Text Representations of Math Tutorial Videos for Clustering, Retrieval, and Learning Gain Prediction

Pichayut Liamthong
Worcester Polytechnic Institute
pliamthong@wpi.edu

Jacob Whitehill
Worcester Polytechnic Institute
jrwhitehill@wpi.edu

ABSTRACT

With the goal of making vast collections of open educational resources (YouTube, Khan Academy, etc.) more useful to learners, we explored how automatically extractable text representations of math tutorial videos can help to categorize the videos, search through them for specific content, and predict the individual learning gains of students who watch them. In particular, (1) we devised novel text representations, based on the output of an automatic speech recognition system, that consider the frequency of different tokens (symbols, equations, etc.) as well as their proximity from each other in the transcript. Unsupervised learning experiments, conducted on 208 videos that explain 18 math problems about logarithms show that the clustering accuracy of our proposed methods reaches 85%, surpassing that of standard TF-IDF features (78% using log normalization). (2) In a video search setting, the proposed text features can significantly reduce the number of videos (up to 88% reduction on our dataset) and amount of video time (up to 82%) that users need to spend looking for desired content in large video collections. Finally, (3) in an experiment on Mechanical Turk with $n = 541$ participants who watched a randomly assigned tutorial video between a pretest & posttest, the text features and their multiplicative interactions with students' prior knowledge provide a statistically significant benefit to predicting individual learning gains.

Keywords: Open educational resources (OER), Crowdsourcing, Information Retrieval

1. INTRODUCTION

Consider a large repository (Khan Academy, edX, etc.) of open educational resources (OERs) such as tutorial videos, and a scenario in which the ultimate goal is to help learners to learn by recommending relevant and high-quality content that matches the students' needs. Knowing *what* the learner needs and providing the *right* content that suits them is crucial. We could estimate automatically the most beneficial content by analyzing their performance on prior exam-

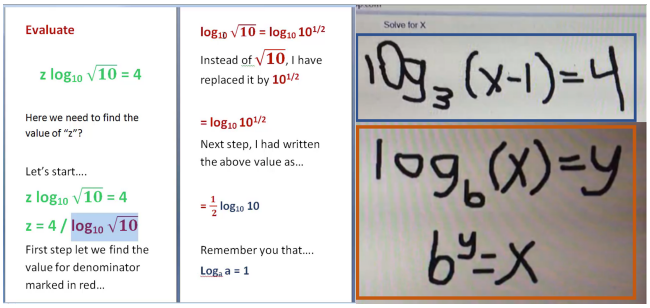

Figure 1: Example videos in our study. Right: Google's Speech-to-Text extracts the text "solve for x ok our problem is log base 3 of x minus 1 equals 4...".

inations. However, a current challenge with contemporary OER repositories is that the content within each resource is typically poorly annotated, with tags that are too general, e.g., "algebra" or "linear equations" rather than "Simplify $\log_{10} 1000$.". Given the high labor and time involved in manual annotation, it is desirable to devise methods of *automatically* analyzing OER content and devising representations that can facilitate efficient search and categorization.

While optimal character recognition and handwriting recognition are both mature fields, they are typically evaluated in much more constrained settings than math tutorials, in which math is mixed with natural language, and extraneous lines and other graphics can exist (see Figure 1). In full-fledged tutorial videos, this segmentation can be very challenging. Our research focuses instead on analyzing the speech transcript of the video (while ignoring other potential audio characteristics such as background noise, pitch, etc.). When a particular expression or equation is presented in a video, there is a high chance that the speaker will also say that expression/equation out-loud to the learners (Figure 1). Rather than manually transcribing the text from the video, we consider only fully automatic approaches based on automatic speech recognition (ASR; we used the Google Speech-to-Text API in our work, more detailed about the pilot test in Appendix B). Hence, the text representations we explore must contend with imperfect transcripts. We then assess the utility of the proposed representations for three tasks: (1) *cluster* the videos automatically into the specific math problems that they explain; (2) *search* through a library of videos for one that explains a particular math problem; and (3) *predict* the individual learning gains of students who

watch the videos in a pretest/treatment/posttest paradigm. In these ways, we hope to make available to students the *right* content that is already available, but not easily findable, among large-scale OER repositories.

We conduct our investigation on a collection [14] of math tutorial videos about logarithms, and another dataset from YouTube on basic algebra. Our goal is not just to make coarse distinctions between videos about "algebra" versus "geometry", but rather fine-grained distinctions about specific math problems. Mirroring our goals from the previous paragraph, our research questions are the following: **RQ1**: How accurately can the devised text features cluster videos into *fine-grained* categories about the specific problem they are solving, and which aspects of these representations are most important? **RQ2**: By how much can we reduce the search time to find a relevant video? **RQ3**: Are the text features predictive of the individual learning gains of students who watch these videos in a pretest/posttest setting?

2. RELATED WORK

Text Representations: There are several prominent text representations used for language modeling: (1) Term frequency and Inverse document frequency (TF-IDF) [12]: TF-IDF features typically do not require training and are thus suitable for unsupervised settings. (2) Word embedding models [8, 7] based on neural networks trained using supervised learning. (3) Sentence-level models such as BERT [8] that capture higher semantics compared to word embeddings.

Video Categorization & Clustering: For categorizing video content automatically, much of the prior work has focused on other fields than math tutorial such as films, sport videos [2], [4], [11]. Most prior methods on video categorization focus on visual aspects such as frame transitions, object detection and segmentation. Some as them use the audio (e.g. [2]) such as the audio frequency and amplitude statistics. We are unaware of any previous research that clustered video content at the low-level tags of individual math problems.

Video Retrieval in OERs: There has been increasing interest in the task of video retrieval of OERs. Many works in have pursued combined feature representations with both textual and visual information [13, 15, 3]. Hürst [3] found that the lecture slides are more useful than the corrected transcriptions. In our work, while we focus solely on text representations, the features we devise could be easily combined with visual features.

Estimating the Effectiveness of OERs: For the task of estimating the effectiveness (e.g., associated learning gains) of viewing tutorial videos, researchers have pursued various approaches, including estimating their effectiveness through correlated measures such as engagement while watching the video [10, 6, 1]. For estimating the effectiveness of OERs in general, one can also use a combined experimental and reinforcement learning-based approach such as bandit algorithms [9]. While Rafferty et al. [9] suggested the potential use of *context* (for example, features of the OERs as well as of the students' prior knowledge) for predicting learning gains, they did not actually pursue that approach.

3. TEXT REPRESENTATIONS

In this paper we explore unsupervised representations of the transcripts of math tutorial videos. When designing the representations, we considered the following characteristics: (1) Similar content should involve similar tokens. A math video whose transcript consists of just "two plus three", for example, is unlikely to be similar to a video whose transcript is "four times x". (2) The most important tokens tend to recur within a video transcript. Conversely, tokens that are uttered only once are often less important or even be transcription errors. (3) The relative order of nearby tokens is important for deciphering the math content. For example, "four over two" and "two over four" are different fractions, but the difference is reflected only in the relative order of tokens, not in their frequencies. For characteristics (1) and (2) above, we created several variations of "1D" text representations that capture which tokens occur more frequently in each video. With the additional characteristic (3), we also explored "2D" text representations that can capture the relative order within a fixed radius from token i w.r.t. token j for each (i, j) pair. We note that extracting the precise mathematical expression from the transcript is inherently ambiguous. For example, the two distinct expressions 2^{x+2} and $2^x + 2$ would likely both be spoken as "two to the x plus two". Fortunately, our objective is not to capture the math content perfectly, but to capture enough of it to enable effective clustering, search, and prediction of learning gains. Below we describe different kinds of unsupervised text representations that vary in terms of token type, order dependency, and summarization method.

3.1 Token Types

3.1.1 Individual Token

As our simplest representation, we call each word (separated by space) a *token*, and then we count the number of math-related tokens, defined as: (1) numbers (digit-only), (2) operations (e.g. $+, -, \times$), or (3) variables (an alphabet). For the operations, we map synonyms to the same token, e.g., 'plus' to '+', 'to the [power]' to '^'. Additionally, we add the words corresponding to each digit 0 to 9 (i.e. 'zero', ..., 'nine') as math-related tokens. For variables, we used a restricted alphabet consisting of $\{b, c, n, m, w, x, y, z\}$ (we omitted 'a' since it is also a common English word).

3.1.2 Expression Token

To infer which math problem in video, it might be useful to extract the entire expression.For example, "2 plus 3" could be considered as one token "2+3" not '2', '+', and '3'. Specifically: (1) We mark all tokens in the transcript as either math-related or non-math-related. Tokens that are labeled as math-related are literals (LIT) and operators (OP) such as plus (+), $\sqrt{}$, etc. (2) For each contiguous sequence of math-related tokens, we read the tokens one-by-one and concatenate them into one expression according to the rule: starting with LIT followed by OP, LIT, ... (alternately).

3.2 Token Count Vector

Given the sequence of tokens in each video, we then compute either a 1D vector or 2D matrix of frequency statistics (which are finally summarized as described in Section 3.3). In the subsections below we let $\mathcal{T}$ be the set of all tokens that appear in any of the videos.

1D (No Order Dependencies): The count vector of each video contains $|\mathcal{T}|$ components, each of which records the frequency of token occurs in video.

2D (First-Order Dependencies): With the goal of encoding the *relative order* of tokens, we computed a 2D *matrix* M, of size $|\mathcal{T}| \times |\mathcal{T}|$, such that M_{ij} is the number of times that token i appears before token j in the transcript. In this approach, we introduced a "radius" parameter k to limit the distance of token pairs (i, j) that need to be considered. For example, if $k = 4$, all token pairs (i, j) such that the distance between i and j is ≤ 4 will be counted, otherwise, ignored.

3.3 Token Summarization Methods

Given the token count vector computed in Section 3.2, we then summarize each count x using a summarization function f. We considered the following functions: (1) **Raw Frequencies**: We let $f(x) = x$. (2) **Binarized Frequencies**: Binarizing the counts x might be less susceptible to noise; hence, we tried setting: $f(x) = 1$ if $x \geq 1$ and $f(x) = 0$ if $x = 0$. (3) **Weighted Frequencies**: It might be beneficial to weight down tokens which appears once because it might be noise from the extraction process; important tokens should be mentioned multiple times in video; token found only once (we call it $t_{=1}$) are either insignificant or incorrectly extracted. Instead of removing $t_{=1}$, we introduced the parameter r to downweight $t_{=1}$. In this case, instead of having the raw frequencies, We fixed the weight of $t_{>1}$ (appear *more than* once) as 1; however, we downweight $t_{=1}$ by r. We thus let $f(x) = 1/r$ if $x = 1$, $f(x) = 0$ if $x = 0$, and $f(x) = 1$ if $x > 1$. Note that $r = 1$ is equivalent to Binarized Frequencies.

4. DATASET

We applied the text representations to two sets of tutorial videos: (1) *Logarithms* and (2) *Algebra*, see Appendix A.

5. CLUSTERING

Given the different feature types, we test whether they serve as an effective basis for clustering the videos. In this section, as ground-truth cluster labels, we took the math problem (there were $K = 18$ unique problems in total) that each video explained as its label. Note, however, that we could also cluster the videos by the *category* of problems that they explain (see Section 4); we do so in Section 7.

Methods: For each of the different text representations, we applied K-means clustering to group the videos into $K = 18$ clusters, followed by the Hungarian algorithm [5] to optimally match the estimated cluster to the ground-truth indices. Since K-means converges to different local minima depending on the random initialization, we executed the algorithm 512 times and then reported the average of accuracy for the clustering with lowest sum of squared distance.

Results: Table 1 shows the clustering accuracy results. All three methods yield accuracies that are much greater than the *random* baseline, which achieved only 18.27% accuracy.

Weighted Frequencies: We tried multiple values of r ($r = 1$ is equivalent to the Binarized Token Counts). We also added $r = 0.5, 0.25$; this contrasts with our intuition for when it weights $t_{=1}$ more; we added this as a sanity check that the accuracy should be getting worse. Table 1 shows that the weighted frequencies increase the accuracy significantly up by 10% on average. $r = 2$ performs the best among $r = 2, 4, 8$. As r gets larger, we see a slight decrease in accuracy.

1D vs. 2D: Table 1 (right) shows clustering accuracy with the 2D approach. For the radius $k = 2$ on the Expression token (and using weighted frequencies with $r = 2$), the accuracy increases around 2% compared to with 1D. However, we can see lots of variance in the accuracy over the different k, and hence the advantage may not be statistically reliable.

Comparison to TF-IDF: Our token summarization methods can be seen as variations of TF-IDF, where only the TF term $f(x)$ is used; in other words, we used a constant 1 for the IDF term. (We experimented with several IDF functions but found that they all worked worse than just 1.) The weighted frequency scheme we tried can be seen as a coarse (piecewise-constant) approximation to the (smooth) log function commonly used as the TF function in TF-IDF. Using TF-IDF (with log for TF and 1 for IDF) and Expression Tokens, the clustering achieved 78.67% for the Expression Token (down about 5% from our *weighted* frequency method). For the Individual Tokens, it performed similarily in accuracy compared to the weighted frequency methods. (See the "log" column in Table 1.) In summary, the results provide some evidence that our text representations may yield a worthwhile accuracy advantage over TF-IDF.

6. SEARCHING

Here we explore whether the proposed text representations could be used to create a simple search engine to reduce the amount of *video time* they would need to watch. Using the text representations, we can build a simple search engine as follows: (1) From each video i in a collection $\mathcal{S}$, we transcribe its speech into text (using Google ASR) and then extract its text representation v_i. Then, (2) for any search query (e.g., "Simplify: $\log_4 16$"), we likewise extract its text representation q using any of the methods presented in Section 3. Finally, (3) we rank all the videos in $\mathcal{S}$ by the *cosine similarity* between v_i and q.

Experiments: Here we consider a general setting in which *multiple* math problems may be explained in a *single* video. A search engine that can pinpoint which *segment* of a video explains the solution could save the user significant time compared to watching the whole video. For this setting, there is a trade-off between granularity and accuracy: the search engine may be more accurate if the segment length is longer, but the user can save more time if the segment returned to them by the search engine is shorter. Hence, we introduced a segment length parameter, L. We divided each video into multiple segments of length L. Each segment has its own (sub-)transcript and its own problem that it explains. Hence, we treat each segment as its own "video". Our goal is to find *any* segment in the video that explains the problem in the user's query q. As a baseline, we used a simulation (averaged over 20 runs) to estimate the sum of the segment lengths (in seconds) that a user would have to watch before finding a relevant segment.

Results on the Algebra dataset: We analyzed the 234 videos of the Algebra dataset that contain multiple problems; in

Token Types	1D								2D					
	Raw	Weighted: r						log	Radius k (for Weighted: $r = 2$)					
		0.25	0.5	1	2	4	∞		1	2	4	8	16	∞
Individual	54.33	25.48	41.35	67.31	72.11	68.75	64.90	73.56	64.90	63.46	59.13	57.69	62.02	49.04
Expression	50.48	20.67	44.71	70.19	**83.65**	83.17	80.29	78.67	83.65	**85.58**	75.96	71.64	73.56	51.44

Table 1: Clustering accuracy on the logarithm videos for 1D text representations with different token types and summarization methods, and the clustering accuracy for 2D representations and different token types (all with weighted summarization: $r = 2$).

total, these videos explain 300 algebra problems. We varied the segment length L over the set {15s, 30s, 1m, 2m, 4m} (see Figure C.1 in Appendix). The results shows that the best text representations were 2D Binarized Individual Token ($k = 8$). In particular, the 2D representations showed an advantage (compare the pairs of {blue, pink}'s *solid* and *dashed* lines). We found that radius $k = 8$ for 2D Representation preforms best across each method. For the Interval Length, the percent decrease, at $L \in \{30s, 1m\}$, in watch time is highest (i.e., the most helpful, see Figure C.1 in Appendix). As L continues to grow, the results go down and at $L = 15s$, the performance drops. This exemplifies the trade-off between segment length and available information.

Results on the Logarithm dataset: In this dataset, each video contains *one* log problem. For each of 18 logarithm problems, we search for *any* of the videos that solve that particular problem. Comparing the results with *random* baseline, the results show the same trend as for the Algebra dataset: The 2D Representation gives the best results. We found, for instance, Binarized Individual Token yields the results of 89.96%, and 93.19% for 1D and 2D ($k = 8$), respectively. The same holds true for Weighted Expression Token ($r = 2$) with the results of 91.25% and 93.20%. For the 1D approach, the best representation was TF-IDF (with log for TF and identity for IDF); the reduction was slightly lower (92.85%).

7. LEARNING GAIN PREDICTION

In this section, we investigate whether the text representation can be used to predict the learning gain of students who watch the videos as an educational intervention. The high-level idea is that the effectiveness of each tutorial video can be estimated by the *interaction* of the *content* within the video and the student's prior knowledge. In contrast to some prior work that predicted the *average* learning gains of a video over many students, here we tackle the arguably harder problem of predicting *individual learning gains* of each student, measured as the difference in test scores on the curriculum before and after watching the video.

The Logarithms dataset (Section 4) contains pretest/posttest scores of students who received a tutorial videos as an intervention. Hence, we use each participant's pretest score and the text representation of the video they watched as predictors to estimate their learning gains (posttest minus pretest score). Rather than use the text representation as a feature vector itself, we instead use the *category label* assigned to the problem (Section 5) by the clustering algorithm as a 0-1 indicator variable with an associated model coefficient; hence, our models can find interactions between a student's prior knowledge and the topic in the video they received.

7.1 Prediction Models

We considered both linear models with mixed effects, as well as deep non-linear models based on neural networks, but we found that the latter overfit too easily and gave unstable results; hence, we present only the linear models. Let $p_{ij}, j = 1, 2, 3$, be student i's prior knowledge (pretest score) within the 3 problem categories (j) on logarithms. Let $c_{ij}, j = 1, 2, 3$, be 0-1 indicator variables that reflect whether student i's assigned video belongs to each category j. (Note that each video is assigned to exactly one of the three categories.) We can compute c_{ij} using either (a) Manually Labeled Categories (MLC) from human annotators, or (b) Automatically Labeled Categories (ALC) from the text representations and clustering algorithm (Section 5).

Prediction Model: We constructed a model that considers multiplicative interactions between the student's prior knowledge p_{ij} in each problem category and the cluster label c_{ij} of the student's assigned video:
$y_i = \sum_{j=1}^{3} (w_j p_{ij} + v_j c_{ij} + u_j (p_{ij} \times c_{ij})) + \epsilon_i$. Importantly, this model contains multiplicative interaction terms $p_{ij} \times c_{ij}$.

Results: We found that the interaction $p_{ij} \times c_{ij}$ using MLC has a statistically significant effect on the learning gain ($F_{11,\,582} = 5.839, p = 5.11e-09$), and so does this interaction using ALC ($F_{11,\,582} = 6.425, p = 4.125e-10$). The RMSE is 0.464, which is slightly better (about 3.1% relative decrease) compared to prediction model 1. Specifically, we found that, for example, u_3 is *negative* and statistically significant ($p = 0.0005$) in the ALC model. The negativity of u_3 means that, if $p_{i3} \times c_{i3}$ is low, then the learning gain is high (and vice versa). In turn, $p_{i3} \times c_{i3}$ is low either because (1) p_{i3} is low and $c_{i3} = 1$, i.e., an individual knows little about topic 3 and receives a tutorial about topic 3, yielding high learning gain; or (2) p_{i3} is high and $c_{i3} = 0$, i.e., an individual already understands topic 3 and receives on another (more helpful) topic, yielding high learning gain. Both the MLC and ALC interactions were stat. sig., suggesting that the text representations can group videos in ways that predict individual learning gains.

8. CONCLUSION AND FUTURE WORK

We have devised novel text representations to represent the content of math tutorial videos. On a dataset of hundreds of math videos and hundreds of students who watched them, we showed that the representation can be used to (1) accurately (around 85%) cluster the videos into the math problems they solve (RQ1); (2) search for specific video content in a large repository of videos, thereby saving the user considerable (up to 88%) search time (RQ2); and (3) predict individual learning gains, in conjunction with features of the students' prior knowledge, with stat. significance (RQ3).

9. REFERENCES

[1] S. Bulathwela, M. Pérez-Ortiz, A. Lipani, E. Yilmaz, and J. Shawe-Taylor. Predicting engagement in video lectures. *arXiv preprint arXiv:2006.00592*, 2020.

[2] S. Fischer, R. Lienhart, and W. Effelsberg. Automatic recognition of film genres. *Technical reports*, 95, 1995.

[3] W. Hürst, T. Kreuzer, and M. Wiesenhütter. A qualitative study towards using large vocabulary automatic speech recognition to index recorded presentations for search and access over the web. In *ICWI*, pages 135–143. Citeseer, 2002.

[4] V. Kobla, D. DeMenthon, and D. Doermann. Detection of slow-motion replay sequences for identifying sports videos. In *1999 IEEE Third Workshop on Multimedia Signal Processing (Cat. No. 99TH8451)*, pages 135–140. IEEE, 1999.

[5] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[6] A. S. Lan, C. G. Brinton, T.-Y. Yang, and M. Chiang. Behavior-based latent variable model for learner engagement. *International Educational Data Mining Society*, 2017.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[8] 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.

[9] A. N. Rafferty, H. Ying, and J. J. Williams. Bandit assignment for educational experiments: Benefits to students versus statistical power. In *International Conference on Artificial Intelligence in Education*, pages 286–290. Springer, 2018.

[10] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé III, and L. Getoor. Learning latent engagement patterns of students in online courses. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1272–1278, 2014.

[11] E. Sahouria and A. Zakhor. Content analysis of video using principal components. *IEEE transactions on circuits and systems for video technology*, 9(8):1290–1298, 1999.

[12] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[13] F. Wang, C.-W. Ngo, and T.-C. Pong. Structuring low-quality videotaped lectures for cross-reference browsing by video text analysis. *Pattern Recognition*, 41(10):3257–3269, 2008.

[14] J. Whitehill and M. Seltzer. A crowdsourcing approach to collecting tutorial videos–toward personalized learning-at-scale. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 157–160, 2017.

[15] H. Yang and C. Meinel. Content based lecture video retrieval using speech and video text information. *IEEE transactions on learning technologies*, 7(2):142–154, 2014.

APPENDIX

A. DATASET

Here we described each dataset we use in the experiment in more detail.

Logarithms: This is the dataset collected by Whitehill & Seltzer [14], which contains both a repository of 208 math tutorial videos about logarithms. Most videos are between 1-3 minutes long. In total the collection spans 18 logarithm problems, with 9 to 17 videos per problem. Relevant only to Section 7, the dataset also contains students' pretest and posttest scores of 541 participants from Amazon Mechanical Turk who watched the videos. There are 226 males, 207 females, and 108 of undefined, with the average age of 33.71 $\pm$9.84. Specifically, each participants were asked to answer 19 logarithm pretest problems, which was classified into 3 main categories: (1) the logarithmic term without variables e.g. $\log_9 1$, (2) the logarithmic term with variables e.g. $\log_w \frac{1}{w}$, and (3) the logarithmic equation e.g. solve for x where $x \log_4 16 = 3$ (category 1, 2 and 3 contain 102, 61, and 45 videos, respectively). Then, they were assigned to *one* random video among 208 logarithm tutorial videos, and were asked to complete a posttest (same level of difficulty as the pretest but slightly different problems).

Algebra: For the search task, we collected another dataset, containing 234 algebra math tutorials on Youtube As of 234 videos, 213 of them contains *one* math problem and 21 of them contains *multiple* math problems (total of 87 math equations); total of 300 expressions on entire dataset. We manually annotated which equation (e.g. $2x^2 - 2x - 12 = 0$, $x + 7 = 10$) each video explains. For videos with multiple math problems, we marked the start end time of each.

B. SPEECH-TO-TEXT TRANSCRIPTION

All the feature types we explore are based on obtaining an *approximate* transcript of the video from an ASR. In particular, we use Google Speech-to-Text API. As a pilot test of its accuracy on the OERs in our dataset, we manually annotated 10 videos (in total of 3044 words in the ground-truth transcripts). Google's API achieved a word error rate (WER) of 5%, which intuitively seemed sufficient, which intuitively seemed sufficient. An example of extracted speech is shown in Figure 1 (caption). After obtaining the transcript for each video in our collection, we then tokenized it and summarized the token frequencies.

C. ADDITIONAL FIGURES

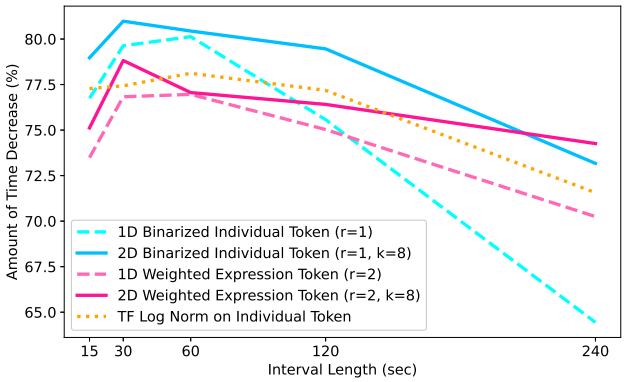

Figure C.1: The decrease in time needed to find specific math content in a set of math tutorial videos. Each line shows a different text representation over different segment lengths.