What is a transformer in ML?

A transformer is a neural network that collects and processes sequential data (like the words in a sentence) as it comes in and transforms one sequence into another sequence.

Understanding how transformers work is a bit more complicated. Before we explain transformers in more detail, though, it's important to cover some key concepts.

First, it's helpful to remember the background of neural networks in NLP. In the past, technologists used recurrent neural networks (RNNs) and Long-Short Term Memory (LSTM) to process language sequentially, similar to transformers. Before transformers were introduced, many technologists used RNNs during language model training since RNN models can learn to use past input and predict what comes next in a sequence.

However, one of the problems with RNN models is that as a sentence becomes longer, the more input that's received and the gap between what's relevant and the model's ability to use it when it's needed widens. When this happens, the model is less likely to successfully predict what comes next.

Transformers solve this problem. Using a technique known as self-attention, transformers focus on distinct parts of the input text during each step of processing sequential data and assign weight to the significance of each part of the sequence. This helps it focus on what matters most. Transformers can also process words simultaneously, drastically improving training speed.

Overview of transformer architecture in NLP

The transformer model architecture breaks down into key components that play a significant part in how a transformer works:

Encoder-decoder architecture

The transformer model is based on an encoder-decoder architecture. The encoder processes the sequential input and creates a representation. The decoder uses the representation to generate an output.

Attention mechanism

The attention mechanism looks at an input sequence and assigns a weight and value to each word based on relevance to distinguish the importance of specific words.

Self-attention

Self-attention is an attention mechanism that compares the different elements in a sequence input, looking for relationships and dependencies between elements to help compute their positions in the output. Unlike other language models, transformer models rely entirely on self-attention.

Multi-head attention

Rather than relying on an isolated attention mechanism, transformers involve multiple parallel attention layers within different layers of the transformer. Each head processes different parts of the input sequence and focuses on representations that have different semantic or syntactic meanings.

Masking

Masking refers to the way transformers can hide future positions in the input sequence so it only pays attention to the words that came before it in the sequence. The decoder can only attend to what it has seen so far, not what's to come.

Here’s how an individual attention mechanism works. When a transformer model receives an input sequence, it associates each element of the sequence with abstract vectors, which represent their semantic meaning.

The model uses three types of vectors:
Query (q)
Key (k)
Value (v)

These vectors are obtained by multiplying input vectors with weight matrices learned during model training. The query vector (q) represents the current word. The key vector (k) works as an index to find the most similar match and identify information that is relevant to the query. The value vector (v) includes the actual information and details about the query.
To find similarities between the query and key, transformers multiply each element of the vector to calculate its dot product and measure similarity. The higher the dot product, the more similar the elements are perceived and the lower the dot product, the more dissimilar the elements are. Next, a softmax function converts the dot product values into a probability distribution on a scale of zero to one, with zero indicating low similarity and one being high similarity.
Finally, the softmax distribution is multiplied by the value vector (v) to assign greater importance to vectors that have high similarity in the comparison of (q) and (k) to help the model focus on the most relevant information. This creates an attention score that helps guide both the learning and output.

Transformer advantages in NLP
Transformers offer many advantages over alternative NLP models, including:

Input length
RNNs often have short memories and start to forget previous inputs, even in long sentences. Transformers can handle larger input sequences because of their self-attention layers and ability to analyze words in parallel.
Prediction accuracy
Because an NLP transformer can distinguish these dependencies, it can more accurately assign context and meaning.

Pre-training and training techniques
hile practitioners still use large text corpora to train transformer models, transformers require less input during training to successfully perform NLP tasks than other neural networks. Pre-training can also help transformers capture knowledge and contextualized representations to fine-tune smaller, task-specific labeled data, reducing overall training time.

Examples of NLP tasks using transformers
Here are a few examples to better understand the tasks transformers are used for and how they can help during model training and pre-training.

1. Using transformers can improve meaning clarity
Words at the beginning of a sentence may impact words at the end of a sentence.

Transformers are particularly useful for analyzing longer sentences and connecting long-range dependencies.

Example: “I saw a statue standing in front of the store with binoculars.”

The sentence itself is ambiguous. Was the statue looking at the speaker with binoculars or was the speaker looking at the statue? Transformer models look at the broader context to improve clarity.

Revised: “I saw a statue standing in front of the store with binoculars and zoomed in on it.”

This example shows how transformers can capture the dependency and context of an input sequence, understanding that the speaker was using a pair of binoculars to look at the statue.

2. Improve sentiment analysis with transformers
Transformers can also help improve sentiment analysis by understanding the correct context.

Example: “The food at the restaurant was good, but the service was terrible.”

This sentence uses mixed sentiments to describe different aspects of the restaurant experience. Without the proper context, some language models may struggle to assign the correct sentiment. With transformers, however, the context can be captured more effectively.

Revised: “The food at the restaurant was OK, but the service was terrible; I would not recommend itHere, the overall sentiment is negative.

3. Fine-tuning responses during pre-training
Technologists can train transformer models using large data sets and then update them with smaller data sets for specific tasks to provide better answers. Teams at Capital One use this technique, known as fine-tuning, to create new experiences and products such as a Slack bot.

Example: “What should I cook after work?”

This question could provide a variety of responses, but it may not match the question’s intent. By providing data on potential answers, you can narrow the focus to find the most likely response. The additional questions that might naturally flow from the first question help train models for better results.

Related question sets: “What are some one-pot meals I can cook?” or “What are some easy recipes to make?”

While these are similar, the subtle differences help refine the model during pre-training and improve future responses.

4. Translation

Transformers are especially valuable in translations, where models are trained to translate English into other languages.

For example, in the sentence, “I enjoy reading books in my free time,” the word enjoy has a nuanced meaning. A language model might struggle with the translation for enjoy, choosing "I like" (gusto) or "I love" (amo), which have very different meanings in Spanish.

A transformer model is more likely to produce a correct output that shows the accurate meaning: "Disfruto leyendo libros en mi tiempo libre."

Beyond the future of transformers

Transformers are changing what’s possible in the NLP domain. While they are used for regular NLP tasks now, technologists create new use cases every day. Researchers and technologists are already exploring ways to reduce memory requirements, create industry-specific models and use advanced pre-training techniques.

Research will shape the future of transformers and researchers are making breakthroughs at an increasing rate. Bayan Buss, VP, Machine Learning Engineering at Capital One, is making significant contributions to the ML domain through his applied research programs. In his research, he has researched transformer models and explored ways to apply deep learning techniques like transfer learning to tabular data. You can learn more about his work in the recently published article Deep learning for transfer tabular models.