SANDIA REPORT

SAND2021-15731
Printed December 2021

Sandia National Laboratories

Document Retrieval and Ranking using Similarity Graph Mean Hitting Times

Daniel M. Dunlavy, Peter A. Chew

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@osti.gov
Online ordering:	http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.gov
Online order:	https://classic.ntis.gov/help/order-methods

Document Retrieval and Ranking using Similarity Graph Mean Hitting Times

Daniel M. Dunlavy
Sandia National Laboratories
Albuquerque, NM
dmdunla@sandia.gov

Peter A. Chew
Galisteo Consulting Group, Inc.
Albuquerque, NM
pachew@galisteoconsulting.com

SAND2021-15731

ABSTRACT

We present a novel approach to information retrieval and document analysis based on graph analytic methods. Traditional information retrieval methods use a set of terms to define a query that is applied against a document corpus to identify the documents most related to those terms. In contrast, we define a query as a set of documents of interest and apply the query by computing mean hitting times between this set and all other documents on a document similarity graph abstraction of the semantic relationships between all pairs of documents. We present the steps of our approach along with a simple example application illustrating how this approach can be used to find documents related to two or more documents or topics of interest.

ACKNOWLEDGMENT

We thank Rich Lehoucq of Sandia National Laboratories for providing guidance on understanding the HITMIX graph mean hitting times algorithm. Galisteo Consulting Group, Inc., gratefully acknowledges the funding and support provided by the Office of Naval Research for the development of this report. This material is based upon work supported by the Office of Naval Research under Contract No. N00014-21-P-2006. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

CONTENTS

1. INTRODUCTION

Given a corpus, or collection, of text documents, analysts often use information retrieval methods to search, organize, and make sense of the contents of the corpus as a whole. Traditional information retrieval involves an analyst providing a set of search, or query, terms of interest and using various methods to find documents that include, or are related to, those terms [4, 27, 30]. However, in many analysis applications an analyst may not know *a priori* what terms are of most interest. In such applications, methods for analyzing text corpora are needed that provide both summary information about the corpus as a whole as well as a means for retrieving documents based on subsets of the corpus that are determined as part of an exploratory analysis process.

In this report, we describe a novel use of existing text analysis techniques coupled with a graph analysis technique for addressing the problem of document corpus analysis without the use of predetermined sets of search terms. Specifically, we leverage standard methods used in text retrieval based on natural language processing and topic modeling to identify summary information about a document corpus. This summary information is provided as a set of topics, each of which is a collection of weights identifying the importance of each term from the full document corpus in that topic. By analyzing this topic summary and the subsets of documents related to each topic, analysts then identify a set of documents of interest for further investigation and analysis. Finally, a graph abstraction of document similarities based on the topics relationships is used to provide a rank ordering of the similarity of each document in the corpus to the set of documents of interest.

This approach to text retrieval and analysis differs from traditional search techniques in that no search terms are required before analysis begins. Analysts are provided with topical summaries of a corpus, from which they can identify subsets of documents of interest. Note that document clustering is a standard approach for finding subsets of documents related by term or topic usage [23, 31]. However, standard clustering methods leverage only pairwise document relationships, whereas our approach can identify relationships between *sets* of documents. For applications where all possible term, topic, and document relationships cannot be identified from a given corpus, and where an analyst's previous experience and expertise with term and topic relationships in a given subject area must be leveraged, our approach provide a means for identifying document relationships based on such information.

The remainder of this reports is structured as follows. In Section 2, we describe the methods used in our approach. In Section 3, we illustrate the use of our method applied to a simple document collection. Finally, in Section 4, we provide conclusions related to our approach as well as some ideas for future research in this area.

2. METHODS

In this section, we describe the methods used in our approach for analyzing text corpora described in Section 1.

The overall process for analysis is as follows:

1. Convert text documents into a vector space representation.
2. Weight document vectors.
3. Compute topic model of document corpus.
4. Present summary of document topics to analyst.
5. Have an analyst indicate documents of interest using topic summary information.
6. Create document similarity graph using topic model.
7. Compute document similarities to set of documents of interest using graph analysis method.

We have implemented the methods in Python using the following packages: `pandas` (for data manipulation), `numpy` (dense linear algebra), `scipy` (sparse linear algebra), `scikit-learn` (vector space representations), `matplotlib` (visualization), and `hitmix` (graph analysis). In the descriptions of the methods below, we identify which packages and capabilities we leverage in our implementations, along with any new software developed specific for the analyses presented here.

2.1. Vector Space Representations of Documents

A corpus of documents can be represented using an $n \times m$ document-term matrix, A, where n is the number of documents and m is the number of terms in the corpus [30]. Such a representation supports a wide variety of analyses of document corpora, as we have demonstrated for retrieval [7, 20], topic modeling [5, 12, 13], named entity recognition [32], part-of-speech recognition [8, 16], clustering [12], classification [3], summarization [9, 17, 19], multilingual analysis [1, 2, 6, 7], and visualization [10, 11] tasks.

We use the `CountVectorizer` class in the `scikit-learn` Python package to create a document-term matrix. In the case studies presented in Section 3, we apply simple preprocessing of the documents to extract whitespace-delimited terms, convert all terms to lowercase, and remove numbers and URLs from the final set of terms used. More advanced document processing based on lemmatization [21] and stemming [29], has demonstrated utility in several text analysis applications [5, 19] and can be applied using the `CountVectorizer` class. However, we limit our use here to simple document processing to illustrate the creation of vector space representations and not provide an exhaustive or optimal matrix creation process.

The document-term matrix created using the process above contains the count of term j appearing in document i as the entry at index (i, j).

2.2. Document and Term Weighting using Pointwise Mutual Information

Using the raw terms counts per document across many text analysis applications has been shown to introduce biases based on document length, novel term usage, and common term usage [4, 18, 19, 25]. Weighting of the document-term matrix entries often leads to improved analyses. Specifically, the weighting scheme based on term frequency and inverse document frequency (TF-IDF) is by far one of the most popular forms of weighting document-term matrices [24].

More recently, Chew *et al.* demonstrated that the use of pointwise mutual information (PMI) for document-term matrix weighting provides several advantages over TF-IDF and other weighting schemes for retrieval tasks [6]. Using PMI, the weighted entry of the document-term matrix is given by

$$w(i,j) = \log_2\left(\frac{p(i,j)}{p(i) \times p(j)}\right) , \tag{1}$$

where $p(j)$ is the probability that a randomly selected term from the corpus is term i, $p(i)$ is the probability that a randomly selected term from the corpus comes from document i, and $p(i,j)$ is the joint probability that a randomly selected term from the corpus is term j and that it came from document j. These probabilities are estimated using the term and document counts provided by the `CountVectorizer` class described in Section 2.1.

2.3. Topic Modeling using Latent Semantic Analysis

Topic models of documents may help to address several challenges associated with using raw term vectors, even weighted as above using PMI, when analyzing relationships between the contents of documents across a corpus. Specifically, topic models define a set of features, called *topics*, that represent combinations of terms to model latent semantic relationships between sets of terms as they occur within and across the documents. The number of topics is often chosen to be much less than the number of terms and documents in the corpus, leading to a reduction in the dimensionality of the document-term matrix.

We use the Latent Semantic Analysis (LSA) [14, 15] to model topics in this work. LSA computes a truncated singular value decomposition of the weighted document-term matrix [4]. Specifically, for an $m \times n$ document-term matrix A, the truncated SVD with t topics is defined as

$$A_t = U_t \Sigma_t V_t^T , \tag{2}$$

where $U_t \in \mathbb{R}^{m \times t}$, $V_t \in \mathbb{R}^{n \times t}$, and $\Sigma_t \in \mathbb{R}^{t \times t}$ is a diagonal matrix. The diagonal elements of Σ_t are scalar weights (called the singular values of A_t) and represent the contributions of each topic to the overall semantic content of the corpus.

In previous work, we have demonstrated that the LSA topic model has helped improve document analysis across many applications, including retrieval [7, 20], clustering [12, 13], classification [3], summarization [9, 17, 19], multilingual analysis [1, 2, 6], and visualization [10].

Computation of the truncated SVD in our implementation is performed using the `svds` method of the `scipy.sparse.linalg` Python package. The choice of t, i.e., the number of topics, is left to the analyst.

2.4. Presenting Topic Summaries to Analysts for Choosing Documents of Interest

Following our recent work on document clustering [5], we summarize the topic models by displaying $\tilde{n} < n$ terms and $\tilde{m} < m$ documents for each of the t topics. Topics are ordered by the singular value associated with each topic in decreasing order, thus displaying the topics contributing to the corpus contents the most at the top of the list (see Section 2.3). The ordering of the terms and documents within each topic are also presenting in decreasing order of the values from the column vectors of V_t and U_t, respectively, thus displaying the most important terms and documents associated with each topic.

An example of this topic model summary display for the Twitter BBC Health dataset from the publicly available UCI Machine Learning Repository[1] for top 3 topics (with $\tilde{n} = 10$ and $\tilde{m} = 3$) is as follows:

```
Topic 0 (102.495): video (0.327) to (0.286) ebola (0.261) for (0.236) nhs (0.235)
in (0.224) the (0.180) of (0.176) health (0.169) cancer (0.156)

    Doc 1445 (0.041): VIDEO: NHS staff to help in Ebola areas
    Doc 966 (0.037): VIDEO: NHS staff set off to help fight Ebola
    Doc 2627 (0.037): VIDEO: The effect of floods on mental health


Topic 1 (76.529): health (0.595) mental (0.462) child (0.143) care (0.116) services (0.082)
cuts (0.077) warning (0.077) nhs (0.069) needs (0.051) priority (0.042)

    Doc 1107 (0.092): Child mental health services 'unfit'
    Doc 279 (0.088): NHS child mental health care pledge
    Doc 115 (0.087): VIDEO: Child mental health services 'broken'

Topic 2 (74.147): ebola (0.522) health (0.360) mental (0.283) uk (0.186) in (0.145)
leone (0.101) vaccine (0.087) sierra (0.083) liberia (0.071) child (0.066)

    Doc 153 (0.064): UK military health worker has Ebola
    Doc 2846 (0.061): UK troops' mental health 'resilient'
    Doc 436 (0.060): UK health worker monitored for Ebola
```

The values in parentheses in the display for topic t following “`Topic` t”, each term, and each document are singular value t, values from the column t of V_t, and values from column t of U_t, respectively. From this display, analysts are asked to identify one or more documents that are of

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/00438/Health-News-Tweets.zip

interest and will serve as the set of documents for which all document similarities will be computed. Note that documents from different topics can be chosen to form this set.

2.5. Document Similarity Graphs

To analyze the relationships between documents, we construct a similarity graph where vertices represent documents and edges represent the semantic similarities between documents [26]. Such graphs have proven useful in many document clustering [5, 33] and topic model visualization [10, 11, 12, 13] applications.

The weight on the edge between vertices (i.e., documents) i and j using the topic model represented by A_t is computed as follows:

$$e_{ij}(t) = \frac{\langle U_t^i \Sigma_t, U_t^j \Sigma_t \rangle}{\|U_t^i \Sigma_t\|_2 \, \|U_t^j \Sigma_t\|_2} \,, \tag{3}$$

where U_t^k is column k of U_t, $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors, and $\| \cdot \|_2$ denotes the L^2 (or Euclidean) norm of a vector. This amounts to the Pearson correlation (or cosine similarity) between the documents using the topic model vectors. Thus, these weights take on values in the range $[-1, 1]$, with the sign corresponding to negative or positive correlation, and the magnitude corresponding to the strength of the correlation between the documents.

In this work, we limit the scope of our analysis to positive correlations only; hence all negative edge weights are replaced by weights of 0. We further allow for analyst-determined thresholding of the weights to create ε*-neighborhood graphs* [33], which contain only the most significant document relationships.

2.6. Document Similarities using Graph Mean Hitting Times

Using the analyst-identified set of documents of interest from Section 2.4 and the document similarity graph from Section 2.5, we can now compute the relationships of interest to the analyst—i.e., the relationships between a set of documents of interest and all other documents in the corpus. To achieve this, we will compute the *mean hitting times* between the set of vertices associated with the documents of interest and all other vertices. Specifically, we leverage the HITMIX method of computing hitting time moments [22], originally developed to solve seed-set expansion problems on graphs, to compute the mean hitting times. HITMIX includes a highly-scalable, linear algebraic approach to computing mean hitting times between a vertex set (called the *hitting set*) and all other vertices, avoiding costly computations based on random sampling and simulation. In the document analysis context here, the mean hitting times represent the relationships between the documents of interest (i.e., the hitting set) and all other documents, with smaller values denoting stronger semantic relationships.

The original HITMIX method assumes a fully connected graph, which is often not the case for document similarity graphs created using the approach defined in Section 2.5, especially when using thresholding. Thus, our implementation of computing hitting time moments includes an initial step of finding the connected components of the graph that contain the vertices associated with the documents of interest. We use the `connected_components` method of the `scipy.sparse.csgraph` Python package for this computation, which implements the efficient, scalable method by Pearce [28].

Our implementation of the HITMIX mean hitting times can be found at https://github.com/sandialabs/hitmix. The output of this implementation is a vector of size *m* (i.e., the number of documents in the corpus) whose values are either 0 (indicating the document is in the hitting set), ∞ (indicating that there is no path in the document similarity graph that connects the vertex to the hitting set) or the mean hitting time of the vertex to the hitting set. These values are used to provide a rank-ordered list of documents as they related to the documents of interest, sorted by semantic similarity (as approximated by the mean hitting time values).

3. CASE STUDY

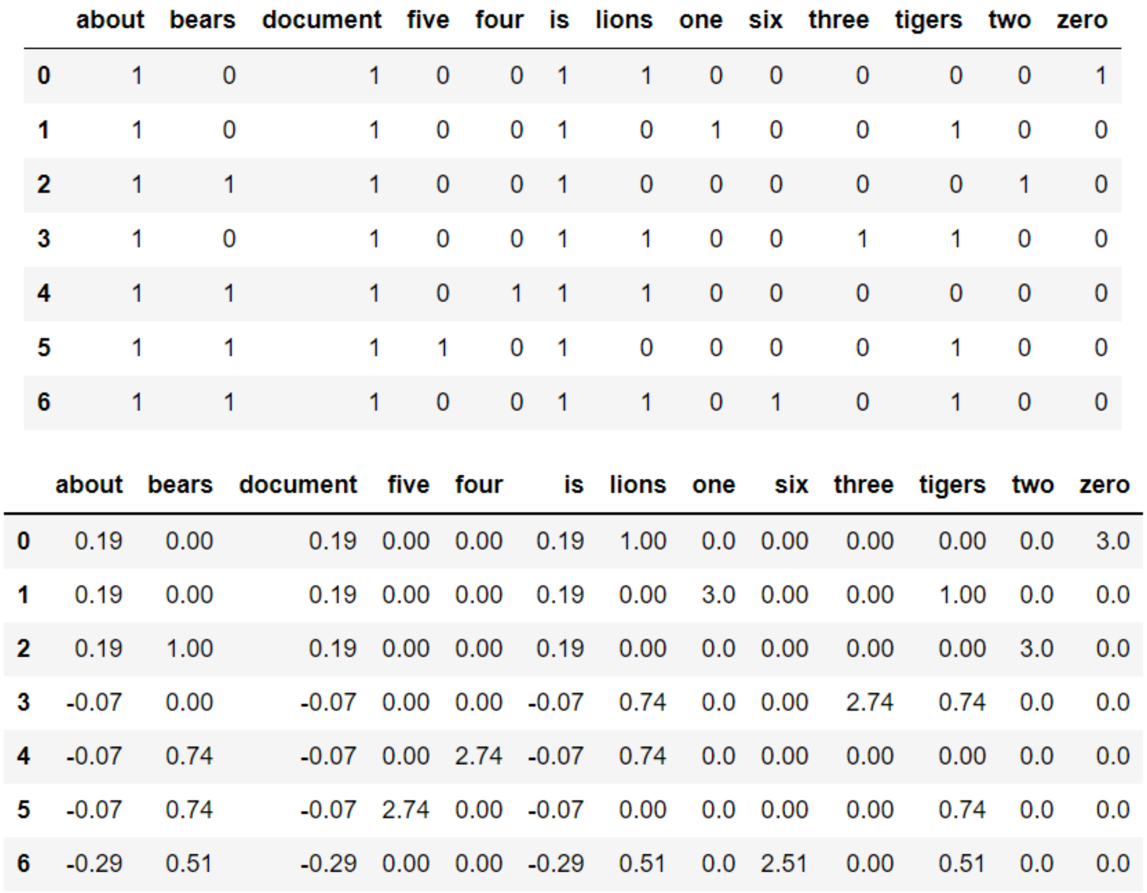
In this section, we present a case study describing the application of our information retrieval approach using a small number of documents that are very simple in terms of the size of the vocabulary used across the documents and the size of individual documents. Due to the simple structure of these documents, it is easy to see similarities and differences across the corpus, and illustration of each of the analysis steps described in Section 2 is also easy to follow. The full set of documents is presented in Table 3-1.

ID	Text
0	Document zero is about lions.
1	Document one is about tigers.
2	Document two is about bears.
3	Document three is about lions, tigers.
4	Document four is about lions, bears.
5	Document five is about tigers, bears.
6	Document six is about lions, tigers, bears.

Table 3-1 The complete set of simple documents used in the case study.

Across the documents, we see that the documents contain some common terms (i.e., *document*, *about*, and *is*), unique terms (i.e., document numbers such as *one*, *two*, etc.), and different combinations of shared terms (i.e., *lions*, *tigers*, and *bears*). We provide detailed results of the steps of the methods from Section 2 to illustrate how the relational structures associated with these different terms are identified, used, and captured in our analyses.

Figure 3-1 presents the document-term matrix of raw term counts in the top portion of the figure (Section 2.1) and PMI-weighted term values (Section 2.2) in the bottom portion of the term values. Note that the common terms all have the same PMI distributions across documents and



	about	bears	document	five	four	is	lions	one	six	three	tigers	two	zero
0	1	0	1	0	0	1	1	0	0	0	0	0	1
1	1	0	1	0	0	1	0	1	0	0	1	0	0
2	1	1	1	0	0	1	0	0	0	0	0	1	0
3	1	0	1	0	0	1	1	0	0	1	1	0	0
4	1	1	1	0	1	1	1	0	0	0	0	0	0
5	1	1	1	1	0	1	0	0	0	0	1	0	0
6	1	1	1	0	0	1	1	0	1	0	1	0	0

	about	bears	document	five	four	is	lions	one	six	three	tigers	two	zero
0	0.19	0.00	0.19	0.00	0.00	0.19	1.00	0.0	0.00	0.00	0.00	0.0	3.0
1	0.19	0.00	0.19	0.00	0.00	0.19	0.00	3.0	0.00	0.00	1.00	0.0	0.0
2	0.19	1.00	0.19	0.00	0.00	0.19	0.00	0.0	0.00	0.00	0.00	3.0	0.0
3	-0.07	0.00	-0.07	0.00	0.00	-0.07	0.74	0.0	0.00	2.74	0.74	0.0	0.0
4	-0.07	0.74	-0.07	0.00	2.74	-0.07	0.74	0.0	0.00	0.00	0.00	0.0	0.0
5	-0.07	0.74	-0.07	2.74	0.00	-0.07	0.00	0.0	0.00	0.00	0.74	0.0	0.0
6	-0.29	0.51	-0.29	0.00	0.00	-0.29	0.51	0.0	2.51	0.00	0.51	0.0	0.0

Figure 3-1 Document-term matrix of raw term counts (top) and PMI-weighted term values (bottom) per document used in the case study.

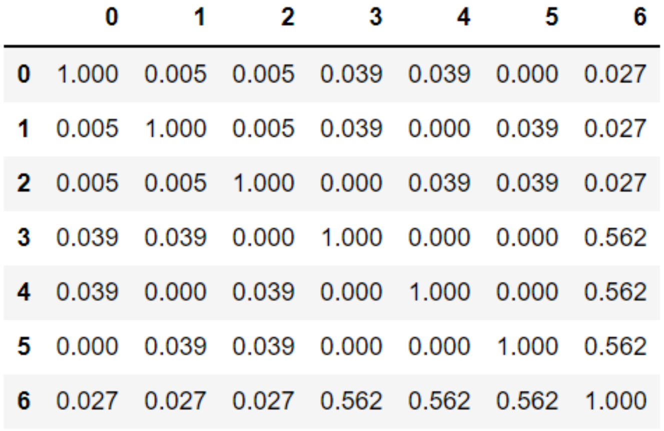
the values are either low or negative, indicating they are not very useful in distinguishing the relationship among the documents. The unique terms have high PMI values, as is expected. Finally, the shared terms also have relatively high PMI values, indicating their importance across documents.

For this small corpus we compute an LSA topic model (Section 2.3) using $t = 6$ topics. In many applications where the document corpus is much larger, we recommend using fewer topics than the maximum that can be computed, but for this example, we compute all available topics that can be computed using the `scipy.sparse.linalg.svds` method (which is one fewer than the minimum of the number of documents and number of terms). Using this topic model, we present the topic summaries (Section 2.4) below:

```
Topic 0 (3.440): zero (0.353) two (0.353) one (0.353) tigers (0.322) lions (0.322)
bears (0.322) four (0.297) three (0.297) five (0.297) six (0.219)

    Doc 2 (0.405) Document two is about bears.
    Doc 0 (0.405) Document zero is about lions.
    Doc 1 (0.405) Document one is about tigers.

Topic 1 (3.201): one (0.691) tigers (0.287) three (0.164) five (0.046) six (0.000)
is (-0.000) document (-0.000) about (-0.000) lions (-0.063) zero (-0.151)

    Doc 1 (0.737) Document one is about tigers.
    Doc 3 (0.192) Document three is about lions, tigers.
    Doc 5 (0.054) Document five is about tigers, bears.

Topic 2 (3.201): two (0.486) one (0.224) five (0.216) bears (0.202) tigers (0.093)
is (0.000) document (0.000) about (0.000) six (-0.000) four (-0.068)

    Doc 2 (0.519) Document two is about bears.
    Doc 5 (0.252) Document five is about tigers, bears.
    Doc 1 (0.239) Document one is about tigers.

Topic 3 (2.980): two (0.414) one (0.414) zero (0.414) is (0.136) document (0.136)
about (0.136) lions (-0.095) bears (-0.095) tigers (-0.095) six (-0.262)

    Doc 2 (0.411) Document two is about bears.
    Doc 1 (0.411) Document one is about tigers.
    Doc 0 (0.411) Document zero is about lions.

Topic 4 (2.791): three (0.750) two (0.274) tigers (0.071) lions (0.040) six (0.000)
is (-0.000) document (-0.000) about (-0.000) zero (-0.098) bears (-0.111)

    Doc 3 (0.765) Document three is about lions, tigers.
    Doc 2 (0.255) Document two is about bears.
    Doc 6 (0.000) Document six is about lions, tigers, bears.

Topic 5 (2.791): four (0.589) one (0.215) three (0.122) lions (0.105) two (0.044)
is (-0.000) document (-0.000) about (-0.000) six (-0.000) bears (-0.018)

    Doc 4 (0.600) Document four is about lions, bears.
    Doc 1 (0.200) Document one is about tigers.
    Doc 3 (0.124) Document three is about lions, tigers.
```

After some analysis, we see that the topics relate most strongly to documents about 0) individual animals, 1) tigers, 2) bears and tigers, 3) common terms in the shorter documents, 4) tigers and lions, and 5) lions. For this case study, we choose the *documents of interest* to be documents 0 and 1—i.e., a document about lions and a document about tigers. The goal is to find other documents that are related to both of these documents by treating them as a set.

Figure 3-2 presents the values in the document similarity graph (Section 2.5). No thresholding was applied in this use case. However, since we use only positive weights in the document similarity graphs (see Section 2.5), the values of 0 correspond to negative weights (i.e., negatively correlated document relationships).



	0	1	2	3	4	5	6
0	1.000	0.005	0.005	0.039	0.039	0.000	0.027
1	0.005	1.000	0.005	0.039	0.000	0.039	0.027
2	0.005	0.005	1.000	0.000	0.039	0.039	0.027
3	0.039	0.039	0.000	1.000	0.000	0.000	0.562
4	0.039	0.000	0.039	0.000	1.000	0.000	0.562
5	0.000	0.039	0.039	0.000	0.000	1.000	0.562
6	0.027	0.027	0.027	0.562	0.562	0.562	1.000

Figure 3-2 Adjacency matrix of the document similarity graph used in the case study.

ID	Mean Hitting Time	Text
0	0.00	Document zero is about lions.
1	0.00	Document one is about tigers.
3	38.01	Document three is about lions, tigers.
6	40.39	Document six is about lions, tigers, bears.
4	40.89	Document four is about lions, bears.
5	40.89	Document five is about tigers, bears.
2	47.03	Document two is about bears.

Table 3-2 Mean hitting times to the set of documents 0 and 1 for the corpus in the case study.

Finally, we compute the mean hitting times (Section 2.6) of the similarity graph. Table 3-2 presents the documents ordered by mean hitting times. Recall that since documents 0 and 1 are in the set of documents of interest (i.e., hitting set), the mean hitting times for those documents are 0, as expected. The document closest to the documents of interest, with respect to mean hitting times, is document 3 about lions *and* tigers but not bears. Then, documents 6 is about all three animals; thus it is about lions *and* tigers but also bears. The next two documents, 4 and 5, are about lions *or* tigers and also bears. And then finally, the farthest document from the hitting set in terms of mean hitting time is document 2, which is not about lions or tigers and only about bears. This seems like a reasonable ordering of the documents if an analysts were searching for documents about lions *and* tigers, as indicated by the documents in the hitting set. If the analysts were to search for documents about lions and tigers separately through two queries of the corpus using traditional information retrieval methods, that would results in two ordered list that would need to be manually merged into a single ordered list. Using our approach described here, a single ordered list is returned that accounts for the relationships of documents to a set of documents, not to a set of terms or a single document.

4. CONCLUSION

We have presented an alternative approach to information retrieval based on document-document relationships rather than the traditional approach of using term-document relationships. We defined the steps of our approach that leverages standard natural language processing and topic modeling to construct a document similarity graph of the semantic relationships between documents. The main contribution of this work was the introduction of using graph mean hitting times between a set of documents of interest and all other documents in a corpus to provide an ordered list of documents that are most related to the set. We illustrated the application of this approach to information retrieval on a simple, synthetically generated small corpus, demonstrating that identifying a set of documents—rather than terms—is a viable approach to document retrieval.

Since the application of the HITMIX for text analysis and information retrieval presented here is a novel use of graph mean hitting times, future work will also include application to a variety of document corpora from multiple topical domains to better assess the effectiveness of this approach more generally.

REFERENCES

[1] Brett W. Bader and Peter A. Chew. Enhancing multilingual Latent Semantic Analysis with term alignment information. In *Proceedings of the 22nd International Conference on Computational Linguistics*, volume 1, pages 49–56, 01 2008. 10, 12

[2] Brett W. Bader and Peter A. Chew. *Algebraic Techniques for Multilingual Document Clustering*, chapter 2, pages 21–36. John Wiley & Sons, Ltd, 2010. 10, 12

[3] Justin D. Basilico, Daniel M. Dunlavy, Stephen J. Verzi, Travis L. Bauer, and Wendy Shaneyfelt. Yucca Mountain Licensing Support Network Archive Assistant. Technical report, Sandia National Laboratories, 2008. SAND2008-1622. 10, 12

[4] Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995. 9, 11

[5] Jonathan Bisila, Daniel M. Dunlavy, Zoe N. Gastelum, and Craig D. Ulmer. Topic modeling with natural language processing for identification of nuclear proliferation-relevant scientific and technical publications. In *Proceedings of the INMM Annual Meeting*, 2020. 10, 12, 13

[6] Peter A. Chew, Brett W. Bader, Stephen Helmreich, Ahmed Abdelali, and Stephen J. Verzi. An information-theoretic, vector-space-model approach to cross-language information retrieval. *Natural Language Engineering*, 17(1):37–70, 2011. 10, 11, 12

[7] Peter A. Chew, Brett W. Bader, Tamara G. Kolda, and Ahmed Abdelali. Cross-language information retrieval using PARAFAC2. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 143–152, 01 2007. 10, 11

[8] Peter A. Chew, Brett W. Bader, and Alla Rozovskaya. Using DEDICOM for completely unsupervised part-of-speech tagging. In *Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, UMSLLS '09, pages 54–62, USA, 2009. Association for Computational Linguistics. 10

[9] John M. Conroy, Daniel M. Dunlavy, and Dianne P. O'Leary. From TREC to DUC to TREC again. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of The Twelfth Text REtrieval Conference, TREC*, pages 293–302. National Institute of Standards and Technology (NIST), 2004. 10, 12

[10] Patricia J. Crossno, Daniel M. Dunlavy, and Timothy M. Shead. LSAView: A tool for visual exploration of latent semantic modeling. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pages 83–90, 2009. 10, 12, 13

[11] Patricia J. Crossno, Andrew T. Wilson, Daniel M. Dunlavy, and Timothy M. Shead. TopicView: Understanding document relationships using Latent Dirichlet Allocation models. In *Proceedings of the IEEE Workshop on Interactive Visual Text Analytics for Decision Making*, October 2011. 10, 13

[12] Patricia J. Crossno, Andrew T. Wilson, Timothy M. Shead, Warren L. Davis, and Daniel M. Dunlavy. TOPICVIEW: Visual analysis of topic models and their impact on document clustering. *International Journal on Artificial Intelligence Tools*, 22(05):1360008, 2013. 10, 11, 13

[13] Patricia J. Crossno, Andrew T. Wilson, Timothy M. Shead, and Daniel M. Dunlavy. TopicView: Visually comparing topic models of text collections. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 936–943, 2011. 10, 11, 13

[14] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. 11

[15] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using Latent Semantic Analysis to improve access to textual information. In *CHI '88: Proc. SIGCHI Conference on Fuman Factors in Computing Systems*, pages 281–285, 1988. 11

[16] Daniel M. Dunlavy and Peter A. Chew. Constrained versions of DEDICOM for use in unsupervised part-of-speech tagging. Technical report, Sandia National Laboratories, 2016. SAND2016-4520. 10

[17] Daniel M. Dunlavy, John M. Conroy, Judith D. Schlesinger, Sarah A. Goodman, Mary E. Okurowski, Dianne P. O'Leary, and Hans van Halteren. Performance of a three-stage system for multi-document summarization. In *Proceedings of the Document Understanding Conference (DUC)*. National Institute of Standards and Technology (NIST), 2003. 10, 12

[18] Daniel M. Dunlavy, Dianne P. O'Leary, John M. Conroy, and Judith D. Schlesinger. QCS: A system for querying, clustering and summarizing documents. Technical report, Sandia National Laboratories, 2006. SAND2006-5000. 11

[19] Daniel M. Dunlavy, Dianne P. O'Leary, John M. Conroy, and Judith D. Schlesinger. QCS: A system for querying, clustering and summarizing documents. *Information Processing & Management*, 43(6):1588–1605, 2007. 10, 11, 12

[20] Daniel M. Dunlavy, Timothy M. Shead, and Eric T. Stanton. ParaText: Scalable text modeling and analysis. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 344–347, New York, NY, USA, 2010. Association for Computing Machinery. 10, 11

[21] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998. 10

[22] Alexander H. Foss, Richard B. Lehoucq, W. Zachary Stuart, J. Derek Tucker, and Jonathan W. Berry. A deterministic hitting-time moment approach to seed-set expansion over a graph. arXiv:2011.09544, 2020. 13

[23] Mamta Gupta and Anand Rajavat. Comparison of algorithms for document clustering. In *2014 International Conference on Computational Intelligence and Communication Networks*, pages 541–545, 2014. 9

[24] Karen Sparck Jones. A statistical interpretation of term specificity and its applications in retrieval. *Journal of Documentation*, 28(1):11–21, 1972. 11

[25] Tamara G. Kolda and Dianne P. O'Leary. A semidiscrete matrix decomposition for latent semantic indexing information retrieval. *ACM Transactions on Information Systems*, 16(4):322–346, October 1998. 11

[26] Thomas K. Landauer, Darrell Laham, and Marcia Derr. From paragraph to graph: Latent semantic analysis for information visualization. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5214–5219, 2004. 13

[27] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008. 9

[28] David J. Pearce. A space-efficient algorithm for finding strongly connected components. *Information Processing Letters*, 116(1):47–52, 2016. 14

[29] Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980. 10

[30] Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. 9, 10

[31] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000. 9

[32] Taylor P. Turpen and Daniel M. Dunlavy. Semisupervised named entity recognition. Technical report, Sandia National Laboratories, 2009. SAND2010-3083P. 10

[33] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. 13

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Technical Library	1911	sanddocs@sandia.gov

Sandia
National
Laboratories