Syntactic Generation of Research Thesis Sketches Across Disciplines Using Formal Grammars

Ismail Babajide Adewumi ¹, Abejide Ade-Ibijola ²

¹ Applied Information Systems Department, School of Consumer Intelligence and Information Systems, College of Business and Economics, University of Johannesburg, Johannesburg, South Africa
² Research Group on Data, Artificial Intelligence, and Innovations for Digital Transformation, Johannesburg Business School, University of Johannesburg, Johannesburg, South Africa
Email: ¹218027873@student.uj.ac.za, ²abejide@jbs.ac.za

Abstract

A part of the prerequisites for granting a degree in higher education institutions, students at postgraduate levels normally carry out research, which they do report in the form of theses or dissertations. Study has shown that students tend to go through difficulties in writing research thesis across all disciplines because they do not fully comprehend what constitutes a research thesis. This project proposes the syntactic generation of research thesis sketches across disciplines using formal grammars. Sketching is a synthesis technique which enables users to deliver high-level intuitions into a synthesis snag while leaving low-level details to synthesis tools. This work extends sketching to document generation for research thesis documents. Context-free grammar rules were designed and implemented for this task. A link to 10,000 generated thesis sketches was presented.

Keywords: Formal grammars, natural language generation, research thesis, thesis sketches, sketching, synthetic data generation, procedural content generation

1. INTRODUCTION

A thesis is an essential academic document for all postgraduate students, regardless of discipline [1], [2], [3]. Unless it is exclusively based on a critical review of existing theories or opinions, a thesis requires some originality [4]. However, the majority of master’s and doctoral students struggle with their thesis because they term it to be an intimidating and potentially boring task [5]. A thesis primary goal, in accordance with Puspita [6], is to clarify the phenomena, truths, and issues that have been investigated in order to derive a conclusion from the study.

Evans et al. [7], stated that a thesis must show logical, systematic, and defensible

This work is licensed under a Creative Commons Attribution 4.0 International License.
reasoning based on reliable and verifiable data presented in such a way that it offers an original contribution to knowledge, as judged by experts in the subject, to pass. A thesis is a typewritten manuscript, typically 100 to 400 pages long, in which a student addresses a specific problem in his or her chosen field [8]. Thesis must show the candidate’s capability to do suitable research, the ability to compile collected data into a comprehensible technique, and the ability to effectively communicate the findings [9].

The results of Bui [10] show that a thesis demonstrates one’s capacity to do original research, evaluate prior literature, gather data, analyze data, publish results, explain conclusions, and infer implications from study results. A thesis should specifically state how it advances knowledge in a particular field [11]. In regard to the thesis, master and doctoral students need to complete research projects in order to be granted their postgraduate degree. Puspita [6] said the purpose of the thesis is to conduct research and write up the results of the investigation. The significance of the thesis is critical since it demonstrates the work’s relevance, potential outcomes, and the major beneficiaries of the study [12].

Thesis is still seen as a threat to master’s and doctorate students across a variety of disciplines. Many students find writing a thesis to be a challenging task, despite the fact that effective writing can help them achieve better academic results [13]. The majority of them struggle to complete theses, especially the literature review component [14] [15]. Particularly at the PhD thesis stage, students struggle to articulate and integrate concepts, as well as maintain consistency plus connectivity between sections and subsections of thesis [16] [17]. Randolph [18] indicated that errors in the literature review can result in defects in the other portions of the thesis. In order to enable readers to understand the distinctive and appropriate strategy, the literature review in each study ought to be written admirably [19]. Bitchener and Basturkmen [20] claims that insufficient fundamental understanding of the thesis and its elements, such as the background of the study and methodology may lead to students facing difficulties with their theses.

The structuring of thesis is generally similar regardless of university or institution and Agarwal et al. [21], suggested that Preface, Introductory Section, Methods and Design (including statistical analysis), Results, Discussion, Conclusions, References, and Appendices should be the format for thesis structure. However, a study by Lestari [22] and Turmudi [23] proves that difficulties within thesis could be caused by a number of things, such as the purpose of the research, the data utilized during the research, the conclusions, or the fact that the entire study is based on a reliable proportion.
In addition to this paper, providing theoretical, practical solution for prospective students or candidate in sketching research thesis across disciplines is our goal and the result of this paper contributed the following to the trends in:

1) the design of grammar rules for the syntactic generation of research thesis sketches,
2) an algorithm for the thesis sketches, based on the new grammar rules,
3) an implementation of the algorithm sketches in .Net Framework version 4.8.1 using C# as the programming language. Functions were dedicated to creating LaTeX (.tex) files, the generated LaTeX files are all compilable and we ensured that the speed of creating synthesized files on disk did not result in deadlock and
4) an evaluation of the generated thesis sketches was carried out and it showed that techniques used achieve accurate thesis sketches.

The structure of this paper is as follows: background and related work is discussed in section II, grammar design of sketches is illustrated in section III which is the methods. The implementation, results and evaluation of the study are discussed in section IV while the conclusion of this study and future work are covered in section V.

2. BACKGROUND OF THE STUDY AND RELATED WORK

2.1 Technology Overview

In our daily activities, technologies have been so helpful in many aspects of our lives. Numerous tools have been created to address issues, such as visualization tools that offer graphic descriptions of programs and summarizing tools that offer a concise written summary of a program’s functioning [24]. The creation and manipulation of strings has gained importance in many different areas of computing [25] [26].

2.1.1 Synthetic Generation of Documents Definitions

Synthetic document or data generation is a useful tool for a range of applications, including software testing, machine learning, and privacy protection [27]. A semi-synthetic document generator is recommended by Journet et al. [28], which generates new documents by changing both the font and background of a group of genuine documents. Yang et al. [29] generates synthetic documents by randomly arranging layout components in LATEXsource files, as well as replacing pieces of elements from pre-existing files in an unpredictable manner. According to a study by Anderson et al. [30], synthetic documents are frequently
produced by constructing artificial values with the similar format as the genuine data using statistical distributions for a set of samples based on tangibly measured data. In addition, synthetic documents generation can generate a large number of documents effortlessly with limited user contribution [31].

2.1.2 Procedural Generation Definition

The term procedural generation is a method used in computing to generate content (basically gaming) which requires important efforts from humans [32]. Also, automatic generation of digital assets for games, imitations built on predetermined algorithms are patterns known as procedural content generation which require little to no human interaction [33]. Procedural content generation, as it relates to Smelik et al. [34] work, is described as any automatically generated asset that is founded on a certain cluster of user-specified input parameters. These algorithms, which convert a little collection of input parameters into a huge array of output data, are known as amplification algorithms according to [35]. In regard to [36] findings, procedural generation offers some advantages such as consumption of memory (content can be generated when needed), development effort (content creation manually is mitigated) and longevity (different generated content is assured whenever user input is predefined). However, the application of procedural generation presents some shortcomings as the outcome can be an unpredictable array of possible game scenarios or content generated, thus, there is a need to put in some constraints that will only allow content to be generated in a correct manner. Therefore, formal grammar was used in executing the syntactic generation of research thesis sketches across disciplines.

2.1.3 Natural Language Generation

The automatic creation of a comprehensible document or text from non-linguistic or textual representations of information is known as natural language generation according to [37]. NLG tasks can be separated into low entropy (summarization, machine translation) and high entropy (story generation, casual conversation) tasks [38]. Through content/text planning, sentence planning, and syntactic realization, NLG increased the quality of texts [39] [40]. Kondadadi et al. [41] claims that NLG consists of sentence planning, which is nothing more than the selection of significant words, meaningful phrases, and sentences, and text planning, which is getting the pertinent material from the domain.

2.1.4 Formal Grammars Definitions

Formal grammars were initially introduced as a method to describe language [42]. In speech recognition, language translation, and language understanding systems,
formal grammars are frequently utilized [43]. The formal notions of grammars serve as the foundation for the broader framework for expressing languages [44]. van Rozen and Heijn [45] said a formal grammar comprises a set of rewrite rules. When applying the rules to a string, each symbol or group of symbols in the string is replaced [45]. According to Ade-Ibibola and Ogbuokiri [44], grammar is a finite non-empty set of production rules that consists of two distinct sets of symbols known as nonterminal symbols, terminal symbols and the start symbol.

2.2 Related Work

In this part of the paper, the review of related literature on sketches and formal grammar were discussed including the application of sketching in general. Sketching is a synthesis technology which combines effortlessly into an imperative programming model, entirely redefining the connection between the synthesizers and the users and possibly conveying synthesis closer to general adoption [46].

2.2.1 What is Sketching?

Sketching is a novel practice of software synthesis that targets gaps among a programmer's high-level understandings about difficulty and the ability of a computer to manage low-level details as well as bridging them together [47]. Sketching allows users to create a sketch, that is, a partial program made up of breaches or holes, and an automated tool will complete the sketch in accordance with a specified description [48] [49]. The research work of Cunningham et al. [50] said sketches have the capacity to both offload cognitive and improve thought coordination. Since a simple description is required to indicate the functionality, sketches encourage clear specifications. With automatic low-level detail synthesis, sketching promises to help complex implementations that could be too tedious to create and maintain [51]. According to Taele et al. [52], integrating intelligent computer-assisted educational systems into sketching modes for academic disciplines will ensure that sketching continues to be an essential part of the learning procedure.

2.2.2. Sketching: AI Tools, Definitions and the Application in Education for Research Thesis across Disciplines?

The genesis of AI (artificial intelligence) goes back to the 1950s [53]. Baker et al. [54] provide a detailed definition of artificial intelligence (AI) and refers AI to computers that carry out intellectual tasks that are normally allied with human thought, mostly learning as well as critical reasoning. Chassignol et al. [55] defined AI as a theoretical framework guiding the evolution and use of computer
systems with the abilities of human beings, mainly, intelligence and the ability to perform responsibilities that involve human intelligence, including visual observation, speech recognition, decision-making, plus translation of languages. The use of AI in education is increasing as well as the attention given to it recently is enormous. As indicated by a report, professionals envisage that AI in education will advance by 43% between 2018 and 2022, in spite of this, Horizon Report on 2019 Higher Education Edition (Educause, 2019) forecasts that AI uses identified with teaching and learning will become considerably and more fundamental compared to the present [56]. Jones et al. [57] states that certain AI could be applied in research, not merely to carry out analytical duties but also to produce testable theories. In addition, AI applications in research can be useful for text and data mining [58]. One of the essential objectives of artificial intelligence is the developments of computational techniques for natural language understanding (NLU) [59]. Natural language understanding defines the intentions as well as extracting the suitable data from the users’ request that was taken as input [60].

2.3 Existing Tools

The use of formal grammars or procedural content generation has been the subject of numerous published publications. Examples of the related work are; regular expression problems and solutions [61], social media profiles for Facebook fields [62], practice problems and solutions in Python [63], and hypothetical sociograms [64]. As a design technique, sketching has also been used in related works such as line by line, part by part: collaborative sketching for designing [65] and the role of sketching in engineering design and its presence on engineering education [66]. However, we could not find syntactic generation of research thesis sketches across disciplines using formal grammars. In addition to works that has been done in sketching and their applications are listed below:

Table 1. Sketching as a Technique and Previous Applications

<table>
<thead>
<tr>
<th>Author/s</th>
<th>Work Related</th>
<th>Title Application</th>
<th>Method used compared to formal grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>LaViola Jr and Zeleznik [67]</td>
<td>A system for the creation and exploration of mathematical sketches.</td>
<td>Graphing, modifying, and solving equations with computational tools are also possible with mathematical sketches.</td>
<td>MathPad2 uses handwritten mathematical expressions; users can leverage their knowledge of mathematical notation in order to create mathematical sketches.</td>
</tr>
<tr>
<td>Ouyang and Chemink</td>
<td>A conditional random field</td>
<td>It operates in real-</td>
<td></td>
</tr>
</tbody>
</table>

Ismail Babajide Adewumi, Abejide Ade-Ibijola | 701
<table>
<thead>
<tr>
<th>Author/s</th>
<th>Work Related</th>
<th>Title Application</th>
<th>Method used compared to formal grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Davis [68]</td>
<td>natural real-time recognition system for chemical drawings.</td>
<td>time and offers visual feedback on the status of chemical structure recognition.</td>
<td>with discriminative training was used to construct a framework for sketch recognition that incorporates a hierarchy of visual cues into a single model.</td>
</tr>
<tr>
<td>Williford et al. [69]</td>
<td>Exploring the potential of an intelligent tutoring system for sketching fundamentals.</td>
<td>Instead of replacing human instructors, this system was created to improve the educational process of learning to sketch.</td>
<td>Grounded theory was utilized for this study.</td>
</tr>
<tr>
<td>Cheema and LaViola [70]</td>
<td>Physicsbook: a sketch-based interface for animating physics diagrams.</td>
<td>It was developed with a sketch-based interface for the purpose of addressing physics difficulties.</td>
<td>Diagram primitives as well as mathematical expressions are used to carry out this study.</td>
</tr>
<tr>
<td>Nelligan et al. [71]</td>
<td>Mechanix: a sketch-based educational interface.</td>
<td>It is an automated system to aid students learning introductory engineering.</td>
<td>Mechanix uses a combination of sketch recognition and numerical questions to evaluate student work and provide personalized feedback.</td>
</tr>
<tr>
<td>Solar-Lezama [49]</td>
<td>Program sketching</td>
<td>It is a synthesis method that can deal with challenging issues from many different fields, such as concurrent data structures, scientific programs, and ciphers.</td>
<td>The sketch synthesizer utilizes counterexample-guided inductive synthesis (cegis).</td>
</tr>
<tr>
<td>Solar-Lezama et al. [51]</td>
<td>Combinatorial sketching for finite programs.</td>
<td>Its ability to create an implementation that is functionally identical to the specification, or a right implementation, is its key characteristic.</td>
<td>This study employs a combinatorial synthesizer that can both express any finite program and sketch any implementation of it.</td>
</tr>
<tr>
<td>Nye et al. [72]</td>
<td>Learning to infer program</td>
<td>This method imitates how people can solve</td>
<td>The method employed in this related work includes an</td>
</tr>
</tbody>
</table>

702 | Syntactic Generation of Research Thesis Sketches Across Disciplines Using Formal
<table>
<thead>
<tr>
<th>Author/s</th>
<th>Work Related</th>
<th>Title Application</th>
<th>Method used compared to formal grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bragg et al.</td>
<td>Program sketching by automatically generating mocks from tests.</td>
<td>It's a unique method that transforms a typical sketch hurdle into a modular sketch hurdle as a result of generation of mocks automatically from harnesses.</td>
<td>This study uses counterexample-guided inductive synthesis (CEGIS), a technique that alternately synthesizes a candidate solution and then employs a validator to verify the claims.</td>
</tr>
<tr>
<td>Murali et al.</td>
<td>Neural sketch learning for conditional program generation.</td>
<td>Gaussian Encoder-Decoder or GED, was proposed to learn a distribution over sketches conditioned on labels.</td>
<td>The approach used is a combination of neural learning and type-guided combinatorial search.</td>
</tr>
</tbody>
</table>

2.4 Gap and Motivation

Regarding the background of the study, masters and doctoral student’s struggle to sketch a research thesis and tools or technologies are required to provide support for prospective students. Such tools will help to mitigate errors in sketching research thesis and provide structure that potential students will follow in sketching their research thesis.

3. METHODS

This work uses the Design Science Research (DSR) methodology. DSR allows a stepwise iteration through the process of identifying a problem, to designing artifacts, and evaluating the designed piece [75], [76]. DSR is also used across disciplines in Information Systems, Management, Engineering, and Technology in general [77], [78], [79]. The problem identified was previously presented in the introduction section of this paper. The newly designed artifact of this paper is a formal grammar and a tool for the syntactic generation of research theses.
sketches across disciplines. The process of generating these sketches is described in Figure 1. In Figure 1, we used existing data (scraped from the internet) to design a file of fragments of multidisciplinary templates. This is passed to a thesis sketching module that is based on a CFG-driven algorithm. As an output, sketches are generated.

![Diagram](image)

**Figure 1.** Process of generating research thesis sketches across disciplines using CFG rules

We relied on primary data sources obtained from different repositories such as Google search, UJoogle, and Google scholar for the creation of the data file for the fragments of multidisciplinary templates. In the next section, we present rules of the CFG used by the sketching algorithm.

4. RESULTS AND DISCUSSION

4.1 Formal Grammar for the Language of Sketches

In this section, we defined grammar for the generation of the language (set of strings) of sketches. We begin by establishing mathematical foundations for the domain of formal language theory.

Context-free grammar (CFG) \( G \) is a four-tuple: \( G = (N, \Sigma, P, S) \). According to context-free grammar (CFG), \( N \) stands for a set of tokens or symbols called non-terminal; they are likewise identified as syntactic variables as they signify phrases and clauses in a sentence. \( \Sigma \) stands for a finite set of symbols or tokens
called terminal symbols. This set is disjoint from $N$, the actual set from which a sentence’s content is constructed. Additionally, $G$ is recognized as the alphabet of the language that a grammar $G$ defines. $P$ stands for a finite set whose elements are referred to as production rules. Each production rule is made up of two parts: left and right sides. The left side of the production rule is a non-terminal, while the right side is a series of terminal and/or non-terminal symbols. As the start rule, $S$ is referred to as such. Let the below expression represent the foundation for the generation of language sketches.

$$S \rightarrow F_p \ C_{set} \ (A_{pen} \ | \ \lambda) \ R_{ef}$$

(1)

Where $S$ is the start symbol, $F_p$ represent front pages of the thesis, $C_{set}$ stands for Chapter set, $A_{pen}$ symbolizes the Appendices of a thesis and $R_{ef}$ signify References. We now define production rules for syntactic generation of research thesis sketches across disciplines using formal grammars.

$$F_p \rightarrow T_p \ D_p \ A_{ck} \ (D_d \ | \ \lambda) \ (A_{br} \ | \ \lambda) \ T_c \ (A_{bs} \ | \ Preface)$$

(2)

Where $T_p$ is the title page, $D_p$ stands for declaration page, $A_{ck}$ represent the acknowledgement section of a thesis, $D_d$ is the dedication section of a thesis, $A_{br}$ indicate abbreviation. $T_c$ stands for table of contents while $A_{bs}$ represent the abstract of a thesis. This phase produces a title page with three different formats to use.

$$T_p \rightarrow t_{p1} \ | \ t_{p2} \ | \ ... \ | \ t_{pn}$$

(3)

On the declaration page, students attest to the originality and self-conducted nature of their research. Also included should be full names, student ID number plus supervisor details.

$$D_p \rightarrow d_{p1} \ | \ d_{p2} \ | \ d_{p3} \ | \ ... \ | \ d_{pn}$$

(4)

An acknowledgement section in a research thesis is a part that identifies all of the people that helped with the study. It is customary in academic literature to thank the efforts of benefactors, departments, and individuals who assisted with the research. Hence, we define the production rules as

$$A_{ck} \rightarrow API(users-input,sample \ text)$$

(5)
We formed a production rule for the dedication of a thesis. The dedication has been referred to as a personal note of thanks to known ones and sometimes about mentioning how they played a role in the process.

\[ D_d \rightarrow d_{d1} \mid d_{d2} \mid d_{d3} \mid \ldots \mid d_{dn} \]  

(6)

At this stage, we define an abbreviation as a condensed version of a word or phrase. A shorter form of a term is called an abbreviation (such as Dr. stands for Doctor or Prof. means professor).

\[ A_{br} \rightarrow a_{br1} \mid a_{br2} \mid a_{br3} \mid \ldots \mid a_{brn} \]  

(7)

The table of contents contains a list of your dissertation’s chapters and important sections, as well as their page numbers. The quality of a thesis structure is determined by the contents page, which should be clear and well-formatted. Hence, we generated a production rule for this section as follow.

\[ T_c \rightarrow \text{template-shuffle}(C_{NT}) \]  

(9)

\[ L_{Fig} \rightarrow \text{template-shuffle}(L_{Fig}) \]  

(10)

\[ L_{Agol} \rightarrow \text{template-shuffle}(L_{Agol}) \]  

(11)

\[ L_{Table} \rightarrow \text{template-shuffle}(L_{Table}) \]  

(12)

In the above grammar, \( C_{NT} \) stands for content, \( L_{Fig} \) represent list of figures, \( L_{Agol} \) means list of algorithms and \( L_{Table} \) implies the list of tables for the design of table of content. A research paper or thesis abstract is a succinct description of the work. It’s a well-written, single-spaced paragraph of roughly 250 words. It highlights the main subject areas, the objective of the study, its importance or applicability, and the main results. Having said that, we created a set of production rules at this stage, which are detailed as follow.

\[ A_{bs} \rightarrow \text{template-shuffle}(A_{bs}) \]  

(13)

The manner which an academic scholar presents himself or herself to the readers is in the introduction of a thesis. A preface is a brief introduction written by the thesis author to convey the thesis’s main topic and writing experience with the audience. The production rule for this phase is written as follow.

\[ P_{face} \rightarrow \text{template-shuffle}(P_{face}) \]  

(14)
A thesis is made up of several chapters that together form a well-supported and convincing answer to the research topic. An introduction, a background of the study, research approach, a report and discussion of results, and a conclusion are typical chapters in a thesis. However, the production rules for this section are the following.

\[
C_{set} \rightarrow (C_1 \ C_2 \ \ldots \ C_5) \ | \ (C_6 \ | \ \lambda) \tag{15}
\]

\[
C_1 \rightarrow \text{gen-introduction(dept)} \tag{16}
\]

\[
C_2 \rightarrow \text{gen-literature review(dept)} \tag{17}
\]

\[
C_3 \rightarrow \text{gen-methods(dept)} \tag{18}
\]

\[
C_4 \rightarrow \text{gen-results(dept)} \tag{19}
\]

\[
C_5 \rightarrow \text{gen-conclusion(dept)} \tag{20}
\]

The definition of \(C_1\) means number of chapters for examples: chapter 1, chapter 2 and the rest of the chapters. Appendices should always come after the references or bibliography, and it provides additional information to the primary thesis. The production rule is written as:

\[
A_{append} \rightarrow A_{append} \ | \ A_{pi} \ | \ \lambda \tag{21}
\]

Referencing is the process of acknowledging the source of information utilized in a piece of writing by citing it. Referencing is significant for a variety of reasons, including but not limited to the following: It permits acknowledgment of other people’s perspectives, ideas, theories, and innovations. It assists readers comprehend what swayed the writer’s thinking and how their ideas were formulated. The production rule for referencing is stated as follow.

\[
R_{f} \rightarrow \text{fn \ (style)} \tag{22}
\]

4.2. Sketching Algorithm

In this section, we present an algorithm for sketching research theses, leveraging the CFG rules outlined in Section 4.1. By applying these rules, we propose a systematic approach that facilitates the creation of well-structured and cohesive research thesis outlines. The propose sketching algorithm, as shown in Algorithm 1.
Algorithm 1. sketch theses

Algorithm 1 refers to the procedure for generating sketch theses. A function is created to generate the sketched thesis, multiple text files are used to store information and serves as input for the Sketch Thesis. The Text files are stored in an array, and they are accessed throughout the thesis sketching process. The output LaTeX file is created, and this file displays the information generated by the Sketch Thesis function, and to access the information in the input text files, a loop is used.

An integer value is created to keep track of the amount of times the loop has occurred and this integer is used to format the LaTeX file. The information that is stored on the main text file, that contains the general section, is read and stored.
onto a string variable. Before the loop starts, latex packages and information about the Title, Author and date is created that is displayed as part of the heading for the output latex file. The main file is accessed at this point and the loop is created.

At this stage of the algorithm, a conditional statement is first used to verify that the line in the input file is not empty so that empty lines are skipped. If an empty line is identified, then the loop searches the next line without executing any command and if the line contains information, then the options in the line are separated so that they will be treated as different items and these items are stored in an array and a random item is selected. Once the item has been selected, the item is then concatenated with a section keyword that is used to create a section in the latex file. The item is written to an output latex file as a section, a newline is added to the section so that the sections do not write onto the same line and a second loop is created within the first loop.

Empty lines are accounted for with the conditional statement, as it was done in the first loop. Lines of information are read for the files that contain the subsection, the lines are separated and randomly selected the same way they were in the first loop. The selected is concatenated with a subsection keyword that is used to create a subsection in the latex file, a newline is added to the subsection and the second completes its first execution.

The integer value is incremented by 1 and once the integer has reached a value of 4, a new page is created to store the rest. The first completes its execution, the first and second loop will continue executing until all the information has been read from all the input files and written to all the output files. When both loops have completed all their executions, a closing keyword statement is added to the output latex file that ends the latex document. An instance of the process class is created to open the Latex editor as well as the latex output file, and the Latex editor and the latex output file is opened to display the final output information.

In the next section, we present the results from the generation of thesis sketches algorithm presented in Section 4.2.

4.3. Implementation

Algorithm in Section 4.2 was implemented in .Net Framework version 4.8.1, using C# as the programming language. Functions were dedicated to creating Tex files and background workers (or multi-threading mechanisms within the language libraries) were used to ensure that the speed of creating synthesised files on disk did not result in deadlock. The generated Tex files are all compilable as they are concatenations made from verified formats with LaTeX file standards,
suitable for the PDFLaTeX compiler. The results are presented in the next section. The generated thesis sketches from the earlier described implementation. Figure 2 shows two generated sketches with variability observable on the lines of the sections of these sketches, while Figure 3 shows a windows folder that contains 40 (out of 10,000) iterations that contain generated thesis sketches. More results of 10,000 generated thesis sketches can be found at: www.tinyurl.com/thesisssketches.

The evaluation of the generated thesis sketches was conducted in two-fold: (1) theoretically by validating that the grammar rules will always provide valid LaTeX documents that are valid sketches, (2) the automatic compilation of the generated sketches using the LaTeX compiler, and (3) by manually inspecting the generated files to see if these are valid sketches.

\begin{verbatim}
\documentclass[article]{article}
\usepackage[utf8]{inputenc}
\title{Thesis Sketch --- Title Here}
\author{Babajide Adeyemi}
\date{\today}
\begin{document}
\maketitle

\section{Introduction}
\subsection{Introduction}
\subsection{Concepts Central to the Study}
\subsection{Motivation of the Study}
\subsection{Research Techniques}
\subsection{Research Significance}
\subsection{Research Question}
\subsection{Document Structure}
\subsection{Conclusion}
\subsection{Related Work}
\subsection{Introduction}
\subsection{Key Terms Definition}
\subsection{Information Systems}
\subsection{Summary}
\section{Research Methodology}
\subsection{Research Design}
\subsection{Introduction}
\subsection{Research Procedure Outline}
\subsection{Research Methods}
\subsection{Research Paradigms}
\subsection{Population of the Study}
\subsection{Sampling Technique and Sampling Frame}
\subsection{Data Collection Process}
\subsection{Data Analysis Method}
\subsection{Summary}
\section{Conclusion}
\section{Findings}
\begin{verbatim}
\documentclass[article]{article}
\usepackage[utf8]{inputenc}
\title{Thesis Sketch --- Title Here}
\author{Babajide Adeyemi}
\date{\today}
\begin{document}
\maketitle

\section{Introduction and Background}
\subsection{Introduction}
\subsection{Background Information}
\subsection{Motivation of the Study}
\subsection{Research Process}
\subsection{Goals of the Research}
\subsection{Research Question}
\subsection{Layout of the Study}
\subsection{Conclusion}
\subsection{Review of Related Literature}
\subsection{Related Work}
\subsection{Introduction}
\subsection{Key Terms Definition}
\subsection{Information Systems}
\subsection{Summary}
\section{Research Methodology}
\subsection{Research Design}
\subsection{Introduction}
\subsection{Research Procedure Outline}
\subsection{Research Methods}
\subsection{Research Paradigms}
\subsection{Population of the Study}
\subsection{Sampling Method and Size}
\subsection{Data Collection Process}
\subsection{Data Analysis Method}
\subsection{Summary}
\section{Conclusion}
\section{Implementation and Results}
\end{verbatim}
\end{document}
\end{verbatim}

Figure 2: Two randomly generated thesis sketches (presented as LaTeX source) from Algorithm 3.2.
5. CONCLUSION

In this paper, we outlined and discussed what a thesis means. The broad literature review of this study allowed us to understand the challenges, significance, and technology used in generating thesis sketches. We discovered that most students struggle with sketching thesis and design science research was used in the methodology phase to present a newly designed artefact for syntactic generation of research thesis sketches across disciplines using formal grammars. We defined formal grammar for the generation of the language of sketches and established the foundations. We presented an algorithm for the sketching of research theses, based on the context-free grammar (CFG) rules. This algorithm produced several sketches, some presented in this paper and others shared in a link to an online repository. These sketches are expected to assist students with a starting point when writing their theses. We have ongoing projects that extend the grammar presented in this work to the syntactic generation of details of sections and subsections of theses. We are also wrapping this tool into an application that can be distributed for students’ use.

ACKNOWLEDGEMENT

The second Author would like to acknowledge the efforts of his research assistant, Tadi Chingwe, during this project.
REFERENCES


Social Sciences, vol. 23, no. 4, pp. 593-603, 2011.


[47] A. Solar-Lezama, "The sketching approach to program synthesis," In


[58] A. M. Cox, "Exploring the impact of Artificial Intelligence and robots on


[70] S. Cheema and J. LaViola, "PhysicsBook: a sketch-based interface for


