Clustering Students' Short Text Reflections: A Software Engineering Course Case Study

Mohsen Dorodchi
University of North Carolina at Charlotte
Charlotte, NC 28223
Mohsen.Dorodchi@uncc.edu

Alexandria Benedict
University of North Carolina at Charlotte
Charlotte, NC 28223
abenedi4@uncc.edu

Erfan Al-Hossami
University of North Carolina at Charlotte
Charlotte, NC 28223
ealhossa@uncc.edu

Andrew Quinn
University of North Carolina at Charlotte
Charlotte, NC 28223
aquinn16@uncc.edu

Sandra Wiktor
University of North Carolina at Charlotte
Charlotte, NC 28223
swiktor@uncc.edu

Aileen Benedict
University of North Carolina at Charlotte
Charlotte, NC 28223
abenedi3@uncc.edu

Mohammadali Fallahian
University of North Carolina at Charlotte
Charlotte, NC 28223
mfallahi@uncc.edu

ABSTRACT
Student reflections can provide instructors with beneficial knowledge regarding their progress in the course, what challenges they are facing, and how the instructor can provide more effectively to the students' needs. Reading every student reflection, however, can be a time-consuming task that may affect the instructor's ability to efficiently address student needs in a timely manner. In this research, we explore the use of clustering and sorting of student reflections to shorten reading time while maintaining a comprehensive understanding of the reflection content. We obtain student reflections from a software engineering course. Next, we generate transformer-based sentence embeddings and then cluster the reflections using K-Means. Lastly, we sort the reflections based on the distance of each reflection from its cluster center. We conduct a small-scale user study with the course's Teaching Assistants and provide promising preliminary results showing a significant increase in reading time efficiency without sacrificing understanding.

Keywords
Natural Language Processing, Student Reflections, Clustering

1. INTRODUCTION
Reflections are an effective way for instructors to detect what their students may be struggling with throughout their courses, gain a perspective on students' impressions of course content, and track their overall progress [9]. However, in order to utilize these benefits to the fullest, instructors would need to manually read through each individual reflection. Manually analyzing reflections can be overwhelming for an instructor, especially in large classroom settings, where timely feedback is needed to address students' possible concerns. Machine learning and knowledge discovery-based methods have been used to assist educators in understanding and helping students [14, 1, 20]. Unsupervised methods in natural language processing (NLP) such as topic modeling, have been used to automatically extract topics from student reflective journals [5]. However, they fall short when it comes to short text, typically around a sentence in length, such as tweets. Recent research has utilized K-means clustering along with transformer-based sentence-embeddings to automatically extract topics from tweets [12, 2]. K-means clustering is often supplemented with a representation of text. Representations can include statistical-learnt representations such as term frequency-inverse document (TF-IDF) [13], neural-learnt representations also known as word embeddings (e.g. Word2Vec [19], Glove [22]), and more recently representations computed from large pretrained transformer deep learning models.

Transformers are deep learning models following the architecture proposed by Vaswani et al. [27]. These models often undergo an unsupervised pretraining on a massive text corpus to create an initial version of the network later fine-tuned for more specific tasks in process called *transfer learning*. Pretrained transformers such as BERT [6], RoBERTa [16], and GPT-3 [4], have achieved state of the art in many natural language processing tasks. Some of these tasks include detecting positive and uplifting discussion on social media (e.g. [17]), determining answers to questions given a passage of text (e.g. [28]), summarizing text (e.g. [15]), and estimating semantic similarity between sentences. For this reason, we select a transformer-based language model to create a semantic representation of student responses.

In this research, we implement an approach using k-means clustering from the scikit-learn library and utilize transformer-based sentence embeddings. We evaluate our approach in a preliminary user study, observing the time taken for teaching assistants to read and analyze student reflections.

2. DATASET

Course. The data used in our research was collected from an undergraduate software engineering course based on the active learning course model proposed in [7]. The number of students enrolled in the course was 108 students. Modules are organized based on the concepts being taught and typically spanned across approximately one week. The course contained 11 modules in total with the topics listed in Table 1. Following the active learning course model presented

Table 1: Concepts taught within each module of the software engineering course.

Module	Topic(s)
1	Introduction to Software Engineering and Agile Development Methods
2	Introduction to Requirements and Modeling
3	Requirement Analysis and Modeling
4	Architecture and Modeling
5	Data Flow Diagrams, Context Diagrams, and UML Diagrams
6	Use Case Diagrams & Extracting Requirements
7	Cloud based Software engineering, Testing, Object-oriented Design Pattern
8	Microservices, feasibility
9	Reliable programming
10	Final exam
11	Final project

in Dorodchi et al. [10, 7], each module is typically divided into multiple scaffolds: prep-work to complete before class including reading assignments and videos to watch, in-class activities, post-lecture activities, including assignments and labs, and a reflection at the end of the module. Labs are more challenging assignments provided to students which require hands-on coding. These lab activities are typically divided into multiple parts. There are a total of 4 labs in this course, with the first lab beginning in Module 2 and the last lab being introduced in Module 8.

Data Collection. A survey questionnaire was provided to students within *Canvas*, the University's Learning Management System (LMS), at the end of each module to allow students to reflect on their learning and challenges. We refer to student responses of this questionnaire as **student reflections** throughout this work. The questions asked of students were:

1. On a scale of 1 to 5, with 5 being Very Active and 1 being Not Active, how engaged would you rate your group this week?

2. What was your biggest challenge this past week? This can include in-class activities, assignments, prep work, studying, time management, motivation, and so on.

3. How can you address the challenge you mentioned above? What can you do to overcome this challenge for next time?

For the purpose of this research, we focused solely on the students' responses to question 2, as this question was free-response and would provide unique responses for the clustering process.

Dataset Statistics. We used two different module reflections from the software engineering course throughout this study: Module 7 reflections and Module 8 reflections. Table 2 showcases our descriptive statistics of our collected student reflection responses corpora. The selected module reflections were comparable in size. Firstly, the response rates to the Module 7 and Module 8 reflections are 94 or 87.0% and 89 or 82.4% responses out of 108 total students after preprocessing respectively. Moreover, the total word counts were 1866, 1390 for Module 7 and Module 8 reflections respectively. We also observe that most student reflections contained between a sentence or two on average in both module reflections. Furthermore, we note that most reflections in our corpus were around a sentence in length.

Table 2: Descriptive statistics of the Module 7 and Module 8 reflections collected from the undergraduate software engineering course.

Module Reflection	Reflection 7	Reflection 8
Responses (%)	94 (87.0%)	89 (82.4%)
Avg. Word Count	19.4	15.4
Avg. Number of Sentences	1.4	1.3
Avg. Words per Sentence	14.0	11.5
Total Words	1866	1390

3. APPROACH

Our overall approach is illustrated in Figure 1. First we collect data from an undergraduate course with 108 students. This is described in more detail in section 2. Then, we perform preprocessing on the data using natural language processing (section 3.1). Next, we generate sentence embeddings (section 3.2), cluster those embeddings (section 3.3), and sort the reflections based on clusters for TA's to view (section 3.4).

3.1 Preprocessing

Before we generate sentence embeddings from our reflections dataset, we first preprocess the data by removing any blank, or null, student responses, and also removing any non-breaking spaces which appear in the text. Next, the student responses are compiled and provided to the model for generating sentence embeddings.

3.2 Sentence Transformers

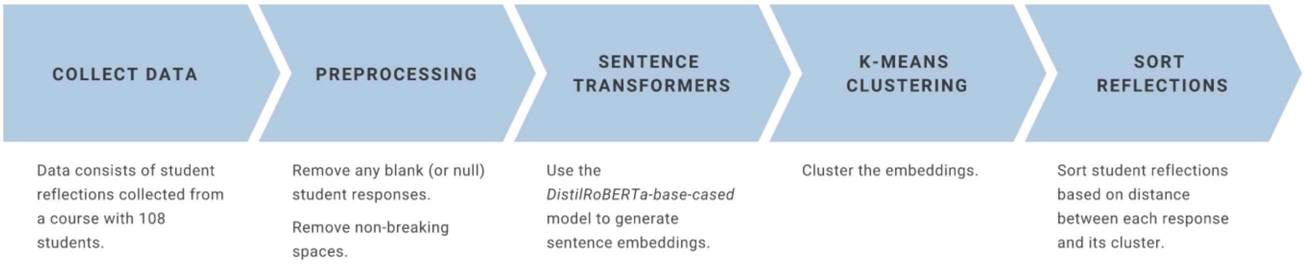


Figure 1: Illustration of our clustering and sorting approach of short student reflections.

Background. Transformer architectures can be computationally inefficient when trying to find the most semantically similar pair in sizable collection of sentences. To address this issue, sentence transformers were developed. Sentence transformers utilize mean pooling which computes the average of all the word-level vectors in the inputted sentence. Pooling helps sentence transformers maintain a fixed size vector as their output. Sentence transformers then undergo a fine-tuning training process using the SNLI dataset [3] containing over 570,000 annotated sentence pairs. The fine-tuning process Siamese and triplet networks [26] are utilized to compute weights during fine-tuning so that sentence embeddings are optimizing for meaningfulness and can be compared with cosine-similarity. Working with sentence-level representations make it easier and more efficient for tasks such as computing the semantic similarity of 2 sentences. Sentence transformers reduce computation time of finding the most similar Quora question from over 50 hours to a few milliseconds using Transformer architectures [23]. Furthermore, Sentence transformers outperform regular transformers on several semantic textual similarity tasks [23].

Approach. We use the sentence-transformers package [23]. We particularly select the `DistilRoBERTa-base-cased` model to get our sentence embeddings. `DistilRoBERTa-base-cased` is a RoBERTa transformer model [16], distilled using [25]. The dimension of the embeddings is 768. In the embedding process, we take each student response which is typically a sentence in length, and convert it into a vector of 768 floats representing the sentence. These embeddings are then used to cluster the reflections as described in the next subsection.

3.3 Clustering

Our earlier step yields a set of embedded student responses one set for module 7 reflections and another for module reflection 8. For each set of embedded student responses from our earlier step, we use K-means clustering using the scikit-learn machine learning library [21]. We compute the cluster centers for each cluster using the embedded student responses, hence cluster centers are represented by an embedding vector of the same shape. We also assign each response to a cluster based on the nearest cluster center.

The number of clusters was determined using the Silhouette method [24] for finding the optimal number of clusters. Using the Silhouette method, we generate 4 clusters for module reflection 7 and 8 clusters for module reflection 8.

3.4 Sorting of Student Reflections

After each student reflection is assigned a cluster, the reflections undergo a sorting process. The goal of the sorting process is to group reflections from most similar to least similar to assist in the reading process. Cluster distances were calculated using the scikit-learn library **`fit_transform`** function which computes and transforms the sentence embeddings to cluster-distance space. This function uses the *euclidean distance formula* for calculating the distance between a student reflection response r and its assigned cluster center r_c, as follows:

$$distance(\mathrm{e}(r), r_c) = \sqrt{\mathrm{e}(r) \cdot \mathrm{e}(r) - (2 * \mathrm{e}(r) \cdot r_c) + r_c \cdot r_c}$$

where $\mathrm{e}(r)$ represents a student response r embedded using sentence-transformers into a vector of 768 elements. r_c represents the computed cluster center assigned to r.

After computation, we sort the reflections using the assigned cluster number to group reflections within the same cluster together. Lastly, we sort the reflections within the same cluster using the *distance* metric in descending order as well. This way reflections are sorted by most semantically similar to the cluster center to least semantically similar to the cluster center. Next, we explore our user study set up and evaluate how well this approach assists in the reading process.

4. RESULTS

4.1 Experimental Setup

In order to measure the efficacy of clustering in the knowledge extraction process, we developed a user study which compares the time efficiency of reading through and extracting topics from student reflections in two formats:

1. Unsorted student reflections exported directly from the LMS.
2. Sorted student reflections sorted based on cluster distances.

First, the method of the user study will be described, and then a summary of the results. Our hypothesis when conducting this study was that clustering can help reduce the cognitive load and increase effectiveness and efficiency of knowledge extraction.

In this user study, four teaching assistants were selected to read through the student reflections of a Software Engineering course. The module 7 and 8 reflections were chosen as the corpora to extract knowledge from, as the TAs had not yet read these in particular.

Each TA was assigned a reflection and a format. For example, TA 1 would read and extract topics from Reflection 7 unsorted, TA 2 would read and extract topics from Reflection 7 clustered/sorted, and so on, as illustrated in Table 3. For the TAs which were assigned the clustered/sorted format, they individually ran the K-Means clustering algorithm first without reading any responses before beginning the process.

Table 3: Assignment of TAs to specified reflection and format type for the knowledge extraction process.

Module Reflection	Unsorted	Sorted
Reflection 7	TA 1	TA 2
Reflection 8	TA 3	TA 4

The free-response question used in particular for this study was:

> "*What was your biggest challenge this past week? This can include in-class activities, assignments, prep work, studying, time management, motivation, and so on.*"

Each TA individually read through each student's reflection response for this question, extracted any new topics mentioned in the student response, and timed themselves accordingly for the duration of the process. Once all TAs had collectively finished, they then met to discuss what topics they found, and compared times and results.

4.2 Evaluation

After comparing results of this study, we derive that by providing instructors with student reflections in a clustered and sorted format, the time needed for knowledge extraction decreases while maintaining the accuracy of identifying topics. Reflection 7, with a total of 94 student responses, took 90 minutes to completely read through and extract topics on the unsorted responses, while only requiring 15 minutes in the sorted and clustered format. Reflection 8 had similar results in which efficiency increased, with a total of 89 responses taking approximately 121.4 minutes on the unsorted format and 20.9 minutes on the clustered and sorted responses. It is important to note that the TA extracting knowledge from Reflection 8 unsorted did not complete within a 90 minute time frame, thus their results were normalized based on how many reflections they did complete. These results are provided in Table 4.

In addition to the increased efficiency of knowledge extraction with a clustered and sorted format, the topics extracted

Table 4: Normalized time taken to fully extract knowledge from all student responses per module reflection.

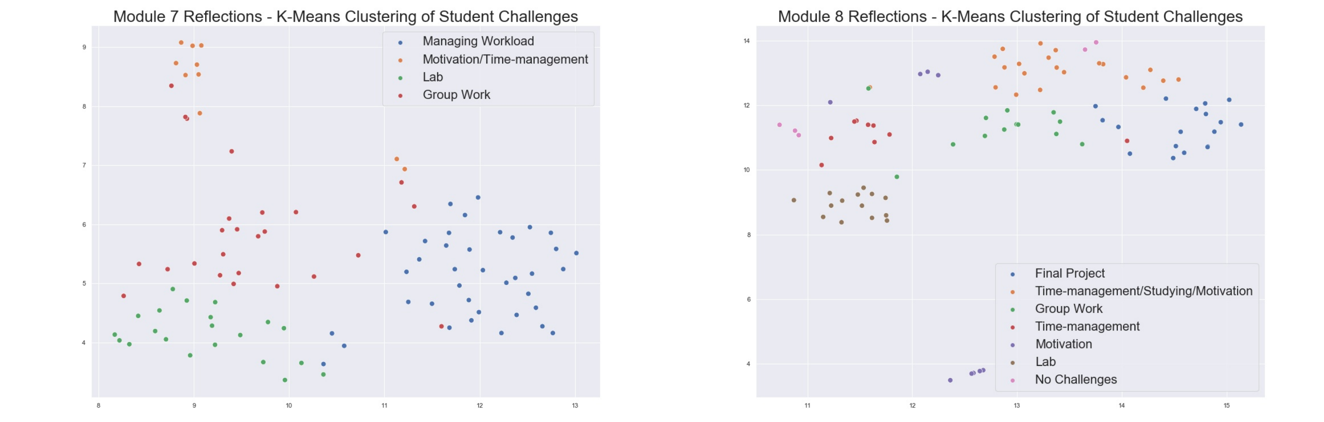
Module Reflection	Unsorted	Sorted	N
Reflection 7	90.0 minutes	15.0 minutes	94
Reflection 8	121.4 minutes	20.9 minutes	89

remained consistent, with a slight improvement in comparison to the unsorted format. Following the portion of the user study which required TAs to individually extract topics from the reflections, they then met afterwards to discuss their similarities and differences in topics. The TAs who analyzed Reflection 7 extracted the same topics from the student responses with no differences. During the discussion, the Reflection 7 TAs took turns sharing the topics they had extracted during the user study, and concluded that they were in 100% agreement with the topics coded. Reflection 8, however, had one topic which was extracted in the clustered and sorted reflections and not in the unclustered/unsorted reflections. The TAs assigned with Reflection 8 noted that this was most likely due to a lack of time to completely analyze all unsorted student reflections, hence displaying how time efficiency can also be beneficial to improving the accuracy of knowledge extraction if given a time-constraint. Despite the improved time efficiency of the clustered and sorted reflection format, no topics were missed.

We utilize the dimension reduction algorithm UMAP [18] to visualize the resulting clusters of student reflections as shown in Figure 2. The student reflections for Module 7 resulted in 4 clusters with 4 major topics including managing workload, motivation and time management, lab work, and group work. The Module 8 student reflections resulted in 8 clusters with each cluster containing a challenge in at least one of the following categories: Lab work, time management, studying, motivation, group work, and some reflections mentioned no challenges whatsoever. Managing workload, motivation, studying, and time management relate to the student's own discerned ability to handle the coursework in general. Lab work and group work were challenges in which students related their troubles more specifically to difficult topics being covered, confusions about instructions, or trouble with communicating among their groups to complete activities. Students who were in the category of "no challenges" noted that they did not have any difficulties or confusion during the span of that module. As displayed in these scatter plots and the major topics described, there are overlaps among several of the clusters. This overlap is created by the similarities in the students' wordings. For example, two student responses within the "Managing Workload" cluster of the Module 7 reflection were:

1. "My biggest challenge has been not procrastinating my work."
2. "The biggest challenge this week was working with the dash and the dashboard framework."

The first student response was the cluster center with a dis-



(a) Module 7 Reflection clusters based on question 2: student challenges.

(b) Module 8 Reflection clusters based on question 2: student challenges.

Figure 2: UMAP scatter plots visualizing student reflection K-Mean clusters

tance of 3.12, and the second student response was one of the farthest points from the cluster center, with a distance of 7.05. Therefore, clusters still maintain semantic similarities to many of the responses with smaller intracluster distances, but contain outliers due to the overlap caused by similar word usages.

5. RELATED WORK

Reflections are a necessary component in active learning courses, as it allows the instructor to track students' impression on the course, activities, and social learning aspects [9]. In Dorodchi et al. [8], student reflections are used in an introductory computer science (CS1) course to test its efficacy as a feature to predict early on which students may be at-risk of failing. By including student reflection data as a feature in a temporal data model, referred to as the student sequence model, the authors were able to increase the accuracy of predicting student outcomes of pass or fail [8]. Despite the advantages of integrating student reflections into a course model, these benefits require the time-consuming process of manually reading through individual reflections and extracting common themes. For this reason, creating an automated process to assist instructors is similarly explored in [5]. Chen et al. [5] presents positive results in exploring the usage of topic modeling for analyzing and extracting knowledge from student reflections. In this particular study, the MALLET toolkit was utilized for the topic modeling process, and the number of clusters K was manually selected. These methods of knowledge extraction are not only effective in an academic environment, but is also used in other applications such as social media mining for COVID-19 related information. Comparatively to the time-sensitive task of analyzing student reflections, clustering can also be used to discover new information from relevant tweets to assist in the decision-making steps that may follow [12]. For this task, Ito et al. [12] and Asgari et al. [2] implement algorithms using K-means clustering and sentence embeddings, which both provide positive results in topic extraction. Our study is distinguished from prior works in that, we collect and cluster short text student reflections and we conduct an educator-centered evaluation where we assess the direct impact of our approach on teaching assistants' reading and analysis time.

6. DISCUSSION & FUTURE WORK

In our research, we implement an approach using k-means clustering and sentence-transformers on student reflections to aid in reducing the labor and time-consumption of manually analyzing reflections. Our study presents promising preliminary results showing that by clustering student reflections based on semantic similarities and sorting by intracluster distance, instructors are able to decrease the time needed to extract topics from the student corpora. However, our study suffers from several limitations. Firstly, our sample size for the user study is very small ($N = 4$) and our results may not generalize to different classes, or reflection corpora. Furthermore, teaching assistants read at different paces. Our results may not generalize to different teaching assistants. To address these limitations we intend to conduct a user study with a significantly larger pool of participants, module reflections, and in multiple courses. In addition, we are planning to utilize fuzzy clustering [11] in the future version as well.

Reflections are fundamental for enhancing learning in classrooms [9], and provides the instructor with instant feedback on student progress. This study focuses on exploring the impact of clustering on student reflections to assist instructors in reducing time costs of analysis. In our future work, we plan to integrate our k-means clustering algorithm into a dashboard tool for instructors and conduct an expanded user study to further evaluate our approach. The dashboard will provide instructors and TAs the functionality to cluster student reflections from the LMS and be guided through the responses.

7. ADDITIONAL AUTHORS

8. REFERENCES

[1] A. Al-Doulat, N. Nur, A. Karduni, A. Benedict, E. Al-Hossami, M. L. Maher, W. Dou, M. Dorodchi, and X. Niu. Making sense of student success and risk through unsupervised machine learning and interactive storytelling. In *International Conference on Artificial Intelligence in Education*, pages 3–15. Springer, 2020.

[2] M. Asgari-Chenaghlu, N. Nikzad-Khasmakhi, and S. Minaee. Covid-transformer: Detecting covid-19 trending topics on twitter using universal sentence encoder.

[3] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[5] Y. Chen, B. Yu, X. Zhang, and Y. Yu. Topic modeling for evaluating students' reflective writing: A case study of pre-service teachers' journals. In *Proceedings of the sixth international conference on learning analytics & knowledge*, pages 1–5, 2016.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] M. Dorodchi, E. Al-Hossami, M. Nagahisarchoghaei, R. S. Diwadkar, and A. Benedict. Teaching an undergraduate software engineering course using active learning and open source projects. In *2019 IEEE Frontiers in Education Conference (FIE)*, pages 1–5. IEEE, 2019.

[8] M. Dorodchi, A. Benedict, D. Desai, M. J. Mahzoon, S. Macneil, and N. Dehbozorgi. Design and implementation of an activity-based introductory computer science course (cs1) with periodic reflections validated by learning analytics. 12 2018.

[9] M. Dorodchi, L. Powell, N. Dehbozorgi, and A. Benedict. *Strategies to Incorporate Active Learning Practice in Introductory Courses*, pages 20–37. 04 2020.

[10] M. M. Dorodchi, N. Dehbozorgi, A. Benedict, E. Al-Hossami, and A. Benedict. Scaffolding a team-based active learning course to engage students: A multidimensional approach. In *2020 ASEE Virtual Annual Conference Content Access. ASEE Conferences, Virtual On line*, 2020.

[11] M. Doroodchi and A. Reza. Implementation of fuzzy cluster filter for nonlinear signal and image processing. In *Proceedings of IEEE 5th International Fuzzy Systems*, volume 3, pages 2117–2122 vol.3, 1996.

[12] H. Ito and B. Chakraborty. Social media mining with dynamic clustering: A case study by covid-19 tweets. In *2020 11th International Conference on Awareness Science and Technology (iCAST)*, pages 1–6. IEEE, 2020.

[13] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

[14] H. Li, W. Ding, S. Yang, and Z. Liu. Identifying at-risk k-12 students in multimodal online environments: A machine learning approach. *arXiv preprint arXiv:2003.09670*, 2020.

[15] Y. Liu and M. Lapata. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[17] K. Mahajan, E. Al-Hossami, and S. Shaikh. TeamUNCC@LT-EDI-EACL2021: Hope speech detection using transfer learning with transformers. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 136–142, Kyiv, Apr. 2021. Association for Computational Linguistics.

[18] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.

[20] N. Nur, N. Park, M. Dorodchi, W. Dou, M. J. Mahzoon, X. Niu, and M. L. Maher. Student network analysis: a novel way to predict delayed graduation in higher education. In *International Conference on Artificial Intelligence in Education*, pages 370–382. Springer, 2019.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[22] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[23] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[24] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[25] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[26] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages

815–823, 2015.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[28] Z. Zhang, J. Yang, and H. Zhao. Retrospective reader for machine reading comprehension. *arXiv preprint arXiv:2001.09694*, 2020.