A SEMANTIC METADATA ENRICHMENT SOFTWARE ECOSYSTEM BASED ON TOPIC METADATA ENRICHMENTS

Ronald Brisebois[1], Alain Abran[1], Apollinaire Nadembega[2] and Philippe N'techobo[3]

[1]École de technologie supérieure, University of Quebec, Montreal, Canada
[2]Network Research Lab., University of Montreal, Montreal, Canada
[3]École Polytechnique de Montréal, Montreal, Canada

ABSTRACT

As existing computer search engines struggle to understand the meaning of natural language, semantically enriched metadata may improve interest-based search engine capabilities and user satisfaction.

This paper presents an enhanced version of the ecosystem focusing on semantic topic metadata detection and enrichments. It is based on a previous paper, a semantic metadata enrichment software ecosystem (SMESE). Through text analysis approaches for topic detection and metadata enrichments this paper propose an algorithm to enhance search engines capabilities and consequently help users finding content according to their interests. It presents the design, implementation and evaluation of SATD (Scalable Annotation-based Topic Detection) model and algorithm using metadata from the web, linked open data, concordance rules, and bibliographic record authorities. It includes a prototype of a semantic engine using keyword extraction, classification and concept extraction that allows generating semantic topics by text, and multimedia document analysis using the proposed SATD model and algorithm.

The performance of the proposed ecosystem is evaluated using a number of prototype simulations by comparing them to existing enriched metadata techniques (e.g., AlchemyAPI, DBpedia, Wikimeta, Bitext, AIDA, TextRazor). It was noted that SATD algorithm supports more attributes than other algorithms. The results show that the enhanced platform and its algorithm enable greater understanding of documents related to user interests.

KEYWORDS

Natural Language Processing, Semantic Topic Detection, Semantic Metadata Enrichment, Text and Data Mining

1. INTRODUCTION

The goal of this paper is to increase the findability of document or content matching user interest using an internal semantic metadata enrichment algorithm. Words themselves are often used inconsistently, having a wide variety of definitions and interpretations. Finding bibliographic references or semantic relationships in texts makes it possible to localize specific text segments using ontologies to enrich a set of semantic metadata related to topics. This paper presents an enhanced implementation of SMESE [1] focusing on semantic topic metadata detection and enrichment.

Semantic topic detection (STD), a fundamental aspect of SIR, helps users to efficiently detect meaningful topics. Initial methods for STD relied on clustering documents based on a core group of keywords representing a specific topic, where, based on a ratio such as tf-idf, documents that contain these keywords are similar to each other [2,3]. Next, variations of tf-idf were used to compute keyword-based feature values, and cosine similarity was used as a similarity (or distance) measure to

DOI: 10.5121/ijdkp.2017.7301

cluster documents. The following generation of STD approaches, including those based on latent Dirichlet allocation (LDA), shifted analysis from directly clustering documents to clustering keywords. Some examples of these advances in STD are presented in [4]. Bijalwan et al. [5], for example, experimented with machine learning approaches for text and document mining and concluded that k-nearest neighbors (KNN), for their data sets, showed the maximum accuracy as compared to naive Bayes and term-graph. The drawback for KNN is that time load is high but it demonstrates better accuracy than others.

A number of approaches are used to perform text mining, including: latent Dirichlet allocation (LDA) [4], tf-idf [2,3], latent semantic analysis (LSA) [6], formal concept analysis (FCA) [7], latent tree model (LTM) [8], naïve Bayes (NB) [9], and artificial neural network (ANN) [10]. This paper consists of a model and an algorithm SATD (Scalable Annotation-based Topic Detection) for topic metadata semantic enrichments. SATD allows the generation of semantic topics using text, relationships and documents analysis. Using simulation, the performance of SATD was evaluated in terms of accuracy of topic detection. For comparison, existing approaches that performs semantic metadata enrichment in terms of topic detection and enrichment were evaluated. Simulation results showed that SATD outperforms these existing approaches.

The remainder of the paper is organized as follows. Section 2 presents the related work. Section 3 describes SATD model and algorithm while Section 4 presents the evaluation through different prototypes. Section 5 concludes the paper and presents some future work.

2. RELATED WORK

Generally, a topic is represented as a set of descriptive and collocated keywords/terms. Initially, document clustering techniques were adopted to cluster content-similar documents and extract keywords from clustered document sets as the representation of topics. The predominant method for topic detection is the latent Dirichlet allocation (LDA) [4], which assumes a generating process for the documents. LDA has been proven a powerful algorithm because of its ability to mine semantic information from text data. Terms having semantic relations with each other are collected as a topic. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities.

The literature presents two groups of text-based topic detection approaches based on the size of the text: short text [11,7,12,13] such as tweets or Facebook posts, and long text [14,4,15-17,8] such as a document or a book. For example, Dang et al. [11] proposed an early detection method for emerging topics based on dynamic Bayesian networks in micro-blogging networks. They analyzed the topic diffusion process and identified two main characteristics of emerging topics, namely attractiveness and key-node. Next, based on this identification, they selected features from the topology properties of topic diffusion, and built a DBN-based model using the conditional dependencies between features to identify the emerging keywords. But to do so, they had to create a term list of emerging keyword candidates by term frequency in a given time interval. Cigarran et al. [7] proposed an approach based on formal concept analysis (FCA). Formal concepts are conceptual representations based on the relationships between tweet terms and the tweets that have given rise to them. Cotelo et al. [12], when addressing the tweet categorization task, explored the idea of integrating two fundamental aspects of a tweet: the textual content itself, and its underlying structural information. This work focuses on long text topic detection.

Recently, considerable research has gone into developing topic detection approaches using a number of information extraction techniques (IET), such as lexicon, sliding window, boundary techniques, etc. Many of these techniques [14,15,17,8] rely heavily on simple keyword extraction from text. For example, Sayyadi and Raschid [14] proposed an approach for topic detection, based on keyword-based methods, called KeyGraph, that was inspired by the keyword co-occurrence graph and efficient graph analysis methods. The main steps in the KeyGraph approach are as follows:

1. The first step is construction of a keyword co-occurrence graph, called a KeyGraph, which has one node for each keyword in the corpus and where edges represent the co-occurrence of the corresponding keywords weighted by the count of the co-occurrences.
2. Secondly, making use of an off-the-shelf community detection algorithm, community detection is taken into account where each community forms a cluster of keywords that represent a topic. The weight of each keyword in the topic feature vector is computed using the tf-idf formula. The TF value is computed as the average co-occurrence of each keyword from the community with respect to the other keywords in that community.
3. Then, to assign a topic to a document, the likelihood of each topic t with the vector of keyword f_t is computed using the cosine similarity of the document.
4. Finally, for each pair of topics, where multiple documents are assigned to both topics, it is assumed that these are subtopics of the same parent topic and are therefore merged.

In other words, KeyGraph is based on the similarity of keyword extraction from text. We note two limitations to the approach, which requires improvement in two respects. Firstly, they failed to leverage the semantic information derived from topic model. Secondly, they measured co-occurrence relations from an isolated term-term perspective; that is, the measurement was limited to the term itself and the information context was overlooked, which can make it impossible to measure latent co-occurrence relations. Salatino and Motta [17] suggested that it is possible to forecast the emergence of novel research topics even at an early stage and demonstrated that such an emergence can be anticipated by analyzing the dynamics of pre-existing topics. They presented a method that integrates statistics and semantics for assessing the dynamics of a topic graph: (1) first, they select and extract portions of the collaboration networks related to topics in the two groups a few years prior to the year of analysis. Based on these topics, they build a topics graph where nodes are the keywords while edges are the links representing co-occurrences between keywords and (2) next, they transform the graphs into sets of 3-cliques. For each node of a 3-clique, they compute the weight associated with each link between pairs of topics by using the harmonic mean of the conditional probabilities. While this is a satisfactory approach to find latent co-occurrence relations, the approach assumes that keywords are topics. Chen et al. [8] proposed a novel method for hierarchical topic detection where topics are obtained by clustering documents in multiple ways. They used a class of graphical models called hierarchical latent tree models (HLTMs). Latent tree models (LTMs) are tree-structured probabilistic graphical models where the variables at leaf nodes are observed and the variables at internal nodes are latent. It is a Markov random field over an undirected tree carried out as follows: (1) first, the word variables are partitioned into clusters such that the words in each cluster tend to co-occur and the co-occurrences can be properly modeled using a single latent variable. The authors achieved this partition using the BUILDISLANDS subroutine, which is based on a statistical test called the uni-dimensionality test (UD-test) and (2) after the islands are created, they are linked up so as to obtain a model over all the word variables. This is carried out by the BRIDGEISLANDS subroutine, which estimates the mutual information between each pair of latent variables in the islands. This allows construction of a complete undirected graph with the mutual information values as edge weights, and finally the maximum spanning tree of the graph is determined [8]. Hurtado et al. [18] proposed an approach that uses sentence-level association rule mining to discover topics from documents. Their method considers each sentence as a transaction and keywords within the sentence as items in the transaction. By exploring keywords (frequently co-occurring) as patterns, their method preserves contextual information in the topic mining process. For example, whenever the terms: “machine”, “support” and “vector” are discovered as strongly correlated keywords, either as “support vector machine” or “support vector”, they assumed that these patterns were related to one topic, i.e., “SVM”. In order to discover a set of strongly correlated topics, they used the CPM-based community detection algorithm to find groups of topics with strong correlations. As in [8], their contribution was limited to simulating existing algorithms. Zhang et al. [15] proposed LDA-IG, an extension of KeyGraph [14]. It is a hybrid relations analysis approach integrating semantic relations and co-occurrence relations for topic detection. Specifically, their approach fuses multiple types of relations into a uniform term graph by incorporating idea discovery theory with a topic modeling method.

1. Firstly, they defined an idea discovery algorithm called IdeaGraph that was adopted to mine latent co-occurrence relations in order to convert the corpus into a term graph.
2. Next, they proposed a semantic relation extraction approach based on LDA that enriches the graph with semantic information.
3. Lastly, they make use of a graph analytical method to exploit the graph for detecting topics. Their approach has four steps: (a) Pre-processing to filter noise and adjust the data format suitable for the subsequent components, (b) Term graph generation to convert the basket dataset into a term graph by extracting co-occurrence relations between terms using the Idea Discovery algorithm, (c) Term graph refining with semantic information using LDA to build semantic topics and tp-izp, inspired by tf-idf, to measure the semantic value of any term in each topic, and (d) Topic extraction from the refined term graph by assuming that a topic is a filled polygon and measuring the likelihood of a document d being assigned to a topic using tf-idf. However, their approach does not include machine learning.

From our review of related work, we conclude that the main drawbacks of existing approaches to topic detection are as follows:

1. They are based on simple keyword extraction from text and lack semantic information that is important for understanding the document. To tackle this limitation, our work uses semantic annotations to improve document comprehension time.
2. Co-occurrence relations across the document are commonly neglected, which leads to incomplete detection of information. Current topic modeling methods do not explicitly consider word co-occurrences because of a computational challenge. The graph analytical approach to this extension was only an approximation that merely took into account co-occurrence information alone while ignoring semantic information. How to combine semantic relations and co-occurrence relations to complement each other remains a challenge.
3. Existing approaches focus on detecting prominent or distinct topics based on explicit semantic relations or frequent co-occurrence relations; as a result, they ignore latent co-occurrence relations. In other words, latent co-occurrence relations between two terms cannot be measured from an isolated term-term perspective. The context of the term needs to be taken into account.
4. More importantly, even though existing approaches take into account semantic relations, they do not include machine learning to find new topics automatically.

The main conclusion is that most of the existing related research is limited to simulations using existing algorithms. None contribute improvements to detect topics more accurately.

Table 1 compares the most known text mining algorithms (e.g., AlchemyAPI, DBpedia, Wikimeta, Bitext, AIDA, TextRazor) with our proposed algorithm in SMESE V3 by keyword extraction, classification and concept extraction.

Table 1. Summary of attribute comparison of existing and SATD algorithms.

Existing algorithms	Keyword extraction	Classification	Concept extraction
AlchemyAPI (http://www.alchemyapi.com/)	x	x	x
DBpedia Spotlight (https://github.com/dbpedia-spotlight)			x
Wikimeta (https://www.w3.org/2001/sw/wiki/Wikimeta)			x
Yahoo! Content Analysis API (out of date) (https://developer.yahoo.com/contentanalysis/)		x	x
Tone Analyzer (https://tone-analyzer-demo.mybluemix.net/)			
Zemanta (http://www.zemanta.com/)			x
Receptiviti (http://www.receptiviti.ai/)			
Apache Stanbol (https://stanbol.apache.org/)			x
Bitext (https://www.bitext.com/)			x

Mood patrol (https://market.mashape.com/ soulhackerslabs/ mood patrol-emotion-detection-from-text)			
Aylien (http://aylien.com/)	x	x	
AIDA (http://senseable.mit.edu/aida/)			x
Wikifier (http://wikifier.org/)			x
TextRazor (https://www.textrazor.com/)			x
Synesketch (http://krcadinac.com/synesketch/)			
Toneapi (http://toneapi.com/)			
SATD algorithm	x	x	x

3. RULE-BASED SEMANTIC METADATA INTERNAL ENRICHMENT ENGINE

This section presents an overview and details of the proposed rule-based semantic metadata internal enrichment engine, including the model and algorithm (SATD) used to process semantic metadata internal enrichment for topic.

The goal of this paper is to extend the SMESE platform [1] through text analysis approaches for topic detection and metadata enrichments. To perform this task, the following tools are needed: (1) topics are a controlled set of terms designed to describe the subject of a document. While topics do not necessarily include relationships between terms, we include relationships as triplets (Entity – Relationship – Entity); for example, Entity "*Ronald*" - relationship:" *likes* " - Entity "*Le petit prince*", and (2) an ontology to provide a representation of knowledge with rich semantic relationships between topics. By breaking content into pieces of data, and curating semantic relationships to external contents, metadata enrichments are created dynamically.

3.1. Rule-based semantic metadata internal enrichment engine overview

The rule-based semantic metadata internal enrichment engine has been designed to find short descriptions, in terms of topics of the members of a collection to enable efficient processing of large collections while preserving the semantic and statistical relationships. Figure 1 shows an overview of the architecture that consists of: (1) User interest-based gateway, (2) Metadata initiatives & concordance rules, (3) Harvesting web metadata & data, (4) User profiling engine and (5) Rule-based semantic metadata internal enrichment engine. The user interest-based gateway is designed to push notifications to users based on the topics found using the user-profiling engine. The rule-based semantic metadata internal enrichment engine performs automated metadata internal enrichment based on the set of metadata initiatives & concordance rules, the engine for harvesting web metadata, the user profile and a thesaurus.

The following sub-sections present the terminology and assumptions, and details of the SATD algorithm.

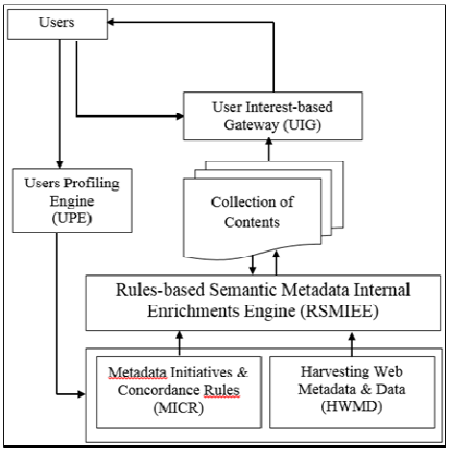


Figure 1. Rule-based semantic metadata internal enrichment engine architecture

3.2. Terminology and assumptions

In this section the following terms are defined:

1. A word or term is the basic unit of discrete data, defined to be an item from a vocabulary indexed by {1, …,V}. Terms are presented using unit-basis vectors that have a single component equal to one and all other components equal to zero. Thus, using superscripts to denote components, the i^{th} term in the vocabulary is represented by an I-vector *w* such that $w^i = 1$ and $w^j = 0$ for $i \neq j$. For example, let V= {book, image, video, cat, dog} be the vocabulary. The video term is represented by the vector (0, 0, 1, 0, 0).
2. A line is a sequence of N terms denoted by *l*. These terms are extracted from a real sentence; a sentence is a group of words, usually containing a verb, that expresses a thought in the form of a statement, question, instruction, or exclamation and when written begins with a capital letter.
3. A document is a sequence of N lines denoted by $D = (w_1,w_2; \ldots,w_N)$, where w_i is the i^{th} term in the sequence coming from the lines. D is represented by its lines as $D = (l_1, \ldots l_i,\ldots,l_K)$.
4. A corpus is a collection of *M* documents denoted by $C = \{D_1, D_2, \ldots, D_M\}$.
5. An emotion word is a word with strong emotional tendency. An emotion word is a probabilistic distribution of emotions and represents a semantically coherent emotion analysis. For example, the word "excitement", presenting a positive and pleased feeling, is assigned a high probability to emotion "joy".

To implement the SATD algorithm, an initial set of conditions must be established:

1. A list of topics $T = \{t_1, \ldots , t_i, \ldots , t_n\}$ is readily available.
2. Each existing document D_j is already annotated by topic. The annotated topics of document D_j are denoted as $T_{Dj} = \{t_p \ldots, t_i , \ldots, t_q\}$ where t_p, t_i, and $t_q \in T$.
3. The corpus of documents is already classified by topics. $C_{ti}=\{\ldots,D_j,\ldots\}$ denotes the corpus of documents that have been annotated with topic t_i. Note that the document D_j may be located in several corpuses.
4. A list of sentiments $S = \{s_1, \ldots , s_i, \ldots , s_S\}$ is readily available.
5. A thesaurus is available and has a tree hierarchical structure.

3.3. Document pre-processing

The objective of the pre-processing is to filter noise and adjust the data format to be suitable for the analysis phases. It consists of stemming, phase extraction, part-of-speech filtering and removal of stop-words. The corpus of documents crawled from specific databases or the internet consists of many documents. The documents are pre-processed into a basket dataset C, called document collection. C consists of lines representing the sentences of the documents. Each line consists of terms, i.e. words or phrases. 'Word' and 'term' are used interchangeably in the rest of this paper.

More specifically, to obtain D_j, the following preprocessing steps are performed: (1) Language detection, (2) Segmentation: a process of dividing a given document into sentences, (3) Stop word: a process to remove the stop words from the text. Stop words are frequently occurring words such as 'a' an', the' that provide less meaning and generate noise. Stop words are predefined and stored in an array, (4) Tokenization: separates the input text into separate tokens, (5) Punctuation marks: identifies and treats the spaces and word terminators as the word breaking characters, and (6) Word stemming: converts each word into its root form by removing its prefix and suffix for comparison with other words. More specifically, a standard preprocessing such as tokenization, lowercasing and stemming of all the terms using the Porter stemmer [19]. Therefore, we also parse the texts using the Stanford parser [20] that is a lexicalized probabilistic parser which provides various information such as the syntactic structure of text segments, dependencies and POS tags.

3.4. Scalable annotation-based topic detection: SATD

The aim of SATD is to build a classifier that can learn from already annotated documents and infer the topics. Traditional approaches are typically based on various topic models, such as latent Dirichlet allocation (LDA) where authors cluster terms into a topic by mining semantic relations between terms. Furthermore, the inability to discover latent co-occurrence relations via the context or other bridge

terms prevents important but rare topics from being detected. SATD combines semantic relations between terms and co-occurrence relations across the document making use of document annotation. In addition, SATD includes: (1) a probabilistic topic detection approach that is an extension of LDA, called BM semantic topic model (BM-SemTopic) and (2) a clustering approach that is an extension of KeyGraph, called BM semantic graph (BM-SemGraph).

SATD is a hybrid relation analysis and machine learning approach that integrates semantic relations, semantic annotations and co-occurrence relations for topic detection. More specifically, SATD fuses multiple relations into a term graph and detects topics from the graph using a graph analytical method. It can detect topics not only more effectively by combing mutually complementary relations, but also mine important rare topics by leveraging latent co-occurrence relations.

SATD is composed of five phases: (1) relevant and less similar documents selection process phase, (2) not annotated documents semantic term graph generation process phase, (3) topics detection process phase, (4) training process phase and (5) topics refining process phase. The following sub-sections present the details of the five phases of the SATD model.

3.4.1. Relevant and less similar documents selection - process phase

For a given topic, a filtering process is performed to avoid using a large corpus of documents that are similar or not relevant. For this reason, only relevant and less similar documents within a corpus are identified. Here, only documents that are already annotated by topic are considered.
An overview of the architecture of the relevant and less similar document selection phase is presented in Figure 2. This phase involves three algorithms:

1. Algo 1 identifies the relevant documents for a given topic.
2. Algo 2 detects less similar documents in the relevant set of documents.
3. Algo 3 ascertains whether the new annotated document with a topic is relevant and less similar to a sub set of relevant and less similar documents of this topic.

First, the most relevant documents of each topic t_i are selected. For each document of a topic t_i, Algo 1 checks whether its most important terms are the same as the most important terms of the topic t_i. To identify the most important terms of a given document D_j, the tf-idf of each term W_i in the corpus C_{ti} is computed using equation (1):

$$f(W_i, D_j, C_{ti}) = TF - IDF(W_i, D_j, C_{ti}) = TF(W_i, D_j) * \log\left(\frac{|C_{ti}| = M_i}{IDF(W_i, C_{ti})}\right) \quad (1)$$

where $TF(W_i, D_j)$, $IDF(W_i, C_{ti})$ and M_i denote the number of occurrences of W_i in document D_j, the number of documents in the corpus C_{ti} where W_i appears, and the number of documents in the corpus C_{ti}, respectively.

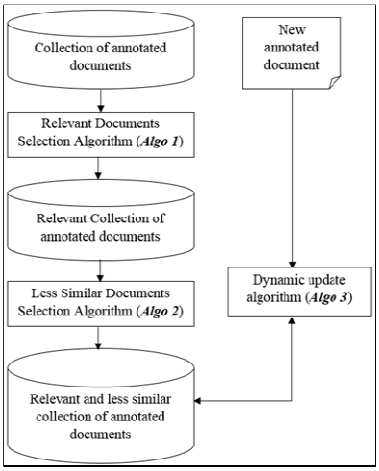


Figure 2. Relevant and less similar document selection process phase – Architecture overview

Equation (1) allows SATD to find, for each document D_j, the vector V_{Dj}= { $(W_a, f(W_a,D_j,C_{ti}))$, …, $(W_i, f(W_i,D_j,C_{ti}))$,…, $(W_{|Dj|}, f(W_{|Dj|},D_j,C_{ti}))$} where in the couple $(W_i, f(W_i,D_j,C_{ti}))$, W_i denotes a term and $f(W_i,D_j,C_{ti}))$ its tf-idf in the whole corpus C_{ti}. To identify the most important terms of a given topic t_i, the tf-itf of each term W_k that appears at least one time in at least one document of corpus C_{ti} is computed with formula (2):

$$g(W_k, t_i) = TF - ITF(W_k, t_i) = TF(W_k, t_i) * \log(\frac{|T| = n}{ITF(W_k)}) \quad (2)$$

where $TF(W_k, t_i)$, $ITF(W_k)$ and |T| denote the number of occurrences of W_k in all the documents of corpus C_{ti}, the number of topics where W_k appears, and the number of topic, respectively.

Equation (2) allows SATD to find, for each topic ti, the vector Vti= { (W1 , g(W1,ti)), …, (Wk, g(Wk,ti)) ,…, (WNi , g(WNi,ti))} where in the couple (Wk , g(Wk,ti)), Wk denotes a term and g(Wk,ti) its tf-itf in the whole corpus T.

Let N_i be the number of terms of the vocabulary of C_{ti} and N_{Dj} = |Dj| be the number of terms of the vocabulary of D_j. In this context, N_i is larger than N_{Dj}. To determine the number of terms to consider the document relevant, SATD computes the standard deviation σ and the average *avg* of the number of distinct terms in the documents for the topics. SATD uses the standard deviation. The standard deviation σ_{ti} of topic t_i is given by equation (3):

$$\sigma_{ti} = \sqrt{\frac{\sum_{j=1}^{|C_{ti}|=M_i}(|D_j| - avg_{ti})^2}{|C_{ti}| = M_i}} \quad (3)$$

where the average number of terms avg_{ti} of topic t_i is computed using equation (4).

$$avg_{ti} = \frac{\sum_{j=1}^{|C_{ti}|=M_i}|D_j|}{|C_{ti}| = M_i} \quad (4)$$

Next, to compute the number of distinct terms to consider, SATD uses equation (5).

$$E_{ti} = avg_{ti} - \sigma_{ti} \quad (5)$$

The score for each document D_j in the topic t_i is computed next:

1. SATD sorts, for each document D_j of corpus C_{ti}, the vector V_{Dj} by $f(W_i,D_j,C_{ti})$ in descending order.
2. SATD computes the BMscore of D_j using equation (6):

$$BMscore(D_j) = \sum_{|E_i|} g(W_i, t_i) \quad (6)$$

where $\sum_{|Ei|}$ are the first $|E_i|$ terms W_i of D_j with the highest value of $f(W_i,D_j,C_{ti})$ in the whole corpus C_{ti}.

In order terms, BMscore is the summation of the tf-itf in the whole corpus C of the first $|E_i|$ terms W_i of D_j with the highest tf-idf in the whole corpus C_{ti}. Finally, based on the BMscore of each document D_j of corpus C_{ti}, SATD selects the most relevant documents of corpus C_{ti}. SATD obtains the sub-corpus C'_{ti} of the most relevant documents using equation (7):

$$C_{ti} = \left[C'_{ti} = \bigcup_{\alpha}\{D_k\}\right] \cup \left[\bigcup_{M_i-\alpha}\{D_j\}\right] \quad (7)$$

where BMscore (Dk) > BMscore (Dj).

Note that α is a threshold determined by empirical experimentation based on the particular document collection. $C'_{ti} = \{D_{k_*}, \ldots, D_{k_i}, \ldots, D_{k_\sigma}\}$ is obtained where $M_i \geq M'_i = \alpha$. Algorithm 1 of appendix A explains, in detail, the selection process of relevant documents for a given topic.
The less similar documents of sub-corpus C'_{ti} for the topic t_i are then selected. SATD defines a similarity threshold β by empirical experimentation based on the particular document collection where C''_{ti} is the sub-corpus of C'_{ti} that contains the less similar documents.

SATD sorts the documents of C'_{ti} according to their BMscore. SATD first puts the document with the largest BMscore in C''_{ti}; then, based on the order of largest BMscore, SATD compares the semantic similarity of each element of C''_{ti} with the rest of element of $C'_{ti}C'_{ti}$. If no document of C''_{ti} is semantically similar to a given document of C'_{ti}, this given document is added to C''_{ti}. When the semantic similarity between two documents is less than or equal to β, SATD assumes they are not similar. Finally, when a new document annotated with topic t_i, is added to the corpus C_{ti}, SATD computes its BMscore in order to ascertain whether this new document must be added to C''_{ti} or not.

For example, let IDF^s_{ti} be the idf vector of the vocabulary of corpus C_{ti} at state *s* and ITF^s be the itf vector of the vocabulary of corpus C at state *s*. The state is the situation of the collection before adding the new document:

$IDF^s_{ti} = (IDF(W_1, C_{ti}), \ldots, IDF(W_k, C_{ti}), \ldots, IDF(W_{Ni}, C_{ti}))$ and

$ITF^s = (ITF(W_1), \ldots, ITF(W_k), \ldots, ITF(W_{Ni}))$. Let TF^s_{ti} be the tf vector of the vocabulary of corpus C_{ti} at the state *s*:

$$TF^s_{ti} = (TF(W_1, t_i), \ldots, TF(W_k, t_i), \ldots, TF(W_{Ni}, t_i)).$$

Based on vector IDF^s_{ti}, SATD computes the TF-IDF of each term *W* of *d* of each term w of *d* using Equation (8):

$$f(W, d, C_{ti}) = TF - IDF(W, d, C_{ti}) = TF(W, d) * \log\left(\frac{|C_{ti}|}{IDF(W, C_{ti}) + 1}\right) \quad (8)$$

Next, SATD ranks the vocabulary of d according to their $f(W, d, C_{ti})$ and selects the E_{ti} terms *W* of d with highest $f(W, d, C_{ti})$. Based on the vectors ITF^s_{ti} and TF^s_{ti}, SATD computes the TF-ITF of each selected term *W* of *d* using equation (9):

$$g(W, t_i) = TF - ITF(W, t_i) = [TF(W, t_i) + TF(W, d)] * log\left(\frac{|T|}{ITF(W_k)}\right) \quad (9)$$

SATD obtains the BMscore(d) of new document *d* by summation of the g(W,t_i) term. If BMscore(d) is greater than the smallest BMscore of C'_{ti} document, SATD uses Algorithm 2 to make a semantic similarity computation and then performs an update of C''_{ti} if necessary.

3.4.2. Not annotated documents semantic term graph generation - process phase

The semantic term graph allows one to convert a set of lines of terms into a graph by extracting semantic and co-occurrence relations between terms. To generate the semantic term graph BM-SemGraph: (1) first the co-occurrence clusters are generated and then optimized, (2) after optimization, the key terms and links between the clusters are extracted and (3) finally, the semantic topic is generated and semantic term graph extracted.

The BM-SemGraph has one node for each term in the vocabulary of the document. Edges in a BM-SemGraph represent the co-occurrence of the corresponding keywords and are weighted by the count of the co-occurrences. Note that, in contrast to existing graph-based approaches, the co-occurrence between A and B is different from the co-occurrence between B and A. This difference allows one to retain the semantic sense of co-occurrence terms. Figure 3 presents an overview of the architecture of

the semantic term graph generation process phase. The term graph process and BM-SemTopic process generate the semantic graph in order to enrich the term graph with semantic information; indeed, the terms graph and semantic graph are merged to provide Semantic term graph, called BM-SemGraph.

The term graph process consists of three steps: (1) Co-occurrence clusters generation, (2) Clusters optimization and (3) Key terms extraction. The BM-SemTopic process consists of two steps: (1) Semantic topic generation and (2) Semantic graph extraction.

Step 1: Co-occurrence clusters generation

For the co-occurrence graph, the assumption is that terms that have a close relation to each other may be linked by the co-occurrence link. The relation between two terms W_i and W_j is measured by their conditional probability. Let D be a document and $V_D = (w_1, w_2; \ldots, w_N)$ be the terms of D and L_D be the number of lines of D.

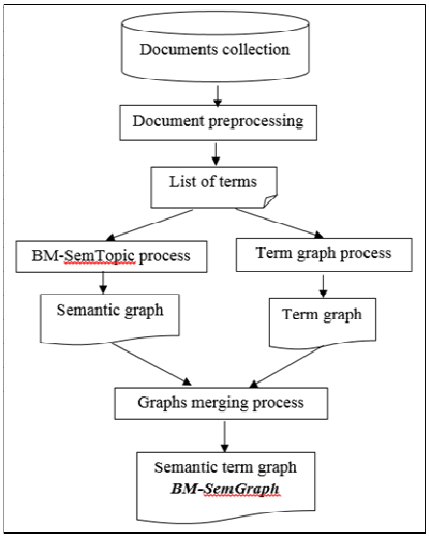


Figure 3. New document semantic term graph process phase - Architecture overview

The conditional probability $p(\overrightarrow{W_i, W_j}^{\varepsilon})$ of $\overrightarrow{W_i, W_j}^{\varepsilon}$ is computed using equation (10) where ε (determined by experimentation) denotes the minimum distance between W_i and W_j and the distance between two terms is the number of terms that appear between them for a given line.

$$p\left(\overrightarrow{W_i, W_j}^{\varepsilon}\right) = \sum_{l=1}^{L_D} \frac{N^{line\ l}\left(\overrightarrow{W_i, W_j}^{\varepsilon}\right)}{\left\lceil \frac{N(line\ l)}{\varepsilon} \right\rceil} \tag{10}$$

where $N^{line\ l}\left(\overrightarrow{W_i, W_j}^{\varepsilon}\right)$ denotes the number of times that W_i and W_j co-occur with a minimum distance ε and where W_i appears before W_j, and *N(line l)* denotes the number of terms of the line l.
To formally define a relation between two terms W_i and W_j, their frequent co-occurrence measured by the conditional probability $p(\overrightarrow{W_i, W_j}^{\varepsilon})$, needs to exceed the co-occurrence threshold. The co-occurrence threshold is also determined by experimentation. Note that frequent co-occurrence is oriented. This allows one to retain the semantic orientation of the links between terms.

Next, the oriented links are transformed into simple links without losing the semantic context. To perform this transformation, three rules are applied - see Figure 4.

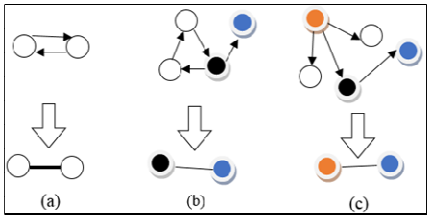


Figure 4. Link transformation rules

In Figure 4a, two nodes with two oriented links are transformed into one simple link. In this case, this type of link cannot be pruned and its weight is given by equation (11):

$$w(W_i, W_j) = p\left(\overrightarrow{W_i, W_j}^{\varepsilon}\right) \mid p\left(\overrightarrow{W_j, W_i}^{\varepsilon}\right) \tag{11}$$

In Figure 4b, where several nodes are linked by oriented links and there is an oriented path to join each of them, only the nodes with a link to other nodes not in the oriented path are retained. The black node becomes the representative of the other nodes.

In Figure 4c, where one node A is linked to several nodes and the links are oriented from A towards the other nodes, node A becomes the representative of the other nodes and the other nodes are removed. This is the case for the red node where the link between the black node and blue node is removed and a new link is added between the red node and the blue node. Let G be a set of nodes where W_i is the representative node. Let G' be the sub set of G which are linked to a node W_j not in G. Figure 5 illustrates G and G'. The weight of the link between W_i and W_j is given by equation (12):

$$w(W_i, W_j) = \sum_{W_k \in G'} p\left(\overrightarrow{W_k, W_j}^{\varepsilon}\right) + p\left(\overrightarrow{W_j, W_k}^{\varepsilon}\right) \tag{12}$$

Equation (12) is applied in the case of Figure 4b and 4c to compute the weight of the link between a representative node and another node. Finally, the rest of the oriented links are transformed into simple links and their weights computed using equation (11).

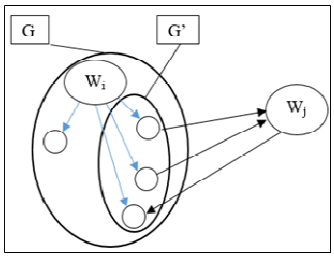


Figure 5. Representation of the computation of weight after removing some nodes

Step 2: Cluster optimization

To enhance quality, clusters should be pruned, such as by removing weak links or partitioning sparse cluster into cohesive sub-clusters. Clusters are pruned according to their connectedness. The link e is pruned when no path connects the two ends of e after it is pruned. As shown in Figure 6, the link between the black node and the green node should be pruned.

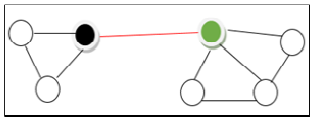


Figure 6. Clusters optimization

Secondly, cliques are identified. In graph theory, a clique is a set of nodes which are adjacent pairs (?) (or a two-by-two set of nodes?) as shown in Figure 7.

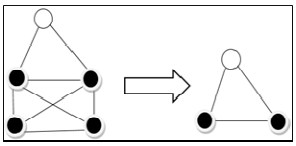


Figure 7. Clique reduction

Let C be the clique and W_i and W_j be the nodes of C that are linked to another node. The weight between W_i and W_j is given by equation (13):

$$w(W_i, W_j) = \underset{W_k \in C \\ W_s \in C}{\text{MAX}}[w(W_k, W_s)] \tag{13}$$

Step 3: Key term extraction

To extract key terms, the relation between a term and a cluster is measured. It is assumed that the weight of a term in a given cluster may be used to determine the importance of this term for the cluster. Let R be the set of nodes of the cluster C where the node W_i is inside. The weight of W_i in the cluster C is given by equation (14):

$$f(W_i) = \sum_{W_j \in R} w(W_i, W_j) \tag{14}$$

To identify a term as a key term, a sort of terms is performed based on their weights regardless of the clusters that they are in. Next, the NumKeyTerm terms that have the largest weights are selected as Key Terms. NumKeyTerm is a parameter.

Step 4: Semantic topic generation

Semantic topic generation combines a correlated topic model (CTM) [21] and a domain knowledge model (DKM) [22], called BM semantic topic model (BM-SemTopic), to build the real semantic topic model. In LDA, a topic is a probability distribution over a vocabulary. It describes the relative frequency each word is used in a topic. Each document is regarded as a mixture of multiple topics and is characterized by a probability distribution over the topics.

A limitation of LDA is its inability to model topic correlation. This limitation stems from the use of the Dirichlet distribution to model the variability among topic proportions. In addition, standard LDA does not consider domain knowledge in topic modeling. To overcome these limitations, BM-SemTopic combines two models: (1) A correlated topic model (CTM) [21] that makes use of a logistic normal distribution and (2) A domain knowledge model (DKM) [22] that makes use of the Dirichlet distribution.

BM-SemTopic uses a weighted sum of CTM and DKM to compute the probability distribution of term W_i on the topic z. The sum is defined by equation (15):

$$h(W_i|z) = \omega CTM(W_i|z) + (1 - \omega)\, DKM(W_i|z) \tag{15}$$

where ω is used to give more influence to one model based on the term distribution of topics.

When the majority of terms are located in a few topics, this means the domain knowledge is important and ω must be small. BM-SemTopic develops the CTM where the topic proportions exhibit a correlation with the logistic normal distribution and incorporates the DKM. A key advantage of BM-SemTopic is that it explicitly models the dependence and independence structure among topics and words, which is conducive to the discovery of meaningful topics and topic relations.

CTM is based on a logistic normal distribution. The logistic normal is a distribution on the simplex that allows for a general pattern of variability between the components by transforming a multivariate normal random variable. This process is identical to the generative process of LDA except that the topic proportions are drawn from a logistic normal distribution rather than a Dirichlet distribution. The

strong independence assumption imposed by the Dirichlet in LDA is not realistic when analyzing document collections where one may find strong correlations between topics. To model such correlations, the covariance matrix of the logistic normal distribution in the BM-SemTopic correlated topic model is introduced.

DKM is an approach to incorporation of such domain knowledge into LDA. To express knowledge in an ontology, BM-SemTopic uses two primitives on word pairs: Links and Not-Links. BM-SemTopic replaces the Dirichlet prior by the Dirichlet Forest prior in the LDA model. Then, BM-SemTopic sorts the terms for every topic in descending order according to the probability distribution of the topic terms. Next it picks up the high-probability terms as the feature terms. For each topic, the terms with probabilities higher than half of the maximum probability distribution are picked up (experiment indicates it is non-sensitive on this parameter).

Step 5: Semantic term graph extraction

To enrich the term graph, the semantic topic needs to be converted into a semantic graph that consists of semantic relations between the semantic terms. To discover these relations, the semantic aspect is included making use of WordNet::Similarity [23]. Based on the structure and content of the lexical database WordNet, WordNet::Similarity implements six measures of similarity and three measures of relatedness. Measures of similarity use information found in a hierarchy of concepts (or synsets) that quantify how much concept A is like (or is similar to) concept B.

First, each generated feature term at step 4 is the candidate for a semantic term where it is assumed the other terms represent the vocabulary associated with the semantic topic. In Figure 8a, the blue node denotes the feature terms of each semantic topic. Next, duplicate terms from the candidates are removed. If there is more than one topic that has the same term W_j in the semantic term candidate, only the topic z with the highest term probability distribution $h(W_j|z)$ is retained W_j as the semantic term candidate. It follows then that following this step the semantic term candidates of different topics are exclusive to each other. Figure 8b shows the remaining candidates by semantic topic.

To remove similar terms, the measure path (one measure of similarity of WordNet::Similarity [23]) is used to evaluate similarity between two terms. The measure path of WordNet::Similarity is a baseline that is equal to the inverse of the shortest path between two concepts. When the semantic term candidates of different topics are identified, the semantic value of each topic's candidates is computed. The semantic value of each term W_i, is given by equation (16):

$$SEM(W_i|z) = TP - ITP(W_k|z) = h(W_i|z) * \log\left(\frac{|Z|}{\sum_{t \in Z} h(W_i|t)}\right) \quad (16)$$

where Z denotes the set of semantic topics. TP-ITP is inspired by the tf-idf formula, where TP is term probability and ITP inverse topic probability.

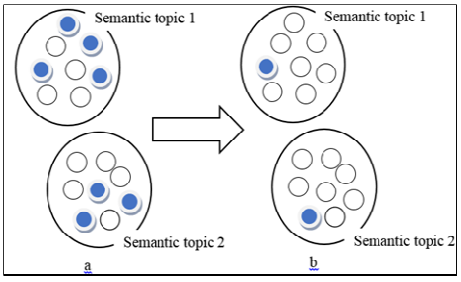


Figure 8. Candidates for semantic term identification (a and b)

Semantic links between semantic terms for the term graph are constructed using the vector measure, one of the measures of relatedness of WordNet::Similarity [23]. The vector measure creates a co–occurrence matrix for each word used in WordNet glosses from a given corpus, and then represents each gloss/concept with a vector that is the average of these co–occurrence vectors.

Let W_i and W_j be semantic terms of the synsets A and B, respectively. Let $\vec{A} = (a_1, \ldots, a_q)$ and $\vec{B} = (b_1, \ldots, b_q)$ be the co–occurrence vectors of A and B, respectively. Let V_z be the set of semantic terms of the semantic topic Z. The weight of the link between W_i and W_j is computed by equation (17):

$$Dis(W_i, W_j \mid z) = \frac{SEM(W_i|z) + SEM(W_j|z)}{\sum_{W_k \in V_Z} SEM(W_k|z)} \times \sqrt{\sum_{l=1}^{n} (a_l - b_l)^2} \quad (17)$$

To discover a semantic relation between two terms, the semantic distance is computed. The semantic distance between two terms is the shortest path between the terms using equation (18):

$$SEMDis(W_i, W_j \mid z) = \underset{pa \in P}{\text{MIN}} \left[\sum_{W_k \in pa} Dis(W_i, W_k \mid z) \right] \quad (18)$$

where *pa*, W_k, and *P* denote a path between W_i and W_j in the thesaurus, a term on a path *pa* and the set of paths *pa* between W_i and W_j, respectively.

To formally define a semantic relation between two terms W_i and W_j, the semantic distance $SEMDis(W_i, W_j \mid z)$ must not exceed the semantic threshold. The semantic threshold is determined by experimentation.

The last process to generate the semantic term graph BM-SemGraph is a merging of the term graph and the semantic graph. The term graph and semantic graph are merged by coupling the co-occurrence relation and the semantic relation. New terms are added as semantic terms and new links are added as semantic links if they do not appear in the term graph. For each link between two nodes W_j and W_k of the merged graph, the weight, called the BM Weight (BMW), for a given topic t_i is computed using equation (19):

$$BMW(W_j, W_k \mid t_i) = \frac{\lambda}{SEMDis(W_j, W_k \mid t_i)} + (1 - \lambda) \times w(W_i, W_j) \quad (19)$$

where λ determined by experimentation.

In order to optimize the clusters of BM-SemGraph, the weak links or partitioning of sparse clusters are removed. At this step, each cluster is considered a topic and the terms of the cluster become the terms of the topic.

3.4.3. Topic detection - process phase

Figure 9 presents the process used by SATD to assign topics to a document.

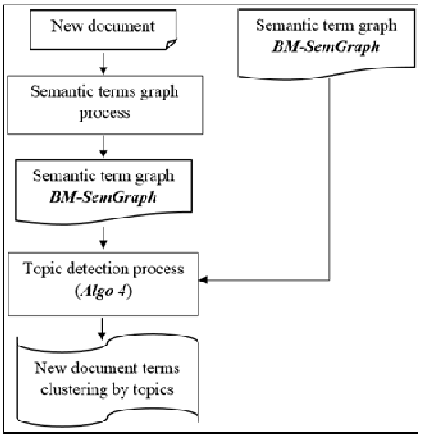


Figure 9. Topic detection process phase - Architecture overview

Topics that may be associated with a new document are detected based on the BM-SemGraph. Note that the BM-SemGraph is obtained using a collection of documents. In this case, the likelihood of detecting topics among a collection of documents is high and must be computed. To accomplish this, the feature vector of each topic based on the clusters of BM-SemGraph is computed. The feature vector of a topic is calculated using the BMRank of each topic term. Let A be the set of nodes of BM-SemGraph directly linked to term W_j in the topic t_i. The score for the term W_j is given by equation (20):

$$BMRank(W_j|t_i) = \frac{\sum_{W_k \in A} BMW(W_j, W_k \mid t_i)}{|A|} \quad (20)$$

The term with the largest BMRank is called the main term of the topic; other terms are secondary terms. The same processes are used to obtain the BM-SemGraph of an individual document *d* and the feature vectors of topics t_j^d. Next, the similarity between each topic t_i and the topics t_j^d of document *d* is computed in order to detect document topics. Let W_i be a master term of topics t_j^d and a master or secondary term of t_i, B be the intersection of the set of terms of BM-SemGraph directly linked to term W_j in the cluster of topic t_i and the set of terms of BM-SemGraph of individual document *d* directly linked to term W_j in the cluster of topic t_j^d, and C be the union of the set of terms of BM-SemGraph directly linked to term W_j in the cluster of topic t_i and the set of terms of BM-SemGraph of individual document *d* directly linked to term W_j in the cluster of topic t_j^d. The similarity between t_i and topic t_j^d is computed with equation (21):

$$Sim(t_i|t_j^d) = \frac{\sqrt{\sum_{W_k \in B} \left(BMW(W_i, W_k \mid t_i) - BMW(W_i, W_k \mid t_j^d)\right)^2}}{\sqrt{\sum_{W_h \in C} \left(BMW(W_i, W_h \mid t_i) - BMW(W_i, W_h \mid t_j^d)\right)^2}} \quad (21)$$

Here, t_i and topic t_j^d are considered to be similar when their similarity $Sim(t_i|t_j^d)$ does not exceed the vector similarity threshold. Finally, the document d is assigned to topics that are similar to its feature vectors.

3.4.4. Training - process phase

The training process establishes a terms graph based on the relevant and less similar documents for a given topic t_i. To form the terms graph for a given topic, preprocessing of its relevant and less similar documents is first carried out, a set of lines is obtained where each line is a list of terms, and the co-occurrence of these terms is then computed. Let Doc be a document and $V_{Doc} = (w_1, w_2; \ldots, w_N)$ be the terms of Doc. The co-occurrence of $co\left(\overleftarrow{W_i, W_j}^{\varepsilon}\right)$ of W*i* and W_j where ε denotes the minimum distance between W_i and W_j is computed using equation (22):

$$co\left(\overleftarrow{W_i, W_j}^{\varepsilon}\right) = \sum_{l=1}^{L_{Doc}} \frac{N^{line\ l}\left(\overleftarrow{W_i, W_j}^{\varepsilon}\right)}{\left\lceil \frac{N(line\ l)}{\varepsilon} \right\rceil} \quad (22)$$

where $N^{line\ l}\left(\overleftarrow{W_i, W_j}^{\varepsilon}\right)$ denotes the number of times that W_i and W_j co-occur with a minimum distance ε, regardless of the order of appearance, and N(line l) denotes the number of terms of line l. A relation between two terms W_i and W_j is formally defined when the computed co-occurrence between them exceeds the co-occurrence threshold determined by experimentation. Figure 10 presents an overview of the training process phase.

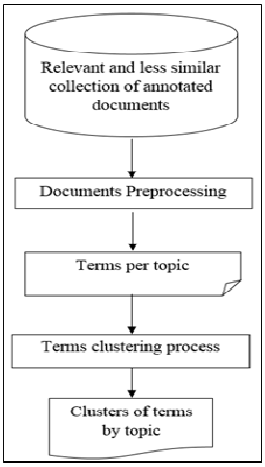


Figure 10. Training process phase - Architecture overview

3.4.5 Topics refining - process phase

Figure 11 presents the process used by SATD to refine the detected topics making use of relevant documents already annotated by humans based on existing or known topics. Following this process, three lists of topics are obtained: a list of new topics, a list of similar existing topics and a list of not similar existing topics. The list of existing topics that match new document detected topics is identified based on the new document detected topics and annotated documents by topic (existing topics). The clusters of terms by topic are identified based on the collection of relevant and less similar documents. Note: each topic is a cluster of terms graph. Therefore, a graph matching technique is a good candidate to perform topic similarity detection. Next, using our graph matching technique, the clusters of terms by topics of relevant and less similar collection of annotated documents which match with CTG are identified, for each cluster of terms graph by topic (CTG) of the new document.

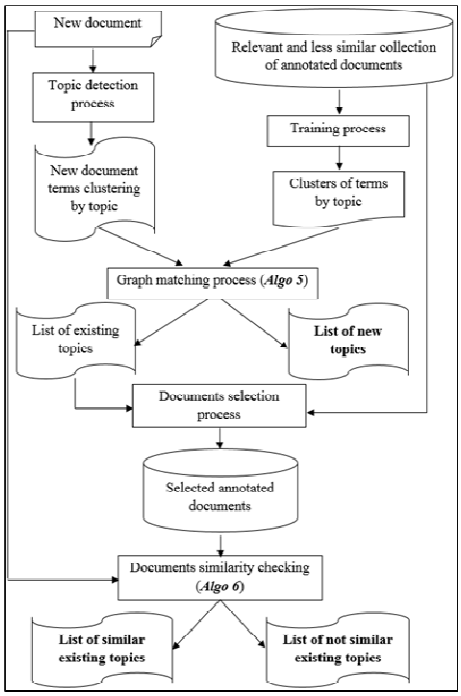


Figure 11. Topic refining process phase - Architecture overview

The matching score between two clusters is then computed. Let H be the new document terms graph and G be the terms graph obtained by a training process applied on the collection of relevant and less similar documents annotated by topics, C_j^d be a cluster of H associated to topic t_j^d and C_i be a cluster of G associated with topic t_i, and W_i and W_j be two terms of cluster C_j^d; the link matching function $g(\overline{W_iW_j})$ between W_i and W_j is defined by equation (23):

$$g: C_j^d \times C_j^d \rightarrow IR$$
$$g(\overline{W_iW_j}) = \begin{cases} MinHopClusterOfti(W_i,W_j) & \text{if path between } W_i,W_j \\ 1 + MaxHopClusterOfti & \text{if not path between } W_i,W_j \end{cases} \quad (23)$$

For a direct link $\overline{W_iW_j}$ (only one hop between W_i and W_j) of cluster C_j^d, the process checks whether there is a path between W_i and W_j in the cluster C_i, regardless of the number of hops:

1. If paths exist between W_i and W_j in the cluster C_i, $g(\overline{W_iW_j})$ is the number of hops of the shortest path between W_i and W_j, in term of hops.
2. Otherwise, $g(\overline{W_iW_j})$ is the number of hops of the longest path that exists in the cluster C_i incremented by 1.

Using the link matching function, the matching score between two clusters C_i^d and C_i is given by equation (24):

$$o: H \times G \rightarrow]0;1]$$
$$o(C_j^d, C_i) = \frac{|C_j^d|}{\sum_{W_i,W_j \in C_j^d} g(\overline{W_iW_j})} \quad (24)$$

where $|C_j^d|$ is the number of links in clusters C_i^d. For a better understanding, consider the term graphs in Figure 12.

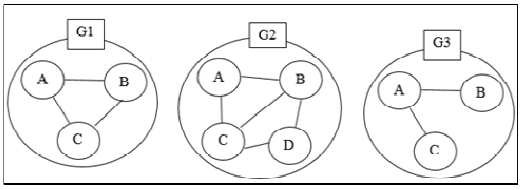


Figure 12. Illustration of term graphs matching score computation

According to Figure 12, o(G1,G2) = 3/3 = 1 while o(G2,G1) = 5/9 and o(G1,G3) = 3/5 while o(G3,G1) = 2/2 = 1. The clusters of H and G whose matching scores exceed a term cluster matching threshold are considered as matching and are assumed to be the same topics. Otherwise, the clusters of H that do not match any clusters of G, are assumed to be new topics. Note that the term cluster matching threshold is determined by experimentation. Based on the H and G clusters that match, the relevant and less similar documents per existing topic that may have the same topic as the new document are identified. Making use of this set of selected documents, the similarity between the new document and each relevant and less similar document of each existing topic i is measured. Let D be the union of the new document *d* and a set of relevant and less similar documents of existing topics t_i that are selected by documents selection and $W = \{W_1, \ldots, W_m\}$ the set of distinct terms occurring in D. The defined m-dimensional vector represents each document of D. For each term of W, its tf-idf is computed using equation (1). This allows one to obtain the vector $\overrightarrow{t_d} = (\text{tfidf}(W_1, d, t_i), \ldots, \text{tfidf}(W_m, d, t_i))$. When documents are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors. Here, cosine similarity is applied to measure this similarity. The cosine similarity is defined as the cosine of the angle between vectors. An important property of the cosine similarity is its independence of document length. Given two documents $\overrightarrow{t_{d1}}$ and $\overrightarrow{t_{d2}}$, their cosine similarity is computed using equation (25):

$$SimCos(\overrightarrow{t_{d1}}, \overrightarrow{t_{d2}}) = \frac{\overrightarrow{t_{d1}} \cdot \overrightarrow{t_{d2}}}{|\overrightarrow{t_{d1}}| \times |\overrightarrow{t_{d2}}|} \quad (25)$$

Note that it is already assumed that when the similarity $SimCos(\overrightarrow{t_{d1}}, \overrightarrow{t_{d2}})$ of two documents d1 and d2 is less than the similarity threshold β, the documents are not similar. The computation of document similarity allows SATD to classify the existing topics into: (1) Similar existing topics and (2) Not similar existing topics.

4. EVALUATION USING SIMULATIONS

This section presents an evaluation of SATD performance using simulations. To perform these simulations, an experimental environment called Libër was used. Libër was developed to provide a simulator to prototype SATD algorithm.

4.1. Dataset and parameters

To evaluate SATD, real datasets from different projects that have digital and physical library catalogues were used. These datasets, consisting of 25,000 documents with a vocabulary of 375,000 words, were selected using average TF-IDF for the analysis. The documents covered 20 topics. The number of documents per topic or emotion was approximately equal. The average number of topics per document was 7 while the average rating emotion number per document was 4. 15,000 documents of the dataset were used for the training phase and the remaining 100 used for the test. Note that the 10,000 documents used for the tests were those that had more annotated topics or a higher rating over emotions.

To measure the performance of topic detection, comparison of detected topics with annotation topics were carried out. Table 2 presents the values of the parameters used in the simulations. The server characteristics for the simulations were: Dell Inc. PowerEdge R630 with 96 Ghz (4 x Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz, 10 core and 20 threads per CPU and 256 GB memory running VMWare ESXi 6.0.

Table 2. Simulation parameters

Parameter	Value	Parameter	Value
ε	3	α	100
NumKeyTerm	8	co-occurrence threshold	0.75
ω	0.5	semantic threshold	1
β	0.7	term cluster matching threshold	0.45
λ	0.6		

4.2. Performance criteria

SATD performance was measured in terms of running time [8] and accuracy [15] [14]. Note that in the library domain, the most important criteria was precision while resource consumption was important for the software providers.

The running time, denoted by Rt, was computed as follows:

$$Rt = Et - Bt$$

where *Et* and denotes the time when processing is completed and *Bt* the time when it started. To compute the accuracy, let $T_{annotated}$ and $T_{detected}$ be the set of annotated topic and the set of detected topics by SATD for a given document d. The accuracy of topics detection, denoted by A_d^t, was computed as follows:

$$A_d^t = \frac{2.|T_{annotated} \cap T_{detected}|}{|T_{annotated}| + |T_{detected}|}$$

Simulation results were averaged over multiple runs with different pseudorandom number generator seeds. The average accuracy, *Ave_acc*, of multiple runs was given by:

$$Ave\_acc = \frac{\sum_{x=1}^{I}\left(\frac{\sum_{d \in TD} A_d^t}{|TD|}\right)}{I}$$

where TD denotes the number of tests documents and I denotes the number of test iterations. The average running time, *Ave_run_time,* was given by:

$$Ave\_run\_time = \frac{\sum_{x=1}^{I} Rt}{I}$$

4.3 Comparison approaches

SATD performance was evaluated in terms of running time and accuracy. The dataset and parameters mentioned above were applied. SATD performance was compared to the approaches described in [15], [14], [4] and [8], referred to as LDA-IG (probabilistic and graph approach), KeyGraph (graph analytical approach), LDA (probabilistic approach) and HLTM, respectively. LDA-IG, KeyGraph, LDA and HLTM were selected because they are text-based and long text approaches. Table 3 presents the characteristics of the comparison approaches. Our prototype approach SATD is the only one that is really semantic and takes into account the correlated topic and domain knowledge.

Table 3. Topic detection approaches for comparison

Approach	Granularity	Description	Training phase	Refining	Semantic	Topic corre-lation	Domain knowledge
LDA-IG [15]	D	P,G	Yes	No	No	No	No
KeyGraph [14]	D	G	Yes	No	No	No	No
LDA [4]	D	P	No	No	No	No	No
HLTM [8]	D	P,G	Yes	No	No	No	No
SATD	C	S,P,G	Yes	Yes	Yes	Yes	Yes

D: document; C: Configurable as desired; P: Probabilistic based; G: Graph based; S: Semantic based.

4.4. Results analysis

Figure 13 presents the average running time of the detection phase when the number of documents used for the tests were varied. Training times were excluded as this phase was performed only one time. However, the SATD training phase required more time than the other approaches. This was justified by the fact that SATD identifies the relevant and less similar documents used for training phase. Figure 13 also shows that the average running time increased with the number of test documents. Indeed, the bigger the number of test documents, the longer the time to perform detection and, ultimately, the higher the average running time.

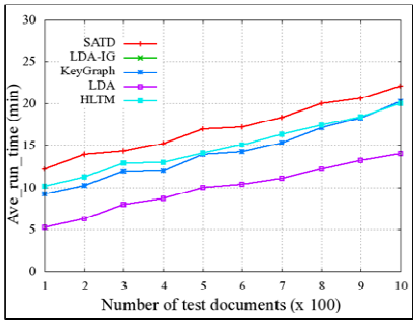


Figure 13. Topic detection - Average running time versus number of documents for test phase

It was also observed that LDA outperforms the other approaches. LDA produced an average of 1.37 sec per document whereas SATD produced an average of 2.62 sec per document. The average relative improvement (defined as [Aver._runtime of SATD − Aver._runtime of LDA]) of LDA compared with

SATD was approximately 1.25 sec per document. The short run times of LDA were due to the fact that LDA did not perform a graph treatment. Graph processing algorithms are very time consuming. Other approaches also outperformed SATD on the running time criteria since SATD performed topic refining in order to increase accuracy.

Figure 14 shows the average accuracy when varying the number of detected topics. For the five approaches, the average accuracy decreased with the number of detected topics. The increase in the number of subjects to detect led to decreased accuracy. However, in terms of accuracy, SATD outperformed the approaches used for comparison. SATD produced an average accuracy of 79.50% per topic while LDA-IG, the best among the approaches used for comparison, produced an average of 61.01% per topic. The average relative improvement in accuracy (defined as [Ave_acc of SATD − Ave_acc of LDA-IG]) of SATD compared to LDA-IG was 18.49% per topic. The performance of SATD is explained as follows: (1) SATD used the relevant documents for training phase, (2) SATD refined its detection topic results by measuring new document similarity with relevant and less similar annotated documents, and (3) SATD combined correlated topic model and domain knowledge model instead of LDA.

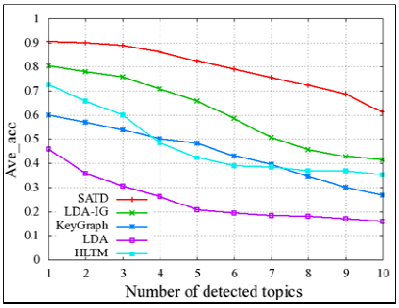


Figure 14. Accuracy for number of detected topics for 5 comparison approaches

Figure 14 also shows that SATD produced an average accuracy of 90.32% for one detected topic and 61.27% for ten detected topics compared to 80.29% and 41.01% respectively for LDA-IG. The gap between SATD accuracy and LDA-IG accuracy was 10.03% for one detected topic and 20.26% for ten detected topics. This meant that SATD was by in large more accurate than LDA-IG in detecting several topics.

The Figure 15 presents the average accuracy when varying the number of training documents of the learning phase. LDA was not included in the scenario since no training phase was performed. Figure 15 shows that the average accuracy increased with the number of training documents. The larger the number of training documents, the better the knowledge about word distribution and co-occurrence and, ultimately, the higher the detection accuracy. However, the accuracy remained largely stable for very high numbers of training documents. When the number of documents of a collection was larger, the number of vocabulary words remained constant, and the term graph did not change. It also shows that HLTM was the approach whose detection accuracy was the first to reach stability at 10,000 training documents. HLTM builds a tree instead of a graph as the other approaches and its tree has less internal roots to identify topics. However, SATD and LDA-IG outperformed HLTM in terms of accuracy.

Figure 15 also shows that SATD outperformed LDA-IG on the accuracy criteria. For example, SATD demonstrated an average accuracy of 73.49% per 2,000 training documents while LDA-IG produced an average accuracy of 50.86% per 2,000 training documents. The average relative improvement of SATD compared to LDA-IG was 22.63% per 2,000 training documents. The better performance of SATD followed from its use of a specific domain knowledge model. SATD did not require a large number of documents for the training phase.

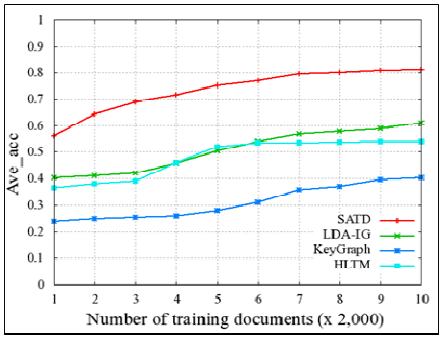


Figure 15. Topic detection - accuracy for number of training documents

In conclusion, the 1.25 sec running time per document increase was a small price to pay for the larger average accuracy of topic detection (18.49%).

5. SUMMARY AND FUTURE WORK

The goal of this paper was to increase the findability (search engines) of user interests using semantic metadata enrichment model and algorithm. Words themselves have a wide variety of definitions and interpretations and are often utilized inconsistently. While topics may have no relationship to individual words, thesauri express associative relationships between words, ontologies, entities and a multitude of relationships represented as triplets. This paper presented an enhanced implementation of SMESE [1] model using SATD engine for topic metadata enrichments.

To help users find interest-based contents, this paper proposes to enhance the SMESE platform [1] through text analysis approaches for topic detection. This paper presents the design, implementation and evaluation of the algorithm SATD focusing on semantic topic extraction. The SATD topic metadata enrichments prototype allows to: (1) generate semantic topics by text, and multimedia content analysis using the proposed SATD (Scalable Annotation-based Topic Detection) algorithm and (2) implement rule-based semantic metadata internal enrichment. Table 1 shows the comparison with most known text mining algorithms (e.g., AlchemyAPI, DBpedia, Wikimeta, Bitext, AIDA, TextRazor) and a new algorithm using keyword extraction, classification and concept extraction. It was noted that SATD algorithm support more attributes than the other algorithms evaluated.

In future work, the focus will be to generate learning-based literature review enrichment and abstract of abstract. It will assess each reference extracting topics to determine her ranking and her inclusion in the literature assistant review. One main goal is to reduce reading load by helping researcher to read only the most related selection of documents to literature review. Using text data mining, machine learning, and a classification model that learn from users annotated data and detected metadata the algorithms will assist the researcher to rank the relevant documents for his literature review for a specific topic and selection of metadata.

REFERENCES

[1] Brisebois R, Abran A, Nadembega A (2017) A Semantic Metadata Enrichment Software Ecosystem (SMESE) based on a Multi-platform Metadata Model for Digital Libraries. Accepted for publication in Journal of Software Engineering and Applications (JSEA) 10 (04)

[2] Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Information Processing & Management 24 (5):513-523. doi:http://dx.doi.org/10.1016/0306-4573(88)90021-0

[3] Niu T, Zhu S, Pang L, El Saddik A (2016) Sentiment Analysis on Multi-View Social Data. Paper presented at the 22nd International Conference on MultiMedia Modeling (MMM), Miami, FL, USA, 4-6 Jan. 2016

[4] Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet Allocation. Journal of Machine Learning Research 3:993–1022

[5] Bijalwan V, Kumar V, Kumari P, Pascual J (2014) KNN based Machine Learning Approach for Text and Document Mining. International Journal of Database Theory and Application 7 (1):61-70. doi:http://dx.doi.org/10.14257/ijdta.2014.7.1.06

[6] Dumais ST (2004) Latent semantic analysis. Annual Review of Information Science and Technology 38 (1):188-230. doi:10.1002/aris.1440380105

[7] Cigarrán J, Castellanos Á, García-Serrano A (2016) A step forward for Topic Detection in Twitter: An FCA-based approach. Expert Systems with Applications 57:21-36. doi:http://dx.doi.org/10.1016/j.eswa.2016.03.011

[8] Chen P, Zhang NL, Liu T, Poon LKM, Chen Z (2016) Latent Tree Models for Hierarchical Topic Detection. arXiv preprint arXiv:160506650 [csCL]:1-44

[9] Moraes R, Valiati JF, Gavião Neto WP (2013) Document-level sentiment classification: An empirical comparison between SVM and ANN. Expert Systems with Applications 40 (2):621-633. doi:http://dx.doi.org/10.1016/j.eswa.2012.07.059

[10] Ghiassi M, Skinner J, Zimbra D (2013) Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. Expert Systems with Applications 40 (16):6266-6282. doi:http://dx.doi.org/10.1016/j.eswa.2013.05.057

[11] Dang Q, Gao F, Zhou Y (2016) Early detection method for emerging topics based on dynamic bayesian networks in micro-blogging networks. Expert Systems with Applications 57:285-295. doi:http://dx.doi.org/10.1016/j.eswa.2016.03.050

[12] Cotelo JM, Cruz FL, Enríquez F, Troyano JA (2016) Tweet categorization by combining content and structural knowledge. Information Fusion 31:54-64. doi:http://dx.doi.org/10.1016/j.inffus.2016.01.002

[13] Hashimoto T, Kuboyama T, Chakraborty B (2015) Topic extraction from millions of tweets using singular value decomposition and feature selection. Paper presented at the 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Hong Kong, China, 16-19 Dec. 2015

[14] Sayyadi H, Raschid L (2013) A Graph Analytical Approach for Topic Detection. ACM Transactions on Internet Technology 13 (2):1-23. doi:http://dx.doi.org/10.1145/2542214.2542215

[15] Zhang C, Wang H, Cao L, Wang W, Xu F (2016) A hybrid term–term relations analysis approach for topic detection. Knowledge-Based Systems 93:109-120. doi:http://dx.doi.org/10.1016/j.knosys.2015.11.006

[16] Bougiatiotis K, Giannakopoulos T (2016) Content Representation and Similarity of Movies based on Topic Extraction from Subtitles. Paper presented at the Proceedings of the 9th Hellenic Conference on Artificial Intelligence, Thessaloniki, Greece, 18-20 May 2016

[17] Salatino AA, Motta E (2016) Detection of Embryonic Research Topics by Analysing Semantic Topic Networks. Paper presented at the Semantics, Analytics, Visualisation: Enhancing Scholarly Data, Montreal, Quebec, Canada, 11 Apri. 2016

[18] Hurtado JL, Agarwal A, Zhu X (2016) Topic discovery and future trend forecasting for texts. Journal of Big Data 3 (1):1-21. doi:http://dx.doi.org/10.1186/s40537-016-0039-2

[19] Porter MF (1980) An algorithm for suffix stripping. Program 14 (3):130-137. Doi: doi: 10.1108/eb046814

[20] de Marneffe M-C, MacCartney B, Manning CD (2006) Generating typed dependency parsers from phrase structure parses Paper presented at the fifth international conference on language resources and evaluation, GENOA , ITALY 22-28 May 2006

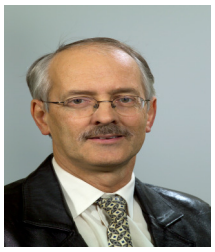
[21] Blei DM, Lafferty JD (2005) Correlated Topic Models. Paper presented at the Proceedings of the 19th Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, 5-8 Dec. 2005

[22] Andrzejewski D, Zhu X, Craven M (2009) Incorporating domain knowledge into topic modeling via Dirichlet Forest priors. Paper presented at the Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Quebec, Canada, 14-18 Jun. 2009

[23] Pedersen T, Patwardhan S, Michelizzi J (2004) WordNet::Similarity: measuring the relatedness of concepts. Paper presented at the Demonstration Papers at Human Language Technology conference/North American chapter of the Association for Computational Linguistics (HLT-NAACL), Boston, Massachusetts, USA, 2-7 May 2004

Authors

Ronald Brisebois is currently a PhD student at the École de Technologie Supérieure (ETS) – Université du Québec (Montréal, Canada). He received a B. Science in Physics at University of Montreal in 1983, a BA in Computer Science at University of Quebec in 1985 and his MBA at Hautes Études Commerciales - HEC (Business School) in 1989. From 1989 to 1995, Ronald Brisebois was a professor of Software Engineering at the University of Sherbrooke. His PhD research focus on semantic web, artificial intelligence, autonomous software architecture, new generation software designing, enriched metadata modeling and software engineering. Renowned entrepreneur in the field of information technology, Ronald Brisebois has held management positions in various top-level firms (Caisses populaires Desjardins). In 1991, he was a professor at the University of Sherbrooke; in 1992, he founded his first company. Cognicase Inc. quickly became one of the largest players in the information technology field in Canada. In 2003, Ronald created Isacsoft/MondoIn, one of the leading providers of integrated solutions for public libraries, academic institutions, specialized and consortia systems worldwide.



Dr. Abran holds a Ph.D. in Electrical and Computer Engineering (1994) from École Polytechnique de Montréal (Canada) and master degrees in Management Sciences (1974) and Electrical Engineering (1975) from University of Ottawa (Canada).He is a professor at the École de Technologie Supérieure (ETS) – Université du Québec (Montréal, Canada). He has over 20 years of experience in teaching in a university environment as well as 20 years of industry experience in information systems development and software engineering management. His research interests include software productivity and estimation models, software engineering foundations, software quality, software functional size measurement, software risk management and software maintenance management. He has published over 400 peer-reviewed papers. He is the author of the books 'Software Project Estimation', 'Software Metrics and Software Metrology' and a co--author of the book 'Software Maintenance Management' (Wiley Interscience Ed. & IEEE-CS Press). Dr. Abran is also the 2004 co-executive editor of the Guide to the Software Engineering Body of Knowledge – SWEBOK (see ISO 19759 and www.swebok.org) and he is the chairman of the Common Software Measurement International Consortium (COSMIC) – http://cosmic-sizing.org/. A number of Dr. Abran research works have influenced international standards in software engineering (i.e., ISO 19761, ISO 19759, ISO 14143-3, etc.)

Dr. Apollinaire Nadembega is currently a guest member of the Network Research Laboratory (NRL) of the University of Montreal. He received his B. E degree in Information Engineering from Computer Science High School, Bobo-Dioulasso, Burkina faso in 2003, his Master's degree in computer science from the Arts and Business Institute, Ouagadougou, Burkina faso in 2007 and his Ph.D. degree in mobile networks from the University of Montreal, Montreal, QC, Canada in 2014. The primary focus of his Ph.D. thesis is to propose a mobility model and bandwidth reservation scheme that supports quality-of-service management for wireless cellular networks. Dr. Nadembega's research interests lie in the field of artificial intelligence, machine learning, networking modelling, semantic web, metadata management system, software architecture, mobile multimedia streaming, call admission control, bandwidth management and mobile cloud computing. From 2004 to 2008, he was a programming engineer with Burkina Faso public administration staff management office.

Philippe started with a three-year training as a computer expert at the institute Leonardo da Vinci in Italy. Then, he joined the University of Parma, where he obtained his Bachelor in Computer Engineering with honors. He was then admitted at Polytechnic of Milan, one of the most prestigious engineering school (24th for Engineering in the world) for a master degree in computer engineering. After his first year, he won a scholarship for a double degree exchange program with the Polytechnic School of Montreal to obtain a second master more focused towards research in Natural Language Processing. In the last two years, he worked as research scientist for Ecole Polytechnique de Montreal, Bibliomondo and Nuance communications.