AI-DRIVEN AUTOMATED SLIDE GENERATION AND TRANSCRIPT-BASED FEEDBACK FOR ENHANCING TEACHING EFFICIENCY AND QUALITY

Claire Shen [1], Ivan Revilla [2]

[1] Saratoga High School, 20300 Herriman Ave, Saratoga, CA 95070
[2] California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

This project aims to automate the creation of lesson slides using AI, reducing the time and effort tutors spend preparing presentations. By leveraging OpenAI's GPT-4-turbo model, the system generates slide content based on a given topic, desired length, and bullet points. Additionally, it analyzes lesson transcripts to assess student engagement and slide effectiveness, offering feedback and actionable tips for tutors to improve their teaching. The system generates slides by parsing the AI's text output and formatting it into a .pptx file using python-pptx. Challenges such as vague content and simplistic design are addressed by refining input prompts and using tools like Designen (Weng et al., 2024) to improve the slides' appearance. The system has proven effective in enhancing instructional quality and student engagement by automating slide creation and providing real-time feedback, allowing tutors to save time and focus more on refining their teaching methods.

KEYWORDS

Artificial Intelligence, Academics, Transcript, Feedback

1. INTRODUCTION

Effective teaching requires significant time and effort, particularly in the preparation of instructional materials such as slides. Research has shown that tutors and instructors often spend a substantial portion of their time creating lesson content, which can detract from their ability to focus on actual teaching and student engagement (Author et al., Year). Automating slide generation presents an opportunity to streamline this process, allowing educators to allocate more time to pedagogical strategies and student interaction.

This project explores the use of artificial intelligence (AI) to automatically generate slides based on instructor-provided inputs, such as topic, number of slides, and bullet points per slide [5]. By leveraging OpenAI's GPT-4-turbo model, we aim to reduce the manual effort required in slide preparation while maintaining content relevance and quality. Additionally, we analyze lesson transcripts to evaluate student comprehension and the effectiveness of the generated slides, providing actionable feedback to tutors for continuous improvement.

Slideshow presentations are a key component of educational lessons, with their usage evident in the display of key points, summarized ideas, and visual aids. Widely used by tutors and educators as instructional aids, they serve as essential tools in providing effective teaching. However, "The

David C. Wyld et al. (Eds): CRYPIS, CBIoT, CAIML, NLCA, NC, WiMo, ICAIT, ICDIPV, ITCSE – 2025
pp. 215-227, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151717

crafting of a compelling presentation requires significant time and effort, from gathering key insights to designing slides that convey information clearly and concisely" (Tushar Aggarwal and Aarohi Bhand, 2025). The amount of time teachers and tutors spend preparing slides materials detracts from their ability to focus on effective teaching methods and student engagement. Therefore, the automation of the slide creation process offers an opportunity for educators to allocate more time and energy to pedagogical strategies and student interaction.

This project explores the use of artificial intelligence (AI) to automatically generate PowerPoint slides for instructors to use in lessons [6]. The user provides requirements such as the topic, the number of slides, and the number of bullet points for each slide, which are then input into OpenAI's GPT-4-turbo model. Through the use of AI, we aim to reduce the manual effort required in slide preparation while maintaining the relevance and quality of content. We achieve a reduction in tutor workload without sacrificing the opportunity to provide a quality learning environment for students. Additionally, we continue to utilize the GPT-4-turbo model in analyzing the lesson transcript to evaluate student engagement and comprehension along with effectiveness of the slides. Tutors receive actionable feedback for improving their teaching methods while the slides are automatically adjusted to aid the tutor in better quality instruction.

PPTAgent (Zheng et al., 2025) aims to reduce the high amount of human effort required to create quality presentations. Their edit-based approach to automating slide generation leverages reference slides in addition to an input document, which are analyzed and combined using a generated outline with specific reference slides and relevant content from the input document. The resulting target slideshow demonstrates good performance in the context of content, design, and coherence, but also exhibits limitations such as occasional design flaws and potential model failure.

DOC2PPT (Fu et al., 2022) presents a sequence-to-sequence approach with multiple components executed in an end-to-end manner to tackle the goal of enhancing human productivity through generating slide presentations from given documents. Their method outperforms other models but still exhibits flaws in slide background design and dependency on user input of relevant documents.

The utilization of Microsoft Excel for lesson transcript analysis by Tsereljav et al. is directed towards automating the time-consuming and error-prone process of lesson transcription and analysis. Features of teacher-student conversation such as word count are derived and analyzed to draw conclusions, achieving speed and accuracy in data processing and simplicity and transparency in the method. However, downsides include the need for manual labor in drawing conclusions from the data generated.

To improve on these works, my project takes full advantage of the use of AI to quickly generate appropriate slideshows with desired content and specified length without requiring the user to provide additional input such as a reference document. In the analysis of lesson transcripts, my system utilizes the OpenAI GPT-4-turbo to generate insightful summaries of student behavior as well as effectiveness of the provided slideshow [7]. Furthermore, my method includes automated teaching feedback, which provides tutors with actionable tips and advice to improve classroom learning.

1. Slide Generation Process

The automated slide generation system follows a structured pipeline:

Input Prompt: The user gives various slide requirements, including:

The main topic of the slides

The total number of slides required

Number of bullet points per slide
Model Utilization: The user input is structured into a prompt, which is then processed using OpenAI's GPT-4-turbo to generate the slide content. The slides material is structured

Response Parsing: The model's output text is parsed to extract the title and content of each slide, ensuring proper formatting for conversion to a PowerPoint.

PowerPoint Creation: The parsed content is converted into a PowerPoint presentation using the Python pptx library. The final output generated is a .pptx file representing the final slides.

2. Lesson Transcript Analysis

To assess the effectiveness of the generated slides, the system analyzes lesson transcripts between the tutor and student using the following steps:

Input Prompt: The transcript is fed into GPT-4-turbo along with the original slide content.

Analysis Criteria: The model evaluates:

Student understanding of the material

Student engagement and interest in the lesson topic

Usefulness of the slides in facilitating learning

Feedback Generation: The model provides 2-3 actionable recommendations for the tutor to improve future lessons.

Slide Content Update: Based on the analysis, the parsed slide text is refined to enhance clarity and relevance.

This methodology ensures a continuous feedback loop, improving both slide quality and teaching effectiveness over time.

With experiment A, we set out to test the capability of the AI system to adapt the difficulty of the slides content. To execute this, we generated three different levels of slides material with our model: introductory, intermediate, and advanced, and compared each output to expert-created slides of the same topic and difficulty. Next, we recruited educators to evaluate both our generated slides and the expert-created slides against the same rubric, revealing the pedagogical qualities of our AI generated slide content through comparison.

Through conducting experiment B, we aim to examine the ability of the AI system to produce quality transcript analysis and consistent slide adjustments. In order to test this, we used manually written transcript annotations by experts as benchmarks. Through comparing the AI generated analysis to these standards, we determine the accuracy of our system in identifying key moments of student-teacher interaction in the lesson [15]. Additionally, we examine whether the model is able to provide specific, actionable feedback supported by effective edits to the slides in contrast to vague, general tips. This process reveals the effectiveness and consistency of our method in providing feedback and enhancing teaching material, using experts' gold standards as reference.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Not Getting the Desired Response from OpenAI

My program relies heavily on OpenAI to generate the contents of the slideshow. Although the requirements such as topic, number of slides, and number of bullets per slide is provided by the user, we face the risk of receiving unsatisfactory output from the AI model. This could involve undetailed content such as vague ideas or points that are lacking in explicitness, or inadequate language/tone in the writing of the slide content. In response to this problem, I could make adjustments to the prompt or provide the AI model with sample slides as a reference to what a desired slide show would look like.

2.2. Transcript Analysis Would Change the Slides too Much

Considering that the edits made to the slides following the lesson transcript analysis should only adjust necessary elements and should maintain the main layout and wording of the original slides, another potential problem of my program would be that the AI model would alter the slide content too much in the improvement process. Therefore, I will adjust the prompt to instruct the model to make minimal changes. To be even safer, I could use the model to generate a number of edit suggestions which could then be passed to the AI along with the slides content as a separate input.

2.3. The Design of the Slides

Since the generated slideshow is meant to be used by the tutor in the lesson with their student, the slides must be visually interesting and engaging to support students' learning experience. "Students viewing text-only PowerPoint slides retained less information than students viewing text-and-image slides…" (Williams et al.) In order to visually enhance the presentation of the lesson material, the slides should have templates that display images relevant to the class topic, and the layout of the information should be logical and effective. To achieve this, I will utilize an automatic template creation pipeline, Desigen, presented by Haohan Weng et al. This will ensure the use of slide backgrounds that "contain the context of a design (e.g., theme-related visual content) while leaving necessary non-salient space for overlaying layout elements" (Weng et al, 2024).

3. SOLUTION

The system operates through two main phases: automated slide generation and lesson transcript analysis, both powered by OpenAI's GPT-4-turbo model.

In the slide generation phase, the user provides an input prompt specifying the topic, desired number of slides, and bullet points per slide. This prompt is processed by OpenAI, which generates a structured slide summary in text format. The system then parses this output to extract slide titles and content, ensuring proper formatting before converting it into a PowerPoint presentation (.pptx) using the Python pptx library.

Following the lesson, the system retrieves the transcript of the interaction between the instructor and student, along with the previously generated slide summary text. These inputs are fed into OpenAI with specific instructions to analyze the discussion. The model evaluates student comprehension, assesses how closely the conversation is aligned with the slide content, and determines the effectiveness of the slides in facilitating learning [8]. Based on this analysis, the system produces a detailed report that includes insights into student understanding, feedback on slide utility, and actionable recommendations for the tutor. Finally, the parsed slide text is updated to reflect improvements, and an updated .pptx file is generated for future use.

This closed loop system ensures continuous refinement of teaching materials while providing tutors with data-driven insights to enhance instructional effectiveness.

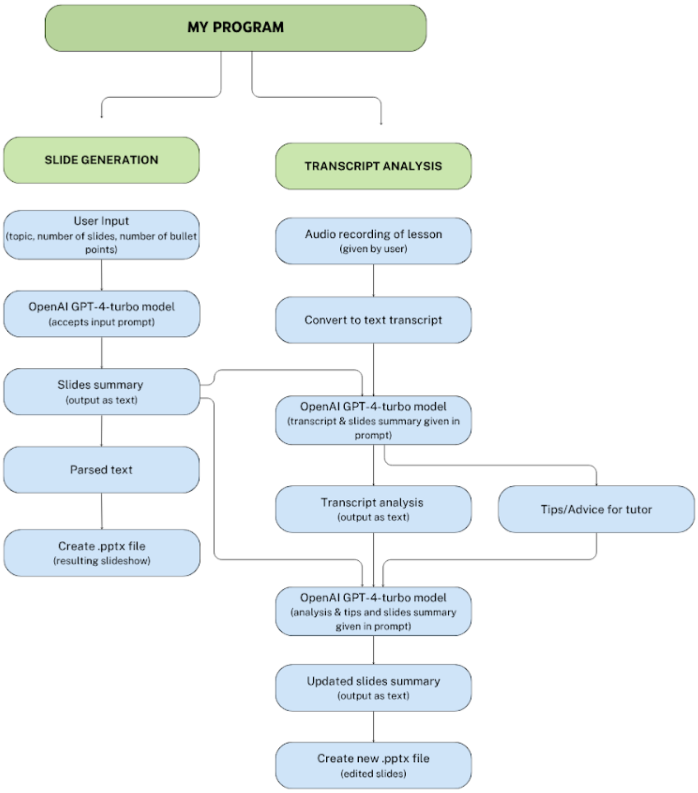

Figure 1. Overview of the solution

Slideshow Generation:

The first major component in this program is the slideshow generation. To automate the slide creation process, the system uses the provided user input, including slide topic, number of slides, and number of bullet points per slide, and feeds the information as a prompt into the OpenAI GPT-4-turbo model [9]. The model then generates the material of the slides as text, defining the title and multiple bullet facts for each slide. After parsing the text response to extract slide titles and content, the material is converted into a .pptx file as the final slides.

```
### Slide 1: Introduction to Cholesterol
- **What is Cholesterol?**
  - A waxy substance found in your blood, essential for building cells.
- **Sources of Cholesterol**
  - Produced by the liver and obtained from animal-derived foods (e.g., meat, dairy).
- **Types of Cholesterol**
  - HDL (Good) cholesterol vs. LDL (Bad) cholesterol.

### Slide 2: Recommended Cholesterol Intake
- **Daily Limits**
  - Less than 300 mg per day for healthy adults.
  - Reduce to 200 mg if high heart disease risk.
- **Importance of Moderation**
  - High levels can lead to heart disease and stroke.

### Slide 3: Impact of Excessive Cholesterol Intake
- **Health Risks**
  - Contributes to plaque, which can clog arteries and lead to heart conditions.
- **Long-Term Effects**
  - Increased risk of coronary artery disease, stroke, and peripheral artery disease.
```

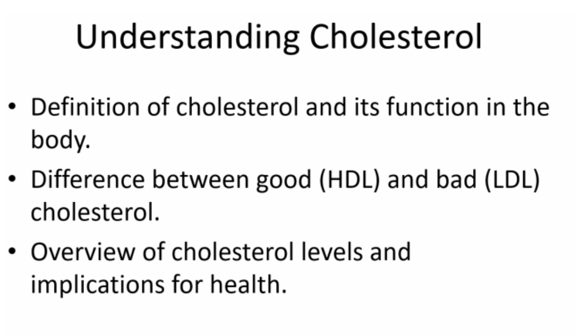

Figure 2. Screenshot of the slides

```python
from openai import OpenAI
from dotenv import load_dotenv
import os

load_dotenv()
client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))

numSlides = "5"
topic = "Cholesterol intake"
bulletPoints = "1-3"

response = client.chat.completions.create(
    model="gpt-4-turbo",
    messages=[
        {
            "role": "user",
            "content": f"Create a {numSlides}-slide PowerPoint presentation on the topic: '{topic}'. Each slide should have a title and
            {bulletPoints} bullet points."
        }
    ]
)

slide_text = response.choices[0].message.content
print(slide_text)
```

```python
def parse_slides(gpt_output):
    slides = []
    current_slide = {"title": "", "content": []}

    for line in gpt_output.strip().splitlines():
        if line.lower().startswith("slide"):
            if current_slide["title"]:
                slides.append(current_slide)
                current_slide = {"title": "", "content": []}
            current_slide["title"] = line.split(":", 1)[-1].strip()
        elif line.strip().startswith("-"):
            current_slide["content"].append(line.strip("- ").strip())

    if current_slide["title"]:
        slides.append(current_slide)

    return slides

from pptx import Presentation

def create_presentation(slides, filename="renewable_energy.pptx"):
    prs = Presentation()

    for slide_data in slides:
        slide = prs.slides.add_slide(prs.slide_layouts[1])  # Title and Content
        slide.shapes.title.text = slide_data["title"]
        content_box = slide.placeholders[1]
        content_box.text = "\n".join(slide_data["content"])

    prs.save(filename)
    print(f"✅ Saved presentation to: {filename}")
```

Figure 3. Screenshot of code 1

In this code, the variables numSlides, topic, and bulletPoints represent the user input specifying the total number of slides required, the main topic of the content, and the number of bullet points to be included on each slide, respectively. Each of these are integrated into the prompt fed into the GPT-4-turbo model, accessed using an OpenAI API key [14]. The slides material is then extracted as text from the model's output, represented by the response variable.

The parse_slides function accepts the slides summary text retrieved from the GPT output and extracts the title and content of each individual slide. The information is returned as an array of dictionaries, representing a list of slides that each have a "title" and "content" component.

The create_presentation function takes in the parsed slides and a specified filename and creates a .pptx file using the python pptx library, saving it to the provided file name.

The program's second component is transcript analysis. The aim of analyzing the lesson transcript is to assess the effectiveness of the generated slides in teaching, in addition to improving students' learning experience through pinpointing areas of improvement for the tutor as well as the slideshow creation. The implementation of this component begins with retrieving the text transcript of the lesson between the tutor and the student. The slides summary from the slideshow generation component is also retrieved. The two are then provided in an OpenAI prompt to be fed into the GPT-4-turbo model with instructions to analyze details of the lesson, including the engagement and interest of the student, the student's comprehension, and the helpfulness of the slides. Additionally, the model provides a few pieces of advice for the tutor on how they could adjust their teaching methods to enhance student engagement and comprehension.

To generate the edited slides with suggested improvements, the given analysis along with the slides summary are inputted once again into GPT-4-turbo as text. The model considers the results of the analysis in addition to the advice given to the tutor to make adjustments to the slides with the goal of aiding the tutor in providing better quality teaching and enhancing student experience. The new slides, which could include added quiz questions, reinforced points, summaries, etc, are outputted as a text and converted to a .pptx file through the same parsing and PowerPoint creation process as the slide generation component.

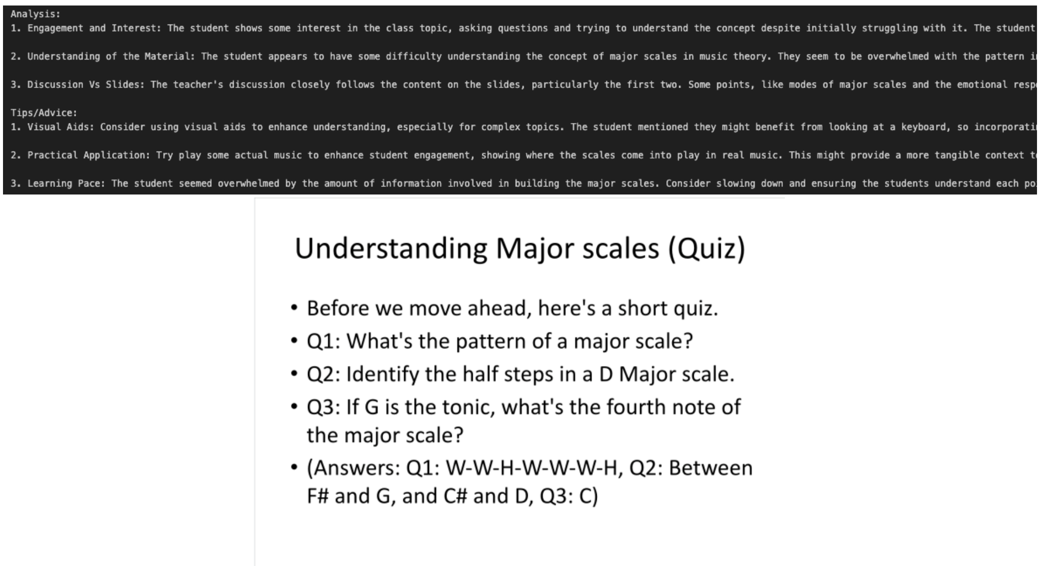

Figure 4. Screenshot of UI

```
load_dotenv()
client = OpenAI(api_key=os.getenv("MY_OPENAI_API_KEY"))

with open("sample_text4.txt", "r", encoding="utf-8") as f:
    transcript = f.read()

with open("presentation_output.txt", "r", encoding="utf-8") as f:
    slide_summary = f.read()

prompt = f"""
You are an education assistant. Analyze the following classroom transcript in the context of the provided PowerPoint slides. Give an analysis of the following:

- The engagement and interest of the students in the class topic.
- The students' understanding of the material being taught.
- How closely the discussion followed the slides and how helpful the slide content was in teaching.
- Provide 2-3 pieces of tips/advice for the tutor on how they can improve student engagement or how they can help students better understand the class content.

Transcript:
{transcript}

Slides:
{slide_summary}
"""

response = client.chat.completions.create(
    model="gpt-4",
    messages=[{"role": "user", "content": prompt}]
)

analysis_text = response.choices[0].message.content
```

```
prompt = f"""
Based on the provided analysis of a lesson transcript and the provided content of the slides used by the teacher, create a new slideshow by adjusting the original slides
according to the tips/advice given to the tutor.
The slides are meant to be shown to the student as the teacher is going through the lesson, and the points listed should represent what the teacher is going to say.
Do not remove any slides. Based on the suggestions from the analysis text, add slides accordingly.
If added content includes quizzes, summaries, etc., be sure to include the specific content on the slides as bullet points. Also include the correct answer to each question
in the slides.

Transcript Analysis: {analysis_text}

Slides summary: {slide_summary}
"""

response = client.chat.completions.create(
    model="gpt-4",
    messages=[{"role": "user", "content": prompt}]
)


new_slides = response.choices[0].message.content
```

Figure 5. Screenshot of code 2

The lesson transcript and generated slides material from OpenAI are both retrieved as text and stored in the variable's transcript and slide_summary respectively. The prompt provides these two variables along with instructions to evaluate the student's engagement and comprehension, assess the effectiveness of the generated slides, and provide pieces of instructional advice to the tutor. After feeding the prompt into GPT-4-turbo, the output of the model yields the desired transcript analysis listed in the form of bullet points and is stored into the analysis_text variable [13]. Following this, a second prompt for making edits to the slides is given to the GPT-4-turbo model with analysis_text and slide_summary to produce the edited slides text, stored in new_slides. The functions parse_slides and create_presentation from component A are utilized again to generate the .pptx file for the final PowerPoint presentation.

4. EXPERIMENT

4.1. Experiment 1

With our current program, the user (tutor) decides the topic and approximate amount of content of the generated slides. However, they hold little control over the quality and depth of the information presented. Thus, the content of the AI generated slides may seem too easy and surface-level in some cases and too challenging in others. In order to facilitate proper learning, it is key that the slides material suits the level of the student.

To test the AI's ability to adapt content difficulty, we will conduct a controlled experiment comparing AI-generated slides at different complexity levels against expert-created benchmarks. For three distinct topics (introductory, intermediate, and advanced), we will generate slide sets using our system's difficulty selector while simultaneously having subject matter experts manually create "gold standard" versions for each level. These expert-curated slides will serve as our control data, providing an objective reference for appropriate depth and accuracy at each complexity tier. We will then recruit educators to blindly evaluate both the AI-generated and expert-created slides using standardized rubrics that assess conceptual appropriateness, terminology difficulty, and pedagogical effectiveness for the target level. The experiment is designed this way to isolate the AI's adaptive capabilities while controlling subject variability, with the expert benchmarks allowing us to quantify how closely the automated system approximates ideal human-adjusted content. By testing across multiple disciplines and involving actual educators in the evaluation, we can identify whether the difficulty adapting works consistently or fails in predictable patterns, guiding targeted improvements to our prompting and validation systems.

The AI generated content demonstrated strong consistency in both structure and thematic coherence, reliably adhering to the requested format such as slide count and bullet point structure. Across multiple slide generations, the system produced uniformly structured presentations with stable titles, content hierarchies. This consistency indicates production ready reliability, however, we have yet to test its production with more complex content. The baseline consistency confirms the system's current suitability for standardized slide generation.

4.2. Experiment 2

In the transcript analysis portion of our program, despite specific instructions provided in the OpenAI prompt, sample testing exposes the risk of the GPT-4-turbo model providing vague and general feedback rather than details specifically tailored to the individual tutor and student. While the advice still proves to be helpful, it lacks personalization and the potential to act as quick, actionable feedback that can be easily adopted by the tutor. Additionally, the updated slides text outputted by the AI system holds uncertainty in that it lacks consistency in execution of the suggested improvements. The factor of inconsistency in slides output must be eliminated in order for the feedback to be effective and beneficial to tutors' teaching and students' learning experience. Both the analysis of the lesson transcript and the editing of the slides exist as key components to improving instruction of lesson material and tutor-student interactions.

To rigorously test the AI's transcript analysis and slide-updating capabilities, we will design an experiment using anonymized real-world tutoring transcripts spanning multiple subjects and teaching styles. These transcripts will be manually annotated by education experts to create ground-truth benchmarks identifying key discussion points, student misconceptions, and required slide modifications. The AI system will process these transcripts to generate both analyses of the lesson (evaluating student understanding and slide effectiveness) and concrete slide updates, which we will compare against the human expert benchmarks using quantitative metrics like detection rate of critical discussion points and qualitative assessment of update specificity. We'll particularly examine whether the AI merely suggests general improvements ("explain this concept better") versus providing actionable, text-level revisions to slide content. The experiment is structured this way to expose inconsistencies in output formatting and substantive gaps between identified teaching needs and implemented slide updates, using expert judgments as the gold standard. This design will reveal whether the system currently adds meaningful value to the teaching feedback loop or requires additional constraints, like output templates or multi-stage prompting, to ensure analyses consistently translate into usable slide improvements. Control data from human experts allows us to distinguish between acceptable variations in teaching approaches and genuine AI shortcomings in educational content adaptation.

5. RELATED WORK

Similar approaches to the problem of automated slide generation include the presentation of PPTAgent by Hao Zheng et al. In their paper "PPTAgent: Generating and Evaluating Presentations Beyond Text-to-Slides," they tackle the issue of high effort cost in creating quality presentations by proposing an edit-based approach using LLMs [10]. Their system leverages retrieved reference slides and an input document to iteratively generate a new slides presentation. Their process of analyzing reference presentations includes "slide clustering and schema extraction" (Zheng et al., 2025). By categorizing slides based on visual and functional similarities, and analyzing content schemas, their system generates an outline with specified reference slides. By combining these reference slides with relevant document content, the new slides are generated iteratively with the information from the document incorporated into the layout of the reference presentation. To evaluate the results of their solution, Zheng et al. compare PPTAgent to two baseline methods: a rule-based approach called DocPres (Bandyopadhyay et al., 2024), and KCTV (Cachola et al., 2024), which is a template-based approach. Their presented evaluation system PPTEval assesses the quality of a presentation through content, design, and coherence. Evaluation results show that "PPTAGENT demonstrates statistically significant performance improvements over baseline methods across all three dimensions of PPTEVAL" (Zheng et al., 2025). However, the authors also mention limitations of their solution: "the model occasionally fails to generate presentations" and results show "occasional design flaws, such as overlapping elements" (Zheng et al., 2025). Furthermore, the success of their system depends on the input of relevant, high-quality documents, which many instructors may not readily have access to. In contrast, my approach to this problem simply requires the tutor to state the topic of the slides, the desired number of slides, and the number of bullet points to be included per slide without having to input any additional material.

To enhance my automated slide generation system, insights from this article suggest valuable improvements while maintaining simplicity. While my current system efficiently creates slides from minimal tutor inputs, incorporating optional reference-based generation could improve output quality without complicating user experience. The system could also benefit from implementing iterative refinement options and basic quality checks inspired by PPTEval to flag potential issues in text density and layout problems.

"DOC2PPT: Automatic Presentation Slides Generation from Scientific Documents" by Tsu-Jui Fu et al. introduces a system aimed towards enhancing human productivity by generating slides presentations from inputted documents [11]. Their "hierarchical sequence-to-sequence approach" features text summarization, image and sentence retrieval, and layout design. The network is composed of "modularized components that are jointly trained in an end-to-end fashion" (Fu et al., 2022), including a Document Reader, Progress Tracker, Object Placer, and Paraphraser. The system functions through encoding sentences and figures from the provided document while tracking progression in both the document and the slides with the use of pointers. The Object Placer decides whether the text or figure to be placed on the current slide as well as its size and location, and the Paraphraser paraphrases sentences to be more concise before inserting it in the slide. The result yields a slide deck with each slide including a title, bullet pointed content, and a figure such as an image or graph as appropriate. In the evaluation of their solution, Fu et al. recruit human evaluators to rate the similarity between the slides presentations generated by their model and ground-truth slide decks. On a scale of 1 to 7, with 1 representing no similarity and 7 representing high similarity, the average ratings for only text, only figures, and text to figures were 4.2, 4.0, and 4.6, respectively. The paper also presents a comparison of their method against two others with differences in the implemented model: one using a non-hierarchical Seq2Seq model, and one featuring a retrieval-based approach. Results show a statistically significant difference in that "The average rating for our approach was significantly greater [...] compared to the other two methods" (Fu et al., 2022). Even so, their solution exhibits shortcomings in that it did not score significantly high in the human evaluated similarity ratings. Additionally, the design of the generated slides, especially the background template, remains basic and potentially boring for students to observe. Furthermore, this approach presents the same issue as the previously discussed PPTAgent in that users are required to provide an input document for desired results.

Fu et al.'s DOC2PPT (2022) shows how a modular system can help generate slides by using separate parts for summarizing text (Document Reader), arranging content (Object Placer), and shortening text (Paraphraser) [12]. Their system did a decent job keeping content from the original documents (similarity scores of 4.2–4.6 out of 7), but it depends heavily on input documents and uses basic slide designs—limitations also seen in PPTAgent. Since my system doesn't rely on input documents, their work suggests three useful ideas:

Add a small paraphrasing tool to make bullet points clearer without needing full documents.

Offer simple layout options, like picking themes or highlighting figures (inspired by the Object Placer), while keeping things mostly text based.

Use slide titles to help track and guide topic flow, improving the order and logic of slides.
While we won't use DOC2PPT's full training, adding these modular parts could improve results without making the system harder to use. Their average scores also show a bigger issue: matching content isn't enough; we also need to measure how useful the slides are for learning.

Other methods directed towards lesson transcript analysis include the utilization of Microsoft Excel for quantitative and qualitative analysis, as described in GereltuyaTsereljav et al.'s "Using Microsoft Excel for Lesson Transcript Analysis." The goal of their work is to automate the "Transcript-Based Lesson Analysis" process to eliminate the time-consuming and error-prone issues of manual transcription and analysis. Tsereljav et al. utilize Microsoft Excel's Visual Basic for Applications and graphical user interface functionalities in the development of their digital tool. In their process, speech to text transcription is used to convert audio and video files of classroom recordings to text through Chimege API. After obtaining a table of teacher and student dialogue, features such as teacher's/student's word count and teacher's question pattern are derived, with "certain variables [...] computed directly through integrated formulas and macros, whereas others require manual recoding prior to analysis" (Tsereljav et al., 2025). This information is then organized into visual graphs and summarized tables, which can be used to draw meaningful conclusions about student-teacher interaction such as how much the teacher dominated the conversation. Results show that this solution using MS Excel provides speed and accuracy in processing and analyzing classroom data. In conclusion, the method presented by Tsereljav et al. is advantageous in its simplicity and adaptability, as well as its transparency: "Unlike AI systems, which often operate as 'black boxes', the Excel tool allows educators to see and understand how data is processed clearly" (Tsereljav et al., 2025). Nevertheless, there are also downsides to this approach since many details such as students' response to teachers' specific teaching methods cannot be identified. Furthermore, the generated data requires manual analysis in contrast to automatically generated conclusions given in AI systems. In view of these flaws, my solution utilizes AI to directly generate analysis of targeted features as well as tips and advice for the tutor as immediate feedback.

6. CONCLUSIONS

One of the current limitations of the system lies in the depth and complexity of the generated slide content, which may not always align perfectly with the intended educational level or subject matter expertise required. While AI can produce structured outlines, ensuring nuanced explanations or advanced technical details still pose a challenge. Additionally, the visual representation of slides is currently very basic, focused primarily on textual content without dynamic design or elements. This can reduce engagement for certain audiences. The slides also lack interactive or animated transitions, resulting in a static presentation format that may feel less polished compared to a manually designed one.

To address these limitations, future implementations could introduce customizable design options, allowing users to import images or select predefined themes to enhance visual appeal. Another planned improvement is the integration of AI-powered slide transitions, which would automatically apply smooth animations between slides. AI-powered slide design can also improve visuals and slide representation to encourage engagement with the audience. These enhancements would not only improve the aesthetic quality of presentations but also make the tool more adaptable to diverse teaching needs, ultimately creating a more engaging and effective learning experience.

This project demonstrates the significant potential of AI in streamlining educational content creation, particularly in automating the time-consuming process of slide preparation. By leveraging OpenAI's capabilities, we have developed a system that not only generates structured slide content efficiently but also provides valuable insights through post-lesson transcript analysis. While current limitations exist, the framework establishes a strong foundation for future enhancements.

The integration of more sophisticated AI-driven features, such as adaptive difficulty levels, automated visual enhancements, and intelligent transitions could further bridge the gap between machine-generated and manually crafted presentations. Most importantly, this tool empowers educators to redirect their efforts from administrative tasks to fostering student engagement and improving learning outcomes. As AS technology continues to evolve, so too, the possibilities for refining and expanding the system, ultimately contributing to more effective and accessible education tools worldwide.

This project underscores the transformative role AI can play in education, not as a replacement for instructors, but as a powerful tool in enhancing teaching efficiency and effectiveness. Future iterations will focus on personalization, interactivity and seamless integration info different educational environments.

REFERENCES

[1] Ravi, Vineeth, et al. "AI pptX: Robust Continuous Learning for Document Generation with AI Insights." arXiv preprint (2020): arXiv-2010.
[2] Dar, Zaema, et al. "Advanced generative AI methods for academic text summarization." 2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI). IEEE, 2024.
[3] Zheng, Hao, et al. "Pptagent: Generating and evaluating presentations beyond text-to-slides." arXiv preprint arXiv:2501.03936 (2025).
[4] Fu, Tsu-Jui, et al. "Doc2ppt: Automatic presentation slides generation from scientific documents." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36. No. 1. 2022.
[5] Dais, Biju P., and C. S. Smitha. "A comparison on techniques for automatic generation of presentation slides." International Journal of Computer Applications 975 (2015): 8887.
[6] Lok, Simon, and Steven Feiner. "A survey of automated layout techniques for information presentations." Proceedings of SmartGraphics 2001.61-68 (2001): 1.
[7] Weng, Haohan, et al. "Desigen: A pipeline for controllable design template generation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.
[8] Tsereljav, Gereltuya, et al. "Using Microsoft Excel for Lesson Transcript Analysis." Lavai-International Journal of Education 21.32 (2025): 41-52.

[9] Wang, Ziyang, et al. "Leveraging AI Technology for Coding the Classroom Observation Record Form of Flanders Interaction Analysis." IEEE Access (2025).
[10] Demszky, Dorottya, et al. "Can automated feedback improve teachers' uptake of student ideas? Evidence from a randomized controlled trial in a large-scale online course." Educational Evaluation and Policy Analysis 46.3 (2024): 483-505.
[11] Williams, Joshua L., et al. "Images and text on PowerPoint slides:"Tracking" their impact on information retention." International Journal of Mobile and Blended Learning (IJMBL) 14.1 (2022): 1-14.
[12] Aggarwal, Tushar, and Aarohi Bhand. "PASS: Presentation Automation for Slide Generation and Speech." arXiv preprint arXiv:2501.06497 (2025).
[13] Shahandashti, Kimya Khakzad, et al. "Evaluating the effectiveness of gpt-4 turbo in creating defeaters for assurance cases." arXiv preprint arXiv:2401.17991 (2024).
[14] Auger, Tom, and Emma Saroyan. "Overview of the OpenAI APIs." Generative AI for Web Development: Building Web Applications Powered by OpenAI APIs and Next. js. Berkeley, CA: Apress, 2024. 87-116.
[15] Teclehaimanot, Berhane, and Torey Hickman. "Student-teacher interaction on Facebook: What students find appropriate." (2011): 19-30.