2015 are shown in Figure 17.

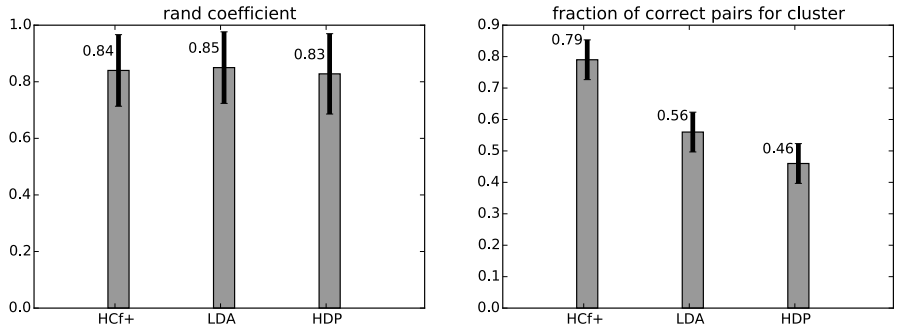


Figure 17: Quality of clusters and LDA/HDP topics according to Rand coefficient

Figure 17 shows that Rand coefficient results are very similar for HC^{f+}, LDA and HDP. However, we reported also statistics about the fraction of document pairs per cluster (i.e., TT) that appear in the same ground truth category as a measure of precision of HC^{f+} clusters and LDA/HDP topics. This statistics shows that our clustering approach performs better in terms of precision with respect to LDA and HDP. The interpretation of this result is that the clusters created by HC^{f+} are based on the keywords that documents have in common. This produces highly homogeneous clusters and, as a consequence, increase the precision of the approach. LDA/HDP topics are instead less homogeneous because are based on latent terminological relations between documents, resulting in a higher recall but a lower precision.

Keyword quality. For evaluation of the keyword quality we measure how much topic keywords match the tags of the ground truth documents. In particular, keyword quality of a topic is the fraction of the topic keywords that correspond to the tags of the topic documents in the ground truth. In HC^{f+}, keywords are intrinsically produced by the clustering mechanism. In case of LDA and HDP, keywords have been produced by exploiting the techniques described in [23]. Given a set of topics $\mathcal{T}_i$, keyword quality $kq(\mathcal{T}_i)$ is calculated as follows:

[4]In the topic modeling terminology, topic denotes the set of similar documents that are grouped because they refer to the same topic.

$$kq(\mathcal{T}_i) = \frac{\sum_{i=1}^{|\mathcal{T}_i|} \mid K_{T_i} \cap \bigcup_{j=1}^{|T_i|} tags(d_j \in T_i) \mid \; / \; \mid K_{T_i} \mid}{\mid \mathcal{T}_i \mid},$$

where K_{T_i} denotes the keywords describing a topic T_i, and $tags(d_j \in T_i)$ is the set of tags provided by the New York Times for a document $d_j \in T_i$.

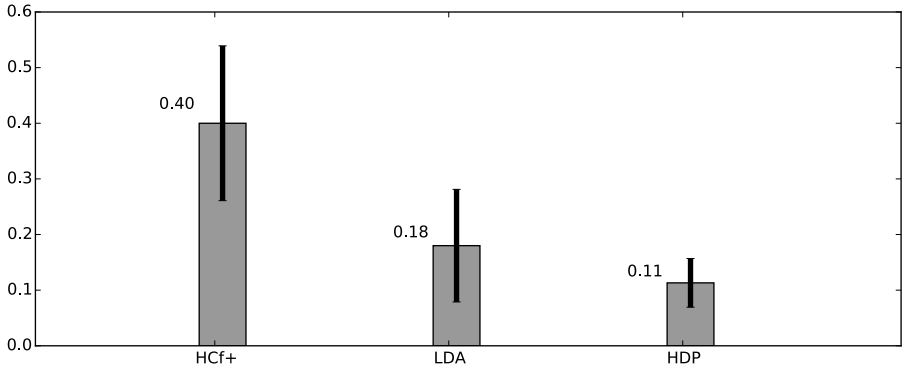


Figure 18: Keyword quality results

The average keyword quality of HC^{f+} and LDA/HDP for all the time windows from 1900 to 2015 is shown in Figure 18. The results show that HC^{f+} topics are associated with good quality keywords, corresponding to NYT tags in about 40% of the cases. In particular, the quality of HC^{f+} topic keywords doubles the quality of keywords provided by LDA/HDP. This result is motivated by the fact that HC^{f+} privileges the keywords that documents have in common as labels for topics, which is a criterion that corresponds in many cases to the criterion used by NYT editors for labeling documents.

8.4. Trend evaluation

For evaluating the quality of trends discovered among topics, we start from the consideration that our trends are a mere consequence of the relative relevance of topics of a time window $\mathcal{W}_i$ and the relative relevance of topics in the previous and subsequent windows $\mathcal{W}_{i-1}$ and $\mathcal{W}_{i+1}$. In particular, having three time windows $\mathcal{W}_{i-1}$, $\mathcal{W}_i$, and $\mathcal{W}_{i+1}$, if the ranking of topics by relevance in $\mathcal{W}_{i-1}$, $\mathcal{W}_i$, and $\mathcal{W}_{i+1}$ is correct with respect to the gold standard, then the trend from $\mathcal{W}_{i-1}$ to $\mathcal{W}_{i+1}$ is also correct. According to this observation, we measured the degree of correlation between the ranking of topic relevance discovered by our approach and the ranking of relevance of the corresponding NYT categories, measured in terms of number of documents per category. In order to measure the ranking correlation, we exploited the Kendall correlation coefficient, which provides a measure of correlation in the interval $[-1; +1]$, with -1 denoting a perfect disagreement between the two rankings, $+1$ denoting a perfect agreement, and 0 denoting independence between the two rankings. The degree of ranking correlation in the time windows from 1900 to 2015 is shown in Figure 19.

We obtained an average correlation of 0.46, which represents a good correlation between the relative relevance of topics discovered by our approach and

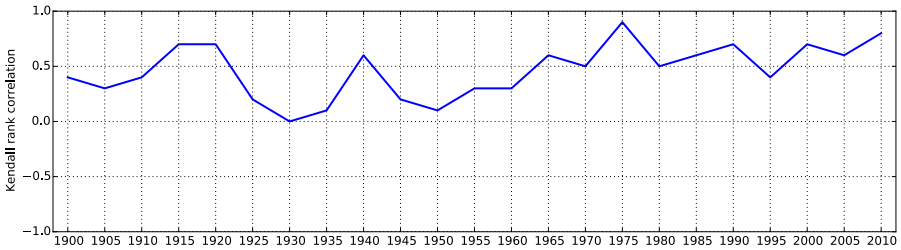


Figure 19: Degree of ranking correlation for topic relevance in the period 1900-2015

the relative relevance of NYT categories in the considered time period. As a consequence, we can conclude that the results of trend evaluation are good, especially for consecutive time windows characterized by a high degree of correlation. This is due to the fact that in our approach we measure the relevance of a topic by measuring how much labels of that topic are specifically used in that time window with respect to the previous time window. Figure 19 shows that this approach is coherent with the distribution in time of the number of occurrences of the NYT tags.

9. Concluding remarks

In this paper, we presented a bootstrapping process based on the HC^{f+} clustering algorithm for exploratory analysis of textual data streams. The topics used for thematic and temporal analysis of documents are obtained through a sequential execution of document clustering, topic discovery, and topic assimilation tasks for each bootstrapping cycle and associated document chunk. Scalability tests show that HC^{f+} can support high-frequent and large textual data streams by providing good quality clusters and keywords even when the size of document chunks to be processed in a single bootstrapping cycle is high (less than 20s are required from processing about 180.000 documents as shown in Figure 16). Evaluation results show that hierarchical unsupervised clustering is a valid approach to topic-based document classification and analysis of textual data streams. In future work, we plan to further extend the evaluation by taking into account a more structured solution for time window setup and modification in relation, for example, to the time period to be analyzed, in order to test different window sizes and related document chunk dimensions.

Appendix A.

Notation	Description
(d_j, t_j)	Document d_j incoming at time t_j
$S = \langle (d_0, t_0), (d_1, t_1), \ldots \rangle$	A textual data stream of incoming documents
$\mathcal{B}_i$	The *i-th* bootstrapping cycle
$\mathcal{W}_i$	The time window triggering $\mathcal{B}_i$
$\mathcal{D}_i$	The document chunk processed in $\mathcal{B}_i$
K_{d_j}	The keyword set featuring a document d_j
$\sigma_d(K_a, K_b)$	Similarity coefficient between two sets of document keywords
$\sigma_{cl}(K_a, K_b)$	Similarity coefficient between two sets of cluster keywords
$\sigma_T(K_a, K_b)$	Similarity coefficient between two sets of topic keywords
σM_i	Similarity matrix for the bootstrapping cycle $\mathcal{B}_i$
κM_i	Keyword matrix for the bootstrapping cycle $\mathcal{B}_i$
$\mathcal{C}_i$	Cluster set discovered in the bootstrapping cycle $\mathcal{B}_i$
$cl_k = (K_{cl_k}, D_{cl_k}, \mathcal{W}_i)$	A cluster belonging to $\mathcal{C}_i$
$\mathcal{T}_i$	Topic set discovered in the bootstrapping cycle $\mathcal{B}_i$
$T_h = (K_{T_h}, D_{T_h}, \mathcal{W}_i)$	A topic belonging to $\mathcal{T}_i$
$\psi(T_h \mid \mathcal{T}_{i-1})$	Specificity of topic T_h with respect to the overall topics of $\mathcal{T}_{i-1} \cup \mathcal{T}_i$
$\phi(K, [s, e])$	Topic filtering operator
$\kappa(k_n, k_m)$	Correlation of keywords k_n and k_m
$\mathcal{G}_i^+$	Keyword correlation graph for the time window $\mathcal{W}_\rangle$
T_k^+	Topic expansion for topic T_k
$\mathcal{G}^{\mathcal{L}}$	Topic link graph
$\overrightarrow{T_k}$	Topic trend for topic T_k
$\overleftarrow{P}(T_k), \overrightarrow{P}(T_k)$	Backward and forward paths of topic T_k, respectively
$rand(\mathcal{C}_i)$	Rand coefficient for the cluster set $\mathcal{C}_i$ with respect to the evaluation ground truth
$kq(\mathcal{T}_i)$	Keyword quality for the cluster set $\mathcal{C}_i$ with respect to the evaluation ground truth

References

[1] C. C. Aggarwal, S. Y. Philip, On clustering massive text and categorical data streams, Knowledge and information systems 24 (2) (2010) 171–196.

[2] Z. Xu, X. Wei, X. Luo, Y. Liu, L. Mei, C. Hu, L. Chen, Knowle: A semantic link network based system for organizing large scale online news events, Future Generation Computer Systems 4344 (2015) 40 – 50. doi:http://dx.doi.org/10.1016/j.future.2014.04.002.
URL http://www.sciencedirect.com/science/article/pii/S0167739X14000636

[3] X. Wang, A. McCallum, Topics over time: a non-markov continuous-time model of topical trends, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2006, pp. 424–433.

[4] S. B. Kaleel, A. Abhari, Cluster-discovery of Twitter Messages for Event Detection and Trending, Journal of Computational Science 6 (2015) 47–57.

[5] W. Gaul, D. Vincent, Evaluation of the Evolution of Relationships between Topics over Time, Advances in Data Analysis and Classification (2016) 1–20.

[6] L. AlSumait, D. Barbarà, C. Domeniconi, On-line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking, in: Proc. of the 8th IEEE Int Conference on Data Mining, Pisa, Italy, 2008, pp. 3–12.

[7] G. Ghosh, S. Banerjee, N. Y. Yen, State transition in communication under social network: An analysis using fuzzy logic and density based clustering towards big data paradigm, Future Generation Computer Systems (2016) –doi:http://dx.doi.org/10.1016/j.future.2016.02.017.
URL http://www.sciencedirect.com/science/article/pii/S0167739X16300309

[8] Y.-B. Liu, J.-R. Cai, J. Yin, A. W.-C. Fu, Clustering text data streams, Journal of computer science and technology 23 (1) (2008) 112–128.

[9] S. Zhong, Efficient streaming text clustering, Neural Networks 18 (5) (2005) 790–798.

[10] J. Allan, Detection as multi-topic tracking, Information Retrieval 5 (2) (2002) 139–157.

[11] K. Nguyen, B. Shin, S. J. Yoo, Hot Topic Detection and Technology Trend Tracking for Patents utilizing Term Frequency and Proportional Document Frequency and Semantic Information, in: Proc. of the Int. Conference on Big Data and Smart Computing (BigComp 2016), Hong Kong, China, 2016, pp. 223–230.

[12] Z. Xu, X. Luo, S. Zhang, X. Wei, L. Mei, C. Hu, Mining temporal explicit and implicit semantic relations between entities using web search engines, Future Generation Computer Systems 37 (2014) 468 – 477, special Section:

Innovative Methods and Algorithms for Advanced Data-Intensive ComputingSpecial Section: Semantics, Intelligent processing and services for big dataSpecial Section: Advances in Data-Intensive Modelling and SimulationSpecial Section: Hybrid Intelligence for Growing Internet and its Applications. doi:http://dx.doi.org/10.1016/j.future.2013.09.027.
URL http://www.sciencedirect.com/science/article/pii/S0167739X13002069

[13] G. Li, W. Zhang, J. Pang, Q. Huang, S. Jiang, Online Web-Video Topic Detection and Tracking with Semi-supervised Learning, in: Proc. of the 14th Pacific-Rim Conference on Multimedia, Nanjing, China, 2013, pp. 750–759.

[14] T. K. Landauer, P. W. Foltz, D. Laham, An introduction to latent semantic analysis, Discourse processes 25 (2-3) (1998) 259–284.

[15] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, the Journal of machine Learning research 3 (2003) 993–1022.

[16] L. Yao, D. Mimno, A. McCallum, Efficient methods for topic model inference on streaming document collections, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, ACM, New York, NY, USA, 2009, pp. 937–946. doi:10.1145/1557019.1557121.
URL http://doi.acm.org/10.1145/1557019.1557121

[17] L. Hong, B. Dom, S. Gurumurthy, K. Tsioutsiouliklis, A time-dependent topic model for multiple text streams, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, ACM, New York, NY, USA, 2011, pp. 832–840. doi:10.1145/2020408.2020551.
URL http://doi.acm.org/10.1145/2020408.2020551

[18] M. Hoffman, F. R. Bach, D. M. Blei, Online learning for latent dirichlet allocation, in: advances in neural information processing systems, 2010, pp. 856–864.

[19] C. D. Manning, P. Raghavan, H. Schütze, Introduction to information retrieval, Vol. 1, Cambridge university press Cambridge, 2008.

[20] A. Ferrara, A. Nikolov, F. Scharffe, Data linking for the semantic web, Semantic Web: Ontology and Knowledge Base Enabled Tools, Services, and Applications 169.

[21] A. Ferrara, L. Genta, S. Montanelli, S. Castano, Dimensional Clustering of Linked Data: Techniques and Applications, Transactions on Large-Scale Data- and Knowledge-Centered Systems XIX (2015) 55–86.

[22] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, Journal of intelligent information systems 17 (2) (2001) 107–145.

[23] D. OCallaghan, D. Greene, J. Carthy, P. Cunningham, An analysis of the coherence of descriptors in topic modeling, Expert Systems with Applications 42 (13) (2015) 5645–5657.