Presentation Slides
Recommender System Design

YIMING FAN
Presentation Slides
Recommender System
Design

YIMING FAN

Master in Machine Learning
Date: June 30, 2018
Supervisor: Mario Romero Vega
Examiner: Tino Weinkauf
Swedish title: Design av ett rekommendationssystem för presentationsbilder
School of Electrical Engineering and Computer Science
Abstract

Using presentation software such as PowerPoint or Keynote to support lectures and presentations has become ubiquitous in both academia and industry. However, designing the visuals of presentation slides is a time-consuming and laborious task; repetitive steps are required for selecting templates, organizing objects, and optimizing layouts. To alleviate these laborious works and to allow users to focus on preparing the contents of presentations, we present SmartPPT, a framework that supports the automatic generation of presentation slides from textual outline. We built a Recommender System model inside the framework that could pick up slide templates for input textual outline. To test its functionality and efficiency, two sets of user study procedures were conducted and shown that SmartPPT is time efficient in generating slides and it outperforms in user satisfaction when compared to user-handcrafted slides and the baseline condition, which was PowerPoint suggested templates.

Sammanfattning
Contents

1 Introduction 2
  1.1 Research Question 3

2 Literature Review 4
  2.1 Recommender Systems 4
  2.2 Deep Learning in Recommender Systems 5
  2.3 Computer-aided Design 5
  2.4 Automatic presentation slide generation 6
  2.5 Knowledge Base 7

3 Research Methods 8
  3.1 Terminology 8
    3.1.1 Presentation slide 8
    3.1.2 Input text 10
    3.1.3 The framework 11
  3.2 Workflow of SmartPPT 11
    3.2.1 Extracting features of input text and templates 12
    3.2.2 Recommendation 13
    3.2.3 Compilation 16
  3.3 User study 17
    3.3.1 Overview of participants 18
    3.3.2 Design differences between user study 1 & 2 18
    3.3.3 Explanation on variables related to elapsed time 19
    3.3.4 Explanation on variables related to user satisfaction 20
    3.3.5 Detailed explanation on feedback 21

4 Results and Discussion 23
  4.1 Overview 23
  4.2 Execution time analysis 23
4.3 User satisfaction analysis ........................................ 24
4.4 Detailed feedback analysis ...................................... 26

5 Conclusion .......................................................... 28

Bibliography .......................................................... 30

A List of input text features ........................................ 35

B List of presentation template features .......................... 37

C User study design .................................................. 39
  C.1 User study 1 design ............................................. 39
    C.1.1 Step 1 ...................................................... 39
    C.1.2 Step 2 ...................................................... 39
    C.1.3 Step 3 ...................................................... 40
    C.1.4 Step 4 ...................................................... 45
  C.2 User study 2 design ............................................. 46
    C.2.1 Step 1 ...................................................... 46
    C.2.2 Step 2 ...................................................... 46
    C.2.3 Step 3 ...................................................... 46
    C.2.4 Step 4 ...................................................... 47
    C.2.5 Step 5 ...................................................... 47
    C.2.6 Step 6 ...................................................... 48
Acknowledgement

Words cannot fully describe my gratitude towards people who provided help while completing my thesis. First, I wish to express my sincere thanks to my thesis supervisor Prof. Mario Romero Vega. It is often challenging to conduct a master’s thesis project outside Sweden. Without our smooth and inspiring remote communication, the thesis work could not have been completed so successfully.

I would also like to thank my thesis examiner Prof. Tino Weinkauf and Prof. Ann Bengtsson, for their valuable suggestions on writing the thesis document.

I am also grateful to Prof. Nan Cao and my fellow labmates, for providing me with all the necessary assistance for the thesis project and sharing sincere and valuable guidance and encouragement to me.

I also thank my parents for the unceasing, long-term support and care. Finally, I am grateful to have such a fiancée, Ms. Yanshan Ji, who accompanied me throughout this journey. Life might be cruel and has no answer after all, but our shared experience will remain, and make us stronger.
Chapter 1

Introduction

We make presentations every day and everywhere. It is essential to design informative and aesthetically pleasing presentation slides to succeed in communication. Currently, widely used software tools for designing slides include: Microsoft PowerPoint, Google Slides, Keynote, and WPS Office. In this thesis, I do not address the style of slides created through \LaTeX.

Usually, there are a number of steps between an idea and a good slide. First, we must transform our ideas into text (or image, video and charts if possible). Then, we must fit the text into suitable, well-selected slide templates. After that, we need to combine slides into a complete presentation, which should:

- Tell a complete story (informative and logically coherent)
- Be perceptually satisfactory (no obvious bugs in design)
- Be aesthetically pleasing (e.g. coherent style / design language)

Generally speaking, the most time consuming task is to choose suitable slide templates for all slides. How are the sentences (and words) related to each other? Do they follow chronological order (such that a “timeline” could suit them)? Is it likely that the sentences fit the scene of a scientific report or an informal occasion? All of the above questions are to be considered when choosing a template.

This has been a difficult task where people spend too much time on choosing and fine-tuning templates for slides. Researchers and scientists typically have to edit their slides manually before presenting on workshops and conferences.
So, what could it be, if a Computer-Aided Design framework could automatically pick up templates and output well-prepared slides from them? The framework could take structured text as input, extract the features inside, and generate a set of most suitable templates via an embedded Recommender System model. Given a list of recommended slides, users could select one or two slide styles with their favored template. As a result, the effort of choosing and fine-tuning proper templates would be well reduced. This thesis formally and rigorously explores these conjectures.

1.1 Research Question

If we employ a Recommender System in a Computer-Aided Design (CAD) framework for presentation slides, what are the costs and benefits measured by task completion time, user experience, and objective and subjective quality of the using the recommender system for slide generation?

The main research question above introduces two subquestions. One, what are the visual features we need to extract from a slide template before employing the recommender system? And two, what are the structural features we need to extract from the input text before inserting them into a slide template?

This thesis follows this structure: section 2 reviews the most relevant related work in order to contextualize this work; section 3 describes the research methods employed to address the main research questions, including formal empirical studies; section 4 presents the results of the empirical studies; section 5 summarizes the conclusion from the current work.
Chapter 2

Literature Review

We present the design of a CAD framework embedded with a recommender system model. Within this context, we will list the work that represents the state-of-the-art of recommender systems in the following two sections. As for the field of CAD, we will list the work that provides useful criterias and guides for the implementation of our framework. Since our framework focuses on automatic presentation slide generation, some milestone papers of that field are also listed as guidelines.

2.1 Recommender Systems

Recommender systems arise from the need of e-commerce. It seeks to predict a user’s potential “preference” or “rating” over an item based on the history of preferences from that user. In recent years, recommender systems have gained increasing attention from academia and industry [19]. Some well-designed recommender systems have brought significant profits [9].

The techniques used in recommender systems are usually categorized into two groups: collaborative filtering (CF) [8] and content-based recommendations [4]. Collaborative filtering seeks to predict user’s preference using previous preference information from many users. In content-based recommendations, items are described by some keywords, and user’s preference is based on her profile indicating which types of items she prefers. Among collaborative filtering techniques, matrix factorization (MF) is the most popular [1] and produces more accurate recommendation results [14]. We will discuss the details
of matrix factorization in the next section.

Typical types of machine learning algorithms used in collaborative filtering include Bayesian ([6], [7]), Decision Tree ([17]), Neighbor-based algorithms ([11]), Neural Networks ([2]) and so on. Some of the algorithms try to reduce the collaborative filtering problem into a classification problem, and classifiers such as support vector machines (SVM) work in some cases [16]. However, traditional machine learning algorithms for collaborative filtering are gradually being replaced by newly designed deep learning algorithms, which this report will introduce at a later point.

2.2 Deep Learning in Recommender Systems

In recent years, deep learning (DL) has made breakthrough in many fields [27], and has been applied to solve collaborative filtering in some researches.

Deep learning models play the role of matrix factorization in collaborative filtering. Users and items are mapped into latent factors via transformations inside deep networks. Consequently, a deep network should be well-trained before being able to produce reasonable recommendations.

The most widely applied deep learning models include: convolutional neural networks (CNN) ([28], [10], [13]), recurrent neural networks (RNN) ([12], [25]) and restricted Boltzmann machine (RBM) [20]. The recent uprising surge of generative adversarial networks (GAN) also floods into the field of recommender systems [24]. Some recommender systems rely solely on deep learning models to make predictions, while some researches integrate deep learning with traditional recommender system models, such as tightly coupled models [24].

2.3 Computer-aided Design

In our CAD framework, input text will be filled into recommended templates to form complete slides. Output slides should be aesthetically pleasing to meet user satisfaction. Yang et al. [26] have summarized style points on generating visual-textual presentation layouts (e.g. magazine covers, posters, PowerPoint slides, etc.).
• **Textual information completeness**: elements should not exceed background boundaries or overlap each other.

• **Visual information maximization**: Images should have proper sizes that preserve important visual information.

• **Spatial layout reasonableness**: Positions of textual elements should obey some aesthetic principles.

• **Perception consistency**: Texts should have distinctive text sizes, fonts, and have high contrast to the background color.

• **Color harmonization**: Similar to Perception Consistency, the combination of colors of elements should be harmonious.

• **Textual information readability**: Textual elements should have proper sizes.

We apply recommender systems to mimic the process of interactive layout suggestions. O’Donovan et al. [18] present a system which produces two types of suggestions: refinement suggestions (small improvement on the current layout) and brainstorming suggestions (layouts of various styles).

### 2.4 Automatic presentation slide generation

Our CAD framework will output presentation slides from structured input text. Below we will list those work that inspire us before implementation.

Masao et al. [15] presented an approach to generate presentation slides from semantically annotated documents. In 2005, Shibata et al. [21] provided a way to generate slides from raw input text. Raw input text is separated into topic- and non-topic parts, and presentation slides are generated with respect to those parts. Sravanthi et al. [22] presented a framework to generate slides from \LaTeX documents. In their framework, documents are first parsed into XML format, then its information is compressed and summarized through a summarizer. Presentation slides are eventually outputted using the summarized content.
Those papers mentioned above are good examples of workflow that transforms input text into presentation slides. Since our framework will focus on matching slide templates with input text, we will presume that input text is well summarized and structured before inputting into the framework.

2.5 Knowledge Base

In this part we will list some other papers that inspired us during literature review.

Our master thesis will serve as part of a PowerPoint slides project, which includes slides recommendation, layout generation, layout refinement, etc. Some work might inspire us in the process of slide layout generation: Cao et al. [5] provided a probabilistic model to automatically generate stylistic manga layout. Qiang et al. [18] presented an approach to generate scientific paper posters. Tokumaru et al. [23] provided a system to facilitate the design of harmonious colors. Beamer and Girju’s work [3] focuses on the process of slide to paper alignment.
Chapter 3

Research Methods

To address the research question, we implemented a CAD framework called SmartPPT. It takes structured text as input, and returns presentation slides as output. In this part, we will show the overall workflow of SmartPPT and how our recommender system works inside the framework. Moreover, we will briefly illustrate the user study methods that address our research question.

3.1 Terminology

Here we provide stipulative, detailed explanation on some terminologies.

3.1.1 Presentation slide

A presentation slide (sometimes “slide” for short) is a single page containing text, shapes, images and/or charts used for presentation.

Slide template

A slide template (sometimes “template” for short) is a single page containing shapes, images and/or charts. A template may contain placeholder text for replacement.

Bullet component

A bullet component is defined as a piece of replicable component in a slide template.
CHAPTER 3. RESEARCH METHODS

Figure 3.1: A slide template. There are 5 bullet components inside the template.

Figure 3.2: Baseline slides.

**Recommended slide**

A recommended slide is defined as a presentation slide generated through the CAD framework.

**Recommended template**

A set of recommended templates are templates which is the output of recommender system. Those templates need to be “compiled” with input text to form recommended slides.

**Baseline slide**

A baseline slide is a slide generated using templates contained in Microsoft PowerPoint, **without** or with slight modification.
3.1.2 Input text

The input text of SmartPPT framework is structured with hierarchical order. Below shows an example of hierarchical input text.

- Page 1 title ... level 0 line
  - Level 1 content ... level 1 line
    - Lorem ipsum ... level 2 line
    - Lorem ipsum ... level 2 line
  - Another level 1 content ... level 1 line
    - Lorem ipsum ... level 2 line
- Page 2 title ... level 0 line
  - Level 1 content: Lorem ipsum, lorem ipsum, lorem ipsum ... level 1 line

Line

A line is the basic component of structured input text. It contains a few tabs, a short dash and its content.

The “level” of a line is defined by the number of tabs before the dash of the line. Every line belongs to the first line above it that has lower level. Lines with level 0 do not belong to any lines; they are page titles.

Page

Each page of content is separated by level 0 lines. For example, there are two sets of recommended slides generated with the above example input.

If we consider input text as having forest-like structure, then each page could be treated as a tree, whose root is their respective level 0 line.

Bullet

Each bullet content inside one page is separated by level 1 lines. In the process of compilation, each bullet corresponds to a piece of replicable component in a template.
3.1.3 The framework

Recommendation process

The recommendation process (sometimes “recommendation” for short) is the process of matching input text with suitable slide templates. Each page of content will be recommended with a set of (usually more than one) recommended templates.

Compilation process

The compilation process (sometimes “compilation / compile” for short) is the process of combining recommended templates with input text.

3.2 Workflow of SmartPPT

Figure 3.3 shows the workflow of our framework. Generally there are three steps before recommended slides are given:
• Step 1: **Initialization.** Input text are parsed, and key features of input text and templates are extracted.

• Step 2: **Recommendation.** Pre-trained Machine Learning model is loaded. Given the feature of input text and templates, the model provide recommended templates. The result of prediction is an array of presentation slide template(s).

• Step 3: **Compilation.** Slides are generated using recommended template(s).

In the following sections, we will illustrate those steps in detail.

### 3.2.1 Extracting features of input text and templates

Feature extraction and pruning is a crucial part in Machine Learning. We need to manually expand features of input text and templates into multidimensional vector space, in order for models to understand them.

**Extracting features of input text**

We extract structural features of input text, and align those features into a vector. The feature vector of input text concerns about its structural features only, meaning that we simply ignore the lexical and semantic features inside the context (that is, we have less to do with Natural Language Processing).

**Extracting features of templates**

Features of templates that are within our care include:

- Position of shapes;
- Height, width, area and their ratio (divided by height and width of slides);
- Number of shapes inside a replicable component (a “bullet”);
- If bullets are left aligned and/or right aligned;
- Colors of shapes.

Those features will be also aligned into vectors. We will provide detailed lists of features of input text and templates in appendix part.
3.2.2 Recommendation

In this step, recommender system will compare features of input text with features of various slide templates. The model returns a list of slide templates that better match the input text. In the context of recommender system, input text is seen as “user” and templates are considered as “items”. Therefore, our task can be described as to select suitable items (slide templates) for a certain user (input text).

In order to deeper address our research question, two recommender system models are implemented using SVM and CNN respectively. In the following sections, we will introduce those models, how they are trained and how they provide recommendations with different approaches.

During runtime of the framework, user could manually select which recommender system model she wishes to use.

Training data

Recommender system models must be trained to predict suitable slide templates for a certain piece of input text. We have invited people with adequate design background to generate training data.

We utilize a set of 11 templates which covers basic layout configurations to generate training data. During training, we generate presentation slides using input text with various structural properties without recommender system. That is, every basic template that is structurally legal with respect to the input text is selected for generating slides. After generating presentation slides, people are supposed to decide which templates better match the input text. Suitable template-text matches are stored as positive data for training.
The structure of training dataset for SVM is slightly different than what for CNN. We train CNN as a binary classifier that detects if a certain template-text match is suitable. Therefore, all positive and negative data are used for training. On the other hands, we divide positive training dataset into 11 categories which corresponds to each page of basic templates, so that SVM could be trained as a 11-class classifier. For example, if basic template 2 matches a certain piece of input text, that row of training data will be labeled as 2. Figure 3.5 briefly shows how training data are structured.

**Support vector machine**

Support vector machine (SVM) is a kind of supervised model. If trained with suitable kernels, SVM could well classify high-dimensional non-linear data with high robustness.

After training, the prediction procedure of SVM-based recommender system is shown below. A label indicating which category the input text belongs to is first predicted. Then we iterate all candidate templates, and calculate their feature similarities with the basic template that label corresponds to. The recommender system will therefore return a list of templates that are closest to the basic template.
Our SVM model is trained with the help of scikit-learn toolbox. The model uses radial basis function (RBF) kernel to achieve non-linear classification.

**Convolutional neural network**

Convolutional neural network (CNN) is a type of feed-forward artificial neural network that has been applied into many fields including computer vision, pattern recognition and recommender systems. In addressing this research question, we have trained a one-dimensional CNN model for detecting whether the template-text match is applicable.

The prediction procedure of CNN-based recommender system is shown below. First we extract both the features of input text and slide template, then feed the aligned data into the model. The model will return a boolean value indicating whether the template matches the input text well. By iterating over all candidate templates, we will eventually receive a list of recommended templates.

Our model is trained with the help of keras framework written in python. Input data goes through one fully connected layer, then one convolutional layer (including convolution and max pooling), then two fully connected layers, then one dropout layer, finally one fully connected layer to complete the prediction process. The model is trained with stochastic gradient descent method and Adam optimizer. After training, the model achieves 78% prediction accuracy on cross-validated data.
3.2.3 Compilation

Once set of recommended templates is produced, we compile them with the input text and output a set of recommended slides.

Check if template is structurally legal for compilation

Before actually combining input text with recommended slide templates, we need to ensure templates have compliant structural properties that could well fit the input text. Concretely speaking, we need to ensure that templates:

- Have number of replicable components not less than the number of bullets;
- Have replicable components, all of which number of text boxes not less than the number of lines inside each bullet;
- Have text boxes, all of which maximum text length not exceed the length of corresponding line.

Above is our stipulative definition for structurally legality of templates with respect to input text. This step of additional check is necessary, since sometimes templates recommended by the model are not compliant with input text.

The remaining of templates, after this step, are usually less than templates outputted by the recommender system. If the number of remaining templates is zero (this usually indicates that the input text has unusual structure), we will compress the input text into a blob, and compile it with a template of minimal design (shown on figure 3.8) - a title box, a big text box and basic layout.
Generating slides

We utilize a node.js package, called PptxGenJS, to help finishing our final step of generating presentation slides. This package is not ready-to-use for our framework, therefore some additional functions are implemented onto the original package in order to serve the framework.

3.3 User study

The motivation of conducting user study is to test our framework’s task completion time, user experience, subjective and objective prediction accuracy, and so on. We have 2 sets of user study procedure. In the following parts we will show the idea behind both studies, and design differences between them. By successfully completing the user study, we will prove that our framework is more efficient in generating slides, comparing to handcrafting. Moreover, we will show that slides generated by the framework are more satisfactory than handcrafted and baseline slides.

This study will be conducted on our private laptop (11-inch Macbook Air, 1.4GHz Intel i5 processor, 4GB memory). The software environment is: Python 2.7, node.js 6.10.2, Microsoft PowerPoint for Mac 15.41.
3.3.1 Overview of participants

We have recruited 13 participants (8 male, 5 female, 2 of them have “design background”). 8 participants among them are 22-26 years old while others are younger or older.

3.3.2 Design differences between user study 1 & 2

Motivation and workflow of user study 1

One of the objectives of user study 1 is to test the framework’s task completion time in various measures. Moreover, we will measure user satisfaction towards recommended slides.

In this user study, we feed input text with various structures into SmartPPT. In each execution of the framework, we will measure overall program execution time, recommendation time and presentation slide generating time under different Machine Learning models, i.e. SVM and CNN.

After each set of recommended slides are produced, we will measure user satisfaction by collecting each participant’s feedback at the end of the study. This variable will be quantified in ordinal scale, and will be compared to satisfaction towards baseline design, which will be measured in user study 2.
Motivation and workflow of user study 2

The main purpose of user study 2 is to set up a “control group” which is contrasted to some conditions as in the previous user study.

First, we will let each participant create their own presentation slide design from scratch, given fixed input text. By measuring elapsed time for completing the design and participant’s satisfaction towards it, it will be convincing to show that SmartPPT does save slide producing time, compared to generating slides by hands.

Then we will show some baseline design, i.e. slides generated with PowerPoint templates. By comparing participant’s satisfaction differences between the baseline design and our recommended ones, we will be able to conclude that our framework is more plausible to users.

3.3.3 Explanation on variables related to elapsed time

In both user study procedures, slides are generated through the CAD framework, by PowerPoint templates or by the participants themselves. In the following sections, we will elaborate the approaches to measure variables, including elapsed time and user satisfaction.

Elapsed time measured during runtime of recommender system

After each run of the framework, we will have the following variables:
• **t_exec**: By calculating the time from beginning to end.

• **t_rec**: By calculating the time elapsed in **Step 2** (see section 3.2).

• **t_gen**: By calculating the time elapsed in **Step 3**.

**Elapsed time measured during handcraft**

In user study 2, we let participant create a presentation slide from scratch, and calculate the overall elapsed time.

**Estimated elapsed time of handcrafting recommended slides**

We will also ask participants to estimate how long it might take for them to handcraft those recommended slides, given their knowledge of presentation slides. The estimated time is recorded into the same interval above.

Every time when we conduct user study 1, we create 8 sets of recommendation slides. Therefore, the participant will be asked the same question 8 times (and we get 8 data each time).

Every time when we conduct user study 2, the participant will be asked to estimate how long it might take for them to handcraft Microsoft PowerPoint template-based (baseline) slides, based on their experience with presentation slides.

The answers are just for reference, since participants might overestimate or underestimate their experience with presentation slides. However, comparing estimated time with actual elapsed time of handcrafting would be interesting.

**3.3.4 Explanation on variables related to user satisfaction**

In order to prove that users have better experience in creating slides with SmartPPT, we need to give stipulative definition of “better user experience” before conducting any meaningful study. Here we define “to have better user experience” as: If a user has better experience toward one slide production process than the other, in the user satisfaction survey, she gives higher point to that process. Notice that having obtained higher average point does not necessarily mean users have better user experience to a certain slide production process.
In the following sections, we will also elaborate what “user satisfaction survey” is, how we collect feedbacks from participants and how feedbacks are quantized.

**User satisfaction towards recommended slides**

In the end of user study 1, participant will be asked if she agree with the sentence: “Overall speaking, I am satisfied with the output of SmartPPT.” The answer will be divided into 5 ordinal intervals: Strongly agree (worth 5 points), agree (4 points), so-so (3 points), disagree (2 points) and strongly disagree (1 point). To calculate their average is to divide the sum of ordinal values by the size of data.

**User satisfaction towards handcrafted slides**

In user study 2, after handcrafted slides are completed, participant will be asked if she agree with the sentence: “Overall speaking, I am satisfied with my own design.” The answer will be divided into the same intervals as above, and the way to calculate their average is the same.

**User satisfaction towards baseline slides**

In user study 2, participant will be asked if she agree with the sentence: “Overall speaking, I am satisfied with the slide provided by PowerPoint.” The answer will be divided into the same intervals as above, and the way to calculate their average is the same.

**Miscellaneous Measures**

Besides questions above, we collect participants’ feedback in other perspectives. For example, the last question in user study 1 is “Do you agree with the sentence: I would like to use this framework to automatically generate presentation slides, if there are any chance.” These answers are seen as reference or “side witness” of user satisfaction, and it would be also interesting to analyze them.

**3.3.5 Detailed explanation on feedback**

In this section, we will illustrate how comments are collected during user study.
Feedback collection process

Every time we conduct user study 1, we show participant 8 sets of recommended slides. Each time after showing one set of slides, participant should answer the question: “How many slides among them do you think might be useful in your presentation tasks?”

If the previous answer is not “None”, we let the participant pick up one slide she considers “useful in potential presentation tasks” and describe its advantages that pleases her.

If the previous answer is not “All of them”, we let the participant pick up one slide she considers “not to be selected in potential presentation tasks” and describe its disadvantages.

As a result, we will receive at most 16 comments after completing one user study.

Feedback selection process

Not all comments are valuable for us. We remove comments like “just good” and “bad design”, since they lack detailed description of how “good/bad” a slide is. This is normal, since some participants might be less critical to presentation slide design or might say shorter sentences compared to other participants.
Chapter 4

Results and Discussion

4.1 Overview

We divide the results into quantitative and qualitative parts for clear illustration. The quantitative results are categorized into task completion time and user satisfaction. The aim of the qualitative analysis is to obtain a thorough understanding of user study feedback. Besides of quantitative analysis of user experiences, we would like to see more detailed, more expressive comments from participants. We will first provide a complete explanation on how we obtain comments from participants, and how those comments are arranged on the spreadsheet. Then we will analyze user comments and reveal the most interesting findings.

4.2 Execution time analysis

In section 3.3.4, we have explained the measurement procedure, i.e. how we obtain the data and what they mean. In this section, we will then proceed to conduct the analysis.

Raw results can be seen in the spreadsheet. Part of the spreadsheet is screenshot and shown below.

- Average $t_{\text{exec}}$ (program execution time) is 7.26 seconds for SVM, and 14.01 seconds for CNN.
  - Both times are on second level.
  - Average measured time of handcrafting those recommended slides is $10 \pm 3$ minutes, whereas when done through the rec-
ommendation system, it takes under 10 seconds. In human factors, this is a meaningful difference in work practices.

• Average $t_{\text{exec}}$ for handcrafting slides is also on minute level. Three participants finished creating their own slide within 15 minutes, while others (3 participants) finished within 30 minutes.
  
  – Based on our observation, participants spend most of the production time adjusting shape positions and searching for color combinations.

• Average $t_{\text{rec}}$ for SVM (0.02 second) is much shorter than CNN (6.54 seconds). This might be due to the large computation cost of the neural network model.

Overall speaking, our recommender system is time-efficient in generating slides.

4.3 User satisfaction analysis

In section 3.3.5, we have explained the measurement procedure, i.e. how we obtain the data and what it means. In this section, we will
then proceed to conduct the analysis.

Raw results can be seen in the spreadsheet.

Seven participants answered the question described in 3.3.5.1 (SmartPPT), while 6 participants answered the question described in 3.3.5.2 (handcraft) and 3.3.5.3 (baseline).

- The average point of user satisfaction towards recommended slides is 3.85, whereas the average of handcrafted slides is 3.33 and baseline slides is 2.17. The p-value of Student’s t-test on average points of recommended slides and baseline slides is 0.0024, indicating that we could reject the null hypothesis that participants’ satisfaction towards recommended slides and baseline slides are the same.
  - Most participants (85%) gave 4 points to recommended slides.
  - The majority of participants (50%) gave 4 points to their handcrafted slides. This is not surprising, since people usually tend to overestimate the design quality of themselves.
  - The majority of participants (50%) gave 3 points to baseline slides.

- Every time we conduct user study 1, we show participant 8 sets of recommended slides. Each time after showing one set of slides, participant should answer the question: “How many slides among them do you think might be useful in your presentation tasks?”

Figure 4.3: User satisfaction score of slide sets. A=baseline slides, B=handcrafted slides, and C=recommended slides.
– The answer is categorized into four intervals: None, one or two slides, three or four slides and all of them.
– The question is asked 8 times per user study. Since we have recruited 7 participants in the study, we have finally collected 56 answers.
– Majority (78%) of the answers are “1-2 pages”.
– Very few answers are “3-4 pages”. Those answers actually come from two participants who are seemingly less critical in judging designs.

As a result, we could conclude that the slides provided by our framework is satisfactory, and “have better user experience” than the baseline condition.

4.4 Detailed feedback analysis

In this section, we will analyze feedbacks collected through process described in section 3.3.6 and reveal some interesting findings.

• Recommended slides could not fit all participants’ tastes. For example, some participants prefer “simple and neat” slides, which is disgusted by others for “having too few components”.

• Features that participants appreciate most:
  – Vivid color combination
  – Good layout design
  – Simple / neat / clear structure

• Features that participants disgust most:
  – Too bizarre layout.
  – Color combination. This is quite subjective, since various participants might prefer different color combinations.
  – Image / textbox discordance.
  – Some slides have too big / too small fonts.
Some flaws could be solved in the future with the help of layout optimization. For example, if a hexagon-like template is filled with 5 or less bullets, we could apply layout optimization to make the slide look pentagon-like, quadrilateral, etc.
Chapter 5

Conclusion

Both user study procedures have proved that SmartPPT performs well in providing templates for academic and formal input text, compared to handcrafted slides and slides with PowerPoint templates. Comparing to those slide production processes, participants are more satisfied with presentation slides generated using our framework. As a result, our research question has been successfully solved.

CNN-based recommender system model is significantly slower in making predictions than SVM-based model. This leads to the consequence, that the overall elapsed time (t_exec) of framework with CNN-based model is around two times of elapsed time with SVM-based model, which is a runtime flaw. The flaw could be explained by the large computation cost due to the multi-layer neural network model.

Another argument on performances between two recommender system models may arise on the relatively small size of training dataset (around 2K). Some classical models like SVM and Random Forests could have very pleasing generalizability when training data is small. However, models like neural networks may require larger training data size, and small dataset may cause its generalizability to decrease.

This thesis could compare performances among more classical Machine Learning models, for example, Random Forests. However, to evaluate performance of Machine Learning model based CAD framework is more complicated than evaluating performance of a Machine Learning model itself. There are various objective criterias to measure a model, but user experience and user satisfaction should be prioritized when we evaluate CAD frameworks. In a word, it is user who indeed uses it. On the other hand, this thesis could compare perfor-
mannances of SmartPPT with automatic generated slides using LaTeX.
Bibliography


1145/3018661.3018665. URL: http://doi.acm.org/10.1145/3018661.3018665.
### Appendix A

#### List of input text features

<table>
<thead>
<tr>
<th>#</th>
<th>Feature</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>txt_len</code></td>
<td>Numeric</td>
<td>The total text length.</td>
</tr>
<tr>
<td>2</td>
<td><code>num_blt</code></td>
<td>Numeric</td>
<td>Number of 1st level elements (“bullets” for short).</td>
</tr>
<tr>
<td>3</td>
<td><code>max_blt_txt_len</code></td>
<td>Numeric</td>
<td>Text length of the largest bullet.</td>
</tr>
<tr>
<td>4</td>
<td><code>min_blt_txt_len</code></td>
<td>Numeric</td>
<td>Text length of the smallest bullet.</td>
</tr>
<tr>
<td>5</td>
<td><code>avg_blt_txt_len</code></td>
<td>Numeric</td>
<td>Average text length of bullets.</td>
</tr>
<tr>
<td>6</td>
<td><code>max_blt_txt_len_r</code></td>
<td>Numeric</td>
<td>Text length of the largest bullet, divided by the total text length.</td>
</tr>
<tr>
<td>7</td>
<td><code>min_blt_txt_len_r</code></td>
<td>Numeric</td>
<td>Text length of the smallest bullet, divided by the total text length.</td>
</tr>
<tr>
<td>8</td>
<td><code>avg_blt_txt_len_r</code></td>
<td>Numeric</td>
<td>Average text length of bullets, divided by the total text length.</td>
</tr>
<tr>
<td>9</td>
<td><code>is_blt_pts_same_len</code></td>
<td>Boolean</td>
<td>True if all bullets have same text length.</td>
</tr>
<tr>
<td>10</td>
<td><code>is_dangled</code></td>
<td>Boolean</td>
<td>True if the title (zero-level element) has zero text length.</td>
</tr>
<tr>
<td>11</td>
<td><code>exist_empty_line</code></td>
<td>Boolean</td>
<td>True if there exists empty line with zero text length.</td>
</tr>
<tr>
<td>12</td>
<td><code>is_title_empty</code></td>
<td>Boolean</td>
<td>True if the title is empty (zero-level element) has zero text length.</td>
</tr>
<tr>
<td>13</td>
<td><code>max_blt_lvl</code></td>
<td>Numeric</td>
<td>The level of bullet that has maximum level.</td>
</tr>
<tr>
<td>14</td>
<td><code>min_blt_lvl</code></td>
<td>Numeric</td>
<td>The level of bullet that has minimum level.</td>
</tr>
<tr>
<td>15</td>
<td><code>max_blt_line</code></td>
<td>Numeric</td>
<td>The number of lines of bullet that has maximum number of lines.</td>
</tr>
<tr>
<td>16</td>
<td><code>min_blt_line</code></td>
<td>Numeric</td>
<td>The number of lines of bullet that has minimum number of lines.</td>
</tr>
<tr>
<td>17</td>
<td><code>avg_blt_line</code></td>
<td>Numeric</td>
<td>Average number of lines of bullets.</td>
</tr>
<tr>
<td>18</td>
<td><code>lv1_txt_len_max</code></td>
<td>Numeric</td>
<td>Maximum text length of level 1 elements.</td>
</tr>
<tr>
<td>19</td>
<td><code>lv1_txt_len_min</code></td>
<td>Numeric</td>
<td>Minimum text length of level 1 elements.</td>
</tr>
<tr>
<td>20</td>
<td><code>lv1_txt_len_avg</code></td>
<td>Numeric</td>
<td>Average text length of level 1 elements.</td>
</tr>
<tr>
<td>21</td>
<td><code>lv1_txt_len_std</code></td>
<td>Numeric</td>
<td>Standard deviation of text length of level 1 elements.</td>
</tr>
<tr>
<td>22</td>
<td><code>lv2_txt_len_max</code></td>
<td>Numeric</td>
<td>Maximum text length of level 2 elements.</td>
</tr>
<tr>
<td>23</td>
<td><code>lv2_txt_len_min</code></td>
<td>Numeric</td>
<td>Minimum text length of level 2 elements.</td>
</tr>
<tr>
<td>24</td>
<td><code>lv2_txt_len_avg</code></td>
<td>Numeric</td>
<td>Average text length of level 2 elements.</td>
</tr>
<tr>
<td>25</td>
<td><code>lv2_txt_len_std</code></td>
<td>Numeric</td>
<td>Standard deviation of text length of level 2 elements.</td>
</tr>
<tr>
<td>26</td>
<td><code>lv3_txt_len_max</code></td>
<td>Numeric</td>
<td>Maximum text length of level 3 elements.</td>
</tr>
<tr>
<td>27</td>
<td><code>lv3_txt_len_min</code></td>
<td>Numeric</td>
<td>Minimum text length of level 3 elements.</td>
</tr>
<tr>
<td>28</td>
<td><code>lv3_txt_len_avg</code></td>
<td>Numeric</td>
<td>Average text length of level 3 elements.</td>
</tr>
<tr>
<td>29</td>
<td><code>lv3_txt_len_std</code></td>
<td>Numeric</td>
<td>Standard deviation of text length of level 3 elements.</td>
</tr>
<tr>
<td>30</td>
<td><code>lv4_txt_len_max</code></td>
<td>Numeric</td>
<td>Maximum text length of level 4 elements.</td>
</tr>
<tr>
<td>31</td>
<td><code>lv4_txt_len_min</code></td>
<td>Numeric</td>
<td>Minimum text length of level 4 elements.</td>
</tr>
<tr>
<td>32</td>
<td><code>lv4_txt_len_avg</code></td>
<td>Numeric</td>
<td>Average text length of level 4 elements.</td>
</tr>
<tr>
<td>33</td>
<td><code>lv4_txt_len_std</code></td>
<td>Numeric</td>
<td>Standard deviation of text length of level 4 elements.</td>
</tr>
<tr>
<td>34</td>
<td><code>lv1_line_std</code></td>
<td>Numeric</td>
<td>Standard deviation of number of lines of level 1 elements.</td>
</tr>
<tr>
<td>35</td>
<td><code>lv2_line_std</code></td>
<td>Numeric</td>
<td>Standard deviation of number of lines of level 2 elements.</td>
</tr>
<tr>
<td>36</td>
<td><code>lv3_line_std</code></td>
<td>Numeric</td>
<td>Standard deviation of number of lines of level 3 elements.</td>
</tr>
<tr>
<td>37</td>
<td><code>lv4_line_std</code></td>
<td>Numeric</td>
<td>Standard deviation of number of lines of level 4 elements.</td>
</tr>
<tr>
<td>38</td>
<td><code>lv1_txt_len_max_r</code></td>
<td>Numeric</td>
<td>Maximum text length of level 1 elements, divided by the total text length.</td>
</tr>
<tr>
<td>39</td>
<td><code>lv1_txt_len_min_r</code></td>
<td>Numeric</td>
<td>Minimum text length of level 1 elements, divided by the total text length.</td>
</tr>
<tr>
<td>40</td>
<td><code>lv1_txt_len_avg_r</code></td>
<td>Numeric</td>
<td>Average text length of level 1 elements, divided by the total text length.</td>
</tr>
<tr>
<td>41</td>
<td><code>lv2_txt_len_max_r</code></td>
<td>Numeric</td>
<td>Maximum text length of level 2 elements, divided by the total text length.</td>
</tr>
<tr>
<td>42</td>
<td><code>lv2_txt_len_min_r</code></td>
<td>Numeric</td>
<td>Minimum text length of level 2 elements, divided by the total text length.</td>
</tr>
<tr>
<td>43</td>
<td><code>lv2_txt_len_avg_r</code></td>
<td>Numeric</td>
<td>Average text length of level 2 elements, divided by the total text length.</td>
</tr>
<tr>
<td>44</td>
<td><code>lv3_txt_len_max_r</code></td>
<td>Numeric</td>
<td>Maximum text length of level 3 elements, divided by the total text length.</td>
</tr>
<tr>
<td>45</td>
<td><code>lv3_txt_len_min_r</code></td>
<td>Numeric</td>
<td>Minimum text length of level 3 elements, divided by the total text length.</td>
</tr>
<tr>
<td>46</td>
<td><code>lv3_txt_len_avg_r</code></td>
<td>Numeric</td>
<td>Average text length of level 3 elements, divided by the total text length.</td>
</tr>
<tr>
<td>47</td>
<td><code>lv4_txt_len_max_r</code></td>
<td>Numeric</td>
<td>Maximum text length of level 4 elements, divided by the total text length.</td>
</tr>
<tr>
<td>48</td>
<td><code>lv4_txt_len_min_r</code></td>
<td>Numeric</td>
<td>Minimum text length of level 4 elements, divided by the total text length.</td>
</tr>
<tr>
<td>49</td>
<td><code>lv4_txt_len_avg_r</code></td>
<td>Numeric</td>
<td>Average text length of level 4 elements, divided by the total text length.</td>
</tr>
<tr>
<td>50</td>
<td><code>lv1_txt_len_sum</code></td>
<td>Numeric</td>
<td>Sum of text length of level 1 elements.</td>
</tr>
<tr>
<td>51</td>
<td><code>lv2_txt_len_sum</code></td>
<td>Numeric</td>
<td>Sum of text length of level 2 elements.</td>
</tr>
<tr>
<td>52</td>
<td><code>lv3_txt_len_sum</code></td>
<td>Numeric</td>
<td>Sum of text length of level 3 elements.</td>
</tr>
<tr>
<td>53</td>
<td><code>lv4_txt_len_sum</code></td>
<td>Numeric</td>
<td>Sum of text length of level 4 elements.</td>
</tr>
<tr>
<td>54</td>
<td><code>lv1_txt_len_sum_r</code></td>
<td>Numeric</td>
<td>Sum of text length of level 1 elements, divided by the total text length.</td>
</tr>
<tr>
<td>55</td>
<td><code>lv2_txt_len_sum_r</code></td>
<td>Numeric</td>
<td>Sum of text length of level 2 elements, divided by the total text length.</td>
</tr>
<tr>
<td>56</td>
<td><code>lv3_txt_len_sum_r</code></td>
<td>Numeric</td>
<td>Sum of text length of level 3 elements, divided by the total text length.</td>
</tr>
<tr>
<td>57</td>
<td><code>lv4_txt_len_sum_r</code></td>
<td>Numeric</td>
<td>Sum of text length of level 4 elements, divided by the total text length.</td>
</tr>
</tbody>
</table>
# Appendix B

## List of presentation template features

<table>
<thead>
<tr>
<th>#</th>
<th>Feature</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>num_group</td>
<td>Numeric</td>
<td>Number of group elements (&quot;groups&quot; for short).</td>
</tr>
<tr>
<td>2</td>
<td>num_elt</td>
<td>Numeric</td>
<td>Height of group, divided by height of template.</td>
</tr>
<tr>
<td>3</td>
<td>wd_r</td>
<td>Numeric</td>
<td>Width of group, divided by width of template.</td>
</tr>
<tr>
<td>4</td>
<td>num_eltnt</td>
<td>Numeric</td>
<td>Number of elements inside a group.</td>
</tr>
<tr>
<td>5</td>
<td>num_sp</td>
<td>Numeric</td>
<td>Number of shape elements inside a group.</td>
</tr>
<tr>
<td>6</td>
<td>num_txt</td>
<td>Numeric</td>
<td>Number of text elements inside a group.</td>
</tr>
<tr>
<td>7</td>
<td>sp_area_r</td>
<td>Numeric</td>
<td>Total area of shape elements, divided by area of a group.</td>
</tr>
<tr>
<td>8</td>
<td>txt_area_r</td>
<td>Numeric</td>
<td>Total area of text elements, divided by area of a group.</td>
</tr>
<tr>
<td>9</td>
<td>area_sp_lf_txt</td>
<td>Numeric</td>
<td>Maximum width of text elements, divided by width of a group.</td>
</tr>
<tr>
<td>10</td>
<td>min_txt_wd_r</td>
<td>Numeric</td>
<td>Minimum width of text elements, divided by width of a group.</td>
</tr>
<tr>
<td>11</td>
<td>max_txt_wd_r</td>
<td>Numeric</td>
<td>Maximum width of shape elements, divided by width of a group.</td>
</tr>
<tr>
<td>12</td>
<td>min_txt_ht_r</td>
<td>Numeric</td>
<td>Minimum height of text elements, divided by height of a group.</td>
</tr>
<tr>
<td>13</td>
<td>max_txt_ht_r</td>
<td>Numeric</td>
<td>Maximum height of shape elements, divided by height of a group.</td>
</tr>
<tr>
<td>14</td>
<td>min_sp_wd_r</td>
<td>Numeric</td>
<td>Minimum width of shape elements, divided by width of a group.</td>
</tr>
<tr>
<td>15</td>
<td>max_sp_wd_r</td>
<td>Numeric</td>
<td>Maximum width of shape elements, divided by width of a group.</td>
</tr>
<tr>
<td>16</td>
<td>min_sp_ht_r</td>
<td>Numeric</td>
<td>Minimum height of shape elements, divided by height of a group.</td>
</tr>
<tr>
<td>17</td>
<td>max_sp_ht_r</td>
<td>Numeric</td>
<td>Maximum height of shape elements, divided by height of a group.</td>
</tr>
<tr>
<td>18</td>
<td>is_height_align</td>
<td>Boolean</td>
<td>If some groups are horizontally aligned.</td>
</tr>
<tr>
<td>19</td>
<td>is_height_all_align</td>
<td>Boolean</td>
<td>If all groups are horizontally aligned.</td>
</tr>
<tr>
<td>20</td>
<td>is_width_align</td>
<td>Boolean</td>
<td>If some groups are vertically aligned.</td>
</tr>
<tr>
<td>21</td>
<td>is_width_all_align</td>
<td>Boolean</td>
<td>If all groups are vertically aligned.</td>
</tr>
<tr>
<td>22</td>
<td>max_txt_len</td>
<td>Numeric</td>
<td>Maximum text length inside a group.</td>
</tr>
<tr>
<td>23</td>
<td>min_txt_len</td>
<td>Numeric</td>
<td>Minimum text length inside a group.</td>
</tr>
<tr>
<td>24</td>
<td>max_txt_sz</td>
<td>Numeric</td>
<td>Maximum text size inside a group.</td>
</tr>
<tr>
<td>25</td>
<td>min_txt_sz</td>
<td>Numeric</td>
<td>Minimum text size inside a group.</td>
</tr>
<tr>
<td>26</td>
<td>is_txt_sm_sz</td>
<td>Boolean</td>
<td>If text elements have the same font size.</td>
</tr>
<tr>
<td>27</td>
<td>is_sp_sm_sz</td>
<td>Boolean</td>
<td>If shape elements have the same font size.</td>
</tr>
<tr>
<td>28</td>
<td>is_sp_sm_wd</td>
<td>Boolean</td>
<td>If shape elements have the same width.</td>
</tr>
<tr>
<td>29</td>
<td>left_grp_r</td>
<td>Numeric</td>
<td>Number of groups that reside on left of the page, divided by the number of groups.</td>
</tr>
<tr>
<td>30</td>
<td>right_grp_r</td>
<td>Numeric</td>
<td>Number of groups that reside on right of the page, divided by the number of groups.</td>
</tr>
<tr>
<td>31</td>
<td>middle_GRP_r</td>
<td>Numeric</td>
<td>Number of groups that reside on middle (horizontally) of the page, divided by the number of groups.</td>
</tr>
<tr>
<td>32</td>
<td>left_grp_r</td>
<td>Numeric</td>
<td>Number of groups that reside on left of the page, divided by the number of groups.</td>
</tr>
<tr>
<td>33</td>
<td>right_grp_r</td>
<td>Numeric</td>
<td>Number of groups that reside on right of the page, divided by the number of groups.</td>
</tr>
<tr>
<td>34</td>
<td>middle_lr_grp_r</td>
<td>Numeric</td>
<td>Number of groups that reside on middle (horizontally) of the page, divided by the number of groups.</td>
</tr>
<tr>
<td>35</td>
<td>up_grp_r</td>
<td>Numeric</td>
<td>Number of groups that reside on top of the page, divided by the number of groups.</td>
</tr>
<tr>
<td>36</td>
<td>down_grp_r</td>
<td>Numeric</td>
<td>Number of groups that reside on bottom of the page, divided by the number of groups.</td>
</tr>
<tr>
<td>37</td>
<td>middle_ud_grp_r</td>
<td>Numeric</td>
<td>Number of groups that reside on middle (vertically) of the page, divided by the number of groups.</td>
</tr>
<tr>
<td>38</td>
<td>ud_lvl</td>
<td>Numeric</td>
<td>Number different y value of groups.</td>
</tr>
<tr>
<td>39</td>
<td>lr_lvl</td>
<td>Numeric</td>
<td>Number different x value of groups.</td>
</tr>
<tr>
<td>40</td>
<td>is_uni_sp</td>
<td>Boolean</td>
<td>If there exists only one shape element inside a group.</td>
</tr>
<tr>
<td>41</td>
<td>is_uni_txt</td>
<td>Boolean</td>
<td>If there exists only one text element inside a group.</td>
</tr>
<tr>
<td>42</td>
<td>max_txt_mg_l</td>
<td>Numeric</td>
<td>Maximum left margin of text elements inside a group.</td>
</tr>
<tr>
<td>43</td>
<td>max_txt_mg_u</td>
<td>Numeric</td>
<td>Maximum top margin of text elements inside a group.</td>
</tr>
<tr>
<td>44</td>
<td>max_txt_mg_d</td>
<td>Numeric</td>
<td>Maximum bottom margin of text elements inside a group.</td>
</tr>
<tr>
<td>45</td>
<td>max_txt_mg_chars</td>
<td>Numeric</td>
<td>Maximum character spacing of text elements inside a group, divided by the width of the group.</td>
</tr>
<tr>
<td>46</td>
<td>max_txt_para_sp</td>
<td>Numeric</td>
<td>Maximum paragraph spacing of text elements inside a group, divided by the height of the group.</td>
</tr>
<tr>
<td>47</td>
<td>min_txt_mg_l</td>
<td>Numeric</td>
<td>Minimum left margin of text elements inside a group.</td>
</tr>
<tr>
<td>48</td>
<td>min_txt_mg_u</td>
<td>Numeric</td>
<td>Minimum top margin of text elements inside a group.</td>
</tr>
<tr>
<td>49</td>
<td>min_txt_mg_d</td>
<td>Numeric</td>
<td>Minimum bottom margin of text elements inside a group.</td>
</tr>
<tr>
<td>50</td>
<td>min_txt_mg_chars</td>
<td>Numeric</td>
<td>Minimum character spacing of text elements inside a group, divided by the width of the group.</td>
</tr>
<tr>
<td>51</td>
<td>min_txt_para_sp</td>
<td>Numeric</td>
<td>Minimum paragraph spacing of text elements inside a group, divided by the height of the group.</td>
</tr>
</tbody>
</table>
Appendix C

User study design

C.1 User study 1 design

C.1.1 Step 1

Present the first 5 pages of User Study slide to warm up the participant. We will tell the participant about the background and workflow of the project, and goal of this study. If the participant confirms that she is well informed of the study processes, we proceed to the next step.

C.1.2 Step 2

In this step, we will ask some basic information about the participant herself.

Step 2.1

Ask the age and sex of the participant.

Step 2.2

Ask if the participant have “design background”. Here we provide stipulative definition of “to have design background” by “to have earned a design major degree, or be currently in a design degree program”.
C.1.3 Step 3

In this step, we will generate 8 sets of recommendation slides for input outline with varying properties:

- Much (100+ words) / less (less than 50 words) content,
- Many (6) / few (1 2) numbers of “bullet”,
- By SVM / CNN.

The participant may change the content of input outline based on her own interest.

Step 3.1

Generate slides with less content and few numbers of “bullet” with SVM. The result slides will be shown to the participant.

Step 3.1.1 During the execution of the program, we measure the overall execution time by calculating the time from the beginning of the program to the end.

Step 3.1.2 Ask the participant the following questions: Are there any recommended slide(s), which design are likely to be adapted in your potential presentation tasks?

For those slide(s) that are selected in the first question:

- Pick up a slide and describe why you may choose it in your potential presentation tasks?
- Based on your familiarity with presentation slides, how long will it take to design such a slide?

For those slide(s) that are not selected in the first question:

- Pick up a slide and describe why you may not choose it in your potential presentation tasks?
- Based on your familiarity with presentation slides, could you avoid designing such a slide?
Step 3.2
Generate slides with less content and few numbers of “bullet” with CNN. The result slides will be shown to the participant.

Step 3.2.1 During the execution of the program, we measure the overall execution time by calculating the time from the beginning of the program to the end.

Step 3.2.2 Ask the participant the following questions: Are there any recommended slide(s), which design are likely to be adapted in your potential presentation tasks?
   For those slide(s) that are selected in the first question:
   - Pick up a slide and describe why you may choose it in your potential presentation tasks?
   - Based on your familiarity with presentation slides, how long will it take to design such a slide?

For those slide(s) that are not selected in the first question:
   - Pick up a slide and describe why you may not choose it in your potential presentation tasks?
   - Based on your familiarity with presentation slides, could you avoid designing such a slide?

Step 3.3
Generate slides with less content and many numbers of “bullet” with SVM. The result slides will be shown to the participant.

Step 3.3.1 During the execution of the program, we measure the overall execution time by calculating the time from the beginning of the program to the end.

Step 3.3.2 Ask the participant the following questions: Are there any recommended slide(s), which design are likely to be adapted in your potential presentation tasks?
   For those slide(s) that are selected in the first question:
- Pick up a slide and describe why you may choose it in your potential presentation tasks?

- Based on your familiarity with presentation slides, how long will it take to design such a slide?

For those slide(s) that are not selected in the first question:

- Pick up a slide and describe why you may not choose it in your potential presentation tasks?

- Based on your familiarity with presentation slides, could you avoid designing such a slide?

**Step 3.4**

Generate slides with less content and many numbers of “bullet” with CNN. The result slides will be shown to the participant.

**Step 3.4.1** During the execution of the program, we measure the overall execution time by calculating the time from the beginning of the program to the end.

**Step 3.4.2** Ask the participant the following questions: Are there any recommended slide(s), which design are likely to be adapted in your potential presentation tasks?

For those slide(s) that are selected in the first question:

- Pick up a slide and describe why you may choose it in your potential presentation tasks?

- Based on your familiarity with presentation slides, how long will it take to design such a slide?

For those slide(s) that are not selected in the first question:

- Pick up a slide and describe why you may not choose it in your potential presentation tasks?

- Based on your familiarity with presentation slides, could you avoid designing such a slide?
Step 3.5
Generate slides with rich content and few numbers of “bullet” with SVM. The result slides will be shown to the participant.

Step 3.5.1 During the execution of the program, we measure the overall execution time by calculating the time from the beginning of the program to the end.

Step 3.5.2 Ask the participant the following questions: Are there any recommended slide(s), which design are likely to be adapted in your potential presentation tasks?
   For those slide(s) that are selected in the first question:
   - Pick up a slide and describe why you may choose it in your potential presentation tasks?
   - Based on your familiarity with presentation slides, how long will it take to design such a slide?

For those slide(s) that are not selected in the first question:
   - Pick up a slide and describe why you may not choose it in your potential presentation tasks?
   - Based on your familiarity with presentation slides, could you avoid designing such a slide?

Step 3.6
Generate slides with rich content and few numbers of “bullet” with CNN. The result slides will be shown to the participant.

Step 3.6.1 During the execution of the program, we measure the overall execution time by calculating the time from the beginning of the program to the end.

Step 3.6.2 Ask the participant the following questions: Are there any recommended slide(s), which design are likely to be adapted in your potential presentation tasks?
   For those slide(s) that are selected in the first question:
- Pick up a slide and describe why you may choose it in your potential presentation tasks?

- Based on your familiarity with presentation slides, how long will it take to design such a slide?

For those slide(s) that are not selected in the first question:

- Pick up a slide and describe why you may not choose it in your potential presentation tasks?

- Based on your familiarity with presentation slides, could you avoid designing such a slide?

**Step 3.7**

Generate slides with rich content and many numbers of “bullet” with SVM. The result slides will be shown to the participant.

**Step 3.7.1** During the execution of the program, we measure the overall execution time by calculating the time from the beginning of the program to the end.

**Step 3.7.2** Ask the participant the following questions: *Are there any recommended slide(s), which design are likely to be adapted in your potential presentation tasks?*

For those slide(s) that are selected in the first question:

- Pick up a slide and describe why you may choose it in your potential presentation tasks?

- Based on your familiarity with presentation slides, how long will it take to design such a slide?

For those slide(s) that are not selected in the first question:

- Pick up a slide and describe why you may not choose it in your potential presentation tasks?

- Based on your familiarity with presentation slides, could you avoid designing such a slide?
Step 3.8
Generate slides with rich content and many numbers of “bullet” with CNN. The result slides will be shown to the participant.

Step 3.8.1 During the execution of the program, we measure the overall execution time by calculating the time from the beginning of the program to the end.

Step 3.8.2 Ask the participant the following questions: Are there any recommended slide(s), which design are likely to be adapted in your potential presentation tasks?

For those slide(s) that are selected in the first question:

- Pick up a slide and describe why you may choose it in your potential presentation tasks?
- Based on your familiarity with presentation slides, how long will it take to design such a slide?

For those slide(s) that are not selected in the first question:

- Pick up a slide and describe why you may not choose it in your potential presentation tasks?
- Based on your familiarity with presentation slides, could you avoid designing such a slide?

C.1.4 Step 4

Ask the participant the following questions:

- Overall speaking, are you satisfied with the output of the recommender system?
- How would you like to use this framework to automatically generate presentation slides, if there are any chance?
C.2 User study 2 design

C.2.1 Step 1

Present the first 5 pages of User Study slide to warm up the participant. We will tell the participant about the background and workflow of the project, and goal of this study.

If the participant confirms that she is well informed of the study processes, we proceed to the next step.

C.2.2 Step 2

In this step, we will ask some basic information about the participant herself.

Step 2.1

Ask the age and sex of the participant.

Step 2.2

Ask if the participant has “design background”. Here we provide stipulative definition of “to have design background” by “to have earned a design major degree, or be currently in a design degree program”.

C.2.3 Step 3

Step 3.1

Ask the participant to generate one page of presentation slide “from scratch”, i.e. from blank slide in Microsoft PowerPoint.

Step 3.2

After the participant finishes creating the slide, calculate the elapsed time.

Step 3.3

Ask the participant the following questions:

• How are you satisfied with your design?
• Which aspect(s) do you think this slide could be improved?

C.2.4 Step 4

Step 4.1
Show the participant the presentation slide generated from Microsoft PowerPoint template.

Step 4.2
Ask the participant the following questions:

• How are you satisfied with this design, i.e. how would you be likely to choose this slide design in your potential presentation tasks?

• Based on your familiarity with presentation slides, how long will it take to design such a slide?

C.2.5 Step 5

In this step, we will generate one set of recommendation slides for input outline with the following properties: Less (less than 50 words) content, and Many (6) numbers of “bullet”.

The participant may change the content of input outline based on her own interest.

Step 5.1
Generate slides with less content and many numbers of “bullet” with SVM. The result slides will be shown to the participant.

Step 5.2
During the execution of the program, we measure the overall execution time by calculating the time from the beginning of the program to the end.
APPENDIX C. USER STUDY DESIGN

Step 5.3
Ask the participant the following questions: Are there any recommended slide(s), which design are likely to be adapted in your potential presentation tasks?

For those slide(s) that are selected in the first question:

- Pick up a slide and describe why you may choose it in your potential presentation tasks?
- Based on your familiarity with presentation slides, how long will it take to design such a slide?

For those slide(s) that are not selected in the first question:

- Pick up a slide and describe why you may not choose it in your potential presentation tasks?
- Based on your familiarity with presentation slides, could you avoid designing such a slide?

C.2.6 Step 6
Ask the participant the following questions:

- Overall speaking, are you satisfied with the output of the recommender system?
- How would you like to use this framework to automatically generate presentation slides, if there are any chance?