Advances in Complex Systems
© World Scientific Publishing Company

TOPOLOGY OF DYNAMIC NETWORKS OF KNOWLEDGE OF MICROLEARNING CONTENT: TOPIC MODELING APPROACHES

PAOLO J. SINGH
Department of Applied Mathematics, Naval Postgraduate School,
1 University Circle, Monterey, California, 93943, USA
paolo.singh@nps.edu

RALUCCA GERA
Department of Applied Mathematics, Naval Postgraduate School,
1 University Circle, Monterey, California, 93943, USA
rgera@nps.edu

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)
Published (Day Month Year)

One of the issues faced when generating personalized learning paths (PLPs) is the organization and tagging of microlearning content. We expand upon network science approaches from previous work in Dynamic Network of Knowledge (DNoK) design by incorporating topic modeling methods to automate keyword extraction. Specifically, we compare the BERT language model with fast keyword extraction methods to assign keywords in common using the original DNoK framework. For instructors, institutions' content curators, and designers of topic-specific learning management systems, we formally define a minimum viable network statistic from which DNoK growth should occur. We continue our comparison of topological growth between networks produced from topic modeling and manually curated keyword methods from previous work. Finally, we examine the growth behavior of the DNoK and propose bounds to support the generation of synthetic models for exploring such networks at scale. Ultimately, we propose that a network science-based approach will facilitate personalized, adaptive learning methods that enable instructors and learning engineers to integrate best-practices from learning and cognitive science within this space.

Keywords: complex networks; micro-learning; dynamic network of knowledge; personalized adaptive learning; keyword relationships; science of science.

1. Introduction

The vast availability of online learning content is attractive for many self-paced and hybrid learning students. Ideally, this flexible environment should enable instructors or course designers to personalize and adapt learning material to enrich the experiences of individual students and foster a lifelong learning mindset beyond the formal classroom. However, the rigid "one-size-fits-all" approach of Massive Online Open Courses (MOOCs) correlates with over 90% dropout rates and highlights the need for adaptive individualization [37]. On the other end of the spectrum, when confronted by the seemingly endless repositories of learning material, learners need some guidance and external motivation via a personalized road map connecting new topics to already learned ones. Some authors have compared modern learning to the "Netflix of learning" [36]. We should not intend to replace classroom learning by simply placing students in front of videos and expecting them to learn.

One of the pitfalls of promoting self-navigated learning in a vast repository of learning content is that a learner is unlikely to "binge watch" the learning material as they would an entertainment series. Moreover, without a guided Learning Management System (LMS), it is difficult to achieve the prescribed or self-described learning objectives [17,36]. There also exist other pragmatic difficulties, such as the time constraints of an instructor or course designer which limit real-time exploration of new content, and in turn the consistent awareness of the latest learning material that becomes available. The amount of microlearning content grows daily with the production of videos, written exposition, and objective-based online courses; by organizing a repository of content to which multiple authors can contribute collaboratively, we may enable discovery of new material and relationships between existing content that enhances learning. Meanwhile, instructors could spend time encouraging exploratory discussion and answering specific questions about student learning to cultivate curiosity rather than manually curating content and assessments.

We contend that in order to find the middle ground between MOOC-style and completely self-guided learning, content repositories must be organized so that attributes such as their relevance to specific learning objectives flow with other material and that content difficulty level can be identified for individual learners. Our network science-based approach develops a framework for organizing potentially vast collections of learning material to enable traditional and novel network science analysis methods to find applicable microlearning content for individual learner profiles. This approach also enables instructors and students to interact with alternative and related topics throughout their learning journey.

In this paper, we present an expanded view of a dynamic network of knowledge (DNoK) design to organize a repository of learning content [31]. We also define a minimum viable DNoK for an instructor or LMS to address a specific set of learning objectives, based on best-practices for teaching and learning. Previous frameworks for the DNoK relied on subject matter expert (SME) curation and/or validation of contents' tagged keywords. This presented obstacles with respect to the scalability of the entire network and the subjectivity of author self-promotion for their respective content via extraneous keyword use. To mitigate this, we now introduce applications of current topic modeling methods for keyword extraction including the Bidirectional Encoder Representations from Transformers (BERT) and two fast extraction methods called YAKE! and WordWise [6,12,19,33]. We offer both qualitative and quantitative comparisons of the results between these methods, and demonstrate the utility of network science analytic approaches, such as traditional community detection, to identify key topics at the macro-level. We examine the growth and topological behavior of the DNoK, and infer how they may continue to behave at scale.

2. Related and Previous Work

Existing methods relied on static repositories of microlearning content while focusing on algorithms to optimize the generation of PLPs within constraints such as available time. Emerging research is centered upon addressing individualized learner profiles [10,25]. These static repositories have been most prevalent for asynchronous learning methods through online access such as those offered within Coursera, Khan Academy, MIT Courseware etc., which are linear MOOC-style courses where learners go through the same learning experience at the same pace of learning standard content. Extending work from these static methods to the presentation of PLPs is centered upon optimization algorithms to maximize achievement of learning objectives within the constraints of the learner profile. However, existing work can still benefit from addressing the associated cognitive level and difficulty level of the material presented, and the perceived flow and pace of the course as PLPs were generated [24]. More recent research proposed a dynamic alternative to static, manually-curated repositories of learning material for creating live and dynamic PLPs right as the learner interacts with the content [31].

In other research areas, individual semantic Knowledge Graphs (KGs) were also used to optimize the generation of PLPs for each learner, but once again they focused on the algorithm for the learners' paths, not the corpus of content that may be presented within the PLP for targeted and adaptive learning [8, 18]. These pursuits are promising, as our network science-based framework presents the environment to identify content-centric knowledge graphs that can complement learners' individual knowledge graphs to bolster their learning. We propose ongoing and future work in this space in Section 5.

Incorporating learning theory and best practices into PLP generation remains in the early stages of development. This is particularly true when considering the diverse ways in which individual students learn most effectively. For example, research addresses requirements for educating the current 'Netflix generation' by creating teaching and learning videos using mixed methods to investigate their effectiveness to support students across distinct STEM disciplines, aiming to improve teaching metrics such as retention and progression [30]. Additionally, work in Cognitive Theory of Multimedia Learning (CTML) emphasized the need for dual encoding information by engaging the reader through multiple modalities at the same time to build mental representations to promote learning [22, 23]. By tailoring PLPs to reflect the unique learning preferences and needs of each student while engaging them through multiple modalities, educators can more effectively foster engagement, and drive meaningful improvements in educational practices. If sufficiently organized, microlearning content can be optimally identified and ordered by educators and learning management systems to facilitate deeper understanding within these theories.

Recent work has also identified the potential to use a network science approach to teach generative AI models, such as using a content repository to generate a semantic knowledge graph to generate new relationships by referencing other areas of science [5]. This approach could be extended to providing specific context for a generative model to answer possible learning path prompts, without the expense involved in training a domain-specific model. The network design of related micro-learning content provides a space to do this toward a specific set of learning objectives. While outside the scope of this particular paper, we address approaches in current and future work in Section 5.

3. Methodology

In this section, we describe the framework design of our Dynamic Network of Knowledge (DNoK) and demonstrate the iterative addition of content. We demonstrate automated methods to represent relationships between microlearning content through the extraction of keywords in common. First, we define a minimum viable NoK in accordance with learning theory by building a base NoK upon which content may be collaboratively added. We then describe our process for using topic modeling methods to extract the keywords which define edges between content in our NoK construct, and how that content is iteratively added.

Expanding upon previous work using a SME-curated network, we specifically examine and reference a DNoK consisting of linear algebra content that is freely available on the WWW, while incorporating CTML and best practices for teaching and learning throughout the process [9, 23, 31].

3.1. *Minimum Viable Network of Knowledge*

For much of this section, we leverage previous work as described in [31]. We first develop the framework of the base NoK (bNoK) by defining its nodes.

Consider a repository of microlearning content. Then a **content node** v^c represents a microlearning content object, which is self-contained content such as a subsection of a text, a single video, or a single blog post. A majority of the nodes in our NoK represent this content, where v_i^c represents content c_i.

Let an **author node** v^a represent the author of the content. Let a **modality node** v^m represent a modality or format in which a microlearning content object may be presented, such as video, text, or code. Note that each content node v_i^c also contains the metadata of the microlearning content c_i, particularly the author-assigned keywords, publication date, and resources, not already represented by an author or modality.

We then assign edges between content nodes and their respective authors and modalities. For each content node, assign an **authorship edge** e^a between each v^c

and respective author v^a. Additionally, for each content node, assign a **modality edge** e^m between each v^c and respective format v^m. (Notation-wise, we may also denote edges by their respective nodes. For example $e^a \equiv v^c v^a$. For this discussion, we elect to refer to edges by the e^a notation.)

We introduce weighted undirected edges between content nodes and their attributes, and between different content nodes throughout the bNoK. Edges that attach content nodes to their modalities and to their authors are assigned weights of 2 and 8, respectively. These assignments are based on tuning of the NoK in [31] for optimizing community detection methods.

Edge weights between different content nodes, on the other hand, directly correlate to the number of topic keywords they have in common. For example, if two content nodes share four keywords (or keyword n-grams) in common, then they will share an edge with weight 4. Throughout this paper, we use "keyword" and "keyword n-grams" interchangeably.

Formally, we define our keyword edges:

Definition 1. Weighted Keyword Edges. Let a keyword k be a keyword shared in the metadata of two content nodes v_i^c and v_j^c. Define a keyword edge e^K with weight m where $m \geq 1$ denotes the count of common keywords between nodes.

Notice that we denote weighted keyword edges with a capital K, where the weight m directly reflects the count of common keywords. The bNoK is the multimodal graph consisting of the nodes and weighted edges defined above:

Definition 2. Base Network of Knowledge (bNoK). Let V be the set of nodes v^c, v^a, v^m, and E be the set of edges e^K, e^a, e^m. Then *bNoK* is the graph $bNoK(V, E)$.

With the node and edge structure defined, we use best-practices in teaching and learning to set the conditions for a Minimum Viable NoK. First, suppose that we are given a full set of learning objectives to be achieved by learners and prescribed by the instructor or learning engineer. Assume a set of content covers all prescribed learning objectives. We adopt the "dual-encoding" doctrine for learning in a multimedia environment and apply the requirement for both audio and visual learning channels to be engaged [22,23]. Then we require a set of written/visual material (ex: textbook or pdf content) and a set of audio material (ex: video or podcast), where each set fully covers the prescribed learning objectives.

Definition 3. Minimum Viable Network of Knowledge. Suppose we have a single learning objective. Then, to achieve this objective, we require a visual content node and an audio content node to support the same learning objective, v_{vis}^c and v_{aud}^c. While the content nodes may share an author, they will have two different modalities. Therefore our *Minimum Viable Network of Knowledge* is the graph $NoK(\{v_{vis}^c, v_{aud}^c, v^a, v_{vis}^m, v_{aud}^m\}, \{e_{vis}^a, e_{aud}^a, e_{vis}^m, e_{aud}^m, e^K\})$ with 5 nodes and 5 edges.

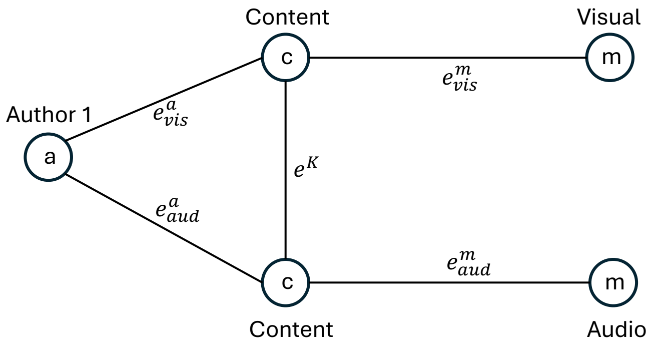

Fig. 1. A generic Minimum Viable Network of Knowledge. Letters indicate node type, e.g. ‘a’ is an author node v^a. Since each single content node of each modality must cover the same learning objective, they will share an edge e^K. The author does not necessarily have to be the same for both modalities; this simplified example represents the minimum possible realization.

Figure 1 shows a generic Minimum Viable Network of Knowledge. Suppose an instructor or learning engineer desires to present multiple options for a learner to experience. PLP development should focus not only on which content is presented to a learner, but also how that content is consumed by the learner. Opening the aperture from just audio and visual choices, an instructor may canvass prospective students for preferred modalities. Surveys conducted at the beginning of potential PLP-driven courses have offered preferences for learning material including research papers, websites, discussion sessions, lectures, raw presentation slides, and ‘do it yourself’ experiences. Many of these can be captured within NoK content presented to learners [10, 11]. In this case, we define the Minimum Desired NoK that may be used as the bNoK:

Definition 4. Minimum Desired Network of Knowledge. Suppose we have a single learning objective. Let m be the minimum number of optional modalities offered to learners, where $m \geq 2$. The *Minimum Desired Network of Knowledge* is the graph $NoK(\{v_1^c, ..., v_m^c, v^a, v_1^m, ..., v_m^m\}, \{e^a, e_1^m, ..., e_m^m, e_1^K, ..., e_m^k\})$ with order $3+2m$ nodes, and size $3+2m$ edges.

In reality, instructors and students will be interested in achieving multiple learning objectives requiring multiple microlearning content objects. The Minimum Viable Network of Knowledge sets a lower bound for extending the bNoK to the Dynamic NoK through iterative content addition.

Before we fully define a content addition process, we discuss keyword extraction as part of the content addition algorithm.

3.2. *Keyword Extraction*

Previous exploration of the DNoK relied on manually SME-curated keywords for edge representation. In this subsection, we discuss the use of automated topic modeling approaches to extract keywords and define common keyword edges.

We examined five keyword extraction methods that are openly available, summarized below. This study is not intended to be a comprehensive comparison of keyword extraction methods as performed in [19] and [26]; rather, we selected representative models across different techniques and speeds to assess their performance in a mathematically topic-specific area (linear algebra). We acknowledge that we deliberately "black-box" our methods for keyword extraction. Since the development and use of new methods for topic and large language models remain a rapidly evolving field, we assume that current models will be supplanted by improved versions that can be harnessed by future DNoK iterations.

- **KeyBERT**. KeyBERT uses BERT embeddings and simple cosine similarity to find the subphrases (or n-grams) in a document that are the most similar to the document itself [12, 28].
- **spaCY**. spaCY is an open source library for performing natural language processing and unsupervised keyword extraction [20].
- **Rapid Automatic Keyword Extraction (RAKE)**. RAKE is part of the Natural Language Toolkit (NLTK) in Python [29].
- **YAKE! - Yet Another Keyword Extraction method**. YAKE! is a lightweight statistical keyword extraction method that does not rely on dictionaries or external corpora [6].
- **WordWise**. WordWise is a relatively fast method that incorporates part of spaCY speech tagging and Sentence-BERT embeddings [33].

We also considered Latent Dirichlet Allocation (LDA) and Term Frequency-Inverse Document Frequency (TF-IDF) for keyword extraction. However, because we want to iteratively build the DNoK by iterating from a comparatively small bNoK, as discussed in Section3.3, we desire unsupervised methods that can process a single document at a time. LDA, for example, performs more accurately when given numerous documents in which we would ideally apply the method to the entire network of $1,000$ or more content nodes [13]. Therefore, we do not apply those methods here.

We first screened the keyword extraction methods on a small set of video transcripts, text extracted from websites, and textbook sections to validate and tune their viability in building a full dynamic NoK. In each case, we tuned the methods to produce keyword n-gram strings between one (ex: 'determinant') and three words (ex: 'elementary row operations') long. To assess their viability in building a bNoK we qualitatively validated their keyword output. When presented with linear algebra topics, some of the methods exhibited difficulty extracting usable keyword n-grams, particularly with video transcripts. For example, when given content surrounding the solutions of linear systems through Gaussian Elimination, spaCY, NLTK, and RAKE output keyword lists of “perf, theoretical, computations, difference, questions, big matrix, textbook might ask.” We considered this output less useful when contrasted with the other methods, which more accurately returned keywords such as “solution, elementary row operation, row echelon form.”

Ultimately, we selected KeyBERT, YAKE!, and WordWise methods to build linear algebra DNoKs for our study, as they produced keywords with the most usable results consistently throughout our sample. However, despite tuning efforts, these keyword extraction methods still produce extraneous n-gram artifacts, such as “row elementary” or “solution row,” as well as colloquialisms used by video narrators. Regardless, we assess that these artifacts would be minimized to a trivial effect on the overall network since the edges between nodes are based on keyword n-grams in common. Thus, in practice, these artifacts would be effectively tuned out.

In summary, for every content video, blog post, or textbook section, we apply our three selected keyword extraction methods. We include generalized terms such as “mathematics,” “applied math,” and “linear algebra” to each method’s list of stop-words, and accepted n-grams between one and three words long. During keyword extraction for each method, we lemmatize each keyword list, respectively–plural forms and different tenses of words such as “transposes,” “transposing,” and “transposed” were replaced with a single instance of “transpose.” As we will examine in Section 4, we did not limit the number of keyword n-grams assignable by the KeyBERT method. For the YAKE! and WordWise extraction methods, we adjusted the default setting to 15, which was the average output of KeyBERT during the initial sample runs.

3.3. *Growth as a Dynamic Network of Knowledge*

With automated methods to extract keywords and assign them to nodes, we now discuss the growth of the network from the bNoK to the Dynamic Network of Knowledge. Suppose we want to collaboratively add content from new authors or diverse modalities into our microlearning content repository. Then we would iteratively add content nodes (or sets of content nodes), with respective edges per Algorithm 1, building on the same ideas used to create the base NoK. This algorithm is an expansion of the rule-based content addition in [31]. Each of the content objects within a set (a single video, blog post, or textbook section) is brought into the network as a node. If a respective author node is not yet in the NoK, then we add it to the node list.

Once content objects and authors are brought into the NoK as nodes, we assign edges between content and their author- and modality-attribute nodes. We then compare keywords between newly added content and each content node existing in previous iterations of content addition. A keyword edge is assigned between the new content node and existing content nodes, with a weight that directly correlates to the number of keywords (or keyword n-grams) they have in common.

Using this algorithm, we can formalize our definitions leading to the DNoK.

Definition 5. Iterative addition of content. Consider the addition of microlearning content nodes via Algorithm 1. Let V_n^{add} denote the set of nodes added during iteration n and E_n^{add} denote the set of edges added during iteration n. I_n is the graph after the nth iteration of node and edge addition, where $E_n = \{E_{n-1} \cup E_n^{add}\}$. Then iteration I_n is the graph

$$I_n(V_{n-1} \cup V_n^{add}, E_n).$$

Note that $bNoK \equiv I_0$. Finally, we formally define the DNoK, the Dynamic Network of Knowledge that grows from the existing bNoK.

Definition 6. DNoK. Consider the graph I_n. Let V_n^{ret} denote the set of retired nodes during iteration n. Let V_n be the set of nodes consisting of the previous iteration $n-1$, adding nodes V_n^{add} and removing nodes V_n^{ret}.

$$V_n = \{V_{n-1} \cup V_n^{add}\} \setminus V_n^{ret}$$

Then the *DNoK* is the subgraph of I_n induced by the resultant set of nodes V_n:

$$DNoK = I_n[V_n]$$

We notice a few things here. First, if $V_n^{ret} = \emptyset$, then the DNoK is simply I_n. Iterative addition/retirement of content also provides a "record" of the dynamic aspect of our network of knowledge. Finally, we may also examine the lower and upper bounds of edge growth in our algorithm.

Lemma 1. *Minimum edge growth per added content node. For each content node added to the DNoK, it must have a respective author and modality edge. Denote the number of nodes added as $p = |V_n^{add}|$.* ***The lower bound of edge growth*** m ***per*** p ***added nodes is*** $2p$***.***

Lemma 2. *Maximum edge growth per added content node. For each content node added to the DNoK, it must have at most one respective author and one modality edge. Let $V_c : V_c \subset V_n$ be the set of all content nodes in the network after the latest iteration of content addition/retirement.* ***Then the upper bound of edge growth*** $\mathbf{m}$ ***per*** $\mathbf{p}$ ***added nodes is*** $(|\mathbf{V_c}| - \mathbf{1}) + \mathbf{2p}$***.***

Corollary 2.1. *Maximum DNoK size. Let $V_c : V_c \subset V_n$ be the set of all content nodes in the latest iteration of content addition/retirement. Each content node must have a respective author and modality edge.* ***The maximum number of edges in the DNoK is given by:***

$$|\mathbf{E_n}| \leq \binom{|\mathbf{V_c}|}{\mathbf{2}} + \mathbf{2}|\mathbf{V_c}|$$

We may reference these bounds as we analyze the growth of the linear algebra DNoKs produced by our three keyword extraction methods.

Algorithm 1 Adding content to the DNoK: Single iteration

Input: contentList: A set of microlearning content with metadata.

Output: Updated Dynamic Network of Knowledge (DNoK) consisting of V: the set of author, modality, and content nodes and E: the set of authorship, modality, and weighted keyword edges.

```
Initialization: Existing DNoK(V,E)
nodes = V
edges = E
for each content in contentList do
   Title = content.metadata.title
   Author = content.metadata.author
   Modality = content.metadata.modality
   Transcript = GetTranscript(content)
   Keywords = ExtractKeywords(Transcript)
   FilteredKeywords = Lemmatize(Keywords)
   newcontentNode = CreateNode(Title, Author, Transcript, FilteredKeywords, Modality, type="content")
   nodes.append(newcontentNode)
   if Author notin nodes[type="author"] do
      newauthorNode = CreateNode(Author,type="author")
      nodes.append(newauthorNode)
   end if
   for each authorNode in nodes[type="author"] do
      if authorNode[Author] == newcontentNode[Author] do
         authorEdge = CreateEdge(newcontentNode,authorNode)
      end if
   end for
   for each modalityNode in nodes[type="modality"] do
      if modalityNode[Modality] == newcontentNode[Modality] do
         modalityEdge = CreateEdge(newcontentNode,modalityNode)
      end if
   end for
   for each contentNode in nodes[type="content"] do
      if contentNode != newcontentNode do
         commonKeywords = contentNode[FilteredKeywords] $\cap$ newcontentNode[FilteredKeywords]
         if commonKeywords != $\emptyset$ do
            weight = length(commonKeywords);
            KWedge = CreateEdge(contentNode, newcontentNode, weight);
            edges.append(KWedge);
         end if
      end if
   end for
end for
return(nodes,edges)
end
```

Our example linear algebra network design focuses on freely available content modeled as the nodes within the linear algebra DNoK. It consists of textbooks such as those by Stephen Leon and Gilbert Strang, video lectures by expositors such as Trefor Bazett and Grant Sanderson, blog posts, and coursework [1–4, 14–16, 21, 27, 32, 35]. To support our model, we make the following assumptions when building our linear algebra DNoK:

- Subject Matter Experts (SMEs) curate the content to be included in and added to the base NoK. This ensures the content is validated with respect to credible, authoritative sources and tagged appropriately (this is usually already done in textbooks, blogs, and YouTube videos by default; however, machine learning methods could be used to support and achieve both validation and tagging).
- Generalized keywords such as “linear algebra”, “mathematics”, “applied mathematics”, etc. that do not refer to specific topic areas within linear algebra are excluded. These keywords are added to stop-word lists in the algorithms discussed in Section 3.2.
- Since we seek micro-learning materials as self-contained content based on learning outcomes, longer pre-existing content, particularly audio/video, should be split into smaller parts, as with sections and sub-sections in a textbook. Generally, microlearning content should be available in 10-15 minute chunks, according to best practices in learning theory [7]. This allows us to incorporate conference lectures, TED talks, and other high-quality content that SMEs have created over time. Segmented microlearning content can be accessed at the learners’ pace.

4. Results and Analysis

In this section, we compare network statistics between our keyword extraction methods, as well as with the original SME-curated DNoK described in [31]. Using the methodology described in Section 3, we started with a bNoK and conducted 107 iterations of content addition for each of the keyword extraction methods. The DNoK produced by the KeyBERT keyword extraction method is shown in Figure 2.

4.1. *Network Statistics*

Table 1 compares the network statistics between our three selected keyword extraction methods and the original SME-curated network. Most notable is that there are far fewer edges with respect to the number of nodes in the SME-curated method when compared to the three automated methods. We assess this difference as an effect of the limit that was placed on the number of SME-curated/validated keywords allowed per content node. Recall that no limit was placed on the number of keywords assigned by KeyBERT. Meanwhile, a limit of 15 keywords was placed on YAKE! and WordWise since that was the average output of KeyBERT dur-

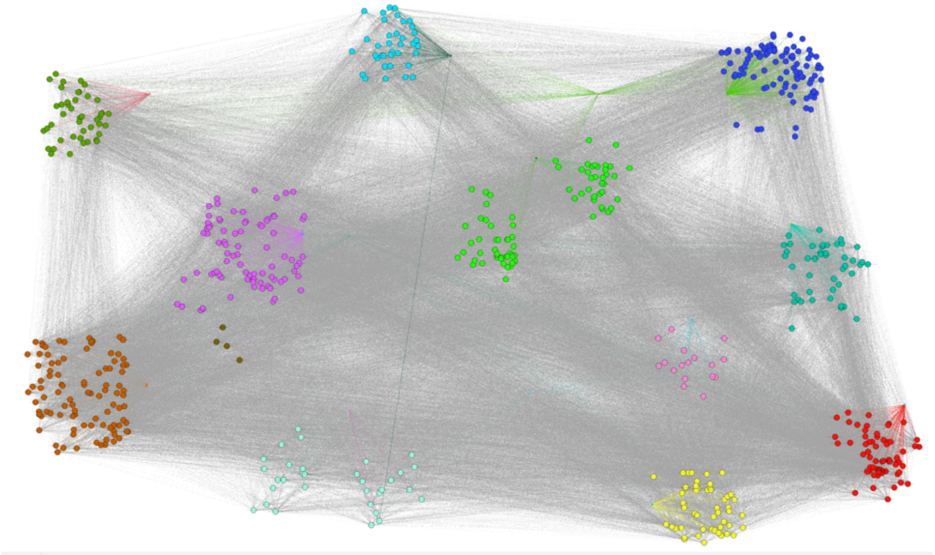

Fig. 2. The DNoK produced by the KeyBERT keyword extraction method, consisting of 665 nodes and 62094 edges. Node color indicates authorship, and they are geographically grouped by author and modality. More network summary statistics are found in Table 1.

ing the initial sampling and tuning. Interestingly, we observe respective maximum keyword-in-common edge weights of 15, 12, 8, and 9 for the KeyBERT, YAKE!, WordWise, and SME-curated method. Despite having the heaviest maximum edge weight, there were less KeyBERT-assigned edges than in the YAKE! and Wordwise DNoKs. Additionally, the small average path and large clustering coefficients show that all methods to create DNoKs create small world networks. This structure supports the creation of personalized paths as there are multiple pathways through content from different authors and modalities.

Figure 3 depicts the edge growth behavior across our methods. Once again, we see significantly faster edge growth per node than the SME-curated keyword method. However, all of the methods remained below approximately half of the upper bound discussed in Section 3.3. The methods described in [31] created a DNoK of 616 nodes while the updated algorithm built DNoKs consisting of 665 total nodes each from updated content. Between the three automated keyword extraction methods, the YAKE! and WordWise methods performed the most similarly with respect to edge growth.

Despite the differences in the overall number of edges, Figures 4(a)-(c), which show the edge weight distributions of the respective DNoKs, indicate similar distributional behavior between the three different methods. Together with the relatively high average clustering coefficients and low average path lengths, as indicated in Table 1, we observe small world network behavior throughout the different DNoK framework methods [34].

Table 1. Comparison of network statistics between the different methods used for this study. The SME-curated keyword network was produced in [31].

	Method			
	KeyBERT	YAKE!	WordWise	SME-Curated
Order (Nodes)	665	665	665	616
Size (Edges)	62094	103319	106574	15649
Node Degree Stats				
Mean	186.7	310.7	320.5	50.8
Max Degree	444	561	563	328
Mean Weighted Degree	239.8	471.3	469.7	146.0
Max Weighted Degree	788	1116	1052	850
Edge Weight Stats				
Mean Edge Weight	1.28	1.52	1.47	2.7
Median	1	1	1	2
Standard Deviation	0.92	1.00	0.88	1.51
Max Edge Weight	15	12	8	9
Average Clustering Coefficient	0.81	0.76	0.79	0.64
Average Path Length	1.81	1.54	1.53	2.14
Diameter	5	4	5	5

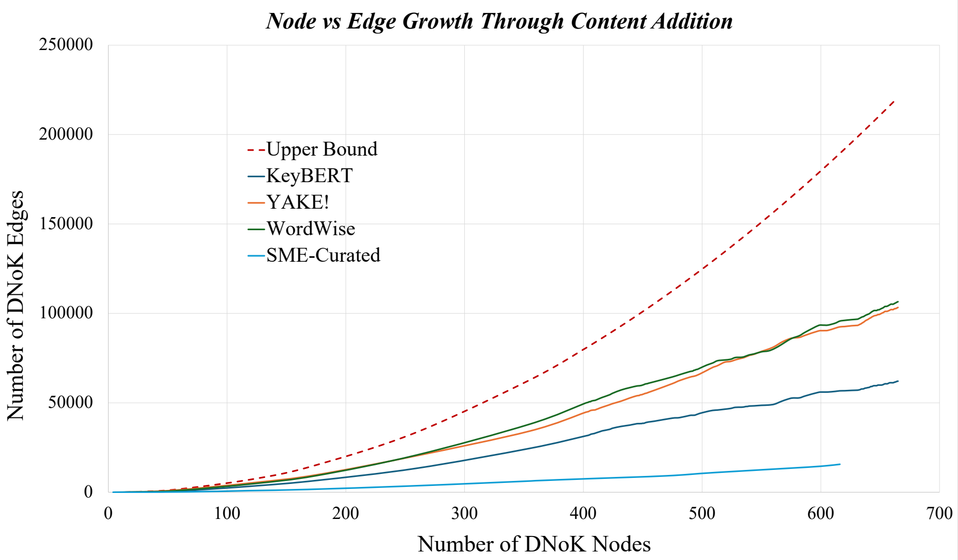

Fig. 3. The edge growth of the DNoK using our three different keyword extraction methods as well as the SME-curated keyword DNoK from Ref. [31]. The upper bound from Corollary 2.1 is displayed with the red dashed line.

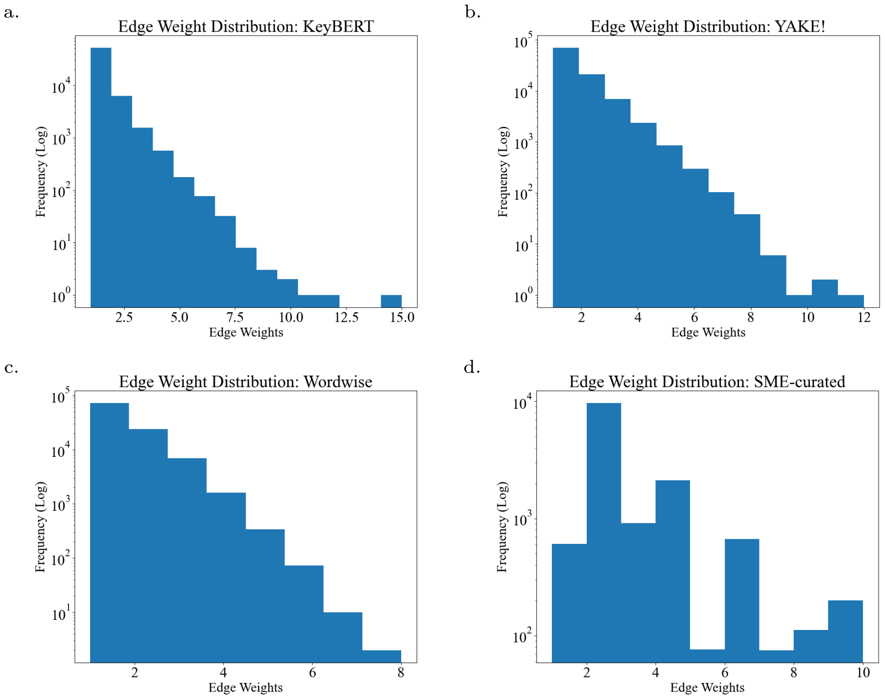

Fig. 4. From the top left, the edge weight distributions of the DNoKs produced by a. KeyBERT, b. YAKE!, c. WordWise, and d. SME-curated keywords, respectively. As with the other network statistics indicated in Table 1, the SME-curated keyword DNoK was the most dissimilar to the three automated methods.

4.2. *Topic identification through community detection and edge-growth analysis*

We conduct two qualitative analyses of the keyword extraction and edge-assignment methods by examining the community detection results of the content nodes, as well as the edge growth rate for niche vs. common topics in the DNoKs.

Recall the visualization of the KeyBERT-produced DNoK in Figure 2. Using the methods discussed in [31], we performed a community detection method on only the content nodes and identified topic-based communities throughout, shown in Figure 5.

We highlight one of the specific topics, “Eigenvalues and Eigenvectors,” in Figure 6. In this example, community detection identified eigenvalue and eigenvector-related content across all authors in the DNoK. This method identified other topics throughout all three method-produced DNoKs in the same fashion.

Table 2 shows the methods’ top modularity classes of topic communities. They are displayed in order of their modularity class size, indicated in percent of the total node population. We assess from these results that all three extraction methods

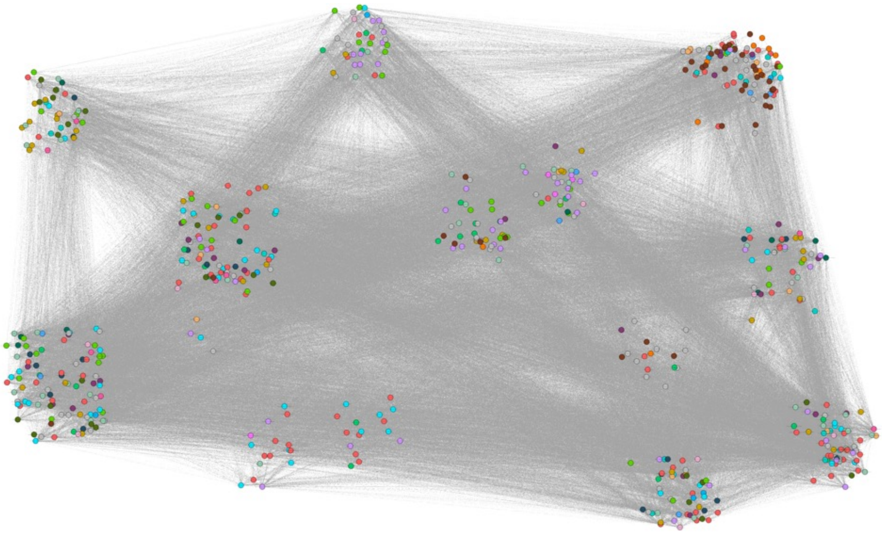

Fig. 5. Community detection using the Louvain Method identified the topic-based communities, indicated in different colors throughout the DNoK produced by the KeyBERT method. Nodes remain grouped with their respective authors, as depicted in Figure 2.

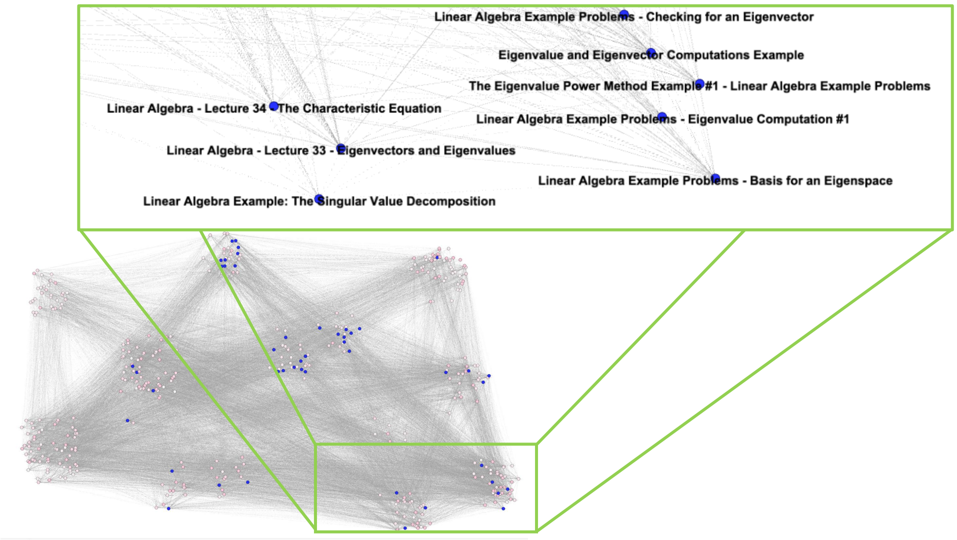
Linear Algebra Example Problems - Checking for an Eigenvector
Eigenvalue and Eigenvector Computations Example
The Eigenvalue Power Method Example #1 - Linear Algebra Example Problems
Linear Algebra - Lecture 34 - The Characteristic Equation
Linear Algebra Example Problems - Eigenvalue Computation #1
Linear Algebra - Lecture 33 - Eigenvectors and Eigenvalues
Linear Algebra Example Problems - Basis for an Eigenspace
Linear Algebra Example: The Singular Value Decomposition

Fig. 6. The topic based community surrounding “Eigenvalues and Eigenvectors” in the DNoK produced by the KeyBERT method is shown in blue. Note the blue topic nodes span different authors’ sets of content.

were able to identify similar key topics at the macro level. More importantly, this indicates that all three automated methods were able to identify content across different authors and modalities in each topic area which could be presented to an individual learner. This demonstrates our DNoK structure supports our goal of CTML-style dual-encoding.

Table 2. The most prevalent topic communities in each of the respective DNoKs. Relative community size is indicated as a percentage of the total content nodes in the respective DNoK. Although the size per topic varied slightly across the three methods, the topics were generally similar across all three methods.

KeyBERT		YAKE		WordWise	
Title	Size	Title	Size	Title	Size
Least Squares Applications	9.62	Least Squares Applications	8.72	Subspaces	8.87
Orthogonality	7.07	Linear Transformations	8.12	Solution of Linear Systems, Least Squares	8.72
Eigenvectors and Eigenvalues	6.32	Subspaces	5.26	Eigenvectors and Eigenvalues, Diagonalization	8.27
Vector Spaces	6.17	Orthogonality	5.11	Least Squares Applications	6.92
Linear Transformations	6.02	Solutions of Linear Systems	4.96	Orthogonality	6.77
Change of Basis, Isomorphism	5.41	Vector Spaces / Subspaces	4.21	Determinants	5.41
Linear Independence	4.81	Linear Independence	4.21	Vector Spaces	3.61
Matrix Inverses	3.91	Determinants	3.61	Linear Independence	2.26
Subspaces	3.46	Diagonalization	3.46	Matrix Inverses	2.26
Gaussian Elimination	3.01	Basis and Dimension	3.31	Gaussian Elimination	2.26

Another qualitative analysis examined the growth rate between niche versus common topics throughout the DNoKs. Table 3 shows the growth rate, in the difference between the number of edges added for each of Gregory Gundersen blogs, which were added into the DNoK one at a time. Dr. Gunderson's blog topics varied between highly niche topics (in relation to linear algebra study), such as "Convex combinations as lines," and general linear algebra concepts, such as "A geometrical understanding of matrices." The results align with expectations that the average edge growth for niche topics was much smaller than that associated with general concepts. From this we can assess the validity of the automated keyword extraction methods, as niche topics would tend to produce fewer keywords in common with other content than the topics covering more general concepts. However, this table also indicates some notable differences in edges added between the methods for a few of the nodes. For example, YAKE! produced 278 more edges for the "High Dimensional Variance" node than the other two methods. This may highlight a need to further examine and tune the keyword extraction methods and potentially refine

Table 3. We list the titles of online expository notes by Dr. Gregory Gundersen [14]. For the most part, niche topics indicated small numbers of added edges per node, while more common concepts display higher numbers of added edges per node. Differences between the methods may highlight some weaknesses in the keyword extraction for that topic.

Title	Edges Added		
	KeyBERT	*YAKE*	*WordWise*
High Dimensional Variance	3	281	3
Matrices as Functions, Matrices as Data	309	376	372
Conjugate Gradient Descent	27	24	20
Understanding Positive Definite Matrices	5	289	338
Convex Combinations as Lines	6	351	334
Linear Independence, Basis, and Gram-Schmidt Algorithm	133	311	295
Why Shouldn't I Invert that Matrix?	284	321	380
Matrix Multiplication as the Sum of Outer Products	278	471	508
Summing Quadratic Forms	9	29	29
Completing the Square	10	51	144
Randomized Singular Value Decomposition	285	283	350
Proof of the SVD	284	294	366
SVD as simply as possible	14	366	386
Woodbury Matrix Identity for Factor analysis	2	7	2
Modeling Repulsion with Detrimental Point Processes	3	8	3
A Geometrical Understanding of Matrices	282	476	400
Two Forms of the Dot Product	157	53	303

the stop-word lists that are being used.

4.3. *Degree Distribution*

As we are interested in the strength of relationships between similar topic content, we examined the weighted degree distribution across the DNoKs produced by our three different methods. To display the dynamic nature of the DNoK growth, Figures 7 through 9 show respective sets of nine histograms indicating the weighted degree distribution at various points in their iterative content addition. In all cases, the distribution tended "right" through the iterations, as both the mean and maximum weighted degrees increased with content addition. The comparative mean and max weighted degrees are shown in Table 4.

The histograms also show some multimodal character, which could indicate the relative "youth" of our networks having only 665 nodes. As part of our ongoing work, we intend to add more content into the respective DNoKs for further study in producing a synthetic DNoK model.

Overall, our results show that automated keyword extraction methods provide a viable method for building a Dynamic Network of Knowledge under our construct. Furthermore, they provide a tool for examining continued growth to study our network design at scale.

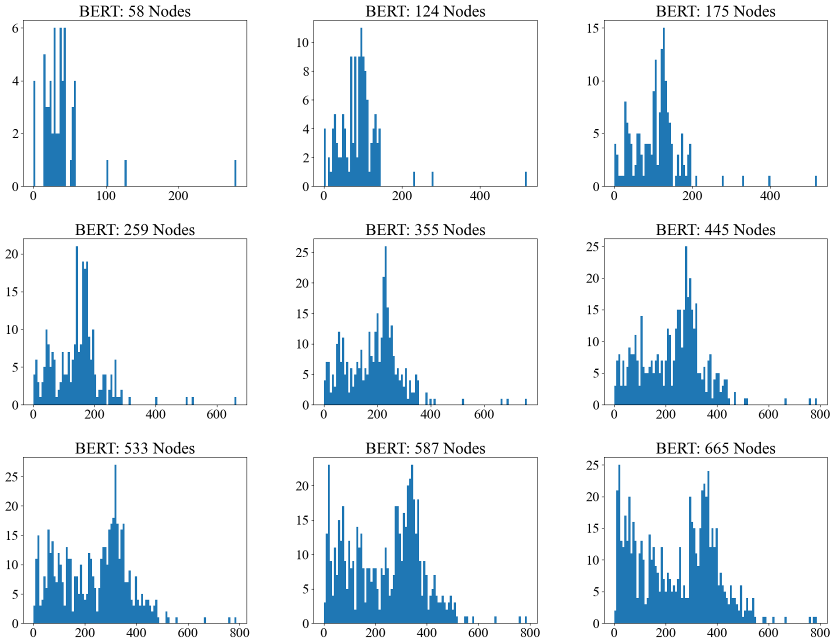

Fig. 7. Starting from the top left, the weighted degree distribution of the KeyBERT-method DNoK at nine iteration points. The mean weighted degree shifted from 38.9 to 237.4.

Table 4. Comparison of mean and max weighted degree across KeyBERT, YAKE!, and WordWise at nine iteration points. All three methods demonstrated increasing mean and max weighted degree.

Nodes	KeyBERT	KeyBERT	YAKE	YAKE	WordWise	Wordwise
	Mean	Max	Mean	Max	Mean	Max
58	38.9	280	107.3	280	61.6	280
124	88.1	520	186.4	520	141	520
175	109	520	219.3	520	179	520
259	145.3	664	283.2	664	263.5	664
355	186.7	760	327.2	760	337.5	760
445	228.4	788	399	843	415.1	824
533	234.1	788	445.5	968	431	918
587	237.7	788	457.8	1016	453.5	985
665	239.5	788	466.7	1104	469.7	1052

5. Conclusions and Future Work

Expanding the work in Dynamic Network of Knowledge design, we define an automated method of extracting keywords from microlearning content and assigning weighted edges/connections between the microlearning content based on their respective keywords in common. By automating the keyword assignment, we removed subjectivity of author-tagging and the potential inconsistency of different SMEs as-

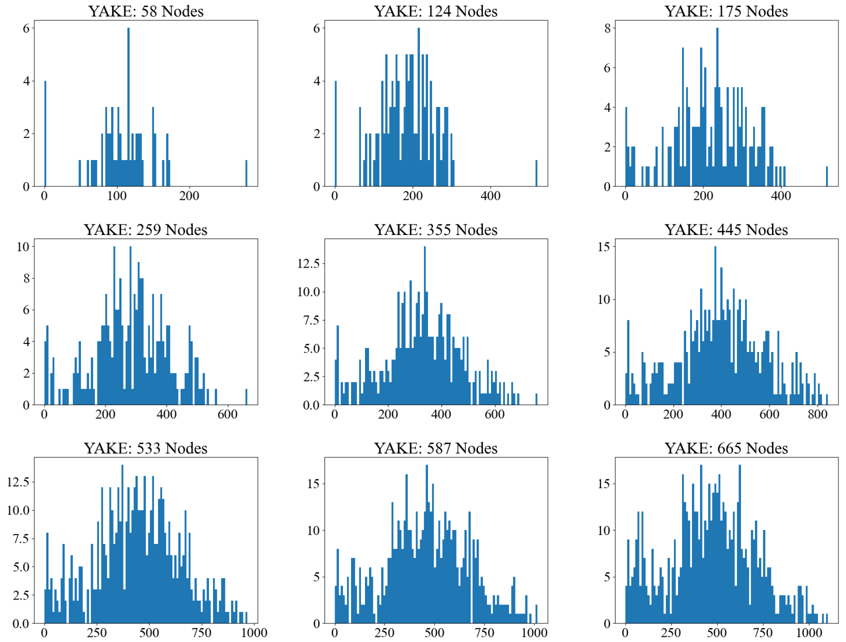

Fig. 8. Starting from the top left, the weighted degree distribution of the YAKE!-method DNoK at nine iteration points. The mean weighted degree shifted from 107.3 to 466.7.

signing keywords.

Furthermore, we found that our three keyword extraction methods–KeyBERT, YAKE! and WordWise– performed similarly in producing DNoKs that allowed for topic identification through community detection. Performance-wise, we found that even though the methods had similar results, on average WordWise was about four to five times faster than KeyBERT in adding content nodes to the DNoK, while YAKE! was almost five-six times faster than KeyBERT. This may lead to the preference of fast-extraction methods over BERT embeddings for this keyword-edge assignment use case, which is useful when crawling data to create dynamic environments that support dynamic personalized learning paths.

The methods are not perfect, however. Some inconsistencies in edge and keyword assignment still appeared, which indicates the need to further tune or even refine the models used for this process. Additionally, in order to examine the behavior of the DNoK at scale via a synthetic model, more content-adding iterations will be beneficial in characterizing the edge weight and degree distributions of the DNoK. We intend to continue building real-world DNoKs to explore the convergence of our data toward theoretical distributions.

PLP algorithms to identify and present optimal content can be computationally expensive [24]. The network environment of our DNoK provides a space where methods more granular than community detection could be used to identify and

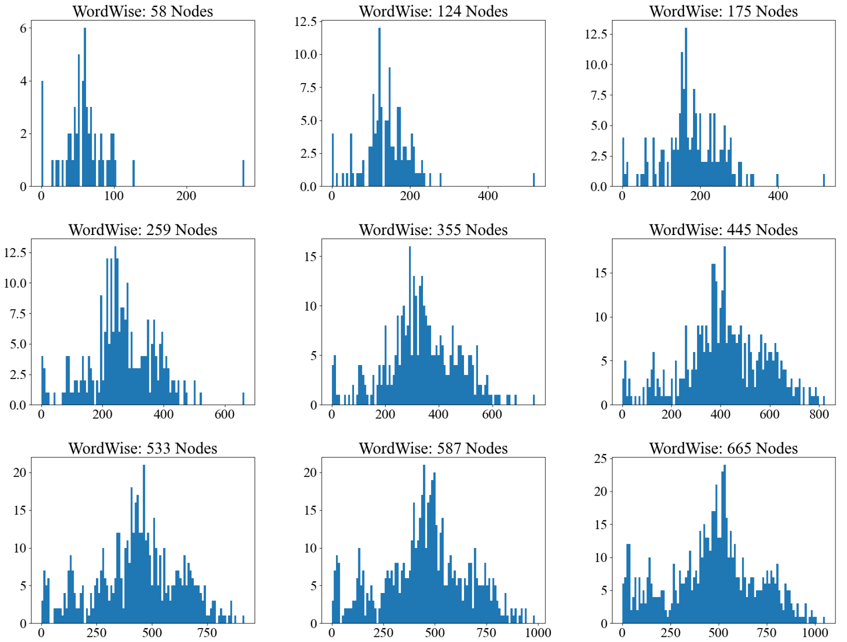

Fig. 9. Starting from the top left, the weighted degree distribution of the WordWise-method DNoK at nine iteration points. The mean weighted degree shifted from 61.6 to 469.7.

quantify knowledge graphs centered on microlearning content. For example, an instructor could canvas students by having them build concept mind maps of their understanding of the relationships between topics. If these mind maps became part of the learner's individual profile, instructors or learning management systems could find DNoK content with complementary knowledge graph structures that would help strengthen existing understanding or seed new relationships between concepts, further enhancing learning.

It is important to note that within the DNoK we can quickly identify content from multiple authors and modalities that discuss the same topics, which serve as a ready set of alternate learning material if additional exposure is desired or needed. Less apparent relationships that exist within the DNoK can also facilitate suggestions in the recommender system style of "you may also be interested in topic X," which could serve to instill habits toward lifelong learning. As the DNoK framework matures, it is useful to consider the structure as a generative context for large language models to identify potential learning paths. While much work is being done to find optimal PLP algorithms, the consistency and flow of the microlearning content itself remains an open question.

Acknowledgments

The authors would like to thank the DoD for partially sponsoring the current research.

ORCID

Paolo J. Singh - `https://orcid.org/0009000498830666`
Ralucca Gera - `https://orcid.org/0000000219562084`

References

[1] 3blue1brown, Essence of linear algebra (2016), `https://www.youtube.com/watch?v=fNk_zzaMoSs&list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab`, accessed May 8, 2024. [Online Videos]. Available: https://www.youtube.com/watch?v=fnkzzaMoSs&list=PLZHQOBOWTQDPD3MizzM2xVFitgF8hEab.

[2] Bazett, T., Linear algebra [full course] (2018), `https://www.youtube.com/watch?v=ZKUqtErZCiU&list=PLHXZ9OQGMqxfUl0tcqPNTJsb7R6BqSLo6`, accessed Jan. 23, 2024. [Online
Videos]. Available: https://www.youtube.com/watch?v=ZKUqtErZCiU&list= PLHXZ9OQGMqxfUl0tcqPNTJsb7R6BqSLo6.

[3] Boyd, S. and Vandenberghe, L., *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares* (Cambridge University Press, 2018).

[4] Brehm, K., Linear algebra [entire course] (2019), `https://www.youtube.com/playlist?list=PLl-gb0E4MII03hiCrZa7YqxUMEeEPmZqK`, accessed Jan. 23, 2024. [Online Videos]. Available: https://www.youtube.com/playlist?list=PLl-gb0E4MII03hiCrZa7YqxUMEeEPmZqK.

[5] Buehler, M. J., Accelerating scientific discovery with generative knowledge extraction, graph-based representation, and multimodal intelligent graph reasoning, *arXiv preprint arXiv:2403.11996* (2024).

[6] Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., and Jatowt, A., Yake! keyword extraction from single documents using multiple local features, *Information Sciences* **509** (2020) 257–289.

[7] Chandler, P. and Sweller, J., Cognitive load theory and the format of instruction, *Cognition and Instruction* **8** (1991) 293–332, doi:10.1207/s1532690xci0804_2, `https://doi.org/10.1207/s1532690xci0804_2`.

[8] Cheng, Y., A learning path recommendation method for knowledge graph of professional courses, in *2022 IEEE 22nd International Conference on Software Quality, Reliability, and Security Companion (QRS-C)* (2022), pp. 469–476, doi:10.1109/QRS-C57518.2022.00076.

[9] Gera, R., Bartolf, D., Tick, S., and Saxena, A., Chunk learning: A tool that supports personalized education, in *Proceedings of the 15th International Conference on Educational Data Mining* (2022), p. 743.

[10] Gera, R., Reith, M., Bartolf, D., Tick, S., and Saxena, A., A vision of personalized education using network science: Co-developing a dynamic network of knowledge (in press.).

[11] Gera, R., Reith, M., Bartolf, D., Tick, S., Singh, P., and Mochocki, S., Personalizing learning through collaborative networks of knowledge, *Education Applications & Developments X* (2025).

[12] Grootendorst, M., KeyBERT: Minimal keyword extraction with BERT. (2020), doi: 10.5281/zenodo.4461265, `https://doi.org/10.5281/zenodo.4461265`.

[13] Gu Yijun, X. T., Study on keyword extraction with lda and textrank combination, *Data Analysis and Knowledge Discovery* **30** (2014) 41, doi:10.11925/infotech.1003-3513.2014.07.06, https://manu44.magtech.com.cn/Jwk_infotech_wk3/EN/abstract/article_3923.shtml.

[14] Gundersen, G., Linear algebra blogs (2022), https://gregorygundersen.com/blog/tags/la/, accessed Jan. 23, 2024. [Online]. Available: https://gregorygundersen.com/blog/tags/la/.

[15] Hamblin, J., Linear algebra lectures (2018), https://www.youtube.com/playlist?list=PLNr8B4XHL5kGDHOrU4IeI6QNuZHur4F86, accessed Mar. 12, 2024. [Online Videos]. Available: https://www.youtube.com/playlist?list=PLNr8B4XHL5kGDHOrU4IeI6QNuZHur4F86.

[16] Hefferon, J., *Linear Algebra* (Jim Hefferon, 2020), accessed May 6, 2024. [Self-published online]. Available: https://www.openintro.org/go?id=linalg4&referrer=hefferon.net.

[17] Hernandez, N., 'netflix for learning' doesn't work. here's why. (2024), https://360learning.com/blog/netflix-for-learning/.

[18] Hou, B., Lin, Y., Li, Y., Fang, C., Li, C., and Wang, X., Kg-plppm: A knowledge graph-based personal learning path planning method used in online learning, *Electronics* **14** (2025) 255.

[19] Issa, B., Jasser, M. B., Chua, H. N., and Hamzah, M., A comparative study on embedding models for keyword extraction using keybert method, in *2023 IEEE 13th International Conference on System Engineering and Technology (ICSET)* (IEEE, 2023), pp. 40–45.

[20] Jugran, S., Kumar, A., Tyagi, B. S., and Anand, V., Extractive automatic text summarization using spacy in python & nlp, in *2021 International conference on advance computing and innovative technologies in engineering (ICACITE)* (IEEE, 2021), pp. 582–585.

[21] Leon, S. J., *Linear algebra with applications, 8th edition* (Pearson Prentice Hall Upper Saddle River, NJ, 2010).

[22] Mayer, R. E., *The Cambridge handbook of multimedia learning* (Cambridge university press, 2005).

[23] Mayer, R. E., The past, present, and future of the cognitive theory of multimedia learning, *Educational Psychology Review* **36** (2024) 8.

[24] Mochocki, S., Reith, M., Borghetti, B., Peterson, G., Jasper, J., and Merkle, L., Computational complexity and AI approaches for the personalized learning path problem (in press.).

[25] Nabizadeh, A. H., Leal, J. P., Rafsanjani, H. N., and Shah, R. R., Learning path personalization and recommendation methods: A survey of the state-of-the-art, *Expert Systems with Applications* **159** (2020) 113596, doi:https://doi.org/10.1016/j.eswa.2020.113596, https://www.sciencedirect.com/science/article/pii/S0957417420304206.

[26] Nadim, M., Akopian, D., and Matamoros, A., A comparative assessment of unsupervised keyword extraction tools, *IEEE Access* (2023).

[27] Panagos, A., Linear algebra example problems (2015), https://www.youtube.com/watch?v=Fg6B01vEN3U&list=PLdciPPorsHuk3Hp7QPPAtTkpW0o1UXQB6, accessed Mar. 12, 2024. [Online Videos]. Available: https://www.youtube.com/watch?v=Fg6B01vEN3U&list= PLdciPPorsHuk3Hp7QPPAtTkpW0o1UXQB6.

[28] Qian, Y., Jia, C., and Liu, Y., Bert-based text keyword extraction, *Journal of Physics: Conference Series* **1992** (2021) 042077, doi:10.1088/1742-6596/1992/4/

042077, `https://dx.doi.org/10.1088/1742-6596/1992/4/042077`.

[29] Rose, S., Engel, D., Cramer, N., and Cowley, W., Automatic keyword extraction from individual documents, *Text mining: applications and theory* (2010) 1–20.

[30] Saunders, F., Gellen, S., Stannard, J., McAllister-Gibson, C., Simmons, L., and Gibson, A., Educating the netflix generation: Evaluating the impact of teaching videos across a science and engineering faculty, in *Engaging Engineering Education: SEFI 48th Annual Conference Proceedings* (SEFI (European Society for Engineering Education), 2020), pp. 431–440.

[31] Singh, P. J. and Gera, R., Designing a dynamic network of knowledge of microlearning content, in *2024 World Conference on Complex Systems (WCCS)* (2024), pp. 1–8, doi:10.1109/WCCS62745.2024.10765575.

[32] Strang, G., *Introduction to Linear Algebra*, 4th edn. (Wellesley-Cambridge Press, Wellesley, MA, 2009).

[33] Tae,
J., Keyword extraction with bert (2021), https://jaketae.github.io/study/keyword-extraction/.

[34] Watts, D. J. and Strogatz, S. H., Collective dynamics of 'small-world'networks, *Nature* **393** (1998) 440–442.

[35] Wrath of Math, Linear algebra (2023), `https://www.youtube.com/watch?v=oXMPQ-6YnGA&list=PLztBpqftvzxWT5z53AxSqkSaWDhAeToDG`, accessed May 6, 2024. [Online Videos]. Available: https://www.youtube.com/watch?v= oXMPQ-6YnGA list= PLztBpqftvzxWT5z53AxSqkSaWDhAeToDG.

[36] Young, S., The netflixification of education (2024), https://youngscholarsacademy.org/blog/the-netflixification-of-education.

[37] Zhang, J., Gao, M., and Zhang, J., The learning behaviours of dropouts in moocs: A collective attention network perspective, *Computers & education* **167** (2021) 104189.