# Task 1.2.1 - Cellula Technologies NLP Internship
# BERT Family
# Short Research Article

Abdelrahman Elaraby

September 2025

## Contents

# 1 Introduction

In these days, Large Language Models (LLMs) has become a trend for their brilliant effectiveness in solving various tasks. And to understand these models we can't ignore a cornerstone in the development timeline of these models which is BERT. It was a model developed by Google in 2018 utilizing the Transformer Architecture by Vaswani et al. in 2017. But why Transformers in the first place? Traditional Natural Language Processing (NLP) architectures like recurrent neural networks (RNNs) and Long-Short-Term-Memory (LSTMs) were good at capturing the word meaning and sequential dependencies; However, they struggled with long term context, parallelization, and scalability. That's why Transformers with their Attention Mechanisms were a turning point, and Models like BERT came to light. Nonetheless, they had their own challenges with regards to their size and computational requirements for real-world deployment, especially in low-resource environments like mobile devices or time-sensitive applications. To address these challenges, researchers have tried to develop variants of BERT but more compact and utilizing lesser resources. Two Prominent variants are DistilBERT and ALBERT, which will be discussed in this article.

# 2 Background on Transformers

The main edge of transformers is that they can process an input sequence of text in parallel compared to sequential processing done earlier by RNNs and LSTMs. But how is that? it is done by the Attention Mechanism. For example, in the sentence "The cat that was chased by the dog ran away." a transformer can directly capture the connection between "cat" and "ran" without having to pass through all the intermediate words.

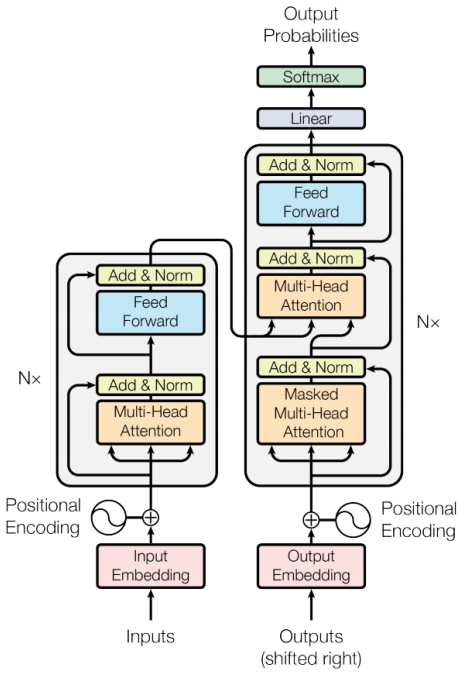


Figure 1: Diagram of a transformer. Image source: *Attention is All You Need* paper.

In the broad sense, the original transformer architecture was composed of two parts, Encoder and Decoder as shown in Figure 1. In the encoder there is a multi-head self-attention, while in the decoder there is the multi-head cross-attention in addition to the self-attention. For the self-attention, the input tokens are projected into queries (Q), keys (K), and values (V). then y computing attention scores (using equation 1), the model determines how much focus each word should place on others in the sequence.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

So, after the attention sub-layer the tokens are represented by new vectors that carries the context with their embeddings; and following that there is another sub-layer which is a feed forward network (FFN) to enhance

those representations more and embed more knowledge to the vectors based on the task the model is tuned on. Residual connections and layer normalization are added after the attention and FFN to stabilize training. Note that because transformers do not process data sequentially like RNNs, they can fully exploit parallel computation, making training on large datasets much faster. Positional encoding is added to the initial embedded tokens since this is the only way to make the transformer aware about the order of tokens. This process of Attention followed by FFN is repeated $N_x$ layers; the original transformer had 6 layers in each of the encoder and decoder. The output of the last layer of the encoder is fed to the cross-attention part of the decoder, where the Queries come from the decoder while the Keys and Values come from the encoder. This Cross-attention in the decoder lets each output token attend to the encoder's hidden states, so the decoder can condition its generation on the input sequence.

# 3 BERT: Bidirectional Encoder Representations from Transformers

## 3.1 Architecture of BERT

BERT is an Encoder-Only Model, which means it only uses the encoder stack of the transformer. BERT comes in different sizes, such as BERT-base (12 layers, 110M parameters) and BERT-large (24 layers, 340M parameters). Unlike other sequence-to-sequence models, and since BERT is an Encoder-Only model, it is not designed to generate texts but rather to produce rich and contextualized embeddings that can be fine-tuned for downstream tasks like classification, sentiment analysis, and question answering.

## 3.2 Pretraining Objectives

BERT was pretrained on a massive corpus (Wikipedia + BookCorpus) and became a learned general-purpose model that can be fine-tuned for downstream tasks. The effectiveness of this model comes from the objectives it was trained on, which were two. The first was Masked Language Modeling (MLM), where a random subset (recommended is 15%) of the tokens are hidden and replaced by a [MASK] token, and the objective of the model is to uncover or predict the original hidden words. The second was Next Sentence Prediction (NSP), the task if the model is to predict whether the second sentence logically follow the first (a binary classification task); this helped BERT capture the inter-sentence relationships.

# 4 DistilBERT

## 4.1 Motivation

Despite the groundbreaking results of BERT in NLP tasks, its large size made it a resource-intensive model and not so practical in real-time application having constrained resources. This led researchers at Hugging Face to propose DistilBERT, a smaller and faster variant of BERT obtained through model compression.

## 4.2 Knowledge Distillation Process

The idea behind Knowledge Distillation is to train a small model (called "student") to mimic a larger model (Called "Teacher"). Instead of training from scratch on ground-truth labels, the student is optimized through a composite loss function consisting of three parts:

1. **Distillation Loss (Soft Targets Loss)**: The student is trained to match the teacher's output probability distribution over the vocabulary. By applying a softmax with temperature scaling, the student captures not only the correct prediction but also the relative probabilities of other words, reflecting the teacher's richer knowledge (generalization capabilities). These small differences in output probabilities over the vocabulary (this uncertainty) is sometimes referred to as the "dark knowledge".

2. **Masked Language Modeling Loss (MLM Loss)**: As in the original BERT, random tokens in the input are masked, and the student predicts the missing words. This ensures the student develops a strong bidirectional language model from the training corpus.

3. **Cosine Embedding Loss (Hidden State Alignment)**: To align internal representations, the student's hidden states are encouraged to resemble those of the teacher. This is achieved by minimizing the cosine distance between corresponding hidden layers of both models. Since BERT-base has 12 layers and DistilBERT

has only 6, each student layer is aligned with every second teacher layer, allowing hidden state similarity to be computed despite the architectural differences.

By combining these three objectives, DistilBERT manages to retain much of BERT's representational power while being significantly smaller and faster. This multi-part loss ensures that the student does not merely approximate outputs, but also inherits the teacher's deeper internal structure and contextual understanding.

### 4.3 Model Characteristics

The DistilBERT reduces the number of parameters by about 40% compared to BERT-base. This is done by reducing the number of layers from 12 (in BERT-base) to 6 (in DistilBERT). Notably, this makes it 60% faster at inference time, while still retaining around 97% of BERT's performance on major NLP benchmarks. Importantly, the DistilBERT still preserves the bidirectional nature of BERT and is trained using Masked Language Modeling (MLM) but not the Next Sentence Prediction (NSP), which is shown to be less critical. So, to sum it up, DistilBERT finds a good balance between efficiency and accuracy making it a highly practical resort for many developers aiming to deploy the models in a resource-constrained environment.

## 5 ALBERT (A Lite BERT)

### 5.1 Motivation

As discussed in the previous section, one of BERT's major limitations is its heavy computational and memory cost, which motivated the researcher to create DistilBERT reducing inference time and model size. However, DistilBERT focused mainly on compression and speed instead of addressing the issue of parameter redundancy in the design of BERT. To tackle this different aspect of inefficiency, Google Research proposed ALBERT (A Lite BERT), which reduces redundancy at the architectural level, enabling the creation of lighter yet deeper models.

### 5.2 Parameter Reduction Techniques

ALBERT introduces two main innovations to reduce redundancy in BERT's architecture:

1. **Factorized Embedding Parameterization**: In BERT, the size of the word embedding matrix grows with both the vocabulary size (V) and the hidden dimension (H). ALBERT decouples these by factorizing the embeddings into two smaller matrices: one for mapping words to a lower-dimensional embedding space (E), and another for projecting them to the hidden size used in the model. This significantly reduces the number of parameters in the embedding layer while maintaining the same representational capacity. In other words, the original vocabulary embedding matrix of size $V \times H$ can be decomposed into a pair of smaller matrices: one of size $V \times E$ and another of size $E \times H$. As a result, instead of requiring $\mathcal{O}(V \times H)$ parameters, the decomposition reduces the complexity to $\mathcal{O}(V \times E + E \times H)$. This method is particularly effective when $H \gg E$. the concept is shown in Figure 2.

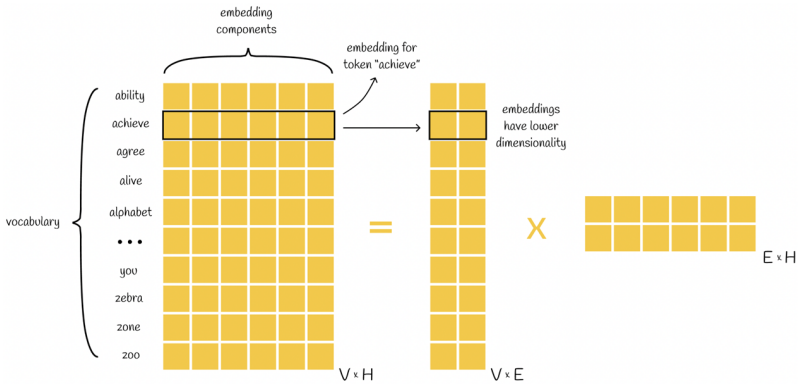


Figure 2: Illustration of factorized embedding parameterization (Source: *Towards Data Science*, "ALBERT" [link])

2. **Cross-Layer Parameter Sharing**: Instead of each transformer layer having its own unique parameters, ALBERT shares the same parameters across layers. This enormously reduces the number of trainable parameters and improves generalization. In practice, there are three options for parameter sharing; the first being the weights of attention and the second being the feed-forward network weight and the third being both attention and FFN sharing. This allows the model to scale to more layers without proportionally increasing parameter count.

3. **Sentence Order Prediction (SOP)**: To replace BERT's Next Sentence Prediction (NSP), which often relied on topic similarity rather than true discourse understanding, ALBERT introduces SOP. In this task, the model distinguishes between correctly ordered consecutive sentences and sentences where the order has been swapped. This encourages learning of inter-sentence coherence and improves performance on tasks requiring discourse-level understanding.

Together, these techniques make ALBERT much more memory-efficient, enabling deeper models with fewer parameters.

### 5.3 Model Characteristics

So, by incorporating the discussed ideas and smart design choices, ALBERT achieves strong performance while using significantly fewer parameters compared to BERT. For example, ALBERT-base has only 12 million parameters compared to the 110 million parameters in BERT-base, yet ALBERT-base performs competitively on known benchmarks like GLUE and SQuAD. Moreover, Larger variants such as ALBERT-xxlarge show that with efficient parameterization extremely deep models can be trained without an incredible memory cost. Furthermore, ALBERT removed the Next Sentence Prediction (NSP) objective and replaced it with a Sentence Order Prediction (SOP) objective, which better captures inter-sentence coherence. Overall, ALBERT shows that by carefully rethinking architectural efficiency, it is possible to scale NLP models in depth while keeping memory usage and training time manageable. In Table 1 you can find details and comparisons between the different sizes of ALBERT and BERT.

Overall, ALBERT shows that by carefully rethinking architectural efficiency, it is possible to scale NLP models in depth while keeping memory usage and training time manageable. In Table 1 you can find details and comparisons between the different sizes of ALBERT and BERT.

Table 1: Comparison of BERT and ALBERT model sizes

| **Model** | **Parameters** | **Hidden Size (H)** | **Layers (L)** | **Embedding Size (E)** |
|---|---|---|---|---|
| BERT Base | 110M | 768 | 12 | 768 |
| BERT Large | 340M | 1024 | 24 | 1024 |
| ALBERT Base | 12M | 768 | 12 | 128 |
| ALBERT Large | 18M | 1024 | 24 | 128 |
| ALBERT XLarge | 60M | 2048 | 24 | 128 |
| ALBERT XXLarge | 235M | 4096 | 12 | 128 |

## 6 Conclusion

In this report, we talked about the basics of transformers, a turning point for building NLP models that can process text sequences in a parallel manner. Following that we illustrated BERT, a pioneering model that utilized the rich bidirectional context learned by the transformer architecture. Building on that we discussed two variant models built to overcome the high-resource demands of BERT. The first was DistilBERT, which used the concept of knowledge distillation to compress the model into a smaller model. The second was ALBERT, which used techniques like factorization and parameter sharing to reduce the number of parameters needed in the model. Both variant models collectively illustrate the rapid advancement of transformer architectures in NLP research and practical applications; they demonstrate the trade-offs between model size, efficiency, and performance.

# References


[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of NAACL-HLT*, 2019.

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

[4] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," *International Conference on Learning Representations (ICLR)*, 2020.

[5] Hugging Face, "Smaller, faster, cheaper, lighter: Introducing DistilBERT, a distilled version of BERT," Medium, 2019. [Online]. Available: https://medium.com/huggingface/distilbert-8cf3380435b5

[6] Towards Data Science, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," 2020. [Online]. Available: https://towardsdatascience.com/albert-22983090d062