

Tutorial Recommendation for Livestream Videos using Discourse-Level Consistency and Ontology-Based Filtering^{*}

Amir Pouran Ben Veyseh^{1,*}, Franck Dernoncourt² and Thien Huu Nguyen¹

¹*Dept. of Computer and Information Science, University of Oregon, Eugene, OR, USA*

²*Adobe Research, Seattle, WA, USA*

Abstract

Streaming videos is one of the methods for creators to share their creative works with their audience. In these videos, the streamer share how they achieve their final objective by using various tools in one or several programs for creative projects. To this end, the steps required to achieve the final goal can be discussed. As such, these videos could provide substantial educational content that can be used to learn how to employ the tools used by the streamer. However, one of the drawbacks is that the streamer might not provide enough details for every step. Therefore, for the learners, it might be difficult to catch up with all the steps. In order to alleviate this issue, one solution is to link the streaming videos with the relevant tutorial available for the tools used in the streaming video. More specifically, a system can analyze the content of the live-streaming video and recommend the most relevant tutorials. Since the existing document recommendation models cannot handle this situation, in this work, we present a novel dataset and model for the task of tutorial recommendation for live-streamed videos. We conduct extensive analyses on the proposed dataset and models, revealing the challenging nature of this task. The dataset and the code are accessible at https://github.com/amirveyseh/Tutorial_Recommendation.

Keywords

Tutorial Recommendation, Video Transcripts

1. Introduction

Streaming platforms, such as Twitch, Behance, and YouTube, are effective tools for creators that equip them with the facility of directly reaching out to their audience to share their creative content. For instance, on Behance¹, creators can share their works on visual projects, such as illustrations and designs, while employing visual content editing tools, such as Photoshop and Illustrator. In these videos, the streamer discusses the details of actions required to fulfill the objective of the creative task (e.g., designing a logo). Depending on the tools and the format that the streamer chooses to present their work, the streaming video can serve as educational content to learn how to use the tools used in the video. For instance, the streamer might review how to draw the sketches for designing a fantasy character or they might discuss the various

VTU'22: The AAAI-2022 Workshop On Video Transcript Understanding, Feb 28, 2022, Virtual

^{*}✉ apouranb@cs.uoregon.edu (A. P. B. Veyseh); franck.dernoncourt@adobe.com (F. Dernoncourt); thien@cs.uoregon.edu (T. H. Nguyen)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

¹<http://www.behance.net>

methods for selecting an object in an image. Thus, these videos can help the audience to learn the nuances of the tools. However, the edit actions might be discussed in different detail. For instance, to add some shapes to an image, the streamer might briefly mention the name of the brush employed to perform this action or he/she might explain the various methods available for this action. As such, a streaming video itself might lack all the details necessary to learn an edit action. One way to fill this gap is to accompany the streaming videos with tutorials in which the details of the actions are presented (see Figure 1). Linking a streaming video with the relevant tutorials helps the audience to learn all aspects of the tools employed in the video.

Given a live-streaming video, we aim to find the relevant tutorial for it. One solution to this question is to employ the existing document recommendation tools [1, 2, 3]. However, one limitation to this approach is that the existing recommendation tools are trained on formal documents, e.g., books or news articles. As such, directly employing these models for the recommendation for the transcripts of the live-streamed videos is not optimal. In particular, unlike formal texts, in the transcripts of a video, there might be incomplete sentences, incorrect words due to ASR (automatic speech recognition) errors, or repeated sentences. These differences require domain-specific models that are designed to handle the challenges of the domain of video transcript. Moreover, another limitation of employing existing document recommendation resources is that there is no evaluation benchmark for this domain, making it more difficult to compare the performance of different models.

To address these shortcomings, in this work, we present the first large-scale tutorial linking dataset for the videos streamed on the Behance platform. More specifically, 47,403 sentences from the transcripts of 24 live-stream videos are annotated. In total, 4,126 sentences are annotated with 3 different tutorials for Photoshop. In addition, we also conduct extensive experiments on the proposed dataset. In particular, we first present the performance of an unsupervised model in which the similarity of the video transcript and the tutorial content is employed for the recommendation. Next, we employ the annotated data, to provide recommendations for the sentences in the video transcripts. Our analysis shows the challenging nature of this domain.

2. Related work

This task could be modeled either as text classification [4] or text similarity [5]. For text classification, the goal is to classify the input text into one of the pre-defined categories. Here, the categories might be defined as the available tutorials. The textual content of the tutorial describes the label of the category. For text similarity, the degree to which the tutorial content is similar to the video transcript is employed to find the most related tutorial for a given transcript. However, these solutions suffer from critical weaknesses which renders them inapplicable or inefficient for our task. First, most of the existing systems for text classification require manually labeled data. However, for our task, there is no human supervision available for training. As such, these methods might not be employed for this task. Second, both text classification and text similarity methods are evaluated on short documents (a few sentences) with formal language (i.e., a news article). However, in our task, the documents might be very large (e.g., transcripts of several hours of videos) and noisy (due to the automatic transcripts). This difference in a domain makes the majority of the solutions inapplicable to our task. Last but not least, the

Transcript

Good morning, be hands happy Monday. How are you today guys what are you planning to do today? Of what I'm planning to do today, I'm going to share with you. Pick Wick compositing in Adobe photo shop for beginners guys let's start. OK, I will be using Adobe Photoshop. Of course, my favorite. Wakeham tablet and also guys today, I will be using. Images I downloaded and licensed from Adobe stock feel free to use any images you like. So let's start. Uh I have open creative cloud library with all my images. I downloaded in licensing from Adobe stock to open any. Image from your creative Cloud Library as a separated document simply double click on your image just like this. And Wala. Now, differently, I would like to bring in other image into my open to document. Uh let me scroll my library up. Just the like this, where is it? Where is it and here I have another image I would like to use? In my today's compositing to add any image from your creative cloud library into your open document simply click on your? Image in drag it into your document just like the it's very simple guys ...

Tutorial

Learn creative compositing techniques for combining images in Photoshop. Learn techniques for adding images to a composite, blending images together using layer masks, and changing the shape of an image by transforming a smart object. Add a central element to your composite using a layer mask with selections and the Brush tool. Add some clouds to the composite for a surreal effect, and paint on a layer mask to shape the clouds. Then learn how to use the Blend If sliders to blend a photo of a sailboat into the scene. Increase perspective in the scene with additional layers that suggest foreground and background elements. Paint on a layer mask to interlace clouds with other objects. Then use a layer Blend Mode to knock out a white background on a layer. Color is an important element in composite. Learn techniques for using fill layers, layer Blending Modes, and clipping masks to control color. Finally, polish your fantasy composite by adjusting lighting and contrast with Curves.

Figure 1: Part of the transcript of the live-streamed video "Adobe Photoshop Compositing for Beginners"

existing similarity-based methods cannot incorporate background knowledge (e.g., an ontology of concepts or keywords in the domain of the videos). Moreover, they ignore the discourse-level consistency between the two texts to compute the similarity score.

3. Data Annotation

3.1. Data Collection

To train and evaluate the model we annotate data from the transcripts of the videos streamed on the Behance platform. The recordings are spilled by specialists and creators to share/discuss their inventive projects. As such, verbal substance from the speakers (in English) is imperative for video understanding. Whereas the recordings have introductory subjects, their substance is impromptu, thus the streamer might cut sentences, examine numerous themes, and utilize casual expressions. The recordings have an average length of 48 minutes. To get the verbal substance of the streamed recordings, we utilize the Microsoft ASR tool. In addition up to, 24 recordings (whose main editing tool is Photoshop) are transcribed. In total, 47,403 sentences are present in the transcribed videos to be annotated by human annotators.

3.2. Data Annotation

To annotate data, we hire expert annotators in Upwork who have experience in using Photoshop and also have experience in data annotation tasks. In total, three annotators were hired for this task. Every video transcript is assigned to the three annotators to link them to relevant

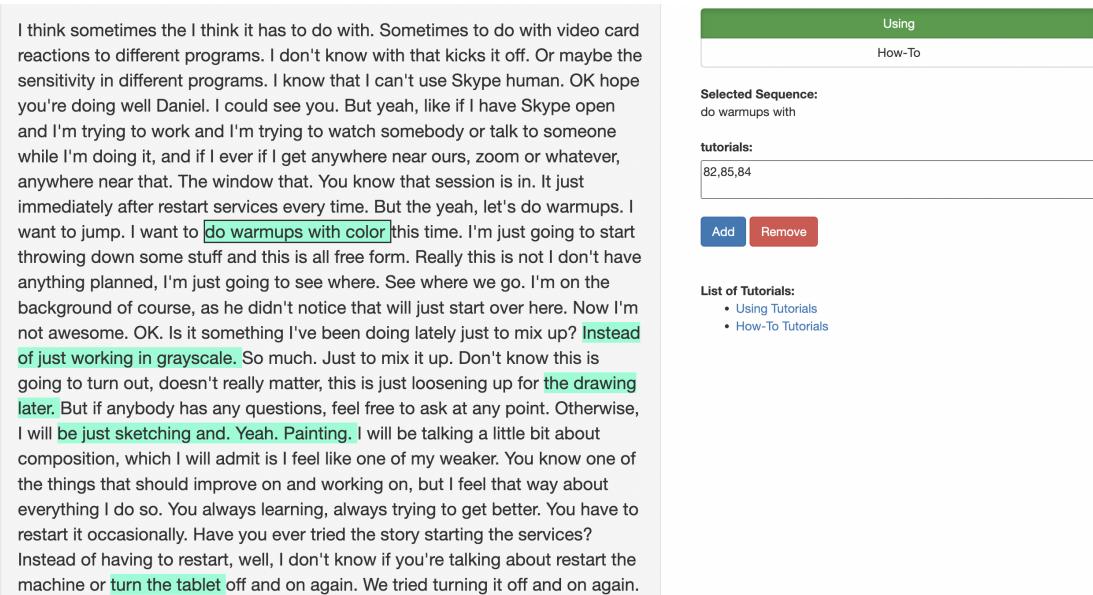


Figure 2: Annotation tool for phrase-level tutorial recommendation

tutorials. More specifically, for the tutorial pool, we employ two types of tutorials available for Photoshop:

- **Using:** In this type of tutorial the usage of the specific tools is discussed. These tutorials are helpful to discuss the details of the tools that are used by the streamer. An example of this type of tutorial is presented in Figure 3. In this work, we employ 290 Using tutorials.
- **How-To:** In this type of tutorial, the process to achieve a final edit action is discussed. For instance, how to design a portrait could be an edit action discussed in a tutorial. For this type of tutorial, multiple editing tools might be employed. An example of this tutorial is presented in Figure 4. In this work, we employ 126 How-To tutorials.

For every transcript, the annotator selects the sentences that might refer to a tutorial and provide at least three tutorial related to the selected sentence. In total, 4,126 sentences with 3 tutorials are annotated.

4. Model

We employ two types of models: (1) An unsupervised model: In this model, the content of the transcript and the tutorial are employed for linking. This method provides a tutorial for the entire transcript; (2) A supervised model: In this model, we employ the annotated data to train a sentence classification model in which the model selects one of the available tutorials for a given sentence. If the sentence does not refer to a tutorial, *None* is selected as the sentence label. The rest of this section provides details of these two models.

Learn the basics of working with layers in Photoshop on the iPad.

Layers contain the images, fonts, and objects that make up a layered file. Layers enable you to move, edit, and work with content on one layer without affecting the content within your other layers. Use layers to perform tasks such as compositing multiple images, adding text to an image, and applying filters and adjustments.

Layers are arranged in a stack and can be viewed by tapping either the compact or detailed layer view from the taskbar in the workspace.

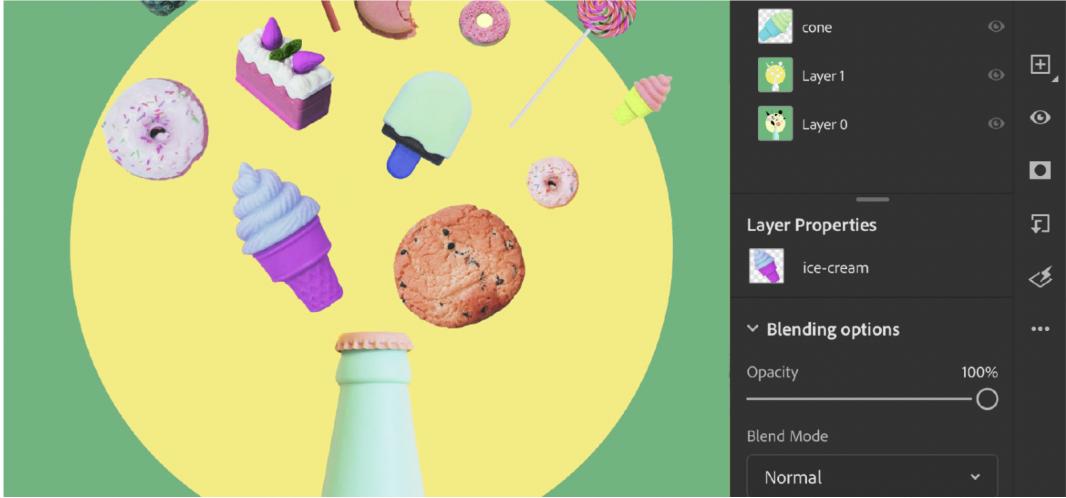


Figure 3: A Using tutorial. The tutorial discusses the application of layers.

4.1. Unsupervised Model Overview

Our proposed model has the following novelties:

- A novel unsupervised deep learning model for the task of tutorial recommendation using video transcripts
- A novel approach to employing domain-specific knowledge for filtering the target tutorials
- A novel method for summarizing the input transcript into a smaller version without using any human-curated labels
- A novel method for computing the similarity between the transcript and the tutorial text based on discourse-level consistency.

4.2. Details

Formally, the input to the system is the transcript $D = [w_1^D, w_2^D, \dots, w_n^D]$, consisting of n words, and a pool of tutorial textual content, i.e., $P = [T_1, T_2, \dots, T_{|P|}]$ where $T_i = [w_1^T, w_2^T, \dots, w_m^T]$ is the textual content of i -th tutorial consisting of m words. The goal is to return the most relevant tutorial from the pool P , i.e., T_{gold} . To create a system for this task, we propose multiple components. Specifically, the proposed system consists of three major components:

How to make a photo composite in Adobe Photoshop

Combining elements from multiple photographs into one image allows you to create something new and unexpected. In this quick tutorial, learn how visual artist Temi Coker combines two photos in Adobe Photoshop to create an image that stretches the imagination.

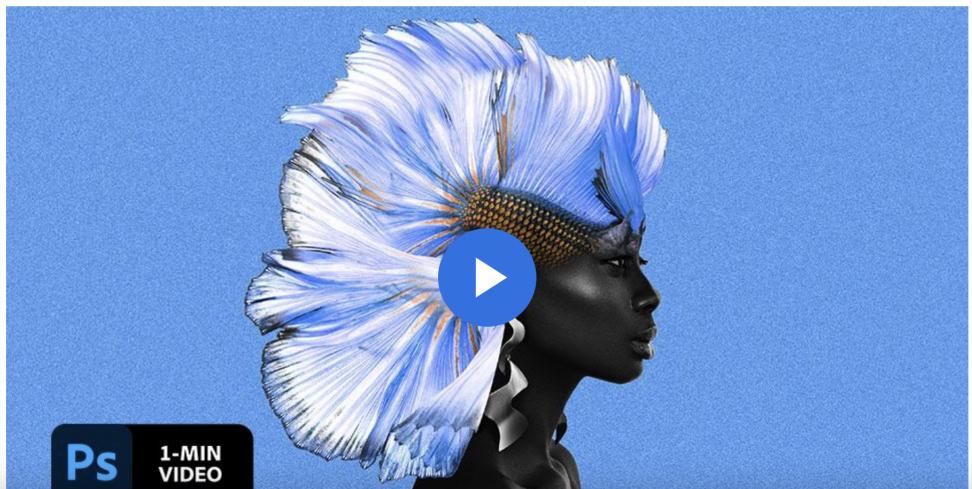


Figure 4: A How-To tutorial. The tutorial discusses the process of creating a composite image.

- Filtering Tutorials: In this component, the goal is to remove the tutorials that are very unlikely to be relevant to a given transcript. In this component, domain-specific knowledge is employed to assess the relevancy between the given transcript and the tutorial textual content.
- Transcript summarizing: In this component, the objective is to summarize the given transcript such that only the most important information that could help find the tutorials is preserved. This component employs an unsupervised deep learning method.
- Tutorial Ranking: Finally, using the summary of the given transcript and the filtered list of relevant tutorials, this component employs various metrics to sort the tutorials based on their relevance and similarity to the transcript.

The rest of this section elaborates more on the details of each of these components.

4.2.1. Filtering Tutorials

The pool of tutorials might contain several irrelevant candidates which could hinder an efficient ranking method. As such, it is necessary to first filter the pool P such that the unlikely candidates are removed. Formally, the objective is to define the filter function Φ such that:

$$\begin{aligned} |\Phi(P)| &< |P| \\ \forall T_i \in \Phi(P), T_j \in P - \Phi(P) \quad R(T_i) &< R(T_j) \end{aligned} \tag{1}$$

where $|\cdot|$ is the size of the pool and $R(x)$ is the rank of x in the sorted pool based on the relevance of the candidates to the given transcript D . To define the function Φ , in a novel method, we propose to employ two types of criteria:

Domain-specific Knowledge: The ontology of names in the domains of interest (i.e., in our work, we use Image Editing Tools such as Adobe Photoshop as the domain of interest) could be employed as the domain-specific knowledge. Specifically, in this work, we propose to employ the available list of tool names for Adobe Photoshop as domain-specific information. To employ this knowledge in function Φ , we suggest first finding the tool names mentioned in the transcript D , i.e., $TN = \{tn_1, tn_2, \dots, tn_{|TN|}\}$ where tn_i is a tool name in transcript D . Next, we define the filter function Φ_{DK} to filter out those tutorials in the pool P which don't mention one of the tool names of TN :

$$\Phi_{DK}(T_i) = \begin{cases} True & \text{if } \exists tn_j \in TN \text{ s.t. } tn_j \in T_i \\ False & \text{otherwise} \end{cases} \tag{2}$$

String Similarity: In addition to the domain-specific knowledge, we seek to incorporate string similarity of the tutorial textual content with the transcript D in the filtering process. More specifically, we first compute the string similarity of the transcript D with tutorial $T_i \in P$ using normalized point-wise mutual information (PMI):

$$\begin{aligned} Sim(D, T_i) &= \sum_{w_i^D \in D} \sum_{w_j^T \in T_i} \frac{PMI(W_i^D, W_j^T)}{n * m} \\ PMI(w_i, w_j) &= \log \frac{COUNT(w_i, w_j)}{COUNT(w_i) * COUNT(w_j)} \end{aligned} \tag{3}$$

where n and m are the number of words in the transcript D and tutorial T_i , $COUNT(W_i)$ is the number of occurrences of the word w_i in all transcripts in the training data and $COUNT(w_i, w_j)$ is the number of occurrences of both words w_i and w_j in a transcript in the training data. Next, we define the following filter function based on similarity:

$$\Phi_{sim}(T_i) = \begin{cases} True & \text{if } Sim(D, T_i) > \delta \\ False & \text{otherwise} \end{cases} \tag{4}$$

where δ is a hyper-parameter to be tuned using development data. Finally, using the two aforementioned filter functions Φ_{DK} and Φ_{sim} , we define the final function Φ :

$$\Phi(T_i) = \begin{cases} True & \text{if } \Phi_{DK}(T_i) \text{ and } \Phi_{sim}(T_i) \\ False & \text{otherwise} \end{cases} \tag{5}$$

4.2.2. Transcript Summarizer

The next component in our system is the transcript summarizer whose goal is to shorten the transcript D such that only the distinctive information is preserved and the redundant portions, which might not be helpful to identify the most related tutorial, are excluded. To this end, we proposed to train a deep learning model to consume the input document D and generate the shorter document D' , such that $|D'| < |D|$. Specifically, a transformer-based language model, i.e., BERT, is trained to encode the words of the document D , i.e., $e_i = \text{BERT}(w_i)$ for all $w_i \in D^2$. In our experiments, we use the last hidden states of the BERT model to obtain the embedding vector e_i . Next, a feed-forward network is utilized to estimate the likelihood of the word w_i to be included in the shorter document D' :

$$P(w_i|D) = FF(w_i) = \sigma(W_1 * e_i + b_1) \quad (6)$$

where σ is the sigmoid activation function, W_1 and W_2 are the weight matrices and b_1 and b_2 are bias. To train the BERT model and the feed-forward layer, since the available resources for this task do not have any labeled data, we resort to unsupervised learning. Concretely, two criteria are employed to train the model:

Distinctiveness: The shorter documents D'_i and D'_j obtained from the original documents D_i and D_j should be as different as possible. Moreover, those portions of the documents D_i and D_j that do not appear in the summary should have the least differences (in other words, as these portions are not informative (e.g., they are chitchat), there will be more similarity between them). To fulfill this requirement, we employ the following loss function:

$$\mathcal{L}_{dist} = \alpha * (\sigma(H'_i) \odot \sigma(H'_j)) - \beta(\sigma(H''_i) \odot \sigma(H''_j)) \quad (7)$$

where σ is the softmax function, α and β are the trade-off parameters and \odot is the hadamard product. The vectors H'_i , H'_j , H''_i and H''_j are the vector representation for the summaries D'_i , D'_j , and the portions of the documents D_i and D_j not included in the summaries, i.e., D''_i and D''_j , respectively. To obtain these vector representations, we use max-pooling on the multiplication of the embedding vectors and the feed-forward network. For instance, H'_i is obtained as follows:

$$\begin{aligned} H'_i &= \text{MAX_POOL}(h_1, h_2, \dots, h_n) \\ h_k &= e_k * FF(e_k) \end{aligned} \quad (8)$$

where n is the number of words in the document D . Note that for the representations of H''_i and H''_j , we replace $FF(e_k)$ with $(1 - FF(e_k))$ in equation 8.

Information Retaining: The process to summarize document D into the smaller version D' is supposed to keep the most important information in D intact. As such, it is expected that the information available in both documents D and D' to have considerable overlap. This criterion can be achieved by increasing the mutual information (MI) between the representations of the D and D' . To fulfill this goal, we exploit contrastive learning. In particular, a discriminator is trained to distinguish positive samples from negative ones where a positive sample is the

²Note that for words consisting of multiple word-pieces, we represent them using the average of their word-piece embeddings obtained from BERT model

concatenation of the representation of the original document D_i and its summary D'_i , i.e., $pos = [H_i : H'_i]$, and the negative sample is the concatenation of the representation of the document D_i with the summary of the randomly selected document D_j , i.e., $neg = [H_i : H'_j]$. Formally, the following loss function is employed to increase the mutual information:

$$\mathcal{L}_{IR} = -(log(\Psi[H_i : H'_i]) + log(1 - \Psi([H_i : H'_j]))) \quad (9)$$

where Ψ is the discriminator. The sum of two losses, i.e., \mathcal{L}_{dist} and \mathcal{L}_{IR} is employed as the final loss function to train the transcript summarizer component:

$$\mathcal{L} = \alpha \mathcal{L}_{IR} + \beta \mathcal{L}_{dist} \quad (10)$$

where α and β are trade-off parameters. At inference time, to summarize the input transcript, we employ the output of the feed-forward network FF , and every word w_i whose corresponding value from FF is higher than a threshold is selected in D' .

4.2.3. Ranking

Finally, given the short document D' and the filtered list of tutorials, i.e., $P' = \Phi(P)$, the final component aims to sort the tutorials based on their relevance to the given transcript. To this end, we employ two types of scores:

- String Similarity: The string similarity between every tutorial $T_i \in P'$ with the summary D' is evaluated using FastText [6] to obtain its string similarity score: $Score_{str} = FastText(T_i, D')$
- Discourse Similarity: The objective of this score is to measure how likely is the tutorial $T_i \in P'$ to complement the summary D' of the input transcript. To this end, first, we train the text classification model C that takes the concatenation of the first and second half of the transcripts D_i , i.e., $D_{i,1}$ and $D_{i,2}$, as a positive sample and the concatenation of $D_{i,1}$ and $D_{j,2}$, where j is selected randomly, as the negative samples³. The model is trained using a similar loss function as equation 9. Next, the trained classifier C is employed in the ranking component by feeding the concatenation of the tutorial T_i and summary D' to the model and its output (i.e., the likelihood of the input to complement each other), is employed as the discourse-level score: $Score_{disc} = C([T_i : D'])$

Finally, to sort the documents, we compute the sum of the two aforementioned scores:

$$Score = Score_{sim} + Score_{disc} \quad (11)$$

The sorted list of the tutorials is returned as the final output of the system.

³words of the documents are encoded using GloVe embedding and the max-pooled embeddings of them are fed into classifier C

Model	Hit@3	Hit@5
String Similarity Sorting	40%	50%
Keyword Sorting	35%	45%
Information-based Sorting	40%	50%
Ours	55%	65%

Table 1

Performance of the unsupervised models

Model	F1
BERT	35%
GloVe	28%

Table 2

Performance of the supervised models

4.3. Supervised Model Overview

To train the model for supervised tutorial linking, we model the task as a sentence classification problem. More specifically, given the words of the sentence S and the title of the tutorial T , the input sequence $[CLS], w_1^S, w_2^S, \dots, w_n^S, [SEP], w_1^T, w_2^T, \dots, w_m^T, [SEP]$ is fed into the BERT_{base} model. Note that for every tutorial in the tutorial pool a separate input is prepared. Next, the $[CLS]$ vector representation is fed into a two-layer feed-forward layer to predict a binary label, e.g., 1/0. The label is 1 if the tutorial is linked with the given sentence in the annotated dataset. Note that we add a special tutorial title *None* for sentences without any linked tutorial.

4.4. Experiments

To evaluate the proposed system, we manually annotated transcripts from the Behance corpus. These are the transcripts of videos streamed on Behance.net and the streamers are all using Adobe Photoshop in their streaming video. We use Adobe Photoshop tutorials (more than 200 tutorials for using or how-to) as the initial pool of tutorials. To provide more insight into the performance of the proposed system, we compare it with the following systems:

- String Similarity Sorting: In this system, the string similarity of the input transcript and the tutorials are measured and it is employed to sort all tutorials in the pool.
- Keyword Sorting: In this system, the tutorials are sorted based on the number of tool names that they have in common with the input transcript.
- Information-based Sorting: In this system, the same PMI-based scoring that is employed in our filtering component is employed to sort all tutorials.

For the supervised model, in addition to the proposed BERT model, we also compare the performance of the model when the words of the sentence S and the title of the tutorial T are presented by GloVe embedding. The max-pooled representation of the input is fed into the feed-forward network.

To evaluate the models, we use Hit@3 and Hit@5 evaluation metrics. The results are shown in Table 1. This table clearly shows that the proposed model significantly outperforms the systems, indicating its effectiveness for this task.

The results for the supervised model are presented in Table 2. This table shows that the contextualized representation of the video transcript and the title are more effective than the GloVe embedding. Nonetheless, both models suffer from low performance which indicates more research is required.

References

- [1] Z. Guan, C. Wang, J. Bu, C. Chen, K. Yang, D. Cai, X. He, Document recommendation in social tagging services, in: Proceedings of the 19th international conference on World wide web, 2010, pp. 391–400.
- [2] D. Kim, C. Park, J. Oh, S. Lee, H. Yu, Convolutional matrix factorization for document context-aware recommendation, in: Proceedings of the 10th ACM conference on recommender systems, 2016, pp. 233–240.
- [3] X. Xu, A. Hassan Awadallah, S. T. Dumais, F. Omar, B. Popp, R. Rounthwaite, F. Jahanbakhsh, Understanding user behavior for document recommendation, in: Proceedings of The Web Conference 2020, 2020, pp. 3012–3018.
- [4] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, arXiv preprint arXiv:1509.01626 (2015).
- [5] O. Shahmirzadi, A. Lugowski, K. Younge, Text similarity in vector space models: a comparative study, in: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE, 2019, pp. 659–666.
- [6] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, arXiv preprint arXiv:1607.04606 (2016).

A. Tutorials

A list of Using and How-To tutorials employed during the annotation are shown in Figures 5 and 6. In our work, in total, 290 and 126 Using and How-To tutorials are employed, respectively.

B. Case Study

In order to provide more insight into the performance of the proposed supervised model, we show the predicted tutorials for the sentence “*I’m gonna work the composition of this and then work on cleaning up the drawing a little bit*” in the paragraph “*Right there. I get it. OK, I get it. I will do this real quick as the last thing. Change this brush. Might get a little bit right there. Something like that. Anyway. I don’t know. So this is what we did for warmups. This guy in this lady. Will leave it at that. But now we will jump back into ours. A party planning here. They actually took a little bit longer than I want to be out about an hour and a half hour and 1520 minutes left. I’m gonna work on the composition of this and then work on cleaning up*

1. [3d Painting Photoshop \(Using\)](#)
2. [3d Panel Settings Photoshop Extended \(Using\)](#)
3. [3d Panel Settings \(Using\)](#)
4. [3d Rendering Saving Photoshop Extended \(Using\)](#)
5. [3d Rendering Saving \(Using\)](#)
6. [3d Texture Editing \(Using\)](#)
7. [3d Workflow Cs6 \(Using\)](#)
8. [ScriptUI \(Using\)](#)
9. [Acquiring Images Cameras Scanners \(Using\)](#)
10. [Actions Actions Panel \(Using\)](#)
11. [Adaptive Wide Angle Filter \(Using\)](#)
14. [Add Edit Text Ipad \(Using\)](#)
15. [Add Edit Text \(Using\)](#)
16. [Add Lighting Effects! \(Using\)](#)
17. [Add Manage Edit Layers \(Using\)](#)
20. [Adding Color Paths \(Using\)](#)
21. [Adding Conditional Mode Change Action \(Using\)](#)
22. [Adding Dynamic Elements Brushes \(Using\)](#)
23. [Adding Swatches Html Css Svg \(Using\)](#)
24. [Adjust Color Tone Levels Curves \(Using\)](#)
26. [Adjust Shadow Highlight Detail \(Using\)](#)
27. [Adjust Vibrance \(Using\)](#)
28. [Adjusting Color Tone Cs6 \(Using\)](#)
29. [Adjusting Crop Rotation Canvas \(Using\)](#)
30. [Adjusting Hdr Exposure Toning \(Using\)](#)
31. [Adjusting Hue Saturation \(Using\)](#)
32. [Adjusting Image Sharpness Blur \(Using\)](#)
33. [Adjusting Pixel Selections \(Using\)](#)
34. [Adjustment Fill Layers \(Using\)](#)
36. [Adobe Color Themes \(Using\)](#)
37. [Adobe Dng Converter \(Using\)](#)
38. [Adobe Stock \(Using\)](#)
41. [Aligning Layers \(Using\)](#)
45. [App Settings Ipad \(Using\)](#)
46. [Apply Brightness Contrast Adjustment \(Using\)](#)
48. [Apply Pattern To Template \(Using\)](#)
49. [Applying Color Balance Adjustment \(Using\)](#)
50. [Applying Smart Filters \(Using\)](#)
51. [Applying Special Color Effects Images \(Using\)](#)
52. [Applying Specific Filters \(Using\)](#)
53. [Arabic Hebrew \(Using\)](#)

Figure 5: A sample list of Using tutorials

the drawing a little bit. Will pull this up. I think I still have it. No, it's not in here. Just to kind of recap. Z. Well, let's see. Let's see if I can find this cleaning up the drawing ". The results are shown in Figure 7, 8 and 9.

0. [3d Model Wrapping \(How-To\)](#)
12. [Add Border Frame Around Photo \(How-To\)](#)
13. [Add Color Fill Layers \(How-To\)](#)
18. [Add Motion Blur Effects \(How-To\)](#)
19. [Add Text Photo \(How-To\)](#)
25. [Adjust Correct Color Balance Photoshop \(How-To\)](#)
35. [Adjustment Layer \(How-To\)](#)
39. [Align Objects Guides \(How-To\)](#)
40. [Align Space Objects \(How-To\)](#)
42. [Amp Up Facebook Profile \(How-To\)](#)
43. [Animate Illustrations \(How-To\)](#)
44. [Animated Gradient \(How-To\)](#)
47. [Apply Filter Effects \(How-To\)](#)
59. [Black And White With Color Photo \(How-To\)](#)
61. [Blur Background For Focal Point \(How-To\)](#)
63. [Brighten A Photo \(How-To\)](#)
64. [Brighten Photo Basics \(How-To\)](#)
74. [Circular Pixel Stretch Effect \(How-To\)](#)
88. [Combine Images \(How-To\)](#)
89. [Combine Pictures \(How-To\)](#)
92. [Composite Add Move Images \(How-To\)](#)
93. [Composite Color Blend \(How-To\)](#)
94. [Composite Images With Layer Blend Mode \(How-To\)](#)
95. [Composite Multiple Images Create Collage \(How-To\)](#)
96. [Composite Nature Photography With Layer Blend Mode \(How-To\)](#)
97. [Composite Photo \(How-To\)](#)
98. [Compositing \(How-To\)](#)
101. [Content Aware Crop Fill \(How-To\)](#)
103. [Content Aware Hide Objects \(How-To\)](#)
110. [Correct Exposure Photoshop \(How-To\)](#)
113. [Create Animated Gif \(How-To\)](#)
114. [Create Animated Self Portrait \(How-To\)](#)
115. [Create Composite Images \(How-To\)](#)
116. [Create Contact Sheets \(How-To\)](#)
117. [Create Digital Painting \(How-To\)](#)
119. [Create Dynamic Text Effects \(How-To\)](#)
123. [Create Poster \(How-To\)](#)
124. [Create Realistic Effects \(How-To\)](#)

Figure 6: A sample list of HowTo tutorials

Change brush size and opacity with pen pressure

If you work with a graphics drawing tablet, such as the Wacom® tablet, you can control painting tools with pen pressure, angle, rotation, or the stylus wheel.

- 1 Select the Brush , Pencil , or other painting tool.
- 2 In the options bar, do either of the following:
 - Click the Tablet Pressure Controls Size button .
 - Click the Tablet Pressure Controls Opacity button .

Note:

Choose Window > Brush to access additional controls that vary the angle, flow, scatter, texture depth, and roundness of the stroke according to pen pressure.



Figure 7: A screenshot of the tutorial page for the sentence shown in the Case Study section.

About painting tools, presets, and options

Adobe Photoshop provides several tools for painting and editing image color. The Brush tool and the Pencil tool work like traditional drawing tools applying color with brush strokes. Tools like the Eraser tool, Blur tool, and Smudge tool modify the existing colors in the image. In the options bar for each of these painting tools, you can set how color is applied to an image and choose from preset brush tips. See [Painting tools gallery](#).

Brush and tool presets

You can save a set of brush options as a preset so you can quickly access brush characteristics you use frequently. Photoshop includes several sample brush presets. You can start with these presets and modify them to produce new effects. Many original brush presets are available for download on the web.

You can quickly choose presets from the Brush Preset picker in the options bar, which lets you temporarily modify the size and hardness of a brush preset.

Save tool presets when you want to store customized brush tip characteristics along with settings from the options bar such as opacity, flow, and color. To learn more about tool presets, see [Create and use tool presets](#).

Brush tip options

Along with settings in the options bar, brush tip options control how color is applied. You can apply color gradually, with soft edges, with large brush strokes, with various brush dynamics, with different blending properties, and with brushes of different shapes. You can apply a texture with your brush strokes to simulate painting on canvas or art papers. You can also simulate spraying paint with an airbrush. You use the Brush Settings panel to set brush tip options. See [Brush Settings panel overview](#).

If you work with a drawing tablet, you can control how color is applied using pen pressure, angle, rotation, or a stylus wheel. You set options for drawing tablets in the Brush Settings panel and options bar.

Figure 8: A screenshot of the tutorial page for the sentence shown in the Case Study section.

When you start Photoshop, the Tools panel appears at the left of the screen. Some tools in the Tools panel have options that appear in the context-sensitive options bar.

You can expand some tools to show hidden tools beneath them. A small triangle at the lower right of the tool icon signals the presence of hidden tools.

You can view information about any tool by positioning the pointer over it. The name of the tool appears in a tool tip below the pointer.

For a pictorial overview of the different tools in Photoshop, see [Tool galleries](#).

Select and display tools

Select a tool

1 Do one of the following:

- Click a tool in the Tools panel. If there is a small triangle at a tool's lower right corner, hold down the mouse button to view the hidden tools. Then click the tool you want to select.
- Press the tool's keyboard shortcut. The keyboard shortcut is displayed in its tool tip. For example, you can select the Move tool by pressing V.

Note:

Pressing and holding a keyboard shortcut key lets you temporarily switch to a tool. When you let go of the shortcut key, Photoshop returns to the tool you were using before the temporary switch.



Figure 9: A screenshot of the tutorial page for the sentence shown in the Case Study section.