Fine-grained Main Ideas Extraction and Clustering of Online Course Reviews

Chenghao Xiao(✉)[0000−0001−7623−8232], Lei shi[0000−0001−7119−3207], Alexandra Cristea[0000−0002−1454−8822], Zhaoxing Li[0000−0003−3560−3461], and Ziqi Pan[0000−0001−8867−1009]

Department of Computer Science, Durham University, UK
{chenghao.xiao,lei.shi,alexandra.i.cristea,zhaoxing.li2,ziqi.pan2}@durham.ac.uk

Abstract. Online course reviews have been an essential way in which course providers could get insights into students' perceptions about the course quality, especially in the context of massive open online courses (MOOCs), where it is hard for both parties to get further interaction. Analyzing online course reviews is thus an inevitable part for course providers towards the improvement of course quality and the structuring of future courses. However, reading through the often-time thousands of comments and extracting key ideas is not efficient and will potentially incur non-coverage of some important ideas. In this work, we propose a *key idea extractor* that is based on fine-grained aspect-level semantic units from comments, powered by different variations of state-of-the-art pre-trained language models (PLMs). Our approach differs from both previous topic modeling and keyword extraction methods, which lies in: First, we aim to not only eliminate the heavy reliance on human intervention and statistical characteristics that traditional topic models like LDA are based on, but also to overcome the coarse granularity of state-of-the-art topic models like top2vec. Second, different from previous keyword extraction methods, we do not extract keywords to summarize each comment, which we argue is not necessarily helpful for human readers to grasp key ideas at the course level. Instead, we cluster the ideas and concerns that have been most expressed throughout the whole course, without relying on the verbatimness of students' wording. We show that this method provides high and stable *coverage* of students' ideas.

Keywords: MOOC · Key ideas extraction · Language models · Automated pipeline

1 Introduction

Identifying key ideas from course reviews is an essential way of obtaining insights into students' learning experience, especially in the context of massive open online courses (MOOCs), where it is hard for students and instructors to have further interaction [16, 1].

However, reading through the often-time thousands of comments and extracting key ideas is not efficient and will potentially incur non-coverage of some

important ideas [12]. This can be due to both aspects of feedback being forgotten throughout the reading due to readers' limited working memory [3] or even being ignored because of readers' perceptions and confirmation bias [8, 30, 10].

In this paper, we propose an *automated key idea extraction pipeline* that can be run with minimal human intervention and interpretation, with the purpose of efficiently covering as many as the most expressed ideas in massive corpus of students' reviews in online courses. While it is not necessarily feasible for course providers to sift through the often-time thousands of comments, it is advisable that they should attend to certain important aspects of concerns that have been most expressed in the comments [23]. We propose such an automated method, facilitated by state-of-the-art NLP algorithms. Moreover, we conduct experiments on the robustness of dimensionality reduction of text embeddings before applying hierarchical clustering, providing empirical and theoretical insights into the selection of this parameter and its impact on efficient *coverage* of ideas, for future users of this method. We also introduce a weighted centroid to select representative phrases for each cluster, and a flexible usage of a coefficient value to attend to under-represented ideas in a cluster.

We argue that for the efficient coverage of the most important aspect-level ideas expressed in massive corpus of online course comments, traditional keyword extraction and topic modeling methods might not work well, which is because the former only studies reducing the size of text instead of the number of documents [32, 25], while the latter suffers from coarse granularity [14]. Facilitated by the state-of-the-art, our research provides a *fine-grained key idea extraction approach* to bridge this gap, while being wording-agnostic.

2 Related Work

Before static embedding methods such as word2vec [28] and contextualized language models such as BERT [9] and RoBERTa [20] were introduced, tasks of natural language processing (NLP) had been strongly relying on statistical characteristics extracted from language. For example, in the field of topic modelling, since Blei *et al.* [5] proposed Latent Dirichlet Allocation (LDA), this probabilistic model had been a major algorithm in topic modeling, whose limitations lie in both the unknown numbers of topic clusters that have to be decided by human through exhaustive experiments, and its statistical discrimination over rare but significant topic keywords - as topical words are not always frequently mentioned at the level of each document. On the other hand, recent development and deeper understanding in word embedding brings state-of-the-art topic modeling algorithms like top2vec [2] and BERTopic [11] to the table, which yield better results and require less human intervention than traditional topic models.

In the field of education, Miller [29] proposed leveraging BERT and k-means for extractive text summarization of lectures. They claimed that many approaches in the field used dated algorithms that produced sub-par results and relied on manual tuning. This provided a good pipeline to address similar extractive summarization scenarios, but we argue that the text representation that

BERT itself produces, typically used in research by directly taking the [cls] (classification) token of BERT, is sub-optimal [31], and while being *de facto*, k-means is not necessarily a panacea for high-dimensional clustering. In this paper, we thus replace them by Sentence Transformers [31] and HDBSCAN [7]. Masala *et al.* [25] proposed extracting and clustering main ideas from student feedback based on a pipeline of KeyBERT-based keyword extraction and K-means context clustering. They first extracted top 10 keywords for each course, then clustered different contexts that mentioned these words. To the best of our knowledge, however, their keyword extraction component still relied on the verbatimness of the wording. This limitation is also addressed in our approach, through clustering directly on high-quality embeddings of fine-grained text.

We argue that for a course level analysis, starting from *top-n* verbatim keywords is not always a good approach as 1-gram keywords extracted might be mostly nouns which are over-general and hard to interpret on their own, while 2-gram or over 2-gram keywords strongly rely on the verbatimness of students' phrasing. For example, "well-organized" and "well-structured" convey close meanings that might otherwise be interpreted by course providers as one aspect. Considering two semantically similar words separately might affect the statistical significance of both of them, leading to both being ignored from *top-n*. By contrast, both being selected in *top-n* might affect the diversity of aspects included, as this prevents other important words from being selected. Therefore, we propose directly applying clustering on the level of fine-grained text, by breaking down each comment into chunks of long phrases or short sentences, which we argue is a good semantic unit that carries semantically interpretable meanings (as opposed to fragmented keywords), while mostly staying in only one aspect (as opposed to document level that covers different aspects which can twist the text embeddings and therefore affect the effectiveness of the clustering).

In line with our intuition, Luo and Litman [23] proposed summarizing students' responses at phrase level, and introduced *student coverage* as an evaluation of the method, based on the assumption that concepts mentioned by more students should receive more attention from the instructor, which chimes in with the purpose of our method. In this paper, we aim to realize these objectives with state-of-the-art algorithms. Moreover, on top of covering concepts that are semantically expressed the most, we also explore using outlier scores in a cluster, to 'listen' to under-represented phrases, as will be described in section 4.1. In summary, we build upon the state of the art in text summarization and natural language processing to propose a novel pipeline, which also overcomes their limitations, and takes into account *readability*, *relevance*, and *coverage* [21].

3 Method

3.1 Corpus

We used the Coursera Course Reviews dataset[1], which comprises over 140k reviews of 1,835 courses, along with their corresponding ratings. For experiments

[1] https://www.kaggle.com/septa97/100k-courseras-course-reviews-dataset

and demonstration of our proposed method, we focus on the field of machine learning and data science, which we filtered by the inclusion of either "machine", or both "data" and "science" in the course names, yielding 12 machine learning- and data science-related courses after we removed "machine design" which is irrelevant in this context. The filtering of data results in 9,980 unique comments with 246,290 tokens.

3.2 Pre-processing

What distinguishes our approach from other topic modeling methods is that we do not apply topic modeling at the entire document level, but instead at a fine-grained level, which requires that we first break each document into long phrases or short sentences. Although our method is mostly based on the state of the arts, the pre-processing step is inspired by a traditional method, RAKE [32], which observed that a document can be parsed into candidate keywords by breaking them down at delimiters and certain stopwords. We further customized our stopword list, removing as many useful words from the list as possible (e.g., opinionated ones like *don't, not, shouldn't*) to prevent them from being deleted during parsing. However, we find that what really matters is that a document is parsed into short sentences using delimiters. The stopwords that further break each short sentence into long phrases are less important, as a word that does not appear in a phrase will appear in the adjacent one anyway, preserving the information to be encoded and processed in later clustering.

3.3 Pipeline

We adopt similar pipeline described in [2], while making a few important adjustments to overcome its coarse granularity. First, as our method is based on fine-grained aspect-level linguistic units after pre-processing, the default doc2vec [18] would intuitively be insufficient to learn phrase embeddings that are semantically meaningful [17]. We thus replace this encoding method by two latest Sentence Transformers [31] models. Second, we find that the optimal number of embedding dimensions to reduce to at phrase level, before applying hierarchical clustering, is different from that on document level, and propose the method to empirically customize this hyperparameter through *coverage*. Lastly, we propose using a *local weighted centroid* to select the most representative phrase, so that readers can cover a big portion of the most important ideas through reading only a few phrases representing the largest clusters. Our pipeline is shown in Fig 1.

Originally introduced using BERT as a backbone, Sentence Transformers (ST) have been shown to yield very effective representations of text when applied to similarity comparison, clustering, and information retrieval tasks [31], as opposed to previous approaches - taking [cls] token of BERT - which yielded suboptimal semantic representation. It was not until recently that new ST methods that yielded significant performance boost based on newer Transformer models have been released. In this work, we empirically evaluate two ST models: one based on MPNet [33] that provides the best sentence embedding and semantic

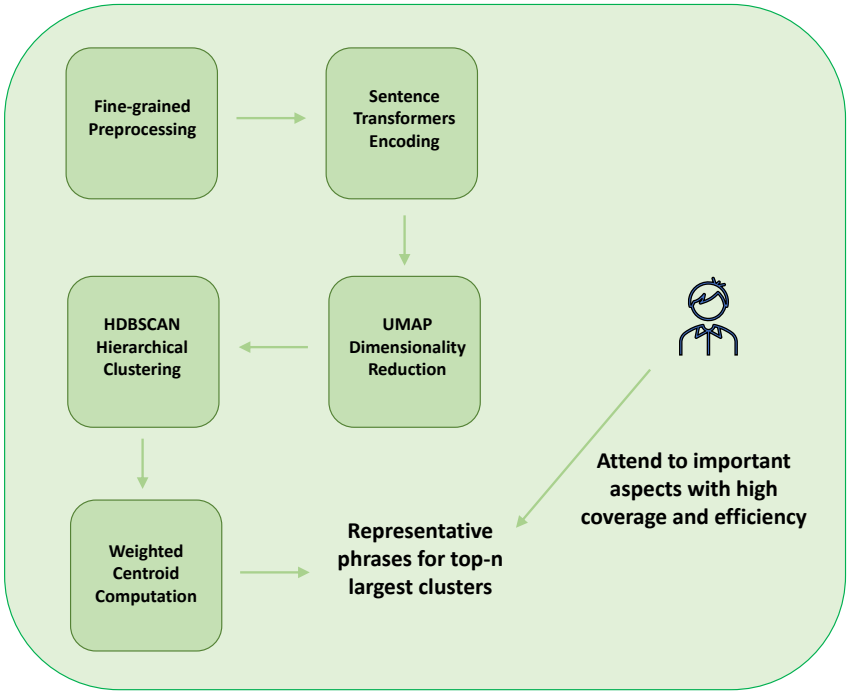

Fig. 1. Automated fine-grained key ideas extraction: pipeline and mechanism

search performance up to the date of this paper, *all-mpnet-base-v2*[2]; and the other being *all-MiniLM-L6-v2*[3], which is based on a distilled model MiniLM [35]. The latter achieves comparable results while being 5 times faster than the former. Thus we believe it is worth evaluating as an alternative for user cases in student feedback reading that require faster encoding of text embedding and output of following analysis results.

After encoding, the data further goes through dimensionality reduction and clustering. For dimensionality reduction, we use UMAP [27], which preserves better global structure of data [2] compared to t-SNE [24] as reflected in distances between clusters. For clustering, we use HDBSCAN [7, 26], a robust hierarchical density-based clustering method which we use to replace the *de facto* k-means used in prior research, whose limitations lie in assumptions of inclusion of all instances and spherical shapes of clusters.

Eqn. (1) demonstrates the way we propose to find the centroid phrase CP in a cluster that shares the highest cosine similarity with our defined weighted centroid embedding CE as computed in Eqn. (2).

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[3] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

$$CP = \arg\max_k \left(\frac{CE \cdot E_k}{\|CE\| \, \|E_k\|}\right), \tag{1}$$

where,

$$CE = \frac{\sum_1^K [(1 - \alpha O_k) E_k]}{K - \alpha \sum_1^K O_k}, \tag{2}$$

where K denotes the number of phrases in a cluster, k denotes the k^{th} entry in that cluster, while O_k and E_k respectively denote its corresponding outlier score and embedding. We further introduce α as a flexible coefficient, to adjust the influence of the outlier score to the computation of the weighted centroid, where by default we set it to 1. However, we did find that α can be used flexibly, to output representative phrases that are far away from the vanilla centroid to obtain *unique ideas*, when it is set to a higher value. Although in [2], it is suggested that at document level, a weighted centroid will not make much of a difference to the vanilla centroid, we find that it does make a difference at the fine-grained aspect level, especially when applied to lower dimensional data after dimensionality reduction using UMAP.

4 Results and Discussions

4.1 Key Ideas Analysis

Results from our methods identified a range of important aspects that have been expressed in reviews of our selected machine learning- and data science-related courses (Table 1). We present the *top*-10 largest clusters from two results: respectively running with dimensionality reduction to 5 and 10 dimensions. [2] observed a best dimension reduction number of 5 for document level embedding clustering, while under our high-granularity context, we observed that it provides the best results when this hyperparameter is set to around 10, as will be further demonstrated through in-depth *coverage* study and empirical interpretation in section 4.2. However, it is shown that our pipeline provides robust performance and a great overlap of topics under these two settings. Notably, we observed that the topics that are not overlapped in the top 10 clusters under these two results can be further found in the rest of their top 15 clusters.

Based on results shown in Table 1, we could easily get an overall idea that in machine learning- and data science-related courses, students express most concerns about: 1) programming language used in the courses, 2) math background required and covered, 3) content structured in the courses such as videos, quizzes and programming exercises, 4) the way and the degree to which instructors successfully convey the essence of the algorithms. These findings are in line with previous research using different methodologies [15, 36, 6, 22, 19]. In courses related to machine learning, students may encounter different kinds of difficulties [19]. For example, students lacking solid mathematical background struggled more with understanding math-related content in the course [15] and developing

computational thinking [36]. Bolliger [6] suggested that an online course can be affected by various factors, such as instructor variables, technical issues, and interactivity, which can be interpreted through combinations of our identified topics as well. For example, identified clusters about Andrew Ng's way of teaching explain instructor variables, technical issues could involve codes debugging issues caused by difficulty of using programming languages like Octave, as expressed in comments, while interactivity could be supported by quizzes and exercises in this context. Lu *et al.* [22] found that flow experience significantly contributed to MOOC satisfaction, which relies on the students' not being distracted and frustrated during learning. In our case, we argue that ideas expressed in the topics identified, such as difficulty in using programming languages and following math intuition due to insufficient background knowledge, could account for no or low flow experience.

However, to acquire deeper insights into opinions in clusters, readers could further go into each cluster to see the well-represented and under-represented phrases in each cluster, through the usage of outlier scores. We provide the example of *Octave* to give a brief idea on how it works (see Table 2).

Table 1. Top-10 largest clusters by clustering on 5 and 10 dimensional embeddings, represented by their weighted centroid phrases. Cluster labels in bold show clusters that have been overlapped in the *top*-10 and thus *double-validated* by two outputs.

	5-d clustering		10-d clustering	
top-n	Weighted centroid	Cluster label (interpreted)	Weighted centroid	Cluster label (interpreted)
1	the essence and purpose of the algorithms	**algorithm**	the guts of the algorithms	**algorithm**
2	the instructor uses Octave	**Octave**	best Coursera course I've ever taken	**best Coursera course**
3	made simple [..] understandable MATLAB	**Matlab**	be afraid of using Octave	**Octave**
4	The best explanation of principles and ideas behind Machine Learning	Machine Learning	Matlab hands-on exercises permit a deeper understanding of the algorithms	**Matlab**
5	I enjoyed very much	**enjoyed**	I enjoyed	**enjoyed**
6	do not want to watch the videos	videos	The exercises	**exercises**
7	allows me to follow the quizzes	quizzes	very well constructed	course structure
8	a statistics background [..]	**math background**	a complete online course	**good MOOC**
9	Great MOOC	**good MOOC**	liked the way Andrew taught us the concept	way of teaching
10	great exercises	**exercises**	being proficient with Linear Algebra	**math background**

Table 2. Sampled elements in an exemplar on the most representative cluster about *Octave* (the 3^{th} largest cluster under 10-d clustering). The Example in bold is the centroid phrase computed by a weighted centroid. We present both well-represented and under-represented examples, showcased by outlier scores

Cluster Examples	outlier score
more chances to practice algorithm prototyping in Octave	0.0
do not like Octave somehow and prefer the Python approach coming	0.0
Octave [...] instead of more modern languages	0.0
force the students to use Octave	0.0
be afraid of using Octave	0.0
in OCTAVE instead of popular languages like R	0.24
But I consistently felt unprepared for applying it in Octave	0.26
alongwith a great introduction to Octave	0.26
awesome assignment submission tool via Octave	0.34
I enjoyed learning Octave and performing the weekly homework	0.40

4.2 Robustness of Dimensionality Reduction

Recent research has indicated that the similarity measures of contextual word embeddings, as opposed to static word embeddings, have been dominated by a small number of what is referred to as the "rogue" dimensions [34]. Furthermore, the typically 768 or similar pre-defined dimensional space of BERT-facilitated embedding methods makes density-based clustering inefficient. It is also not difficult to intuitively picture how hierarchical density-based clustering will tend to only put phrases that are extremely close to one another into the same cluster, due to the vast geographical space created by high dimensions. This tends to, thereby, make the clusters no more than a collection of some almost semantically identical, or even worse, verbally identical phrases. Reducing dimensionality before apply clustering, however, greatly compresses the semantic space, making clusters that are otherwise separated in high dimensional space have to "accept" one another and merge to a large cluster.

We speculate that on a high granularity level, the number of embedding dimensions used in hierarchical clustering can be interpreted and utilized as a strong indicator of reader's tolerance towards semantic difference, and therefore the acceptance of larger cluster with semantically diverse, yet aspect-wise similar expressions. In general, a lower number of dimensions indicates higher tolerance, and leads to more otherwise separated clusters merged into one.

The results are demonstrated in Fig. 2, where we compare a few representative dimension numbers through their corresponding sizes of the biggest clusters, and average sizes of the *top*-10 clusters, which is extremely significant for measure of *coverage* of ideas. Coverage to more frequently expressed students' ideas through fewer *first* n clusters can lead to instructors' getting important opinion aspects from comments with higher efficiency. We empirically find that the coverage stays stable when dimension is reduced to around 5-20, yielding similar *top-n* cluster phrases and sizes, while 10 dimensions provide the best results

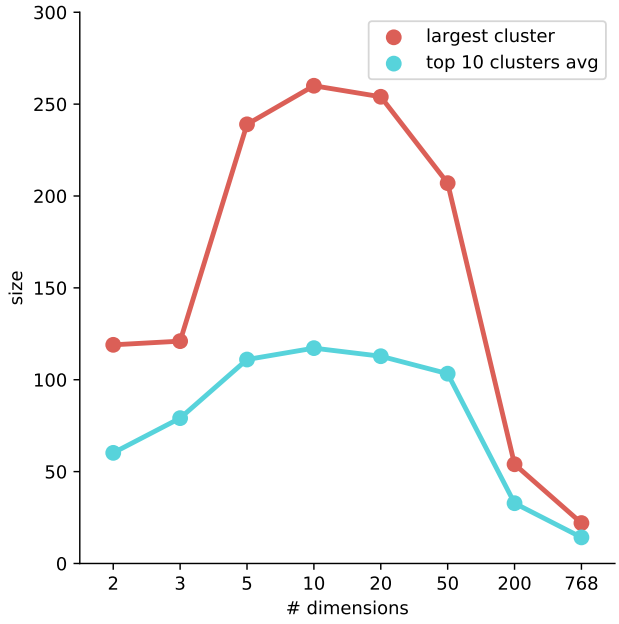

Fig. 2. Reducing to different numbers of dimensions before applying hierarchical clustering yields *coverage* of ideas through *top-n* clusters with different efficiency

in our case, indicated by both cluster quality and *coverage*. We suggest future studies to start from this range and find a customized value for their specific courses.

4.3 A glimpse of the algorithm's wording-agnostics

In this section, we briefly demonstrate the superiority of our algorithm in terms of how agnostic it is towards different wordings to convey similar meanings that are by their nature supposed to be clustered into one aspect.

Using the 7^{th} largest cluster under 10-dimensional clustering as an example, we randomly select 13 phrases out of the 71 phrases in that cluster (see Table. 3). It is clearly shown that wordings of reviews in this cluster are highly diverse, while our approach facilitates to understand them as conveying close meanings, albeit phrases in this cluster consist of no verbatim wordings.

5 Conclusion

In this work, we propose a novel pipeline for online course providers to receive insights into students' opinions, concerns and experience from online courses, and thus be able to attend to the most important aspects of comments efficiently.

Table 3. Diverse wordings of reviews in a sample cluster, whose weighted centroid phrase is computed as *Very well constructed*, while phrases in that cluster include almost no verbatim forms of the same wording

Cluster Examples	outlier score
thoughtfully made	0.0
very well put together	0.0
carefully created	0.0
Very well constructed	0.0
Well constructed with good practicals	0.04
really well crafted	0.27
Very well designed with a clear focus	0.43
also well-built with a lot of warm-support and encouragement	0.52
Exceptionally well arranged	0.54
very polished and it makes participating easy and smooth	0.54
meticulously curated	0.56
Excellent selfcontained	0.57
badly designed	0.58

We empirically present the effectiveness of combining state-of-the-art embedding encoders, dimensionality reduction, and clustering algorithm on the *fine-grained aspect level*. We also present an empirical study on the robustness of embedding dimension selection that could optimize runtime without losing much semantic information of aspect-level linguistic units, and being more wording-agnostic for higher efficiency of student *coverage*.

Structured on the state-of-the-art, our proposed method contributes to achieving high coverage of important ideas, while being agnostic to students' wordings. We plan to deploy this pipeline in real-life classes and create teaching assistant-generated gold-standard summaries [23] for evaluation of algorithm-generated idea coverage against human readers' perceptions. Notably, our approach aims to extract information from course reviews, while we highlight that intrinsic biases in course evaluations do exist [13, 4]. We encourage researchers in this field to build upon our method to detect and mitigate biases in course evaluations.

We also envision two possible directions of future work. First, we envision fine-tuning Sentence Transformer models with domain-specific text datasets, to make domain-specific aspects more positionally accurate in the semantic space, for facilitating better evaluations of courses in highly specialized domains. While in our case, reviews related to machine learning and data science courses might not be highly different from day-to-day writing, highly domain-specific ones like medical courses might require language models' deeper understanding about the field, to extract accurate clusters. Second, we envision efforts in human-AI interaction: if deploying our proposed method in industrial settings, we encourage to enable users (course providers, instructors, etc.) to accept, reject, and merge clusters. Such data can then be recorded and used to learn a feature-based activation layer [9, 31] for the system to provide more personalized cluster recommendations.

References

1. Anderson, A., Huttenlocher, D., Kleinberg, J., Leskovec, J.: Engaging with massive online courses. In: Proceedings of the 23rd international conference on World wide web. pp. 687–698 (2014)
2. Angelov, D.: Top2vec: Distributed representations of topics. arXiv preprint arXiv:2008.09470 (2020)
3. Baddeley, A.D.: Working memory and reading. In: Processing of visible language, pp. 355–370. Springer (1979)
4. Baker, R., Dee, T., Evans, B., John, J.: Bias in online classes: Evidence from a field experiment. Economics of Education Review **88**, 102259 (2022)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. the Journal of machine Learning research **3**, 993–1022 (2003)
6. Bolliger, D.U.: Key factors for determining student satisfaction in online courses. International Journal on E-learning **3**(1), 61–67 (2004)
7. Campello, R.J., Moulavi, D., Zimek, A., Sander, J.: Hierarchical density estimates for data clustering, visualization, and outlier detection. ACM Transactions on Knowledge Discovery from Data (TKDD) **10**(1), 1–51 (2015)
8. Del Vicario, M., Scala, A., Caldarelli, G., Stanley, H.E., Quattrociocchi, W.: Modeling confirmation bias and polarization. Scientific reports **7**(1), 1–9 (2017)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186 (2019)
10. Frost, P., Casey, B., Griffin, K., Raymundo, L., Farrell, C., Carrigan, R.: The influence of confirmation bias on memory and source monitoring. The Journal of general psychology **142**(4), 238–252 (2015)
11. Grootendorst, M.: Bertopic: leveraging bert and c-tf-idf to create easily interpretable topics (2020). URL https://doi. org/10.5281/zenodo **4381785**
12. Hasan, K.S., Ng, V.: 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). pp. 1262–1273 (2014)
13. Hassan, T.: On bias in social reviews of university courses. In: Companion Publication of the 10th ACM Conference on Web Science. pp. 11–14 (2019)
14. Jiang, D., Shi, L., Lian, R., Wu, H.: Latent topic embedding. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 2689–2698 (2016)
15. Kim, S.W.: Kepler vs newton: Teaching programming and math to almost all-majors in a single classroom. In: 2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE). pp. 956–957 (2020). https://doi.org/10.1109/TALE48869.2020.9368332
16. Kop, R.: The challenges to connectivist learning on open online networks: Learning experiences during a massive open online course. The International Review of Research in Open and Distributed Learning **12**, 19–38 (2011)
17. Lau, J.H., Baldwin, T.: An empirical evaluation of doc2vec with practical insights into document embedding generation. arXiv preprint arXiv:1607.05368 (2016)
18. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196. PMLR (2014)

19. Lishinski, A., Yadav, A., Enbody, R.: Students' emotional reactions to programming projects in introduction to programming: Measurement approach and influence on learning outcomes. In: Proceedings of the 2017 ACM Conference on International Computing Education Research. pp. 30–38 (2017)
20. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
21. Liu, Z.: Research on keyword extraction using document topical structure. Beijing: Tsinghua University (2011)
22. Lu, Y., Wang, B., Lu, Y.: Understanding key drivers of mooc satisfaction and continuance intention to use. Journal of Electronic Commerce Research **20**(2) (2019)
23. Luo, W., Litman, D.: Summarizing student responses to reflection prompts. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1955–1960 (2015)
24. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
25. Masala, M., Ruseti, S., Dascalu, M., Dobre, C.: Extracting and clustering main ideas from student feedback using language models. In: International Conference on Artificial Intelligence in Education. pp. 282–292. Springer (2021)
26. McInnes, L., Healy, J., Astels, S.: hdbscan: Hierarchical density based clustering. Journal of Open Source Software **2**(11), 205 (2017)
27. McInnes, L., Healy, J., Saul, N., Großberger, L.: Umap: Uniform manifold approximation and projection. Journal of Open Source Software **3**(29) (2018)
28. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
29. Miller, D.: Leveraging bert for extractive text summarization on lectures. arXiv preprint arXiv:1906.04165 (2019)
30. Oswald, M.E., Grosjean, S.: Confirmation bias. Cognitive illusions: A handbook on fallacies and biases in thinking, judgement and memory **79** (2004)
31. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992 (2019)
32. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual documents. Text mining: applications and theory **1**, 1–20 (2010)
33. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: Mpnet: Masked and permuted pre-training for language understanding. Advances in Neural Information Processing Systems **33**, 16857–16867 (2020)
34. Timkey, W., van Schijndel, M.: All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 4527–4546 (2021)
35. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Advances in Neural Information Processing Systems **33**, 5776–5788 (2020)
36. Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., Wilensky, U.: Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology **25**(1), 127–147 (2016)