Semantic Web 0 (0) 1
IOS Press

Large-Scale Semantic Exploration of Scientific Literature using Topic-based Hashing Algorithms

Editor(s): Tomi Kauppinen, Aalto University, Finland; Daniel Garijo, University of Southern California, USA; Natalia Villanueva, The University of Texas at El Paso, USA
Solicited review(s): Anita de Waard, Elsevier Labs

Carlos Badenes-Olmedo [a,*], José Luis Redondo-García [b] and Oscar Corcho [a]
[a] *Ontology Engineering Group, Universidad Politécnica de Madrid, Boadilla del Monte, Spain*
E-mails: cbadenes@fi.upm.es, ocorcho@fi.upm.es
ORCIDs: 0000-0002-2753-9917, 0000-0002-9260-0753
[b] *Amazon Research, Cambridge, UK*
E-mail: jluisred@amazon.com
ORCID: 0000-0002-7413-447X

Abstract. Searching for similar documents and exploring major themes covered across groups of documents are common activities when browsing collections of scientific papers. This manual knowledge-intensive task can become less tedious and even lead to unexpected relevant findings if unsupervised algorithms are applied to help researchers. Most text mining algorithms represent documents in a common feature space that abstract them away from the specific sequence of words used in them. Probabilistic Topic Models reduce that feature space by annotating documents with thematic information. Over this low-dimensional latent space some locality-sensitive hashing algorithms have been proposed to perform document similarity search. However, thematic information gets hidden behind hash codes, preventing thematic exploration and limiting the explanatory capability of topics to justify content-based similarities. This paper presents a novel hashing algorithm based on approximate nearest-neighbor techniques that uses hierarchical sets of topics as hash codes. It not only performs efficient similarity searches, but also allows extending those queries with thematic restrictions explaining the similarity score from the most relevant topics. Extensive evaluations on both scientific and industrial text datasets validate the proposed algorithm in terms of accuracy and efficiency.

Keywords: Document Similarity, Information Search and Retrieval, Clustering, Topic Models, Hashing

1. Introduction

Huge amounts of documents are publicly available on the Web offering the possibility of extracting knowledge from them (e.g. scientific papers in digital journals). Document similarity comparisons in many information retrieval (IR) and natural language processing (NLP) areas are too costly to be performed in such huge collections of data and require more efficient approaches than having to calculate all pairwise similarities.

In this paper we address the problem of programmatically generating annotations for each of the items inside big collections of textual documents, in a way that is computationally affordable and enables a semantic-aware exploration of the knowledge inside it that state-of-the-art methods relying on topic models are not able to materialize.

Most text mining algorithms represent documents in a common feature space that abstracts the specific

*Corresponding author. E-mail: cbadenes@fi.upm.es.

sequence of words used in each document and, with appropriate representations, facilitate the analysis of relationships between documents even when written using different vocabularies. Although a sparse word or n-gram vectors are popular representational choices, some researchers have explored other representations to manage these vast amounts of information. Latent Semantic Indexing (LSI) [17], Probabilistic Latent Semantic Indexing (PLSI) [25] and more recently, Latent Dirichlet Allocation (LDA) [11], which is the simplest probabilistic topic model (PTM) [10], are algorithms focused on reducing feature space by annotating documents with thematic information. PLSI and PTM also allow a better understanding of the corpus through the topics discovered, since they use probability distributions over the complete vocabulary to describe them. However, only PTM's are able to identify topics in previously unseen texts.

One of the greatest advantages of using PTM in large document collections is the ability to represent documents as probability distributions over a small number of topics, thereby mapping documents into a low-dimensional latent space (the K-dimensional probability simplex, where K is the number of topics). A document, represented as a point in this simplex, is said to have a particular topic distribution. This brings a lot of potential when applied over different IR tasks, as evidenced by recent works in different domains such as scholarly [23][19], health [45] [38] [50], legal [43][20], news [24] and social networks [47][15]. This low-dimensional feature space could also be suitable for document similarity tasks, especially on big real-world data sets, since topic distributions are continuous and not as sparse as discrete-term feature vectors.

Exact similarity computations for most topic distributions require to have complexity $O(n^2)$ for neighbours detection tasks or $O(kn)$ computations when k queries are compared against a dataset of n documents. Computation can be an approximate nearest neighbor (ANN) search problem. ANN search is an optimization problem that finds nearest neighbors of a given query q in a metric space of n points. Due to the low storage cost and fast retrieval speed, hashing is one of the most popular solutions for ANN search [34] [4] [60]. This technique transforms data points from the original feature space into a binary-code space, so that similar data points have larger probability of collision (i.e. having the same hash code). This type of formulation for the document similarity comparison problem has proven to yield good results in the metric space due to the fact that ANN search has been designed to handle distance metrics (e.g. cosine, Euclidean, Manhattan) [49][46][29], even in high-dimensional simplex spaces handling information-theoretically motivated metrics (e.g. Hellinger, Kullback-Leibler divergence, Jensen-Shannon divergence) as demonstrated by [39].

However, the smaller space created by existing hashing methods loses the exploratory capabilities of topics to support document similarity. The notion of topics is lost and therefore the ability to make thematic explorations of documents. Moreover, metrics in simplex space are difficult to interpret and the ability to explain the similarity score on the basis of the topics involved in the exploration can be helpful. While other models based on vector representations of documents are simply agnostic to the human concept of themes, topic models can help finding the reasons why two documents are similar.

Semantic knowledge can be thought of as knowledge about relations among several types of elements, including words, concepts, and percepts [22]. Since topic models create latent themes from word co-occurrence statistics in corpus, a topic (i.e latent theme) reflects the knowledge about the word-word relations it contains. This abstraction can be extended to cover the knowledge derived from sets of topics. The topics obtained via state-of-the art methods (LDA) are hierarchically divided into groups with different degrees of semantic specificity in a document. Documents can then be annotated with the semantic inferred from the topics detected, and from their relation between topics inside each hierarchy level. Let's look at a practical example to clarify this idea. A topic model is created from texts labeled with Eurovoc [1] categories. This model[2] annotates texts with categories inferred from their topic distributions. For the document *"Commission Decision of 23 December 2003.. on seeds and propagating material of gramineae, Triticum aestivum.."* [3], the top5 categories are: (1) *research*, (2) *sugar*, (3) *fats*, (4) *textile_industry* and (5) *marketing* . In contrast to these categories that standard topic modelling methods are able to offer, a 3-level hierarchical set of topics would be: (1) *research*, (2) *sugar* and *fats*, and (3) *textile_industry* and *marketing*. The knowledge provided by each of these annotations is derived from the relations between the topics that compose it. Based on these semantic annotations, the content-based similarity among documents is calcu-

[1] http://publications.europa.eu/resource/dataset/eurovoc
[2] http://librairy.linkeddata.es/jrc-en-model/
[3] https://eur-lex.europa.eu/eli/dec/2004/57(1)/oj

lated and the exploration of large document collections is performed following an ANN search.

Thus, in this paper, we propose a hashing algorithm that (1) groups similar documents, (2) preserves their topic distributions, and (3) works over unseen documents. Therefore our contributions are:

- a **novel hashing algorithm** based on topic models that not only performs efficient searches, but also introduces semantic in the hierarchy of concepts as a way to restrict those queries and provide explanatory information.
- an optimized and easily customizable open-source **implementation of the algorithm** [9]
- **data-sets** and **pre-trained models** to facilitate other researchers to replicate our experiments and validate and test their own ideas [9]

2. Document Similarity

In the probability simplex space created from topic models, documents are represented as vectors containing topic distributions. Distance metrics based on vector-type data such as Euclidean distance (l_2), Manhattan distance (l_1), and angular metric (θ) are not optimal in this space [39]. Information-theoretically motivated metrics such as Kullback-Leibler (KL) divergence (Eq.1) (also known as relative entropy), Jensen-Shannon (JS) divergence (Eq.2) (as its symmetric version) and Hellinger (He) distance (Eq.3) are often more reasonable [39]:

$$KL(P,Q) = \sum_{i=1}^{K} p(x_i) \log \frac{p(x_i)}{q(x_i)} \tag{1}$$

$$JS(P,Q) = \frac{1}{2} KL\left(p, \frac{p+q}{2}\right) + \frac{1}{2} KL\left(q, \frac{p+q}{2}\right) \tag{2}$$

$$He(P,Q) = \sum_{i=1}^{K} \left(\sqrt{p(x_i)} - \sqrt{q(x_i)}\right)^2 \tag{3}$$

where P and Q are two known distributions, K is the dimensionality of P and Q, and p_i and q_i are the values of the i_{th} component of P and Q, respectively.

He distance is also symmetric and, along with JS divergence, are usually used in various fields where a comparison between two probability distributions is required. However, all these metrics are not well-defined distance metrics, that is, they do not satisfy triangle inequality [14]. This inequality considers $d(x,z) <= d(x,y) + d(y,z)$ for a metric d [22]. It places strong constraints on distance measures and on the locations of points in a space given a set of distances. As a metric axiom the triangle inequality must be satisfied in order to take advantage of the inferences that can be deduced from it. Thus, if similarity is assumed to be a monotonically decreasing function of distance, this inequality avoids the calculation of all pairs of similarities by considering that if x is similar to y and y is similar to z, then x must be similar to z.

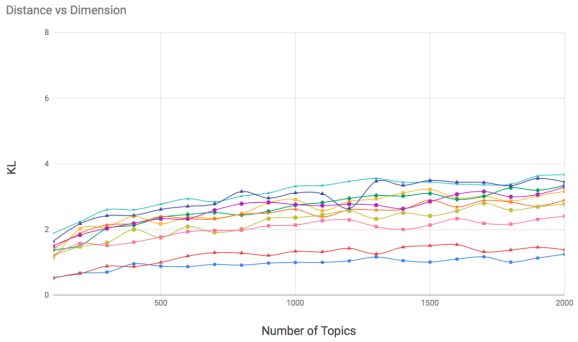

Fig. 1. Distance values based on KL-divergence between 10 pair of documents from topic models with 100-to-2000 dimensions.

S2JSD was introduced by [18] to satisfy the triangle inequality. It is the square root of two times the JS divergence:

$$S2JSD(P,Q) = \sqrt{2 * JS(P,Q)} \tag{4}$$

However, making sense out of the similarity score is not easy. As shown in figures 1 to 4, given a set of pairs of documents, their similarity scores vary according to the number of topics. So the distances between those pairs fluctuate from being more to less distant when changing the number of topics.

Distances between documents generally increase as the number of dimensions of the space increases. This is due to the fact that as the number of topics describing the model increases, the more specific the topics will be. Topics shared by a pair of documents can be broken down into more specific topics that are not shared by those documents. Thus, similarity between pairs of documents is dependent on the model used to represent them when considering this type of metrics. We know that absolute distances between documents vary when we tune hyperparameters differently, but in this study

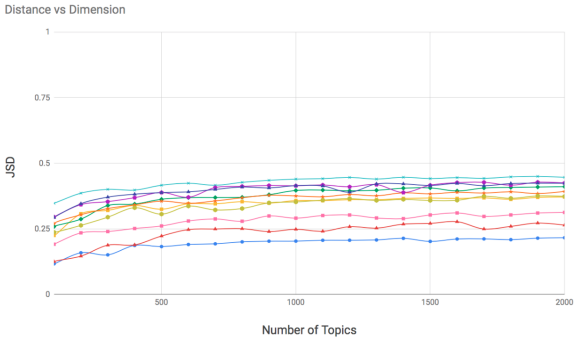

Fig. 2. Distance values based on JS-divergence between 10 pair of documents from topic models with 100-to-2000 dimensions.

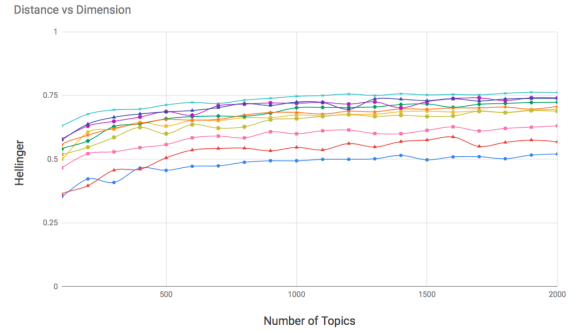

Fig. 3. Distance values based on He-divergence between 10 pair of documents from topic models with 100-to-2000 dimensions.

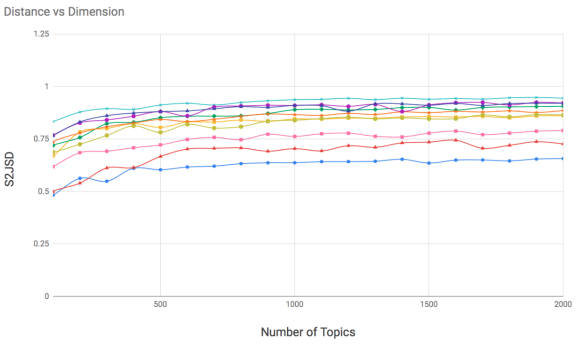

Fig. 4. Distance values based on $S2JSD$ between 10 pair of documents from topic models with 100-to-2000 dimensions.

we also see that "relative distances" also change: e.g. for model M1, A is closer to B than C, but according to a M2 trained in the same corpora with different parameters , A is closer to C than B (cross-lines in figs 1-4). This behaviour highlights the difficulty of establishing absolute similarity thresholds and the complexity to measure distances taking into account all dimensions. Distance thresholds should be model-dependent rather than general and metrics flexible enough to handle dimensional changes. These challenges are tackled through the proposed hashing algorithms by means of clusters of topics to measure similarity, instead of directly using their weights.

3. Hashing Topic Distributions

Hashing methods transform the data points from the original feature space into a binary-code Hamming space, where the similarities in the original space are preserved. They can learn hash functions (data-dependent) or use projections (data-independent) from the training data [56]. Data-independent methods unlike data-dependent ones do not need to be re-calculated when data changes, i.e. adding or removing documents to the collection. Taking large-scale scenarios into account (e.g. Document clustering, Content-based Recommendation, Duplicate Detection), this is a key feature along with the ability to infer hash codes individually (for each document) rather than on a set of documents.

Data-independent hashing methods depend on two key elements: (1) data type and (2) distance metric. For vector-type data, as introduced in section 2, based on l_p distance with $p\epsilon[0,2)$ lots of hashing methods have been proposed, such as p-stable Locality-Sensitive Hashing (LSH) [16], Leech lattice LSH [3], Spherical LSH [51], and Beyond LSH [5]. Based on the θ distance many methods have been developed such as Kernel LSH [31] and Hyperplane hashing [53]. But only few methods handle density metrics in a simplex space. A first approach transformed the He divergence into an Euclidean distance so that existing ANN techniques, such as LSH and k-d tree, could be applied [30]. But this solution does not consider the special attributions of probability distributions, such as Non-negative and Sum-equal-one. Recently, a hashing schema [39] taking into account the symmetry has been proposed, non-negativity and triangle inequality features of the S2JSD metric for probability distributions. For set-type data, Jaccard Coefficient is the main metric used. Some examples are K-min Sketch [33], Min-max hash [28], B-bit minwise hashing [32] and Sim-min-hash [59].

All of them have demonstrated efficiency in the search for similar documents, but none of them allows the search for documents (1) by thematic areas or (2) by similarity levels, nor they offer (3) an explanation about the similarity obtained beyond the vectors used to calculate it. Binary-hash codes drop a very precious information: the topic relevance.

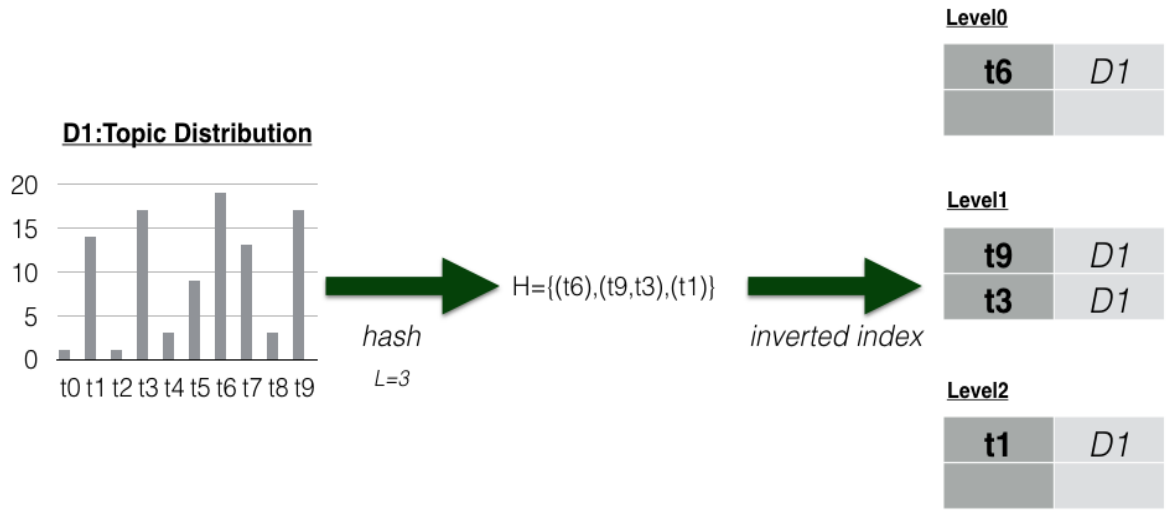

Fig. 5. Hash method based on hierarchical set of topics from a given topic distribution

A new hierarchical set-type data is proposed. Each level of the hierarchy indicates the importance of the topic according to its distribution. Level 0 contains the topics with the highest score. Level 1 contains the topics with highest score once the first ones have been eliminated, and so on. From a vector of components, where each of the components is the score of topic t, a vector containing set of topics is proposed, where each of the dimensions means a topic relevance. Thus, for the topic distribution $q = [0.3, 0.15, 0.4, 0.15]$, a hierarchical set of topics may be $h = \{(t2), (t0), (t1, t3)\}$. It means that topic $t2$ (0.4) is the most relevant, then topic $t0$ (0.3) and, finally, topics $t1$ (0.15) and $t3$ (0.15). This is just an example about the data structure that will support the different hashing strategies. In section 3.3 some approaches to create hash codes based on this data structure are described.

3.1. Data Type

A traditional approach to text representation usually requires encoding of documents into numerical vectors. Words are extracted from a corpus as feature candidates and based on a certain criterion they are assigned values to describe the documents: term-frequency, TF-IDF, information gain, and chi-square are typical measures. But this causes two main problems: huge number of dimensions and sparse distribution. The use of topics as feature space has been extended to mapping documents into low-dimensional vectors. However, as shown in Figures 1 to 4, the distance metrics based on probability densities vary according to the dimensions of the model and reveal the difficulty of calculating the similarity values using the vectors with the topic distributions.

Since hashing techniques can transform both vector and set-based data [39] [28] into a new space where the similarity (i.e. closeness of points) in the original feature space is preserved, a new set-based data structure is proposed in this paper. It is created from clusters of topics organized by relevance levels and it aims to extend the ability of building queries with topic-based restrictions over the searching space while maintaining high level of accuracy.

The new hierarchical set-type data describes each document as a sequence of sets of topics sorted by relevance. Each level of the hierarchy expresses how important those topics are in that document. In the first level (i.e level 0) are the topics with the highest score. In the second level (i.e level 1) are the topics with the highest score once the first ones have been removed, and so on. In this work, several clustering approaches have been considered to assign topics to each level.

In a feature space created from a PTM with eight topics, for example, each data point p is described

by a eight-dimensional vector with the topic distributions: $vp = [t0, t1, t2, t3, t4, t5, t6, t7]$. Then, given a point $q1 = [0.18, 0.15, 0.2, 0.05, 0.14, 0.11, 0.09, 0.08]$, the three-level hierarchical set of topics may be $h = [\{t2\}, \{t0\}, \{t1, t4\}]$. It means that $t2$ is the most relevant topic, then topic $t0$ and finally topics $t1$ and $t4$. This is just an example about the data structure that will support the hashing strategies. In section 3.3 some approaches to create hash codes based on this data structure are described.

Domain-specific features such as vocabulary, writing style, or speech type, have a major influence on the topic models, but not in the hashing algorithms described in this article. The methods for creating hash codes are agnostic of these particularities since they are only based on the topic distributions generated by the models.

3.2. Distance Metric

Since documents are described by set-type data, the proposed distance metric is based on the Jaccard coefficient. This metric computes the similarity of sets by looking at the relative size of their intersection as follows:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

where A and B are set of topics.

More specifically, d_J is based on the Jaccard distance, which is obtained by subtracting the Jaccard coefficient J from 1:

$$d_J(A, B) = 1 - J(A, B) \tag{6}$$

The proposed distance measure d_H used to compare hash codes created from set of topics is the sum of the Jaccard distances d_j for each hierarchy level, i.e. for each set of topics:

$$d_H(H_1, H_2) = \sum_{l=1}^{L} \Big(d_J(H_1(x_l), H_2(x_l)) \Big) \tag{7}$$

where H_1 and H_2 are hash codes, $H_1(x_l)$ and $H_2(x_l)$ are the set of topics up to level l for each hash code H and L is the maximum hierarchy level. A corner case is $L = T$, where T is the number of topics in the model.

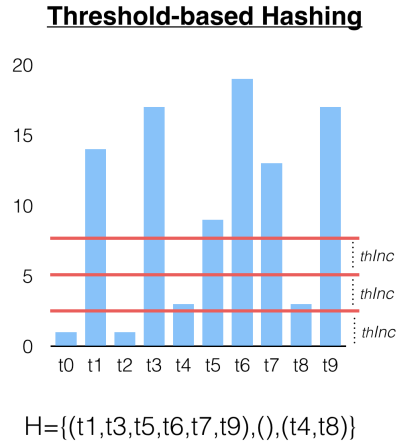

Fig. 6. Threshold-based Hierarchical Hash (L=3)

3.3. Hash Function

The hash function clusters topics based on relevance levels. Three approaches are proposed depending on the criteria used to group topics: threshold-based, centroid-based and density-based.

3.3.1. Threshold-based Hierarchical Hashing Method

This approach is just an initial and naive way of grouping topics by threshold values into each relevance level. They can be manually defined or automatically generated by thresholds dividing the topic distributions as follows:

$$th_{inc} = \frac{1}{(L+1) \cdot T} \tag{8}$$

where L is the number of hierarchy levels, and T the number of topics.

If $L = 3$ and $T = 10$ for a topic distribution td defined as follows:

$$td = [0.017, 0.141, 0.010, 0.172, 0.030, 0.090, 0.199, 0.133, 0.031, 0.171] \tag{9}$$

Then, a threshold-based hierarchical hash H_T, with an automatically created threshold defined by equation 8, is equals to $H_T = \{(t1, t3, t5, t6, t7, t9), (), (t4, t8)\}$ with $th_{inc} = 0.025$ (Fig 6).

3.3.2. Centroid-based Hierarchical Hashing Method

This approach assumes topic distributions can be partitioned into k clusters where each topic belongs to the cluster with the nearest mean score. It is based on

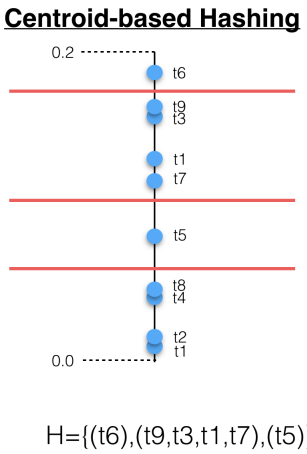

Fig. 7. Centroid-based Hierarchical Hash (L=3)

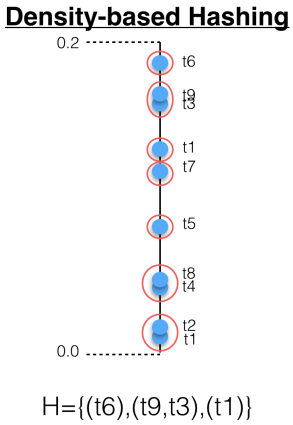

Fig. 8. Density-based Hierarchical Hash (L=3)

the k-Means clustering algorithm, where k is obtained by adding 1 to the number of hierarchy levels. Unlike the previous method, threshold values used to define the hierarchy levels may vary between documents, i.e. for each topic distribution, since they are calculated for each distribution separately.

Following the previous example, if $L = 3$ and $T = 10$ for a topic distribution td defined in equation 9, then a centroid-based hieararchical hash H_C equals to $H_C = \{(t6), (t9, t7, t3, t1), (t5)\}$ (Fig 7).

3.3.3. Density-based Hierarchical Hashing Method

This approach also considers relative hierarchical thresholds for each relevance level. Now, a topic distribution is described by points in a single dimension. In this space, topics closely packed together are grouped together. This approach does not require a fixed number of groups. It only requires a maximum distance (eps) to consider two points close and grouped together. This value can be estimated from the own distribution of topics (e.g. variance).

Following the above example, if $L = 3$ and td is the topic distribution defined in equation 9, then a density-based hierarchical hash H_D is equals to $H_D = \{(t6), (t9, t3), (t1)\}$ when eps equals to the variance of the topic distribution (Fig 8).

3.4. Online-mode Hashing

Hashing methods are batch-mode learning models that require huge data for learning an optimal model and cannot handle unseen data. Recent work address online mode by learning algorithms [26] that get hashing model accommodate to each new pair of data. But these approaches require the hashing model to be updated during each round based on the new pairs of data.

Our methods rely on topic models to build hash codes. These models do not require to be updated to make inferences about data not seen during training. In this way, the proposed hashing algorithms can work on large-scale and real-time data, as the size and the novelty of the collection does not influence the annotation process.

4. Experiments

As mentioned above (Section 2), it is difficult to interpret the similarity score calculated by metrics in a probability space. Since all of them are based on adding the distance between each dimension of the model (eq. 1, 2 and 3), distributions that share a fair amount of the less representative topics may still get higher similarity values than those that share the most representative ones specially if the model has a high number of dimensions.

Figures 9 and 10 show overlapped topic distributions of two pairs of documents. In the first case (fig 9), none of the most representative topics of each document is shared between them. However, the similarity score calculated from divergence-based metrics (eq 2) is higher than in the second case (fig 10), where the most representative topic is shared (topic 26). This behavior is due to the sum of the distances between the less representative topics (i.e. topics with a low weight value) being greater than the sum of the distances between the most representative ones (i.e. topic with a high weight value). In high-dimensional models, that sum may be more representative than the one obtained with the most

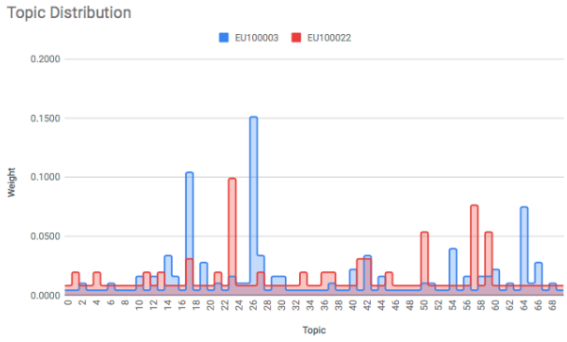

Fig. 9. Topic Distribution of two documents. Similarity score, based on JSD, is equals to 0.74

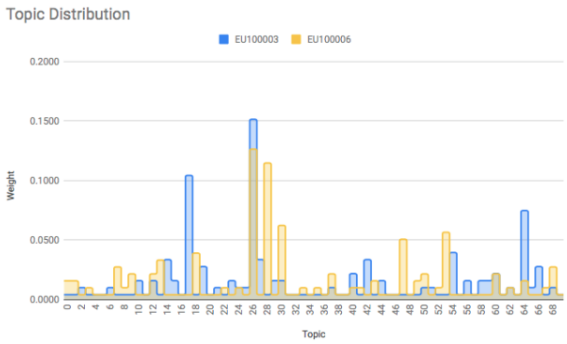

Fig. 10. Topic Distribution of two documents. Similarity score, based on JSD, is equals to 0.71

relevant topics, which are fewer in number than the less relevant ones.

The following experiments aim to validate that **hash codes based on hierarchical set of topics not only make it possible to search for similar documents with high accuracy, but also to extend queries with new restrictions and to offer information that helps explaining why two documents are similar**.

4.1. Datasets and Evaluation Metrics

Three datasets [9] are used to validate the proposed approach. The OPEN-RESEARCH[4] dataset consist of 500k research papers in Computer Science, Neuroscience, and Biomedical randomly selected from the Open Research Corpus [54]. The CORDIS[5] dataset contains 100k documents describing research and innovation projects funded by the European Union under a framework programme since 1990. The PATENTS dataset consists of 1M patents randomly selected from the USPTO[6] collection. For each dataset, documents are mapped to two latent topic spaces with different dimensions using LDA. We perform parameter estimation using collapsed Gibbs sampling for LDA [21] from the open-source librAIry [7] software. It is a framework that combines natural language processing (NLP) techniques with machine learning algorithms on top of the Mallet toolkit [40], an open-source machine learning package. The number of topics varies to study their influence on the performance of the algorithm (i.e. CORDIS-70 indicates a latent space created with 70 topics).

Experiments use JS divergence as an information-theoretically motivated metric in the probabilistic space created by topic models. Since it is a smoothed and symmetric alternative to the KL divergence, which is a standard measure for comparing distributions [13], it has been extensively used as state-of-the-art metric over topic distributions in literature [52][1][39]. Our upper bound is created from the brute-force comparison of the reference documents with all documents in the collection to obtain the list of similar documents.

In this scenario the goal is to minimize the accuracy loss introduced by hashing algorithms. Since this is a large-scale problem and an accuracy-oriented task, recall is not a good measure to be considered and precision is only relevant for sets much smaller than the total size of data (between 3-5 candidates).

All the experimental results are averaged over random training/set partitions. For each topic space, 100 documents are selected as references, and the remaining documents as search space. As noted above, only p@5 will be used to report the results of the experiments.

4.2. Retrieving Similar Documents

It is challenging to create an exhaustive gold standard, given the significant amount of human labour that is required to get a comprehensive view of the subjects being covered in it. In order to overcome this problem, the list of similar documents to a given one is obtained after comparing the document with all the documents of the repository and sorting the result. We have observed that different distance functions perform similarly in this scenario (figs 1 to 4), so we have decided to use only the JS divergence (eq. 2) in our experiments.

[4]https://labs.semanticscholar.org/corpus/
[5]https://data.europa.eu/euodp/data/dataset/cordisref-data

[6]https://www.uspto.gov/learning-and-resources/ip-policy/economic-research/research-datasets

OPEN-RES-100 (p@5)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	0.22	0.20	**0.86**	1.00	0.66	0.80
3	0.23	0.20	**0.87**	1.00	0.81	1.00
4	0.27	0.20	**0.89**	1.00	0.86	1.00
5	0.27	0.20	**0.92**	1.00	0.89	1.00
6	0.27	0.20	**0.94**	1.00	0.92	1.00

Table 1

Precision at 5 (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on Open Research dataset using a model with 100 topics. LEVEL column indicates the number of hierarchies used.

OPEN-RES-500 (p@5)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	0.23	0.20	**0.76**	0.80	0.67	0.80
3	0.24	0.20	**0.80**	1.00	0.71	0.80
4	0.25	0.20	**0.83**	1.00	0.74	0.80
5	0.25	0.20	**0.86**	1.00	0.81	1.00
6	0.24	0.20	**0.89**	1.00	0.86	1.00

Table 2

Precision at 5 (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on Open Research dataset using a model with 500 topics. LEVEL column indicates the number of hierarchies used.

CORDIS-70 (p@5)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	0.18	0.20	**0.92**	1.00	0.66	0.70
3	0.20	0.20	**0.92**	1.00	0.80	0.80
4	0.22	0.20	**0.94**	1.00	0.86	1.00
5	0.23	0.20	**0.91**	1.00	0.89	1.00
6	0.19	0.20	**0.92**	1.00	0.91	1.00

Table 3

Precision at 5 (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on CORDIS dataset using a model with 70 topics. LEVEL column indicates the number of hierarchies used.

CORDIS-150 (p@5)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	0.19	0.20	**0.88**	1.00	0.78	0.80
3	0.19	0.20	**0.92**	1.00	0.80	1.00
4	0.25	0.20	**0.91**	1.00	0.82	1.00
5	0.25	0.20	**0.91**	1.00	0.83	1.00
6	0.27	0.20	**0.91**	1.00	0.86	1.00

Table 4

Precision at 5 (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on CORDIS dataset using a model with 150 topics. LEVEL column indicates the number of hierarchies used.

Only the top N documents obtained from this method are used as reference set to measure the performance of the algorithms proposed in this paper. The value of N is equals to 0.5% of the corpus size (i.e. if the corpus size is equal to 1000 elements, only the top 5 most similar documents are considered relevant for a given document). This value has been considered after reviewing datasets used in similar experiments [30][39]. In those experiments, the reference data is obtained from existing categories, and the minimum average between corpus size and categorized documents is around 0.5%.

Once the reference list of documents similar to a given one is defined, the most similar documents through the proposed methods (i.e. threshold-based hierarchical hashing method (thhm), centroid-based hierarchical hashing method (chhm) and density-based hierarchical hashing method (dhhm)) are also obtained. An inverted index has been implemented by using Apache Lucene[7] as document repository. The source code of both the algorithms and tests is publicly available [9].

Let's look at an example to better understand the procedure. We want to measure the accuracy and data size ratio used to identify the top5 similar documents to a new document d_1 from a corpus of 1000 documents . The similarity between $d1$ and all the documents in the corpus is calculated based on JS divergence. The top50 (0.5%) documents with the highest values will be the set of documents considered as similar to d_1. As we are going to use an ANN-based approach, we need the hash expressions of all documents to measure similarity. The data structure proposed in this work is a hierarchy of sets of topics, so that the most similar documents are those that share most of the topics at the highest levels of the hierarchy.

The representational model for this example only considers 8 topics, that is, a document is described by a vector with 8 dimensions where each dimension corresponds to a topic (i.e $[t0, t1, t2, t3, t4, t5, t6, t7]$) and its value will be the weight of that topic in the document, for example $d_1 = [0.18, 0.15, 0.2, 0.05, 0.14, 0.11, 0.09, 0.08]$. The hierarchy level ($L$) will be equal to 2, i.e. the hash expression has two hierarchical sets of topics: $h = \{h_0, h_1\}$.

According to methods described at Section 3.3, there are 3 ways to create the hierarchical hash codes for documents:

1. threshold-based (*thhm*): 2 thresholds are defined as described in section 3.3.1, for example 0.15 and

[7] http://lucene.apache.org

0.1 . h_0 includes the topics with a weight greater than 0.15, and h_1 the remaining topics with a weight greater than 0.1. Then $h_0 = \{t0, t1, t2\}$ and $h_1 = \{t4, t5\}$. Based on the hash expression $h = \{(t0, t1, t2), (t4, t5)\}$, the documents that share more topics in those levels (i.e $h0 = (t0$ OR $t1$ OR $t2)$, $h1 = (t4$ OR $t5))$ or in other levels but with less relevance are ordered. Since there are many topics in the expression, potentially many documents are similar when sharing at least one of them. This increases the data ratio. Accuracy is also affected, as the algorithm is not able to bring under the same bucket similar documents. In short, the hash expression is not representative of the document, for the given exploratory task.

2. centroid-based (*chhm*): sets of topics are created using a clustering algorithm based on centroids as described in section 3.3.2. The cardinalities of the hierarchical groups are generally more uniform with this method. Since $k = L + 1 = 3$ in this example, $h_0 = \{t0, t2\}$ and $h_1 = \{t1, t4\}$. The number of representative topics at each level of the hierarchy is usually lower, and this causes the data ratio used to discover similar documents to decrease as well. This approach increases the precision because now the hierarchy is more selective to distinguish similar documents. However, the size of region of similar candidates is still high.
3. density-based (*dhhm*): now the clustering algorithm is based on how dense certain regions in the topic relevance dimensions are as described in section 3.3.3. It can group topics that have unbalanced distributions and, therefore, generates more discriminating hash expressions than with the previous algorithm. In the example, we would have a hash expression like this: $h_0 = \{t2\}$ and $h_1 = \{t0\}$. This significantly reduces the data ratio used to discover similar documents and does not excessively penalize accuracy. Obviously, increasing L (i.e. number of hierarchies) increases precision, but with $L > 3$ that gain is not so significant.

As it can be seen in tables 1 to 6, the mean and median of precision are calculated to compare the performance of the methods. In this assessment environment, the variance is not robust-enough because score values don't follow a normal distribution. We consider the result obtained as significant, based on the fact that mean and median values are fairly close. The centroid-based method (chhm) and the density-based method (dhhm)

PATENTS-250 (p@5)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	0.03	0.00	**0.71**	0.80	0.67	0.80
3	0.08	0.00	**0.91**	1.00	0.90	1.00
4	0.11	0.00	**0.95**	1.00	**0.95**	1.00
5	0.12	0.00	0.95	1.00	**0.96**	1.00
6	0.11	0.00	**0.97**	1.00	**0.97**	1.00

Table 5

Precision at 5 (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on Patents dataset using a model with 250 topics. LEVEL column indicates the number of hierarchies used.

PATENTS-750 (p@5)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	0.02	0.00	**0.77**	0.80	0.76	0.80
3	0.04	0.00	0.94	1.00	**0.95**	1.00
4	0.06	0.00	**0.97**	1.00	**0.97**	1.00
5	0.08	0.00	**0.97**	1.00	**0.97**	1.00
6	0.06	0.00	**0.97**	1.00	**0.97**	1.00

Table 6

Precision at 5 (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on Patents dataset using a model with 750 topics. LEVEL column indicates the number of hierarchies used.

OPEN-RES-100 (data-ratio)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	99.8	99.9	45.2	45.9	**4.9**	2.5
3	99.9	99.9	74.4	77.6	**13.4**	10.7
4	99.9	99.9	87.4	90.2	**27.2**	22.8
5	99.9	99.9	95.4	96.3	**49.9**	42.6
6	99.9	99.9	97.9	98.7	**72.2**	65.8

Table 7

Data size ratio used (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on Open Research dataset and 100 topics.

OPEN-RES-500 (data-ratio)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	95.9	96.3	22.2	22.1	**1.4**	0.3
3	99.1	99.2	43.9	43.7	**5.1**	4.1
4	99.6	99.6	57.1	57.3	**11.7**	10.3
5	99.6	99.6	70.7	70.7	**28.8**	22.0
6	99.9	99.9	81.5	80.6	**50.3**	40.1

Table 8

Data size ratio used (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on Open Research dataset and 500 topics.

CORDIS-70 (data-ratio)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	99.9	99.9	51.3	56.3	**5.1**	5.0
3	99.9	99.9	84.8	89.5	**10.5**	10.6
4	99.9	99.9	96.1	97.6	**20.8**	19.5
5	99.9	99.9	98.9	99.4	**35.0**	32.7
6	99.9	99.9	99.7	99.8	**53.1**	51.2

Table 9

Data size ratio used (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on CORDIS dataset and 70 topics.

CORDIS-150 (data-ratio)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	99.9	99.9	40.9	41.2	**3.1**	2.9
3	99.9	99.9	75.3	76.7	**6.2**	6.1
4	99.9	99.9	90.0	92.1	**12.1**	11.8
5	99.9	99.9	96.4	96.9	**21.6**	20.6
6	99.9	99.9	98.1	98.9	**36.5**	33.9

Table 10

Data size ratio used (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on CORDIS dataset and 150 topics.

PATENTS-250 (data-ratio)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	99.9	99.9	43.2	32.7	**35.1**	23.0
3	99.9	100.0	82.4	100.0	**78.2**	100.0
4	99.9	100.0	96.5	100.0	**95.1**	100.0
5	99.9	99.9	99.2	100.0	**98.9**	100.0
6	100.0	100.0	99.8	100.0	**99.7**	100.0

Table 11

Data size ratio used (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on Patents dataset and 250 topics.

PATENTS-750 (data-ratio)

LEVEL	THHM		CHHM		DHHM	
	mean	*median*	*mean*	*median*	*mean*	*median*
2	99.9	100.0	35.2	23.6	**31.8**	19.9
3	99.9	99.9	81.4	99.8	**79.6**	98.8
4	99.9	99.9	96.5	99.9	**95.5**	99.5
5	97.7	96.6	99.0	99.9	**98.6**	99.7
6	99.1	98.6	99.7	99.9	**99.5**	99.8

Table 12

Data size ratio used (*mean* and *median*) of threshold-based (THHM), centroid-based (CHHM) and density-based (DHHM) hierarchical hashing methods on Patents dataset and 750 topics.

show a similar behaviour to the one offered by the use of brute force by means of JS divergence.

In terms of efficiency, we consider the times to compare pairs of topic distributions constant, and we focus on the number of comparisons needed. Thus, algorithms with larger candidate spaces will be less efficient than others when the accuracy in both is the same. Tables 7-12 show the percentage of the corpus used by each of the algorithms to discover similar documents. Tables 1-6 show the accuracy of each algorithm for each of these scenarios. Density-based algorithm (dhhm) shows better balance between accuracy and volume of information (efficiency). It uses smaller samples (i.e lower ratio size) than others in all tests and even when it only uses a subset that is a 6.2% (Table 10) of the entire corpus, it obtains an accuracy of 0.808 (Table 4).

The precision achieved by the algorithm based on density (dhhm), which is much more restrictive than the others, suggests that few topics are required to represent a document in order to obtain similar ones. In addition, the number of topics does not seem to influence the performance of the algorithms, since their precision values are similar among the datasets of the same corpus. This shows that hashing methods based on hierarchical set of topics are robust to models with different dimensions.

The behavior of the algorithms have also been analyzed when the number of topics in the model varies. Models with 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000 topics were created from the CORDIS corpus. For each model, the p@5 of the hashing methods is calculated taking into account the hierarchy levels: 2, 3, 4, 5 and 6. Figures 11 to 13 show the results obtained for each algorithm. It can be seen how the performance, i.e precision, of each of the algorithms is not influenced by the dimensions of the model.

4.3. Exploration

In a certain domain, we may want to retrieve similar documents to one given. For example, searching for articles in the Biomedical domain that are similar to an article about Semantic Web. In terms of topics this kind of search requires to narrow down the initial search space to a subset with only documents that contain the topics that better describe the queried domain.

Existing hashing techniques based on a binary-code Hamming space do not allow to customize the search query beyond the reference document itself. However, the algorithms proposed in this work allow adding new restrictions to the initial query based on the reference document, since they use a hierarchy of set of topics as hash codes.

Through the following example we describe the workflow to enable such retrieval operations. For sim-

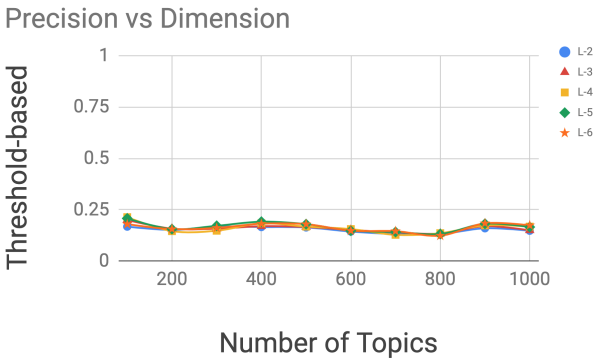

Fig. 11. Precision at 5 (*mean*) of threshold-based hashing method when number of topics varies in CORDIS dataset.

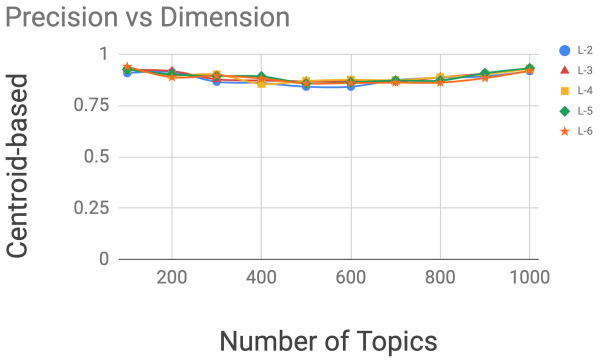

Fig. 12. Precision at 5 (*mean*) of centroid-based hashing method when number of topics varies in CORDIS dataset.

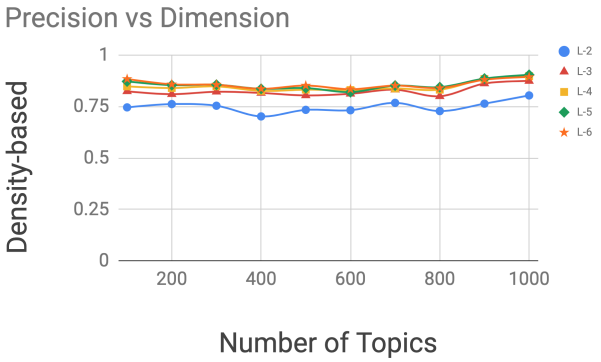

Fig. 13. Precision at 5 (*mean*) of density-based hashing method when number of topics varies in CORDIS dataset.

plicity we consider hash expressions with only two hierarchy levels. The reference document d_1 has the following hash expression: $h = \{h_0, h_1\} = \{(t10), (t18)\}$.

The first query, $Q1$, searches for documents similar to the reference document d_1 among all documents in the corpus. One of the ways to formalise this query looks like this: $Q1 = h_0 : t10\hat{}100$ or $h_0 : t18\hat{}50$ or $h_1 : t10\hat{}50$ or $h_1 : t18\hat{}100$. It sets a maximum boost (100) when the same restrictions as the reference document ($t10$ in h_0 and $t18$ in h_1) are fulfilled, and a lower boost (50) for the others ($t18$ in h_0 and $t10$ in h_1). In the specific case of applying this query to the CORDIS dataset, we observed that most of the retrieved documents included topic t18 (fig 14).

But if we were only interested in similar documents to d_1 that have topic $t10$, we could restrict the previous query $Q1$ to express this condition in the following way: $Q2 = (h_0 : t10\hat{}100$ or $h_1 : t10\hat{}50)$ and $(h_1 : t10\hat{}50$ or $h_1 : t18\hat{}100)$. The result obtained by $Q2$ (fig 14) shows that the condition has been considered since there is a balance between topics $t10$ and $t18$ among the documents similar to d_1.

This type of restrictions based on the semantics offered by topics in the hash expression get enabled thanks to the methods proposed in this work.

OPEN-RESEARCH-100			
hash	**q1**	**q2**	**ratio**
thhm	499,755	160,660	67.8
chhm	356,111	1,976	99.44
dhhm	49,068	766	98.43

Table 13

Number of documents similar to a given one (q1) and also in a specific domain (q2) for threshold-based (thhm), centroid-based (chhm) and density-based (dhhm) hierarchical hashing methods.

5. Conclusions

The usefulness of topics created by probabilistic models when exploring collections of scientific articles on large-scale has been widely studied in the literature. Each document in the corpus is described by probability distributions that measure the presence of those topics in their content. These vectors can also be used to measure the similarity between documents by using metrics such as Jensen-Shannon divergence. But with large amounts of items in the collection, discovering the entire set of nearest neighbors to a given document would be infeasible. Due to the low storage cost and fast retrieval speed, hashing is one of the popular solutions for approximate nearest neighbors. However, existing hashing methods for probability distributions only focus on the efficiency of searches from a given document, without handling complex queries or offering hints about why one document is considered more similar than another. A new data structure is proposed to represent

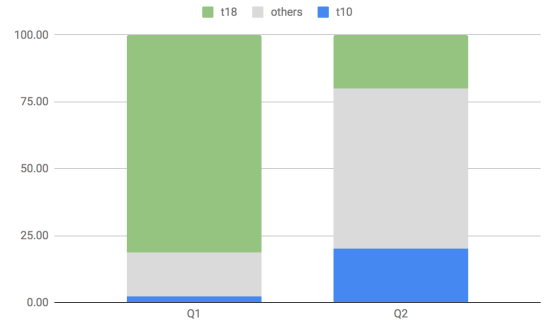

Fig. 14. Most relevant topics in similar documents from using a document as query (Q1) and setting topic t10 as mandatory (Q2).

hash codes, based on topic hierarchies created from the topic distributions. This approach has proven to obtain high-precision results and can accommodate additional query restriction. This way of encoding documents can also help to understand why two documents are similar, based on the intersection of topics at hierarchies of relevance.

In this paper we have focused on (1) comparing the performance of topic-based hashing methods with respect to the distance metrics based on probability distributions (e.g. JS divergence), (2) their ability to support more complex queries based on topic-based filters and (3) the expressiveness of their annotations (topics hierarchically divided into groups with different degrees of semantic specificity) to justify the relations obtained.

A manually annotated corpus with content similarity relations would further confirm the ability of the metrics proposed in this paper to reflect similarity as humans perceive it. Ongoing work on this line includes the creation of questionnaires [8] to more accurately capture how similar two documents are from the perspective of human evaluators who read them both. This is an ambitious task that need to deals with the evaluators' own interpretation of similarity. What an expert perceives as different (since his knowledge in the domain allows him to identify discrepancies between the two texts), may be considered as similar by an inexperienced user that might not be able to capture those fine grained differences.

The next steps in our research are to extend the metric proposed in this paper from the point of view of the perception of similarity that a human makes, and to perform a more in-depth investigation about the meaning of the topics grouped by levels of relevance.

[8] http://librairy.linkeddata.es/survey

6. Acknowledgments

This research was supported by the Spanish Ministerio de Economía, Industria y Competitividad and EU FEDER funds under the DATOS 4.0: RETOS Y SOLUCIONES - UPM Spanish national project (TIN2016-78011-C4-4-R)

References

[1] N. Aletras, T. Baldwin, J. H. Lau, and M. Stevenson. Evaluating topic representations for exploring document collections. *Journal of the Association for Information Science and Technology*, 68(1):154–167, 2017.

[2] N. Aletras and M. Stevenson. Measuring the Similarity between Automatically Generated Topics. In *EACL*, pages 22–27, 2015.

[3] A. Andoni and P. Indyk. Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 459–468. IEEE, 2006.

[4] A. Andoni, P. Indyk, H. L. Nguyen, and I. Razenshteyn. Beyond Locality-Sensitive Hashing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, 6 2013.

[5] A. Andoni, P. Indyk, H. L. NguyáźĚn, and I. Razenshteyn. Beyond Locality-Sensitive Hashing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1018–1028. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1 2014.

[6] A. Andoni and I. Razenshteyn. Optimal Data-Dependent Hashing for Approximate Near Neighbors. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing - STOC '15*, pages 793–801, New York, New York, USA, 2015. ACM Press.

[7] C. Badenes-Olmedo, J. L. Redondo-Garcia, and O. Corcho. Distributing Text Mining tasks with librAIry. In *17th ACM Symposium on Document Engineering (DocEng)*, 2017.

[8] C. Badenes-Olmedo, J. L. Redondo-Garcia, and O. Corcho. Efficient Clustering from Distributions over Topics. In *9th International Conference on Knowledge Capture (K-CAP)*, page 8, 2017.

[9] C. Badenes-Olmedo, J. L. Redondo-Garcia, and O. Corcho. Large-scale Topic-based Search Java Algorithm, 2019.

[10] D. Blei, L. Carin, and D. Dunson. Probabilistic topic models. *IEEE Signal Processing Magazine*, 27(6):55–65, 2010.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003.

[12] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the Web. *Computer Networks and ISDN Systems*, 29(8-13):1157–1166, 9 1997.

[13] S.-H. Cha. Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):1–8, 2007.

[14] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing - STOC '02*, page 380, New York, New York, USA, 2002. ACM Press.

[15] X. Cheng, X. Yan, Y. Lan, and J. Guo. BTM : Topic Modeling over Short Texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2928–2941, 2014.

[16] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry - SCG '04*, page 253, New York, New York, USA, 2004. ACM Press.

[17] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. Harshman. Indexing by Latent Semantic Analysis. *JASIS*, 41(6):391–407, 1990.

[18] D. Endres and J. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 7 2003.

[19] C. J. Gatti, J. D. Brooks, and S. G. Nurre. A Historical Analysis of the Field of OR/MS using Topic Models. *CoRR*, abs/1510.0, 2015.

[20] D. Greene and J. P. Cross. Exploring the political agenda of the european parliament using a dynamic topic modeling approach. *Political Analysis*, 25(1):77–94, 2016.

[21] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl:5228–35, 2004.

[22] T. L. Griffiths, M. Steyvers, and J. B. Tenenbaum. Topics in semantic representation. *Psychological Review*, 114(2):211–244, 2007.

[23] D. Hall, D. Jurafsky, and C. D. Manning. *Studying the History of Ideas Using Topic Models*. Association for Computational Linguistics, 2008.

[24] J. He, L. Li, and X. Wu. A self-adaptive sliding window based topic model for non-uniform texts. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, volume 2017-Novem, pages 147–156, 2017.

[25] T. Hofmann. Probabilistic Latent Semantic Indexing. *SIGIR*, pages 50–57, 1999.

[26] L. K. Huang, Q. Yang, and W. S. Zheng. Online Hashing. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6):2309–2322, 2018.

[27] P. Indyk and R. Motwani. Approximate nearest neighbors. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing - STOC '98*, pages 604–613, New York, New York, USA, 1998. ACM Press.

[28] J. Ji, J. Li, S. Yan, Q. Tian, and B. Zhang. Min-Max Hash for Jaccard Similarity. In *2013 IEEE 13th International Conference on Data Mining*, pages 301–309. IEEE, 12 2013.

[29] K. Krstovski and D. A. Smith. A Minimally Supervised Approach for Detecting and Ranking Document Translation Pairs. In *Workshop on Statistical MT*, 2011.

[30] K. Krstovski, D. A. Smith, H. M. Wallach, and A. McGregor. Efficient Nearest-Neighbor Search in the Probability Simplex. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval - ICTIR '13*, pages 101–108, New York, New York, USA, 2013. ACM Press.

[31] B. Kulis and K. Grauman. Kernelized Locality-Sensitive Hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1092–1104, 6 2012.

[32] P. Li and C. König. b-Bit minwise hashing. In *Proceedings of the 19th international conference on World wide web - WWW '10*, page 671, New York, New York, USA, 2010. ACM Press.

[33] P. Li, A. B. Owen, and C.-H. Zhang. One Permutation Hashing. *Advances in Neural Information Processing*, 2012.

[34] W. Liu, C. Mu, S. Kumar, and S.-F. Chang. Discrete Graph Hashing. *NIPS*, 2014.

[35] W. Liu, C. Mu, S. Kumar, and S.-F. Chang. Discrete Graph Hashing. *Advances in Neural Information Processing Systems*, pages 3113–3121, 2014.

[36] Y. Liu, J. Cui, Z. Huang, H. Li, and H. T. Shen. SK-LSH. An efficient index structure for Approximate Nearest Neighbor Search. *Proceedings of the VLDB Endowment*, 7(9):745–756, 5 2014.

[37] N. Ljubešić, D. Boras, N. Bakarić, and J. Njavro. Comparing measures of semantic similarity. *Proceedings of the International Conference on Information Technology Interfaces, ITI*, pages 675–681, 2008.

[38] H.-m. Lu, C.-p. Wei, and F.-y. Hsiao. Modeling healthcare data using multiple-channel latent Dirichlet allocation. *JOURNAL OF BIOMEDICAL INFORMATICS*, 60:210–223, 2016.

[39] X. Mao, B.-S. Feng, Y.-J. Hao, L. Nie, H. Huang, and G. Wen. S2JSD-LSH: A Locality-Sensitive Hashing Schema for Probability Distributions. In *AAAI*, 2017.

[40] A. McCallum. MALLET: A Machine Learning for Language Toolkit, 2002.

[41] A. Niekler and P. Jähnichen. Matching results of latent dirichlet allocation for text. In *Proceedings of ICCM*, pages 317–322, 2012.

[42] R. OâĂŹDonnell, Y. Wu, and Y. Zhou. Optimal Lower Bounds for Locality-Sensitive Hashing (Except When q is Tiny). *ACM Transactions on Computation Theory*, 6(1):1–13, 3 2014.

[43] J. OâĂŹNeill, C. Robin, L. OâĂŹBrien, and P. Buitelaar. An analysis of topic modelling for legislative texts. *CEUR Workshop Proceedings*, 2143, 2017.

[44] M. Paul and R. Girju. Topic Modeling of Research Fields: An Interdisciplinary Perspective. In *Recent Advances in Natural Language Processing*, pages 337–342, 2009.

[45] M. J. Paul and M. Dredze. Discovering health topics in social media using topic models. *PLoS ONE*, 9(8), 2014.

[46] S. Petrovic, M. Osborne, and V. Lavrenko. Streaming First Story Detection with application to Twitter. *NAACL*, 2010.

[47] D. Ramage, S. Dumais, and D. Liebling. Characterizing Microblogs with Topic Models. *Icwsm*, pages 1–8, 2010.

[48] D. Ramage, E. Rosen, J. Chuang, C. D. Manning, and D. A. McFarland. Topic Modeling for the Social Sciences. In *Twenty-Third Annual Conference on Neural Information Processing Systems*, pages 1–4, 2009.

[49] D. Ravichandran, P. Pantel, and E. H. Hovy. Randomized Algorithms and NLP: Using Locality Sensitive Hash Functions for High Speed Noun Clustering. *ACL*, 2005.

[50] M. D. Tapi Nzali, S. Bringay, C. Lavergne, C. Mollevi, and T. Opitz. What Patients Can Tell Us: Topic Analysis for Social Media on Breast Cancer. *JMIR medical informatics*, 5(3):e23, 7 2017.

[51] K. Terasawa and Y. Tanaka. Spherical LSH for Approximate Nearest Neighbor Search on Unit Hypersphere. In *Algorithms and Data Structures*, pages 27–38. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[52] W. B. Towne, C. P. Rosé, and J. Herbsleb. Measuring Similarity Similarly: LDA and Human Perception. *ACM Transactions on Intelligent Systems and Technology ACM Reference Format ACM Trans. Intell. Syst. Technol*, 7(2):1–25, 2016.

[53] S. Vijayanarasimhan, P. Jain, and K. Grauman. Hashing Hyperplane Queries to Near Points with Applications to Large-Scale Active Learning. *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, 36(2):276–288, 2 2014.

[54] A. Waleed, G. Dirk, B. Chandra, B. Iz, C. Miles, D. Doug, D. Jason, E. Ahmed, F. Sergey, H. Vu, K. Rodney, K. Sebastian, L. Kyle, M. Tyler, O. Hsu-Han, P. Matthew, P. Joanna, S. Sam, W. Lucy, Lu, W. Chris, Y. Zheng, v. Z. Madeleine, and E. Oren. Construction of the Literature Graph in Semantic Scholar. In *NAACL*, 2018.

[55] H. M. Wallach, D. M. Mimno, and A. McCallum. Rethinking LDA: Why Priors Matter. In *Neural Information Processing Systems (NIPS)*, pages 1973–1981, 2009.

[56] J. Wang, W. Liu, S. Kumar, and S.-F. Chang. Learning to Hash for Indexing Big DataâĂŤA Survey. *Proceedings of the IEEE*, 104(1):34–57, 1 2016.

[57] L. Xing and M. J. Paul. Diagnosing and Improving Topic Models by Analyzing Posterior Variability. *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 6005–6012, 2018.

[58] T. Zhang, Guo-Jun Qi, Jinhui Tang, and J. Wang. Sparse composite quantization. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4548–4556. IEEE, 6 2015.

[59] W.-L. Zhao, H. Jégou, and G. Gravier. Sim-min-hash. In *Proceedings of the 21st ACM international conference on Multimedia - MM '13*, pages 577–580, New York, New York, USA, 2013. ACM Press.

[60] Y. Zhen, Y. Gao, D.-Y. Yeung, H. Zha, and X. Li. Spectral Multimodal Hashing and Its Application to Multimedia Retrieval. *IEEE Transactions on Cybernetics*, 46(1):27–38, 1 2016.