METRIC LEARNING FOR EDUCATION TEXT CLASSIFICATION

Greg Rolwes
Saint Louis University
St. Louis, MO 63103
gregory.rolwes@slu.edu

December 2, 2020

ABSTRACT

Text classification is a problem domain with a wealth of applications, especially in the context of online education. However, while classifying content based on its domain is beneficial for categorization, using such a classification for a recommendation system ignores the nuance contained within a lesson and over-generalizes its contents. To create a basis for more individualized and accurate lesson recommendations, I apply metric learning approaches to generate an embedding space that outperforms a standard classification loss using the same model.

1 Introduction

Online education rests on the internet's capacity for vast amounts of freely accessible content. This movement is reflected in the rise in popularity of massive open online courses (MOOCs) [1]. However, MOOCs have struggled to generate content that engages students in a meaningful way and matches their individual needs and interests leading to low student retention [2]. To properly filter the online lessons recommended to students, a representation that captures the numerous interconnected parts involved in information organization is necessary. While courses and lessons tend to be categorized into discrete fields, many of these fields bleed over into others. As an obvious example, calculus and physics are highly dependent on one another but statistics not as much so, despite both calculus and statistics being a part of mathematics (see Figure 1). Therefore, its necessary to create a continuous, rather than discrete, representation of lessons, while maintaining some amount of discrete structure for the sake of identification. With such representations, courses and lessons can be managed on a more granular level to individually target students based on their interests within one or more fields.

I use the MOOCCube dataset of Chinese lesson transcripts along with the Web of Science dataset of scientific paper excerpts to generate embedding spaces for identification through comparison to the surrounding embeddings, and compare this to a standard discrete classification approach using the same model.

In the absence of student lesson preferences alongside transcripts, this project is a foundation for accurate and considerate lesson recommendation systems. The objective is to generate an embedding space that accurately classifies lessons while providing a realistic representation of their content in relation to the other lessons.

2 Related Work

Deep Learning and MOOCs Considering the varying contents of lessons and the diverse backgrounds of students, recent approaches have attempted to personalize MOOCs through classification of students [3], while others have attempted standard classification of video transcripts with convolutional neural networks [4].

Metric learning and text classification. Metric learning through triplet loss is a well-known technique originally used in image classification [5] and has been transferred to text classification in a number of instances [6, 7]. Wohlwend, et al. apply some metric learning techniques to one of the same datasets (WOS), but with a modified label set and a

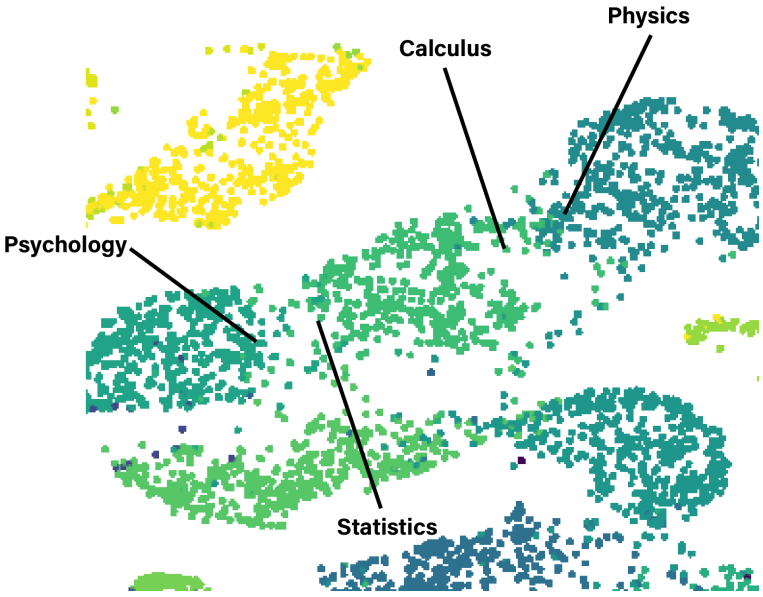

Figure 1: A sample embedding space. While calculus and statistics maintain the same overarching label, a calculus lesson might be considered more related to a physics lesson than statistics. This allows for a level of inter-class variability that properly reflects education content.

slightly different application. Text classification has been applied to MOOCs in [8] for student comment classification and in [4] for video transcript classification.

3 Method

3.1 Data

MOOCCube contains data from 706 XuetangX courses, 689 of which have Chinese video transcripts for each video-based lesson, making 41,605 transcripts of varying length and subject-matter [9].

WOS (Web of Science) contains 46,985 paragraph-long English excerpts from scientific papers categorized into 134 fields.

Train/Test Splits When performing classification (through the classifier or P@1 of the embeddings), a train set of 70% of the data was used and a test set of the remaining data set aside until all experimentation was complete. When attempting to determine the embedding performance on unseen classes, the data was split into a 50/50 train/test class-disjoint split (as is the typical approach in metric learning).

3.2 Setup

For all experiments I used an embedding layer with a 256-dimension output fed to a two-layered, bidirectional LSTM with a hidden state size of 200. The LSTM feeds to a dropout layer followed by a fully connected layer and sigmoid activation. These dimensions and all other parameters were manually tuned over the course of dozens of trials.

Classification vs Embedding The sole difference between the classification trials and the embedding trials was the output of the fully connected layer. In the case of classification the output was the size of the number of classes as is standard, whereas in the case of embedding generation for metric learning the outputs were 256-dimensional embeddings. Even the sigmoid activation is kept in the metric learning (embedding) trials to truly compare the effects of the triplet loss and embedding space, although this activation is atypical for metric learning and likely hindered performance.

Table 1: Classification accuracy.

	Classification	Metric Learning
MOOCCube	34.4	50.8
WOS	53.7	67.0

Loss For the classification task, a standard categorical cross entropy loss was used. For embedding tasks, standard triplet loss with batch all triplet mining was used. Although they are far from the most sophisticated loss functions, these were chosen to more accurately compare the direct effect of using document embeddings rather than standard classification.

3.3 A Brief Overview of Triplet Loss

Triplet loss aims to create clusters of classes by pulling same-label examples closer and pushing away examples of a differing label. This way the model learns one (or more if multiple clusters form) representations of a class while using an N-dimensional embedding space to rank inter-class examples by similarity. Specifically, triplet loss is calculated by taking an anchor example of class A, a positive example also of class A, and a negative example of class B. The loss is then calculated based on the similarity or distance (normally as cosine similarity) between the anchor and the other two examples, like so:

$$\mathcal{L}(A, P, N) = max(\parallel f(A) - f(P) \parallel^2 - \parallel f(A) - f(N) \parallel^2 + \alpha, 0)$$

Where α is the margin parameter, which was set to 0.1 throughout the trials. Batch all triplet loss involves performing this loss calculation on every possible triplet within a batch.

3.4 Evaluation

Classification I used three varying forms of evaluation. In the first, simple classification was used as a baseline comparison. I then performed a comparable metric learning approach. At test time, I find the closest train embedding to each test embedding, and consider the train embedding's label to be the chosen label for the test embedding. This is equivalent to precision at 1 of the test example within the train embedding space. I also evaluate precision at 5 and 10 to get a better sense of how well grouped the embedding space is.

Evaluation on Unseen Classes To account for the standard metric learning evaluation, I also run trials with the datasets as 50/50, class-disjoint splits. At test time, I compare each test example to the test embedding space, rather than the train embedding space, as its class will only be contained within its own space. While this method is not comparable to the standard classification approach, it extends the use case to never before seen topics as would be necessary in many real-world applications. For this method I compute precision at 1, 5, and 10 as before. See Table 2 for results.

4 Results

4.1 Classification

As shown in Table 1, the embedding model with P@1 drastically outperforms the standard classification model. Considering both methods use the same model (and subsequently the same parameters besides the loss), the triplet loss over the embedding space can be attributed to this substantial improvement. In the case of the embedding space, all documents of the same label are not forced to be considered identical by the model. Essentially, the embedding space allows for variation within a class, as each document does not need to be extremely similar to every document within its class, but rather just some documents in its class. In Figure 2, for example, a single class can be spread into multiple clusters. In standard classification all documents of a class must fit into a single bucket.

4.2 Seen vs Unseen Class Precision

The precision metrics for these trials is shown in Table 2. One of the most obvious observations is the WOS evaluation on seen classes is fairly consistent in precision from 1 to 10. This indicates very well clustered classes. An example found to be nearest one example of the same class, is also very likely to be surrounded by 9 others. MOOCCube, on the other hand, sees a significant drop in precision between P@1 and P@5. While transcripts can be correctly identified more than 50% of the time (in the seen case), the surrounding embeddings are not as likely to be of the same class. This could potentially be due to the extreme number of classes in MOOCCube (689 when working with seen classes, half

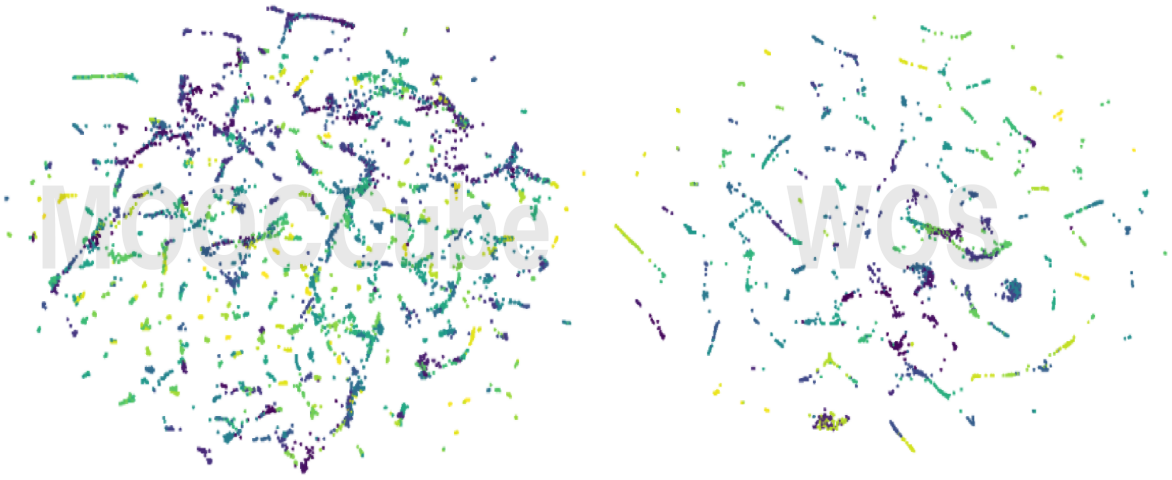

Figure 2: The embedding spaces of MOOCCube (left) and WOS (right) when evaluating on seen classes. Notice that MOOCCube is more cluttered, while WOS has found more compact clusters. This can be partially attributed to the number of classes (689 vs 134) and high similarity between classes in MOOCCube.

Table 2: Precision on seen vs unseen classes.

	Seen Classes			Unseen Classes		
	P@1	P@5	P@10	P@1	P@5	P@10
MOOCCube	50.8	41.0	37.2	59.4	44.3	38.5
WOS	67.0	64.5	63.9	17.1	11.1	9.3

that for unseen) and the high potential of overlap between educational topics (see Introduction and Figure 1). Figure 2 provides a visualization of these embedding spaces.

5 Conclusion and Further Work

In this work I demonstrated the superior text classification potential of a basic metric learning loss compared to standard classification. In addition, this approach to text classification was able to determine the relatedness of documents from unseen classes with reasonable success. Applied to education, metric learning can act as a foundation for lesson recommendation systems that consider content on a more individual level and rely less on overarching labels. Further work collecting data that connects lesson transcripts to student preference and/or success is needed to implement such a system. In addition, future work should implement some of the more sophisticated metric learning methods on this and other educational text datasets.

References

[1] Joseph Chapes. Online video in higher education: Uses and practices. In Joyce P. Johnston, editor, *Proceedings of EdMedia + Innovate Learning 2017*, pages 1133–1138, Washington, DC, June 2017. Association for the Advancement of Computing in Education (AACE).

[2] Kate S. Hone and Ghada R. El Said. Exploring the factors affecting mooc retention: A survey study. *Computers Education*, 98:157 – 168, 2016.

[3] K. Kaabi, F. Essalmi, M. Jemni, and A. A. Qaffas. Personalization of moocs for increasing the retention rate of learners. In *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, pages 1–5, 2020.

[4] H. Chatbri, M. Oliveira, K. McGuinness, S. Little, K. Kameyama, P. Kwan, A. Sutherland, and N. E. O'Connor. Educational video classification by using a transcript to image transform and supervised learning. In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2017.

[5] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. *CoRR*, abs/1404.4661, 2014.

[6] Zongze Ren, Zhiyong Chen, and Shugong Xu. Triplet based embedding distance and similarity learning for text-independent speaker verification, 2019.

[7] Jeremy Wohlwend, Ethan R. Elenberg, Samuel Altschul, Shawn Henry, and Tao Lei. Metric learning for dynamic text classification, 2019.

[8] Y. Ziming, C. Yan, and Z. Qiang. Study on text classification of mooc course comments based on chinese character-level convolutional networks. In *2018 International Computers, Signals and Systems Conference (ICOMSSC)*, pages 679–681, 2018.

[9] Jifan Yu, Gan Luo, Tong Xiao, Qingyang Zhong, Yuquan Wang, Wenzheng Feng, Junyi Luo, Chenyu Wang, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jie Tang. MOOCCube: A large-scale data repository for NLP applications in MOOCs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3135–3142, Online, July 2020. Association for Computational Linguistics.