K-Means Clustering for Mind Map Generation for Visual Learners

Subhadra Vadlamannati
Mercer Island High School

Abstract
Many visual learners and English language learners desire automated, personalizable, and topical visual aids in conjunction with written text to fully understand material. Unfortunately, there is a lack of methods available to transform vital academic text into a visual format for such learners, with current methods primarily focusing on text-to-text generation, e.g., summarization. In this study I present MindTree, a k-Means clustering-based algorithm to automatically generate informative mind maps for any length of textbook or article text. MindTree picks out the key topics from long and complicated texts and organizes them in a hierarchical and logical mind map, drawing connections between related topics. MindTree's algorithm additionally finds latent, or "hidden" topics within the text that may not be explicitly mentioned as well.

Keywords
Mind maps, natural language processing, clustering, k-means

Related Work
In the literature on educational psychology one can find numerous references to learning styles or learning preferences and how to take advantage of them to drive better outcomes for a student. The VARK (Visual, Aural, Read/Write, and Kinesthetic) method is one of the most popular categorizations of learning styles (Fleming and Mills, 1992). According to the VARK learning style categorization, Visual learners prefer explanation of concepts diagrammatically or through pictures while Aural learners prefer to listen to what their teachers say. Persons whose learning style aligns with R (Read/Write) learn best when they read text material and take notes by themselves and Kinesthetic learners learn best by doing things with their hands. The VARK questionnaire has been used by various educational institutions since its inception (Fleming & Bonwell, 2019) and co-developer Neil Fleming also proposed techniques for teaching and test taking that caters to the preferences of learners (Fleming, 1995). Similarly, Felder and Silverman (Felder and Silverman, 1988) described how learners utilize different mechanisms to

send, receive, and process information. They developed the Felder-Silverman learning style model, which rates students' learning styles according to a scale that consists of four dimensions: Sensing-Intuitive, Visual-Verbal, Active-Reflective, and Sequential-Global.

While some researchers challenge the usefulness of adopting teaching methodologies to match learners' preferences, citing lack of evidence in educational outcomes (Pashler et al., 2008; Kratzig et al. 2006; Riener & Willingham, 2010), there is a general consensus that learning styles do exist, and that providing material to learners in formats that can help them engage better with content is desirable (Coffield et al., 2004). Visual learning is a common learning style where individuals better understand and retain information presented in visual formats like pictures, diagrams, flowcharts, timelines, films and demonstrations (Shabiralyani et al., 2015). Research shows that visual learning can improve outcomes for students, as concepts and relationships are made clearer to them with visual elements (Carifio & Perla, 2007; Ibrahim et al., 2012). However, exclusively visual teaching methods may not be ideal for all students. Hence, a mix of modalities is likely most effective for a general student population (Coffield et al., 2004).

The use of visual aids to facilitate learning is not a new concept, and there are several existing methods available for transforming academic text into more digestible formats for learners with reading disabilities (McNamara, 2001; Schnotz, 2002). One such method is text summarization, which automatically generates a shortened version of a given text by identifying the most important sentences or phrases (Nenkova, A., & McKeown, K., 2011). While text summarization has shown promising results for improving accessibility and learning outcomes for learners with reading disabilities, it primarily focuses on generating text-based summaries and may not be as appealing for visual learners. Additionally, summarization does not take into account words or concepts outside of the corpus, which limits the potential of improving the overall understanding of the concepts being taught in the text.

Another approach that has been explored in the literature is the use of concept maps or mind maps to present information in a hierarchical and visual format (Buzan, 2005). Concept maps present information in a hierarchical format with directional labeled arrows emphasizing relationships between concepts and are more formal in nature. Mind maps are very similar to concept maps but often incorporate images, colors, and keywords to enhance visual appeal and aid in memory retention (Guerrero, 2023). They are designed to be visually engaging and to stimulate creative thinking. Some studies have found that concept maps can improve learning outcomes for students with disabilities such as dyslexia and attention-deficit/hyperactivity disorder (ADHD) by helping them organize and make connections between different pieces of information (Novak & Gowin, 1984; Sutton et al., 2013, Chan et al.

1990). Others have shown that computer assisted concept maps help accelerate English Language Learning (Liu, P. L., Chen, C. J., & Chang, Y. J., 2010).

The creation of mind maps is typically an arduous manual task for an instructor. Shyian-Shyong et al. created a concept map that a student could leverage by correlating test scores and concept relationships (Shian-Shyong et al., 2007). Their starting point was a list of concepts and student test scores in quizzes that tested these concepts. Shao et al. took this concept further by automatically generating the list of concepts from the tests rather than starting from a predetermined list using clustering algorithms and association rules mining (Shao et al., 2018). Based on my literature survey, existing methods for automatically generating concept maps or mind maps may be limited in their ability to handle long and complex texts. The proposed approach in this paper, MindTree, aims to address these limitations by automatically generating informative mind maps for any length of textbook or article text. By selecting the key topics and subtopics and organizing them in a hierarchical and logical manner, MindTree provides a more accessible and visually engaging format for learners.

Large Language Models (LLMs) have recently proven to be very effective in interpreting and processing a corpus of text and images to perform a broad variety of tasks including text simplification, answering related questions etc (Kasneci et al., 2023). However, current LLMs are restricted by the amount of context they can hold in memory and the associated computational costs, as well as the lack of reliability in their output. This limits their ability to process large amounts of text.

The proposed approach in this study offers a novel and potentially more effective solution for transforming long form academic text into a digestible format for visual learners.

Introduction

The paper provides a detailed analysis on how k-Means clustering can be paired with word embedding technologies to create less repetitive and more relevant mind maps for visual learners. Additionally, MindTree improves upon existing methodologies of generating concept maps which are limited in their ability to handle long and complex texts. MindTree automatically generates informative mind maps for any length of textbook or article text. MindTree picks out the key topics from lengthy and complicated texts and organizes them in a hierarchical and logical mind map, drawing connections between related topics and including brief descriptions of the places in the text where these topics come up.

The proposed approach offers a novel and potentially more effective solution by employing a k-Means clustering algorithm (Lloyd, 1982) to identify the main topics of a text corpus. K-Means clustering was selected over hierarchical clustering algorithms for sub-topic generation

because of its ability to efficiently cluster large datasets (Sun et al. 2008). As new content is continually added to the system, the volume of data needing to be grouped into subtopics grows rapidly. K-Means is able to quickly assign data points to predefined clusters versus the slower, sequential clustering of hierarchical algorithms (Kobren et al. 2017). Additionally, k-Means is less prone to propagating errors that can occur when using hierarchical clustering as by reassigning data points across iterations, k-Means can self-correct early bad clustering. This allows for more consistent, stable clusters - an important consideration for generating coherent, meaningful sub-topics. Finally, because our approach is user-focused, allowing them to select the k parameter (the number of topics) allows direct control over the desired sub-topic granularity, as opposed to other density-based clustering algorithms.

The technical details of the algorithm are discussed in the methodology section. All code and data associated with this paper can be viewed and downloaded using the links in Appendix C.

Preprocessing Steps

The first step in the process of topic identification is to eliminate stop words—irrelevant parts of speech, such as prepositional phrases and conjunctions—from the text corpus. To do this, a list of stop words was downloaded from Punkt and the Natural Language Toolkit (NLTK) and removed from the text during preprocessing. The next step was to identify words in the text corpus that were similar to one another. The GloVe word embeddings approach, a method that embeds words in a vector space to unveil relationships between them (Pennington, J., Socher, R., & Manning, C., 2014) was used for this.

However, this is just one version of the implementation, and various other word embeddings like BERT (Devlin et al. 2019), roBERTa (Liu et al., 2019), or ELMO (Peters et al., 2018) can also be used. These vectors are pre-trained to predict other words in a given context, relying on the number of times those words co-occur with each other. Therefore, GloVe models are used to determine the words that are the most relevant in each text.

When passing the words of the corpus into the GloVe model, Python scripts were used to remove words that do not appear in the entire GloVe vocabulary. Subsequently, unnecessary text and links were filtered out from the generated output. The remaining words were sorted alphabetically, and their corresponding embeddings were used to create a matrix of embeddings for the corpus. Note that in my approach there are two "dictionaries": one of which includes all the corpus words and their corresponding GloVe embeddings, and the other of which is the entire GloVe dictionary, which includes (almost) all of the words in the English language and their k-dimensional embeddings. These dictionaries will subsequently be referred to as 'corpus dictionary' and 'GloVe dictionary,' respectively.

The GloVe model comes in four different sizes, which correspond to the dimensionalities of the matrices the words are embedded in, (50d, 100d, 200d, 300d). I studied the effect of dimensionality on the performance of the model and presented the results in the Results section.

Clustering with k-Means Algorithm

The k-Means algorithm (Lloyd, 1982) from Scikit-learn was used to cluster the words in the corpus dictionary based on the closeness of their embedding values. To see an explanation of why k-Means was selected over other clustering algorithms, see the "Alternate Approaches" section where a review of other approaches tried as part of this study are presented. The k-Means algorithm was initiated at least ten times for each test to ensure stability amongst the clusters generated. Then, words were weighted by the number of times they appeared in the text for both GloVe and corpus dictionary testing. This weighting was especially important because past approaches did not consider the frequency at which words occurred in the text. As MindTree is created for topic summarization, it's important to have relevant topics in the text. We ran the model with three cluster sizes to evaluate its impact on the output quality.

Identification of Main and Latent Topics

Vectors for the centers of the clusters were then computed using k-Means and the *n* closest word vectors in the corpus dictionary (these form the main topics) and *n* closest words in the entire GloVe dictionary (these form the "latent" or hidden topics) to the center were found. Variations of the Minkowski p-norm distance metrics were used to find the closest main topics to the cluster centers (see the results section for an analysis on this).

The Minkowski p-norm distance is calculated using the following formula:

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

When $p = 1$, this distance is called the L1 norm or the Manhattan distance between two points. When $p = 2$, this distance is considered the L2 norm, or the Euclidean distance. As p increases, the Minkowski p-norm converges to the maximum value of the absolute values of the elements in the vector (the Chebyshev distance, or the infinity norm). This is because the largest element in the vector dominates the sum and determines the norm, as the other elements become increasingly insignificant with increasing p. In other words, the norm becomes increasingly sensitive to the largest element as p increases.

As finding the distances for various values of p manually would be

computationally intensive, I used a KDTree by recursively partitioning the data along different dimensions. A KDTree is a binary search tree where *k* refers to the dimensionality of the data stored in the tree (2D, 3D, etc). At each node in the tree, I split the current subset of word vectors along one of the dimensions, alternating between dimensions as I went further down the tree. Once constructed, this KDTree allowed me to quickly find the n closest vectors (words) to any query vector, such as a cluster center. The function returned a list containing the n words closest to each cluster center according to Euclidean distance.

Outside of the corpus itself, I also further refined this process by considering a use case where there may be hidden, or "latent" topics within a given corpus. For example, a text could be a cookbook, but never mention the word "cooking" once, even though it's an important main topic! This could be problematic, because we're choosing main topics from words that are strictly within the corpus itself. I solved this by additionally adding all of the words in the word embedding corpus to my graph. I then calculated the distances between all of the words in the glove embedding space to the words in the corpus itself using a KDTree, as it is a very efficient data structure to determine the *n* closest neighbors in a Euclidean space. Then, I chose the top *n* words that had the smallest distances to the words in the corpus, to get a more general idea of the main topics.

Next, a list of subtopics underneath each main topic is generated in a similar fashion by finding the nearest neighbors to each of the main topic words within a certain threshold of similarity. The similarity of the subtopic words to the main topics was computed using various similarity functions, like cosine similarity and Euclidean distance between the points in the [50, 100, 200, 300]-dimensional space, after running all the words in the corpus through a GloVe embedding space. The closest words were then extracted for each main topic, and then used as the list of the subtopics.

After generating the lists of subtopics and main topics, the resulting list was then pruned using NLTK to delete duplicate subtopics and subtopics that were the same as the main topics. I then visualized the data using Python functionality with the NetworkX package, creating edges between subtopics under the same main topics.

Algorithm

To see a more complete and written form of the algorithm, see Appendix B.

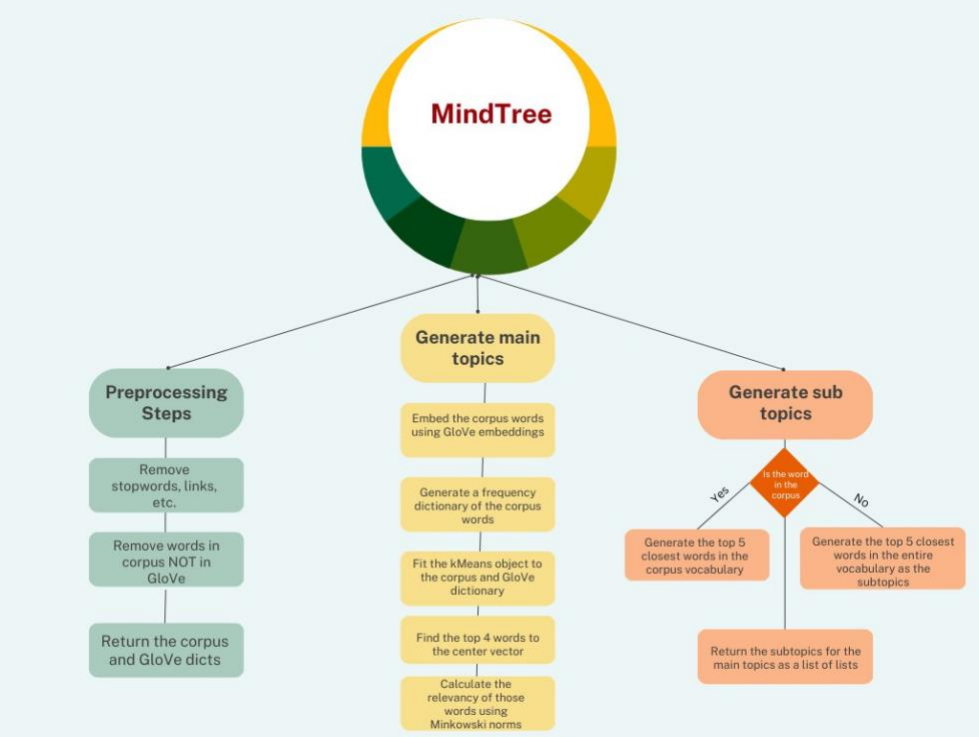


FIGURE 1. Diagrammatic Representation of the Algorithm

Results

The main metric used to evaluate model performance is the averaged distance of the main topics from the cluster center (noted as "Averaged Distance" on the graphs). In some cases, the inertia, a measure of the cluster quality obtained from the k-Means algorithm, was also included. (Inertia, which is used to evaluate the performance of the k-Means algorithm, measures the sum of squared distances between each point and its nearest cluster centroid.) The objective of k-Means is to minimize inertia, which means that lower inertia values are better (Arthur, D., & Vassilvitskii, S., 2007). A low inertia value indicates that the clusters are tightly packed, and the points within each cluster are similar to each other. Similarly, a lower value of the average distance additionally indicates that the main topics were of better objective quality.

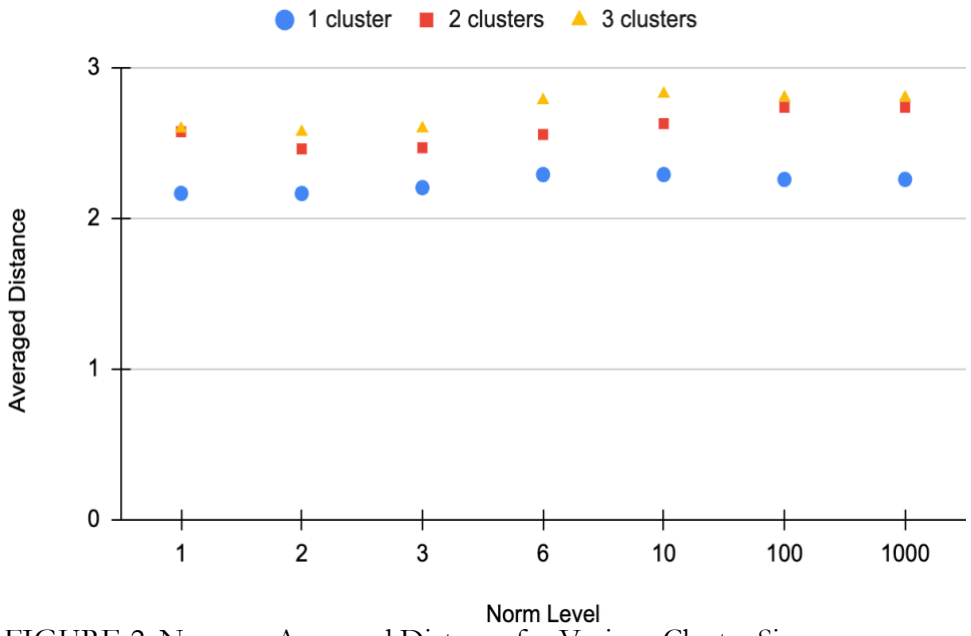


FIGURE 2. Norm vs Averaged Distance for Various Cluster Sizes

In this chart, it is evident that the Minowski *p*-norm affects the averaged distance from the cluster center. We see a general trend across the norm levels, with the highest performance (lowest average distance) consistently reached at norm level 2 and the performance plateauing around norm level 10-1000. Recall that when $p=2$, it corresponds with a measure of Euclidean distance.

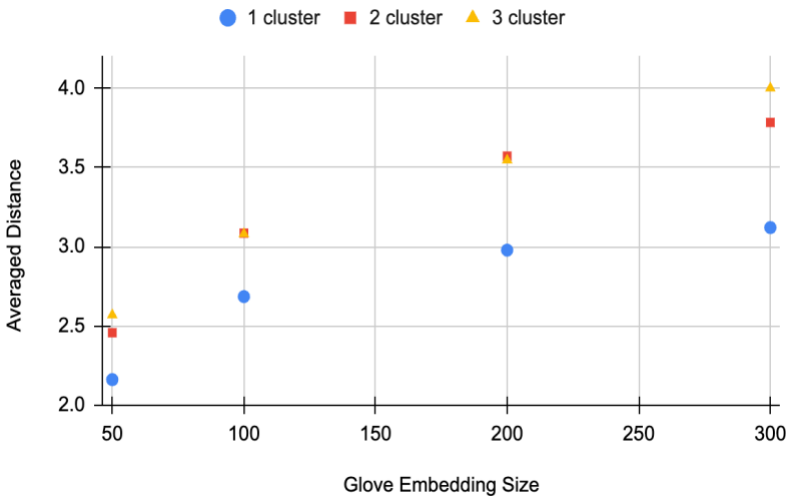
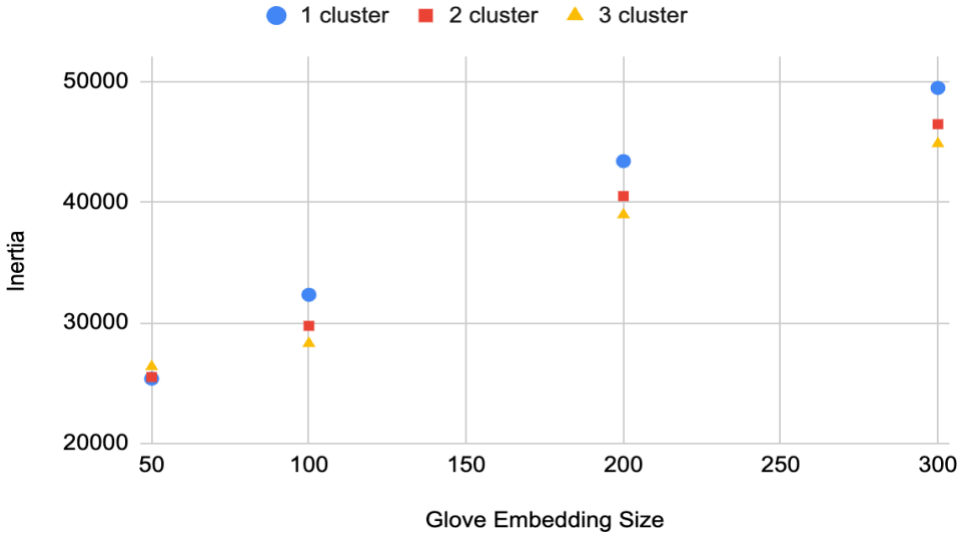


FIGURE 3. Glove Embedding Size vs Averaged Distance for Various Clusters

FIGURE 4. Glove Embedding Size vs Inertia for Various Clusters



These graphs demonstrate the relation between GloVe embedding size and the averaged distance/inertia from the cluster center. Increasing the dimensionality of GloVe embedding size can cause an increase in k-Means inertia because it can make the clustering task more difficult and increase the variability of the data. K-Means clustering is based on minimizing the sum of squared distances between the data points and their assigned cluster centers, known as the "inertia" of the clustering.

When the dimensionality of the embedding space is increased, the space becomes sparser and the number of possible combinations of features or attributes increases. This can make it more difficult for k-Means to find meaningful clusters and can cause it to converge to suboptimal solutions with higher inertia values.

Moreover, increasing the dimensionality of the embedding space can also increase the amount of noise or irrelevant features in the data, making it harder to identify the true underlying clusters. This can result in k-Means assigning points to incorrect clusters, leading to higher inertia values.

Alternate Approaches

Before eventually landing on the final approach for MindTree, various precursors were tested and evaluated to determine the best possible way to generate mind maps using a clustering approach.

Main Topics

As a first pass approach, the text was initially preprocessed to remove unnecessary words and then found the top four most frequently occurring words in the corpus as the main topics. This gave me a base idea of the most common words, and a somewhat accurate description of the main

topics, but this approach was very naive and did not consider the relations between words. Additionally, the most common words in a given text are not necessarily the main topics of the text.

After this approach, I implemented HLDA (Hierarchical Latent Dirichlet Allocation) in order to determine efficiency and effectiveness. HLDA is an extension of LDA (Latent Dirichlet Allocation) that models a hierarchical structure of topics (Blei, D. M., Ng, A. Y., & Jordan, M. I., 2003). In HLDA, each topic can have multiple subtopics, and these subtopics can have their own subtopics, forming a hierarchical tree structure (Blei, D. M., & Jordan, M. I., 2003). The main idea behind HLDA is to allow for a more flexible and fine-grained representation of the topics in a corpus, which can be useful for tasks such as document classification, topic modeling, and information retrieval. However, HLDA was not suitable for my task, and often generated duplicated topics in an unstructured format (see Appendix A for detailed results). Additionally, hierarchical classifications do not consider the relation between subtopics that could be related to multiple main topics, so it would have not been suitable later down the line when drawing connections between the main topics and subtopics. Furthermore, HLDA often results in extreme redundancy, generating several "main topics" that often have no distinction from one another, as found in my results as well as previous research (Yoshida et al. 2023). Another key issue with HLDA is that it relies on unsupervised learning algorithms to infer the topic hierarchy from the data (Griffiths et al., 2004). As a result, the hierarchy extracted may not align well with human judgment or the actual semantic relationships between topics.

After these two approaches, I eventually decided to use clustering as my final approach to generate main topics, with a mix between explicit topics from the corpus and latent topics found from the overall GloVe embeddings.

Sub-Topics

As a first pass approach, I simply decided to look at the top 20 most commonly occurring words in the corpus that were outside of the main topics and connected them to the main topics at random. This gave me a valid baseline to look at the generally common words of the corpus, but the methodology was extremely simple and was akin to a word cloud.

Conclusion

The proposed approach, MindTree, provides a novel solution to improve accessibility of academic text for visual learners by automatically generating informative mind maps for any length of textbook or article text. MindTree uses k-Means clustering algorithm on top of GLoVe embeddings to identify key topics and subtopics and organizes them in a hierarchical and logical manner, drawing connections between related topics. Through this study we established the relation between key

parameters in GLoVe and k-Means models to the performance of mind map generation. Further investigations could examine the integration of MindTree with additional word embedding methods such as BERT or ELMO. A logical succession to the current study would be to conduct human subject research analyzing learning outcomes and gauging learner preferences for assorted MindTree yields under varied parameter settings.

Science Examples

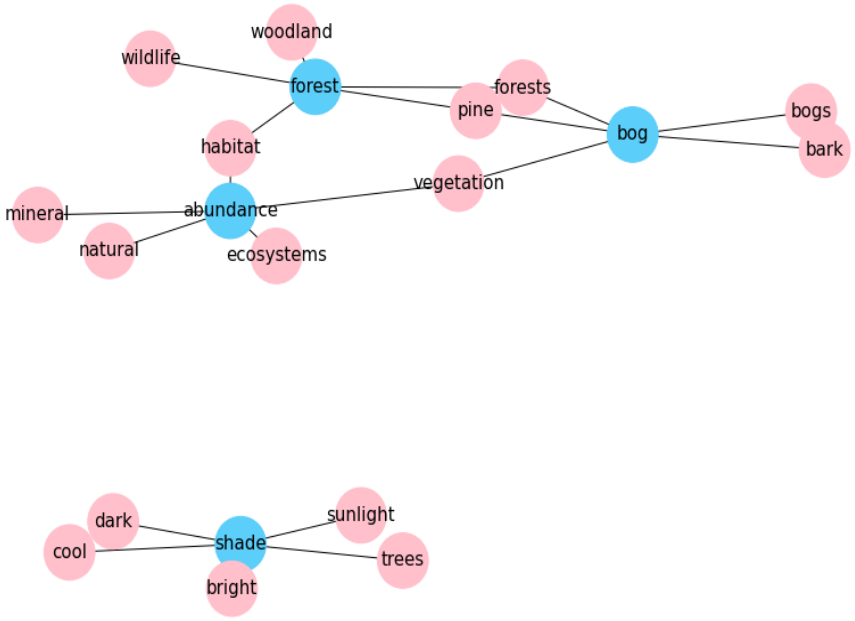


FIGURE 5. Science Mind-Map Without Latent Topics

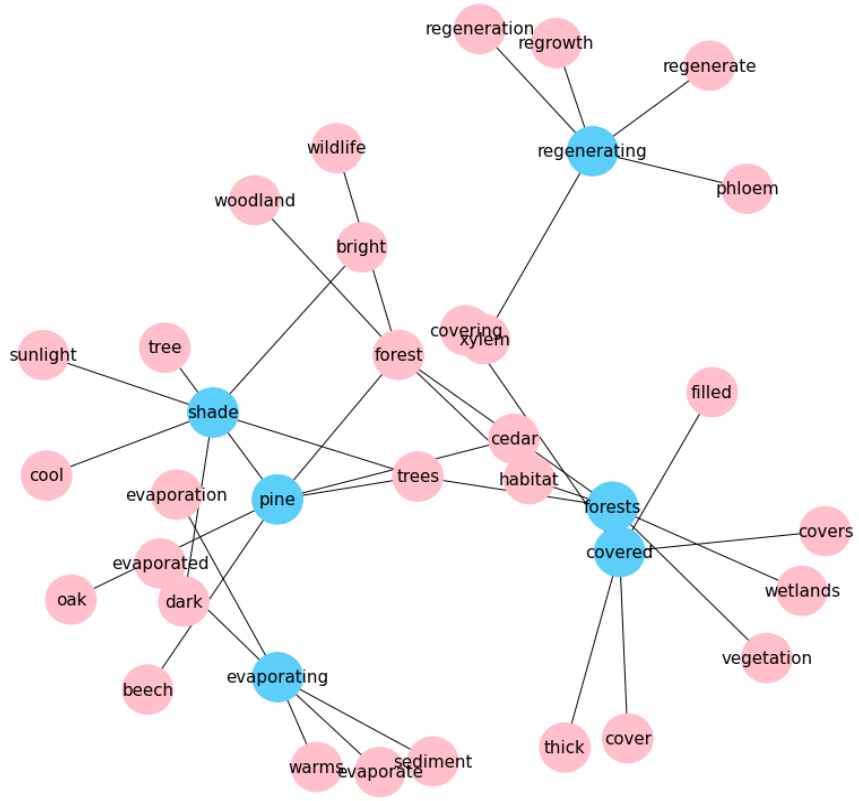


FIGURE 6. Science Mind-Map With Latent Topics

History Examples

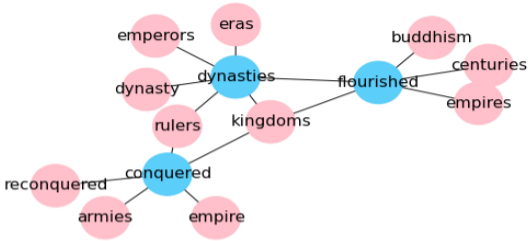


FIGURE 7. History Mind-Map Without Latent Topics

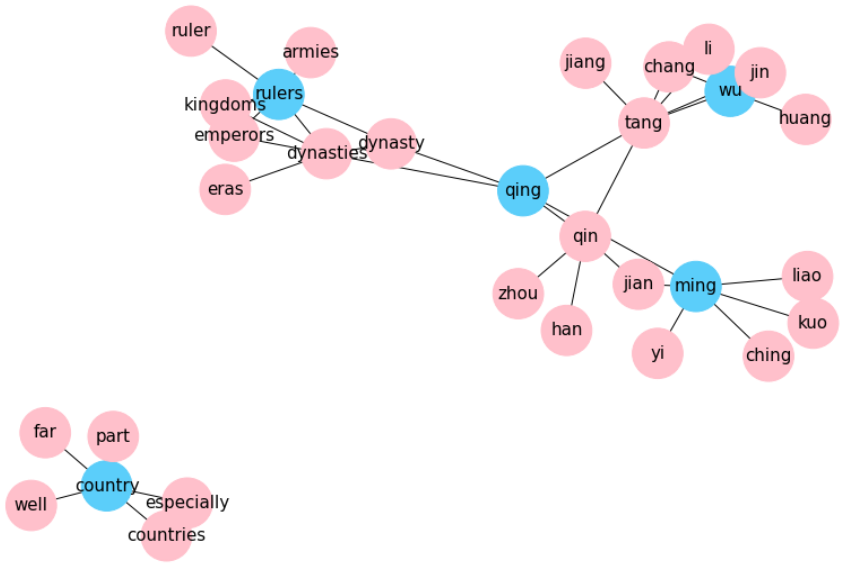


FIGURE 8. History Mind-Map With Latent Topics

References

Fleming, N.D. & Mills, C. (1992). Helping Students Understand How They Learn. The Teaching Professor, Vol. 7 No. 4, Magma Publications, Madison, Wisconsin, US

Fleming, N.D & Bonwell, C. (2019). How Do I Learn Best?: a student's guide to improved learning

Fleming, N.D; (1995), I'm different; not dumb. Modes of presentation (VARK) in the tertiary classroom, in Zelmer,A., (ed.) Research and Development in Higher Education, Proceedings of the 1995 Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA),HERDSA, Volume 18, pp. 308 - 313

Pashler, H., McDaniel, M., Rohrer, D., & Bjork, R. (2008). Learning Styles: Concepts and Evidence. Psychological Science in the Public Interest, 9(3), 105-119. https://doi.org/10.1111/j.1539-6053.2009.01038.x

Krätzig, G. P., & Arbuthnott, K. D. (2006). Perceptual learning style and learning proficiency: A test of the hypothesis

Riener, C., & Willingham, D. (2010). The myth of learning styles

Coffield, & Ecclestone, Kathryn & Moseley, & Hall, Elaine. (2004). Learning styles and pedagogy in post 16 education: a critical and systematic review.

McNamara, D. (2001). Cesare Cornoldi and Jane Oakhill, eds., - Reading comprehension difficulties: Processes and intervention. Hillsdale, NJ: Erlbaum, 1996. (xxiii + 365 pp). Journal of Pragmatics. 33. DOI: 10.1016/S0378-2166(00)00026-6. https://www.researchgate.net/publication/263415272_Cesare_Cornoldi_and_Jane_Oakhill_eds_-_Reading_comprehension_difficulties_Processes_and_intervention_Hillsdale_NJ_Erlbaum_1996_xxiii_365_pp

Schnotz, W. (2002). Towards an Integrated View of Learning From Text and Visual Displays. Educational Psychology Review. 14. 101-120. 10.1023/A:1013136727916. https://www.researchgate.net/profile/Wolfgang-Schnotz/publication/226902934_Towards_an_Integrated_View_of_Learning_From_Text_and_Visual_Displays/links/5cd967c4a6fdccc9dda762e6/Towards-an-Integrated-View-of-Learning-From-Text-and-Visual-Displays.pdf

Nenkova, A., & McKeown, K. (2011). Automatic summarization. Foundations and Trends® in Information Retrieval, 5(2-3), 103-233. https://www.cis.upenn.edu/~nenkova/1500000015-Nenkova.pdf

Buzan, T. (2005). Mind map handbook. Great Britain: Thorsons.

Guerrero, J. M. (2023). Chapter 1 - What is mind mapping. In Guerrero, J. M., Mind Mapping and Artificial Intelligence; Academic Press, 2023; pp 1-29.https://doi.org/10.1016/B978-0-12-820119-0.00006-6.

Novak, J. D., & Gowin, D. B. (1984). Learning how to learn. Cambridge

University Press.
Sutton, C., & Spooner, F. (2013). The effect of concept mapping on reading comprehension skills of students with high-incidence disabilities. Learning Disabilities Research & Practice, 28(3), 108-117.
Chan, L. K. S., Cole, P. G., & Morris, J. N. (1990). Effects of Instruction in the Use of a Visual-Imagery Strategy on the Reading-Comprehension Competence of Disabled and Average Readers. Learning Disability Quarterly, 13(1), 2–11. https://doi.org/10.2307/1510388
Liu, P. L., Chen, C. J., & Chang, Y. J. (2010). Effects of a computer-assisted concept mapping learning strategy on EFL college students' English reading comprehension. Learning Disabilities Research & Practice. Computers & Education 54, 436–445 http://59.64.36.71/lc/koPage/ko4212/1/436-445Effects%20of%20a%20computer-assisted%20concept%20mapping%20learning%20strategy%20on%20EFL%20college%20students%E2%80%99%20English%20reading%20comprehension%20%20.pdf
Shian-Shyong Tseng, Pei-Chi Sue, Jun-Ming Su, Jui-Feng Weng, Wen-Nung Tsai. (2007). A new approach for constructing the concept map, Computers & Education. https://doi.org/10.1016/j.compedu.2005.11.020.
Shao, Z., Li, Y., Wang, X., Zhao, X., Guo, Y. (2018). Research on a New Automatic Generation Algorithm of Concept Map Based on Text Clustering and Association Rules Mining. In: Huang, DS., Bevilacqua, V., Premaratne, P., Gupta, P. (eds) Intelligent Computing Theories and Application. ICIC 2018. Lecture Notes in Computer Science(), vol 10954. Springer, Cham. https://doi.org/10.1007/978-3-319-95930-6_44.
Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, Gjergji Kasneci (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences. Volume 103, 2023. https://doi.org/10.1016/j.lindif.2023.102274
S. Lloyd, "Least squares quantization in PCM," in IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129-137, March 1982. https://doi.org/10.1109/TIT.1982.1056489
Kobren, A., Monath, N., Krishnamurthy, A., & McCallum, A. (2017, August). A hierarchical algorithm for extreme clustering. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 255-264).

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics. https://aclanthology.org/D14-1162.pdf

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. https://doi.org/10.48550/arXiv.1907.11692

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics. https://aclanthology.org/N18-1202.pdf

Nurrokhim, M. F., Riza, L. S., & Rasim. "Generating Mind Map from an Article Using Machine Learning." Journal of Physics: Conference Series, vol. 1280, no. 3, IOP Publishing Ltd, 2019, article 032023. doi: 10.1088/1742-6596/1280/3/032023. https://iopscience.iop.org/article/10.1088/1742-6596/1280/3/032023/pdf

Hu, Mengting, et al. "Efficient Mind-Map Generation via Sequence-to-Graph and Reinforced Graph Refinement." *arXiv preprint arXiv:2109.02457* (2021).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022. https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

Blei, D. M., & Jordan, M. I. (2003). Hierarchical topic models and the nested Chinese restaurant process. In Proceedings of the 16th International Conference on Neural Information Processing Systems (NIPS 2003), 17(4), 473-480. https://arxiv.org/pdf/0710.0845.pdf

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 1027-1035. https://theory.stanford.edu/~sergei/papers/kMeansPP-soda.pdf

Yoshida T, Hisano R, Ohnishi T (2023) Gaussian hierarchical latent

Dirichlet allocation: Bringing polysemy back. PLOS ONE 18(7): e0288274. https://doi.org/10.1371/journal.pone.0288274

Jigui Sun, Jie Liu and Lianyu Zhao, "Clustering algorithms Research", Journal of Software, vol. 19, no. 1, pp. 48-61, January 2008.

Appendix

A: HLDA Generations

[{'topic_id': 0, 'words': ['emoji', 'new', 'propos', 'peopl', 'mean', '']},
{'topic_id': 0, 'words': ['emoji', 'make',
'text', 'keyboard', 'releas', '']}, {'topic_id': 0, 'words': ['new', 'text', 'say',
'commun', 'current', '']}, {'topic_id':
0, 'words': ['emoji', 'propos', 'get', 'consortium', 'updat', '']}, {'topic_id': 0,
'words': ['new', 'heart', 'thi',
'mean', 'peopl', '']}, {'topic_id': 0, 'words': ['emoji', 'use', 'sure', 'current',
'need', '']}, {'topic_id': 0, 'words':
['new', 'heart', 'get', 'propos', 'consortium', '']}, {'topic_id': 0, 'words':
['new', 'emoji', 'heart', 'make', 'mean',
'']}, {'topic_id': 0, 'words': ['emoji', 'heart', 'thi', 'use', 'face', '']},
{'topic_id': 0, 'words': ['heart', 'make', 'thi', 'releas', 'get', '']}]

B: Algorithm Write-up

- Start with original article text
    - Preprocess article text, removing stopwords, links, and strings with numbers
- Generate main topics
    - Embed the corpus words using GloVe embeddings
    - Generate a frequency dictionary of the corpus words
    - Fit the k-Means object to the corpus or GloVe embeddings
        - Pass in the frequency dictionary of the corpus words to the k-Means algorithm
    - Generate a vector of the centers of the k-Means algorithm
    - Find the top 4 words to the center vector
        - In the GloVe embeddings
        - In the corpus embeddings
    - If there are relevant words in the GloVe embeddings *not* present in the corpus embeddings list…
        - Calculate the relevancy of those words using cosine similarity.
        - If they meet a certain threshold (e.g. 95% similarity)…
            - Add them to the list of main topics
    - Output a list of the main topics
- Generate sub topics
    - For each word in the main topics…
        - Generate the word vector for the word via GloVe embeddings
        - If the word is in the corpus…
            - Generate the top 5 closest words in the corpus as the subtopics

        - Else…
            - Generate the top 5 closest words in the entire vocabulary as the subtopics
    - Return a list of lists, with each sublist containing the 5 subtopics for each main topic in the text

C: Code and Data Links

China Document Dataset
Science Document Dataset
Code