



New chat



The paper introduces **BERT (Bidirectional Encoder Representations from Transformers)**, a new language representation model designed to pre-train deep bidirectional representations from unlabeled text. Unlike previous models like OpenAI GPT, which use unidirectional language models, BERT is trained to jointly condition on both left and right context in all layers. This bidirectional approach allows BERT to achieve state-of-the-art results on a wide range of natural language processing (NLP) tasks by fine-tuning with just one additional output layer.



Key Contributions:

1. **Bidirectional Pre-training:** BERT uses a **masked language model (MLM)** objective, where some tokens in the input are randomly masked, and the model predicts the original vocabulary id of the masked word based on its context. This allows the model to learn deep bidirectional representations.
2. **Next Sentence Prediction (NSP):** BERT also includes a **next sentence prediction task**, where the model predicts whether one sentence follows another. This helps the model understand relationships between sentences, which is crucial for tasks like question answering and natural language inference.
3. **State-of-the-Art Performance:** BERT achieves new state-of-the-art results on 11 NLP tasks, including GLUE, MultiNLI, and SQuAD, with significant improvements over previous models.

Model Architecture:

- BERT is based on the **Transformer** architecture, specifically a multi-layer bidirectional Transformer encoder.
- Two model sizes are reported: **BERTBASE** (12 layers, 768 hidden units, 12 attention heads, 110M parameters) and **BERTLARGE** (24 layers, 1024 hidden units, 16 attention heads, 340M parameters).
- BERTBASE is designed to be comparable in size to OpenAI GPT for fair comparison, but BERT uses bidirectional self-attention, whereas GPT uses constrained (left-to-right) self-attention.



1810.04805v2.pdf

PDF 757KB

Extract and explain contents

Pre-training Tasks:

1. Masked Language Model (MLM):

- 15% of tokens are masked, and the model predicts the original token.
- To avoid a mismatch between pre-training and fine-tuning, the masked token is replaced with:
 - [MASK] token 80% of the time.
 - A random token 10% of the time.
 - The original token 10% of the time.

2. Next Sentence Prediction (NSP):

- The model predicts whether two sentences are consecutive or not.
- This task helps the model understand sentence relationships, which is useful for tasks like question answering and natural language inference.

Fine-tuning:

- BERT can be fine-tuned for various downstream tasks by simply adding a task-specific output layer.
- Fine-tuning is computationally efficient and can be done in a few hours on a single GPU or TPU.
- BERT achieves state-of-the-art results on tasks like GLUE, SQuAD, and SWAG by fine-tuning the pre-trained model on task-specific data.

Experiments:

- **GLUE:** BERT achieves significant improvements over previous models, with BERTLARGE pushing the GLUE score to 80.5%.
- **SQuAD:** BERT achieves new state-of-the-art results on both SQuAD v1.1 and v2.0, with improvements in F1 scores.
- **SWAG:** BERT outperforms previous models on the SWAG dataset, which evaluates commonsense reasoning.

Ablation Studies:

- The paper includes ablation studies to understand the impact of different components of BERT, such as the importance of the NSP task and the effect of model size.
- The results show that both the MLM and NSP tasks are crucial for BERT's performance, and larger models consistently improve performance across tasks.

Conclusion:

- BERT demonstrates that deep bidirectional pre-training is highly effective for a wide range of NLP tasks.
- The model's ability to fine-tune with minimal task-specific modifications makes it a powerful tool for NLP applications.

Implementation Details:

- BERT is pre-trained on a combination of the BooksCorpus (800M words) and English Wikipedia (2,500M words).
- The model is trained using Adam optimization with a learning rate of 1e-4, and fine-tuning is done with task-specific learning rates.

Comparison with Other Models:

- BERT is compared to **ELMo** and **OpenAI GPT**. Unlike ELMo, which uses a feature-based approach, BERT and GPT are fine-tuning approaches. However, BERT's bidirectional nature and the inclusion of the NSP task give it a significant advantage over GPT.

Code and Pre-trained Models:

- The code and pre-trained models for BERT are available at <https://github.com/google-research/bert>.

In summary, BERT represents a significant advancement in NLP by introducing a deep bidirectional pre-training approach that achieves state-of-the-art results across a wide range of tasks. Its ability to fine-tune with minimal task-specific modifications makes it a versatile and powerful tool for language understanding.

explain it step by step with an example



Let's break down **BERT (Bidirectional Encoder Representations from Transformers)** step by step with an example to make it easier to understand.

Step 1: Problem BERT Solves

Traditional language models like **OpenAI GPT** are **unidirectional**, meaning they only consider the context from left-to-right (or right-to-left). This limits their ability to fully understand the meaning of words in a sentence because they can't see the full context from both directions. For example, in the sentence:

"The bank of the river was crowded with people."

The word "**bank**" could mean either the side of a river or a financial institution. A unidirectional model might misinterpret it because it only sees the left context ("The bank of the..."). BERT solves this by being **bidirectional**, meaning it looks at both the left and right context simultaneously.

Step 2: BERT's Architecture

BERT is based on the **Transformer** architecture, which uses **self-attention** to process input sequences. Here's how it works:

1. Input Representation:

- BERT takes a sequence of tokens (words or subwords) as input.
- Each token is represented by three embeddings:
 - **Token Embedding:** Represents the word itself.
 - **Segment Embedding:** Indicates whether the token belongs to sentence A or sentence B (useful for tasks involving pairs of sentences).
 - **Position Embedding:** Represents the position of the token in the sequence.
- Example: For the sentence "The bank of the river was crowded with people," BERT breaks it into tokens like ["The", "bank", "of", "the", "river", "was", "crowded", "with", "people"].

2. Bidirectional Self-Attention:

- BERT uses **self-attention** to process the entire sequence at once, allowing each token to attend to all other tokens in the sequence.
 - This means the word "bank" can attend to both "The" (left context) and "river" (right context), helping it understand the correct meaning.
-

Step 3: Pre-training BERT

BERT is pre-trained on two tasks:

Task 1: Masked Language Model (MLM)

- BERT randomly masks 15% of the tokens in the input sequence and tries to predict the masked tokens based on the surrounding context.
- Example:
 - Input: "The [MASK] of the river was crowded with people."
 - BERT predicts the masked word: "bank."
- To avoid overfitting to the [MASK] token, BERT sometimes replaces the masked word with:
 - A random word (10% of the time).
 - The original word (10% of the time)

Task 2: Next Sentence Prediction (NSP)

- BERT is trained to predict whether one sentence follows another.
 - Example:
 - Sentence A: "The bank of the river was crowded with people."
 - Sentence B: "People were enjoying the sunny weather."
 - BERT predicts whether Sentence B logically follows Sentence A.

Step 4: Fine-tuning BERT

After pre-training, BERT can be fine-tuned for specific tasks by adding a task-specific output layer. Here's how it works for different tasks:

Example 1: Sentence Classification (e.g., Sentiment Analysis)

- Task: Classify the sentiment of a sentence as positive or negative.
 - Input: "The movie was fantastic!"
 - BERT processes the sentence and uses the [CLS] token's representation (a special token added at the beginning of the input) to predict the sentiment.

Example 2: Question Answering (e.g., SQuAD)

- Task: Given a question and a passage, find the answer in the passage.
 - Input:
 - Question: "What is the capital of France?"
 - Passage: "France is a country in Europe. Its capital is Paris."
 - BERT processes the concatenated sequence of the question and passage and predicts the start and end positions of the answer ("Paris").

Example 3: Named Entity Recognition (NER)

- Task: Identify entities like names, locations, or organizations in a sentence.
 - Input: "John lives in New York."
 - BERT processes the sentence and predicts labels for each token:
 - "John" → PERSON
 - "lives" → O (not an entity)
 - "in" → O
 - "New York" → LOCATION

Step 5: Why BERT Works So Well

1. Bidirectional Context:

- BERT looks at both left and right context, which helps it understand the meaning of words more accurately.
- Example: In "The bank of the river," BERT can see both "The" and "river" to understand that "bank" refers to the side of a river.

2. Transfer Learning:

- BERT is pre-trained on a large corpus of text (BooksCorpus and Wikipedia), so it learns general language representations.
- During fine-tuning, it adapts these representations to specific tasks with minimal additional training.

3. Versatility:

- BERT can be fine-tuned for a wide range of tasks (e.g., classification, question answering, named entity recognition) without significant changes to its architecture.

Step 6: Example Walkthrough

Let's walk through an example of fine-tuning BERT for **question answering** (SQuAD):

1. Input:

- Question: "What is the capital of France?"
- Passage: "France is a country in Europe. Its capital is Paris."

2. Preprocessing:

- BERT combines the question and passage into a single sequence:

[CLS] What is the capital of France? [SEP] France is a country in Europe. Its capital is Paris. [SEP]

Copy

- The `[CLS]` token is used for classification, and `[SEP]` separates the question and passage.

3. BERT Processing:

- BERT processes the sequence and generates contextual embeddings for each token.

4. Output:

- BERT predicts the start and end positions of the answer in the passage.
- In this case, it predicts the start position at "Paris" and the end position at "Paris."

5. Result:

- The answer is "Paris."

Step 7: Key Takeaways

- BERT is **bidirectional**, meaning it looks at both left and right context, which helps it understand language better.
- It is pre-trained on two tasks: **Masked Language Model (MLM)** and **Next Sentence Prediction (NSP)**.
- BERT can be fine-tuned for various NLP tasks by adding a task-specific output layer.
- It achieves state-of-the-art results on tasks like question answering, sentiment analysis, and named entity recognition.

By combining deep bidirectional pre-training with fine-tuning, BERT provides a powerful and flexible framework for a wide range of NLP applications.



New chat

Message DeepSeek

DeepThink (R1)

Search

U

↑

AI-generated, for reference only