A Topological Collapse for Document Summarization

Hui Guan*, Wen Tang*, Hamid Krim*, James Keiser†, Andrew Rindos‡ and Radmila Sazdanovic§
*Department of Electrical and Computer Engineering
§Department of Mathematics
North Carolina State University, Raleigh, NC 27606
Email: {hguan2, wtang6, ahk}@ncsu.edu, rsazdanovic@math.ncsu.edu
† Laboratory for Analytic Sciences, Raleigh, 27606, Email: jmkeiser@ncsu.edu
‡IBM, Durham, NC 27703, Email: rindos@us.ibm.com

***Abstract*—As a useful tool to summarize documents, keyphrase extraction extracts a set of single or multiple words, called keyphrases, that capture the primary topics discussed in a document. In this paper we propose DoCollapse, a topological collapse-based unsupervised keyphrase extraction method that relies on networking document by semantic relatedness of candidate keyphrases. A semantic graph is built with candidates keyphrases as vertices and then reduced to its core using topological collapse algorithm to facilitate final keyphrase selection. Iteratively collapsing dominated vertices aids in removing noisy candidates and revealing important points. We conducted experiments on two standard evaluation datasets composed of scientific papers and found that DoCollapse outperforms state-of-the-art methods. Results show that simplifying a document graph by homology-preserving topological collapse benefits keyphrase extraction.**

I. Introduction

Big data raises the problem of dealing with data explosion, especially huge amount of natural language text data for efficient data analysis. Document summarization aims at reducing a text document to a concise representation, such as keyphrases or sentences, that retains the most important points of the original document for faster document analysis. As a useful tool to summarize documents, keyphrase extraction extracts a set of single or multiple words, called keyphrases, that capture the primary topics discussed in a document. Keyphrase Extraction is a basic step for various natural language processing problems such as document clustering, document classification and information retrieval [1]. Therefore, automatically assigning keyphrases to documents is an active and wide research field.

In this paper we propose DoCollapse a topological collapse-based unsupervised keyphrase extraction method that relies on a semantic graph representation of the document. A semantic network is built with the keyphrase candidates extracted from an input document and then reduced to its core using topological collapse algorithm to facilitate final keyphrase selection. Though there are a few studies on unsupervised methods, especially graph-based ranking [2] and topic-based clustering approaches [3] to extract keyphrases, neither of them consider the topological properties of a document network.

978-1-5090-1749-2/16/$31.00 ©2016 IEEE

The key step of DoCollapse is a distributed topological collapse algorithm based on a vertex dominance criterion [4]. A vertex v is dominated by vertex w if all vertices that share an edge with v also share an edge with vertex w. We assume that in a document semantic graph, if one candidate keyphrase dominates another one, then the dominating candidate should convey more important information and thus, is more likely to be a keyphrase. Iteratively collapsing dominated vertices aids in removing noisy candidates and revealing important points. Specifically, removing dominated vertices does not change the homology of the graph [5]. We will show that simplifying a document complex by preserving topological structures/homology benefits keyphrase extraction.

II. Related Work

State-of-the-art approaches for keyphrase extraction contain mainly two steps: candidate keyphrases extraction using heuristics and keyphrases ranking based on statistical learning that requires manually created training set. Recent years have seen an increase in the popularity of unsupervised keyphrase extraction approaches because they don't require training data and are generally domain dependent.

We focus on state-of-the-art graph-based keyphrase extraction approaches. These approaches rely on the assumption that a candidate keyphrase is important if it is related to a large number of candidates and also these related candidates are important. In TextRank [2], a document is represented as a graph in which vertices are words and edges represent co-occurrence relations. The importance of each vertex is computed using the PageRank algorithm [6], commonly used to rank web pages. The top-ranked vertices are assembled to phrases as their assigned keyphrases. TopicRank [7] is a recent improvement of TextRank. TopicRank proposes to represent a document as a complete graph where vertices are topics and each topic is a cluster of phrases that have certain percentage of overlapping words.

A related novel approach is Topological Data Analysis (TDA). TDA applies topology to study coarse geometric features of data by providing a global description of a dataset with significant structural information. Little work has been done on application of topological tools on text and documents. Zhu et al. [8] introduced a new document representation that captures the semantic "tie-backs" in a text document. In [9], the authors demonstrated that topic detection with a topological representations of twitter data generated by Mapper algorithm [10], can obtain competitive results compared to state-of-the-art topic detection methods. The structures of document collections are studied using persistent homology in [11]. Chiang [12] structured the semantic space of high dimensional data into a simplicial complex that is effective on document clustering. To the best of our knowledge, our work is the first to study the topological structure of a single document for keyphrase extraction.

III. DoCollapse

DoCollapse is our proposed unsupervised approach to extract keyphrases that represent the most important topics in the document. The method takes advantage of a distributed topological collapse algorithm introduced by Wilkerson et al. [5] to reveal core structure of a single document. We conjecture that the key information of a document is preserved in the core structure after collapse and the duplicated or non-important information is removed to reduce noise. The flowchart for extracting keyphrases is shown in Fig. 1. Given a single document, several preprocessing steps including tokenization and Part-Of-Speech tagging are applied and candidate phrases are extracted with syntactic constraints. Then, a similarity graph is built with the set of candidate keyphrases and collapses to facilitate candidate phrase selection.

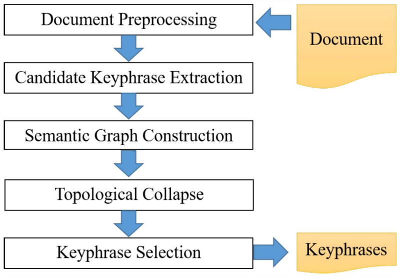

Fig. 1. Processing steps of DoCollapse

A. Candidate Keyphrase Extraction

Candidate keyphrase extraction identifies a single word or multiple consecutive words from a document to build a candidate pool for keyphrases selection. A single document is a sequence of lexical units or tokens, $d = (t_1, t_2, \cdots, t_m)$. In our case, a token is a word. Since the same word may occur in different position of a document, different tokens can be the same word, and denoted as $t_i = t_j, i \neq j$.

A candidate keyphrase is defined as a single word or multiple consecutive words that meet certain syntactic constraints. The contraints require that words in a phrase have to be the longest consecutive sequence of nouns or nouns joined by a preposition. Adjectives before nouns will also be included in a candidate keyphrase. Let two candidates composed of l and q words be $p_i = (t_i, t_{i+1}, \cdots, t_{i+l-1})$ and $p_j = (t_j, t_{j+1}, \cdots, t_{j+q-1})$. We define two candidates $p_i, p_j, i \neq j$ as identical if they have the same length, as well as the same sequence of tokens:

$$p_i = p_j \iff l = q \text{ and } t_j = t_i, \cdots, t_{j+l-1} = t_{i+q-1}.$$

Candidate keyphrase extraction maps a sequence of tokens to a sequence of candidate keyphrases: $\tilde{d} = (p_1, p_2, \cdots, p_{\tilde{n}})$. Note that the sequence might have identical candidates. So a set of candidate keyphrases from document d is denoted as $P_d = \{p_1, p_2, \cdots, p_n\}, p_i \neq p_j$ if $i \neq j, \forall p_i, p_j \in P_d$.

B. Semantic Graph Construction

Let $G = (V, E)$ be a semantic graph of document d where V is the set of candidate keyphrases P_d. The edge between two candidate keyphrases depends on their semantic relatedness, indicating whether they are associated with each other. Researchers have computed semantic relatedness using external sources such as WordNet [13] and Wikipedia [3][1].

We take an easier and stricter approach [7]: two candidate keyphrases are related if they have an overlap ratio over a specific threshold thr. This approach requires that two candidates have to share some common words. Similar to Jaccard similarity coefficient, our candidate keyphrase *Overlap Ratio (OR)* is defined as:

$$OR(p_i, p_j) = \frac{|\{t_i, \cdots, t_{i+l-1}\} \cap \{t_j, \cdots, t_{j+q-1}\}|}{|\{t_i, \cdots, t_{i+l-1}\} \cup \{t_j, \cdots, t_{j+q-1}\}|}. \quad (1)$$

Then the set of edges will be $E = \{(p_i, p_j) | OR(p_i, p_j) \geq thr, \forall p_i, p_j \in P_d, p_i \neq p_j\}$.

C. Topological Collapse

Topological Collapse is the key step of DoCollapse for removing noisy candidate keyphrases while preserving key information of a document. It takes advantage of the strong collapse algorithm [5] to reduce a semantic graph of a document to its core.

Since topological collapse is for flag complexes, we introduce the concept of a flag complex, a higher dimensional generalization of a graph. Given a graph $G = (V, E)$, a K-simplex σ is the convex hull of $K + 1$ affinely independent points $\sigma = conv\{v_0, v_1, \cdots, v_K\}$ and $dim(\sigma) = K$. For example, a 2-simplex is a filled-in triangle; a 3-simplex is a solid tetrahedron. A flag complex contains a simplex σ whenever all pairs of vertices in σ are connected by an edge in G. Mathematically, a flag complex is defined as:

$$X(G) = \{\sigma =(v_{i_\bullet}, v_{i_1}, \cdots, v_{i_{\bullet dim(\sigma)}}) | (v_{i_j}, v_{i_r}) \in E,$$
$$\text{for all } 0 \leq j, r \leq dim(\sigma)\}.$$

A flag complex can be uniquely built based on the semantic graph of a document. 2-simplices are added first based on triangles formed by three edges, 3-simplices are added based on tetrahedrons formed by four 2-simplices and then followed by higher dimensional simplices. Simplices in the flag complex mean common words shared by candidate keyphrases. Let's define the neighbors of a vertex by $\mathcal{N}(v) = \{u \in V | (u, v) \in E\} \cup \{v\}$. A vertex v is dominated by its neighbor vertex

w if and only if $\mathcal{N}(v) \subseteq \mathcal{N}(w)$. The topological collapse algorithm iteratively collapses dominated vertices information to the dominating one, simplifying the flag complex by eliminating redundant parts that do not affect topological structure. The topological collapse algorithm to get the core graph $G_C = (V_C, E_C)$ is shown as Algorithm 1.

Algorithm 1 Topological Collapse Algorithm

```
$\forall v \in V, \text{label}(v) \leftarrow p_v$
$V_C \leftarrow V, E_C \leftarrow E$
while True do
    del $\leftarrow \emptyset$
    for $v \in V_c$ do
        for $u \in \mathcal{N}(v)$ do
            if $\mathcal{N}(u) \subseteq \mathcal{N}(v)$ then
                del $\leftarrow u$
                label$(v) \leftarrow$ label$(u)$
    if del is $\emptyset$ then
        Break
    else
        $E_C \leftarrow \{(u,v) | (u,v) \in E_C, u, v \notin \text{del}\}$
```

Each vertex is labeled with one candidate keyphrase in the original semantic graph G while also being labeled with candidate keyphrases from dominated vertices in the core graph G_C. Through the collapse procedure, related information distributed to several vertices in G is combined together to represent information of the remaining vertex in G_C. The core structure of a document represents its key information with each vertex being a topologically non-removable "topic". Fig. 2 shows a toy example of topological collapse. The information in dominated vertices flows to the dominating vertex; those dominating vertices form the core of a document.

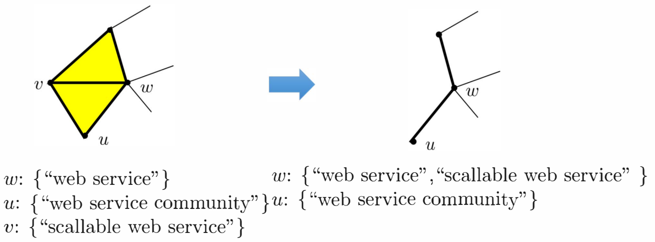
v
w
u
w
u
w: {"web service"}
u: {"web service community"}
v: {"scallable web service"}
w: {"web service","scallable web service" }
u: {"web service community"}

Fig. 2. A toy example of topological collapse. Vertices v, u are dominated by vertex w. When v is removed, the information of v flowed to vertex w. After u is collapsed, the labels of w will be {"web service", "web service community", "scalable web service"}.

D. *Keyphrase Selection*

Keyphrase Selection is the last step in assigning keyphrases to a document. Only the most representative k candidate keyphrases are selected from the core graph G_C. We utilize TF-IDF (Term Freqeuncy-Inverse Document Frequency) weights of candidate keyphrases to compute vertex score and select out candidates from the top k ranked vertices as keyphrases. The TF-IDF weight of keyphrase candidate p from document d in corpus D is calculated by [14]:

$$\text{TF-IDF}(p, d|D) = tf(p, d) * \log_2(\frac{N}{N_p}), \tag{2}$$

where $tf(p, d)$ is frequency of $p \in P_d$ in document d, N is the number of documents in corpus D and N_p is the number of documents that contain candidate p. For simplicity, we use TF-IDF(p).

Given a vertex $v \in V_C$ and its candidate keyphrase labels label(v), the vertex score is calculated:

$$S(v) = \sum_{p_i \in \text{label}(v)} \mathbb{1}(p_v \preceq p_i) * \text{TF-IDF}(p_i), \tag{3}$$

where $p_i \preceq p_j, p_i = (t_i, t_{i+1}, \cdots, t_{i+l-1}), p_j = (t_j, t_{j+1}, \cdots, t_{j+q-1})$ if and only if $\{t_i, t_{i+1}, \cdots, t_{i+l-1}\} \subseteq \{t_j, t_{j+1}, \cdots, t_{j+q-1}\}$. Then assigned keyphrases are a set of labels for top ranked vertices V_k: $\{p_v | v \in V_k\}$.

IV. Experimental Evaluation

A. *Datasets*

To compare DoCollapse against other unsupervised methods, we evaluate several methods on two standard evaluation datasets: SemEval-2010 and NUS Corpus. SemEval-2010, built for the keyphrase extraction campaign [15], is composed of 284 scientific articles from ACM Digital Libraries. The dataset is divided into three sets for trial, train and test. The test set contains 100 documents and is used in our experiment. NUS Corpus contains 211 scientific conference papers collected by Nguyen [16]. Only a subset of the corpus consisting of 151 documents contains both author assigned keyphrases and reader assigned keyphrases. This set of 151 doucments is used in our experiments. For both datasets, the reference keyphrases are the combination of author and reader assigned keyphrases.

B. *Preprocessing*

We used python Natural Language ToolKit [17] for preprocessing document and extracting noun phrase chunks as candidate keyphrases. The preprocessing steps contain: 1) sentence segmentation, 2) word tokenization, 3) Part-of-Speech (POS) tagging. POS tagging is for candidate keyphrase extraction for choosing the longest consecutive sequence of nouns and adjectives chunks. Data Statistics are shown in Table I. Tokens, Keys, Candidates and Matches refer to the average number per document. Matches show the number of candidate keyphrases that match the reference keyphrases.

TABLE I
DATA STATISTICS

Dataset	Documents	Tokens	Keys	Candidates	Matches
SemEval-2010	100	9398.6	14.4	841.4	9.59
NUS Corpus	151	8295.1	13.4	809.9	8.87

C. Baselines

We use three baseline methods to compare with our DoCollapse approach: TF-IDF, TextRank [2] and TopicRank [7]. TF-IDF is a strong baseline to extract keyphrases from a document by considering not only a single document, but all documents from a reference corpus. The method orders keyphrase candidates by TF-IDF weight, calculated in Equation 2, and picks the top k candidates with highest weight as the keyphrases for the document. Since the flag complex used to describe an input document for DoCollapse contains a graph structure for that document, it is natural to compare it with the two state-of-the-art graph-based keyphrase extraction methods: TextRank and TopicRank. The two algorithms are described in Section II.

D. Evaluation Measures

The performances of keyphrase extraction methods are evaluated using *exact match* in terms of Matches (M), Precision (P), Recall(R) and F-measure(F) with top 15 keyphrases extracted (k=15), similar to [2][7]. Both extracted and reference keyphrases are stemmed to their root form to reduce the number of mismatches. Let TP be the number of correctly assigned keyphrases, FP be the number of wrongly assigned keyphrases, and FN be the keywords that are missed using using our method. The definitions of M, P, R, F are:

$$M = TP, \qquad P = \frac{TP}{TP + FP},$$

$$R = \frac{TP}{TP + FN}, \qquad F = \frac{2PR}{P + R}.$$

E. Results

The results of DoCollapse and three baselines on the two datasets SemEval-2010 and NUS Corpus are shown in Table II and Table III. The similarity threshold thr is empirically set to be 0.3. Overall, our method outperforms TF-IDF and significantly outperforms TextRank and TopicRank on both datasets. We can see that TF-IDF performs better than TextRank, which is consistent with the work in [7]. TextRank tends to be more sensitive to noisy candidates since it takes each candidates independently. TopicRank aims at ranking topics instead of words to identify the set of keyphrases that cover the main topics of the a document. Intuitively, candidates that represent the main topics of a document are more likely to be keyphrases. Our approach is a further improvement of TopicRank by forming main topics of a document by topological collapse. Though the final keyphrase selection step is based on the same TF-IDF weights as TF-IDF method, DoCollapse outperforms TF-IDF at 0.001 level using Student's t-test at NUS corpus. Note that the criterion for determining dominated vertices needs only local neighborhood information, which makes the topological collapse algorithm easily distributed and computationally efficient.

To observe the effect of similarity threshold thr defined in Section III-B, we evaluated the performance of DoCollapse under different parameter settings for both datasets. Table IV shows the results of DoCollpase on NUS Corpus with similarity threshold thr ranging from 0.15 to 0.5. M_C is the number of matches between reference keyphrases and candidate keyphrases represented by the vertices in the core graph. $|V_C|$ denotes the number of vertices in the collapsed semantic graph. The larger the similarity threshold thr, the smaller the number of vertices in the core graph G_C. This is because the semantic graph G contains fewer edges with a larger thr and thus fewer vertices/candidates need to be kept to preserve topological structures or holes. With increasing thr, more noisy or redundant candidate keyphrases are removed, leaving a smaller size of candidates to perform keyphrase selection. For both dataset, $thr \in (0.25, 0.5)$ gives comparatively good performance. The results of DoCollapse are consistently better than the baseline methods under different parameter settings. More study needs to be done for the generality of the best similarity threshold $thr = 0.3$ on various corpus types.

TABLE II
EVALUATION RESULTS ON SEMEVAL-2010

Methods	M	P	R	F
TF-IDF	2.32	15.47	16.57	15.85
TextRank	1.51	10.07	10.49	10.17
TopicRank	1.87	12.47	13.54	12.87
DoCollapse	**2.52**	**16.8**	**18**	**17.22**

TABLE III
EVALUATION RESULTS ON NUS CORPUS

Methods	M	P	R	F
TF-IDF	2.62	17.44	21.61	18.57
TextRank	1.7	11.35	14.25	12.09
TopicRank	1.92	12.8	16.07	13.66
DoCollapse	**3.23**	**21.51**	**26.13**	**22.64**

TABLE IV
EVALUATION RESULTS WITH DIFFERENT SIMILARITY THRESHOLDS

thr	M	P	R	F	M_C	$\|V_C\|$
0.15	2.58	17.17	19.68	17.6	4.95	313.5
0.2	2.91	19.38	22.94	20.13	5.81	315.7
0.25	3.1	20.66	24.39	21.41	6.64	294.7
0.3	**3.23**	**21.51**	**26.13**	**22.64**	6.68	227.7
0.35	3.08	20.57	24.32	21.36	5.47	122.5
0.4	3.13	20.99	24.86	21.82	5.41	120.1
0.45	2.91	19.52	23.02	20.25	4.77	91.1
0.5	2.91	19.52	23.02	20.25	4.77	91.1

The last question to investigate is: why does topological collapse work? Intuitively, DoCollapse works well because it takes into account the semantic meaning of candidate keyphrases. Researchers found a remarkable feature of the semantic graph: phrases related to the main topics of the document form densely connected communities while non-important phrases are weakly connected or even isolated [3]. Based on this property, our core graph G_C, as a densely connected community, represents key points of a document, while those vertices removed during topological collapse procedure are from weakly connected communities that convey less important information. Gamble et al. [4] also demonstrated that the collapse algorithm based on node dominance criteria could reveal the core structure of social and information networks that is important with respect to network flow and the global structure of the network. Observationally, frequent candidate keyphrases are contained in the core, and core vertices often have high degree, which is consistent with the fact that keyphrases are often frequent in a document. An example semantic graph is shown in Fig. 3 and the corresponding core graph in Fig. 4. Reference keyphrases are labeled and marked with red star in both graphs. More than 70% of vertices are removed while 10 out of 12 keyphrases are still kept in the core.

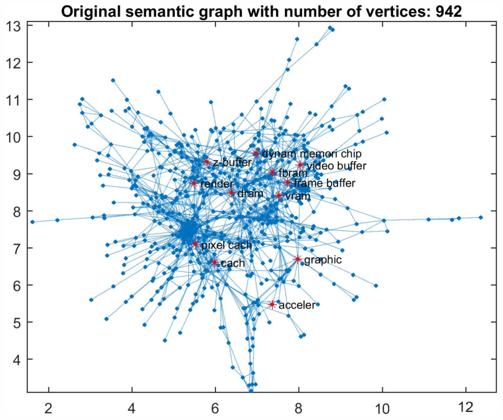

Fig. 3. Main component of a semantic graph for a scientific paper

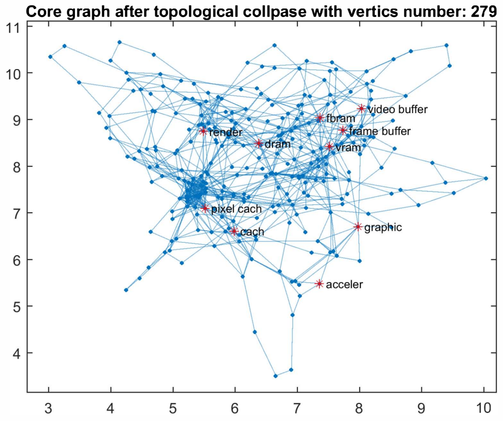

Fig. 4. Core graph after collapse

V. Conclusion

In this paper, we introduced DoCollapse, a topological collapse based unsupervised method for document summarization. The important and novel feature of our method is that it reveals the key information of a document by reducing the document similarity graph to its core while keeping its topological structure. The vertices in the core graph represent the main topics of a doucment and are more likely to be keyphrases than those removed vertices. Topological collapse algorithm is easily distributed and computationally efficient. A topological tool was used in document summarization for the first time, and we showed that it significantly outperforms the state-of-the-art graph-based keyphrase extraction approaches.

Acknowledgment

The authors would like to thank the financial support of Laboratory for Analytic Sciences and IBM. We also thank the anonymous reviewers for insightful comments and suggestions.

References

[1] K. S. Hasan and V. Ng, "Automatic keyphrase extraction: A survey of the state of the art," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 1262–1273.

[2] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2004.

[3] M. Grineva, M. Grinev, and D. Lizorkin, "Extracting key terms from noisy and multitheme documents," in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 661–670.

[4] J. Gamble, H. Chintakunta, A. C. Wilkerson, H. Krim, and A. Swami, "Node dominance: Revealing community and core-periphery structure in social networks," *CoRR*, vol. abs/1509.07435, 2015.

[5] A. C. Wilkerson, T. J. Moore, A. Swami, and H. Krim, "Simplifying the homology of networks via strong collapses," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5258–5262.

[6] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Comput. Netw. ISDN Syst.*, vol. 30, no. 1-7, pp. 107–117, Apr. 1998.

[7] A. Bougouin, F. Boudin, and B. Daille, "Topicrank: Graph-based topic ranking for keyphrase extraction," in *International Joint Conference on Natural Language Processing (IJCNLP)*, 2013, pp. 543–551.

[8] X. Zhu, "Persistent homology: An introduction and a new text representation for natural language processing," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI '13. AAAI Press, 2013, pp. 1953–1959.

[9] P. Torres-Tramón, H. Hromic, and B. R. Heravi, "Topic detection in twitter using topology data analysis," in *Current Trends in Web Engineering*. Springer, 2015, pp. 186–197.

[10] G. Singh, F. Mémoli, and G. E. Carlsson, "Topological methods for the analysis of high dimensional data sets and 3d object recognition," in *Symposium on Point-Based Graphics*, 2007, pp. 91–100.

[11] H. Wagner, P. Dłotko, and M. Mrozek, "Computational topology in text mining," in *Computational Topology in Image Context*. Springer, 2012, pp. 68–78.

[12] I.-J. Chiang, "Discover the semantic topology in high-dimensional data," *Expert Systems with Applications*, vol. 33, no. 1, pp. 256–262, 2007.

[13] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995.

[14] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[15] S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin, "Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, ser. SemEval '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 21–26.

[16] T. D. Nguyen and M.-Y. Kan, "Keyphrase extraction in scientific publications," in *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*. Springer, 2007, pp. 317–326.

[17] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st ed. O'Reilly Media, Inc., 2009.