Semantic HMC: Ontology-described hierarchy maintenance in Big Data context

Rafael Peixoto[1, 2], Christophe Cruz[2], Nuno Silva[1]

[1] GECAD - ISEP, Polytechnic of Porto, Porto, Portugal
{rafpp,nps}@isep.ipp.pt
[2] LE2I UMR6306, CNRS, Univ. Bourgogne Franche-Comté, F-21000 Dijon, France
christophe.cruz@u-bourgogne.fr

Abstract. One of the biggest challenges in Big Data is the exploitation of Value from large volumes of data that are constantly changing. To exploit value, one must focus on extracting knowledge from these Big Data sources. To extract knowledge and value from unstructured text we propose using a Hierarchical Multi-Label Classification process called Semantic HMC that uses Ontologies to describe the predictive model including the label hierarchy and the classification rules. To not overload the user, this process automatically learns the ontology-described label hierarchy from a very large set of text documents. This paper aims to present a maintenance process of the ontology-described label hierarchy relations with regards to a stream of unstructured text documents in the context of Big Data without relearn all the hierarchy.

Keywords. Maintenance, multi-label classification, hierarchy induction, ontology, machine learning

1 Introduction

The exponential growth of the amount of data available on the web requires new forms of processing to enable enhanced decision making, insight discovery and optimization. The term of Big Data is mainly used to describe datasets that cannot be processed using traditional tools.

To extract knowledge from Big Data sources we propose to use a Semantic HMC process [1, 2] that is capable of Hierarchically Multi-Classify a large Variety and Volume of unstructured data items. Hierarchical Multi-Label Classification (HMC) is the combination of Multi-Label classification and Hierarchical classification [13]. The Semantic HMC process is unsupervised such that no previous labelled examples or enrichment rules to relate the data items with the labels are required. The label hierarchy and the enrichment rules are automatically learned from the data through scalable Machine Learning techniques.

The automatic concept (label) hierarchy extraction from unstructured documents is not a trivial process and proper techniques for document analysis and representation are required. In the context of Big Data, this task is even more challenging due to Big

Data characteristics. An increasing number of V-dimensions has been used to characterize Big Data further [3, 4]. Volume, Velocity, Variety are usually used to characterize the essence of Big Data [3, 5]. Volume concerns the large amount of data that is generated and stored through the years by social media, sensor data and other sources [3]. Velocity concerns the high speed of data production. Variety relates to the various types of data composing the Big Data. These types include semi-structured and unstructured data representing 90% of his content [5] such as audio, video, web page and text, as well as traditional structured data (e.g. XML, JSON or CSV).

Automatic concept hierarchy learning from text (i.e. taxonomy induction) has been extensively studied [6, 7]. In Big Data however, relearning the hierarchy for each new document is infeasible due to the Big Data characteristics. Most of the existing literature focuses in learning concept hierarchies from texts. Few works has been published in maintaining the hierarchy once it is created without relearning the whole hierarchy. This paper focuses in maintaining hierarchical relations of an automatically learned ontology-described concept hierarchy with regards to a stream of unstructured text documents in Big Data context. The process is implemented using technologies for Big Data that distributes the process by several machines in order to reach high performance and scalability.

The rest of the paper covers four sections. The second section presents the background knowledge and related work. The third section describes the hierarchy maintenance process from a stream of text documents. The fourth section describes the implementation in a scalable and distributed platform to process Big Data. Finally, the last section draws conclusions and suggests further research.

2 Background and related work

This section introduces some background and discusses the current related work about automatic concept hierarchy learning from text.

Two main different aspects of concept hierarchy extraction from text can be distinguished: (1) concept extraction and filtering (2) hierarchy creation. Several methods exists in literature for extracting concepts (i.e. terms) from a set of text documents, including linguistic [8–10] and statistical approaches [11, 12]. Many types of semantics can be captured using statistics-based methods [13] that offer better scaling performance than their alternatives [14]. Several statistic-based methods exist to create hierarchical relations between concepts, including [14, 15]:

- Hierarchical clustering that starts with one cluster and progressively merges the closest clusters.
- Subsumption methods that construct the concept broader-narrower relations based on the co-occurrence of concepts [16].

More advantages and drawbacks of each method are deeply studied by De Knijff et al. [14].

Most work in this area focus on the hierarchy creation and few works have been done on the hierarchy maintenance since it is created, even less in Big Data context.

Liu et al. [17] automatically induce a hierarchy by combining phrases extracted from the collection using clustering with context knowledge derived from a knowledge base. Recent efforts have been made to organize text corpora using evolving multi-branch trees [18], known as topic trees. Cui et al. [19] present a visual topic analytics system, called RoseRiver, to help users better understand the hierarchical topic evolution at different levels of granularity.

Compared with existing approaches, maintaining an automatically induced hierarchy from Big Data requires simple and greatly scalable methods due to its high volume and velocity of data production. Due to the simplicity and its relation between the processing speed and ability to provide good concept hierarchical relations [14], the subsumption method is used to learn and maintain the relations between concepts. To the best of our knowledge, our approach is the first one to apply a subsumption method in order to maintain the concept hierarchy relations (Hierarchy Maintenance) with regards to a stream of unstructured text documents in Big Data.

3 Hierarchy Maintenance

This section describes the hierarchy maintenance process in detail. The label hierarchy is automatically learned based on a Description Logic ontology presented in Table 1. The $Item$ class represents data items to be labeled/classified and is populated with data items (i.e. text document) in the assertional level (Abox). The $Term$ class defines the extracted terms from data items and it is populated in the assertion level (Abox) with terms (e.g. words extracted from text documents, symbols representing subjects/objects in photos). The $Label$ class defines the terms that are considered to classify the items i.e. the most relevant terms. The $broader$ and $narrower$ relations define the subsumption hierarchy between terms. The $hasTerm$ relation links the asserted Items to the asserted Terms.

Table 1. Label hierarchy ontology

DL concepts	Description
$Item \sqsubseteq \top$	Data item to label (e.g. document)
$Term \sqsubseteq \top$	Extracted terms (e.g. word)
$Label \sqsubseteq Term$	Terms used to classify the items
$Term \sqsubseteq \exists broader.Term$	Broader relation between terms
$Term \sqsubseteq \exists narrower.Term$	Narrower relation between labels
$broader \equiv narrower^{-}$	Broader and narrower are inverse relations
$Item \equiv \exists hasTerm.Term$	Relation that links data items to the terms
$Item \sqcap Term = \emptyset$	Term is disjointed from Item

In previous work [2] the hierarchy relations (i.e. the hierarchy created using a traditional subsumption method) are induced from a collection of documents C using the co-occurrence of terms in documents. The terms in text documents can be either a unigram (i.e. a word, "CEO") or more complex n-grams (i.e. composed words, "chief executive officer"). This paper proposes to maintain the hierarchy relations by using a stream Stm of new documents. The new documents from Stm are included in the collection C originating a new collection C' impacting the term co-occurrence and consequently the hierarchy. In order to persist the co-occurrence values, a term co-occurrence frequency matrix is used to represent the co-occurrence of any pair of terms in a collection of documents [2].

Three steps comprise the maintenance process: (i) Co-occurrence matrix maintenance that calculates the impact of new documents in the co-occurrence matrix, (ii) Hierarchical relationship maintenance that calculates how the changes in the co-occurrence matrix impact the hierarchical relations, and (iii) Change detection that detects hierarchical changes and updates the hierarchy. These maintenance steps are described in the following subsections.

3.1 Co-occurrence matrix maintenance

The co-occurrence matrix used by the subsumption method is impacted with the inclusion of new text documents in the document collection. The set of documents C is a collection of n documents. The sub-set of $documents \subseteq C$ that contains a $term \in Term$ is represented by a set of vectors in the form: $vector_{term} < d_1, d_2, \dots, d_n >$.

The co-occurrence frequency matrix cfm is used to represent the co-occurrence of any pair of terms $(term_i, term_j)$ in a collection C of documents, such that:

$$cfm_C(term_i, term_j) = \left|\left\{doc \in C : doc \in vector_{term_i} \wedge doc \in vector_{term_j}\right\}\right| \quad (1)$$

where $vector_{term_i}$ is the document vector for $term_i$ and $vector_{term_j}$ is the vector for term $term_j$. Let m denote the number of different terms in a collection of documents C, the term co-occurrence matrix for the collection C is a $m \times m$ symmetric matrix. The main diagonal of the co-occurrence matrix $cfm_C\ (term_i, term_i)$ denotes the occurrence of $term_i$ in all documents of C. Hence the main diagonal $cfm_C\ (term_i, term_i)$ is the maximum value for the line $term_i$ and the column $term_i$ of the co-occurrence table.

However, in document streaming context, the collection C will be constantly evolving. For each new text document doc' in the stream, the collection of documents C will evolve for a C' collection impacting the co-occurrence matrix. Assuming that no new terms are extracted from the new document, the set of terms $term_i$ in the new document doc' is defined as:

$$setTerms_{doc'} = \{term_i \in Term : term_i \in doc'\} \quad (2)$$

All vectors of documents $vector_{term}$ where $term \in setTerms_{doc'}$ are updated by adding the new document doc' such as: $vector_{term} < d_1, d_2, d_n, doc' >$. Hence for each pair of terms $(term_i, term_j) | \ term_i, term_j \in setTerms_{doc'}$ the co-occurrence matrix is impacted such as:

$$cfm_{C'}(term_i, term_j) = cfm_C(term_i, term_j) + 1 \tag{3}$$

3.2 Hierarchical relationship maintenance

The hierarchical relations are maintained according to the evolved co-occurrence matrix without re-processing the entire hierarchy. A method based on [16] is used, which exploits the co-occurrence matrix to calculate the hierarchical relations between terms where term $term_x$ potentially subsumes term $term_y$ if (Fig. 1(A)):

$$(P_C(term_x | term_y) \geq \text{st}) \wedge (P_C(term_y | term_x) < \text{st}) \tag{4}$$

where:

- $P_C(term_x | term_y)$ is the conditional proportion (number) of the documents from collection C common to $term_x$ and $term_y$, in respect to the number of documents in $term_y$ such that:

$$P_C(term_x | term_y) = \frac{term_x \cap term_y}{term_y} = \frac{cfm_C(term_x, term_y)}{cfm_C(term_y, term_y)} \tag{5}$$

- $P_C(term_y | term_x)$ is the conditional proportion (number) of the documents from collection C common to $term_x$ and $term_y$, in respect to the number of documents in $term_x$ such that:

$$P_C(term_y | term_x) = \frac{term_x \cap term_y}{term_x} = \frac{cfm_C(term_x, term_y)}{cfm_C(term_x, term_x)} \tag{6}$$

- $st \in [0,1]$ is the subsumption co-occurrence threshold.

Hence if $term_x$ appears in at least proportion st of the documents in which $term_y$ also appears, and $term_y$ appears in less than proportion st in which $term_x$ appears, then the hierarchical relationship between x and y is statistically relevant and $term_x$ potentially subsumes $term_y$, i.e. $term_y \prec term_x$. By applying this method to all terms, the result is a subsumption hierarchy of terms. The hierarchy relations can be induced with all statistically relevant hierarchical relationships (i.e. DAG) or with only some relationships (i.e. Tree).

Since the hierarchy is created from the initial collection of documents C it must be maintained according to the stream of new documents. Two types of changes in the co-occurrence matrix impact the conditional proportions used to calculate the hierarchal relations. That changes are (1) changes in the co-occurrence $cfm_C(term_i, term_j)$ where $term_i \neq term_j$ and (2) changes in the main diagonal $cfm_C(term_i, term_i)$.

Change in the Co-occurrence.

A change in co-occurrences $cfm_C(term_i, term_j)$ where $term_i \neq term_j$, impact the numerator of the two conditional proportions described by equations (5) and (6), used by the subsumption method (4) to calculate the hierarchical relations. Hence, only the $P_C(term_i|term_j)$ and $P_C(term_j|term_i)$ proportions used to create the hierarchical relationships between $term_i$ and $term_j$ are impacted.

As an example, consider the set of terms $\{term_1, term_2, term_3, term_4, term_5\}$ with the induced hierarchy relation as depicted in Fig.1 (A). A co-occurrence change in the matrix $cfm_C(term_1, term_2)$ will impact the relationship between the terms $term_1$ and $term_2$ (Fig.1 (B)).

Change in the main diagonal.

A change in the matrix main diagonal $cfm_C(term_i, term_i)$ for $cfm_{C'}(term_i, term_i)$ will impact the denominator of the two conditional proportions described by equations (5) and (6), used by the subsumption method (4) to calculate the hierarchical relations. Hence all proportions $P_C(term_i|term_n)$ and $P_C(term_n|term_i)$, where $term_n \in Term$ and $term_n \neq term_i$, are impacted. In other words, all the relationships between that term and all other terms from the collection C are impacted.

As an example, consider the set of terms $\{term_1, term_2, term_3, term_4, term_5\}$ with the induced hierarchy relation as depicted in Fig. 1(A). A main diagonal change in the matrix $cfm_C(term_2, term_2)$ impacts the hierarchical relationships with all other terms in the set (Fig. 1(C)). The broken line in Fig.1 (C) represents the impacted relation between $term_2$ and $term_3$ even it are not consider a hierarchical relation regarding the collection C.

3.3 Change Detection

Only the statistically relevant hierarchical relationships obeying to the subsumption method are induced as relations in the output hierarchy. After recalculating all the impacted proportions by C' (i.e. $P_{C'}(term_x|term_y)$ and $P_{C'}(term_y|term_x)$) the changes to the hierarchy relations regarding the previous collection C are detected. A change can be of two types: "Add", when a new hierarchy relation is induced and "Delete", when a hierarchy relation is no longer induced.

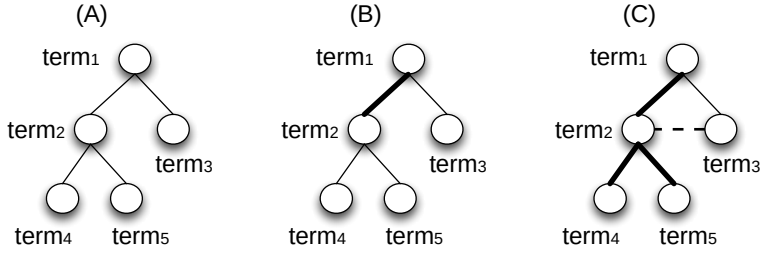

Fig. 1. Impact the hierarchical relations (example).

An "*add*" change is detected when a relationship between two terms is not induced as a hierarchy relation in C but is induced in C'. On the other hand a "delete" change is detected when a relation is induced as hierarchy relation in C but is not induced in C'.

For example, consider the set of terms $\{term_1, term_2, term_3, term_4\}$ for the initial collection of documents C and that the subsumed terms of t_2 are calculated with a subsumption threshold $st = 65$. The proportions of $term_2$ and all other terms ($term_y$) in the collection C as well as the subsumption threshold are depicted in Fig. 2.

According with the subsumption method, one can observe in Fig. 2 that $term_1$ and $term_4$ are subsumed terms of $term_2$ as $(P_C(term_2|term_1) > st) \wedge (P_C(term_1|term_2) < st)$ and $(P_C(term_2|term_4) > st) \wedge (P_C(term_4|term_2) < st)$. However $term_3$ does not subsume $term_2$ because the subsumption condition with $st = 65$ is not granted $(P_C(term_2|term_4) > st) \wedge (P_C(term_4|term_2) > st)$. Hence the initial hierarchichal relations created for the $term_2$ are $term_1 \prec term_2$ and $term_4 \prec term_2$.

With the arrival of new documents with $term_1$ and $term_2$, the collection C will evolve to C'. Considering the impacted proportions as depicted in Fig. 3:

- A *delete* change is identified between $term_2$ and $term_1$ where the relation is no longer induced $(P_{C'}(term_2|term_1) < st) \wedge (P_{C'}(term_1|term_2) < st)$;
- An *add* change is detected between $term_2$ and $term_3$ where the hierarchical relationship is induced $(P_{C'}(term_2|term_3) > st) \wedge (P_{C'}(term_3|term_2) < st)$.

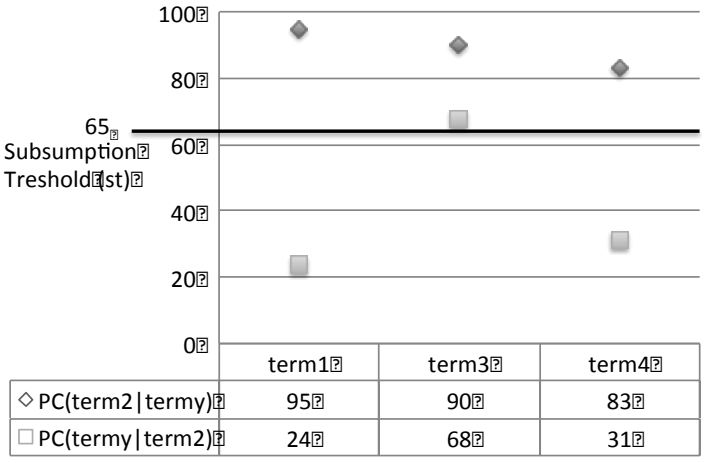

Fig. 2. Calculate the subsumed terms of $term_2$ in the collection C (example).

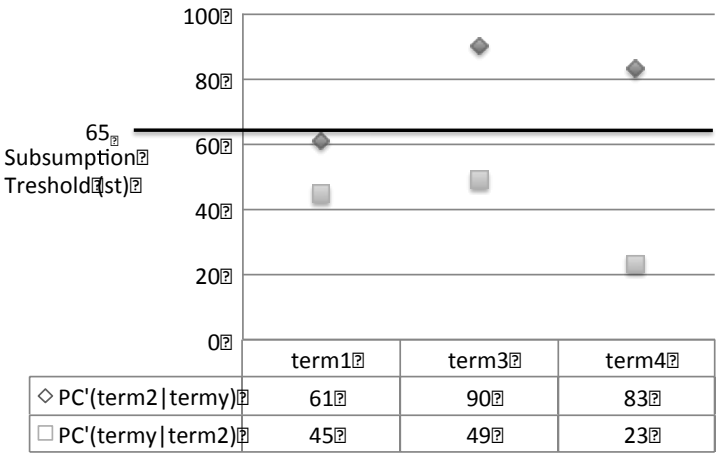

Fig. 3. Calculate the subsumed terms of $term_2$ in the collection C'.

4 Implementation

In this section an implementation of the proposed hierarchy maintenance process is described. The proposed process is implemented in a scalable and distributed platform for Big Data. The hierarchical relations are maintained using documents streamed using a distributed and highly scalable broker Apache Kafka (http://kafka.apache.org).

The distributed real-time computation system Apache Storm (https://storm.apache.org) is used to process the text document streaming in order to maintain the hierarchy. The Storm architecture is based in: Tuples (Storm data item abstraction); Stream (unbound list of Tuples); Spout (source of a Stream); Bolts (Process the Streams of Tuples from Spouts to create new Streams) and Topologies (a directed graph of Spouts and Bolts). Four components comprise the maintenance topology (Fig. 4): a spout (in square) and three bolts (in ellipsis).

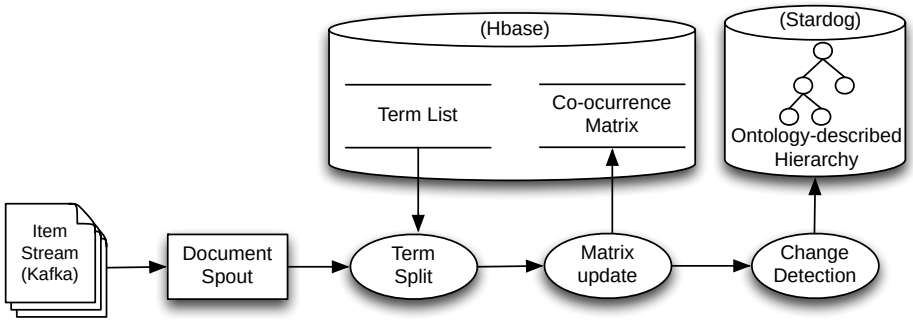

Fig. 4. Hierarchy Maintenance Topology

The spout reads the streamed documents from the Kafka broker providing an interface between Storm and Kafka. The first bolt splits the documents in terms using a pre-defined term list. Several methods exists in the literature for extracting the terms from a set of text documents, including linguistic [8–10] and statistical approaches [11, 12] but obtaining this list is out of the scope of this paper. The second bolt updates the co-occurrence matrix according with section 3.1. The matrix is stored in a distributed and scalable Big Data store Apache HBase (http://hbase.apache.org). HBase is a non-relational column-oriented database that is built on the top of HDFS (Hadoop Distributed File System). Its column-oriented architecture is a good choice to fit the co-occurrence matrix as it provides efficient random access to each table cell. Also the native method to increment counters in the database is used to maintain the matrix regarding the new documents. The last bolt detects the hierarchical changes as described in the previous section. The change detection is processed on the fly where no proportions are stored. This change can be automatically applied to the hierarchy or recommended to a supervisor.

5 Conclusions

To maintain the hierarchical relation automatically induced from text documents in Big Data context without relearn the whole hierarchy, this paper proposes a maintenance process from a stream of text documents. Three steps comprise the maintenance process (1) Co-occurrence matrix maintenance, (2) Hierarchical relationship maintenance and (3) Change detection.

The prototype was successfully implemented in a scalable and distributed platform to process Big Data. Our future research efforts will focus in studying the relevance and pertinence of using the proposed maintenance process in the Semantic HMC process to maintain the ontology-described concept hierarchy used to classify documents in Big Data context.

Acknowledgements. This project is founded by the company Actualis SARL, the French agency ANRT and through the Portuguese COMPETE Program under the project AAL4ALL (QREN13852).

References

1. Hassan, T., Peixoto, R., Cruz, C., Bertaux, A., Silva, N.: Semantic HMC for big data analysis. Big Data (Big Data), 2014 IEEE International Conference on. pp. 26–28 (2014).
2. Peixoto, R., Hassan, T., Cruz, C., Bertaux, A., Silva, N.: Semantic HMC: A Predictive Model using Multi-Label Classification For Big Data. To appear in The 9th IEEE International Conference on Big Data Science and Engineering (IEEE BigDataSE-15) (2015).
3. Chen, M., Mao, S., Liu, Y.: Big Data: A Survey. Mob. Networks Appl. 19, 171–209 (2014).
4. Hitzler, P., Janowicz, K.: Linked data, big data, and the 4th paradigm. Semant. Web. 4, 233–235 (2013).

5. Syed, A., Gillela, K., Venugopal, C.: The Future Revolution on Big Data. Future. 2, 2446–2451 (2013).
6. Medelyan, O., Manion, S., Broekstra, J., Divoli, A., Huang, A.L., Witten, I.H.: Constructing a focused taxonomy from a document collection. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 7882 LNCS, 367–381 (2013).
7. Caraballo, S.A.: Automatic Construction of a Hypernym-labeled Noun Hierarchy from Text. Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics. pp. 120–126. Association for Computational Linguistics, Stroudsburg, PA, USA (1999).
8. Hearst, M. a.: Automatic Acquisition of Hyponyms ftom Large Text Corpora. Proc. 14th Conf. Comput. Linguist. 2, 23–28 (1992).
9. Toutanova, K., Manning, C.D.: Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. Proc. Jt. SIGDAT Conf. Empir. Methods Nat. Lang. Process. Very Large Corpora. 63–70 (2000).
10. Cimiano, P., Staab, S., Tane, J.: Automatic acquisition of taxonomies from text: FCA meets NLP. Proceedings of the International Workshop & Tutorial on Adaptive Text Extraction and Mining (2003).
11. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. 24, 513–523 (1988).
12. Maedche, A., Volz, R.: The ontology extraction & maintenance framework Text-To-Onto. Proc. Work. Integr. Data …. 1–12 (2001).
13. Halevy, a., Norvig, P., Pereira, F.: The Unreasonable Effectiveness of Data. IEEE Intell. Syst. 24, (2009).
14. De Knijff, J., Frasincar, F., Hogenboom, F.: Domain taxonomy learning from text: The subsumption method versus hierarchical clustering. Data Knowl. Eng. 83, 54–69 (2013).
15. Meijer, K., Frasincar, F., Hogenboom, F.: A Semantic Approach for Extracting Domain Taxonomies from Text. Decis. Support Syst. (2014).
16. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '99. 206–213 (1999).
17. Liu, X., Song, Y., Liu, S., Wang, H.: Automatic taxonomy construction from keywords. Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. 1433–1441 (2012).
18. Wang, X., Liu, S., Song, Y., Guo, B.: Mining evolutionary multi-branch trees from text streams. Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '13. 722 (2013).
19. Cui, W., Liu, S., Member, S., Wu, Z., Wei, H.: How Hierarchical Topics Evolve in Large Text Corpora. IEEE Trans. Vis. Comput. Graph. 20, 2281–2290 (2014).