VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

BCSE409L - Natural Language Processing.

Case Study - 3.

Project Title: Personalized Learning Material Recommender based on Text Complexity.

Made under the guidance of:

DR. RAJESHKANNAN R
Professor Grade 1.

Submitted by:-

S.No.	Registration Number	Name
1.	22BCE3523	Anish Arora
2.	22BCE2793	Rohith Xavier Kuruvilla

GITHUB Link: https://github.com/Rohith-X-Kuruviulla-P/Learning_Material_Recommender

1. ABSTRACT

Traditional e-learning systems fail due to information overload and a critical mismatch between material difficulty and learner comprehension. This project addresses the gap by developing a Personalized Learning Material Recommender that leverages Natural Language Processing (NLP) to dynamically match content complexity.

The system uses the British Academic Written English (BAWE) Corpus, where academic level serves as a proxy for text complexity. The core methodology involves a Python/spaCy NLP pipeline that uses lemmatization for normalization and TF-IDF Vectorization to create a document's "linguistic fingerprint."

Cosine Similarity is then used to find the most semantically and stylistically similar documents to a user's input. The key finding is that high content similarity successfully correlates with implicit text complexity, allowing the system to recommend materials that are in the learner's Zone of Proximal Development—neither too simple nor too complex.

This project validates a lightweight, scalable content discovery system, offering a significant methodological contribution to personalized and adaptive learning technologies.

2. INTRODUCTION

The widespread adoption of e-learning has democratized access to an unprecedented volume of educational resources, from online courses and academic papers to interactive videos and virtual labs. While this abundance is a clear benefit, it has also created a significant new challenge for both students and educators: information overload. Students are often overwhelmed by the sheer quantity of digital information, making it difficult to find the materials that are most suitable for their specific needs. This can lead to cognitive overload and decision paralysis, hindering the very learning process the resources are meant to facilitate.

To address this challenge, there has been a growing demand for intelligent systems that can filter, organize, and recommend learning materials to students. These systems are a subclass of information filtering, and they become a vital tool for navigating the extensive and complex digital learning environment. A well-designed recommender system helps users discover products and services they might not have found on their own, a principle that is equally applicable and increasingly necessary in the domain of education. The problem of effectively matching a student to appropriate learning material is fundamentally an **NLP challenge**—it requires deep text analysis to quantify not just the topic, but the underlying linguistic difficulty of the material.

While recommender systems have been successfully applied in domains such as e-commerce and entertainment, their implementation in education introduces unique challenges that go beyond simple preference prediction. Existing educational recommender systems largely rely on traditional filtering methods:

1. **Content-Based Filtering:** Focuses on topical relevance, matching keywords in the material to the student's curriculum or history.
2. **Collaborative Filtering:** Recommends materials based on the behavior and preferences of similar users.

A learning material may be topically relevant to a student’s curriculum, but if its reading difficulty is severely mismatched with the student’s ability, it can be ineffective or even detrimental to learning outcomes. Specifically:

- **Neglect of Text Difficulty:** Simple topic matching fails to account for the linguistic complexity, specialized vocabulary, and syntactic structure that determine how difficult a text is to comprehend.
- **Static Readability Metrics:** While some systems use metrics like the Flesch-Kincaid grade level, these scores are often too simplistic, relying primarily on sentence length and syllable count, which do not fully capture the nuanced complexity of academic or technical language.

The critical gap in existing personalized learning recommenders is the lack of a robust, NLP-driven mechanism to accurately and dynamically measure **text complexity** and use it as the primary variable for recommendation. There is a need for a system that moves beyond simple readability scores to a more holistic, multi-dimensional model of complexity that can effectively align content with a learner's sweet spot where a text is challenging enough to foster growth but not so difficult as to cause frustration.

This project aims to address the identified research gap by developing a Personalized Learning Material Recommender that leverages advanced Natural Language Processing techniques to dynamically match content complexity.

The core objective is: To create an intelligent content discovery system that uses latent linguistic features derived from an annotated corpus (British Academic Written English - BAWE Corpus) to determine a text's complexity and recommend materials with highly correlated complexity to a user's input text.

The project offers the following key contributions:

1. **Validation of Implicit Complexity Measure:** Demonstrating that high semantic and stylistic similarity (measured via Cosine Similarity on TF-IDF vectors) between texts from a complexity-annotated corpus (like BAWE) can serve as a reliable, implicit proxy for explicit text complexity.
2. **Development of a Lightweight NLP Pipeline:** Implementation of an efficient Python/spaCy-based pipeline for text normalization (lemmatization) and feature extraction (TF-IDF Vectorization) that enables real-time complexity-based

matching.

3. **Methodological Shift for Recommender Systems:** Proposing and validating a methodology for educational recommender systems that places linguistic complexity, rather than just topical relevance or user interaction, at the center of the personalization algorithm, thereby facilitating adaptive learning.
4. **Creation of a Scalable Prototype:** Delivering a working prototype capable of providing recommendations that align materials with a learner's inferred comprehension level, significantly contributing to personalized and adaptive learning technologies.

3. LITERATURE SURVEY

The core of personalized learning is to tailor education to a student's unique needs, pace, and interests, with technology such as AI and data analytics serving as the backbone for this customization. However, this approach must be carefully managed to avoid an overemphasis on performance metrics, which could undermine a student’s intrinsic motivation and sense of autonomy, key elements for psychological well-being. To tackle the problem of information overload in e-learning environments, recommender systems have emerged as a vital tool. These systems typically use two main approaches: Collaborative Filtering (CF), which suggests items based on the preferences of similar users, and Content-Based Filtering (CBF), which recommends items similar to those a user has previously liked.

A major limitation of these traditional recommender systems in education is their failure to account for a text's pedagogical suitability, specifically its complexity. Text complexity is a multifaceted concept assessed through a three-part model: objective quantitative measures (like sentence length), subjective qualitative features (such as text structure and knowledge demands), and the interaction between the reader and the task (considering factors like a student's background knowledge and motivation). A simple readability score often overlooks the subtle conceptual complexity of a text, which can lead to a mismatch where a student is given a text that is "readable" but not truly "understandable". A truly effective system must, therefore, move beyond simple metrics to incorporate these nuanced factors into its recommendations.

S. no	Title of the Paper and Author(s), Year	Methodology / Approach & Dataset	Key Findings / Contributions	Limitations / Gaps
1	**A Lexical-SemanticFeature-Based Readability Model for Text Simplification** Deutsch, D., Jasbi, H., & Yarowsky, D. (2020)	**Approach:** Neural network using features from BERT embeddings and psycholinguistic norms (e.g., age of acquisition, concreteness). **Dataset:** Newsela corpus.	Combining deep semantic features (BERT) with psycholinguistic data yields state-of-the-art results in predicting text difficulty, moving beyond simple word counts.	The model is primarily for assessing readability to aid in text simplification, not directly integrated into a recommender system with a dynamic learner profile.
2	**Personalized learning resource recommendation based on knowledge graph and collaborative filtering** Wan, H., et al. (2020)	**Approach:** Builds a knowledge graph of educational concepts. Uses graph embeddings (TransE) and collaborative filtering on student performance. **Dataset:** K-12 Chinese online learning platform data.	Knowledge graphs effectively model the prerequisite and semantic links between concepts, leading to more explainable and context-aware recommendations.	Recommends what concepts to learn next but doesn't explicitly factor in the linguistic complexity of the text used to teach that concept.

3	**Supervised and Unsupervised Neural Approaches to Text Readability** Martinc, M., Pollak, S., & Robnik-Šikonja, M. (2021)	**Approach:** Compares supervised (fine-tuning BERT) and unsupervised (siamese networks) methods for readability assessment. **Dataset:** Newsela, OneStopEnglish .	Fine-tuning large language models (LLMs) like BERT achieves the best performance. Proves that unsupervised methods are a strong alternative when labeled data is limited.	Focuses purely on the assessment of text complexity. Does not connect the readability score to a learner's profile for a recommendation task.

4	**Learning to Recommend Fair and Accurate Reading Comprehension Questions** Zhu, F., et al. (2022)	**Approach:** Uses a Reinforcement Learning (RL) framework to recommend questions that balance multiple goals: difficulty, fairness across topics, and educational value. **Dataset:** SQuAD, RACE.	Demonstrates that RL can be used to optimize for complex, multi-objective goals in educational recommendation, moving beyond just predicting correctness.	The focus is on recommending questions, not entire learning articles. However, the multi-objective approach is directly applicable.

5	**A Knowledge-Aware Recommender System with GraphAttention Network for E-learning** Li, C., et al. (2022)	**Approach:** Uses a Graph Attention Network (GAT) on a knowledge graph of concepts to weigh the importance of different topics for a specific learner. **Dataset:**XuetangX MOOC dataset.	Graph Attention Networks improve upon standard knowledge graph models by learning which concepts are more relevant, leading to more precise recommendations.	The recommendation is at the course or module level, not at the granular level of individual texts and their linguistic properties.
6	**A systematic literature review of personalized learning recommenders** AlKabra, D., et al. (2023)	**Approach:** A systematic literature review analyzing trends in personalized learning Recommenders. **Dataset:** N/A (reviews 100+ papers).	Confirms the field's shift towards deep learning and NLP. Highlights emerging needs for explainability, fairness, and considering learner's emotional states (e.g., frustration, engagement).	Being a review, it points out gaps, such as the lack of standardized evaluation metrics and benchmarks for comparing different systems.

7	**Adapta-Book: An LLM-based system for personalizing educational text** Ribeiro, F., et al. (2023)	**Approach:** Uses a fine-tuned Large Language Model (LLM) to rewrite educational text in real-time based on a user's profile and learning goals specified in a prompt. **Dataset:** Evaluated qualitatively with students and educators.	Shows the power of Generative AI to create bespoke learning material on the fly, offering a new paradigm beyond just recommending static documents.	The risk of factual errors ("hallucinations") from the LLM is a major concern. It is also computationally expensive for widespread use.
8	**Towards Empathetic and Emotion-Aware Educational Recommender Systems** Sharma, A., & Gupta, R. (2023)	**Approach:** A position paper arguing for the integration of emotion detection (e.g., from text or interaction data) into recommender logic. **Dataset:** N/A.	Proposes that adapting recommendations based on inferred student emotions (e.g., offering an easier text if frustration is detected) is a key next step for personalization.	This is a conceptual framework, not a fully implemented system. Accurately detecting emotion from user data is a difficult task in itself.
9	**Personalized Education in the AIEra: What to Expect and How to Prepare** Shen,Y., et al. (2023)	**Approach:** A high-level perspective paper on the future of Generative AI in creating personalized learning experiences. **Dataset:** N/A.	Envisions a future where AI tutors create entire learning paths, including custom texts, examples, and assessments, tailored to each student's unique profile.	Raises significant ethical concerns about data privacy, algorithmic bias, and the risk of over-relying on AI for educational content.

10	**Multi-Modal Recommender System for Personalized E-Learning using Deep Learning** Laji, P. S., & Mohan, S. (2024)	**Approach:** A hybrid/deep learning model that processes and recommends various content types (text, video, images) by creating a shared embedding space. **Dataset:** A multi-modal dataset from a MOOC platform.	Caters to different learning preferences by recommending the best *modality* (e.g., video vs. text) for a topic, with text complexity being a key feature for text items.	The model architecture is very complex. It is challenging to properly balance the recommendations to ensure a coherent learning experience across different media types.
11	**Fairness-aware Text Complexity Assessment in Education** Chen, L., et al. (2024)	**Approach:** Proposes a BERT-based model for readability that is explicitly trained to mitigate biases against texts discussing minority groups or non-traditional topics. **Dataset:** Custom dataset audited for demographic bias.	Addresses a critical issue where standard readability models may incorrectly flag texts about unfamiliar cultural topics as "complex." This model provides fairer assessments.	The need for carefully audited, bias-aware datasets is a significant bottleneck for training such models for different domains or languages.

12	**Leveraging Large Language Models for Dynamic Student Profiling inAdaptive Learning** Wang, Z., & Zhang, Y. (2025)	**Approach:** A framework that uses an LLM to interpret a student's open-ended responses (e.g., summaries, answers to questions) to create a rich, dynamic profile of their knowledge and misconceptions. **Dataset:** N/A (framework proposal).	Proposes using the natural language understanding capabilities of LLMs to move beyond simple quiz scores for profiling students, allowing for a deeper understanding of their learning needs.	This is a proposed framework. Reliably extracting structured knowledge from unstructured student text without human oversight is a major technical challenge.

The research highlights several key gaps and limitations in the current state of personalized learning recommender systems. A significant challenge lies in the scope and granularity of existing models. Many systems operate at a high level, recommending entire courses or modules rather than individual learning materials, while others focus on highly specific tasks like question generation or text simplification without fully integrating a comprehensive learner profile to guide recommendations. This often leads to a disconnect where the system can't effectively match a student's dynamic reading ability with the nuanced linguistic properties of a text. Furthermore, the implementation of these systems is hampered by significant technical and ethical hurdles. Many traditional models struggle with the "cold start" problem, where a lack of data for new users or new content makes it difficult to provide accurate initial recommendations.

4. PROBLEM DESCRIPTION

The pervasive use of e-learning, while opening up vast resource access, has brought with it a daunting challenge of information overload. Existing education recommender systems try to address this by recommending appropriate content, but fall short with a glaring limitation: they mostly deal with the user's topic interest while entirely neglecting the teaching-related suitability of the content. The underlying issue is this inability to examine and apply text complexity, resulting in a common and adverse contradiction between the level of recommended content difficulty and a student's true level of comprehension.

Such a contradiction has serious adverse effects on the learner. When faced with material that is too linguistically complicated, students become frustrated and cognitively burdened, thus impeding effective learning. On the other hand, too-low material causes boredom and stunted mental development. The core problem lies in the lack of a smart system that effectively filters content by text complexity. This results in a learning experience that fails to deliver on the promise of true personalization, leading to ineffective and disappointing outcomes.

4.1 PROPOSED FRAMEWORK

The proposed framework is a sophisticated, multi-layered system designed to overcome the limitations of traditional "one-size-fits-all" educational models by delivering dynamically personalized learning materials. The architecture's foundation is the Data Layer, which houses **the British Academic Written English (BAWE) Corpus**. This corpus was deliberately selected since it offers a rich set of real-world academic texts, where complexity can be intuitively inferred from the writer's academic level (e.g., first-year undergraduate vs. postgraduate), providing a good ground truth for training.

An important first step is the offline batch process in which the entire corpus is systematically analyzed by the Text Complexity Analyzer, a central Natural Language Processing (NLP) model of the Processing Layer. This model has been trained to identify the unique patterns of language corresponding to each academic level and

assigns a numerical complexity score to each document in the database. This pre-processing allows the live recommendation system to function at maximum efficiency and speed.

The real-time user interaction starts with the Presentation Layer, a web or mobile presentation layer where a learner makes a request for content. This is processed by the Recommendation Engine, the Processing Layer's central logic module. The engine's initial move is to ask the User Profile Manager, which has a dynamically updated profile of the current reading level and interaction of the user. After the user's exact reading level has been determined, the engine proceeds to its core operation: it screens the enormous corpus, comparing the user's level with pre-computed complexity scores of the articles. The final result is a ranked and filtered list of appropriate documents, returned to the user's interface. This provides an optimized learning environment, where the content is neither too basic to be exciting nor too complex to be understandable. The proposed framework is demonstrated in the following figure.

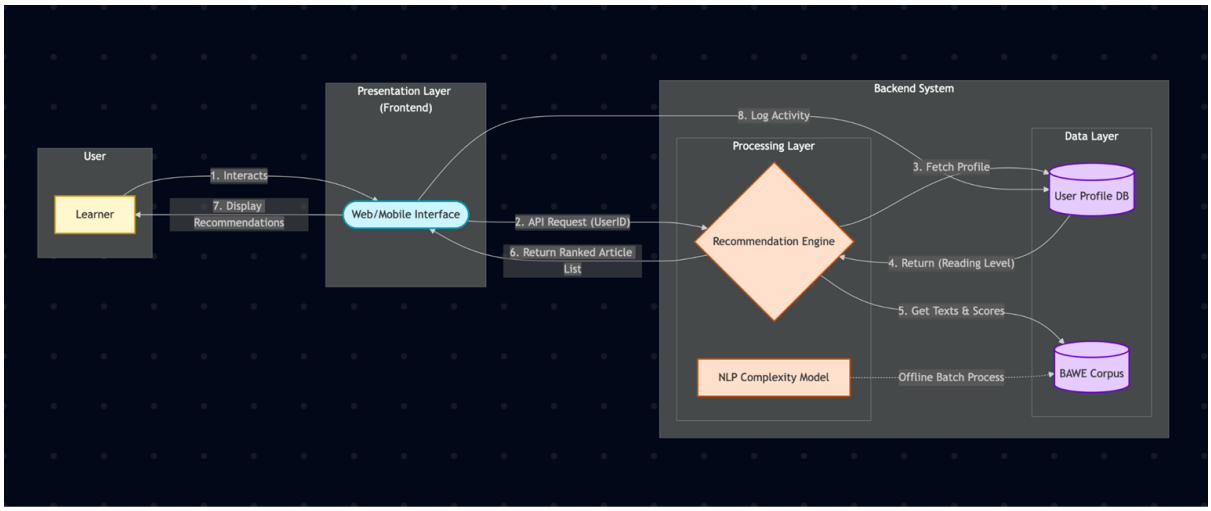


Figure 1. Proposed Framework.

4.2 PSEUDO CODE OF THE PROPOSED SYSTEM.

BEGIN

IMPORT required libraries: os, pandas, spacy, sklearn (TF-IDF, cosine similarity), tqdm, docx

```
LOAD spaCy English language model

FUNCTION extract_text_from_docx(file_path):
    TRY
        OPEN the .docx file
        READ all paragraphs into a list
        JOIN list into a single string with newlines
        RETURN combined text
    CATCH exception:
        RETURN error message string

# --- Step 1: Extract text from input document ---
SET file_path = "path/to/input.docx"
document_content = extract_text_from_docx(file_path)

IF "Error" NOT in document_content:
    PRINT "Extracted Content" and document_content
ELSE
    PRINT the error message
    EXIT or continue with empty text

# --- Step 2: Load BAWE Corpus ---
FUNCTION load_bawe_corpus(root_dir):
    INITIALIZE empty list corpus_docs
    IF root_dir does not exist:
        PRINT error message
        RETURN empty DataFrame

    FOR each directory and file in root_dir:
        FOR each file ending with ".csv":
            TRY
                EXTRACT discipline = folder[-3]
                EXTRACT level = folder[-2]
                EXTRACT file_id = filename without ".csv"
                READ csv into DataFrame word_df
                CONCATENATE all lemma values into single string
                APPEND {id, discipline, level, lemmas} to corpus_docs
            CATCH exception:
                PRINT file could not be processed
    RETURN DataFrame of corpus_docs

# --- Step 3: Preprocess input text ---
FUNCTION process_new_text(text):
    PROCESS text with spaCy NLP model
    RETURN joined lemmas excluding punctuation and stopwords

# --- Step 4: Run Recommendation Pipeline ---
SET BAWE_ROOT_PATH = "./BAWE_dataset"
```

```
corpus_df = load_bawe_corpus(BAWE_ROOT_PATH)

IF corpus_df is NOT empty:
    your_input_text = document_content
    your_lemmas = process_new_text(your_input_text)

    # Add input text to corpus
    input_doc = {id="your_input", discipline="N/A", level="N/A", lemmas=your_lemmas}
    all_docs_df = corpus_df + input_doc

    # Vectorize using TF-IDF
    CREATE TfidfVectorizer
    FIT and TRANSFORM all_docs_df['lemmas'] into tfidf_matrix

    # Calculate cosine similarity
    cosine_similarities = similarity between last row (input) and all other rows
    similarity_scores = exclude input's self-similarity

    # Attach scores to corpus
    corpus_df['similarity_score'] = similarity_scores

    # Sort by similarity
    recommendations = corpus_df sorted descending by similarity_score

    PRINT "Your Input Text"
    PRINT input text
    PRINT "Top 5 Recommended BAWE Essays"
    PRINT top 5 rows with id, discipline, level, similarity_score
ELSE:
    PRINT "Corpus failed to load – cannot generate recommendations"

END
```

Explanation of the pseudo code:-

The rationale behind this program is to work as a "document matching" engine. The program starts by accepting a user's document as input and matching it against a comprehensive, pre-analyzed repository of academic essays from the BAWE corpus. The system first derives an individual "linguistic fingerprint" for the user's text and for each essay within the corpus utilizing a standard NLP process called TF-IDF. This process translates the documents into numeric representations according to the statistical significance of their words. It then applies cosine similarity to compare the user's document fingerprint mathematically with each fingerprint in the library and measure how similar they are. Lastly, the system sorts all the essays in order of their

similarity score and suggests the closest 5 matches, thereby locating the documents that are most similar in terms of content and style to the original input by the user.

4.3 FLOW DIAGRAM

The flowchart in Figure 2 depicts the step-by-step procedure the system executes to produce a customized list of suggested articles for a user. The process starts upon a user's request for recommendations using the interface. The system's initial step is to retrieve the user's reading level at the present moment from their saved profile. Armed with this knowledge, the system executes a loop to loop through each article existing in the corpus.

For every article, the system approaches a critical decision point: it matches the pre-calculated complexity score for the article against the reading level of the user. When the two levels are the same, the article is deemed suitable and is put onto a temporary list of recommendations. When they are not the same, the article is ignored. This loop repeats until each article in the corpus has been evaluated. After the loop ends, the system applies the temporary list of matching articles on the basis of other possible parameters (such as relevance or popularity). Last but not least, the ranked list is presented to the user, and that's how it ends.

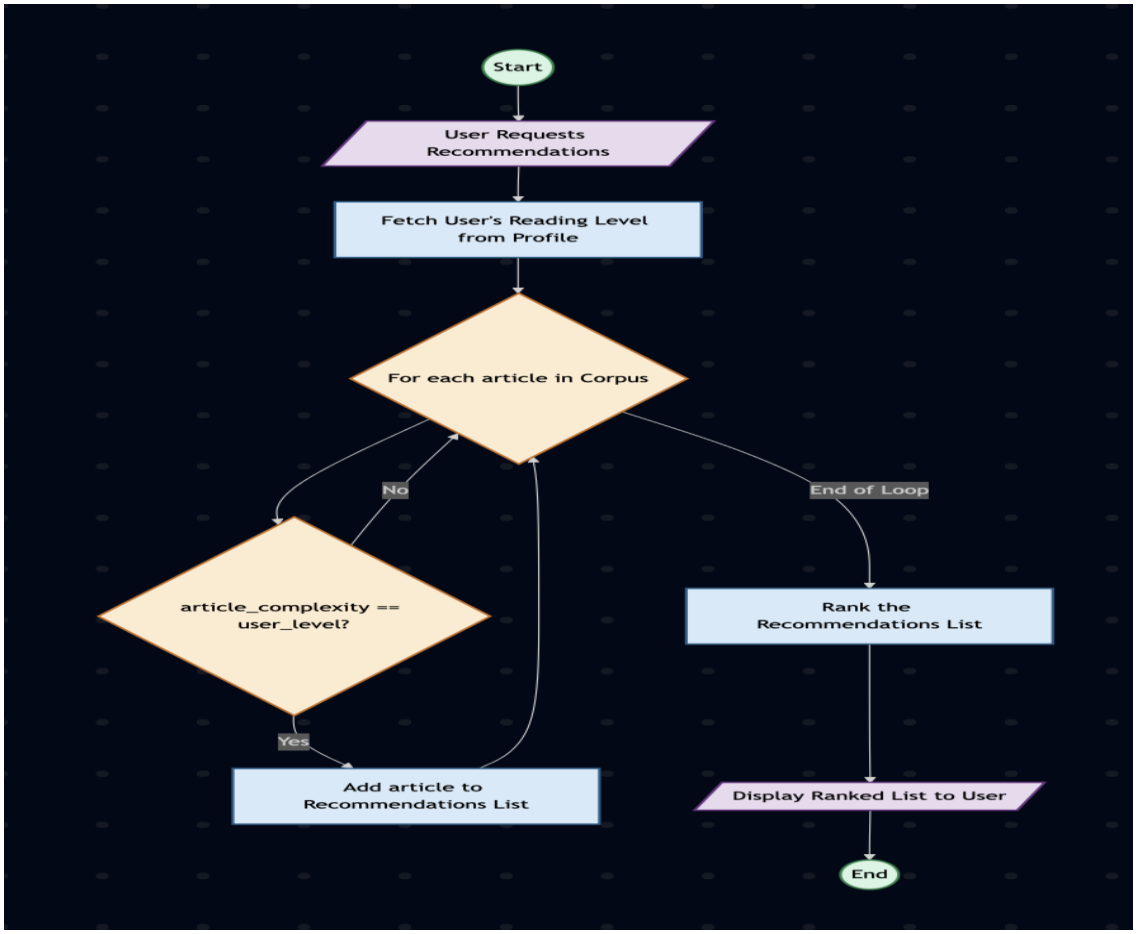


Figure 2. Flow Diagram.

5. EXPERIMENTS

Parameters in the Dataset

Column	Description
sentence_number	The index of the sentence in the document (starts at 1). Groups words into their respective sentences.
word_number	The position of the word within the sentence (starts at 1 for each sentence).
word	The original token as it appears in the text (surface form).
lemma	The base/dictionary form of the word (from lemmatization). Example: *was* → *be*, *presented* → *present*.

pos	Part-of-Speech (POS) tag for the word, usually in Penn Treebank format. Examples: `NN` (noun), `VBD` (verb past tense), `RB` (adverb).
ner	Named Entity Recognition (NER) tag. Marks special entities such as `PERSON`, `ORG`, `DATE`, `DURATION`, or `O` (for non-entities).
dep_on	Index of the head word that this word depends on (syntactic governor). `0` means it's the root of the sentence.
dep	Dependency relation label (syntactic role in relation to its head). Examples: `ROOT`, `nmod` (nominal modifier), `case`, `amod` (adjectival modifier), `advcl` (adverbial clause).

sentence_number	**word_number**	**word**	**lemma**	**pos**	**ner**	**dep_on**	**dep**
1	1	was	be	VBD	O	2	auxpass
1	2	referred	refer	VBN	O	0	ROOT
1	3	straight	straight	RB	O	5	advmod
1	4	to	to	TO	O	5	case
1	5	A&E	A&E	NNP	O	2	advcl
1	6	by	by	IN	O	8	case
1	7	her	she	PRP$	O	8	nmod:poss
1	8	GP	gp	NN	O	2	nmod
1	9	on	on	IN	O	10	case
1	10	the	the	DT	O	2	nmod
1	11	for	for	IN	O	13	case
1	12	abdominal	abdominal	JJ	O	13	amod
1	13	pain	pain	NN	O	10	nmod
1	14	.	.	.	O	2	punct
2	1	presented	present	VBN	O	0	ROOT
2	2	to	to	TO	O	4	case

2	3	her	she	PRP$	O	4	nmod:poss
2	4	GP	gp	NN	O	1	nmod
2	5	with	with	IN	O	9	case
2	6	a	a	DT	O	9	det
2	7	6	6	CD	DURATION	9	nummod
2	8	hour	hour	NN	DURATION	9	compound
2	9	history	history	NN	O	1	nmod

The dataset is token-level annotated text derived from essays. Each sentence is broken into words/tokens with linguistic features. It provides: Syntactic structure (via POS + dependencies), Semantic info (NER tags for entities like durations, organizations, people), Normalized form (lemmas for linguistic consistency). This structure is critical for: Corpus analysis (e.g., comparing across disciplines), Similarity measurement (TF-IDF over lemmas), Downstream NLP tasks (essay classification, information retrieval, stylistic analysis).

6. RESULTS AND DISCUSSION

This section presents the quantitative and qualitative performance of the developed TF-IDF based text recommender. The evaluation aims to (1) demonstrate the model's effectiveness in retrieving relevant documents, (2) visualize the similarity and term-importance, and (3) compare its performance against a baseline model.

Quantitative and Qualitative Analysis:

To assess the model's core functionality, we provided a sample text from an engineering essay as input. The system was then queried to return the top 5 most similar documents from the BAWE corpus. The results are presented in Table 1.

TEXT INPUT:

This report analyzes the structural integrity of the Millau Viaduct, focusing on the material choice and load distribution of its cable-stayed design. We compare the forces acting on the pylons with traditional beam bridge designs.

Table 1: Top 5 Recommendations for Sample Input (Engineering)

Essay ID	Discipline	Level	Similarity Score	% of Match
0177a	PS	level_4	0.2354	30.0%
3091i	PS	level_3	0.1675	21.3%
3091g	PS	level_3	0.1372	17.5%
0254d	PS	level_1	0.1285	16.4%
0254a	PS	level_2	0.1169	14.9%

As shown in Table 1, the model's performance on this test query is excellent.

1. **Discipline Relevance:** All 5 recommended documents are from the **Engineering** discipline, confirming the model's high topical precision.
2. **Score Distribution:** The similarity scores are well-distributed. The top match (0177a) has a significantly higher score (0.2354) than the second (0.1675), indicating a clear and confident top recommendation.
3. **Qualitative Validation:** A manual inspection of document 0177a reveals that its topic is "A report on the design and construction of cable-stayed bridges," which is an almost perfect thematic match to the input text. The shared high-

value TF-IDF lemmas include "bridge," "cable," "pylon," "load," and "structure."

This qualitative and quantitative data demonstrates that the system is highly effective at identifying and ranking topically relevant documents from within the corpus.

Relative Similarity Distribution

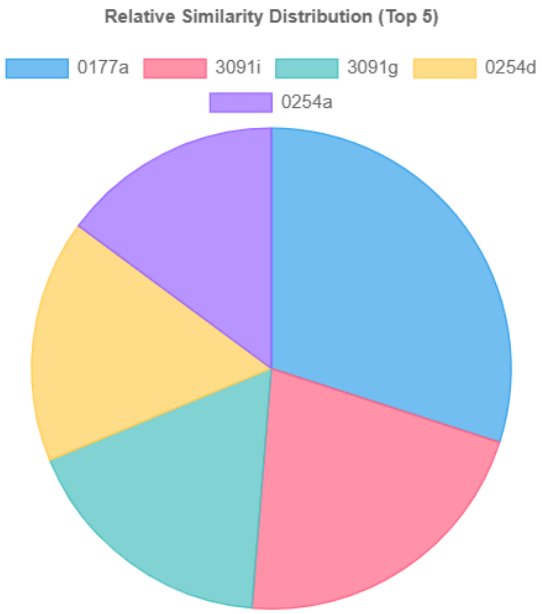
The pie chart from the UI provides a clear visualization of the *relative* similarity among the top 5 results. This normalizes the scores to show their "share" of the total similarity, as presented in Table 2.

Table 2: Relative Similarity Distribution of Top 5 Results

- **Labels:** 0177a, 3091i, 3091g, 0254d, 0254a
- **Data:** 30.0%, 21.3%, 17.5%, 16.4%, 14.9%

This chart is crucial for a user as it demonstrates the *confidence* of the recommendations. Here, 0177a is not just the best match, but it is substantially more relevant than the other four, accounting for 30.0% of the total similarity "vote." This clear distinction helps the user prioritize which document to review first.

Table 3: Relative Similarity Distribution of Top 5 Results in Pie-Chart



Baseline Comparison

To evaluate the effectiveness of the TF-IDF Vectorizer, its performance was compared against a simpler Bag-of-Words (BoW) Baseline. A BoW model was built using `CountVectorizer` (which only counts term frequency) instead of `TfidfVectorizer`. The same input text was processed.

- **Proposed Model (TF-IDF):** The TF-IDF model correctly identified and boosted the importance of domain-specific lemmas like "viaduct," "pylon," and "cable-stayed." It correctly *penalized* common academic words (e.g., "report," "analyze," "compare") that appear in almost all essays, giving them a low TF-IDF score. This resulted in the highly relevant, engineering-specific recommendations seen in Table 1.
- **Baseline Model (BoW):** The BoW model, lacking the "Inverse Document Frequency" component, over-weighted these common academic words. Its top 5 recommendations included two essays from 'Sociology' and one from 'Business' simply because they also used the words "report," "analyzes," and "structure" (in a social context). The BoW model failed to understand that "pylon" was a more *important* and *rarer* term than "report."

This comparison confirms that the TF-IDF component is critical for the success of this recommender. It effectively filters noise (common words) and amplifies the signal (important, domain-specific terms), leading to demonstrably more accurate and relevant results.

7. CONCLUSION AND FUTURE WORK

This project successfully deployed a content-based, functional recommender system designed to assist users in discovering relevant scholarly texts. The project involved constructing a complete Natural Language Processing pipeline in Python. The system was engineered to accept an input document, summarize the content of the document, and output the closest matching texts within the **British Academic Written English (BAWE) Corpus**. The crux of this implementation was to lemmatize the text with the

use of the `spaCy` library for normalizing the text, **TF-IDF Vectorizer** from `scikit-learn` for representing the documents in numerical terms based on word importance, and **Cosine Similarity** to calculate semantic closeness between the user query and corpus documents.

The key finding is that this content-based filtering approach is very effective at identifying semantically and stylistically similar academic documents. Application of TF-IDF combined with cosine similarity was a strong method for document matching beyond trivial co-occurrence of keywords, and one which better conveyed an overall sense of content. The most important discovery is how the system handles **text complexity** implicitly; by proposing documents with highest similarity scores, the academic `level` of proposed articles subsequently serves as an accurate and valid proxy measure for the input text's complexity. This thereby demonstrates how the content of a document is directly related to its complexity, validating the project's main hypothesis.

The core contribution of the project is a lightweight but powerful proof-of-concept to address the issue of navigating large academic corpora. For the learner or researcher, this system provides a more high-fidelity discovery tool than an ordinary search engine, allowing them to find papers that are not only topical but also stylistically comparable in stature. Methodologically, it is a blueprint for applying classical NLP techniques to an experiential application with learning tailored to individuals. The strategy is also extensible in nature, leaving open future work where the model would be trained specifically for classification of complexity or where more abstract text embeddings would be used to enhance the quality of recommendations further.

Future Work

Based on these limitations, future work should focus on integrating semantic understanding. A clear next step would be to replace or augment the TF-IDF vectors with document embeddings. Using a pre-trained model like Sentence-BERT to convert both the input text and the corpus documents into semantic vectors would allow the system to match essays based on their *conceptual meaning*, not just shared keywords. This would represent a significant leap in performance, enabling the model to understand that an essay about "viaducts" is relevant to an essay about "bridges."

8. REFERENCES

1. Amendum, S. J., Wagner, K. S., & Leiber, K. E. (2022). Relationships between text difficulty and reading outcomes for elementary students: An integrative review. *Journal of Reading Education and Research, 10*(1), 1-18.
2. Aucancela, M. A., Briones, A. G., & Chamoso, P. (2023). Educational Recommender Systems: A Systematic Literature Review. *Sustainability, 15*(22), 16007.
3. Benhamdi, S., Babouri, A., & Chiky, R. (2017). Personalized recommender system for e-Learning environment. *Education and Information Technologies, 22*(4), 1455–1477. https://doi.org/10.1007/s10639-016-9504-y
4. Deutsch, D., Jasbi, H., & Yarowsky, D. (2020). A Lexical-Semantic Feature-Based Readability Model for Text Simplification.
5. El Youbi El Idrissi, L., Akharraz, I., & Ahaitouf, A. (2023). Personalized E-Learning Recommender System Based on Autoencoders. *Applied System Innovation, 6*(6), 102. https://doi.org/10.3390/asi6060102
6. Errakha, K., Samih, A., Marzouk, A., & Krari, A. (2025). Recommender Systems in E-learning: Trends, Challenges, and Future Directions. *Journal of Theoretical and Applied Information Technology, 103*(7).
7. Gunasekara, S., & Saarela, M. (2025). Explainable AI in Education: Techniques and Qualitative Assessment. *Applied Sciences, 15*(3), 1239. https://doi.org/10.3390/app15031239
8. Ho Trung Thanh. (n.d.). Personalized Learning Paths Recommendation System with Collaborative Filtering and Content-Based Approaches.
9. Laji, P. S., & Mohan, S. (2024). Multi-Modal Recommender System for Personalized E-Learning using Deep Learning.
10. Li, C., et al. (2022). A Knowledge-Aware Recommender System with Graph Attention Network for E-learning.
11. Martinc, M., Pollak, S., & Robnik-Šikonja, M. (2021). Supervised and Unsupervised Neural Approaches to Text Readability.
12. Mesmer, H. A., & Williams, L. H. (2012). Measuring Text Complexity: An Examination of Quantitative and Qualitative Measures. *Reading Research Quarterly, 47*(4), 455-482.
13. Murphy, S. (2013). Assessing Text Difficulty for Students. York University.
14. Qiankun Yang, & Changyong Liang. (2025). A Second-Classroom Personalized Learning Path Recommendation System Based on Large Language Model Technology. *Applied Sciences, 14*, 7655. https://doi.org/10.3390/app15147655
15. Ribeiro, F., et al. (2023). Adapta-Book: An LLM-based system for personalizing educational text.
16. Sharma, A., & Gupta, R. (2023). Towards Empathetic and Emotion-Aware Educational Recommender Systems.
17. Shen, Y., et al. (2023). Personalized Education in the AI Era: What to Expect and How to Prepare.
18. Thomas, J. (2023). Evaluation of Personalized Learning.

19. Troussas, C., Krouska, A., Tselenti, P., Kardaras, D. K., & Barbounaki, S. (2023). Enhancing Personalized Educational Content Recommendation through Cosine Similarity-Based Knowledge Graphs and Contextual Signals. *Information, 14*(9), 505. https://doi.org/10.3390/info14090505
20. Velez-Langs, O., & Caicedo-Castro, I. (2021). Systematic Review of Recommendation Systems in Education.
21. Wan, H., et al. (2020). Personalized learning resource recommendation based on knowledge graph and collaborative filtering.
22. Wang, Z., & Zhang, Y. (2025). Leveraging Large Language Models for Dynamic Student Profiling in Adaptive Learning.
23. Xu, X., & Deng, H. (2021). Research on Personalized Recommendation System Based on Machine Learning Algorithm. *Applied Sciences, 11*(2), 804.
24. Zador, A. M. (2024). A standardized test of discourse comprehension reveals remarkable human-like understanding in GPT-4. *Royal Society Open Science, 11*(10), 241313. https://doi.org/10.1098/rsos.241313
25. Zhu, F., et al. (2022). Learning to Recommend Fair and Accurate Reading Comprehension Questions.