Structural Vetting of Academic Proposals

Opeoluwa Iwashokun[1]
Department of Applied Information Systems
College of Business and Economics, University of Johannesburg
Johannesburg, South Africa

Abejide Ade-Ibijola[2]
Research Group on Data, Artificial
Intelligence, and Innovations for Digital Transformation
Johannesburg Business School, University of Johannesburg
Johannesburg, South Africa

***Abstract*—Increasing postgraduate enrollments gives rise to many proposal documents required for vetting and human supervision. Reading and comprehension of large documents is a boring and somewhat difficult task for humans which can be delegated to machines. One way of assisting supervisors with this routine screening of academic proposals is to provide an artificial intelligent (AI) tool for initial *structural vetting* — checking if sections of proposals are complete and appear where they are supposed to. Natural Language Processing (NLP) techniques in AI for document vetting has been applied in legal and financial domains. However, in academia, available tools only perform tasks such as checking proposals for plagiarism, spellings or grammar, word editing, and not structural vetting of academic proposal. This paper presents a tool named Auto-proofreader that attempts to perform the task of structural document review of proposals on behalf of the human expert using formal techniques and document structure understanding hinged on context free grammar rules (CFGs). The experimental results on a corpus of 20 academic proposals using confusion matrix technique for evaluation gives an overall of 87% accuracy. This tool is expected to be a useful aid in postgraduate supervision for vetting students' academic proposals.**

***Keywords*—Document structure; context free grammar; postgraduate supervision; artificial intelligence; natural language processing**

I. Introduction

Natural language processing (NLP) has been applied in document vetting across domains such as legal practice [1], [2], [3]. These areas of research in extracting information, text summarising and text vetting of documents is a difficult task for humans when several pages of a document or many documents are involved or short time is available. Daramola [4] observed that postgraduate supervisors are faced with the salient task of vetting many students' proposals to conform to certain academic standards, amidst other key roles in the University and must find the right balance for work and effectiveness. In recent times, South Africa experienced an increasing headcount in the number of post-graduate students' enrollment, impacting on the process of screening of submitted proposals [5]. There are yet many sub-standard proposals submissions by postgraduate student novice writers which is impacting negatively on the screening, feedback and vetting time of the assigned supervisors [4], [6]. The timeliness and effectiveness of screening these proposals can be assisted by an intelligent tool performing the specific task of vetting proposals based on prescribed proposal format guideline constructed as context free grammar (CFG) rules.

A. Challenges Facing Supervisors in the Process of Vetting Academic Proposals

Supervision is a critical component of postgraduate studies and many supervisors continue to grapple with promoting research ideas in students' academic writing and students' writing standards [7]; especially when they are still novice writers and in the first year of research writing [4]. There is an increasing pressure on postgraduate supervision as the number of enrollments continue to increase exponentially and universities are under more pressure for more quality research output [8]. Supervisors continue to assess the preparedness and candidature of student enrollees with the vetting outcomes of their proposals. They observed that students often submit proposals that are unacceptable (often too long or too short, poorly written or not well organised and often *missing critical proposal sections*). This is attributed to students not reading or understanding the guideline/instruction format or other peculiar reasons [4], [7]. Supervisors are expected to play a multi-faceted role in their discharge of duties. They are stretched thin as they support students' handling responsibilities and other academic responsibilities for the university [4], [9]. A better approach is to reduce drudgery by introducing technological tools for replacing traditional approach in supervising students [10] and specifically for vetting the structure of academic proposals.

B. What has been done?

1) Support for Students to improve Writings before Proposal Submission: There are lots of approaches and support programmes for scholars to develop writing techniques in the best way possible. There are provisions for writing workshops [11], writing groups [11], informal and online support services [11], mentoring programmes [12], writing editors, various grammar and spell-check productivity tools [13], [14]. These were supports for scholars to improve their writings which is as important as support for supervision. The findings of a review of students and supervisors by Hey-Cunningham *et al.* [10] explained that providing innovative solutions to improve the feedback mechanism in supervision is very important.

2) Support Toolkit for Supervisors: A survey conducted by Hey-Cunningham *et al.* [10] expresses that the theme common to many supervisors was the need for enhancements of supervisory approaches to academic writing standards for which the authors proposed innovations for a timely and effective feedback in supervision. Supervisors engage many generalised tools (e.g. plagiarism checker, editor review tools and grammar cum spell check error tools) to provide revision

TABLE I. CATEGORY OF EXISTING TOOLS FOR POSTGRADUATE SUPERVISION

Supervision tool	Functions
Plagiarism checkers e.g. Turnitin	It performs only text extraction for similarity check index against other literature. It is not a self-check document tool.
Feedback review tools e.g. Microsoft's word reviewer	It can assist a human reviewer to perform a document self-check, but it can be time-consuming especially when reviewing proposals of many students.
Grammar and spell check tools e.g. Grammarly, document proWriting aid, etc.	It can only assist the human reviewer to proofread the content of a proposal but not the structure or format layout of the proposal document the proposal document.

supports when screening submitted proposals of students. The perceived functions of these tools are laid out in Table I

C. The New Norm of Technological Aids in Postgraduate Supervision

Productivity tools are replacing and improving traditional and unconventional methods of performing higher education supervision [15], [16]. The effect of COVID-19 pandemic has even made it more necessary. For instance, one-to-one supervision meetings now commonly take place virtually and proposal document review process are more commonly done with various online collaborative productivity tools [17], [18], [16]. Productivity tools has been very effective in engaging dialogue between supervisors and their students, but has been proven to be time-consuming and tiresome when examining a large batch of students' proposals on a computer [19]. Many grammar and spell-check tools are also used by students and supervisors for fine-tuning grammar and editing spelling errors that may not be easily tracked by the eyes of a human reviewer. Popular examples are Grammarly, Microsoft word spell check, pro-Writing aid and language tool.

D. Gap

Researchers have used various forms of text processing technique to automatically extract and analyse documents such as business documents [20], clinical notes [21], legal documents [2], [22] and so on. NLP techniques have been used to perform text information extraction [23], named entity recognition [24], language to SQL translator [25], [26], summarisation [27], classification and examination of other textual contents such as CVs [28], invoices [20] and social media texts [29]. These NLP techniques and others have been largely used around the text content of a document, and sometimes short-text based documents. We considered document understanding an AI task and we have found no tool for vetting of large-content academic text based document such as proposals or similar academic writings. The question then is: *"how can we aid the vetting of academic proposals using existing NLP techniques?"*

E. Proposed Solution

In this paper, we have designed an approach which breaks up a proposal document into `tokens` that are basic recognisable `symbolic parts` of an academic proposal document. We also determine if the input proposal document parts satisfies a valid structure, defined in a proposal guide, by constructing a CFG for the acceptance of a valid proposal structure and REGEX to accept valid terminal symbols. A simple parse tree representation for a proposal document is given in Fig. 1. The document itself as the root node of the parse tree contains Section parts of a proposal document. The algorithm for document parsing is implemented using an existing PDF library named `iTextSharp` Java PDF library.

Similar technique was used for CV slicing [28], metadata extraction of PDF books [30], meta-analysis of clinical notes [21], business invoice document processing [20] and legal documents [27].

F. Contribution to Knowledge

This paper contributes to knowledge in the following ways:

1) the production of a simplistic CFG for recognising an academic proposal structure,
2) design of an approach for the automatic discovery of the document's structure of academic proposals, useful for other large text based document,
3) it promotes further research on automatic slicing of text documents using grammar based rules,
4) design of an algorithm for end-to-end automatic vetting of the logical sections and elements (i.e. structure) of an academic proposal and
5) it describes the implementation and evaluation of a software tool for vetting of academic proposals.

The rest of this document is organised as follows. Section II explains some background to this research problem and related research efforts in text documents comprehension. Section III contains the design concept for this work, while Section IV shows the result of the test of the software tool on proposal documents and the output results generated. An evaluation of these results is done in Section V, while we conclude and state further future work in Section VI.

II. BACKGROUND AND RELATED WORK

There is no directly related research work in vetting academic proposals. This section only explains various related NLP techniques and examples in past research for information extraction and summarisation of documents in general. Then highlights on the formal grammar approaches that has been applied by various researchers for automatic document discovery of large text documents. bluelightAnd lastly discusses some existing proofreading tools.

A. Automatic Document Discovery

NLP has gained a lot of popularity using computational methods to process spoken or written form of text by humans, with applied use cases in various business and enterprise based applications [31]. This field in artificial intelligence (AI) is gaining a lot of momentum and one key applied use is in the area of text processing for information extraction, summarisation or classification [23]. It is used in legal field for similar case matching and text summarisation of legal documents [2]. NLP techniques has been applied successfully for automatic comprehension of clinical notes [21], slicing and information extraction from curriculum vitae (i.e. CVs) [28], understanding

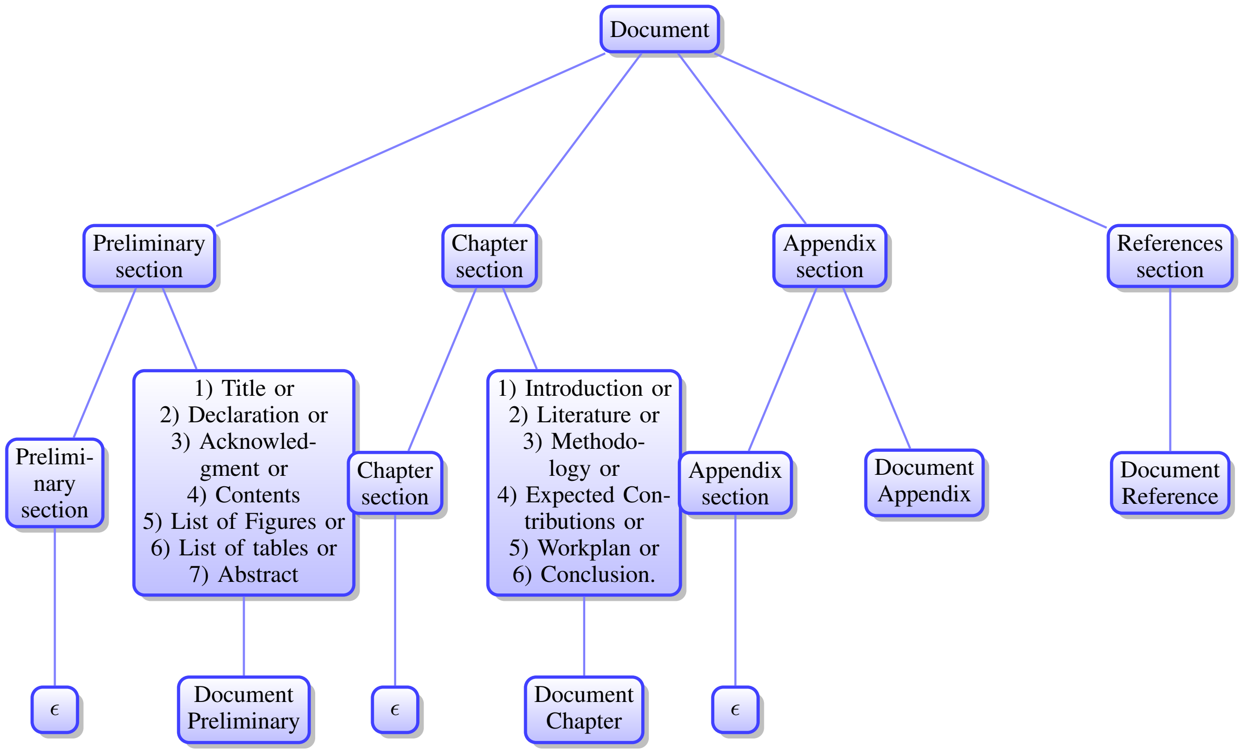

Document
Preliminary section
Chapter section
Appendix section
References section
Prelimi-nary section
1) Title or 2) Declaration or 3) Acknowled-gment or 4) Contents 5) List of Figures or 6) List of tables or 7) Abstract
Chapter section
1) Introduction or 2) Literature or 3) Methodo-logy or 4) Expected Con-tributions or 5) Workplan or 6) Conclusion.
Appendix section
Document Appendix
Document Reference
ϵ
Document Preliminary
ϵ
Document Chapter
ϵ

Fig. 1. Proposal Document Parse Tree.

and improvement of requirements documents [32], extracting parts of business (invoice) documents [20], finance chat messaging [33] and so on.

B. Related Work

1) Document Vetting of Legal Proceedings Document: Legal expert systems that use NLP for relevant information extraction and summarization for legal consumption now exist. The computational language processing of legal text documents has been very useful in drafting and analysing legal documents, classifying documents based on relevance to legal case and legal documents discovery and legal citations extractions [2], [3], [27]. More recently, a process tagged "Technology Assisted Review" is associated with the legal profession focused on categorizing legal documents and files based on relevance to case or legal information required [3]. It has become popular and replaced manual review of documents in the legal profession with a more effective automated approach.

2) Automatic Extraction from Business Documents: Glenda and Shilpa [1], implemented an NLP based document vetting process for Banks thereby reducing staff work load and increasing efficiency. NLP techniques have been successfully applied to processing business invoice documents [20], IT Projects system requirements document vetting [32] and automatic comprehension of business finance chats [33]. The applied use of NLP for document processing in finance has become very rapid and important.

3) Automatic Comprehension of Clinical Notes: Modern medicine has embraced NLP techniques for systematic reviews of several clinical trials with great measures of success, known as NLP-enhanced clinical trials research [34]. In a similar vein, a simple NLP translator was designed to decrypt clinical notes, creating friendly user plain texts from complex medical reports [21].

4) Automatic Slicing of CVs: Curriculum Vitae (CV) and resumes are structured documents that contain certain key elements information such as work experiences that talent-hunt specialist usually look out for in CVs during a job advert placement. Emil St. *et al.* [35] applied NLP text extraction techniques on several CV documents to determine candidate professional qualifications, which is useful for review and ease of vetting the relevance of the CV to the role advertised . At another instance, an NLP-based tool was designed to extract the logical sections of CVs using a set of CFG rules [28].

C. Some Background on Document Structure Parsing

The order of arrangement of a document largely explains its structure and can be described using a tree model or an hierarchical pattern [36], [37]. Anjewierden's [36] approach extracted the characteristics of text fonts and lines of text to discover the structure or document style by clustering tokens (in this case, a non-space characters or strings of the English alphabet identified in PDF document) into document elements identifiable on a document page. This was achieved using the characteristics of the token's dimensional co-ordinates on the document, which are upper left position co-ordinates of token and bottom right position co-ordinates on a document's page. These characteristics are syntactically analysed using rules by categorizing text into chunks of meaningful elements of the document that reveals the document structure. According to Anjewierden [36], a set of document object texts is sorted on

TABLE II. Existing Proofreading Tools and Techniques

	Tool	Main function	Technique
1.	Grammarly [38], [39], [40], [41]	Grammar, style and spell check	It combines rules, patterns and AI techniques in machine learning and deep learning
2.	Pro-write Aid [42]	Grammar, style and spell check	AI techniques in machine learning techniques
3.	Typely	Grammar, style and spell check	AI techniques in machine learning techniques aimed at high precision
4.	Custom-built proof-reader tool e.g Chinese text automatic proofreader [43]	Grammar, style and spell check	entity recognition and Knowledge graph

their co-ordinate positions on a document page, then parsed through a set of shallow grammars to detect specific elements of the document's logical structure.

D. Discussion of Some Existing Proofreading Tools

There are quite a number of existing proofreader tools available as add-on or online tools for general writing problems on grammars, spellings and styles. Grammarly is quite popular for automatic proofreading in academic writing [38], [39], [40], [41]. Karyuatry and Rizqan [40] explains that Grammarly provides feedback on grammar errors and styling mistakes based on similarity patterns in real time. It is mainly available as a web based tool and built on AI system of a variety of natural language processing (NLP) statistic and machine learning based techniques. Pro-writing Aid is a similar proofreading tool for corrections in punctuation, grammar and style [42]. It is also available online and provides teachers and or students with feedback reports, with which they can improve writing. Proofreaders generally rely on AI techniques to come up with suggested corrections as feedback. Table II summarizes the functions and techniques of some existing tools. The table shows that existing proofreader tools have created a niche for improving writing by providing the feedback on grammar, punctuation and style. However, the novelty of our proposed tool seeks to provide feedback based on the organization (i.e. structure) and sections of an academic research proposal.

E. Definition of Terms

Definition 1: The document logical structure [36] is the layout for its constituents consisting of paragraphs, item lists, sections, tables etc which is easily identified by humans but has to be discovered computationally.

Definition 2: The document (logical) elements [36] is the proper thematic units that can be annotated and form part of the document logical structure, it is also referred to as document text segment. We can identify headings, sub-headings, paragraphs, tables, lists of information, page numbers and headers or footers as structural elements of a document.

Definition 3: A context free grammar (CFG) [44], [45] is a given grammar G defined by a four tuple given as $G = (N, \Sigma, P, S)$ where:

1) N is a set of non-terminals,
2) Σ is a finite set of terminal symbols, which are the nodes of the grammar (such as symbols, alphabets and numbers),
3) P is a set of Productions and
4) S is a non-terminal start symbol.

.

Definition 4: Regular languages (RL) and regular expressions (REGEX) [44], [45] denotes regular languages. Both are represented by formulas involving the operations of concatenation, union and Kleene star. Regular languages are such languages that can be accepted by a finite automation. In formal terms, a regular language (L) for a given grammar over an alphabet set is Σ defined as any of the following:

1) a singleton language a where $a \in \Sigma$,
2) if A is a regular language, then kleene star of the language (A^*) is a regular language,
3) the empty set of a language A, denoted as $\{\}$ or $\{\lambda\}$ and
4) if A and B are regular languages, then $A \cup B$ the and $A \cap B$ are also regular languages.

III. Design

Fig. 2 describes the steps for structural vetting of academic proposals. The design presents an end-to-end technique of information extraction and vetting of an academic proposal.

A. Overview

An input proposal document is parsed (see Fig. 1 for parse tree structure), by an algorithm into meaningful document symbolic parts consisting of document's sections, chapters and pages. The leaf nodes of the parse tree structure are the document's pages symbols. In our implementation, a valid document page symbol acceptance is further determined by a REGEX parser for recognising and determining the *elements* within the symbolic string. Such elements that may be contained within a proposal document's page symbolic string are document's title, document's author, supervisor names, proposal date, section title, paragraph's heading, paragraph, figure and so on. *See Fig. 3 and 4 for the description of our proposal document elements*. The elements of a proposal document as presented in these figures are the building blocks for a proposal document organization. They appear in a pre-defined order for an organized paper. The structure of a proposal document is accepted or rejected after full parsing of the input proposal document into parts that validates its structure (i.e. arrangement). The correctness of the elements' structure is based on the CFG Production rules defined and successful passing by REGEX matching of elements contained in the document's parts.

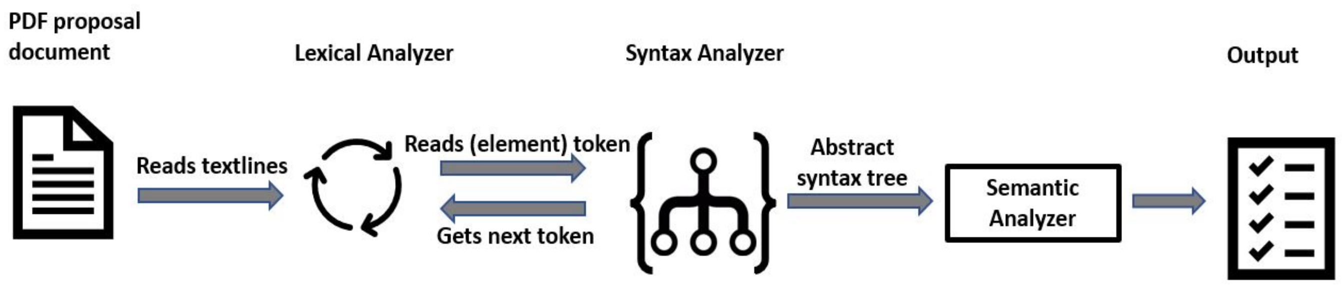


Fig. 2. Proposal Structural Vetting Design

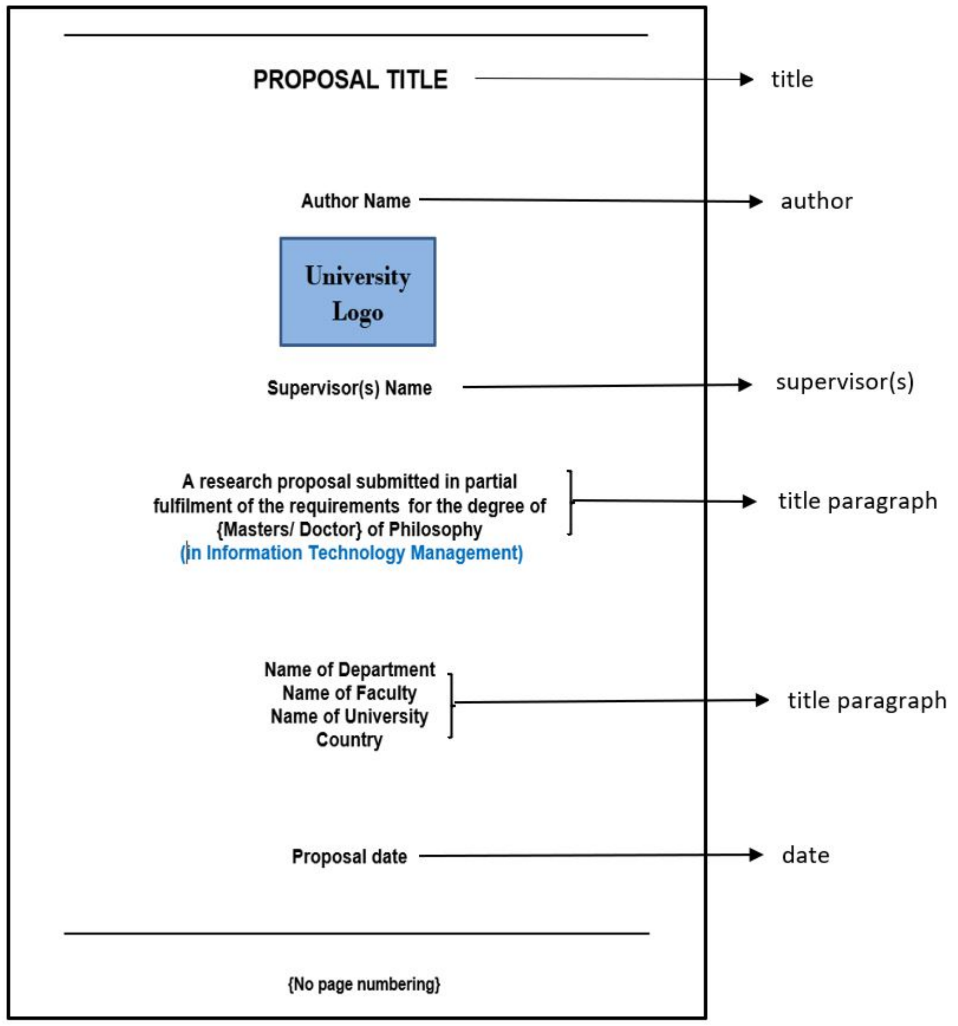


Fig. 3. Structure of Title Page

B. Lexical Analysis of PDF Proposal Document

The input document is tokenized into strings of texts identified by text line sequence on the document page. The text's feature (appearance) on the page described in the Table III is used to categorize text and "line of texts" of a page into meaningful document elements. For instance, a proposal document title is made up of line(s) of text in bold and appears on the first page of a proposal document. The input proposal is analyzed into all the elements that may be contained in a document and as described in Fig. 3 and 4.

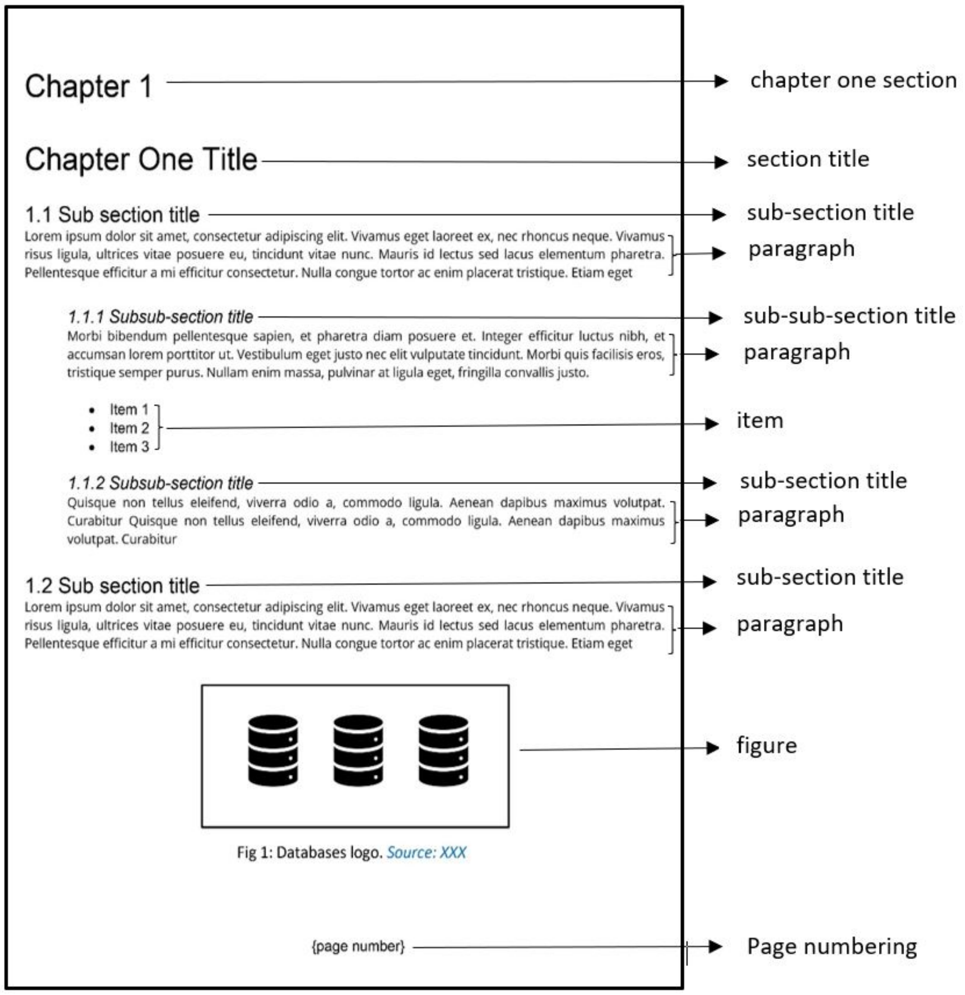


Fig. 4. Structure of Chapter Pages

TABLE III. Document Component/Element List of Characteristics

Characteristics	Description
Font bold	the thickness of the text
Font size	text height
Line top spacing	height of space between the current line and the previous line.
Line bottom spacing	height of space between the current line and the next line.
Line left align	width of space between the document layout left indent and the line left co-ordinate position.
Line right align	width of space between the document layout right indent and the line right co-ordinate position
Line align type	determined as left, center or right based on the left and right indent

C. Description of Proposal Document Abstract Parse Tree using CFG

To explain the parse tree structure and its nodes, we implement a CFG with a four-tuple as given below:

$$G = (N, \Sigma, P, S) \tag{1}$$

with the following representations:

1) Set of non-terminal variables (N): a collection of all the non-terminal symbols for the production which represents proposal document's section. The symbols are described as Preliminary Section (S_p), Chapter Section (S_c), Appendix-Section (S_a) and References-Section(S_r). $N = \{S_p, S_c, S_a, S_r\}$.
2) Finite set of terminal symbols (Σ): a set of all symbols representing the proposal document terminals. These terminals are symbolic of proposal document pages and described as Title Page (t_p), Declaration Page (d_p), Acknowledgment Page (a_p), Content Page (c_p), List of Figures (l_f), List of Tables (l_t), Abstract Page (a_{bp}), Introduction Chapter Page (c_{intro}), Literature Chapter Page (c_{lit}), Methodology Chapter Page (c_{method}), Expected Contribution Page (c_{contr}), Workplan Chapter Page (c_{plan}), Conclusion Chapter Page (c_{concl}), Appendix Page (p_a) and Reference Page (p_r)

3) Set of Productions (P): are the Production rules (P) which defines the structure of the elements in the pages of the proposal document.

$$S_{doc} \longrightarrow S_p S_c S_a S_r \tag{2}$$

$$S_p \longrightarrow t_p(d_p|\lambda)(a_p|\lambda)c_p(l_f|\lambda)(l_t|\lambda)a_{bp} \tag{3}$$

$$S_c \longrightarrow c_{intro}{\cdot}c_{lit}{\cdot}c_{method}{\cdot}(c_{contr}|\lambda){\cdot}(c_{plan}|\lambda){\cdot}c_{concl} \tag{4}$$

$$S_a \longrightarrow p_a|p_a S_a|\lambda \tag{5}$$

$$S_r \longrightarrow p_r \tag{6}$$

4) Non-terminal start symbol (S): The start symbol for this grammar is the input document proposal, denoted by symbol S_{doc} .

D. REGEX for Terminal Symbols Matching

The terminal symbols of the CFG tuple are further broken down into regular expressions for recognising the page elements implemented by the symbols. The terminal symbols can be substituted with the following symbols in REGEX below.

$$t_p = e_t e_a e_s e_{tp}^{1,2} e_d \tag{7}$$

$$d_p = e_{st} e_p^{+} e_n \tag{8}$$

$$a_p = e_{st} e_p^{+} e_n \tag{9}$$

$$c_p = e_{st} e_{ta} e_n \tag{10}$$

$$l_f = e_{st}(e_i|e_{ta})e_n \tag{11}$$

$$l_t = e_{st}(e_i|e_{ta})e_n \tag{12}$$

$$a_{bp} = e_{st} e_p^{+} e_n \tag{13}$$

$$c_{intro} = e_{st}((e_p|e_{sst}|e_f|e_i|e_{ta})^{+} e_n)^{+} \tag{14}$$

$$c_{lit} = e_{st}((e_p|e_{sst}|e_f|e_i|e_{ta})^{+} e_n)^{+} \tag{15}$$

$$c_{method} = e_{st}((e_p|e_{sst}|e_f|e_i|e_{ta})^{+} e_n)^{+} \tag{16}$$

$$c_{contr} = e_{st}((e_p|e_{sst}|e_f|e_i|e_{ta})^{+} e_n)^{+} \tag{17}$$

$$c_{plan} = e_{st}((e_p|e_{sst}|e_f|e_i|e_{ta})^{+} e_n)^{+} \tag{18}$$

$$c_{concl} = e_{st}((e_p|e_{sst}|e_f|e_i|e_{ta})^{+} e_n)^{+} \tag{19}$$

$$p_a = e_{st} e_f^{+} e_n \tag{20}$$

$$p_r = e_{st} e_r^{+} e_n \tag{21}$$

TABLE IV. REGULAR EXPRESSIONS FOR DOCUMENT ELEMENTS PATTERN MATCHING

SN	Document Elements	Description	Example
1.	Title (e_t)	(\w\s)$^{+}$	Structural vetting …
2.	Author (e_a)	(By:|By)?(\s \w)$^{+}$	Joe Smith
3.	Supervisor (e_s)	(Supervisor:|$Supervisors$:)? (\s\w)$^{+}$	Prof Tim
4.	Title-paragraph (e_{tp})	((\w \s)$^{+}$ [\n])$^{+}$	This research paper …
5.	Section-title (e_{st})	(\s \w)$^{+}$	Introduction
6.	Sub-section-title (e_{sst})	\d^{+}.\d^{+} (\s \w)$^{+}$	1.1 Background
7.	Paragraph (e_p)	((\w \s)$^{+}$ [\n])$^{+}$	The advent of ICT …
8.	Page Number (e_n)	\d^{+}	2
9.	Date (e_d)	{Date API}	3rd Dec 2021
10.	Ref item (e_r)	{Reference API}	K.Van (2020), "The choice" …
11.	Figure (e_f)	{Image}	Image
12.	Item list (e_i)	((\w \s)$^{+}$[\n])$^{+}$	1. Name 2. Subject
13.	Table (e_{ta})	((\w \s)$^{+}$[\n])$^{+}$	Table

The symbols \w, \s, \n, \d matches single word, single space, a newline character and single digit respectively

E. REGEX for Document Elements Matching

The elements contained in a proposal document's pages can be matched with corresponding regular expression defined in the Table IV. The regular expressions stated are used to represent the language of the structure in English alphabet strings for every element that may be contained in a proposal document. We define a language acceptor algorithm which accepts or rejects the element token based on the matching regular expression. The language acceptor determines if the element is well-formed and which element class it belongs to. The proposal document is vetted structurally correct if all the parsed elements are accepted by the language acceptor.

F. Document Parsing

This section explains possible derivations of a proposal document given the grammar defined and its set of Production rules. Given the input proposal document which is of the start symbol S as given below, then the document can be parsed as follows:

$$S_{doc} \Longrightarrow S_p \cdot S_c \cdot S_a \cdot S_r \quad (rule\ 2) \tag{22}$$

A proposal document can be made up of the following section parts: Preliminary Section(S_p), Chapter Section(S_c), Appendix Section(S_a), Reference Section(S_r).

$$\Longrightarrow t_p \cdot d_p \cdot c_p \cdot a_{bp} \cdot S_c \cdot S_a \cdot S_r \quad (rule\ 3) \tag{23}$$

The Preliminary Section can be made up of Title Page (t_p), Declaration Page (d_p), Contents Page (c_p) and Abstract Page (a_{bp})

$$\Longrightarrow t_p{\cdot}d_p{\cdot}c_p{\cdot}a_{bp}{\cdot}c_{intro}{\cdot}c_{lit}{\cdot}c_{method}{\cdot}c_{plan}{\cdot}c_{concl}{\cdot}S_a{\cdot}S_r \quad (rule\ 4) \tag{24}$$

The Chapter Section can be made up of Introduction Chapter (c_{intro}), Literature Review Chapter (c_{lit}), Methodology Chapter (c_{method}), Workplan Chapter (c_{plan}) and the Conclusion Chapter (c_{concl}).

$$\Longrightarrow t_p{\cdot}d_p{\cdot}c_p{\cdot}a_{bp}{\cdot}c_{intro}{\cdot}c_{lit}{\cdot}c_{method}{\cdot}c_{plan}{\cdot}c_{concl}{\cdot}S_r \quad (rule\ 5) \tag{25}$$

The Proposal document may not contain an Appendix Section.

$$\Longrightarrow t_p{\cdot}d_p{\cdot}c_p{\cdot}a_{bp}{\cdot}c_{intro}{\cdot}c_{lit}{\cdot}c_{method}{\cdot}c_{plan}{\cdot}c_{concl}{\cdot}p_r \quad (rule\ 6) \tag{26}$$

The Derivation 26 is an instance of a valid proposal structure and has been successfully parsed by the four-tuple grammar defined for the structural vetting of a proposal document. The expressions contained in Derivations 22 to 26 shows the complete parsing. Further understanding of the document's structure is done by recognising the document *elements* contained in each symbolic string of the language generated by the parser, refer to Derivations 7 to 21. Elements are also matched by their corresponding REGEX description in Table IV

G. Algorithms

This algorithm for the automatic structural vetting of an academic proposal document takes as input, the proposal document, and outputs the document with highlights of any structural defects that may have been picked. We present the algorithm as given below:

Algorithm 1: Structural Vetting Algorithm

```
Data: PDF proposal document, $S_{doc}$
Result: Highlighted PDF proposal document, $S_{hl}$
1  Parse input document into document parts using
     PDFLibrary.Extract($S_{doc}$)
2  if ValidParse(Document parts) then
3      if REGEX_Match(Elements contained in
         Document Parts) then
4          if REGEX_Match(words contained in
             Document Elements) then
5              PROCESS COMPLETE: Document
                 verified successful
6              Display success message
7          end
8          else
9              ERROR DETECTED: Highlight error on
                 Proposal
10         end
11     end
12     else
13         ERROR DETECTED: Highlight error on
             Proposal
14     end
15 end
16 else
17     ERROR DETECTED: Highlight error on Proposal
18 end
19 Document_file_location ⟵
     Save_Highlighted_Document_toDisk($S_{hl}$)
20 return Return $S_{hl}$
```

IV. Implementation and Results

The algorithms presented in this paper were implemented using C sharp Microsoft's .Net library. The software embeds an iTextSharp version 7.0 library used for PDF document tokenization and manipulation of text. The Auto-proofreader tool interface displaying its menu controls is presented in Fig. 5.

Table V shows the results from the implementation of a corpus of 20 academic proposals. All the results vary in the degree of the number of error or success feedback reported by the tool. All the elements in each document have been parsed against defined rules and REGEX patterns based on a given instruction format guide.

A. Description of Dataset

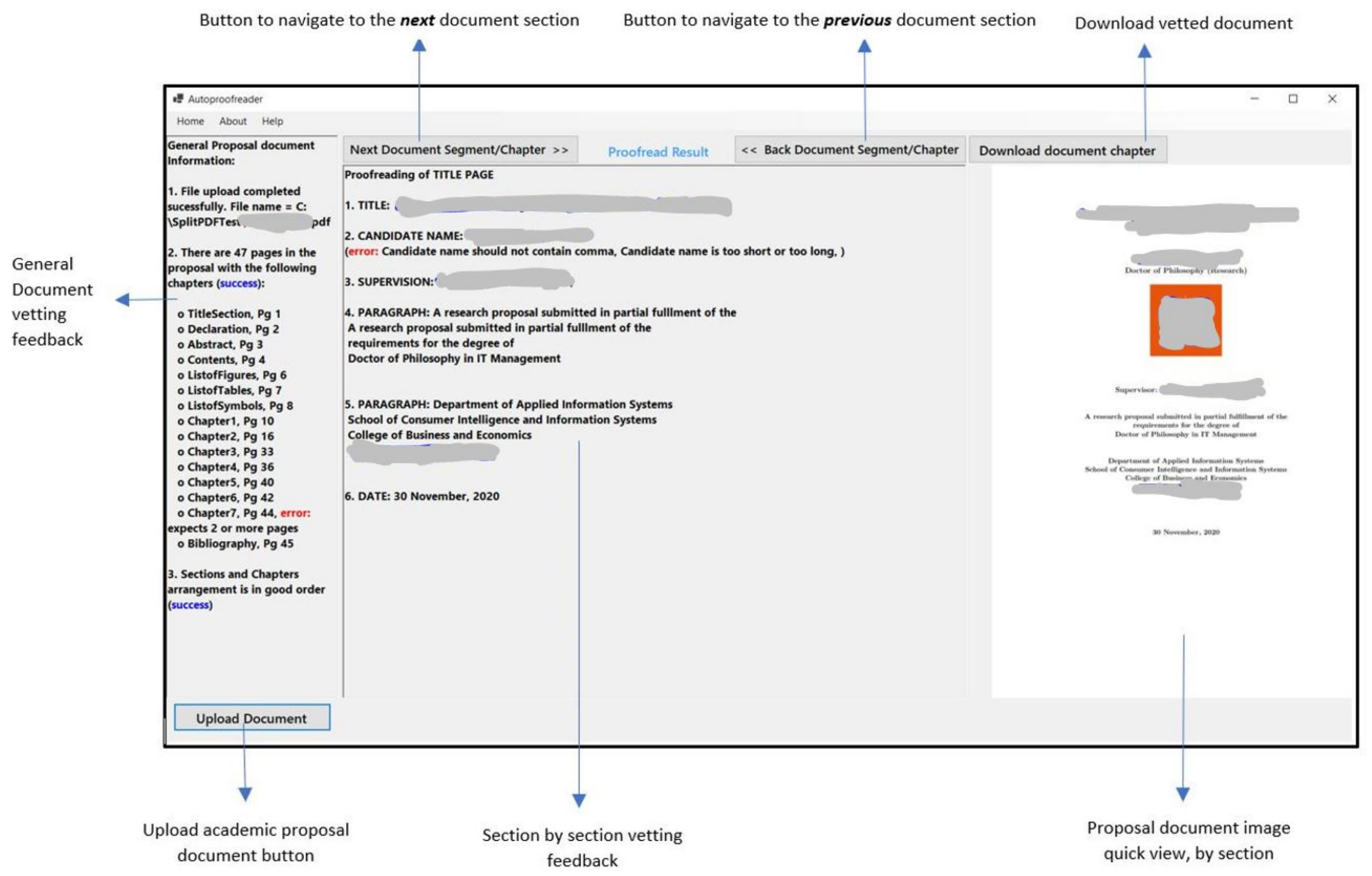
A corpus of ten (20) proposals collected from the online digital ecommons of various Universities was used to carry out the experimental result. The dataset consists of academic proposals of MSc and PhD post-graduate students in Information Systems' related discipline that were accessed on Universities institutional repository (IR) or available on online digital-common. These are academic proposal thesis submitted between year 2015 and 2021.

TABLE V. Results of Vetting of 20 Academic Proposals

S/N	Document Elements	TP	TN	FP	FN	Prec	Rec	Acc
1	Title	10	5	5	0	0.67	1.00	0.75
2	Author	5	5	9	1	0.36	0.83	0.50
3	Supervisor	15	5	0	0	1.00	1.00	1.00
4	Title-Paragraph	21	16	0	3	1.00	0.88	0.93
5	Section-Title	111	0	0	13	1.00	0.90	0.90
6	Sub-Section	2041	50	0	826	1.00	0.71	0.72
7	Paragraph	4019	255	0	538	1.00	0.88	0.88
8	Page-No	3011	0	0	0	1.00	1.00	1.00
9	Date	6	1	0	13	1.00	0.32	0.35
	Total	9239	337	14	1394	0.99	0.87	0.87

B. Result and Discussion

The difference in the number of correctly identified document elements by the tool differs significantly for some logical elements due to the complexities of computationally identifying them. The tool detected and proofread the structural items on the preliminary pages with better precision and accuracy. The overall tool precision and accuracy is given as 0.99 and 0.87, respectively. Refer to expressions given in Equation 27 and 29


Button to navigate to the *next* document section
Button to navigate to the *previous* document section
Download vetted document
General Document vetting feedback
Autoproofreader
Home About Help
Next Document Segment/Chapter >>
Proofread Result
<< Back Document Segment/Chapter
Download document chapter
Upload Document
Upload academic proposal document button
Section by section vetting feedback
Proposal document image quick view, by section

General Proposal document Information:

1. File upload completed sucessfully. File name = C: \SplitPDFTest[redacted]pdf

2. There are 47 pages in the proposal with the following chapters (success):

- o TitleSection, Pg 1
- o Declaration, Pg 2
- o Abstract, Pg 3
- o Contents, Pg 4
- o ListofFigures, Pg 6
- o ListofTables, Pg 7
- o ListofSymbols, Pg 8
- o Chapter1, Pg 10
- o Chapter2, Pg 16
- o Chapter3, Pg 33
- o Chapter4, Pg 36
- o Chapter5, Pg 40
- o Chapter6, Pg 42
- o Chapter7, Pg 44, error: expects 2 or more pages
- o Bibliography, Pg 45

3. Sections and Chapters arrangement is in good order (success)

Proofreading of TITLE PAGE

1. TITLE: [redacted]

2. CANDIDATE NAME: [redacted]
(error: Candidate name should not contain comma, Candidate name is too short or too long,)

3. SUPERVISION: [redacted]

4. PARAGRAPH: A research proposal submitted in partial fullment of the
A research proposal submitted in partial fullment of the
requirements for the degree of
Doctor of Philosophy in IT Management

5. PARAGRAPH: Department of Applied Information Systems
School of Consumer Intelligence and Information Systems
College of Business and Economics
[redacted]

6. DATE: 30 November, 2020

[redacted]

[redacted]
Doctor of Philosophy (Research)

[redacted]

Supervisor: [redacted]

A research proposal submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in IT Management

Department of Applied Information Systems
School of Consumer Intelligence and Information Systems
College of Business and Economics
[redacted]

30 November, 2020

Fig. 5. Autoproofreader Tool and Menus.

V. Evaluation

We present the evaluation for the tool in terms of accuracy for correctly classifying/ recognising syntactically right or wrong elements in the document. We present the confusion matrix model evaluation with the performance metrics of sensitivity, precision and accuracy given below. Table VI gives a brief explanation of how the confusion matrix is applied for evaluating the tool.

Overall Precision:

$$\frac{TP}{(TP+FP)} = \frac{9239}{(9239+14)} \approx 0.99 \tag{27}$$

Overall Recall:

$$\frac{TP}{(TP+FN)} = \frac{9239}{(9239+1394)} \approx 0.87 \tag{28}$$

Overall Accuracy:

$$\frac{(TP+TN)}{(TP+TN+FP+FN)} = \frac{(9239+337)}{(9239+337+14+1394)} \approx 0.87 \tag{29}$$

where:
TP = True Positives

[h!]

TABLE VI. Confusion Matrix Evaluation

Item	Auto-proofreader Tool(T/F)	Benchmark (P/N)	Description
TP	T = Identified	P = Correct Elements	Identified correct Elements
TN	T = Identified	N = Incorrect Elements	Identified incorrect Elements
FP	F = Did not identify	P = Correct Elements	Did not identified correct Elements
FN	F = Did not identify	N = Incorrect Elements	Did not identified incorrect Elements

TN = True Negatives
FP = False Positives
FN = False Negatives

VI. Conclusion and Future Work

Identifying errors in a (lengthy) textual document is an expert task that can be made into an artificial intelligent (AI) tool which can then assist human reviewers (i.e. Supervisors) to more effective and productive, especially when faced with many documents (i.e. academic proposals). The tool will not only be useful for academic proposals but can be refined for many varying template-based documents. In this paper, we have presented the technique and design of the tool for vetting an academic proposal for layout or structure-based errors. This technique slices a proposal document into its minute structural components (which we have termed as document elements contained a document section) and performs a check of correctness. The technique allows the slicing of the academic proposals into separate sections as documents and lastly allows the download of a **vetted** academic proposal document which can be used as a feedback to Students' candidate after an automatic vetting. The designed software tool was evaluated on twenty (20) proposal documents which gives an accuracy of 86%. The accuracy of the tool is only based on specific elements for evaluation but it can be made more robust with vetting more components (or elements) contained in a document section such as images (or figures) and tables. The software design is template specific but can be extended for various kind of template-driven text document vetting.

In the future, we will explore the structural vetting of more document logical elements: figures/images, tables and citations. We will also extend the tool to perform rule-based sentence level grammar vetting.

Acknowledgment

This research was partially supported by the Global Excellence Stature (GES) awards and National Research Fund (NRF) with grant number - 119041

References

[1] S. P. Glenda Rosy Clements, "Application of natural language processing in document vetting," *PSYCHOLOGY AND EDUCATION*, vol. 57, pp. 5651–5658, 11 2020.

[2] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and M. Sun, "How does nlp benefit legal system: A summary of legal artificial intelligence," 2020.

[3] R. Dale, "Law and word order: Nlp in legal tech," *Natural Language Engineering*, vol. 25, pp. 211–217, 01 2019.

[4] O. Daramola, "Lessons from postgraduate supervision in two african universities: An autoethnographic account," *Education Sciences*, vol. 11, p. 345, 07 2021.

[5] G. van Rensburg, P. Mayers, and L. Roets, "Supervision of postgraduate students in higher education," *Trends in Nursing*, vol. 3, 11 2016.

[6] N. S. Sudheesh K, Duggappa Duggappa Rani, "How to write a research proposal," pp. 631–634, 09 2016.

[7] M. Bushesha, H. Mtae, J. Msindai, and S. Mbogo, "Challenges facing supervisors and students in the process of writing theses/dissertations under odl: Experiences from the open university of tanzania," *Huria: Journal of the Open University of Tanzania*, vol. 12, pp. 118–131, 2012.

[8] G. van Rensburg, P. Mayers, and L. Roets, "Supervision of postgraduate students in higher education," *Trends in Nursing*, vol. 3, 11 2016.

[9] M. d. K. Jan Botha, Gabriele Beata Vilyte. (2019). [Online]. Available: "https://theconversation.com/digital-training-can-help-supervisors-lift-phd-output-126391"

[10] A. J. Hey-Cunningham, M.-H. Ward, and E. J. Miller, "Making the most of feedback for academic writing development in postgraduate research: Pilot of a combined programme for students and supervisors," *Innovations in Education and Teaching International*, vol. 58, no. 2, pp. 182–194, 2021. [Online]. Available: https://doi.org/10.1080/14703297.2020.1714472

[11] K. Wilmot and H. Lotz-Sisitka, *Supporting Academic Writing Practices in Postgraduate Studies. A sourcebook of academic writing support approaches and initiatives*, 2015.

[12] H. D. Kohn, "A mentoring program to help junior faculty members achieve scholarship success," *American Journal of Pharmaceutical Education*, vol. 78, 2014.

[13] N. Bak, *Research Proposal Guide*, 04 2015.

[14] L. Karyuatry and M. Rizqan, "Grammarly as a tool to improve students' writing quality: Free online-proofreader across the boundaries," *JSSH (Jurnal Sains Sosial dan Humaniora)*, vol. 2, p. 83, 05 2018.

[15] K. Siau and Y. Ma, "Artificial intelligence impacts on higher education," 05 2018.

[16] R. F. Heller, *The Distributed University for Sustainable Higher Education*. Springer, 2022.

[17] D. Maor and J. K. Currie, "The use of technology in postgraduate supervision pedagogy in two australian universities," *International Journal of Educational Technology in Higher Education*, vol. 14, no. 1, pp. 1–15, 2017.

[18] J. Miranda, C. Navarrete, J. Noguez, J.-M. Molina-Espinosa, M.-S. Ramírez-Montoya, S. A. Navarro-Tuch, M.-R. Bustamante-Bello, J.-B. Rosas-Fernández, and A. Molina, "The core components of education 4.0 in higher education: Three case studies in engineering education," *Computers & Electrical Engineering*, vol. 93, p. 107278, 2021.

[19] P. Race, "Some pros and cons of 'track changes' feedback on work returned to students electronically," 03 2014.

[20] B. P. Rasmus, "End to end information extraction from business documents," 2019.

[21] S. Abbott and A. Ade-Ibijola, "Algorithms and a tool for automatic decryption of clinical notes," *2019 6th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pp. 137–143, 2019.

[22] S. Kubeka and A. Ade-ibijola, "Automatic Comprehension and Summarisation of Legal Contracts," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 2, pp. 19–28, 2021.

[23] S. Singh, "Natural language processing for information extraction," 2018.

[24] H. Jiang, Y. Hua, D. Beeferman, and D. Roy, "Annotating the tweebank corpus on named entity recognition and building nlp models for social media analysis," *arXiv preprint arXiv:2201.07281*, 2022.

[25] G. Obaido, A. Ade-Ibijola, and H. Vadapalli, "Talksql: A tool for the synthesis of sql queries from verbal specifications," *2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, pp. 1–10, 2020.

[26] T. Revanth, K. V. Sai, R. Ramya, R. Chava, V. Sushma, and B. Ramya, "Nl2sql: Natural language to sql query translator," pp. 267–278, 2022.

[27] A. Gheewala, C. Turner, and J.-R. de Maistre, "Automatic extraction of legal citations using natural language processing," in *Proceedings of the 15th International Conference on Web Information Systems and Technologies*, ser. WEBIST 2019. Setubal, PRT: SCITEPRESS - Science and Technology Publications, Lda, 2019, p. 202–209. [Online]. Available: https://doi.org/10.5220/0008052702020209

[28] M. Cronje and A. Ade-Ibijola, "Automatic slicing and comprehension of cvs," in *2018 5th International Conference on Soft Computing & Machine Intelligence (ISCMI)*. IEEE, 2018, pp. 99–103.

[29] K. Ahmad, M. A. Ayub, K. Ahmad, J. Khan, N. Ahmad, and A. Al-Fuqaha, "Merit-based fusion of nlp techniques for instant feedback on water quality from twitter text," *arXiv preprint arXiv:2202.04462*, 2022.

[30] A. Alamoudi, A. Alomari, S. Alwarthan, and A. Rahman, "A rule-based information extraction approach for extracting metadata from pdf books," *ICIC Express Letters*, vol. 12, pp. 121–132, 02 2021.

[31] H. Assal, J. Seng, F. Kurfess, E. Schwarz, and K. Pohl, "Semantically-enhanced information extraction," in *2011 Aerospace Conference*, 2011, pp. 1–14.

[32] K. Verma, A. Kass, and R. Vasquez, "Using syntactic and semantic analyses to improve the quality of requirements documentation," *Semantic Web*, vol. 5, pp. 405–419, 01 2014.

[33] A. Ade-Ibijola, "Finchan a grammar-based tool for automatic comprehension of financial instant messages," *Proceedings of the Annual Conference of the South African Institute of Computer Scientist and Information Technologists*, vol. 0, no. 1, p. 0, 2016.

[34] X. Chen, H. Xie, G. Cheng, L. Poon, M. Leng, and F. L. Wang, "Trends and features of the applications of natural language processing techniques for clinical trials text analysis," *Applied Sciences*, vol. 10, p. 2157, 03 2020.

[35] E. S. Chifu, V. R. Chifu, I. Popa, and I. Salomie, “A system for detecting professional skills from resumes written in natural language,” in *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2017, pp. 189–196.

[36] A. Anjewierden, “Aidas: incremental logical structure discovery in pdf documents,” *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pp. 374–378, 2001.

[37] H. Langer and P. Bayerl, “Text type structure and logical document structure,” 07 2004.

[38] S. O. M. Perdana, Indra and F. A. Masri, “Effectiveness of online grammarly application in improving academic writing: Review of experts experience.” *International Journal of Social Sciences*, 2021.

[39] M. Nova, “Utilizing grammarly in evaluating academic writing: A narrative research on efl students’ experience,” *Premise: Journal of English Education*, vol. 7, p. 80, 04 2018.

[40] L. Karyuatry and M. Rizqan, “Grammarly as a tool to improve students’ writing quality: Free online-proofreader across the boundaries,” *JSSH (Jurnal Sains Sosial dan Humaniora)*, vol. 2, p. 83, 05 2018.

[41] T. Nur Fitria, “”grammarly” as ai-powered english writing assistant: Students’ alternative for english writing,” *Metathesis Journal of English Language Literature and Teaching*, vol. 5, pp. 65–78, 05 2021.

[42] A. Nasution and S. Fatimah, “The use of pro writing aid web in editing students writing,” *Journal of English Language Teaching*, vol. 7, no. 2, pp. 362–368, 2018.

[43] Y. Luo, J. Yu, and X. Cheng, “The research of chinese text proofreading system model,” 12 2020.

[44] A. Ade-Ibijola, “Synthesis of regular expression problems and solutions,” *International Journal of Computers and Applications*, vol. 42, no. 8, pp. 748–764, 2020. [Online]. Available: https://doi.org/10.1080/1206212X.2018.1482398

[45] J. C. Martin, *Introduction to languages and the theory of computation, fourth edition*. New-York: McGraw-Hill, 2010.