HCLTech | Supercharging Progress™

Application of transfer learning in iLog Analyzer using BERT models

Table of contents

Abbreviations

Short Form	Abbreviation
BERT	Bidirectional Encoder Representations from Transformers
CLS	Classification Token
iLA	Intelligent Log Analyzer
LLM	Large Language Model
MLM	Masked Language Modelling
NSP	Next Sentence Prediction
PAD	Padding Token
SEP	Separator Token
SLA	Service Legal Agreement
SME	Subject Matter Expert

Introduction

In recent times, the proliferation of code has resulted in the generation of large volumes of log files across numerous applications, services, systems, devices, networks and more. Analyzing these log files to identify the issues, determining the root cause and proposing solutions is essential. iLog Analyzer(iLA), an AI tool, leverages ML and Deep Learning (DL) algorithms to analyze log files and detect anomalies, identify root causes and recommend fixes. Despite its capabilities, the tool has limitations that require additional effort for deployment across different applications.

This paper explores the multifaceted applications of large language models (LLMs), shedding light on their integration across various domains and emphasizing the advantages of integrating the Bidirectional Encoder Representations from Transformers (BERT) model into the existing iLA framework. By considering the adoption of LLMs, particularly BERT, for immediate problem resolution, this paper aims to demonstrate how BERT-pre-trained LLMs can address the current challenges and potentially revolutionize the iLA.

Business challenges

In today's rapidly evolving landscape of technology and customer service, swiftly and effectively addressing challenges is more critical than ever. The support and maintenance teams face the pressing task of reducing the effort and resources required to analyze the log files, triage them to the right team and resolve them within the constraints of the Service Legal Agreement using available engineers. The iLA tool addresses these challenges but has certain limitations that mandate additional investment for deployment. Specifically, the tool must be pre-configured to understand the log file data structure and structure them into the format that iLA can process. Additionally, iLA AI models can only predict based on the data they are trained on, lacking contextual learning capabilities.

Introducing pre-trained LLMs can address these limitations by enabling contextual learning through transfer learning, thus facilitating quicker deployment with reduced Subject Matter Expert (SME) effort. This contribution aims to fuel the ongoing discussion on how the BERT pre-trained model can transform real-time issue resolution, ushering in a new era of enhanced customer experiences and operational efficiency through the fundamentals of transfer learning.

Problem statement

A significant challenge in the existing iLA tool is the inability to learn new data without input from SME's effort. This limitation arises from iLA's difficulty in capturing the semantic meaning and relational context from the available complex log data, which varies widely. Consequently, iLA's prediction accuracy diminishes with new data, as its ML algorithms rely on text similarity to the learned data, lacking the depth of contextual and masked learning. This shortfall necessitates continuous SME feedback to improve the learning process incrementally, even for similar errors occurring in a different context.

This paper addresses the limitation of iLA in understanding the contextual relationship within the data by leveraging transfer learning using the BERT language model powered by the transformer architecture. Integrating transfer learning into our error analysis pipeline highlights our commitment to leveraging cutting-edge techniques and methodologies to address the evolving challenges of log analysis in today's digital landscape. The BERT pre-trained model employs a transformer encoder that simultaneously processes all the words in a sentence through bidirectional learning, allowing the model to understand the word's context based on the surroundings.

Solution

Conventional AI algorithms that rely on sequential learning methods often need help to capture complex relationships in log data. LLMs like BERT offer a solution by employing self-attention mechanisms to better understand relationships and nuances within the data. This streamlines the process of identifying errors and root causes and recommending solutions to help resolve the errors promptly.

Applying BERT LLMs and employing self-attention mechanisms through transfer learning enables iLA to better understand the relationships and nuances within the data. This approach streamlines error identification and root causes and recommends solutions promptly.

In our proposed approach, transfer learning using the BERT pre-trained model is a pivotal strategy for increasing the effectiveness of error log analysis. By utilizing pre-trained language models such as BERT, which is trained on a vast corpora of text, we can use its comprehensive understanding to enhance the learning of complex relationships in error log data.

The pre-trained BERT model allows us to tap into the wealth of semantic patterns and nuances gathered from diverse textual data, accelerating the learning process. Using insights from a broader domain of language understanding significantly enhances our error analysis capabilities.

Why use the BERT pre-trained model for iLA

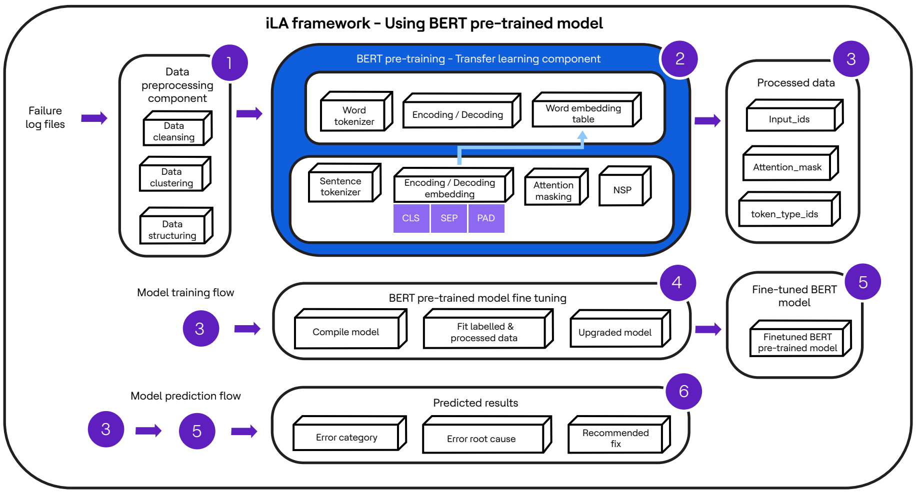
Transfer learning is the key to the BERT pre-trained model and occurs in two main phases: pre-training and fine-tuning. BERT is pre-trained on an extensive corpus of unsupervised text data using Masked Language Modelling (MLM), allowing it to grasp contextual relationships and nuances bidirectionally.

This enables BERT to understand language structure and semantics without requiring prior knowledge of new data. Using BERT, iLA can effectively learn and understand complex relationships in error lags from various scenarios. For example, 'Unable to connect to the database, invalid credentials' and 'Unable to connect to the database, the DB server is not reachable' can be comprehended bidirectionally with BERT understanding the data surrounding each word.

This pre-trained knowledge is transferred to iLA enhancing its ability to interpret complex log data across different scenarios. Moreover, through masked learning, iLA can accurately analyze and predict new data with similar error messages and minor differences based on past learning, incorporating multi-intelligence into the iLA framework.

Contextual learning architecture of iLA

LLM BERT pre-trained model architecture embedded in iLA AI model framework. With this architecture, iLA can capture the semantic meaning and the relative position and learn the contextual information in the given log data through transfer learning.



The log messages are structured and cleansed before input into the transfer learning component depicted in the architecture. The pre-trained model performs the following steps, utilizing the learning acquired on a large corpus.

First, word tokenization is applied using the BERT tokenizer, on which the BERT model was trained, to split each log message into word tokens. These tokens are assigned a unique ID from the tokenizer vocabulary creating a word embedding table as part of the encoding and decoding process.

Second, the sentence tokenization process splits the log messages into sentences. Special tokens Classification Token (CLS) (indicating the start of the sentence) and Separator Token (SEP) (marking the beginning of the following sentence) process the input data. Padding Token (PAD) tokens pad sentences to a uniform length, ensuring input consistency, which is crucial for the BERT model.

Attention masking is then applied to control which parts of the input sequence should focus on during processing. It assigns different weights to tokens, indicating their importance in the input sequence. The model then calculates the weighted sum to prioritize relevant tokens irrespective of their position. Positional embeddings are added to each token to indicate its position in the sequence.

Next Sentence Prediction (NSP) is applied when the input sequences have multiple sentences to understand the contextual relationship within the log messages. Additional information, such as segment labels, is added to the embeddings to achieve this. BERT uses segment embeddings to distinguish between different segments of input. Tokens from the first sentence receive a segment embedding of 0, whereas the tokens from the second sentence will have a pre-defined embedding of 1.

Data vectorization is performed by converting tokenized, embedded and segmented error messages into numerical vectors and numerical IDs and organizing them into tensors or arrays.

Finally, batching and loading occur by organizing and grouping multiple log messages to optimize computational resources and speed up the training or prediction process.

BERT pre-training – Transfer learning in iLA

Transfer learning occurs during the pretraining phase of the BERT model, which utilizes unsupervised data.

The BERT pre-trained model transfers learned knowledge to processing structured and unlabelled log data.

Several key steps are involved in this phase to facilitate learning from the input data. First, the text data undergoes tokenization, split into individual tokens or sub-words.

Next, each token is embedded into a high-dimensional vector space, capturing semantic information and contextual relationships.

The model then undergoes training through tasks like MLM, predicting masked tokens based on context and NSP and predicting if two sentences follow each other. These tasks help the model understand language context and coherence. Finally, token embeddings and positional encodings process through a multilayer bidirectional transformer encoder.

This encoder utilizes self-attention mechanisms to capture nuances in the data and understand the context. The model effectively learns to understand language through these steps, making it well-suited for downstream tasks.

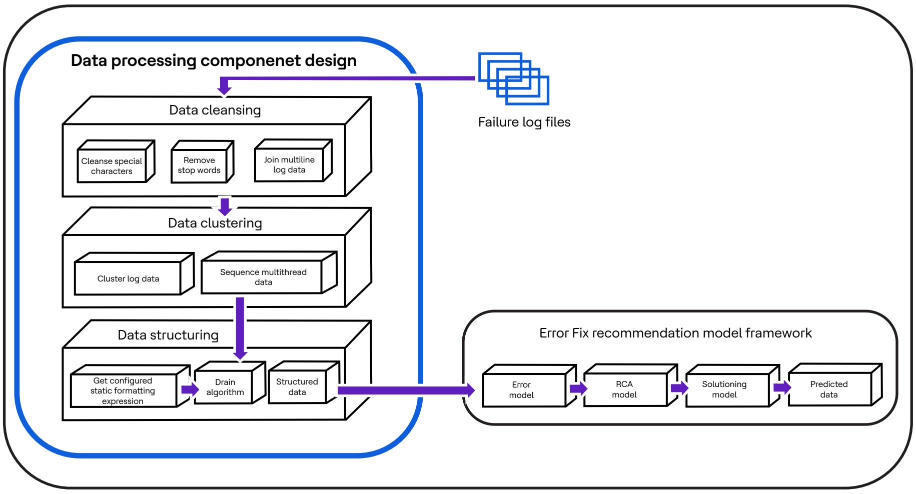
BERT model fine tuning

Subsequently, BERT undergoes fine-tuning on smaller labeled datasets for specific downstream tasks such as sentiment analysis or named entity recognition. This process involves adjusting the pre-trained BERT parameters to align more closely with the intricacies of the target task.

Leveraging the knowledge distilled during pretraining, BERT efficiently adapts to new tasks, enhancing its performance across diverse Natural Language Processing applications.

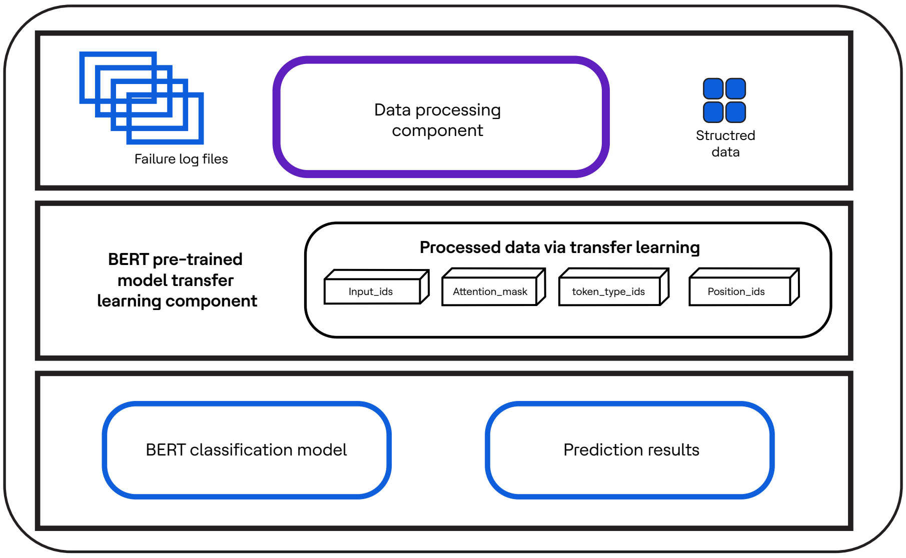
BERT pre-trained model - Inference Process Flow

In the existing iLA, the data preprocessing component only performs data cleansing, clustering and structuring. The structured data is then passed directly to the model framework for prediction.



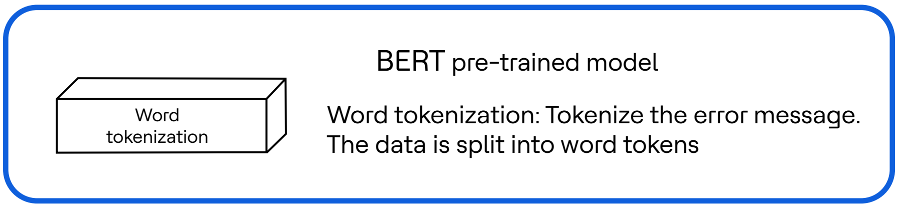
However, the model cannot learn the nuance of the language and the contextual information in the sentences available in structured data. It needs to understand the relationship between the data and, because of this limitation, every new data source requires the SME's feedback to learn and predict.

Flow of input data from data processing component to model framework in iLA with BERT pre-trained LLM

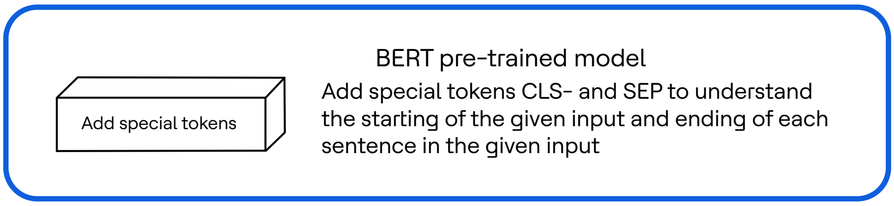


The log files flow into the data processing component and get processed and structured data is received as output.
The structured data then passes into the transfer learning component and the error messages are further processed to understand the contextual relationship.

Example - An error message from the structured data is demonstrated.
Error message: "Database connection failed. Invalid username/password"



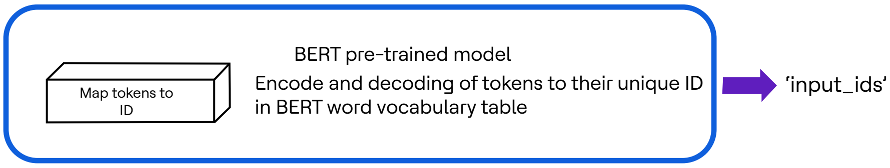
Tokenized error messages: ['database', 'connection', 'failed', '.', 'invalid', 'user', '##name', '/', 'password']



Tokenized error messages with special tokens added.
[[CLS]'database', 'connection', 'failed', '.', [SEP] 'invalid', 'user', '##name', '/', 'password'[SEP]]

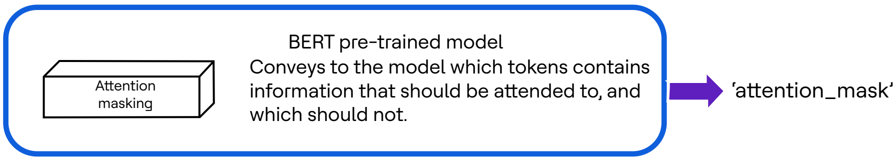
Find the max length of the error messages in the given structured data. The length of all the error messages should be equal for BERT to process and understand the data. Hence, padding is performed on the uneven data to ensure that the length of all the error messages is equal. PAD token is applied to pad all the error messages to max length. Let us assume that the max length is 20.

Tokenized error messages with special tokens PAD added.
[[CLS]'database', 'connection', 'failed', '.', [SEP] 'invalid', 'user', '##name', '/', 'password'[PAD][SEP]]



Encoded token IDs for given error message: [101,7809, 4434, 3478, 102,1011, 19528, 5310, 18442, 1013, 20786, 102, 0,0,0,0,0,0,0,0]

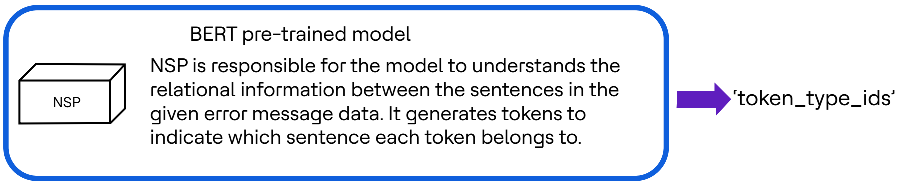
[CLS] uses 101 and [SEP] uses 102. 0s are added to structure the error message to the maximum length.



Encoded token IDs for given error message: [101,7809, 4434, 3478,102, 1011, 19528, 5310, 18442, 1013, 20786, 102, 0,0,0,0,0,0,0,0]

Attention masks for the encoded token IDs: [1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0]

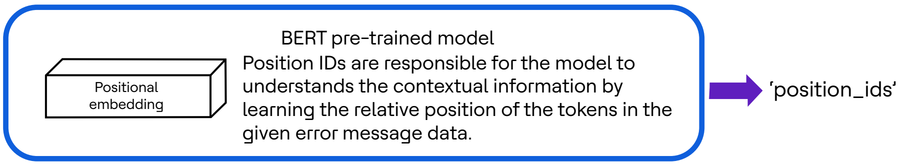
The attention mask communicates to the model that IDs with '1' should be considered for learning and computing and those with '0' should not be considered for learning. This way, the model ignores the padded data.



Encoded token IDs for given error message: [101,7809, 4434, 3478, 102, 1011, 19528, 5310, 18442, 1013, 20786, 102, 0,0,0,0,0,0,0,0]

token_type_ids of NSP for the encoded token IDs: [0,0,0,0,1,1,1,1,1,1,1,1]

The NSP generates the IDs representing the sentence sequence in the given input. This helps the model identify the relationship between the sentences and decide the prediction by learning this relationship. '0' indicates that the tokens belong to the first sentence and '1' indicates that the token belongs to the second sentence, etc.

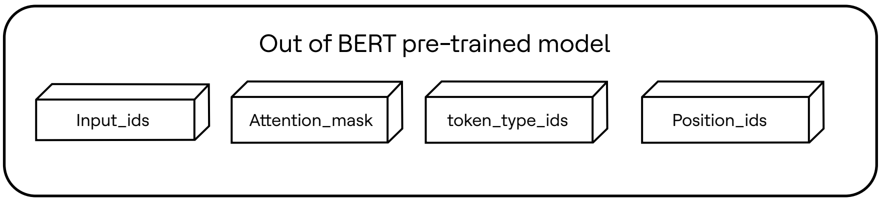


Encoded token IDs for given error message: [101,7809, 4434, 3478, 102, 1011, 19528, 5310, 18442, 1013, 20786, 102, 0,0,0,0,0,0,0,0]

position_ids of positional embedding for the encoded token IDs: [0,1,2,3,0,1,2,3,4,5,6,7]

Positional embeddings provide a positional value for the token in each sentence of the error message. This allows the model to understand the token position in the given error message data and helps identify the word's meaning in line with the given context.

This is an optional parameter, and when it is not provided, the system automatically generates a positional embedding to offer it to the model.



The input_ids, attention_mask, token_type_ids and Position_ids generated for the given data capture the semantic relationship through the transfer learning process, gaining more intelligence to understand the nuance of the language and contextual information. This will enable iLA to help understand new data and help predictions based on similar data learned in the past. The model can correlate the past learned data to predict the new data. This feature will help iLA overcome the limitation of missing and capturing the semantic relationship and contextual information of the language data provided for prediction. Fine tuning considerations for BERT pre-trained model

Fine-tuning the BERT pre-trained model for domain-specific tasks with labeled data requires carefully selecting hyper-tuning parameters and fine-tuning techniques. The model architecture plays a crucial role in achieving high performance. Inappropriate training data selection can lead to overfitting, mainly if the target dataset is small and the number of parameters is large. To mitigate this, proper learning techniques must be applied. One challenge for iLA is the need for labeled data. An active learning technique for fine-tuning the BERT-based classification model addresses this challenge. Conducting a detailed study on this area will enhance the performance and accuracy of the BERT-based.

Advantages of applying BERT in iLA

The key advantages of applying the BERT pre-trained model in iLA include acquiring transfer learning, contextual proficiency and knowledge transfer. These benefits enable iLA to enrich its understanding of the relationship between the data and the contextual information, which will help decide the correct label to predict.

Contextual proficiency: BERT's bidirectional contextual understanding will enable iLA to comprehend sentence meaning and relationships, elevating the precision of breaking and categorization tasks.

Knowledge transfer: By commencing with a pre-learned BERT model, iLA gains from knowledge transfer, where the model applies insights acquired from a broad spectrum of language data.

Conclusion

The existing iLA AI models must be trained from scratch, leveraging labeled datasets for every application and domain. By adopting the BERT pre-trained model in iLA, the pre-trained model facilitates us to tap into the wealth of semantic patterns and nuances the model has gathered from its training on diverse textual data. By doing so, we effectively shortcut the learning process, using the insights garnered from the broader domain of language understanding to support our error analysis capabilities.

Integrating the BERT pre-trained model, iLA can capture semantic, contextual relationships and patterns within the input text, enhancing its prediction performance on new data. BERT's proficiency in understanding the context enables iLA to make nuanced predictions when fine-tuned on small-labeled datasets for every domain-specific task.

By fine-tuning BERT on a small set of labeled error datasets, iLA can predict the error types, error route causes and possible fixes. Each prediction function yields a label corresponding to the predicted class, providing interpretable and actionable insights into log errors, possible causes and potential fixes.

The model's ability to generalize knowledge enables it to make informed predictions on unseen error messages, enhancing its utility for real-world applicability. However, to maximize the prediction accuracy, mastering advanced fine-tuning techniques, selecting the appropriate method and acquiring commendable knowledge when selecting training data are crucial.

Finally, with BERT pre-trained model integration, iLA can be poised to redefine how to swiftly address challenges across diverse industries and find the right balance between efficiency, ethical considerations and user satisfaction.

References

- Understanding BERT — Word Embeddings | by Dharti Dhami | Medium
- https://huggingface.co/google-bert/bert-large-uncased
- Glossary — transformers 2.11.0 documentation (huggingface.co)
- https://medium.com/@bijit211987/advanced-techniques-for-fine-tuning-llms-46f849c6ece8
- https://web.stanford.edu/class/cs224n/final-reports/final-report-169728058.pdf
- https://arxiv.org/abs/1810.04805v2
- https://www.kaggle.com/code/au1206/fine-tuning-bert-text-classification
- https://medium.com/@yashvardhanvs/classification-using-pre-trained-bert-model-transfer-learning-2d50f404ed4c

Author information

Sasikala Sethu

Sasikala Sethu is working at HCLTech, as an AGM with PTS Solutions CoE. She has vast experience in Microsoft Technologies, Data Science and Analytics and a passion for training. She has more than 24 years of experience in IT with expertise in providing solutions to any given problem statement and architecting the solution with loosely coupled design. Her interest lies in ideating and implementing AI solutions to solve industry problems.

Srinivasan S

Srinivasan S is a mechanical engineer who graduated in 2022 with experience of one year and expertise in Python and SQL. His interest lies in exploring the fields of data analytics, ML and AI. He is particularly interested in understanding the intricacies of ML, DL algorithms and LLMs.

HCLTech | Supercharging Progress™

hcltech.com

HCLTech is a global technology company, home to 222,000+ people across 60 countries, delivering industry-leading capabilities centered around Digital, Engineering and Cloud powered by a broad portfolio of technology services and software. The company generated consolidated revenues of $12.3 billion over the 12 months ended December 2022. To learn how we can supercharge progress for you, visit hcltech.com.