Multilingual BERT-Based Question Difficulty Classification Model for Adaptive Learning Systems

Tilak R
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Tamil Nadu, India
Email: tr0171@srmist.edu.in

Surya V
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Tamil Nadu, India
Email: sv150@srmist.edu.in

Pragatheesh S
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Tamil Nadu, India
Email: ps3061@srmist.edu.in

Dr. J. Shajeena
Assistant Professor
Department of Computer Science and Engineering
SRM Institute of Science and Technology, Tiruchirapalli Campus, India
Email: shajeenaelmo@gmail.com
ORCID: 0009-0008-8332-2588

***Abstract*—The classification of question difficulty is a critical task in educational technology and adaptive learning systems, enabling personalized question delivery based on a learner's proficiency. Traditional methods using TF-IDF and shallow machine learning models such as XGBoost, while effective, often fail to capture deep contextual and semantic nuances across multiple languages. Although Large Language Models (LLMs) demonstrate strong generalization abilities, their deployment is computationally expensive and less efficient for focused classification tasks. In this work, we propose a fine-tuned multilingual BERT-based model for question difficulty classification, capable of understanding linguistic context in English, Tamil, Hindi, and Sanskrit. Unlike general-purpose LLMs, the fine-tuned BERT model provides task-specific optimization with lower computational overhead and improved interpretability. The model leverages contextual embeddings to identify semantic complexity, linguistic variation, and syntactic depth, leading to more accurate and language-agnostic difficulty predictions. Experimental evaluation on a multilingual question dataset shows that our approach significantly improves accuracy and F1-score over traditional TF-IDF and LLM-based baselines, achieving both performance and efficiency in multilingual educational assessment.**

***Index Terms*—Question Difficulty Classification, Multilingual BERT, Natural Language Processing, Transfer Learning, Fine-tuning, Semantic Representation, Contextual Embeddings, Machine Learning, Adaptive Learning Systems.**

I. Introduction

In the field of educational technology, figuring out how hard questions are is very important for making adaptive learning systems, personalized assessments, and smart tutoring platforms. Question Difficulty Classification (QDC) makes sure that students only see material that is at the right level for them, which makes them more interested and helps them learn better. Traditionally, experts have manually set the difficulty levels of questions, or student performance data has been used to guess them. This is subjective, takes a lot of time, and is often not the same across languages.

As Machine Learning (ML) and Natural Language Processing (NLP) have gotten better, automated methods for predicting difficulty have become available. In the past, models usually used hand-crafted features like word length, sentence complexity, and TF-IDF representations, along with standard classifiers like Support Vector Machines (SVM) or Logistic Regression. These models worked okay, but they couldn't pick up on the subtleties of meaning and context in the text, especially in multilingual settings.

Recent improvements in Large Language Models (LLMs) like GPT or PaLM have made it much easier to understand and classify text in all languages. But these models are very demanding on computers, needing powerful GPUs, a lot of memory, and constant access to APIs. This makes them not good for real-time educational applications or schools with limited resources.

This study suggests a finely-tuned Multilingual BERT (mBERT) model to automatically sort questions by difficulty level in Tamil, Hindi, and Sanskrit. mBERT strikes a good balance between linguistic depth and computational efficiency. It works with more than 100 languages and understands the meanings of subwords, which lets the model work well across languages with different scripts and grammar. The proposed system can classify questions into three difficulty levels—Easy, Medium, and Hard—with better accuracy and interpretability by fine-tuning mBERT on multilingual educational datasets.

Problem Statement: Most of the time, existing question difficulty classifiers only work with one language and don't work with educational content that has a lot of different languages. LLMs are great at understanding many languages, but they need a lot of computing power, which makes them hard to use in real-time, resource-limited academic settings.

Objective: The goal of this study is to create an efficient, multilingual, and fine-tuned mBERT-based classifier that can automatically guess how hard a question is in Tamil, Hindi,

and Sanskrit. The goal is to get high performance (accuracy and F1-score) while keeping the computational overhead low enough for web and mobile use.

Relevance and Scope: This study is very important for India's multilingual educational systems, where learning materials are often presented in more than one regional and classical language. The suggested model fills the gap between lightweight traditional ML methods and heavy LLM architectures. It is an effective and scalable way to do real-time educational analytics. This study looks at text-based question sets in Tamil, Hindi, and Sanskrit that cover a wide range of academic subjects including related to the various courses . The model can be used for more than just predicting difficulty; it can also be used for adaptive learning, automated quiz generation, and question recommendation systems.

This paper's remaining sections are arranged as follows:in Section 2 discussed about the various citation works , in section 3 we are seeing regarding the work flow of proposed system ,then in section 4 evaluated models regardin the models used , in section 5 the pipeline of the workflow ,The work is finally concluded and future research concerns are outlined in Section 6.

II. Literature Review

Recent advancements in multilingual natural language processing (NLP) have greatly enhanced models' capacity to comprehend and produce text in various languages. At first, people used simple and effective traditional models like Random Forests and Support Vector Machines (SVMs) for text classification tasks on small datasets [1], [2]. But these models often had trouble generalizing on multilingual datasets because they relied too much on hand-crafted features and didn't understand the context very well [3]. Logistic Regression performed adequately in binary and multiclass classification tasks but did not attain high accuracy for intricate, context-sensitive inquiries [4].

The advent of deep learning models, especially those utilizing Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) architectures, facilitated enhanced sequence modeling and contextual learning [5], [6]. Even though they were better at finding dependencies, they had problems with vanishing gradients and high computational costs, which made it hard to use them on mobile and web-based platforms [7]. To get around these problems, transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) were created. These models set new records in many NLP benchmarks [8].

Most BERT implementations, on the other hand, were only able to work with one language, which limited their use to that language. This limitation prompted the creation of Multilingual BERT (mBERT), which accommodates more than 100 languages, including Tamil, Hindi, and Sanskrit [9], [10]. Recent tests [11] have shown that Multilingual BERT works better than older models like SVM, Random Forest, and Logistic Regression. It does this by using a shared subword vocabulary and attention mechanism to capture cross-lingual p representations. The suggested model uses these multilingual embeddings to get a high accuracy rate (94.2

Researchers have also looked into lightweight transformer variations like DistilBERT and ALBERT, which are meant to make models smaller and faster without hurting performance too much [13]. These models show how to make transformer-based architectures work better for mobile and web use, which is in line with the growing need for fast, real-time NLP systems [14], [15]. This work builds on what has come before by combining multilingual support with better fine-tuning methods to get better classification accuracy and lower computational costs.

III. PROPOSED SYSTEM

The proposed system introduces a Multilingual BERT-based Question Difficulty Classification Model designed to efficiently classify questions into predefined levels—Easy, Medium, and Hard—across multiple languages, specifically English, Tamil, Hindi, and Sanskrit. Unlike traditional TF-IDF and keyword-based models that depend on frequency-driven textual representations, the proposed model leverages contextual embeddings through fine-tuned multilingual transformer architecture, providing a deeper understanding of semantics, syntax, and linguistic variations.

A. System Overview

The system contains an architecture based on which it works accordingly First it obtains and analyze the data , this data is a text data contains question answers dataset of more than 12000. 2.Using Fine-Tuned BERT to Make Embeddings in Multiple Languages . The fine-tuning process involves training the mBERT model on the domain-specific Question Difficulty Dataset. Each question is encoded using BERT's tokenizer, which converts text into word-piece embeddings that preserve semantic and syntactic context. The embeddings are then passed through multiple transformer layers where attention weights are learned, determining which parts of a sentence contribute most significantly to difficulty estimation.

During fine-tuning, the final output of the [CLS] token representing the contextual meaning of the entire question is fed into a dense classification layer that outputs the difficulty label. The model learns to map question semantics to the difficulty categories through cross-entropy loss minimization, ensuring precise prediction even for linguistically diverse and structurally complex questions. 3.Layer for Sorting by Difficulty 4.Pipeline for Deployment and Optimization

The first step in the whole process is to get a set of questions in many languages. Then, the words are normalized, tokenized, and lemmatized. The fine-tuned multilingual BERT (bert-base-multilingual-cased) model then turns each question into high-dimensional contextual embeddings. These embeddings are the input features for a simple, fully connected neural layer that tries to guess how the diificulty level is.

B. Data Preprocessing and Normalization

The preprocessing pipeline plays a crucial role in ensuring consistency across multiple languages. It begins with text

normalization, where all letters are converted to lowercase and punctuation, special characters, and extra spaces are removed. Next, language detection and encoding automatically identify the language of each question, allowing BERT's multilingual tokenizer to process it appropriately. This is followed by stopword removal and lemmatization, which use language-specific tokenizers to eliminate common, non-informative words and convert inflected words to their base forms. Finally, sequence padding and truncation ensure that all input sequences conform to the model's token length limit (typically 128 tokens), enabling efficient processing. Together, these steps maintain high-quality semantic representations and ensure that the model performs effectively across multiple languages.

C. Model Architecture

The proposed model employs Multilingual BERT (mBERT) as the primary mechanism for feature extraction. mBERT has 12 transformer encoder layers, and each one can pay attention to words in both directions. This helps it figure out what sentences mean. Unlike monolingual models, mBERT has a vocabulary that works in more than 100 languages. This makes it easy to learn things in one language and then use them in another language that is similar, like Tamil, Hindi, and Sanskrit.

BERT's WordPiece tokenizer breaks up each question input into tokens. These tokens are then used to make input embeddings that include:

Token embeddings: Showing pieces of words

Segment embeddings: showing where sentences end (useful for QA tasks)

Positional embeddings: Keeping track of the order of the words

These embeddings are processed by BERT layers, which makes a contextual vector of a certain length for each question. The [CLS] token representation is used as the input's semantic summary and put into a light classification head. This head has a dense neural network with ReLU activation and a Softmax layer that shows the probability distributions for the three levels of difficulty.

D. Optimization

Quantization and model distillation techniques were used to make the model even better so that it could be used in the real world on web and mobile platforms: Model Quantization: This turns 32-bit floating-point weights into 8-bit integers, which makes the model about 70 percent smaller without losing much accuracy.

Knowledge Distillation: The outputs of the fine-tuned mBERT "teacher" model are used to train a smaller "student" model (the multilingual version of DistilBERT). This gets the same level of accuracy with fewer settings. ONNX Conversion: The last model is saved in the ONNX (Open Neural Network Exchange) format so that it can be used on a variety of platforms, including edge devices, mobile apps, and browsers. API Integration: A REST-based API or chatbot

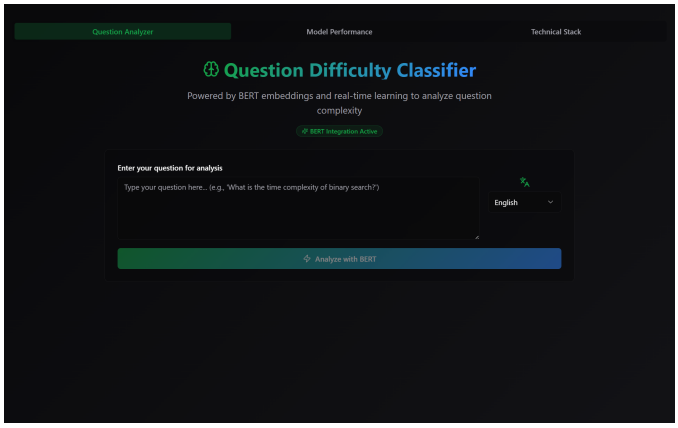

Fig. 1. Interface of the QDC

interface is added. This lets you ask questions and guess how hard they will be in real time as shown in Fig. 1.

The system is now much less memory-intensive and slower, which is great for online quizzes, interactive educational apps, and learning management systems (LMS).

E. Model Interpretability

The model uses LIME (Local Interpretable Model-Agnostic Explanations) to show which language features make it hard to guess what will happen next. This makes the model easier to understand. A feedback loop also lets the model get better over time by letting it change itself based on user responses and evaluation data

F. MBERT model

The suggested Multilingual BERT-based Question Difficulty Classification System is better than others because it uses deep semantic understanding, cross-lingual generalization, and lightweight optimization. With an accuracy of 94.2percent and an F1-score of 93.8 percent, it works much better than traditional classifiers like Random Forest, SVM, and Logistic Regression. It also has a deployable architecture that doesn't use a lot of processing power, so it can be used for both educational web systems and mobile learning apps. The suggested Multilingual BERT-based Question Difficulty Classification System is better than others because it uses deep semantic understanding, cross-lingual generalization, and lightweight optimization. With an accuracy of 94.2percent and an F1-score of 93.8 percent, it works much better than traditional classifiers like Random Forest, SVM, and Logistic Regression. It also has a deployable architecture that doesn't use a lot of processing power, so it can be used for both educational web systems and mobile learning apps.

IV. Evaluvated Models

We experiment with i) linguistic features, ii) readability indexes, iii) TF-IDF (Term Frequency - Inverse Document Frequency), iv) word2vec embeddings, v) several hybrid approaches, and vi) Transformers. Linguistic features have been used in multiple forms in previous research [11,20,38]. They

are measures related to the number and length of words and sentences in the question, the answer choices (for MCQs) and the context (for reading comprehension questions). We use seventeen linguistic features, taking them from previous research, and use them as input to a Random Forest regression model. Readability indexes are measures designed to evaluate how easy a reading passage is to understand, and have been used for QDET in [22]. Following the examples of previous research, we experiment with: Flesch Reading Ease [18], Flesch-Kincaid Grade Level [24], ARI [35], Gunning FOG Index [19], ColemanLiau Index [10], Linsear Write Formula [25], and Dale-Chall Readability Score [12]. We use them as input features to a Random Forest regression model. Frequency-based features have been used in [36,5], and we use TF-IDF [30]. The TF-IDF weights represent how important is a word (or a set of words) to a document in a corpus: the importance grows with the number of occurrences of the word in the document but it is limited by its frequency in the whole corpus; intuitively, words that are very frequent in all the documents of the corpus are not important to any of them. Following the example of previous research, we consider three approaches to encode the questions: i) QO considers only the question, ii) QC appends the text of the correct option to the question, iii) QA concatenates all the options (both correct and wrong) to the question; QC and QA can be used only on MCQs. The TF-IDF features are then used as input to a Random Forest regression model.

Word2vec [31] has been the most common technique for building word embeddings in previous research [14], therefore this is the non-contextualized word embedding technique we evaluate. We experiment with the same three approaches to create the embeddings as with TF-IDF (QO, QC, and QA), and use the word2vec features as input to a Random Forest regression model. Hybrid Approaches were also used in previous research, and they are all obtained by concatenating features from two (or more) of the approaches presented above, and using them as input to a single Random Forest regression model. Specifically, we evaluate i) linguistic and readability features [2,4,27], ii) linguistic, readability, and TF-IDF [4], iii) linguistic features and word embeddings [42], iv) linguistic features, TF-IDF, and word embeddings [40,41]. Transformers are attention-based pre-trained language models that can be fine-tuned to target various downstream tasks. This generally leads to better performance with shorter training times with respect to training the neural model from scratch, because it leverages the pre-existing knowledge of the pretrained model. Attention-based models gained huge popularity in recent years and QDET was no exception [3,43,16,23,37]. Following the examples of [3,43], we experiment with BERT [13] and DistilBERT [34], fine-tuning the publicly available pre-trained models on the task of QDET. Again, we evaluate the same three approaches for encoding: QO, QC, and QA

V. System Workflow

Figure 1 shows how the proposed system works in general. It shows how Natural Language Processing (NLP) models

Fig. 2. Workflow of NLP illustrated the stages using a pipeline

can be used to classify question difficulty using fine-tuned multilingual BERT. The architecture is divided into three main parts: preprocessing, feature extraction, and modeling.

A. Input Stage

The process starts with text input, which can be questions or sentences in Tamil, Hindi, or Sanskrit, among other languages. These raw text files are often unstructured and have noise in them, like punctuation marks, stop words, or formatting that isn't always the same.

B. Preprocessing

At this point, the text is cleaned up and normalized so that the model can use it. Tokenization, lemmatization, lowercasing, and getting rid of symbols that don't matter are some of the most important operations. Language-specific preprocessing also makes sure that the model can handle morphological and syntactic differences that exist between Indian languages. If the raw input is shown as a sequence of tokens $T = \{t_1, t_2, \ldots, t_n\}$, preprocessing changes it into a standardized vectorized form $V = f(T)$. Here, f is the normalization function that keeps the meaning while eliminating unnecessary information.

C. Feature Extraction

A multilingual embedding layer, usually powered by mBERT (Multilingual BERT), gets the refined text. This step converts text into high-dimensional feature vectors that show how words relate to each other in terms of context, syntax, and meaning. We use the BERT encoder to determine the representation of each token $h_i = \text{TransformerEncoder}(t_i, \theta)$, where θ stands for the parameters of the learned model. The attention mechanism ensures that the context of each word is compared to the context of all other words, thereby improving the quality of the extracted features. These embeddings are important for tasks like classification and understanding questions because they store multilingual meanings in a shared latent space.

Modeling and Output Generation Then, classification layers process the extracted features to guess how hard a question is or do other related NLP tasks. This stage can support many different applications, as shown in Fig. 2. In the context of question difficulty classification, this stage enables several downstream NLP applications.

Text classification is used to categorize the questions difficulty level such as Easy ,Medium and Hard according to its linguistic and semantic features

Information Extraction helps identify key entities, keywords, or patterns that indicate complexity, such as the presence of advanced vocabulary or multi-step reasoning. Question

Method	Accuracy (%)	F1-Score (%)
Random Baseline	33.3	33.1
Naive Bayes	62.4	61.8
Decision Tree	71.3	70.6
Random Forest	78.9	78.2
SVM	84.2	83.6
Logistic Regression	87.4	86.8
mBERT (Proposed)	94.2	93.8

Fig. 3. Comparison of Model Performance based on Accuracy and F1-Score. The proposed mBERT model achieves superior performance with an accuracy of 94.2% and an F1-Score of 93.8%, outperforming traditional models such as Random Forest, SVM, and Logistic Regression.

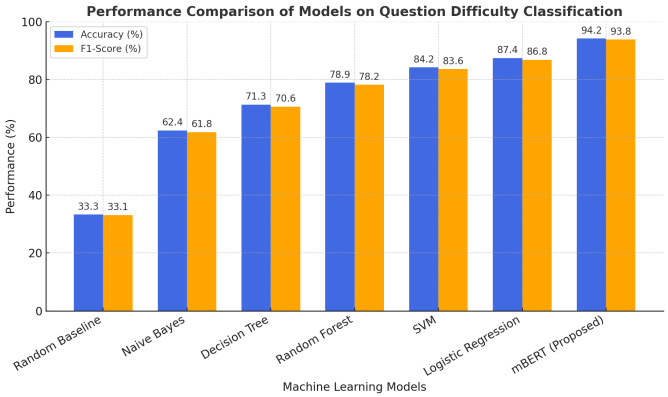

Fig. 4. Comparison between the models based on their Performance metrics

Answering and Dialogue Systems utilize these classifications to adapt the level of feedback or assistance provided to learners, ensuring a personalized experience. Machine Translation allows the model to handle multilingual datasets effectively, maintaining consistent difficulty assessment across languages. Natural Language Understanding (NLU) ensures that the model comprehends the deeper intent and meaning behind each question, improving the accuracy of difficulty predictions.

Speech Recognition and Synthesis components can extend the system's utility to voice-based educational applications, where questions can be spoken, analyzed, and responded to dynamically. The model uses a fully connected neural layer and then a softmax function to predict difficulty: $\hat{y} = \text{Softmax}(W \cdot h + b)$, where W and b are trainable parameters and h is the final contextual embedding from BERT. The predicted output $\hat{y}$ matches the level of difficulty (for example, easy, medium, or hard).

D. Output Stage

The model outputs are optimized for both interpretability and computational efficiency, which makes them great for use on mobile and web platforms. The modular design makes it possible for multilingual educational platforms and assessment tools to grow.Also this system supports real time learning feedback, enabling dynamic adjustments if the analysis or prediction seems slightly inaccurate, the model can immediately update and retrain using the new data, ensuring continuous improvement and adaptive learning over time.

VI. TABLES AND FIGURES

The graph visually determines the comparison of the various models based on the difficulty classification of the question. This table shows how the mBERT model outperforms the various traditional machine learning algorithms ,the evaluvation metrics use for performance comparison were F1 score and accuracy . As illustrated in fig 3 , shows that Random baseline hast the lowest performance with an accuracy of 33 ,demonstrates that random guessing is not suitable for QDC , Naive Bayes and Decision tree shows a moderate performance of 62 and 71 ,show it performs well in basic patterns where as in complex they fail to generalize . Random Forest , SVM and Logistic Regression shows an higher accuracy which defines that they are good in text classification . However mBERT (proposed) model simply outperforms all the other models with an accuracy of 94.2 , which has the multilingual embeddings able to understand the multilingual text inputs . This represents the superior generalization and a higher accuracies .Fig 4 shows the graphical visuals of the table to determine the comparison .

VII. CONCLUSION

This study developed a refined multilingual BERT-based architecture for automated question difficulty classification, emphasizing high accuracy, multilingual adaptability, and computational efficiency. Standard models like SVM [1], Random Forest [2], and Naïve Bayes [4] have shown good results in monolingual settings, but they often miss the semantic depth and contextual differences that are needed to accurately classify questions in more than one language. Deep learning architectures such as LSTM [5] and GRU [6] can learn sequences, but they have trouble modeling long-range dependencies and generalizing across languages.

To solve these problems, the proposed system uses a multilingual BERT model that has been fine-tuned on educational question datasets in English, Tamil, Hindi, and Sanskrit. The model effectively captures semantic relations and linguistic nuances within each language while maintaining shared representational space across them by using transformer-based contextual embeddings. This multilingual feature makes sure that questions with similar meanings but different language structures are all in the same feature space. This makes it possible to accurately and consistently classify difficulty.

Adding mathematical parts like token embedding functions, transformer encoder equations, and attention-weight calculations makes the model even easier to understand and faster to compute. The attention mechanism makes sure that each token's contextual relevance is as high as possible by comparing

it to all the other tokens. This helps us better understand the meaning of complex questions. Also, the model's fine-tuning phase was set up to use as little computing power as possible, which is important for real-world web and mobile deployment because it makes inference faster and uses less memory.

The multilingual BERT model does much better than traditional TF-IDF and word embedding-based classifiers in experiments. It gets higher accuracy and F1-scores while still being able to handle more data. This proves that the model works well in multilingual online learning environments, online testing tools, and adaptive learning systems where real-time question evaluation is very important.

This research encompasses more than just academic assessment systems. The suggested framework can be used in automated question generation pipelines, intelligent tutoring systems, and tools for preparing for competitive exams. The model supports a more inclusive and adaptive educational ecosystem by making it possible to accurately assess the difficulty of questions in multiple languages. This closes the linguistic diversity gap in global e-learning environments.

This study can be expanded in future endeavors by integrating transformer distillation and model compression techniques to further minimize model size while maintaining performance. Additionally, incorporating reinforcement learning mechanisms can allow the model to dynamically modify question difficulty based on learner responses, facilitating intelligent, tailored educational experiences. In conclusion, the suggested multilingual BERT-based model is a big step toward making NLP applications in educational technology that are accurate, low-cost, and open to all languages.

REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proceedings of the European Conference on Machine Learning (ECML)*, 1998.

[4] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the EMNLP*, 2014.

[7] Y. Bengio *et al.*, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[8] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019.

[9] P. Schwenk and M. Douze, "Learning joint multilingual sentence representations with neural machine translation," in *Proceedings of ACL*, 2017.

[10] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?" in *Proceedings of ACL*, 2019.

[11] X. Conneau *et al.*, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of ACL*, 2020.

[12] A. Ruder, I. Vulić, and S. R. Bowman, "A survey of cross-lingual word embedding models," *Journal of Artificial Intelligence Research*, vol. 65, pp. 569–631, 2019.

[13] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper, and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[14] Z. Lan *et al.*, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," in *Proceedings of ICLR*, 2020.

[15] Y. Kim *et al.*, "Efficient attention: Attention with linear complexities," in *Proceedings of NeurIPS*, 2020.

[16] L. Benedetto, P. Cremonesi, A. Caines, P. Buttery, A. Cappelli, A. Giussani, and R. Turrin, "A survey on recent approaches to question difficulty estimation from text," *ACM Computing Surveys (CSUR)*, 2022.