

An Investigation on Automatically Assessing an Application Tutorial's Difficulty

by

Shahed Anzarus Sabab

A thesis submitted to The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements of the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
November 2019

© Copyright 2019 by Shahed Anzarus Sabab

Abstract

Online step-by-step tutorials play an integral role in how users learn feature-rich software applications (e.g., Photoshop, AutoCAD, Fusion360). However, when searching for a tutorial, users can find it difficult to assess whether a given tutorial is designed for their level of software expertise. Novice users can struggle when a tutorial is out of their reach, whereas more advanced users can end up wasting time with overly simple, first-principles instruction. To assist users in selecting tutorials based on expertise, I investigate the feasibility of using machine learning techniques to automatically assess and label a tutorial's difficulty level. Using Photoshop as a testbed, I develop a set of distinguishable tutorial features and use these features to train a classifier that can label a tutorial as either Beginner or Advanced with 85% accuracy. To illustrate a potential application of my classifier, I developed a tutorial selection interface called *TutVis*. *TutVis* annotates each tutorial with its difficulty level, along with visual representations of other tutorial features that contribute to this difficulty assessment. An initial evaluation comparing *TutVis* to two other interfaces (which varied in the number of different tutorial features displayed) showed a strong preference for and use of *TutVis*'s novel features.

Table of Contents

Abstract	i
Table of Contents	iii
List of Figures	vii
List of Tables	ix
Acknowledgements	xi
Chapter 1 – Introduction	1
1.1. Research Questions	2
1.2. Methodology and Approach	2
1.2.1. Investigating Differentiable Features	3
1.2.2. Model Generation and Evaluation	3
1.2.3. Development of the Prototype	4
1.2.4. Tutorial Selection Study	4
1.3. Contributions	4
Chapter 2 – Related Work	7
2.1. Characterizing and Classifying Software Expertise	7

2.2. Detection of Expertise.....	8
2.3. Improving the Usability of Software Tutorials	9
2.4. Summary	11
Chapter 3 – Investigating Differentiable Features	13
3.1. Data Collection	14
3.2. Data Preprocessing.....	15
3.3. Feature Engineering	17
3.3.1. Tutorial Topics.....	17
3.3.2. Command Ratio (CR)	20
3.3.3. Word Repetition (WR).....	20
3.3.4. Text Difficulty (TD)	21
3.3.5. Tutorial Length (Len)	21
3.4. Feature Analysis between Advanced vs Beginner Tutorials	21
3.5. Summary	22
Chapter 4 – Model Generation and Evaluation.....	25
4.1. Impact of Individual Feature on Classifier Accuracy	26
4.2. Impact of Combining Feature Sets on Classifier Accuracy.....	27
4.3. Impact of Number of Training Samples on Classifier Accuracy.....	28
4.4. Generalizing to 3D Modeling Tutorials.....	29
4.5. Summary	30

Chapter 5 – Development of the Prototype.....	31
5.1. Transforming Text Difficulty, Length, Word Repetition, Command Ratio into Interface Components	32
5.2. Transforming Topics into Interface Components	33
5.2.1. Generating Tutorial Clusters.....	35
5.2.2. Analyzing Tutorial Clusters to Generate Labels.....	35
5.2.3. Advanced Vs Beginner Topics: Some High-Level Differences	43
5.3. <i>TutVis</i> : Tutorial Selection Interface	45
5.4. Summary	46
Chapter 6 – Tutorial Selection Study.....	49
6.1. Participants.....	49
6.2. Study Conditions and Tutorials	50
6.3. Procedure	52
6.4. Results.....	54
6.4.1. Subjective Response: Preferences and Confidence Levels.....	54
6.4.2. Individual Component Usage	55
6.4.3. Perspective on the Utility of Individual Components.....	57
6.5. Summary	60
Chapter 7 – Conclusion.....	61
7.1. Contributions.....	62

7.2. Limitations and Future Research Directions.....	63
7.2.1. Developing a Recommender System	63
7.2.2. Investigating Other Features	63
7.2.3. Reducing the Impact of Misclassification.....	64
7.2.4. Automating Manual Effort.....	64
7.2.5. Exploring the Long Term Effect.....	65
Bibliography	67
Appendix A – Clear Topics	83
Appendix B – Similar Topics	84
Appendix C – Fuzzy Topics	85
Appendix D – Evolution of the Tutorial Representation	86
Appendix E – Research Ethics Board Approval.....	87
Appendix F – TCPS 2: CORE Certificate	88
Appendix G – Poster Advertising the Study.....	89
Appendix H – Consent Form	90
Appendix I – Instructions for Different Prototypes	92
Appendix J – Isomorphic Scenarios for Tutorial Selection Tasks.....	93
Appendix K – Demographics Questionnaire	94
Appendix L – Study Questionnaire.....	95
Appendix M – Semi-Structured Interview Sample Questions.....	98

List of Figures

Figure 1: Different Stages of the feature investigation.....	14
Figure 2: Distribution of the collected Photoshop Tutorials.....	15
Figure 3: Different Preprocessing Steps	16
Figure 4: Process of developing the command dictionary.....	18
Figure 5: An example of LDA topic model output for 3 sample topics.	19
Figure 6: Model Performance using individual features.....	26
Figure 7: Model performance using combined features.	27
Figure 8: Learning Curves. *error bar represents s. d.....	28
Figure 9: An example of LDA topic model output for 3 sample topics. The shaded fields represent dominant topic.....	34
Figure 10: Topic Labeling using Tutorial Clusters.....	35
Figure 11: Sample code of the tutorials of topic 5.....	36
Figure 12: General concept of labeling different topics (i.e., tutorial clusters)	40
Figure 13: A general concept of classifying advanced vs beginner tutorial clusters (i.e., Photoshop).	43
Figure 14: The <i>TutVis</i> interface, which presents a list of tutorials with difficulty (A), title (B), thumbnail image (C), topics covered (D), length, text difficulty, commands usage (E)	

and most frequently used tools (F). TutVis also provides filtering options (G,H) and a search bar (I) 45

Figure 15: The *baseline* interface, which presents a list of tutorials with title (B), thumbnail image (C), and most frequently used tools (F). This interface also provides a search bar (I) 50

Figure 16: The *TutDiff* interface, which presents a list of tutorials with difficulty (A), title (B), thumbnail image (C), and most frequently used tools (F). *TutDiff* also provides filtering options (G) and a search bar (I) 51

Figure 17: Self-reported interface components used. 55

Figure 18: Interface components used in the different tasks according eye-gaze and think-aloud data (in TutVis only) 56

List of Tables

Table 1: Mean differences between Advanced vs Beginner tutorials.....	22
Table 2: Different representation of Text Difficulty.....	32
Table 3: One set of tutorial selection scenarios	53

Acknowledgements

I am grateful to the almighty God for giving me the patience to pursue my higher degree while staying away from my family for over two years.

I would like to thank Dr, Andrea Bunt, for her constant support and encouragement throughout the entire time of my MSc program at the University of Manitoba. Following her guidance, I have developed self-confidence, time management, and skills to enhance productivity. I would also like to thank her for the financial support during my stay. I extend my gratitude to my thesis committee members, Dr. Danny D. Mann and Dr. Olivier Tremblay-Savard for their precious time and feedback.

I am thankful to all of my HCI Lab friends and Dr. James Young for their support of all kinds. I feel privileged because of being surrounded by the positive energy of the HCI Lab which never let me feel alone. Thank you, Adnan, Ananta, Anik, Annalena, Cheng, Chris, Dan, Denise, Diljot, Ellie, Lena, Lorena, Mahya, Patrick, Rahat, Raquel, Stela, Taylor, and Volodymyr.

Last but not the least, I want to thank my family, especially my parents, who always believed in me and supported me in my hard time.

Chapter 1

Introduction

Online tutorials have emerged as one of the most popular and heavily used resources for learning and using feature-rich software applications (e.g., Autocad, Photoshop, Fusion360, etc.). [7,50]. There is an abundance of tutorials online (e.g., over 28,160 video & text tutorials on the popular aggregator site tutplus.com) and, in comparison to other resources like forums or Q&A sites, they typically describe full workflows, illustrating the step-by-step progression of a task.

Despite the benefits offered by online tutorials, it can be difficult for users to locate and identify tutorials that are appropriate for their current level of software expertise [22,36,80]. For example, advanced tutorials often assume certain software skills and knowledge of the application's vocabulary [25,31]. When a novice tries to follow a tutorial with this assumed knowledge, s/he can experience cognitive overload [53,62], frustration [49], and limited task success [39]. Expert users, on the other hand, are more interested in compact workflow representations, and in tutorials that cover more advanced or novel techniques [31,39,46].

Existing online tutorials often fail to provide expertise or difficulty information to guide a user's search for an appropriate tutorial. For example, when sampling from over 8,000 Photoshop tutorials on tutplus.com, I found that only 8% provided the user with any difficulty information. To address this problem, I investigate whether a system could classify a tutorial's difficulty automatically. Given the highly structured nature of many feature-rich tutorials, with their step-based [50], and command-oriented workflows [42], my approach relies on machine learning to uncover properties of advanced vs. beginner tutorials.

1.1. Research Questions

The goal of my thesis is to investigate the feasibility of automatically labeling the tutorial's difficulty using machine learning techniques. While doing the investigation, I had the following research questions:

- 1) What are the features that differentiate tutorials designed for experts from tutorials designed for novices?
- 2) How can I develop a machine learning model that can automatically classify the tutorial's difficulty levels?
- 3) How can I leverage the developed model to assist users in the tutorial selection?

1.2. Methodology and Approach

Using Photoshop tutorials as the testbed, I approached my research questions by i) investigating and extracting differentiable features of advanced vs beginner tutorials ii) training different machine learning models using the extracted features of the tutorials and

evaluating different models' performance, iii) developing a prototype which presents the model's predicted difficulty level along with the visual representation of the extracted features, and iv) elicit users' response on the prototype from a tutorial selection study. What follows is a summary of each of these thesis components.

1.2.1. Investigating Differentiable Features

I started my investigation by consulting prior research on measuring software expertise [30,36,46] and learnability [31,39,54]. Initially collecting Photoshop tutorials, I identified and engineered a set of differentiable features that I extracted from the tutorial's text. Here, my analysis included both video and text tutorials; however, in the case of video tutorials, I only considered textual transcripts. Finally, after my analysis, I settled upon five different feature sets including topics, length, text difficulty, word repetition, and the density of command references.

1.2.2. Model Generation and Evaluation

I investigated the impact of the extracted features on classifier accuracy. Specifically, I trained different models using 750 tutorials with existing difficulty labels (obtained from 9 online tutorial repositories) using different feature combinations. Using 10-fold cross-validation, I found that the best model achieves an accuracy of 85% when classifying an arbitrary tutorial as either beginner or advanced. From another investigation, I uncovered that this performance could be improved by introducing more training data to the model. I also evaluated the generalizability of the feature sets to the second type of feature-rich software, 3D modeling software (e.g., Fusion 360).

1.2.3. Development of the Prototype

To illustrate a user-centered application of the classifier, I created a prototype tutorial browsing interface called *TutVis*. *TutVis* aims to guide tutorial selection by annotating each tutorial with its automatically generated difficulty label, along with interface components that summarize other tutorial features (i.e., those leveraged by the classifier). To present the tutorial features to a user, such as topics, length, command ratio, text difficulty, word repetition, I presented an approach (i.e., refined by a series of pilot testing) to transform the features into interface components of *TutVis*.

1.2.4. Tutorial Selection Study

In a proof-of-concept user evaluation with 12 participants, I compared *TutVis* to two other tutorial selection interfaces that displayed subsets of the annotations (e.g., only the difficulty labels). The results suggest that participants prefer having *TutVis*'s full set of interface components and that they use the interface components to increase their selection confidence.

1.3. Contributions

In summary, this thesis contributes to the following:

- 1) I identify and investigate features (e.g., topic, length, text difficulty) that differentiate feature-rich software tutorials designed for experts from those designed for beginners.
- 2) I illustrate that these features can be leveraged by a machine-learning model for an 85% classification accuracy.

- 3) I show how the classifier’s decision and its features can be interpreted (in particular, the machine-generated topics) and presented through the *TutVis* system.
- 4) I provide initial insight from a proof-of-concept evaluation on how *TutVis* impacts tutorial selection tasks.

The remainder of this thesis is organized in six chapters: Chapter 2 summarizes prior work related to this thesis, Chapter 3 describes the investigation process of differentiable features, Chapter 4 describes the model generation and evaluation process, Chapter 5 discusses the development of the prototype, Chapter 6 summarizes the tutorial selection study and Chapter 7 concludes the thesis.

Chapter 2

Related Work

The coverage of my related work focuses on three main areas: characterizing software expertise, detection of expertise, and improving the usability of software tutorials.

2.1. Characterizing and Classifying Software Expertise

Earlier research has acknowledged that the detection of individual differences can significantly improve software learning and task efficiency [16,19,20]. In software learning research, one difference that has received recent attention is the study of user expertise [3,30,31,39]. Ericsson et al. defined user expertise as “the characteristics, skills, and knowledge that distinguish experts from novices and less experienced people” [24]. Based on this definition Grossman et al. defined software expertise as “The characteristics, skills, and knowledge that distinguish experts from novices, considered across the entire scope of functionality that the software provides” [30].

Prior work has recognized the wide range of expertise that users bring to their experiences with feature-rich software. Building on Nielsen's categorization of general user interface expertise [60], Grossman et al. classified feature-rich software expertise according to the following dimensions: experience with computers, experience with the software's interface, domain knowledge and experience with similar software [31]. Moreover, considering the familiarity, frequency, and efficiency of software usage, Grossman et al. presented low-level metrics of four different expertise profiles: core expert, isolated expert, naïve expert and knowledgeable expert [30].

Guided by the prior research, I acknowledge the differences across the wide range of software expertise. However, to capture the most prevalent differentiation, I chose to work with two significant levels – advanced and beginner.

2.2. Detection of Expertise

Prior research has investigated different ways to detect software expertise. Masarakal et al. introduced a seven-point self-assessment scale where users rated themselves through task questionnaires [51]. This technique is very common in testing software usability [13,14,70] and user experience [1] but lacks reliability [61]. Among other techniques, expert judgment has been leveraged in previous research to detect expertise levels. For example, Wang et al. assessed the task (i.e., produced by topic modeling) expertise using expert judgment and used this knowledge to recommend similar tutorials [77]. Another method of measuring software expertise involves controlled task assessment, based on the performance analysis of the users in a laboratory setting [28,36]. Unlike self-assessment, expert judgment and

laboratory task assessment are reliable but impractical outside the laboratory setting [21,30]. Therefore, I leverage the automatic detection of expertise.

Prior research has looked at the feasibility of automatically detecting software expertise, which is a key step for supporting users of differing skill levels. One area of focus has been on capturing and analyzing low-level interface operations. Examples of such expertise indicators include the time to perform commands [30], the rate of interface actions [35], pauses, or dwells [64], mouse motions [28], and menu access times [36]. My work aims to accommodate different skill levels by automatically assessing the difficulty of tutorials available online.

Other research has investigated how users of different skill levels utilize a feature-rich software application's command set. Lawson et al.'s study of spreadsheet use found expertise-related workflow differences [46]. Matejka et al.'s study of command usage behavior found that command usage frequency can be an indicator of software expertise [48]. I leverage these findings to investigate command-oriented tutorial features that serve to discriminate between beginner and advanced tutorials.

2.3. Improving the Usability of Software Tutorials

Many software users, especially newcomers, often struggle in locating a relevant tutorial for a given task [39]. Given the ubiquity and important role of tutorials in software learning, a wide body of work has looked at how to support tutorial use and retrieval.

In supporting tutorial use, prior work has explored integrating tutorials with the target applications, for example, through overlays that help users find tutorial commands [37,69], or techniques that use application context to control a video tutorial's progression [64].

Prior work has also focused on reducing workload by automating certain mechanical tutorial steps [11,42], motivating tutorial use by adding gamification elements [47], and augmenting tutorials with input from the user community [10,44,65].

Some prior approaches have explored annotating software tutorials to make it easier for users to select, appraise, and navigate them. Examples of previously explored tutorial annotations include commands covered [26,63], UI events [5,32], other users' viewing patterns [40], and the location of workflow steps within a video [41,79]. This prior work has leveraged a mix of automated (e.g., [26,63,65]) and crowdsourcing techniques (e.g., [15,41]) to create the annotations.

Despite all the research in improving user interaction with tutorials, there is very little prior work on providing users with information about the difficulty level of the application content covered in the tutorial. One exception is Social CheatSheet [75], a system for creating and sharing software instructions and tutorials, which proposed a social voting mechanism to classify an instruction set's difficulty level. Also highly relevant to my work is Wang et al.'s work on identifying tutorial tasks [77]. Their approach leveraged command usage logs and topic modeling to identify latent tutorial topics. They then had experts assign human-readable labels to the topics, consisting of the task covered and its difficulty. My work differs in that I use machine learning to classify a tutorial's difficulty level automatically. My approach also does not require access to usage logs. Also, my work adds insights into how tutorial difficulty information can affect novice and expert users' tutorial selection tasks.

2.4. Summary

Previous research has characterized different software expertise levels and detection techniques. Guided by earlier work, my thesis goal is to detect software tutorial's difficulty automatically. Prior work has leveraged menu access time, command invocations, mouse motions and rate of interface actions in expertise detection. I extend this body of work by focusing on different distinguishable aspects of online tutorials. Previous work has investigated the feasibility of different tutorial annotations to improve tutorial navigation and quality of the contents. My objective is to annotate tutorials with difficulty levels and assist users in tutorial selection.

Chapter 3

Investigating Differentiable Features

My thesis goal is to investigate the feasibility of automatically labeling an application tutorial's difficulty. In this chapter, I describe the data that I collected for classifier training, my data preprocessing strategies, my feature investigation and extraction process (i.e., feature engineering), and the analysis of the extracted features to see any statistically significant differences between advanced vs. beginner tutorials. Figure 1 shows the method overview for the feature investigation and extraction process.

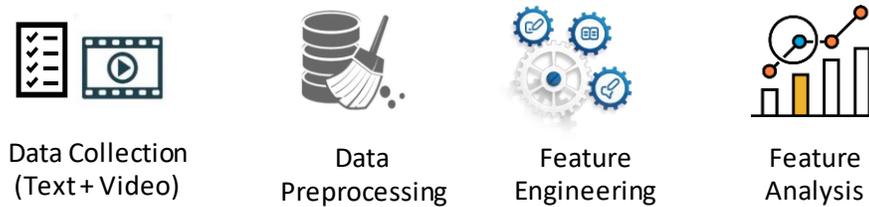


Figure 1: Different Stages of the feature investigation.

3.1. Data Collection

I began by collecting a corpus of already labeled tutorials for use as ground truth for classifier training and testing purposes. The initial investigation was confined to Photoshop tutorials as it is widely used and frequently studied in feature-rich software research [10,11,18,41,69].

To ensure high-quality difficulty labels, I consulted only tutorial sources that appeared to have a strict editorial process or accepted tutorials from only experienced authors. In my final sample, I included tutorials from 9 sources: Adobe, envatotuts+, tutvid, tutpad, Creative Bloq, PSD Vault, Pelfusion, 99 designs, and Photoshop Star. As a proof-of-concept, I focused on building a classifier to distinguish between two classes, a choice motivated by the fact that six of my sources used this level of labeling granularity (e.g., “Advanced/Beginner”). The remaining three sources used three difficulty levels (e.g., “Advanced/Intermediate/Beginner”). For these sources, I labeled both the “Intermediate” and “Advanced” tutorials as “Advanced” in my corpus. My final corpus had 750 tutorials (i.e., 375 advanced and 375 beginner), with equal distributions of video and text tutorials across each difficulty level (70% text and 30% video tutorials). Figure 2 shows the distribution of the collected data.

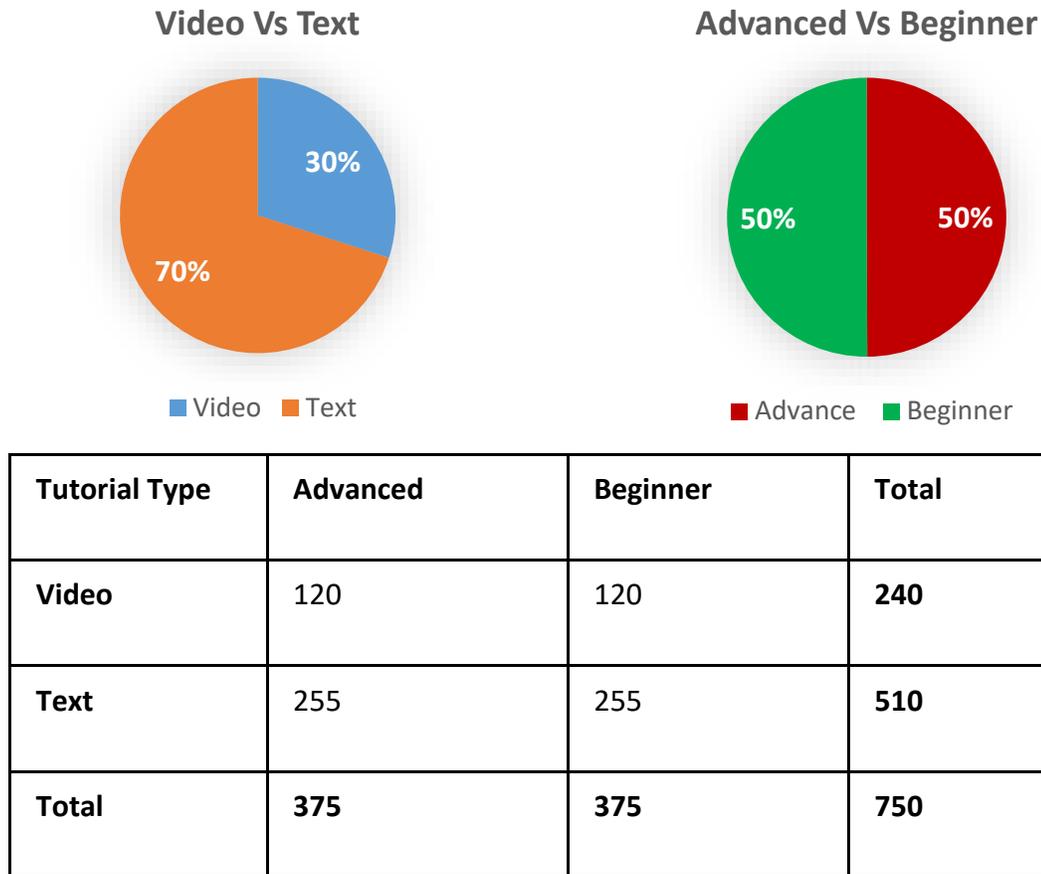


Figure 2: Distribution of the collected Photoshop Tutorials

3.2. Data Preprocessing

My next step was data preprocessing. In a classification task, data preprocessing leads to significant improvements by removing sources of noise [74]. Guided by informal experimentation, I performed four preprocessing steps on my data. Figure 3 summarizes my preprocessing steps. I briefly discuss each step in the following paragraph.

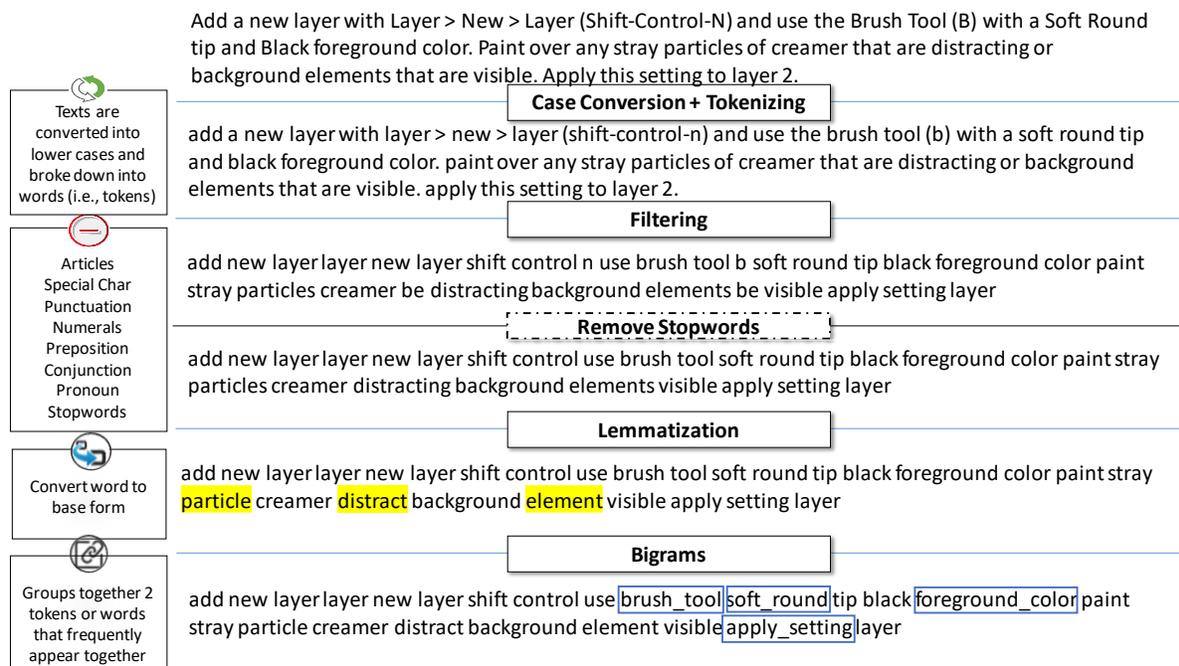


Figure 3: Different Preprocessing Steps

In the first step, I converted all text (including the transcript for the video tutorials) into lower case and divided the text into tokens (i.e., small pieces or words). In the given example in Figure 3, ‘add’, ‘new’, ‘layer’, ‘use’, ‘soft’, ‘tip’, ‘brush’ are the tokens. In the filtering step, similar to prior work [57,72], I removed special characters, articles, punctuation, numerals, prepositions, conjunctions, pronouns and stopwords. For example, ‘>’, ‘(,)’, ‘be’, ‘and’, ‘with’, ‘2’ etc. are removed from the text at the filtering step. In the third step, I converted words into their base forms (known as lemmatization [4]). From the given example, the highlighted words such as ‘particle’, ‘distract’ and ‘element’ are the base form of ‘particles’, ‘distracting’ and ‘elements’. In the fourth step, I created bigrams of words [9], by grouping together frequently co-occurring words. For example, ‘brush_tool’ groups together two different tokens such as ‘brush’ and ‘tool’ (i.e., in Figure 3, the bigrams are shown by the enclosed boxes).

3.3. Feature Engineering

After preprocessing, I created a set of potential features to train the classifier. By investigating prior work on software expertise and learnability (e.g., [30,31,46,54,75]) and conducting informal feature investigations, I settled on: topics, commands, word repetition, text difficulty, and length. I briefly discuss my motivation for each feature, and how I developed the feature from the tutorial text in the following subsections.

3.3.1. Tutorial Topics

Prior work has pointed to a potential relationship between a tutorial's higher-level topic and its difficulty level. For example, an analysis of comments that users post to online tutorials indicated that the user community views certain tutorials as covering expert techniques [43]. Wang et al.'s work on identifying tutorial tasks via command usage logs showed that when experts were asked to provide human-readable labels for the machine-generated topics, their labels included both task and difficulty information [77].

Inspired by this prior work, I used topic modeling to generate a set of topics that I leveraged in classifying tutorial's difficulty. Due to its ability to capture the hidden structure of the text [59,77], I used the topic modeling algorithm, LDA [6] (using Gensim [82]).

LDA assumes each document (i.e., tutorial) as a mixture of topics where these topics are present in different proportions. These proportions are called topic distribution probabilities. For example, if LDA represents any tutorial by topic 1: 0.7, topic 2: 0.2, and topic 3: 0.1 (where 0.7, 0.2 and 0.1 are the probability values) that means topic 1 contributes the most in the given tutorial. I generated two different models using this topic-modeling technique: 1) A Topics-All model which considered all of the preprocessed text and 2) A

Topic-Commands model, which considered only command references. Figure 5 shows the general concept of the LDA.

To extract command references for the Topic-Commands model, I applied techniques from prior work on automatically identifying direct and indirect references (i.e., the tutorial says “adjust the blending mode” instead of the actual command “set blending mode”) [26,63]. I created a Photoshop command dictionary consisting of both direct and indirect command references. My method of creating the command dictionary is shown in Figure 4.

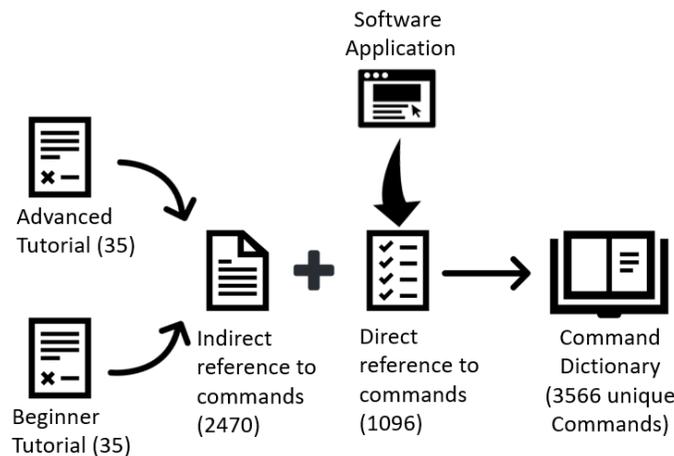


Figure 4: Process of developing the command dictionary

I collected the direct command references from the application interface (i.e., Photoshop). The list of commands in the Photoshop interface is divided into three sections, such as “Tools”, “Panels” and “Commands”. The list of commands can be accessed via Edit > keyboard Shortcut > Summarize. From that list, I collected all the commands enlisted to “Tools”. From the “Commands” and “Panels” sections, I only collected the last member of the menu hierarchy. For example, if the menu hierarchy of a command is “Layer>Smart

Objects>Convert to Smart Object” then I only enlisted “Convert to Smart Object” in the command dictionary. Thus, I collected 1096 unique direct commands from the Photoshop interface. To collect examples of indirect references, I manually annotated a subset of 70 Photoshop tutorials (35 Advanced and 35 Beginner). I added an additional 2470 indirect command references to the dictionary via this hand-annotation approach. Finally, all together, I collected 3566 unique direct and indirect commands, which I enlisted in the command dictionary.

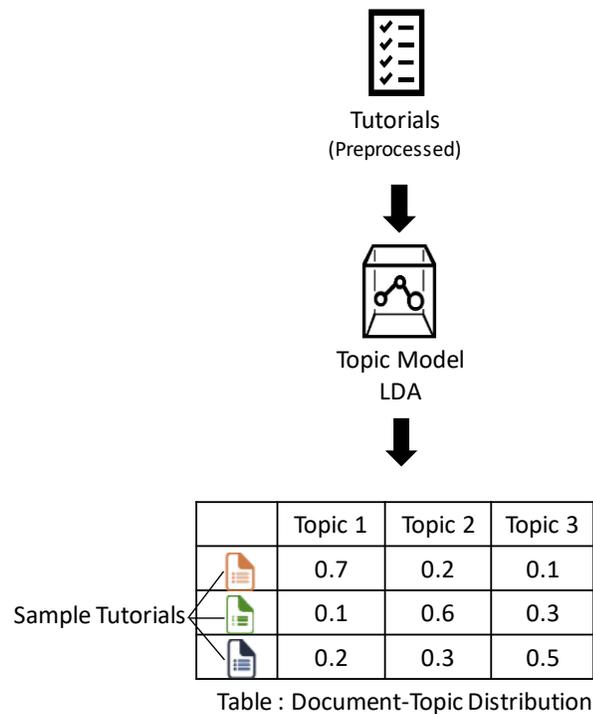


Figure 5: An example of LDA topic model output for 3 sample topics.

I used both sources of text (all preprocessed text and only command references) as input to LDA. To specify the number of topics for LDA to generate, I used an evaluation metric called topic coherence [56,58,76], which measures the human-interpretability of the topics.

Using this metric, I generated 30 LDA topics. As output, LDA generates a document-topic distribution matrix that I used for my classification. Figure 5 shows an example of this matrix.

3.3.2. Command Ratio (CR)

Matejka et al.’s study of command usage behavior found a connection between a user’s expertise level and the frequency in which they used different commands [48]. To investigate whether tutorials designed for experts might also make heavier usage of commands than those designed for novices, I chose to explore differences in how often tutorials refer to commands. To account for tutorial length, I used a tutorial’s command ratio (CR), which represents the percentage of words in the tutorial that refer to a Photoshop command. The calculation is made using the following:

$$\text{Command Ratio (CR)} = \frac{\text{Number of Total Words in Commands}}{\text{Number of Total Words}} \times 100$$

3.3.3. Word Repetition (WR)

I conducted an informal investigation and found that advanced tutorials tended to focus on specific effects or tasks (e.g., “Creating a Sketch Effect”) whereas the beginner tutorials were often broader (e.g., “Demonstrating the use of Different Retouching Tools in Photoshop”). To try to capture some of this difference, I created a feature based on word repetition:

$$\text{Word Repetition (WR)} = \frac{\text{Number of Repeated Words}}{\text{Number of Total Words}} \times 100$$

I speculated that there might be more repeated words in the advanced tutorials owing to their more focused nature. On the other hand, it is also possible that beginner tutorials might contain more repetition to reinforce key concepts.

3.3.4. Text Difficulty (TD)

Also, based on my informal investigation, I speculated that advanced tutorials might use more complex language. To capture this, I used a consensus score of 7 different formulas as advocated in prior work [23] (i.e., Flesch Reading Ease, Flesch-Kincaid Grade Level, Fog Scale, SMOG Index, Coleman-Liau Index, Automatic Readability Index, Linsear Write Formula). The score considers average sentence length, average number of syllables per word, percentage of words having 3+ syllables, etc. It penalizes text having polysyllabic words and long, complex sentences. This score has a scale from 1-12, with higher values representing more complex text.

3.3.5. Tutorial Length (Len)

Finally, my informal investigation suggested that advanced tutorials tended to be lengthier than beginner tutorials, prompting to include the tutorial length as one of my features. I represent tutorial length as the number of words present (i.e., word count). I used word count primarily because this feature could be easily calculated from the videos (i.e., video transcripts) or text tutorials.

3.4. Feature Analysis between Advanced vs Beginner Tutorials

For features that could be summarized using means (e.g., command ratio, length, word repetition, and text difficulty), I looked for statistically significant differences between the advanced and beginner tutorials in the dataset (using 2-tailed Independent T-Tests).

Table 1 shows that advanced tutorials are significantly longer and have more repeated words than beginner tutorials. Contrary to my speculation, beginner tutorials use more complex language (according to the readability measures); however, the size of the effect (as measured by Cohen’s *d*) is small. I did not find a significant difference in the density of command references (i.e., command ratio) between advanced and beginner tutorials.

	Adv mean (s.d.)	Beg mean (s.d.)	Sig	Cohen’s d
Command Ratio	33.3 (10)	34.3 (11.2)	$p = 0.10$	0.1
Length	2275.8 (1124.1)	1461 (841.8)	$p < 0.001$	0.8
Word Repetition	71.7 (8.5)	68 (7.9)	$p < 0.001$	0.5
Text Difficulty	7.5 (1.6)	8.1 (1.7)	$p < 0.001$	0.4

Table 1: Mean differences between Advanced vs. Beginner tutorials.

3.5. Summary

I investigated and automatically extracted features from the collected Photoshop tutorial’s text (i.e., 750 tutorials) after preprocessing. My final set of features includes – topics, command ratio, word repetition, text difficulty, and length. I analyzed the differences of command ratio, length, word repetition, and text difficulty between advanced and beginner tutorials using 2-tailed independent T-tests. My findings suggest that advanced tutorials are significantly lengthier and contain more repeated words than beginner tutorials. On the other side, I found beginner tutorials are significantly more difficult to read (i.e.,

according to text difficulty) than the advanced tutorials. However, the effect size of this difference is not very substantial according to Cohen's *d*. I did not find any significant difference in the command references (i.e., command ratio) between these two groups.

Chapter 4

Model Generation and Evaluation

This chapter describes the performance of different models that I generated to classify the difficulty of a Photoshop tutorial. I investigate i) the feasibility of automatically classifying a tutorial as either advanced or beginner; and ii) the discriminatory power of the different features, both in isolation and in combination.

Due to its robustness and that it tends to be less prone to overfitting than some other approaches (e.g., Decision Tree, Naïve Bayes), I used Random Forest for the classification [8]. I optimized classifier parameters using Grid Search [67]. To evaluate the model's performance, I used a standard cross-validation approach, with 10 folds (using StratifiedKfold [83]). In other words, each model was trained and validated through 10 trials, where each trial used a different 90% of the data as training samples and the

remaining 10% of the data as testing samples. Because of my balanced dataset, I report accuracy as my performance metric.

4.1. Impact of Individual Feature on Classifier Accuracy

I initially investigated the impact of the individual feature sets (topics, length, word repetition, text difficulty, and command ratio) on classifier performance. As a reminder, for the topics, I have two models: Topics-All and Topics-Commands.

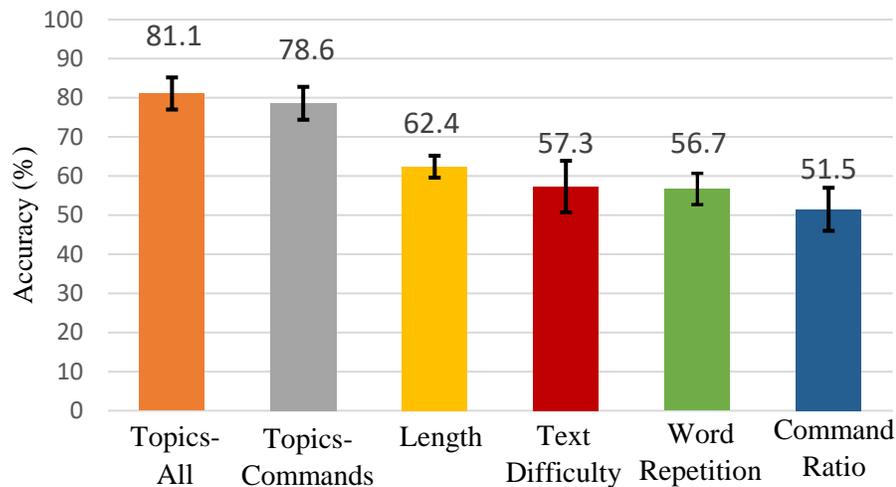
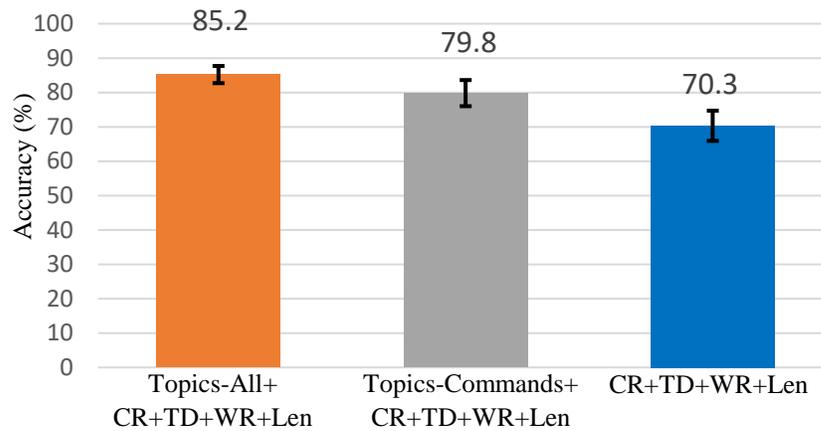


Figure 6: Model Performance using individual features.

From Figure 6, we can see that my classifier achieved the best performance (i.e., accuracy = 81.1%, s.d. = 3.8) when it was trained using the topics derived from all of the text. The accuracy dropped slightly (to 78.6%, s.d. = 4.2) when considering only the command references. In other words, topics are the most informative feature, and the difficulty information is not only confined to the Photoshop command references. Conversely, the command ratio was the least informative feature, resulting in baseline accuracy (i.e., 50%

in this 2-class classification problem). The models trained with the other feature sets (text difficulty, word repetition, and length) also did not perform well. Thus, while there were significant differences in mean values for these tutorial features, these differences were not strong enough to distinguish between advanced and beginner tutorials.

4.2. Impact of Combining Feature Sets on Classifier Accuracy



CR: Command Ratio, TD: Text Difficulty, WR: Word Repetition, Len: Length

Figure 7: Model performance using combined features.

I also investigated the impact of combining different features on classifier accuracy. Figure 7 shows that the classifier performed best (achieving 85.2% accuracy, s.d.=2.5) when I included all of my features. In this highest-performing model, the topics were derived from all of the text. Accuracy dropped slightly (to 79.8 %, s.d.= 3.8) when using the command-only topic distributions. These results indicate that while some of my features lack discriminatory power when used in isolation (see Figure 6), they performed better when used in combination.

4.3. Impact of Number of Training Samples on Classifier Accuracy

My next investigation is focussed on the number of training samples required to generate a good fit model. For this, I used the concept of learning curves [2,78], which shows how the model's performance changes as the training dataset size increases. In learning curves, a model is evaluated on a training dataset and a validation dataset. Here, I chose 10-fold cross-validation to split the data into training and validation sets. Figure 8 shows the learning curves for our best model (CR, TD, WR, Len, and Topics-All). In this figure, the X-axis represents the number of training samples, and the Y-axis represents the accuracy score. The top line indicates the performance on the training data and the bottom line indicates the performance on the validation data. Here, the training curve indicates how well the model is learning, and the validation curve indicates how well the model is generalizing to the unseen data.

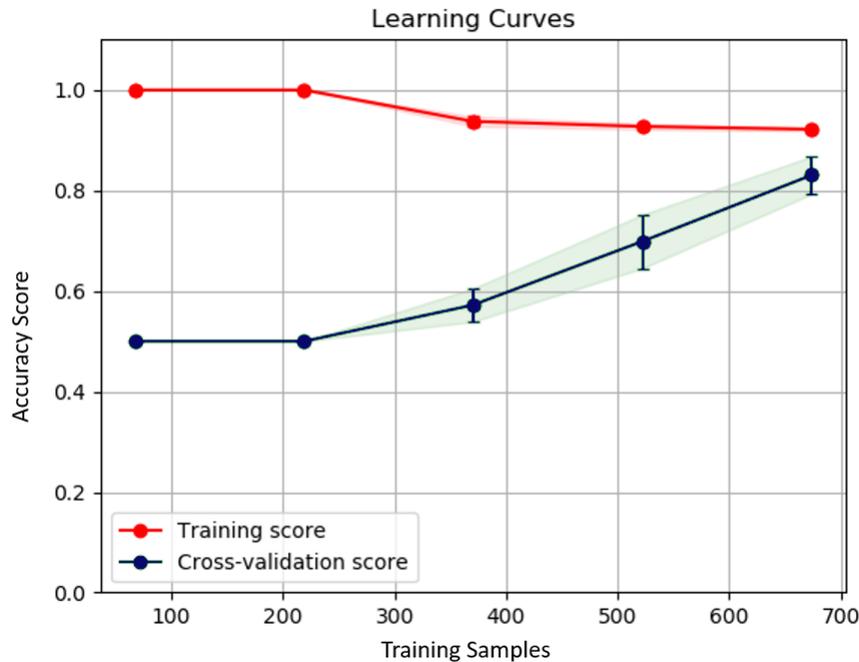


Figure 8: Learning Curves. *error bar represents s. d.

By analyzing Figure 8, we can see that when the number of training samples is 60, the model has the validation score of 0.5 (50%) and the training score as 1.0 (100%). At this point, the model perfectly fits the training data but has not learned enough to be able to classify unseen data. We can see the performance improvement of the model for the unseen data with the increment of training samples. For example, at 520 training samples, the validation score reaches to 0.715 (71.5%). However, the training score has encountered a sudden drop at this point. A certain drop is acceptable because a model that learns the training data too closely often suffers from overfitting.

We see that to achieve an accuracy of 80%, a minimum of 630 training samples are needed. While using 675 training samples, the validation score jumps to 0.846 (~85%), which is still on the rise. Analyzing both training and validation scores at this point, we see that there is a gap between the training score and the validation scores, which is known as the variance [84]. For a good fit model, the variance should be as low as possible. In my case, we see the validation curve has not yet faced the plateau effect, which is an indicator that if I provide more training data, the model is likely to achieve better performance.

4.4. Generalizing to 3D Modeling Tutorials

To investigate the generalizability of my features, I evaluated my best model's performance (CR, TD, WR, Len, and Topics-All) using tutorials for a different feature-rich application: 3D modeling software. For this purpose, I collected 210 labeled tutorials for the application Fusion 360 (Advanced 105, Beginner 105, 90% video tutorials) and constructed a Fusion 360 command dictionary. The data preprocessing and feature engineering procedures were identical to those described in Section 3.2 and Section 3.3, with the exception that LDA

produced 20 topics (guided again by the topic coherence score). With this dataset, my classifier achieved an average of 81.4% accuracy (s.d.= 9.2) when trained/tested using 10-fold cross-validation. This accuracy provides encouraging initial evidence that my feature sets and classification techniques generalize beyond Photoshop to other kinds of feature-rich software.

4.5. Summary

I generated different machine learning models using my engineered feature sets of Photoshop tutorials. I used the random forest classifier to build and 10-fold cross-validation to evaluate the models. I investigated the contribution of the different feature sets (individual vs. combined) in the model's performance (i.e., accuracy). After the performance analysis, I found my best model has an accuracy of 85% and uses all of the engineered features (e.g., topics, command ratio, word repetition, text difficulty, and length) to classify advanced vs. beginner Photoshop tutorials. My investigation on the amount of training data indicated the possibility of performance improvement with more training samples. To find out the generalizability of my feature sets, I also trained another model using 210 Fusion 360 tutorials (i.e., 3D modeling software), which was able to classify advanced vs. beginner tutorials at 81% accuracy.

Chapter 5

Development of the Prototype: *TutVis*

This chapter discusses the development of my tutorial browsing interface prototype, *TutVis*. *TutVis* summarizes the model's generated decisions (i.e., tutorials difficulty) along with the model's features through visual interface components. The model's features are numerical values that need further transformation to present them in *TutVis*. I investigated different approaches for the transformation. After a series of testing, I found out that the subsets of my model's features, i.e., length, text difficulty, command ratio, and word repetition can be presented through a three-level scale. However, I needed to interpret my model's feature - topics further to transform them into a meaningful visual interface component of *TutVis*. In the following subsections, I present my approach of the transformations.

5.1. Transforming Text Difficulty, Length, Word Repetition, Command Ratio into Interface Components

I investigated different approaches to visually represent the subset of my model's features, i.e., text difficulty, length, word repetition and command ratio. Some of the approaches include presenting values as integers, representing values through percentages, relative comparison from the average distributions and converting the numerical value into a three-level scale. Table 2 presents the approaches that I tried to represent one of the model's features - Text Difficulty.

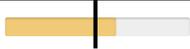
Type	Indicator	Illustration
Integer		The text difficult score of this tutorial is 8 out of 12.
Integer		The tutorial has a text difficulty score of 8 and should be appropriate for the 13-15 age group.
Percentage		Considering the text difficulty, the tutorial is just above 50%.
Comparison from the average		The text difficulty of this tutorial is slightly above average.
Three-level scale		The tutorial is fairly easy to go through.

Table 2: Different representation of Text Difficulty

After trying different scales, I decided to transform the features, i.e., text difficulty, command ratio, length, and word repetition into a three-level scale, i.e., “low”, “medium” and “high”. I chose this technique because I found that this way of presentation was simpler, meaningful, and easy to interpret. For example, a tutorial presenting “high” value for length can be interpreted as a lengthy tutorial. Similarly, a low value for text difficulty can be interpreted as a tutorial having more simpler text structure, which is expected to be easier to go through.

5.2. Transforming Topics into Interface Components

My model’s performance analysis revealed that topic distribution (generated via LDA) was my most informative tutorial feature. As a reminder, LDA generates latent words for each topic and applies a generic label (e.g., “Topic 1”, “Topic 2” in Figure 9). However, the generic labels for the topics and their distribution lack interpretation. For example, in Figure 9, the first sample tutorial has a value of 0.7 for “Topic 1”. Here, the label - “Topic 1” does not have any meaning, and therefore, the value is hard to connect with a meaningful semantic. So, I needed further interpretation to be able to present them in *TutVis*. In the following, I describe how I went from this LDA output to the human-readable labels that I used in my *TutVis* system.

There are different methods for topic labeling; for example, labels can be generated by humans manually [68,77] or through automated techniques [45,55]. I use a manual approach for labels because they often give users more insights into the nature of the topics than ones that are automatically generated [34].

My first approach was to focus on the top latent words from a topic-word distribution table that LDA generates automatically (e.g., “scene”, “resize”, “composite”, and “matte in Figure 9). Prior work reports success in using software experts to assign topic labels to sets of latent words that consist only of precise software command names of 3d design application [77]. I tried this approach with my *Topics-All* model (as this model performed best in the classification task) but found it difficult to connect the latent words produced with a meaningful semantic label, in part because the latent words included a number of generic Photoshop terms (such as scene, matte, animation, timeline). Instead, I devised my approach to topic labeling that involved: 1) creating clusters of tutorials based on LDA output, and then 2) qualitatively analyzing the tutorials in each of the clusters.

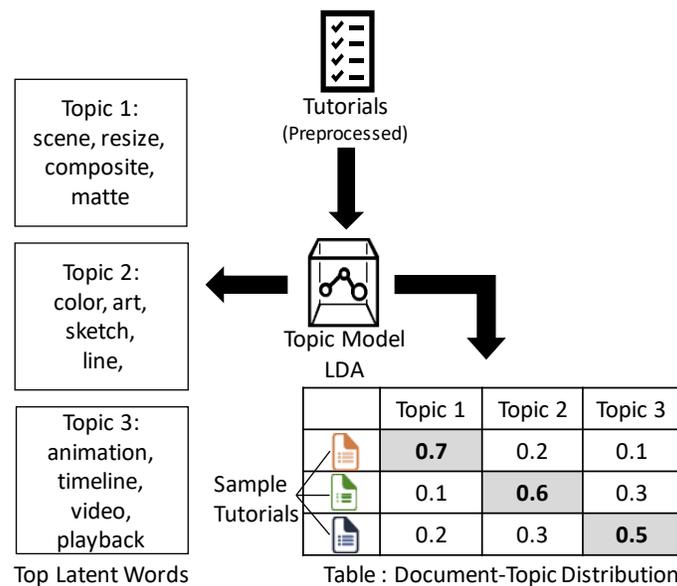


Figure 9: An example of LDA topic model output for 3 sample topics. The shaded fields represent dominant topic.

5.2.1. Generating Tutorial Clusters

LDA classifies each document (i.e., tutorial) as a mixture of topics, where each topic is contributing a different amount. This mixture is represented as a probability distribution. For example, in the sample document-topic distribution table in Figure 9, the first tutorial is represented by the topic distribution: Topic 1: 0.7, Topic 2: 0.2, and Topic 3: 0.1. From this distribution, I define the dominant topic as the topic having the highest probability value within this distribution. For this sample tutorial, Topic 1 is the dominant topic (see the shaded values in Figure 9’s Document-Topic distribution table). Following this technique, I defined the dominant topics for all the tutorials in my corpus. To look for semantic relationships, I created tutorial clusters, based on tutorials with the same dominant topic. Figure 10 shows sample tutorial clusters, where each cluster has tutorials with the same dominant topic.

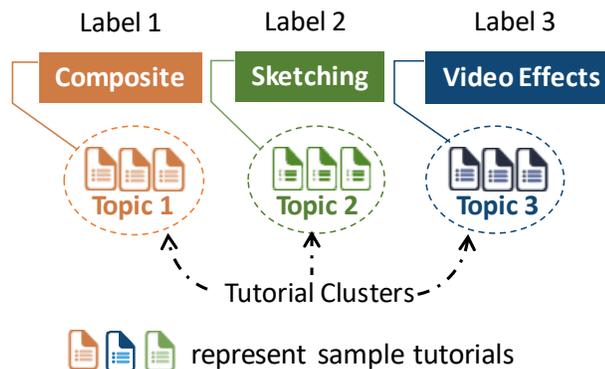


Figure 10: Topic Labeling using Tutorial Clusters

5.2.2. Analyzing Tutorial Clusters to Generate Labels

After the generation of the tutorial clusters, I analyzed them qualitatively for commonalities in the Photoshop tasks that they covered. I used these commonalities to label the LDA topics.

For my analysis, I focused on the top tutorials (i.e., ordered by the probability values) in each cluster, as they were the most representative of that cluster’s topic. I used qualitative analysis involving open coding [73]. During my coding, I consulted the tutorials’ titles, commands used, high-level tasks performed, image cues, and any end goal specified by the tutorial author. I coded at least three top tutorials under each cluster, examining more tutorials are necessarily to find clear patterns. After open coding, I identified common themes of each cluster, which I used for labeling. In the following paragraphs, I show my coding strategy that I followed to label topic 5 of my LDA model output.

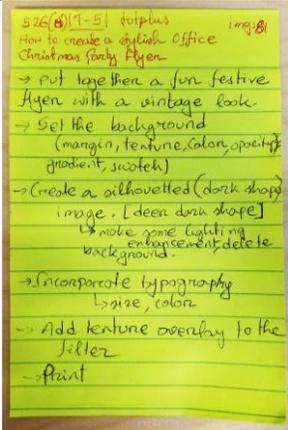
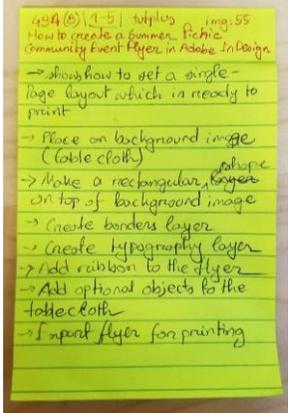
Notes	Image Cue	
Tutorial 1		
Tutorial 2		

Figure 11: Sample code of the tutorials of topic 5.

Figure 11 shows my annotations for two top tutorials under topic 5. While coding a tutorial, I collected title, high-level tasks performed, and end goal in handwritten notes. I also collected the image cues of these tutorials separately. I used the notes for thematic analysis and the image cues as an illustration of the result.

For collecting the title, I looked into the tutorial's heading. While analyzing the tutorials, I noticed that tutorial author often defines the objective at the starting or ending note of any tutorial, which I referred to as the author's end goal. For example, in one tutorial, the author describes:

“In this tutorial, we'll use Adobe InDesign, Photoshop and Illustrator to put together a fun festive flyer with a vintage look ... for promoting office parties or other holidays. This flyer ...”

From the statement, I summarized the end goal of this tutorial was to design a festive-looking flyer with a vintage vibe. To deduce the higher-level tasks performed, I looked at the workflow information, which is often provided as sub-steps in any tutorial. For example, in the first sample tutorial, the provided sub-steps are as follows: “how to set up a flyer in InDesign, how to create a silhouetted image, how to incorporate typography into a flyer design, how to add a texture overlay to your flyer, conclusion”. From the given information, I deduced the sub-steps to be: set the background, create a silhouetted image, incorporate typography, add texture and export (see Figure 11). However, I also found some tutorials which did not have any explicit sub-steps. For those, I came up with some sub-steps by analyzing the types of tasks performed and the commands used. For example, in one tutorial, the author provides the following instructions:

“I’m going to use this image with the Mountaineer so I would like to change the background ... I’m going to create a quick selection and use that selection for masking out the background of this image... I prefer to use the quick selection tool... creating the selection by dragging the parts that I would like to select... the magic wand and quick selection tool works the same [for the selection task]”

From this, I decided the sub-step to be: Selecting a part of an image.

After my tutorial coding process, I used thematic analysis to deduce the common themes of the topic. For example, the tutorials in Figure 11 follow a common workflow, such as: creating the background, decorating the background by adding objects and effects, creating text layer and export. The end goals for these tutorials are to design posters or flyers (see the image cues of the sample tutorials in Figure 11). Therefore, considering the themes, I chose to label topic 5 as – *Flyer & Poster Design*. However, I also encountered some tutorials where the end goal was not explicitly related to the topic label. For example, the topic *Animation & Video Effects* grouped tutorials where the authors provided the following end goals:

“In this tutorial, we will design a simple news iPhone app, and then animate it for client presentation and export it as a GIF file.” [tutorial 1]

“In this tutorial, I will teach you how to lift an object from its background using the automated Content Aware Fill and the good old copy and paste technique [creating glitch effect]” [tutorial 2]

“In this tutorial we'll take a video clip and transform it into a doodle-filled video inspired by Skrillex and Diplo's Where Are Ü Now with Justin Bieber”

[tutorial 3]

For these tutorials, the end goals seemed different from each other. So to label this cluster, I leveraged the latent words given by the LDA model in addition to the tutorial's end goal and workflow. For this cluster, the LDA model enlists words such as “animation”, “timeline”, “frame”, “video”, “gif” (i.e., top five words). By relating to the latent words, I noticed that the tutorials followed a common workflow to create an animation or video effects. After analyzing, I ended up with a common theme of this cluster and accordingly, I named this cluster as *Animation & Video Effects*.

During the topic labeling process, I found clusters that represented high-level Photoshop tasks that were clearly distinguishable from other clusters (about 30% of the topics, covering 30% of the tutorials in my dataset). Labeling these clusters was relatively straightforward. Two examples are *Flyer and Poster Design* and *Drawing Pixel Art*. The tutorials within each cluster had common sub-steps and end goals, but there were large differences across the two clusters. Figure 12 (see Type 1: Clear Topics) shows the general concept of labeling these topics. I present these topics with my generated labels and the top ten latent words (i.e., given by LDA) in Appendix A.

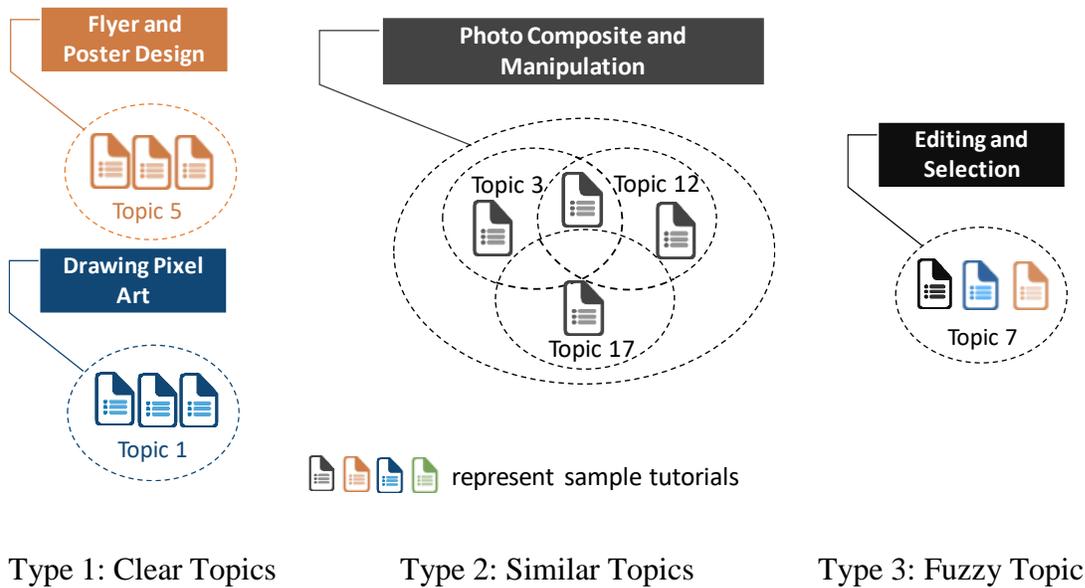


Figure 12: General concept of labeling different topics (i.e., tutorial clusters)

Among other tutorial clusters, I saw clear tasks within the topic but did not see enough semantic differences relative to some other clusters to warrant unique labels. While there were likely subtle differences in these tutorial clusters, the tutorials generally seemed to follow the same sub-tasks to achieve similar end results. For these clusters (about 60% of the topics, covering 66% of the tutorials), I grouped subsets of the clusters together and assigned a common label. For example, I assigned the label *Photo Composite and Manipulation* to 5 different clusters. Figure 12 (see Type 2: Similar Topics) presents the general concept of labeling these topics. In the following paragraphs, I present the rationale behind labeling these tutorial clusters as *Photo Composite and Manipulation*.

Below I present a list of the end goals of tutorials from five different clusters:

*“The image we’re going to create is inspired by a scene from the movie *Lovely Bones*, by Peter Jackson... Since we’re going for a fantasy world, I planned to*

use images of desert dunes to create the snow-scape. This will allow us to have the kind of surreal wavy lines in our landscape that would be impossible to obtain from real pictures of snow” [tutorial from topic 3]

“In this tutorial, you will discover how to combine advanced masking techniques, blending modes, adjustment layers and clever use of filters to part the sea and create a surreal photo manipulation” [tutorial from topic 12]

“In this tutorial, we will show you how to re-create that scene (movie scene) where a coast break apart and fall into the sea) using selection of stock photos” [tutorial from topic 17]

“In this tutorial I’ll show you how to use photo manipulation techniques in Adobe Photoshop to create a dreamy scene featuring a medieval woman with a dove carrying a letter... we’ll add the sky and landscape, import the bridge, model, castle and blend all of these elements together...” [tutorial from topic 24]

“In this tutorial we will be teaching how to integrate elements from different sources to create a realistic photo manipulation with dark conceptual elements. You will learn some lighting and blending techniques ...” [tutorial from topic 30]

As we can see, all the tutorials have similar end goals that involve manipulating photos and creating hypothetical or surreal scenery by combining those. I found similar high-level tasks to accomplish the end result within this group. The high-level tasks were: select the background scene, extract items from different images, blend items together, adjust

lighting, and add effects. Therefore, after analyzing all the collected information, I concluded the label for this group to be *Photo Composite and Manipulation*. I report all of these topics with my generated labels, and top ten latent words (i.e., given by LDA) in Appendix B

I also came across clusters where the top tutorials in the cluster were quite different from one another (about 10% of the topics, covering 3% tutorials of the corpus). I handled these cases by labeling them generically according to their commonalities (e.g., *Editing and Selection*). For example, in one topic cluster, I found one tutorial entitled “Photoshop CC Tutorial – Advanced How to Select Hair” – where the author shows the use of the “quick selection tool” to select delicate details. In another tutorial entitled “Glowing PS4 Controller” from the same cluster, the author shows different editing steps (e.g., color-adjustment, filter) to create a glowing effect. Here, the two tutorials from the same cluster seemed to provide different themes. So, I investigated a few more tutorials under this cluster entitled “How to Create Amazing Text with Mixer brush”, “Advanced Lighting Techniques in Photo Editing”, and “Advanced tutorial: How to select Difficult Hair in Photoshop CC”. After the analysis, I was unable to find any common pattern by analyzing their end goals, workflows, and the latent words provided by LDA. Therefore, I looked into a generic name that could best suit this cluster. I ended up naming this cluster as *Editing and Selection*. Figure 12 (see Type 3: Fuzzy Topics) shows the general concept of labeling these topics. I report all of these topics with my generated labels, and the top ten latent words in Appendix C.

The LDA topic modeling produced 30 topics; however, after the manual labeling process (i.e., merging and naming), I ended up with 18 topics. To verify the semantics of my labels,

I solicited feedback from a Photoshop expert who was not involved in the labeling process. I provided the expert with four randomly selected tutorials per topic (of the top ten for that topic) and asked him to verify the relevance of my labels to tasks being demonstrated. I used the expert's feedback to make some minor wording adjustments to my labels.

5.2.3. Advanced Vs. Beginner Topics: Some High-Level Differences

I examined the tutorial clusters to get a sense of any key differences between the topics covered by advanced and beginner tutorials. In my analysis, I considered a topic consisting of primarily advanced tutorials if at least 70% of its tutorial cluster was comprised of advanced tutorials (and vice-versa for beginner topics/clusters). I selected the 70% threshold heuristically as it seemed to provide a reasonable starting point. Figure 13 shows the general concept of classifying predominant advanced vs. beginner tutorial clusters (i.e., Photoshop).

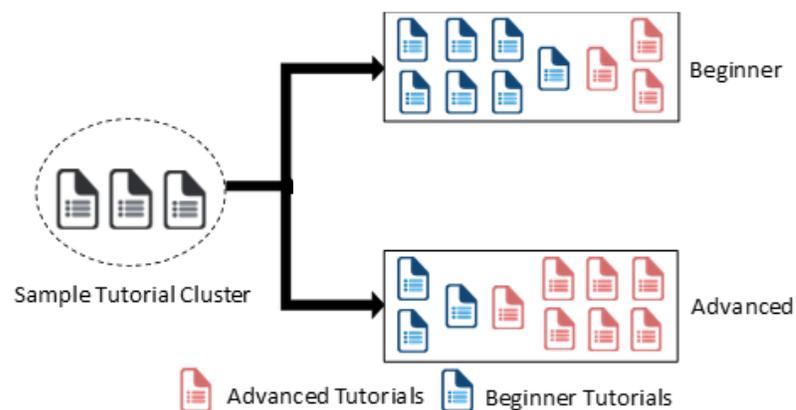


Figure 13: A general concept of classifying advanced vs beginner tutorial clusters (i.e., Photoshop).

I found that some of the advanced topics of Photoshop used special techniques to preserve an image's source content so that the same image could be reused even after modification.

For example, I noticed the use of “smart object”, which enables users to perform non-destructive editing in creating 3d objects (e.g., of a wine bottle, glass, and loaf). Some advanced topics assumed existing “how-to” knowledge, such as knowing about different photo manipulation techniques and how to use basic tools (e.g., pen tool, brush tool). Others involved using additional complex software (e.g., Cinema 4D, 3Ds Max, Modo).

In beginner topics, I found most of them provided comprehensive descriptions, without any assumption of existing knowledge. For example, tutorials in the *flyer and poster design* topic conveyed complete workflow guidance to the users, providing detailed instructions, and demonstrating the use of basic tools. I also noticed in beginner topics, the images tended to undergo fewer changes. For example, the beginner topic *photo editing and retouching* deals with fine-tuning different parameters, such as brightness, contrast or removing unwanted items from an image. In contrast, in the advanced topic *photo manipulation*, images underwent significant changes, particularly in terms of the image’s overall content. One example included changing the features of a person’s body (i.e., adding neon horns, creating surreal stitched eye effect). Some advanced topics combined multiple techniques that were covered in isolation in beginner tutorials. For example- *photo manipulation* often combines different photo masking, editing and retouching techniques to match the creator’s imagination. The future investigation could leverage this technique to uncover potential insight on advanced vs beginner topics in other domains (e.g., Fusion 360, AutoCAD, MAYA). Besides, this technique could also be used as feature extraction for classifying advanced vs. beginner topics.

5.3. TutVis: Tutorial Selection Interface

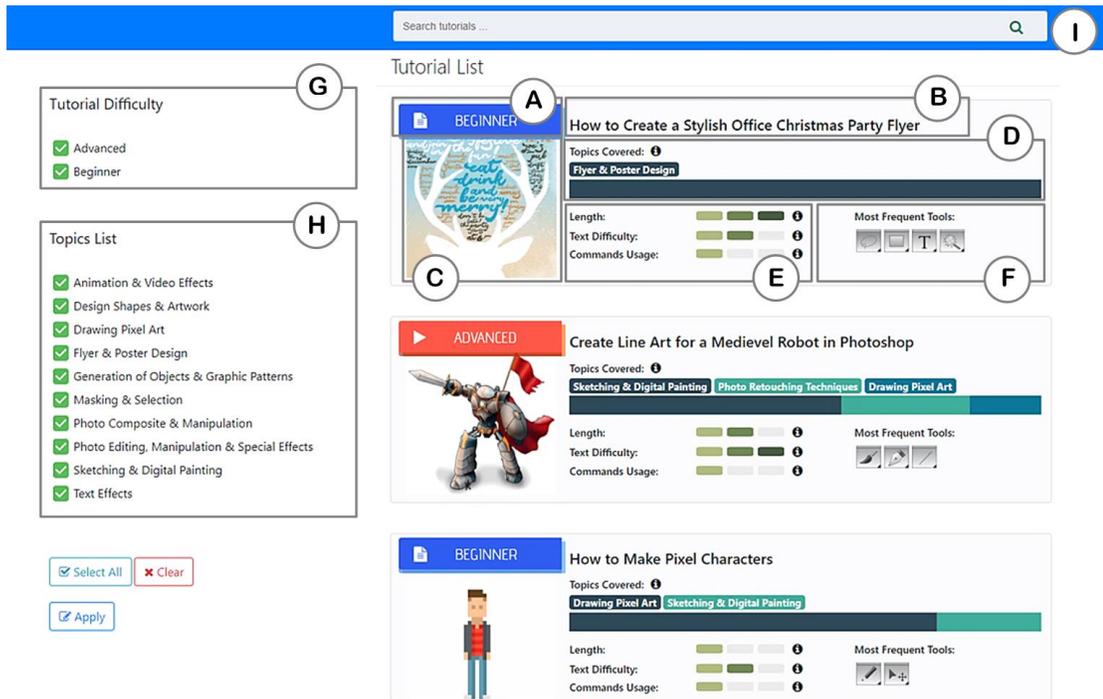


Figure 14: The *TutVis* interface, which presents a list of tutorials with difficulty (A), title (B), thumbnail image (C), topics covered (D), length, text difficulty, commands usage (E) and most frequently used tools (F). *TutVis* also provides filtering options (G,H) and a search bar (I)

To illustrate how my classifier and its features could be used to help users select tutorials, I developed the *TutVis* prototype. As shown in Figure 14, *TutVis* uses the classifier to annotate each tutorial with an automatically generated difficulty assessment. *TutVis* also summarizes other features that contributed to this difficulty assessment through interface components representing: the topics covered, the text difficulty, the length, and commands usage (renamed from command ratio in section 3.3.2 based on pilot testing). The visual representations of these features were refined iteratively based on pilot testing (examples of alternatives explored are provided in Appendix D). For topics, I chose to include only those which contributed at least 10% to the tutorial's overall topic distribution, resulting in

tutorials having at most three topics listed (Figure 14, D the stack bar shows the distribution of the topics). To present the length, text difficulty and commands usage, I converted their numerical values into low, medium and high scale (as shown in section 5.1). I did not include my model's word repetition feature after pilot testing with different visual representations revealed that users found this feature difficult to understand.

Building on prior work on command-oriented tutorial selection interfaces [42,63], in addition to the model's features, *TutVis* also lists the frequently used tools, as well as the title and the tutorial's output image (i.e., thumbnail). Users can click on a tutorial for a more detailed view and can hover to obtain more information on the different interface components. *TutVis* allows users to filter tutorials according to topic and difficulty. It also has a search bar where users can search different tutorials by the general topic or title. The searching supports approximate substring matching (i.e., fuzzy string searching) and presents result with the closest match. In case of presenting the results, it prioritizes tutorials having the exact topic name or title and sorts the tutorial list accordingly.

5.4. Summary

I transformed my model's features into interface components of a tutorial browsing prototype, *TutVis*. This prototype annotates tutorials with length, text difficulty, command usage, topics, and frequently used tools. To present three of my model's features - length, text difficulty, and commands usage, I used a three-level scale (i.e., high, medium, low). I devised an approach of interpreting topics as high-level Photoshop tasks by qualitatively investigating tutorial clusters. After interpretation, I presented the top three topics with my generated labels and their distribution in *TutVis*.

Chapter 6

Tutorial Selection Study

This chapter discusses the user study that I conducted to evaluate the utility of my prototype, *TutVis*. My goal was to gain insight into the value of the difficulty labels in helping users select a tutorial from a tutorial repository, as well as the representations of the different tutorial features (i.e., topics, length, text difficulty, commands usage). This study was approved by the university's research ethics board (see Appendix E for the approval and Appendix F for the certificate).

6.1. Participants

I recruited 12 participants (8 male, 4 female) through advertisements posted on a local university campus (see Appendix G for the poster advertising the study), via social media and through word of mouth. All participants were required to have some familiarity with

Photoshop. Among our participant pool, 5 self-reported as beginners (i.e., use Photoshop once a month or less), 5 as intermediates (i.e., use Photoshop at least once a week), 2 as experts (i.e., use Photoshop daily). Participants received \$20 (cash or gift card) for their participation.

6.2. Study Conditions and Tutorials

My study had a within-subjects design with three conditions (*Baseline*, *TutDiff*, and *TutVis*). In each condition, participants were provided with a different interface for browsing a set of Photoshop tutorials. The three conditions differed in the number of tutorial features that were displayed:

1. *Baseline*: each tutorial was annotated with only the title, thumbnail image, and most frequently used tools (see Figure 15).

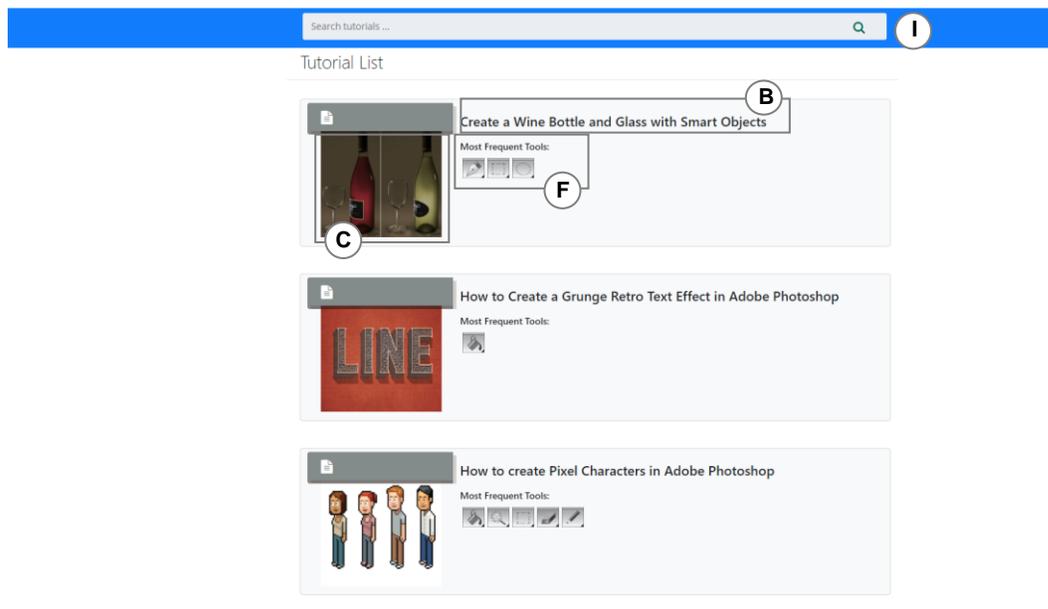


Figure 15: The *baseline* interface, which presents a list of tutorials with title (B), thumbnail image (C), and most frequently used tools (F). This interface also provides a search bar (I)

2. *TutDiff*: all information in the *Baseline* interface plus the auto-generated difficulty labels (advanced/beginner) (see Figure 16).

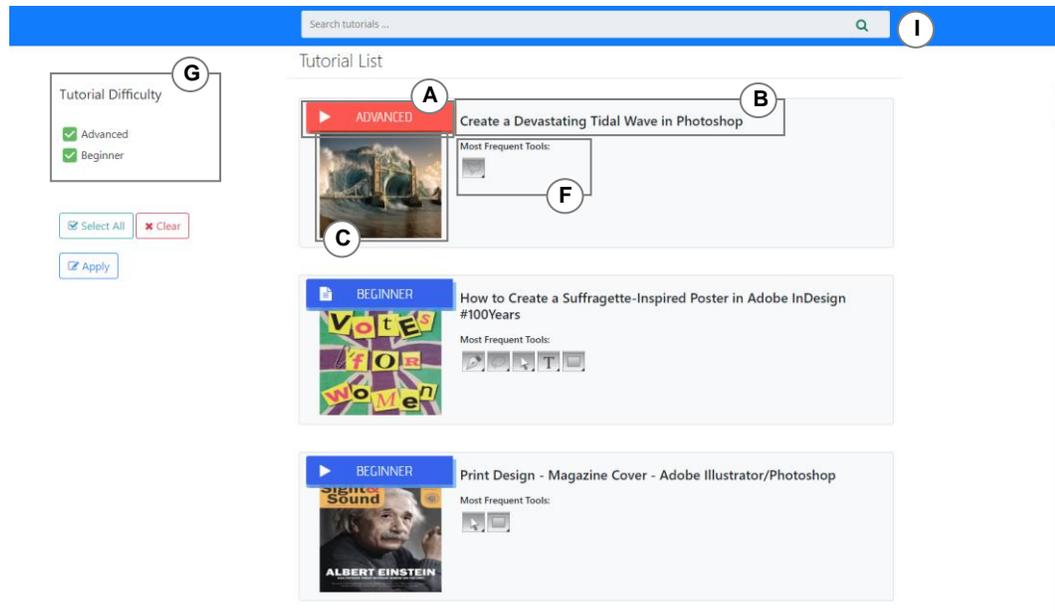


Figure 16: The *TutDiff* interface, which presents a list of tutorials with difficulty (A), title (B), thumbnail image (C), and most frequently used tools (F). *TutDiff* also provides filtering options (G) and a search bar (I)

3. *TutVis*: the complete *TutVis* system as described in Section 5.3. The additional annotations available in this condition can be found in Figure 14; D, E (i.e., topics, length, text difficult, and commands usage).

Each tutorial selection interface contained a list of 50 tutorials. I had three mutually exclusive sets of varied tutorials (in terms of topics, difficulty, length, etc.), which I randomly assigned to each condition. To replicate my model's overall performance (85% accuracy), each set had 7 tutorials with incorrect difficulty labels (i.e., misclassified as advanced or beginner). The order of interface condition was fully counterbalanced across participants.

6.3. Procedure

I began by asking participants to sign in a consent form (see Appendix H) and giving them a demographic questionnaire (see Appendix K) to complete. After completing the demographic questionnaire, participants were asked to complete three tutorial selection tasks per-interface condition (i.e., nine in total). Before getting to work with each interface condition, participants were given a brief instruction (see Appendix I) on the available features. Each tutorial selection task presented a different scenario and asked the participants to find a tutorial accordingly. My scenarios were motivated by findings from the previous research on the different reasons that users search for tutorials online (e.g., [18,43]). The first focused on a scenario with a sense of urgency, the second involved an exploratory search and the third focused on wanting a tutorial of particular difficulty. I created three isomorphic scenario sets, which I iteratively refined and pilot tested. Table 3 shows one of the scenario sets (see Appendix J for all the sets).

To focus the study time on tutorial selection data, I asked participants to spend around 7-10 minutes per selection task but did not require them to complete their selected tutorial. This technique follows previously established methodology for evaluating tutorial selection interfaces [42].

Task	Task Description
Sense of urgency (1 st Task)	Suppose you are assigned the task of creating an advertisement for a fundraising occasion. You want to complete this task quickly. Select a tutorial that you think would serve as the best starting point for you.
Sense of exploratory search (2 nd Task)	Suppose you are free for the whole afternoon, and you are interested in learning about digital drawing. Find a tutorial, which would give you some insight into digital drawing.
Sense of difficulty (3 rd Task)	Suppose you have a friend who has never used Photoshop before. Recently, he asked for your help in finding tutorials on how to change an image background. Find a suitable tutorial for your friend.

Table 3: One set of tutorial selection scenarios

Participants were asked to think-aloud while searching for tutorials. I also recorded participants' eye gaze information using a Tobii Eye Tracker 4C. After each condition, participants completed a short questionnaire where they reported i) on which interface components they used, and ii) their confidence level in their tutorial selections using a 5-pt Likert scale (see Appendix L). After completing all three conditions, participants took part in a semi-structured interview, where I asked about their experiences with the three interfaces (see Appendix M for the sample questions). Each study session lasted approximately 1.5 hours.

6.4. Results

6.4.1. Subjective Response: Preferences and Confidence Levels

In the interview, I asked participants to rank the three interfaces according to their subjective preferences. All 12 participants ranked *TutVis* as their most preferred interface. At the other end of the spectrum, the *Baseline* condition had very little support, with 11 participants rating it as their least preferred of the three.

I also compared participants' tutorial selection confidence levels (reported on a 5-pt Likert scale) using Friedman's two-way ANOVA with Interface as the within-subject factor. I found a statistically significant main effect of Interface on selection confidence ($\chi^2(2) = 11.267, p = 0.004$). Posthoc comparisons (Bonferroni adjusted) indicated that participants felt more confident when using *TutVis* ($mean = 4.7, s.d. = 0.5$) than when using *Baseline* ($mean = 3.6, s.d. = 0.8, p = 0.006$). There were also trends suggesting that participants were more confident with *TutVis* than with *TutDiff* ($mean = 4.1, s.d. = 0.9, p = 0.068$), and that they were more confident with *TutDiff* than with *Baseline* ($p = 0.084$).

6.4.2. Individual Component Usage

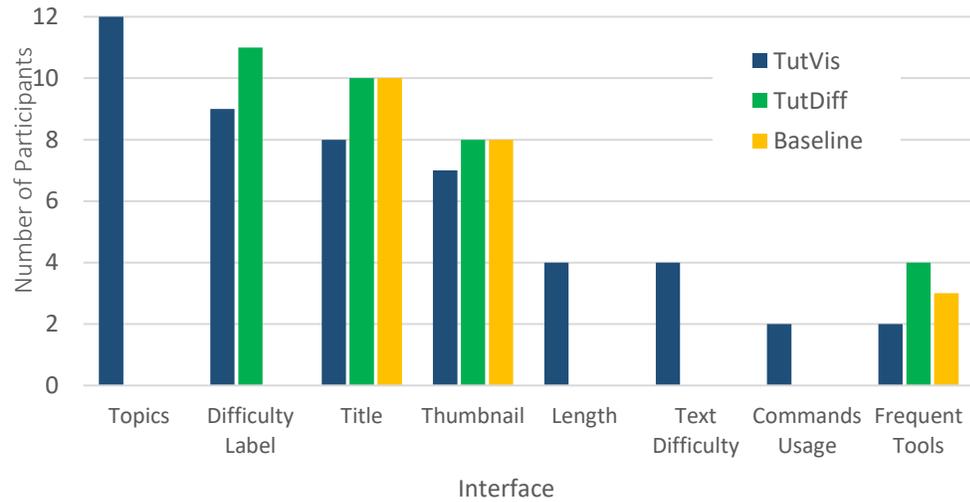


Figure 17: Self-reported interface components used.

I also investigated how participants used different interface components during the tutorial selection tasks.

Figure 17 summarizes responses from the post-condition questionnaire, which asked participants to indicate which of the available interface components they had used during that condition. Figure 17 presents data from all three conditions. However, as a reminder, not all features were available in each condition (see section 6.2 for details). When the difficulty labels were present (in *TutVis* and *TutDiff*), the majority of participants reported using them, particularly with *TutDiff* (11/12 participants). The topics, which were available in *TutVis* only, were very popular - all 12 participants reported using them in that condition. As would be expected, participants reported using title and thumbnail in all conditions. However, their reported usage of these components decreased with *TutVis*, where some seemed to instead rely more on the topic labels. Other components (e.g., length, text

difficulty, commands usage, and frequent tools) were not as heavily reported (2-4 participants depending on condition).

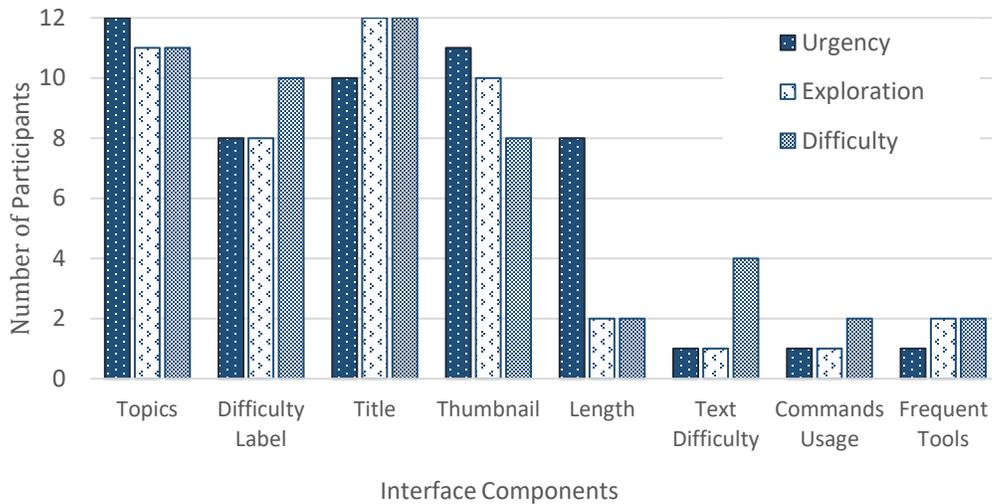


Figure 18: Interface components used in the different tasks according eye-gaze and think-aloud data (in TutVis only)

To provide further insight into how the tutorial selection scenario impacted interface component usage, I turned to the think-aloud transcripts and the eye-gaze data. To analyze the gaze data, I leveraged heatmaps generated by a software extension of Tobii [85]. I considered only those components with the longest fixation duration as determined by the application (i.e., dwells of at least 2.2 milliseconds; guided by [17]). Following previous work on combining eye-gaze and think-aloud data [12], I retained only the fixations where the participant also mentioned using the component to guide their selection. This was to disregard cases where, for example, the participant might have fixated because they found a component confusing. I instead use my interview data to shed light on components participants found confusing. I conducted this analysis on the *TutVis* data only, since this

condition contains all interface components. Figure 18 shows that while there was some variation in component usage across tasks, there were no dramatic differences. The one notable exception is the length component, which was used by 8 participants in the task that conveyed a sense of urgency, and by only 2 participants in the other tasks. Figure 18 also shows that the majority of participants used the difficulty labels in all three tasks, as opposed to only in the task that emphasized the expertise of the target user. The figure also suggests heavier reliance on the titles and thumbnails than was indicated in the self-reports.

6.4.3. Perspective on the Utility of Individual Components

The semi-structured interviews provided further insight into why participants used the different components. I elaborate on some of these reasons below. In the quotes below, *B* represents a beginner Photoshop user, *I* an intermediate, and *E* an expert.

Topics Provide a Useful Preview: Participants were enthusiastic about the topic information. One of their main reasons was that this information tended to be more useful/accurate than the title in summarizing the tutorial’s emphasis:

“... just like character design [task] [...] when you go through the topic once and there will be an animation or something like that [a topic related to animation and design] then you can know that this one is related to character design. That’s useful” – (P7-E)

The participant felt the topics served a similar function as the preface of a book:

“it is giving you a type of outline [...]. It is like a preface to a book. Like when you start reading, you should know the contents.” – (P10-B)

Difficulty Labels Help with Filtering and Uncovering Advanced Techniques:

Participants particularly liked using the difficulty labels as a way to streamline the list of tutorials to only those that would match the desired expertise level:

“He is from different background [beginner user]. He might flip if provided with more technical jargon [advanced tutorial] [...] So, it was like a more simpler way [filter by tutorial difficulty labels]” - (P10-B)

The difficulty labels were also appreciated by the expert participants, who wanted a tutorial that would go beyond just accomplishing a task:

“For event flyers [task scenario] I think that one [beginner tutorial] is really fit for the task. But in my mind, if I am doing this [...] I vote to have something more stylish more attractive [...] eye catchy. So that’s why I am choosing this [advanced tutorial]” - (P7-E)

During the interview, I also asked participants how they would feel about misclassified difficulty labels, given the classifier's overall accuracy (85%). I found participants who self-reported themselves as experts or intermediates were not concerned with misclassification. They felt that they either had the knowledge to further assess the tutorial before committing to it or could cope with various levels of difficulty:

“For me, it [misclassification] does not matter too much [...] it’s always the contents that matter the most for all” - (P4-I)

“I think that [following misclassified tutorial] is not difficult for me here because I can follow each level [advanced or beginner]” – (P7-E)

Participants worried more about misclassifications related to beginner tutorials, where it could lead to struggles in completion.

“If I am sharing a tutorial to someone else like in it said a grandparent [sharing advanced tutorial to beginners] and it is actually advanced [...] that’s not gonna be very good” – (P1-B)

Usage of Length Varied According to Participant Expertise: The length component was mostly used in the scenario with a sense of urgency (see Figure 18). The expert participants indicated that they were searching for a short tutorial because they did not need in-depth explanations of the task or tool usage:

“What I do sometimes if I need to look at something [then] length is very important. [...] if they [designers] have to first understand how to make a selection [then] there are those tutorials on YouTube that are like 1 or 2 minutes videos. Most of them go through that. They are not going to those videos that are like 30 minutes and that explain what selection tools are and how you can work” – (P5-E)

Conversely, beginner participants were more interested in the long tutorials that show step-by-step changes:

“[...] it [a short tutorial] does not describe how to create a canvas. So, this one might not be the best [...] the fantasy scene [a long tutorial] oh, it describes the tools you are [going to] use step by step [...] length is definitely helpful” – (P6-B)

Other components had limited value: Most of the participants did not use the most frequent tools (see Figure 17). Beginner participants lacked the knowledge of the tools, whereas the more advanced participants found that they got a better sense of the tutorial by looking at the topics, title, and thumbnail. Participants reported not using the text difficulty, because they felt that they could cope with various text difficulty levels. Most of my participants had difficulty understanding the command usage feature.

6.5. Summary

The results from my initial user study suggest value in providing users with both automatically generated difficulty labels and information on features that contribute to this classification. My full-featured *TutVis* interface was preferred over the *Baseline* version as well as the version with only the difficulty levels present. I also found that the full set improved selection confidence over the *Baseline*, with trends indicating that the more information users had, the more confident they seemed. My think-aloud and eye-gaze data indicated that of my novel interface components, the topics and difficulty labels were the most heavily used. The use of tutorial length during the selection process was more task- and user-dependent. Given that my other interface components (readability and command usage) had very little use or qualitative support, future versions should likely remove them to reduce the selection interface's visual complexity.

Chapter 7

Conclusion

Online tutorials are learning aids for the feature-rich software [39,43,50]. However, in the large pool of available tutorials, most of them do not have any difficulty levels, which is needed to guide different expertise of users to achieve task success [39] and remove frustration [49]. Prior work has leveraged an online voting mechanism [75] and expert's judgment [77] to find out the difficulty levels. In my approach, I investigated the feasibility of automatically labeling online tutorials as advanced or beginner leveraging machine learning techniques. In the following subsections, I conclude by summarizing my contributions and by discussing some of the future research directions of this thesis.

7.1. Contributions

The goal of my thesis was to present an automatic, machine-learning approach to labeling an online software tutorial's difficulty. In this thesis, my contribution was four-fold. I briefly summarize each of the contributions in the following paragraphs.

I initiated my investigation by collecting pre-labeled Photoshop tutorials from various sources. Guided by previous works on software expertise and learnability [30,31,46,54,75], and by conducting informal feature investigations on the collected tutorials, I settled on: topics, commands, word repetition, text difficulty, and length as my features. I analyzed the feature differences using a 2-tailed independent T-test and found that advanced tutorials are significantly lengthier and contain significantly more repeated words than the beginner tutorials.

I developed different machine learning models using the combination of my engineered feature sets. I found my best model could correctly classify advanced vs. beginner tutorials at 85% accuracy while testing with 10-fold cross-validation. My best model leveraged all feature sets, e.g., topics, command ratio, word repetition, text difficulty, and length. From my analysis on the quantity of the training data, I found out the performance of my model can still be improved with more training data. To find out the generalizability of my approach, I developed another model using Fusion 360 tutorials. I showed that this model could get accuracy up to 81%.

I investigated ways to present classifier features and its decision to the users. I demonstrated an application of my classifier by embedding it in the tutorial browsing interface *TutVis*. To represent my features length, command ratio, and text difficulty as

interface components of *TutVis*, I transformed the numerical values into low, medium, and high scale. To present my feature – topics, I devised an approach of interpreting the numerical topic distributions into high-level Photoshop topics. I uncovered some high-level differences of advanced vs. beginner topics through my analysis.

To investigate the utility of my system, *TutVis*, I conducted a tutorial selection study with 12 participants. My study findings indicated that users appreciated having information on a tutorial’s difficulty level and its high-level topics. The combination of difficulty labels and topics had the potential to be particularly powerful in the context of feature-rich software since a user’s software expertise can vary substantially according to the topics [31].

7.2. Limitations and Future Research Directions

7.2.1. Developing a Recommender System

I demonstrated an application of my classifier by embedding it in my *TutVis* tutorial browsing interface. I uncovered that users found tutorial’s difficulty and topics helpful in selecting tutorials. Beyond supporting tutorial browsing through annotations, another potential application of my classifier would be to embed it inside a recommender system. Such a system could use recent advances in expertise [29,30] and task detection [38,71,77] to automatically recommend tutorials.

7.2.2. Investigating Other Features

Since my current work mainly relies on text-based features, there are a number of opportunities to explore additional classification features. For example, my qualitative analysis suggested that a beginner tutorial might spend more time on tool demonstrations.

It might be possible to use existing techniques [64] to identify and quantify tool demonstrations in a tutorial. When looking for further properties of an advanced tutorial, the classifier might also consider references to external software, or look for references to commands that are particularly unique, as measured by community usage logs [52]. Future work could also leverage advances in computer vision to generate new visual features about tutorial difficulty by analyzing objects in images and video frames [5].

7.2.3. Reducing the Impact of Misclassification

In conjunction with exploring new tutorial features, future work can systematically examine the impact of misclassified tutorials. My study provides only high-level subject impressions of the potential implications of misclassification, which is that the classifier might need to be particularly conservative when labeling a tutorial as beginner. Novice users might be more negatively impacted by a tutorial that does not match their skill level, and they might experience greater frustration or even become discouraged. In contrast, expert users might be able to leverage their existing software knowledge to more easily detect misclassifications. One way to alleviate the impact of the misclassifications would be to augment the automatically generated labels with community-based feedback about tutorial difficulty (e.g., as explored in Vermette et al. [75]).

7.2.4. Automating Manual Effort

In considering the generalizability of my approach, I reflect on the manual effort required. My command dictionary involved some manual effort. While I could extract command names from the software, I manually annotated a subset of tutorials (70 in total) to include examples of indirect references. This command dictionary was used to calculate one of the features in my best performing model (i.e., Command Ratio). Assigning human-readable

labels to the LDA topics also involved a non-trivial amount of human labor, as I had to hand analyze a subset of tutorials within each topic to look for common themes. Given participant enthusiasm for this component of the *TutVis* browsing interface, future work could explore ways to automate this manual labeling to eliminate the need for expert inspection. One could also imagine using crowd workers [41] to assign labels, using the tutorial clustering method to guide this effort.

7.2.5. Exploring the Long Term Effect

I presented an automatic, machine-learning approach to labeling an online software tutorial's difficulty. I showed my developed tutorial features could be leveraged to classify advanced vs. beginner Photoshop tutorials at 85% accuracy. My system, *TutVis* represents only one point in the design space of how this expertise information might be used to support tutorial selection. Future work should verify the generalizability of my study findings to larger sample size. Deploying *TutVis* would also enable to collect more ecologically valid data on how *TutVis* supports real-world tutorial browsing and selection. Future work should also explore the feasibility and utility of finer-grained difficulty assessments by collecting suitably-labeled training data (e.g., advanced, intermediate, beginner tutorials) and using multi-class classifiers [27,33,66,81]. With ongoing advances in software expertise detection, my approach paves the way for new technologies that match users with online resources that best suit their current levels of software expertise.

Bibliography

1. Anshu Agarwal and Andrew Meyer. 2009. Beyond usability: Evaluating emotional response as an integral part of the user experience. In *Conference on Human Factors in Computing Systems - Proceedings*, 2919–2930. <https://doi.org/10.1145/1520340.1520420>
2. Michel Jose Anzanello and Flavio Sanson Fogliatto. 2011. Learning curve models and applications: Literature review and research directions. *International Journal of Industrial Ergonomics* 41, 573–583. <https://doi.org/10.1016/j.ergon.2011.05.001>
3. Catherine A Ashworth. 1992. Skill as the Fit Between Performer Resources and Task Demands: A Perspective from Software Use and Learning. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, 444–449.
4. V. Balakrishnan and E. Lloyd-Yemoh. 2014. Stemming and lemmatization: A comparison of retrieval performances. In *Proceedings of SCEI Seoul Conferences*.
5. Nikola Banovic, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2012. Waken: reverse engineering usage information and interface structure from software videos. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 83–92.
6. DM Blei, AY Ng, MI Jordan - Journal of machine Learning Research, and

- Undefined 2003. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3: 993–1022.
7. Doris U. Bolliger and Supawan Supanakorn. 2011. Learning styles and student perceptions of the use of interactive online tutorials. *British Journal of Educational Technology*. <https://doi.org/10.1111/j.1467-8535.2009.01037.x>
 8. Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1: 5–32. <https://doi.org/10.1023/A:1010933404324>
 9. Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics* 18, 4: 467–479.
 10. Andrea Bunt, Patrick Dubois, Ben Lafreniere, Michael Terry, and David Cormack. 2014. TaggedComments: Promoting and Integrating User Comments in Online Application Tutorials. In *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI'14*, 4037–4046. <https://doi.org/10.1145/2556288.2557118>
 11. Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2012. MixT: automatic generation of step-by-step mixed media tutorials. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 93–102.
 12. Lynne Cooke. 2010. Assessing Concurrent Think-Aloud Protocol as a Usability Test Method: A Technical Communication Approach. *IEEE Transactions on*

-
- Professional Communication* 53, 3: 202–215.
<https://doi.org/10.1109/TPC.2010.2052859>
13. M.F. Costabile, M. De Marsico, R. Lanzilotti, V.L. Plantamura, and T. Roselli. 2005. On the Usability Evaluation of E-Learning Applications. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 6b-6b.
<https://doi.org/10.1109/HICSS.2005.468>
 14. Gennaro Costagliola, Andrea De Lucia, Filomena Ferrucci, Carmine Gravino, and Giuseppe Scanniello. 2008. Assessing the usability of a visual tool for the definition of e-learning processes. *Journal of Visual Languages and Computing* 19, 6: 721–737. <https://doi.org/10.1016/j.jvlc.2008.01.003>
 15. Andrew Cross, Mydhili Bayyapunedi, Dilip Ravindran, Edward Cutrell, and William Thies. 2014. VidWiki: enabling the crowd to improve the legibility of online educational videos. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, 1167–1175.
 16. Andrew Dillon and Charles Watson. 1996. User analysis in HCI - The historical lessons from individual differences research. *International Journal of Human Computer Studies* 45, 6: 619–637. <https://doi.org/10.1006/ijhc.1996.0071>
 17. Soussan Djamasbi, Marisa Siegel, and Tom Tullis. 2010. Generation Y, web design, and eye tracking. *International journal of human-computer studies* 68, 5: 307–323.
 18. Volodymyr Dziubak, Patrick Dubois, Andrea Bunt, and Michael Terry. 2016. Switter: Supporting Exploration of Software Learning Materials on Social Media.

- In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, 1209–1220.
19. DE Egan, LM Gomez - Individual differences in cognition, and undefined 1985. Assaying, isolating, and accommodating individual differences in learning a complex skill. *Academic Press New York*.
 20. Dennis E. Egan. 1988. Individual Differences In Human-Computer Interaction. In *Handbook of Human-Computer Interaction*. Elsevier, 543–568. <https://doi.org/10.1016/B978-0-444-70536-5.50029-4>
 21. Hillel J. Einhorn. 1974. Expert judgment: Some necessary conditions and an example. *Journal of Applied Psychology* 59, 5: 562–571. <https://doi.org/10.1037/h0037164>
 22. Michael Ekstrand, Wei Li, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2011. Searching for software learning resources using application context. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, 195. <https://doi.org/10.1145/2047196.2047220>
 23. Adam E. M. Eltorai, Syed S. Naqvi, Soha Ghanian, Craig P. Ebersson, Arnold-Peter C. Weiss, Christopher T. Born, and Alan H. Daniels. 2015. Readability of Invasive Procedure Consent Forms. *Clinical and Translational Science* 8, 6: 830–833. <https://doi.org/10.1111/cts.12364>
 24. K Anders Ericsson. 2006. *An Introduction to The Cambridge Handbook of Expertise and Expert Performance: Its Development, Organization, and Content*. Cambridge

University Press.

25. Laura Faulkner and David Wick. 2005. Cross-user analysis: Benefits of skill level comparison in usability testing. *Interacting with Computers* 17, 6: 773–786. <https://doi.org/10.1016/j.intcom.2005.04.004>
26. Adam Fourney, Ben Lafreniere, Richard Mann, and Michael Terry. 2012. Then click ok!: extracting references to interface elements in online documentation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 35–38.
27. Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. 2011. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition* 44, 8: 1761–1776. <https://doi.org/10.1016/j.patcog.2011.01.017>
28. Arin Ghazarian and S. Majid Noorhosseini. 2010. Automatic detection of users' skill levels using high-frequency user interface events. *User Modeling and User-Adapted Interaction* 20, 2: 109–146. <https://doi.org/10.1007/s11257-010-9073-5>
29. Jun Gong, Fraser Anderson, George Fitzmaurice, and Tovi Grossman. 2019. Instrumenting and Analyzing Fabrication Activities, Users, and Expertise. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, 1–14. <https://doi.org/10.1145/3290605.3300554>
30. Tovi Grossman and George Fitzmaurice. 2015. An Investigation of Metrics for the

- In Situ Detection of Software Expertise. *Human–Computer Interaction* 30, 1: 64–102.
31. Tovi Grossman, George Fitzmaurice, and Ramtin Attar. 2009. A Survey of Software Learnability: Metrics, Methodologies, and Guidelines. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, 649. <https://doi.org/10.1145/1518701.1518803>
 32. Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2010. Chronicle: capture, exploration, and playback of document workflow histories. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, 143–152.
 33. Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. 2009. Multi-class AdaBoost. *Statistics and Its Interface* 2, 3: 349–360. <https://doi.org/10.4310/SII.2009.v2.n3.a8>
 34. Abram Hindle, Christian Bird, Thomas Zimmermann, and Nachiappan Nagappan. 2015. Do topics make sense to managers and developers? *Empirical Software Engineering* 20, 2: 479–515. <https://doi.org/10.1007/s10664-014-9312-1>
 35. Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. 1998. The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 256–265.
 36. Amy Hurst, Scott E. Hudson, and Jennifer Mankoff. 2007. Dynamic detection of novice vs. skilled use without a task model. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*, 271.

<https://doi.org/10.1145/1240624.1240669>

37. Caitlin Kelleher and Randy Pausch. 2005. Stencils-based tutorials: design and evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 541–550.
38. Md Adnan Alam Khan, Volodymyr Dziubak, and Andrea Bunt. 2015. Exploring personalized command recommendations based on information found in Web documentation. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, 225–235.
39. Kimia Kiani, George Cui, Andrea Bunt, Joanna McGrenere, and Parmit K. Chilana. 2019. Beyond “One-Size-Fits-All”: Understanding the Diversity in How Software Newcomers Discover and Make Use of Help Resources. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*: 1–14. <https://doi.org/10.1145/3290605.3300570>
40. Juho Kim, Philip J Guo, Carrie J Cai, Shang-Wen Daniel Li, Krzysztof Z Gajos, and Robert C Miller. 2014. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 563–572.
41. Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J Guo, Robert C Miller, and Krzysztof Z Gajos. 2014. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, 4017–4026.

42. Nicholas Kong, Tovi Grossman, Björn Hartmann, Maneesh Agrawala, and George Fitzmaurice. 2012. Delta: a tool for representing and comparing workflows. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1027–1036.
43. Ben Lafreniere, Andrea Bunt, Matthew Lount, and Michael Terry. 2013. Understanding the Roles and Uses of Web Tutorials. In *Seventh International AAAI Conference on Weblogs and Social Media*.
44. Benjamin Lafreniere, Tovi Grossman, and George Fitzmaurice. 2013. Community enhanced tutorials: improving tutorials with multiple demonstrations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1779–1788.
45. Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. 2011. Automatic labelling of topic model. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, 1536–1545.
46. Barry R. Lawson, Kenneth R. Baker, Stephen G. Powell, and Lynn Foster-Johnson. 2009. A comparison of spreadsheet users with different levels of experience. *Omega* 37, 3: 579–590. <https://doi.org/10.1016/j.omega.2007.12.004>
47. Wei Li, Tovi Grossman, and George Fitzmaurice. 2012. GamiCAD: a gamified tutorial system for first time autocad users. In *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*, 103.

<https://doi.org/10.1145/2380116.2380131>

48. Wei Li, Justin Matejka, Tovi Grossman, Joseph A. Konstan, and George Fitzmaurice. 2011. Design and evaluation of a command recommendation system for software applications. *ACM Transactions on Computer-Human Interaction* 18, 2: 1–35.
49. EA Locke, GP Latham - American Psychologist, and Undefined 2002. 2002. Building a practically useful theory of goal setting and task motivation: A 35-year odyssey. *American psychologist* 57, 9: 705.
50. Matthew Lount and Andrea Bunt. 2014. Characterizing Web-Based Tutorials: Exploring Quality, Community, and Showcasing Strategies. In *Proceedings of the 32nd ACM International Conference on The Design of Communication CD-ROM*, 6.
51. M Masarakal. 2010. Improving expertise-sensitive help systems. Retrieved May 7, 2018 from <http://ecommons.usask.ca/handle/10388/etd-03152010-120307>
52. Justin Matejka, Wei Li, Tovi Grossman, and George Fitzmaurice. 2009. CommunityCommands: command recommendations for software applications. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology - UIST '09*, 193. <https://doi.org/10.1145/1622176.1622214>
53. Richard E. Mayer and Roxana Moreno. 2003. Nine Ways to Reduce Cognitive Load in Multimedia Learning. *Educational Psychologist* 38, 1: 43–52. https://doi.org/10.1207/S15326985EP3801_6

54. J McGrenere and G Moore Interface. 2000. Are we all in the same "bloat"? *Graphics interface 2000*: 187--196.
55. Qiaozhu Mei, Xuehua Shen, and Chengxiang Zhai. 2007. Automatic Labeling of Multinomial Topic Models. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*.
56. David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew Mccallum. 2011. Optimizing Semantic Coherence in Topic Models. In *Proceedings of the conference on empirical methods in natural language processing*, 262–272.
57. Daša Munková, Michal Munk, and Martin Vozár. 2014. Influence of Stop-Words Removal on Sequence Patterns Identification within Comparable Corpora. In *International Conference on ICT Innovations*, 67–76. https://doi.org/10.1007/978-3-319-01466-1_6
58. David Newman, Jey, Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic Evaluation of Topic Coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 100–108.
59. David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. 2010. Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries - JCDL '10*, 215. <https://doi.org/10.1145/1816123.1816156>
60. Jakob. Nielsen and Jakob. 1993. *Usability engineering*. AP Professional.

-
61. Richard E. Nisbett and Timothy D. Wilson. 1977. Telling more than we can know: Verbal reports on mental processes. *Psychological Review* 84, 3: 231–259. <https://doi.org/10.1037/0033-295X.84.3.231>
 62. Fred Paas, Alexander Renkl, and John Sweller. 2003. Cognitive Load Theory and Instructional Design: Recent Developments. *Educational Psychologist* 38, 1: 1–4.
 63. Amy Pavel, Floraine Berthouzoz, Björn Hartmann, and Maneesh Agrawala. 2013. Browsing and Analyzing the Command-Level Structure of Large Collections of Image Manipulation Tutorials. In *Citeseer, Tech. Rep.*
 64. Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F Cohen. 2011. Pause-and-play: automatically linking screencast video tutorials with applications. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 135–144.
 65. Luca Ponzanelli, Gabriele Bavota, Andrea Mocci, Massimiliano Di Penta, Rocco Oliveto, Mir Hasan, Barbara Russo, Sonia Haiduc, and Michele Lanza. 2016. Too long; didn't watch!: extracting relevant fragments from software development video tutorials. In *Proceedings of the 38th International Conference on Software Engineering*, 261–272.
 66. Anita Prinzie and Dirk Van den Poel. 2008. Random Forests for multiclass classification: Random MultiNomial Logit. *Expert Systems with Applications* 34, 3: 1721–1732. <https://doi.org/10.1016/J.ESWA.2007.01.029>
 67. Philipp Probst, Marvin N. Wright, and Anne-Laure Boulesteix. 2019.

- Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9, 3. <https://doi.org/10.1002/widm.1301>
68. Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, 248--256.
69. Vidya Ramesh, Charlie Hsu, Maneesh Agrawala, and Björn Hartmann. 2011. ShowMeHow: translating user interface instructions between applications. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 127–134.
70. Arif Raza, Luiz Fernando Capretz, and Faheem Ahmed. 2012. An open source usability maturity model (OS-UMM). *Computers in Human Behavior* 28, 4: 1109–1121. <https://doi.org/10.1016/j.chb.2012.01.018>
71. J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*. Springer, 291–324.
72. Alexandra Schofield, Måns Magnusson, and David Mimno. 2017. Pulling Out the Stops: Rethinking Stopword Removal for Topic Models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 432–436.
73. Anselm Strauss and Juliet Corbin. 1990. *Basics of qualitative research*. Sage

publications.

74. Alper Kursat Uysal and Serkan Gunal. 2014. The impact of preprocessing on text classification. *Information Processing & Management* 50, 1: 104–112. <https://doi.org/10.1016/J.IPM.2013.08.006>
75. Laton Vermette, Shruti Dembla, April Y Wang, Joanna Mcgrenerere, and Parmit K Chilana. 2017. Social CheatSheet: An Interactive Community-Curated Information Overlay for Web Applications. In *Proceedings of the ACM: Human-Computer Interaction (I,I), Computer-Supported Cooperative Work and Social Computing (CSCW)*.
76. Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009. Evaluation Methods for Topic Models. In *Proceedings of the 26th annual international conference on machine learnin*, 1105--1112.
77. Xu Wang, Benjamin Lafreniere, and Tovi Grossman. 2018. Leveraging Community-Generated Videos and Command Logs to Classify and Recommend Software Workflows. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 285. <https://doi.org/10.1145/3173574.3173859>
78. Geoffrey I. Webb, Claude Sammut, Claudia Perlich, Tamás Horváth, Stefan Wrobel, Kevin B. Korb, William Stafford Noble, Christina Leslie, Michail G. Lagoudakis, Novi Quadrianto, Wray L. Buntine, Novi Quadrianto, Wray L. Buntine, Lise Getoor, Galileo Namata, Lise Getoor, Xin Jin, Jiawei Han, Jo-Anne Ting, Sethu Vijayakumar, Stefan Schaal, and Luc De Raedt. 2011. Learning Curves in Machine

- Learning. In *Encyclopedia of Machine Learning*. Springer US, Boston, MA, 577–580. https://doi.org/10.1007/978-0-387-30164-8_452
79. Sarah Weir, Juho Kim, Krzysztof Z Gajos, and Robert C Miller. 2015. Learnersourcing subgoal labels for how-to videos. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 405–416.
80. Ryen W. White, Susan T. Dumais, and Jaime Teevan. 2009. Characterizing the influence of domain expertise on web search behavior. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining - WSDM '09*, 132. <https://doi.org/10.1145/1498759.1498819>
81. Ting-Fan Wu, Chih-Jen Lin, and Ruby C. Weng. 2004. Probability Estimates for Multi-class Classification by Pairwise Coupling. *Journal of Machine Learning Research* 5, Aug: 975–1005.
82. gensim: Topic modelling for humans. Retrieved September 19, 2019 from <https://radimrehurek.com/gensim/>
83. sklearn.model_selection.StratifiedKFold — scikit-learn 0.21.3 documentation. Retrieved October 1, 2019 from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html
84. Learning Curves - Advice for Applying Machine Learning | Coursera. Retrieved November 5, 2019 from <https://www.coursera.org/lecture/machine-learning/learning-curves-Kont7>

85. Tobii Ghost - Stream with Eye Tracking. Retrieved September 11, 2019 from <https://gaming.tobii.com/software/ghost/>

Appendix A

Clear Topics

Topic Number	Name	Top 10 Words
T1	Drawing Pixel Art	pencil, pixel_art, isometric, character, outline, shade, volume, diagonal, extend, define
T5	Flyer & Poster Design	poster, page, vintage, indesign, typography, bleed, pantone, poster_design, file_export, paper
T11	Introducing Interface & Basics	menu_choose, interface, option, edit, crop, workspace, dialog_box, hover, check_mark, panel
T15	Shading, Texture & Color Blending	splash, gradient_map, hardness_flow, explosion, multiply, blend_mode, stylish_light, alt_clipping, palette, overlay
T16	Masking & Selection	quick_selection, mask, smart_radius, subtract_selection, check_colorize, refine_edge, lasso_tool, fine_tune, stamp_tool, refinement
T22	Introducing Layers & Colors	brightness, rgb, histogram, channel, adjustment, contrast, highlight, correction, curve, percentage
T25	Photo Retouching Techniques	photograph, compare, healing_brush, bridge, feature, retouch, detail, option, show, important_thing
T26	File Organization, Share and Export	library, profile, web, collection, facebook, creative_cloud, plug, download, update, save
T29	Animation & Video Effects	animation, timeline, frame, video, gif, playback, loop, glare, motion, outline

Appendix B

Similar Topics

Topic Number	Name	Top 10 Words
T2+T13	3D Designs & Effects	extrusion, cinema, modo, diffuse_texture, high_pass, bitmap, texture, render, viewport, polygon
T8+T21	Generation of Objects & Graphic Patterns	ruler, clipping, smart_object, gaussian_blur, apply_transformation, mockup, canvas, neon_tube, shape, filter
T3+T12+T17+T24+T30	Photo Composite & Manipulation	building, import_asset, matte_painting, resize_position, free_transformation, scene, smoke, source-folder, lasso, rgb_composite
T4+T19+T23	Photo Editing, Manipulation & Special Effects	thumbnail, manipulation, brightness_contrast, man_portrait, threshold_level, camera_raw, puppet_wrap, effect, adjust, subject
T6+T27+T28	Sketching & Digital Painting	color, digital_art, artist, rough_sketch, expression, line_art, skin_tone, incorporate, motif, focal_point
T9+T10	Design Shapes & Artwork	elliptical_marquee, contract, selection_active, feather, circle, geometric, illustrator, design, stroke, triangle
T14+T20	Text Effects	preset_manager, angle_distance, bevel_emboss, text, style, pattern_overlay, global_setting, rasterize_type, font, write

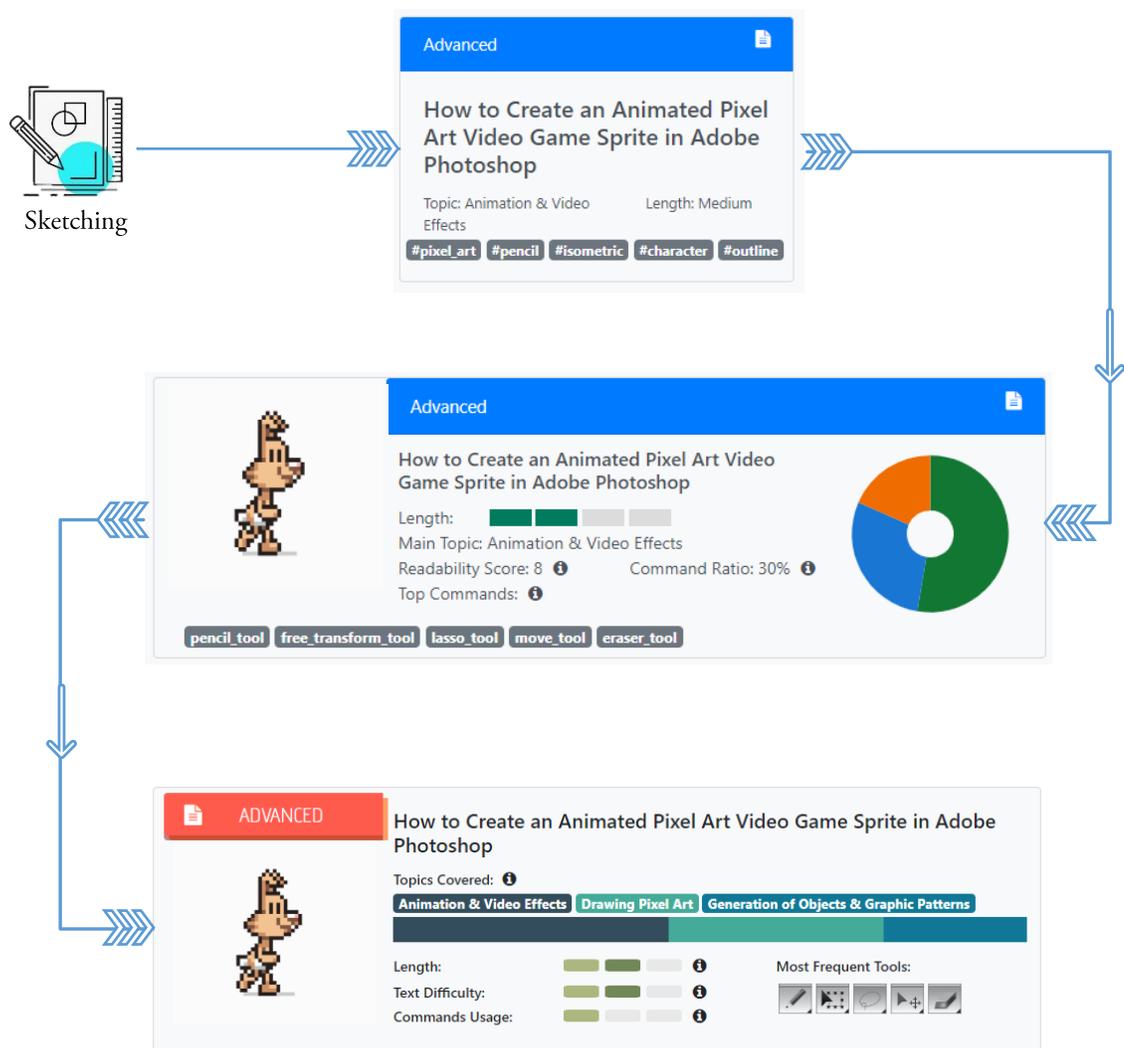
Appendix C

Fuzzy Topics

Topic Number	Name	Top 10 Words
T7	MIX: Editing & Transformation	step, copy, add, fill, duplicate, merge, position, warp, resource, move
T18	MIX: Editing & Selection	sort, drag, difficult, stuff, command, powerful, bunch, fact, great, hit

Appendix D

Evolution of the Tutorial Representation



Appendix E

Research Ethics Board Approval



Human Ethics
208-194 Dafoe Road
Winnipeg, MB
Canada R3T 2N2
Phone +204-474-7122
Email: humanethics@umanitoba.ca

PROTOCOL APPROVAL

TO: Andrea Bunt
Principal Investigator

FROM: Julia Witt, Chair
Joint-Faculty Research Ethics Board (JFREB)

Re: Protocol J2019:030 (HS22757)
"Interfaces for Online Tutorial Selection"

Effective: May 13, 2019

Expiry: May 13, 2020

Joint-Faculty Research Ethics Board (JFREB) has reviewed and approved the above research. JFREB is constituted and operates in accordance with the current *Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans*.

This approval is subject to the following conditions:

1. Approval is granted for the research and purposes described in the application only.
2. Any modification to the research or research materials must be submitted to JFREB for approval before implementation.
3. Any deviations to the research or adverse events must be submitted to JFREB as soon as possible.
4. This approval is valid for one year only and a Renewal Request must be submitted and approved by the above expiry date.
5. A Study Closure form must be submitted to JFREB when the research is complete or terminated.
6. The University of Manitoba may request to review research documentation from this project to demonstrate compliance with this approved protocol and the University of Manitoba *Ethics of Research Involving Humans*.

Funded Protocols:

- Please mail/e-mail a copy of this Approval, identifying the related UM Project Number, to the Research Grants Officer in ORS.

Appendix F

TCPS 2: CORE Certificate



Appendix G

Poster Advertising the Study



We are looking for participants who have working experience with
Photoshop

Purpose of the Study:

You are invited to participate in a study that aims to explore different ways to help people find Photoshop tutorials. The study will take approximately 90 minutes.

Role of the Participants:

You will be asked to select tutorials under different scenarios using different interfaces.

Help us with our study and get \$20 in cash or gift cards!

Research approved by the University of Manitoba Joint Faculty Research Ethics Board

Tutorial Selection Study

Appendix H

Consent Form



DEPARTMENT OF COMPUTER SCIENCE

Winnipeg, Manitoba
Canada R3T 2N2

UNIVERSITY
OF MANITOBA

Research Project Title: An Investigation on Assessing the Expertise Level of Application Tutorials

Researchers:

Dr. Andrea Bunt, Associate Professor, Department of Computer Science, University of Manitoba,

Shahed Anzarus Sabab, Masters Research Assistant, Department of Computer Science, University of Manitoba

Please take the time to read this carefully and to ensure you understand all the information.

This consent form, a copy of which will be left with you for your records and reference, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, you should feel free to ask. Please take the time to read this carefully and to understand any accompanying information.

The goal of this study is to gather information about different prototypes designed to help users choose appropriate tutorials. If you have any questions or concerns at this time or any time during the study, please feel free to ask the researcher for clarification.

As, a part of this study, you will be given some scenarios and will be asked to choose a Photoshop tutorial for that given scenario from a set of potential candidates. For each scenario, you will be provided with a prototype that displays some information about each tutorial. At the end of the tasks, we will ask you to participate in a semi-structured interview about your experiences with the prototypes. The whole study will take approximately 90 minutes. The risks of this study will be no greater than every life.

Participation in this study is voluntary. After signing this consent form, you will receive a \$20 gift card or \$20 in cash for your participation.

Data collected for this study will be retained for a period of maximum three years in a locked cabinet or password-protected computer in a locked office or laboratory in the EITC building, University of Manitoba, to which only researchers associated with this study (Dr. Andrea Bunt and Shahed Anzarus Sabab) have access. In addition, the University of Manitoba may look at research records to see that the research is being done in a safe and proper way. We intend to present results as part of a thesis and as academic publications. Again, no personal information about your involvement will be included.



UNIVERSITY
OF MANITOBA

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a subject. By doing this you also confirm that you are at least 17 years of age. In no way does this waive your legal rights nor release the researchers, sponsors, or involved institutions from their legal and professional responsibilities. During the study you are free to withdraw at any time, and/or refrain from answering any questions you prefer to omit. After the study, you can withdraw your feedback within 2 days of your participation.

Even by withdrawing, you will keep your compensation. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation.

This research has been approved by the University of Manitoba Joint Faculty Research Ethics Board. If you have any concerns or complaints about this project you may contact Dr. Andrea Bunt at _____ the Human Ethics Coordinator (HEC) at _____ or at _____. A copy of this consent form has been given to you to keep for your records and reference.

We wish to record our discussions using a standard voice recorder. The audio will serve as a reference point in our data analysis, allowing us to review the discussion in detail. We also wish to record your eye gaze using the vision-based Tobii Eye Tracker 4c eye-tracker. This will provide us with information on the interface components that you looked at during the study tasks. Any information you choose to contribute is completely confidential and will be used for anonymized research analysis. We may use anonymized quotes for purposes of dissemination; your name will not be included or in any other way associated with the data presented in the results. By signing this consent form, you agree that you understand this and that we may use the recorded audio for data analysis purposes only.

- I wish to receive a summary of the findings.
- I wish to receive a copy of the transcript of the audio recording to confirm its accuracy.

Please write your email address if you checked a box above:

Participant's email address: _____

Participant's signature: _____ Date: _____

Researcher's signature: _____ Date: _____

Appendix I

Instructions for Different Prototypes

Prototype	Instructions
<i>Baseline</i>	This interface presents title of the tutorial, output image (what will the tutorial be creating), most frequently used tools, types of the tutorial (either it is text or video) in the tutorial's list. You can search by the title of the tutorials by using the search bar. You can hover over any of the items or icons for more information.
<i>TutDiff</i>	This interface presents the title of the tutorial, output image (what will the tutorial be creating), most frequently used tools, types of the tutorial (either it is text or video), and a system-generated assessment of the difficulty of the tutorial. You can search by the title of the tutorials by using the search bar. You can also filter the tutorials by advanced or beginner from the left panel. You can hover over any of the items or icons for more information.
<i>TutVis</i>	This interface presents the title of the tutorial, output image (what will the tutorial be creating), most frequently used tools, types of the tutorial (either it is text or video). In addition to this information, this interface provides automatically-generated information such as the difficulty of the tutorial, the covered topics, length, text difficulty, and commands used. You can hover over any of the items or icons for more information. You can search for tutorials by using the search bar, or you can also filter tutorials by the difficulty levels and/or topics they cover.

Appendix J

Isomorphic Scenarios for Tutorial Selection Tasks

Set	Specific Task	Exploratory Task	Difficulty Task
Set 1	Suppose you are assigned the task of creating an advertisement for a fundraising occasion. You want to complete this task quickly. Select a tutorial that you think would serve as the best starting point for you.	Suppose you are free for the whole afternoon, and you are interested in learning about digital drawing. Find a tutorial that would give you some insight into digital drawing.	Suppose you have a friend who has never used Photoshop before. Recently, he asked for your help in finding tutorials on how to change an image background. Find a suitable tutorial for your friend.
Set 2	Suppose you and your friends are planning to make a T-shirt for an upcoming event. You want to design a logo for the T-shirt. You want to complete this task quickly. Find a tutorial that could help you to get some ideas on how to design the logo.	Suppose you have recently been inspired by the scenery in a Sci-Fi movie, and you would like to create something similar using stock images. You have got your weekend free, and you intend to dedicate your time into it. Find a tutorial that would help you to explore your imagination.	Suppose you are a professional. You have a new client who wants you to add a new filter to his portrait. Find a tutorial that you can follow to ensure high-quality output for your client.
Set 3	Suppose you are working on a gaming project with a tight deadline. Your current project requires you to create a character for your upcoming game. Now find a tutorial that would help you to create the character.	Suppose you have got two days off from your office. You want to invest your free time to create a piece of digital art to add to your portfolio. Find a tutorial that can serve as a starting point.	Suppose one of your grandparents, who is not tech-savvy recently asked your help to guide him in compiling a video in Photoshop. Find a tutorial which can help your grandparent to finish the task.

Appendix K

Demographics Questionnaire

Please enter your age *

Short answer text

Enter your gender you identify as *

- Male
- Female
- Others

How often do you use Photoshop? *

- Daily
- Once a week
- Once a month
- Less than once a month

How would you define yourself as a Photoshop user? *

- Novice
- Beginner
- Intermediate
- Expert

What do you use Photoshop for? *

- 3D Designs & Effects
- Animation & Video Effects
- Designing flyer, poster, magazine.
- Design Shapes and Artwork
- Generation of Objects & Graphic Patterns
- Photo composites
- Photo Editing, Retouching & Manipulation
- Pixel Art
- Shading, Texture & Color Blending
- Sketching and digital painting
- Text Effects
- Other...

Appendix L

Study Questionnaire

Interface Condition: *Baseline*



Please select the interface components you used when searching for tutorials? *

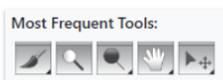
Search Box



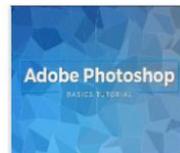
Title of the tutorial



Most Frequent Tools



Thumbnail Image



Tutorial Type (Video/Text)



Other...

How confident are you that you picked the correct tutorials in the tasks? *

not confident at all 1 2 3 4 5 completely confident

Interface Condition: *TutDiff*



Please select the interface components you used when searching for tutorials? *

Search Box



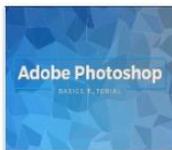
Filter



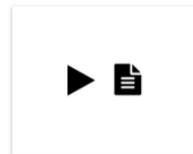
Title of the tutorial



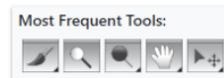
Thumbnail Image



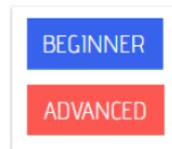
Tutorial Type (Video/Text)



Most Frequent Tools



Difficulty of the tutorial (Advanced/Beginner)



Other...

How confident are you that you picked the correct tutorials in the tasks? *

not confident at all 1 2 3 4 5 completely confident

○ ○ ○ ○ ○

Interface Condition: *TutVis*



⋮

Please select the interface components you used when searching for tutorials? *

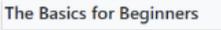
Search Box



Filter



Title of the tutorial



Thumbnail Image



Tutorial Type (Video/Text)



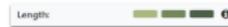
Most Frequent Tools



Difficulty of the tutorial (Advanced/Beginner)



Length



Text Difficulty



Commands Usage



Topics



Other...

How confident are you that you picked the correct tutorials in the tasks? *

1 2 3 4 5

not confident at all ○ ○ ○ ○ ○ completely confident

Appendix M

Semi-Structured Interview Sample Questions

- Which of these prototypes did you like? Why?
- Can you recall any interface components which seem useful to you while looking for the tutorials? How are they helpful?
- Which of the components you did not find useful? Why?
- How did you feel about the auto-generated information?
- Did you trust that they were accurate? Why/Why not?
- What if the information is incorrect? How much of an issue would this be for you?
- What was the prototyping missing that would have helped you select a tutorial?
- Can you rank the three interfaces based on your preference? What is your reasoning behind this preference?