Assessing the Disciplinary Depth of a Text: A Transformer Based Approach and a new Dataset

ABSTRACT

This paper investigates means of assessing the disciplinary depth of a text, which reflects the level of proficiency of the author and the expected reader's capacity to understand the text. This measure can prove useful for adjusting the pedagogical content to the student's level of disciplinary mastery, as well as for assessing student answers. First we apply a pre-trained BERT-based classifier to categorize texts as either beginner-level or expert-level, leveraging BERT's deep contextual understanding of language. In this step, BERT generates embeddings for the text, which are then used to classify the texts based on their complexity. We employ a centroid-based approach, where we compute the cosine similarity between the embeddings of the test texts and the centroids of each class (beginner and expert) derived from the pre-trained BERT model. This method captures the contextual meaning of the texts and classifies them based on their proximity to the pre-defined centroids of each class. In the second step, we fine-tune BERT on our specific dataset to further enhance the model's performance for the classification task. This fine-tuning allows the model to learn nuances in text complexity, improving its accuracy in distinguishing between beginner and expert-level texts. Both methods—using the pre-trained BERT model and the fine-tuned model—are evaluated independently on data collected from the English and Simple English Wikipedia across several topics, with the results demonstrating that BERT's context-aware approach leads to a more robust and nuanced classification system.

ACM Reference Format:
. 2025. Assessing the Disciplinary Depth of a Text: A Transformer Based Approach and a new Dataset. In . ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

KEYWORDS

Binary Text Classification, Contextual Embeddings, BERT, Natural Language Processing (NLP), Semantic Similarity, Cosine Similarity, Expert vs. Beginner Text Classification, Word Frequency Distribution, Transformer Models, Embeddings, Text Complexity, Machine Learning for Text, Linguistic Features, Wikipedia Text Classification, Centroid, Accuracy, F1 score

1 INTRODUCTION

Personalized education and adaptive interfaces have the potential to revolutionize learning by tailoring content to the individual needs and expertise levels of learners. Traditional one-size-fits-all approaches often fail to engage learners effectively, especially when there are significant differences in their knowledge and skills. Adaptive systems, which adjust content based on a learner's proficiency, can improve engagement and learning outcomes across a range of applications, including online platforms, intelligent tutoring systems, educational games, and adaptive learning environments. These systems can also support automated feedback, personalized assessments, and dynamic course design, helping to optimize learning paths for each individual. Additionally, adaptive learning technologies can be applied in teacher training programs, where instructors' teaching styles and strategies can be fine-tuned to match the needs of their students.

The challenge, however, lies in accurately identifying the depth of a learner's expertise. Text classification is a foundational technology driving many of today's most widely used tools. From search engines that categorize web pages based on relevance, to AI assistants like ChatGPT that analyze and respond to a wide range of queries, text classification is integral to enabling intelligent and context-aware systems. Moreover, businesses rely on these techniques for tasks like spam detection, email filtering, customer feedback analysis, and more. However, most text classification systems are domain-specific, meaning they require significant adjustments or retraining to work across different topics or industries.

The recent advent of pre-trained transformer models, such as BERT, with attention mechanisms has significantly advanced text classification. While these models have shown impressive performance on topic-specific classification tasks, there is still a gap when it comes to generalizing classifiers that can distinguish texts based on their level of expertise—from beginner-level content to expert-level content—independently of the text's specific subject matter.

This paper seeks to design a text expertise classifier capable of distinguishing between texts that are suited for beginner or expert audiences, without being dependent on the specific domain or subject of the text. The proposed study evaluates the performance of contextual embeddings from BERT's pre-trained and fine-tuned language models. By considering a general level of expertise our classifier aims to improve upon existing systems that are often constrained by topic boundaries.

2 REAL-WORLD APPLICATIONS

The ability to classify text expertise on a general level, without being tied to specific topics, has wide-reaching applications. For instance:

- **Educational Tools:** Automatically categorizing content as either beginner or expert-level could help personalize learning materials for students, adapting content to their level of understanding or automating students' essay scoring.
- **Content Curation:** Platforms like news aggregators or knowledge-sharing websites (e.g., Stack Overflow, Wikipedia) could use such a system to recommend articles or tutorials

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

based on a user's expertise level, improving user engagement and experience.

- **Online Learning Platforms:** Online Learning Platforms can benefit from adaptive learning systems that adjust course material based on a learner's progress. For example, if a student struggles with a particular concept, the system can offer additional resources, exercises, or alternative explanations, while more advanced learners can skip ahead to more challenging content.

3 RESEARCH OBJECTIVES AND QUESTIONS

To address the challenge of creating a topic-independent text expertise classifier, we aim to investigate approaches based on contextual models. Specifically, we aim to design a text classifier using the BERT's language model, considering general level of expertise (beginner vs. expert).

This leads us to the following research questions:

1. **Research Question 1:** Can we leverage a pre-trained BERT model to classify text complexity, using its contextual embeddings to distinguish between beginner-level and expert-level texts?
   This question investigates whether the pre-trained BERT model, with its deep contextual understanding, can be used directly to classify texts based on their complexity, using cosine similarity between the embeddings of test texts and the centroids of predefined beginner and expert classes.
2. **Research Question 2:** Can we fine-tune a BERT model on a specific dataset to improve its performance in distinguishing between beginner and expert sections within multi-topic texts?
   This question explores whether fine-tuning BERT on a specialized dataset can enhance its ability to detect subtle nuances in text complexity, leading to more accurate classification of sections within multi-topic texts as either beginner or expert level.

4 RELATED WORK

We categorize previous work on the problem of identifying novice-expert texts into two broad classes: non-trained and trained text classifiers.

Non-trained classifiers are generally statistical methods [8] such as those based on word frequency distributions [2, 3, 5] or statistical properties like TF-IDF. They rely on feature extraction and statistical analysis. While they require no training, they struggle with generalization on more complex data due to a lack of semantic understanding, reliance on assumptions and an inability to capture nuanced text patterns or semantic relationships. For example, models like TF-IDF represent text as a bag of words (BoW), failing to incorporate contextual meaning or sentence structure [4].

In contrast, trained models such as Word2Vec, attempt to address some of these limitations by capturing semantic relationships between words based on their contextual usage. Word2Vec models words that frequently appear in similar contexts as semantically similar in the vector space. However, these methods suffer from unidirectional limitations, as they do not consider the full context of a word's usage in both directions. This issue is partly resolved by bidirectional models such as BiLSTMs, which represent words in a context-sensitive manner, allowing for a richer understanding of word meanings. Statistical feature extraction is often combined with machine learning models to improve classification accuracy, as seen in models like [6], where statistical methods augment the semantic capabilities of machine learning classifiers.

Attention-based text classifiers represent a more recent advance, with the introduction of the attention mechanism enhancing model interpretability and performance. Attention models assign higher importance (i.e., attention scores) to words that are more relevant for classification, such as highly polarized terms in binary classification tasks. For example, researchers investigated how attention scores are assigned to polarized words, showing that these words carry more weight in the model's decision-making process [7].

Recent work on large language models (LLMs) like GPT has also shown promising results in binary text classification tasks [1]. These models, while powerful, often require fine-tuning to achieve optimal performance. The combination of LLMs with statistical features has also been explored to further enhance classification performance. In particular, hybrid or joint models that integrate both contextual understanding from LLMs (such as BERT) and statistical features have gained attention. For example, a joint model approach to classify citizen complaints, combining BERT's contextual embeddings (e.g., [CLS] token) with keyword enhancement derived from word embeddings and TF-IDF scores [9]. This hybrid method improves classification accuracy by leveraging both contextual and statistical information, where the weighted sum of both embeddings is passed to a fully connected neural network (FCNN) for final classification.

5 METHODOLOGY

The first method is based on BERT-base-uncased which is a pre-trained transformer model that generates contextual word embeddings for text classification by considering the full sentence context, treating words case-insensitively. The BERT tokenizer converts text into tokens, making it compatible with BERT's input requirements. The second method, also based on BERT-base-uncased, has a classification head on top of the pre-trained BERT and fine-tunes the model end-to-end on the labeled train dataset using CrossEntropyLoss and the Adam optimizer, with early stopping to prevent overfitting. For both methods' classification, we extract the final generated [CLS] token embeddings representing each document, or each chunk of document if its token length exceeds BERT's 512 token limit. The [CLS] embeddings for "beginner" and "expert" texts are averaged to form class centroids. The calculated cosine similarity between the generated test document [CLS] embeddings and the class centroids predicts the class with the highest similarity.

5.1 Data Description

We introduce a dataset consisting of two types of texts: expert-level texts and beginner-level texts. The expert-level texts are sourced from the full English Wikipedia, while the beginner-level texts are sourced from the Simple English Wikipedia. The dataset contains two columns: text and label. The text column includes the actual text data, and the label column indicates the level of the text. A label of 0 corresponds to beginner-level texts, and a label of 1 corresponds to expert-level texts.

The dataset spans a variety of topic categories reflecting the breadth of content covered in Wikipedia, and each category is represented in both versions of the text, ensuring a broad and diverse selection of content. Topics may include, but are not limited to, science, history, arts, technology and economy.

The corpus contains 1030 texts and 515 texts per class. The average length of a beginner level text is 1490.19 words and the average length of an expert level text is 8435.12 words. The dataset can be found under github link: https://github.com/inesgoddi/text-expertise-model/tree/main/data_bert

5.2 Research Question 1: Can we leverage a pre-trained BERT model to classify text complexity, using its contextual embeddings to distinguish between beginner-level and expert-level texts?

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model that generates contextualized word embeddings by considering the full context of a word in a sentence, rather than just its immediate neighbors. The "base" version refers to the model's size, with 12 layers and 110 million parameters, and "uncased" indicates that the model does not differentiate between uppercase and lowercase letters (i.e., it treats "hello" and "Hello" the same).

The BertModel and BertTokenizer from HuggingFace's transformers library are used to load the pre-trained BERT model (bert-base-uncased) through which we obtain the hidden states of the model. The text is initially tokenized and split into non-overlapping chunks of a specified of 510 tokens. Each chunk is then enriched by the special tokens [CLS] at the start and [SEP] at the end, respecting BERT's 512 maximum token length. For each chunk, the model is run, and the embedding of the [CLS] token (the first token in each sequence) is extracted. This process is applied to all texts in the train dataframe, with the ability to filter based on labels (e.g., "beginner" vs. "expert"). The mean of the [CLS] embeddings is computed across all chunks to form a single vector that represents the entire documents for each label, it is computed by averaging the embeddings of the relevant chunks.

The same extraction and aggregation process is applied to the test dataset. For each document in the test set we extract the [CLS] embeddings of each chunk of the document. The embeddings are then aggregated by taking the mean of all [CLS] embeddings for each document. This results in a list of aggregated [CLS] embeddings and their associated labels (either "beginner" or "expert").

To evaluate the model's performance on the test set, we compare the aggregated [CLS] embeddings of test documents against the precomputed mean embeddings for "beginner" and "expert" labels, which were calculated on the training set. Both the test embeddings and the training embeddings (mean embeddings for "beginner" and "expert") are normalized to unit vectors to ensure fair cosine similarity computation. For each test document, the cosine similarity between the document's [CLS] embedding and the mean embeddings for both classes ("beginner" and "expert") is computed. The document is classified as belonging to the class (either "beginner" or "expert") with the higher cosine similarity to its [CLS] embedding. The classification is compared against the true label, and the count of correct predictions is tracked.

5.3 Research Question 2: Can we fine-tune a BERT model on a specific dataset to improve its performance in distinguishing between beginner and expert sections within multi-topic texts?

In this method, our aim is to fine-tune the previous model by creating a custom classification model, the BertClassifier class, which is built upon the pre-trained BERT model this time usingBertForSequenceClassification from Hugging Face, which includes BERT's architecture plus a classification head. A custom Dataset class is created to handle the chunking efficiently to respect the 512 tokens per input sequence limit: The raw text data is tokenized using the BERT tokenizer converting text into a sequence of tokens compatible with the BERT model. The chunk_tokens function splits tokenized text into smaller chunks, each containing a maximum of max length tokens (510 tokens in this case). This ensures the input sequences fit within BERT's 512 token limit after the addition of the special tokens [CLS] and [SEP] while preserving semantic meaning. The Dataset class constructs the training and validation datasets by iterating over the input training and validation DataFrames, tokenizing and chunking each document, and storing the tokenized chunks along with the corresponding labels and document IDs. These datasets will be used in the training and validation loops. The model takes the tokenized input and an attention mask to process the text then extracts the [CLS] token embedding from the last hidden state of BERT. A dropout layer is applied to the [CLS] embedding to prevent overfitting during training. The [CLS] embeddings of each chunk are passed to the classification head to output the logits for each class.

The training process is organized as follows: The DataLoader is responsible for batching the input data, applying padding to the token sequences, and ensuring each batch contains both input data and labels. This is handled via a custom collate_fn function. To handle larger batch sizes without exceeding memory limits, gradient accumulation is used. Gradients are accumulated over multiple steps and updated every accumulation steps batches. An early stopping mechanism is implemented to halt training if the validation loss does not improve after a certain number of epochs. This helps prevent overfitting and ensures efficient use of computational resources.

In the training loop, the model is trained for a specified number of epochs. The training loop performs the Forward Pass by processing the input then generating the output logits. The loss is computed using CrossEntropyLoss, and gradients are backpropagated for optimization. The accuracy of the model is calculated by comparing predicted labels to the ground truth. After accumulating gradients over multiple steps, the optimizer updates the model's parameters using the Adam optimizer.

The validation phase runs similarly to the training loop but without backpropagation. The validation loss and accuracy are computed after each epoch, and the model's performance is monitored to decide whether to stop training early.

At the last training epoch, the train function outputs the [CLS] embeddings of each chunk associated to its label. The mean [CLS] embeddings for each class are computed by averaging the embeddings of all training samples for that class.

The test set evaluation follows these steps: The Dataset class is used to process the test data, similar to the training data processing. For each test sample, we extract the [CLS] embedding using our fine-tuned BERT model. The classification decision for each test sample is based on the cosine similarity between the document's [CLS] embedding and the precomputed mean [CLS] embeddings for each class ("beginner" or "expert").The document is assigned to the class with the higher cosine similarity.

6 RESULTS

The accuracy and the F1 Score are used to evaluate both models.

The following table summarizes the classification accuracy of the different methods evaluated in this study: Pre-trained BERT (Centroid-based) and and Fine-tuned BERT (Centroid-based).

Method/Approach	Accuracy		
	Beginner	Expert	Overall
BERT (Pre-trained)	0.84	0.62	0.74
BERT (Fine-tuned)	0.95	0.98	0.96

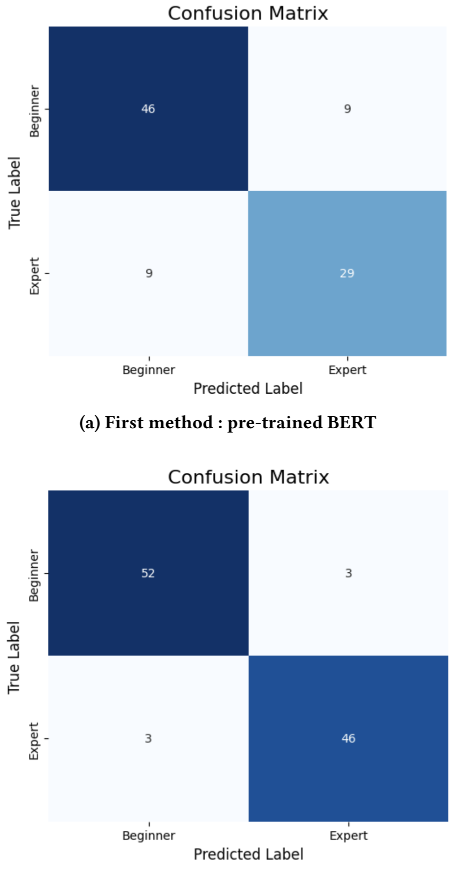
Table 1: Accuracy of Different Classification Methods

The following table summarizes the F1 Score of the different methods : Pre-trained BERT (Centroid-based) and Fine-tuned BERT (Centroid-based).

Method/Approach	F1 Score
BERT (Pre-trained)	0.73
BERT (Fine-tuned)	0.96

Table 2: F1 Score of Different Classification Methods

The following figures display the confusion matrix of each method: Pre-trained BERT (Centroid-based) and and Fine-tuned BERT (Centroid-based).



(a) First method : pre-trained BERT

(b) Second method : fine-tuned BERT

Figure 1: Confusion Matrix of each method

7 FUTURE WORK

In light of the promising results from analyzing contextual embeddings for expertise classification, several open questions remain that could lead to future research avenues:

(1) **Can we develop a hybrid model that integrates statistical features derived from Zipf's law with the contextual knowledge learned by BERT to create a more refined and accurate text classifier?**
This question explores whether combining the simplicity and mathematical properties of statistical models like Zipf with the deep contextual understanding of BERT could produce a model that leverages the strengths of both approaches, resulting in better classification accuracy.

(2) **Can we design a system that uses the semantic dimensions captured by BERT's embeddings to identify topic-specific expertise within multi-topic texts, distinguishing between beginner and expert sections?**
This question focuses on whether BERT's fine-grained representations of language can be leveraged to detect expertise levels within specific topics, even in texts that cover multiple

domains. By analyzing these embeddings, we could identify which sections of a multi-topic text are aimed at novice readers and which are intended for experts.

8 CONCLUSION

In this paper, we explored two approaches for binary text classification aimed at distinguishing texts based on their depth of expertise—whether they reflect a beginner or expert level of understanding—using the BERT model. The pretrained BERT model, with its pre-trained embeddings, provided a contextually sensitive approach by capturing semantic relationships between text elements, offering insights into the content's underlying meaning. The fine-tuned BERT, with its attention-based architecture, showcased the most advanced method by leveraging contextual embeddings to evaluate the nuanced depth of the text, considering both syntactic and semantic factors that contribute to the overall level of expertise.

Our findings indicate that while the pre-trained BERT based classifier approach can be useful for basic classifications, the fine-tuned BERT model outperforms it significantly by better capturing the depth of understanding within the text. This model is able to assess not just the superficial complexity of language but the underlying depth of knowledge and expertise embedded in the content. The results highlight the importance of moving beyond simple classification into a more sophisticated evaluation of expertise, particularly in applications such as educational tools, where distinguishing between varying levels of learner understanding is critical. Future research could focus on further fine-tuning these models to improve their ability to assess depth across a broader range of domains and more diverse datasets, ultimately leading to more accurate and context-aware classifiers for educational contexts.

REFERENCES

[1] Harika Abburi, Michael Suesserman, Nirmala Pudota, Balaji Veeramani, Edward Bowen, and Sanmitra Bhattacharya. 2023. Generative ai text classification using ensemble llm approaches. *arXiv preprint arXiv:2309.07755* (2023).

[2] Jaume Baixeries, Brita Elvevåg, and Ramon Ferrer-i Cancho. 2013. The evolution of the exponent of Zipf's law in language ontogeny. *PloS one* 8, 3 (2013), e53227.

[3] Roderick Edwards and Laura Collins. 2011. Lexical frequency profiles and Zipf's law. *Language Learning* 61, 1 (2011), 1–30.

[4] JRKC Jayakody, VGTN Vidanagama, Indika Perera, and HMLK Herath. 2023. Empirical Analysis for the Selection of Baseline Performances for Short Text Classification. In *2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS)*. IEEE, 335–340.

[5] Arghavan Moradi Dakhel, Michel C. Desmarais, and Foutse Khomh. 2021. Assessing developer expertise from the statistical distribution of programming syntax patterns. In *Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineering*. 90–99.

[6] Dadi Ramesh and Suresh Kumar Sanampudi. 2022. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review* 55, 3 (2022), 2495–2527.

[7] Xiaobing Sun and Wei Lu. 2020. Understanding attention for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3418–3428.

[8] Muthuraman Thangaraj and Muthusamy Sivakami. 2018. Text classification techniques: A literature review. *Interdisciplinary journal of information, knowledge, and management* 13 (2018), 117.

[9] Yuanhang Wang, Yonghua Zhou, and Yiduo Mei. 2023. A joint attention enhancement network for text classification applied to citizen complaint reporting. *Applied Intelligence* 53, 16 (2023), 19255–19265.