Efficient Exploration of Scientific Articles using Topic-based Hashing Algorithms

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
[b] *Amazon Research, Cambridge, UK*
E-mail: jluisred@amazon.com

Abstract. Searching for similar documents and exploring major themes covered by groups of documents are common actions when browsing collections of scientific articles. This manual knowledge-intensive task may become less tedious and may even lead to unexpected findings if algorithms are applied to help researchers. Most text mining algorithms represent documents in a common feature space that abstracts away from the specific sequence of words used in them. Probabilistic Topic Models reduce that feature space by annotating documents with thematic information. On this low-dimensional latent space some locality-sensitive hashing algorithms have been proposed to perform document similarity search. However, thematic information is hidden behind binary 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 to extend 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

In recent years, the data in the Internet keeps growing exponentially. 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 perform in such huge collections of data and require more efficient approaches than having to calculate all pairwise similarities.

Most text mining algorithms represent documents in a common feature space that abstracts the specific sequence of words used in each document and, with appropriate representations, facilitate the analysis of relationships between documents even 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) [13], Probabilistic Latent Semantic Indexing (PLSI) [19] and more recently, Latent Dirichlet Allocation (LDA) [8], which is the simplest probabilistic topic model (PTM) [7], 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 is able to identify topics in previously unseen texts.

One of the greatest advantages 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 doc-

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

ument, represented as 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 [17][15], health [35] [30] [40], legal [33][16], news [18] and social networks [37][11]. 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 $O(n^2)$ comparisons for near-neighbor detection tasks or $O(kn)$ computations when k queries are compared against a data set 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 [27] [3] [50]. 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) [39][36][22], even in high-dimensional simplex spaces handling information-theoretically motivated metrics (e.g. Hellinger, Kullback-Leibler divergence, Jensen-Shannon divergence) as demonstrated by [31].

However, existing hashing methods lose the exploratory feature that topic models offer and the explanatory power that topics have to support the document similarity. Searching for similar documents in a domain described by a set of topics cannot be performed using binary hash codes. Moreover, metrics in simplex space are difficult to interpret and the ability to explain the similarity score can be helpful. While other distances only using vectors are simply agnostic to the human concept of themes, topic models can offer a reason of why two documents are similar. Thus, in this paper, we will propose a hashing algorithm that (1) groups similar documents (2) preserving their topic distributions (3) even for previously unseen documents. We make the following contributions in this paper:

- **novel hashing algorithm** based on topic distributions that not only performs efficiently searches, but also allows to extend those queries with new restrictions and provides explanatory information about the similarity.
- source-code and data-sets to facilitate other researchers to replicate our experiments and validate their own ideas [1]

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 [31]. 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) (Eq.3) distance are often more reasonable [31]:

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

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

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

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

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, so do not satisfy triangle inequality [10]. To solve it, [14] introduced a new metric called $S2JSD$ that satisfies the symmetry, non-negativity and triangle inequality. It is the square root of two times the *JS* divergence as follows:

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

[1] https://github.com/cbadenes/Large-scale-Topic-based-Search

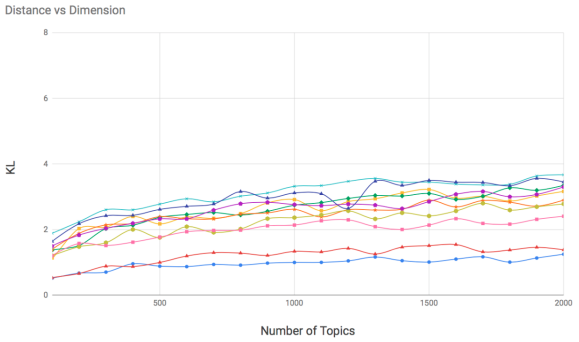

Fig. 1. Distance values based on KL-divergence between 10 pair of documents from topic models with 100-to-2000 dimensions.

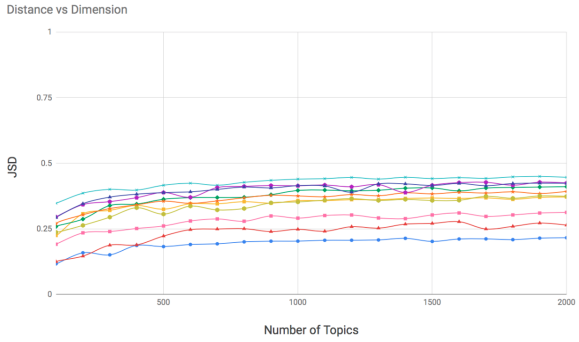

Fig. 2. Distance values based on JS-divergence between 10 pair of documents from topic models with 100-to-2000 dimensions.

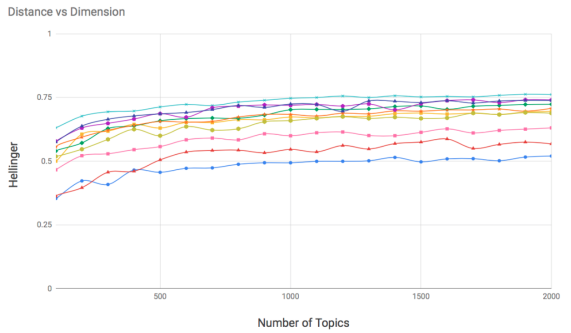

Fig. 3. Distance values based on He-divergence between 10 pair of documents from topic models with 100-to-2000 dimensions.

However, making sense out of the similarity score is not easy. As shown in figures 1, 2, 3, and 4 , given a set of pairs of documents, their similarity scores vary according to the space dimensions, that is, according to the number of topics. So those pairs fluctuate from being more to less distant when changing the number of topics.

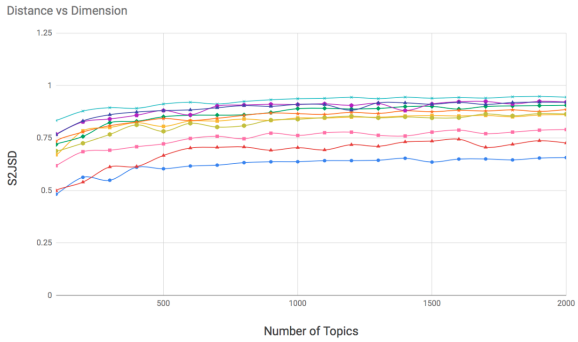

Fig. 4. Distance values based on $S2JSD$ between 10 pair of documents from topic models with 100-to-2000 dimensions.

Distance between documents generally increases 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. This behaviour highlights the difficulty of establishing absolute similarity thresholds and the complexity to measure distances taking into account all dimensions. These thresholds should be model-dependent rather than general and metrics flexible enough to handle dimensional changes. These challenges are taken over the proposed hashing algorithm by means of similarity levels as described next.

3. Hashing Topic Distributions

Hashing methods transform the data points from the original feature space into a binary-code Hamming space, where the similarity in the original space is preserved. They can learn hash functions (data-dependent) or use projections (data-independent) from the training data [46]. Data-independent unlike data-dependent methods 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 perform 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) [12], Leech lattice LSH [2], Spherical

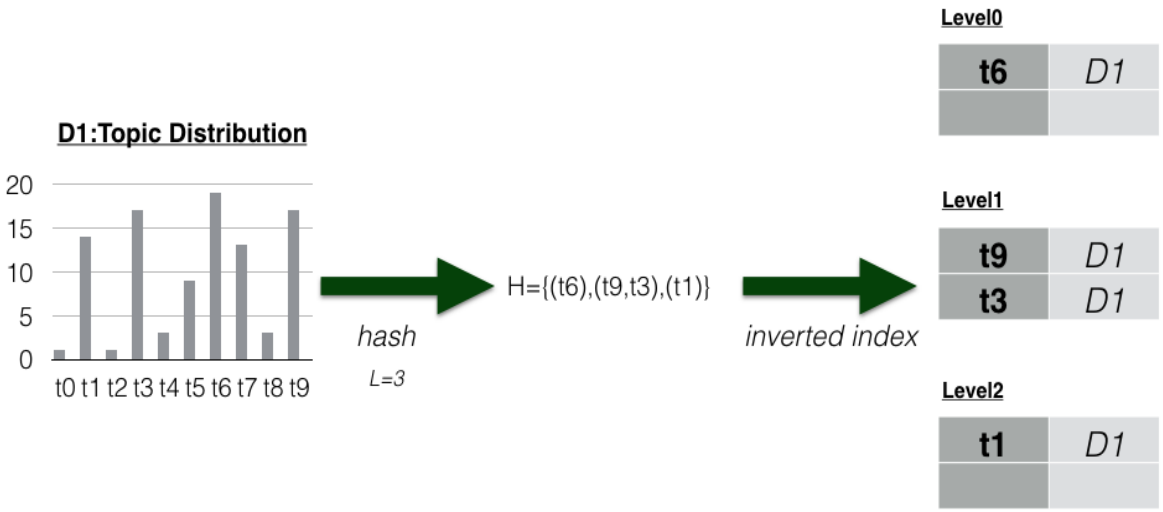

Fig. 5. Hash method based on hierarchical set of topics from a given topic distribution

LSH [41], and Beyond LSH [4]. Based on the θ distance many methods have been developed such as Kernel LSH [24] and Hyperplane hashing [43]. But only few works 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 [23]. But this solution does not consider the special attributions of probability distributions, such as Non-negative and Sum-equal-one. Recently, a hashing schema has been proposed [31] taking into account the symmetry, 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 [26], Min-max hash [21], B-bit minwise hashing [25] and Sim-min-hash [49].

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

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 3.2 some approaches to create hash codes based on this data structure are described.

3.1. 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.

3.2. 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.

3.2.1. Threshold-based Hierarchical Hashing Method

This approach is just an initial and naive way of grouping topics by threshold values for 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).

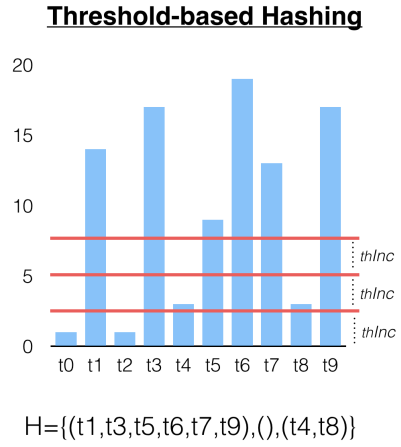

Fig. 6. Threshold-based Hierarchical Hash (L=3) from a Topic Distribution

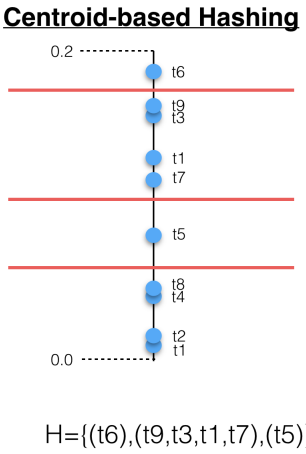

Fig. 7. Centroid-based Hierarchical Hash (L=3) from a Topic Distribution

3.2.2. 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 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 hiearchical hash H_C equals to $H_C = \{(t6), (t9,t7,t3,t1), (t5)\}$ (Fig 7).

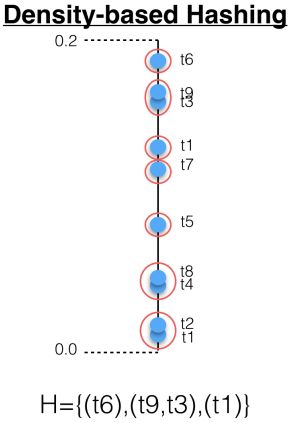

Fig. 8. Density-based Hierarchical Hash (L=3) from a Topic Distribution

3.2.3. *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 7).

4. Experiments

As mentioned above (Section 2), it is difficult to interpret the similarity score calculated by metrics in 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 9), 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 relevant topics, which are fewer in number than the less relevant ones.

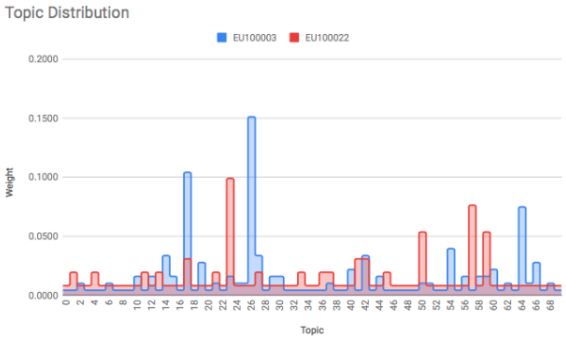

Fig. 9. Topic Distribution of two documents. Similarity score, based on JSD, is equals to 0.74

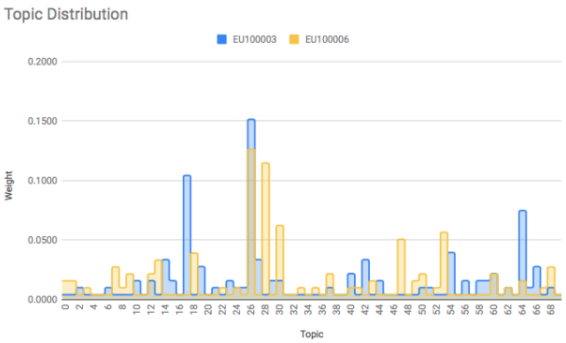

Fig. 10. Topic Distribution of two documents. Similarity score, based on JSD, is equals to 0.71

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 explain why two documents are similar.

4.1. *Data Sets and Evaluation Metrics*

Three datasets[2] are used to validate the proposed approach. OPEN-RESEARCH[3] dataset consist of 500k

[2]https://github.com/cbadenes/Large-scale-Topic-based-Search
[3]https://labs.semanticscholar.org/corpus/

research papers in Computer Science, Neuroscience, and Biomedical randomly selected from the Open Research Corpus [44]. CORDIS[4] dataset contains 100k documents describing research and innovation projects funded by the European Union under a framework programme since 1990. PATENTS dataset consist of 1M patents randomly selected from the USPTO[5] collection. For each dataset, documents are mapped to two latent topic spaces with different dimensions using LDA. The number of topics varies to measure their influence on the performance of the algorithm (i.e. CORDIS-70 indicates a latent space created with 70 topics) .

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@N is used to illustrate.

4.2. Retrieving Similar Documents

The creation of a gold-standard for measuring content similarity is a challenge in itself, since in large-scale corpora is rarely including this type of annotations. 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. The similarity metric used in experiments is JSD, due to is used in literature [42][1][31] and therefore the one used to build our reference test set to investigate how good the topic based hashing functions are performing compared to the state of the art.

Only the top N documents 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 top5 most similar documents are considered relevant for a given document). This value has been considered after reviewing datasets used in similar experiments [23][31]. 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. The inverted index has been implemented by using Apache Lucene[6] as document repository. The source code of both the algorithms and tests is publicly available[7].

As can be seen in tables 1, 2 and 3, precision of both the centroid-based method (chhm) and the density-based method (dhhm) to identify similar documents is close to results based on JS divergence.

The number of candidates to be considered as similar documents for each hashing algorithm helps to un-

	OPEN-RES-100			OPEN-RES-500		
L	thhm	chhm	dhhm	thhm	chhm	dhhm
2	0.226	**0.848**	0.660	0.228	**0.764**	0.676
3	0.230	**0.906**	0.812	0.250	**0.826**	0.714
4	0.280	**0.924**	0.864	0.267	**0.830**	0.745
5	0.278	**0.924**	0.896	0.252	**0.860**	0.816
6	0.276	**0.938**	0.926	0.288	**0.893**	0.869

Table 1

p@5 on Open-Research datasets for threshold-based (thhm), centroid-based (chhm) and density-based (dhhm) hierarchical hashing methods. L is the hierarchical level.

	CORDIS-70			CORDIS-150		
L	thhm	chhm	dhhm	thhm	chhm	dhhm
2	0.180	**0.928**	0.668	0.196	**0.918**	0.784
3	0.208	**0.936**	0.808	0.198	**0.920**	0.808
4	0.222	**0.930**	0.862	0.254	**0.920**	0.824
5	0.230	**0.910**	0.894	0.256	**0.920**	0.836
6	0.194	**0.920**	0.912	0.276	**0.912**	0.864

Table 2

p@5 on CORDIS datasets for threshold-based (thhm), centroid-based (chhm) and density-based (dhhm) hierarchical hashing methods. L is the hierarchical level.

	PATENTS-250			PATENTS-750		
L	thhm	chhm	dhhm	thhm	chhm	dhhm
2	0.020	**0.740**	0.672	0.123	**0.776**	0.764
3	0.080	0.890	**0.908**	0.050	0.934	**0.943**
4	0.128	0.942	**0.947**	0.200	0.957	**0.961**
5	0.208	0.971	**0.975**	0.500	**0.950**	0.937
6	0.230	**1.000**	0.990	0.000	**0.977**	0.933

Table 3

p@5 on Patents datasets for threshold-based (thhm), centroid-based (chhm) and density-based (dhhm) hierarchical hashing methods. L is the hierarchical level.

[4] https://data.europa.eu/euodp/data/dataset/cordisref-data

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

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

[7] https://github.com/cbadenes/Large-scale-Topic-based-Search

	OPEN-RES-100			OPEN-RES-500		
L	thhm	chhm	dhhm	thhm	chhm	dhhm
2	99.8	43.2	**4.9**	95.9	23.0	**1.4**
3	99.9	75.0	**13.4**	99.1	42.2	**5.1**
4	99.9	86.2	**27.2**	99.7	58.4	**10.8**
5	99.9	95.3	**49.9**	99.7	71.0	**28.1**
6	99.9	97.9	**72.2**	99.9	79.5	**46.0**

Table 4

Data size ratio used from Open Research datasets by threshold-based (thhm), centroid-based (chhm) and density-based (dhhm) hierarchical hashing methods.

	CORDIS-70			CORDIS-150		
L	thhm	chhm	dhhm	thhm	chhm	dhhm
2	99.9	52.1	**5.0**	99.9	40.8	**3.0**
3	99.9	83.6	**10.5**	99.9	74.3	**6.2**
4	99.9	96.0	**20.8**	99.9	90.5	**12.1**
5	99.9	99.1	**35.0**	99.9	96.2	**21.6**
6	99.9	99.7	**53.1**	99.9	98.1	**36.5**

Table 5

Data size ratio used from Cordis datasets by threshold-based (thhm), centroid-based (chhm) and density-based (dhhm) hierarchical hashing methods.

	PATENTS-250			PATENTS-750		
L	thhm	chhm	dhhm	thhm	chhm	dhhm
2	100.0	43.3	**35.1**	99.9	35.0	**31.8**
3	99.9	82.5	**78.2**	99.8	78.2	**76.2**
4	99.9	95.9	**94.2**	99.9	91.7	**90.2**
5	99.8	98.5	**97.7**	96.1	94.6	**92.9**
6	100.0	99.3	**98.9**	98.5	97.4	**96.4**

Table 6

Data size ratio used from Patents datasets by threshold-based (thhm), centroid-based (chhm) and density-based (dhhm) hierarchical hashing methods.

derstand its performance. Thus, algorithms with larger candidate spaces will be less efficient than others when the accuracy in both is the same, therefore lower ratio size is preferred at similar levels of accuracy. Tables 4, 5 and 6 show the average ratio of data used by algorithms. Density-based algorithm (dhhm) uses smaller datasets than others even using only a subset that is a 6.2% (Table 5) of the entire corpora with an accuracy of 0.808 (Table 2).

The precision achieved by the algorithm based on density (dhhm), much more restrictive than others, suggests that few topics are required to represent a document in order to obtain other similar ones. In addition,

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 7

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.

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.

4.3. Exploration

Similar documents to a given one may be required in a given domain. For example, searching for articles in Biomedical domain similar to an article about Semantic Web. It requires, in terms of topics, to reduce the initial search space to a subset with only documents that contain the topics that best describe the searched domain.

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

The experiment consists of comparing the distribution of the most relevant topics in similar documents when searching from a given document, and when limited to a domain defined by a few topics. Given a document whose most relevant topic is t10, the distribution of the most relevant topics in similar documents is obtained (Q1). The search is then reduced to a given domain, e.g. documents containing the topic t10. The distribution of the most relevant topics in similar documents is obtained again (Q2). As shown in fig 11, the presence of topic t10 equals that of topic t18. This shows that this extra restriction has been correctly considered in the query. Table 7 shows a reduction in the number of candidates between the second query and the first one.

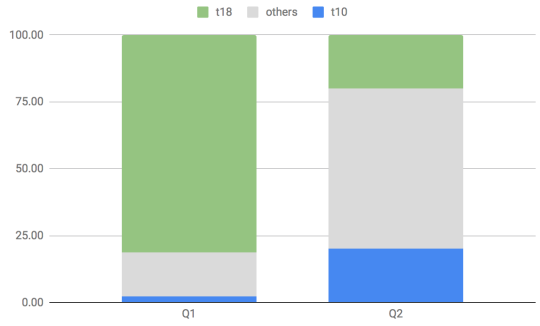

Fig. 11. Most relevant topics in similar documents from using a document as query (Q1) and setting topic t10 as mandatory (Q2).

5. Conclusions

The usefulness of topics created by probabilistic models is well known when exploring collections of scientific articles on large-scale. Each document in 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. In large-scale applications, it is usually time-consuming or impossible to return the exact nearest neighbors to a given document. 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 explaining why one document is considered more similar than another. A new data structure is proposed to represent hash codes based on topic hierarchies created from the topic distributions. It has proven to be a high-precision approach that can be also extended by adding extra query restrictions. This way of coding documents can also help to understand why two documents are similar, based on the intersection of topics at levels of relevance.

The next steps in this research line are to validate the metric proposed in this paper from the point of view of the perception of similarity that makes a human, and to understand the meaning of the topics grouped by levels of relevance.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

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

[18] 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.

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

[20] 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.

[21] 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.

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

[23] 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.

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

[25] 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.

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

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

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

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

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

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

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

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

[38] 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.

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

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

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

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.