Dublin Business School

Natural Language Processing

Transformers – HuggingFace Models

2023-24

terri.hoare@dbs.ie

Transformers – Hugging Face

Introduction

Several breakthroughs in deep learning have given birth to powerful language models such as GPT-3 and GPT-4. Applications include text generation, text classification, summarization, translation, question answering, chatbots, natural language understanding (NLU), and more.

Three key underpinning technology innovations underpin GPT's: transformers, transfer learning, and model hubs such as Hugging Face.

.

Transformers – Hugging Face

Transformers

The transformer is a neural network architecture proposed in 2017 in a groundbreaking paper called “Attention Is All You Need”, published by a team of Google researchers. In just a few years it swept across the field, crushing previous architectures that were typically based on recurrent neural networks (RNNs). The Transformer architecture is excellent at capturing patterns in long sequences of data and dealing with huge datasets—so much so that its use is now extending well beyond NLP, for example to image processing tasks.

.

Transformers – Hugging Face

Transformers

Google now uses BERT to enhance its search engine by better understanding users' search queries. Similarly, the GPT family of models from OpenAI have repeatedly made headlines in mainstream media for their ability to generate human-like text and images.

Like many great scientific breakthroughs, it was the synthesis of several ideas, like **attention**, **transfer learning**, and the **scaling up neural networks**, that were challenges in the research community at the time.

.

Transformers – Hugging Face

Transfer Learning

For most projects, it’s possible to download a model that has been pretrained on a generic dataset. The model can then be fine-tuned on a smaller dataset.

Pretraining has been mainstream in image processing since the early 2010s, but in NLP it was restricted to word embeddings (i.e., dense vector representations of individual words). For example, the word “bear” had the same pretrained embedding in “teddy bear” and in “to bear.” In 2018, several papers proposed full-blown language models that could be pretrained and fine-tuned for a variety of NLP tasks.

Transformers – Hugging Face

Model Hubs

Hugging Face is an open-source model hub supporting both PyTorch and TensorFlow models. It makes it easy to download a state-of-the-art pretrained model from the Hugging Face Hub, configure it for your task, fine-tune it on your dataset, and evaluate it.

Use of the library is growing quickly: in Q4 2021 it was used by over five thousand organizations and was installed using pip over four million times per month. The library ecosystem has expanded beyond NLP to include image processing; time series forecasting; reinforcement learning and graph learning. As of Q1 2024 it included more than 60 thousand models.

Tunstall, Lewis; Werra, Leandro von; Wolf, Thomas (2022). Natural Language Processing with Transformers. O'Reilly.

Transformers – Hugging Face

Model Hubs

Hugging Face is an open-source model hub supporting both PyTorch and TensorFlow models. It makes it easy to download a state-of-the-art pretrained model from the Hugging Face Hub, configure it for your task, fine-tune it on your dataset, and evaluate it.

Use of the library is growing quickly: in Q4 2021 it was used by over five thousand organizations and was installed using pip over four million times per month. The library ecosystem has expanded beyond NLP to include image processing; time series forecasting; reinforcement learning and graph learning. As of Q1 2024 it included more than 60 thousand models.

Tunstall, Lewis; Werra, Leandro von; Wolf, Thomas (2022). Natural Language Processing with Transformers. O'Reilly.

Exploring Transformers

In 2017, researchers at Google published a paper that proposed a novel neural network architecture for sequence modelling, the Transformer which outperformed recurrent neural networks (RNNs) on machine translation tasks, both in terms of translation quality and training cost.

In parallel, an effective transfer learning method called ULMFiT showed that training long short-term memory (LSTM) networks on a very large and diverse corpus could produce state-of-the-art text classifiers with little labelled data.

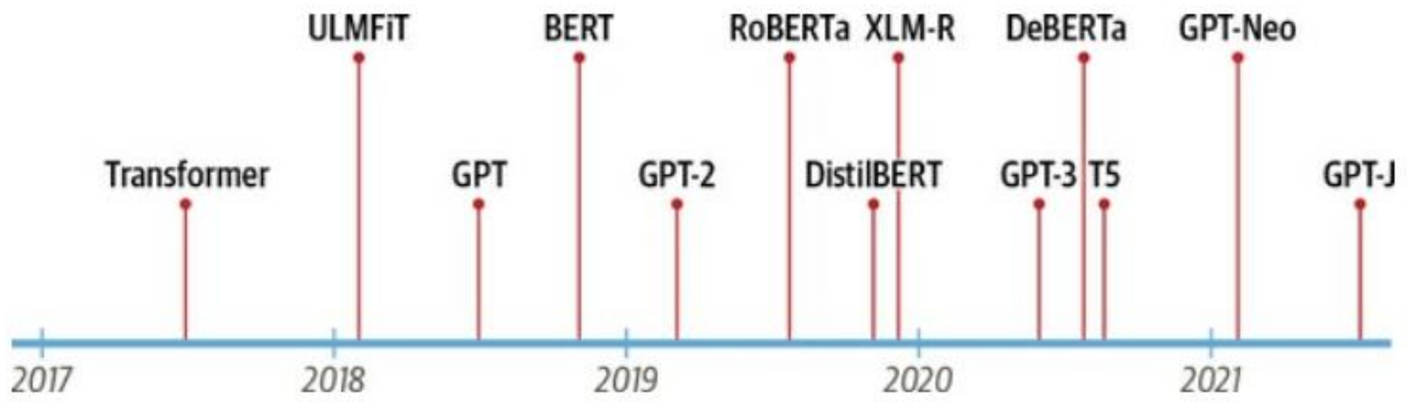
Exploring Transformers

These advances were the catalysts for two of today’s most well-known transformers: the Generative Pretrained Transformer (GPT) and Bidirectional Encoder Representations from Transformers (BERT).

By combining the Transformer architecture with unsupervised learning, these models removed the need to train task-specific architectures from scratch and broke almost every benchmark in NLP by a significant margin. Many models resulted from these innovations shown next slide.

.

Exploring Transformers



Transformers Timeline (Tunstall et al, 2022, p. 23

Note: GPT-4 released in ‘03 2023 containing more than a trillion parameters.

Tunstall, Lewis; Werra, Leandro von; Wolf, Thomas (2022). Natural Language Processing with Transformers. O'Reilly.

Exploring Transformers

Key concepts include the encoder-decoder framework, Attention mechanisms and Transfer learning

.

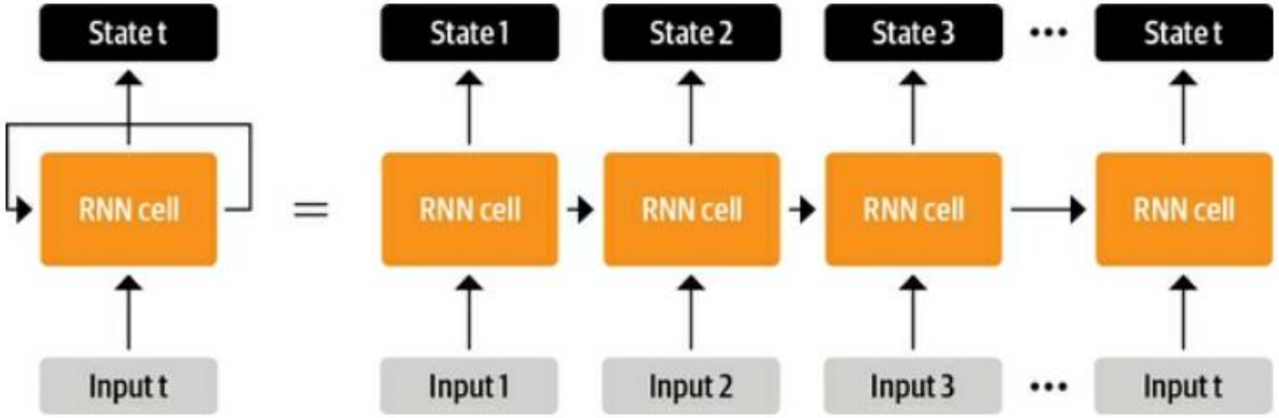
Exploring Transformers

The Encoder – Decoder Framework – LSTM’s

Prior to transformers, recurrent architectures such as LSTMs were the state of the art in NLP. These architectures contain a feedback loop in the network connections that allows information to propagate from one step to another, making them ideal for modelling sequential data like text. See Figure next slide.

Exploring Transformers

The Encoder – Decoder Framework - RNN



Unrolling an RNN in time
(Tunstall et al, 2022, p. 25

Tunstall, Lewis; Werra, Leandro von; Wolf, Thomas (2022). Natural Language Processing with Transformers. O'Reilly.

Exploring Transformers

The Encoder – Decoder Framework

The RNN receives some input (which could be a word or character), feeds it through the network, and outputs a vector called the hidden state.

At the same time, the model feeds some information back to itself through the feedback loop, which it can then use in the next step. This can be more clearly seen if we "unroll" the loop. The RNN passes information about its state at each step to the next operation in the sequence. This allows an RNN to keep track of information from previous steps and use it for its output predictions.

Exploring Transformers

The Encoder – Decoder Framework

These architectures were (and continue to be) widely used for NLP tasks, speech processing, and time series. (Refer Andrej Karpathy's blog post, "The Unreasonable Effectiveness of Recurrent Neural Networks".

The Unreasonable Effectiveness of Recurrent Neural Networks (karpathy.github.io)

The Encoder – Decoder Framework

One area where RNNs played an important role was in the development of machine translation systems, where the objective is to map a sequence of words in one language to another.

This kind of task is usually tackled with an **encoder-decoder** or sequence-to-sequence architecture which is well suited for situations where the input and output are both sequences of arbitrary length.

Exploring Transformers

The Encoder – Decoder Framework

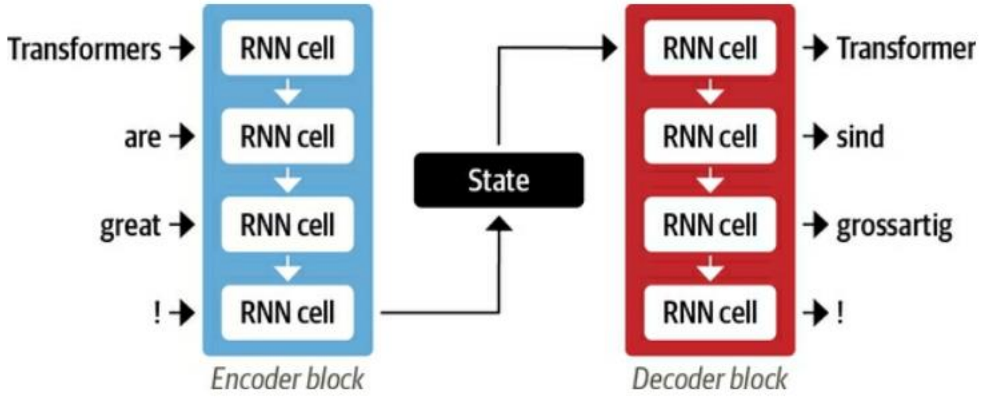
The **encoder** encodes the information from the input sequence into a numerical representation that is often called the last hidden state. This state is then passed to the **decoder** which generates the output sequence.

In general, the encoder and decoder components can be any kind of neural network architecture that can model sequences. Refer Figure next slide translating the English sentence “Transformers are great!” into the German “Transformer sind grossartig!”

Exploring Transformers

The Encoder – Decoder Framework

The input words are fed sequentially through the encoder and the output words are generated one at a time, from top to bottom.



An encoder-decoder architecture with a pair of RNNs (Tunstall et al, 2022, p. 26)

Tunstall, Lewis; Werra, Leandro von; Wolf, Thomas (2022). Natural Language Processing with Transformers. O'Reilly.

Exploring Transformers

The Encoder – Decoder Framework

Although elegant in its simplicity, one weakness of this architecture is that the final hidden state of the encoder creates an information bottleneck: it has to represent the meaning of the whole input sequence because this is all the decoder has access to when generating the output. This is especially challenging for long sequences, where information at the start of the sequence might be lost in the process of compressing everything to a single, fixed representation. This is solved by allowing the decoder to have access to all the encoder's hidden states. The general mechanism for this is called **Attention**, an important feature of Transformers.

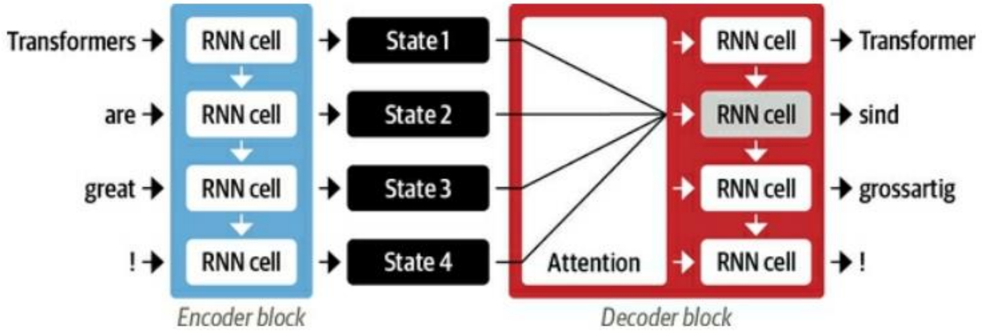
Exploring Transformers

Attention Mechanisms

The main idea behind attention is that instead of producing a single hidden state for the input sequence, the encoder outputs a hidden state at each step that the decoder can access. However, using all the states at the same time would create a huge input for the decoder, so some mechanism is needed to prioritize which states to use. This is where attention comes in: it lets the decoder assign a different amount of weight, or “attention,” to each of the encoder states at every decoding timestep. See next slide.

Exploring Transformers

Attention Mechanisms



An encoder-decoder architecture with an attention mechanism for a pair of RNNs (Tunstall et al, 2022, p. 27)

Exploring Transformers

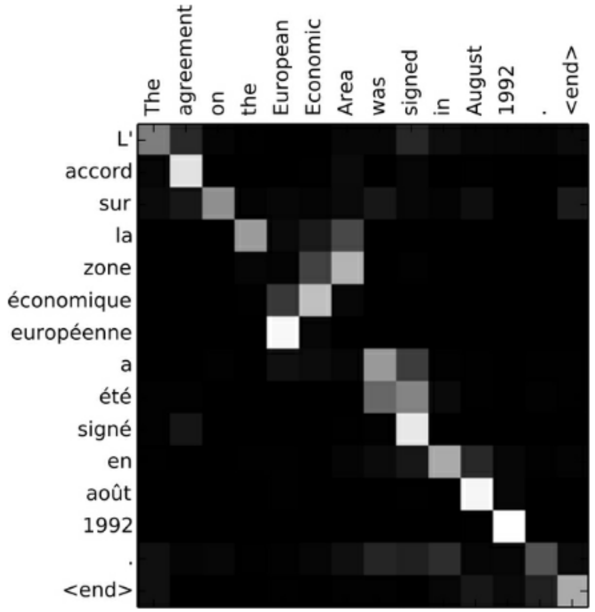
Attention Mechanisms

By focusing on which input tokens are most relevant at each timestep, these attention-based models can learn nontrivial alignments between the words in a generated translation and those in a source sentence. See next slide.

.

Exploring Transformers

Attention Mechanisms



The figure shows how the decoder can correctly align the words “zone” and “Area”, which are ordered differently in the two languages.

RNN encoder-decoder alignment of words in English and the generated translation in French (Tunstall et al, 2022, p. 28)

Exploring Transformers

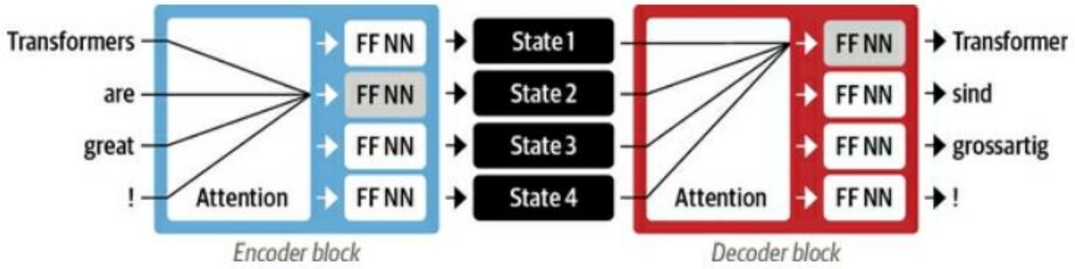
Attention Mechanisms

Although attention enabled the production of much better translations, there was still a major shortcoming with using recurrent models for the encoder and decoder: the computations are inherently sequential and cannot be parallelized across the input sequence.

Using the transformer, a new modelling paradigm was introduced which dispensed with recurrence altogether and relied instead entirely on a special form of attention called self-attention. The main idea is to allow attention to operate on all the states in the same layer of the neural network as shown in the figure (next slide).

Exploring Transformers

Attention Mechanisms



Encoder-decoder architecture of the original Transformer (Tunstall et al, 2022, p. 29)

Both the encoder and the decoder have their own self-attention mechanisms, whose outputs are fed to feed-forward neural networks (FF NNs). This architecture can be trained much faster than recurrent models and paved the way for many of the recent breakthroughs in NLP.

.

Exploring Transformers

Attention Mechanisms

In the original Transformer paper (Vaswani et al, 2017), the translation model was trained from scratch on a large corpus of sentence pairs in various languages. However, in many practical applications of NLP we do not have access to large amounts of labelled text data to train models on.

Transfer Learning was needed to power the transformer revolution.

.

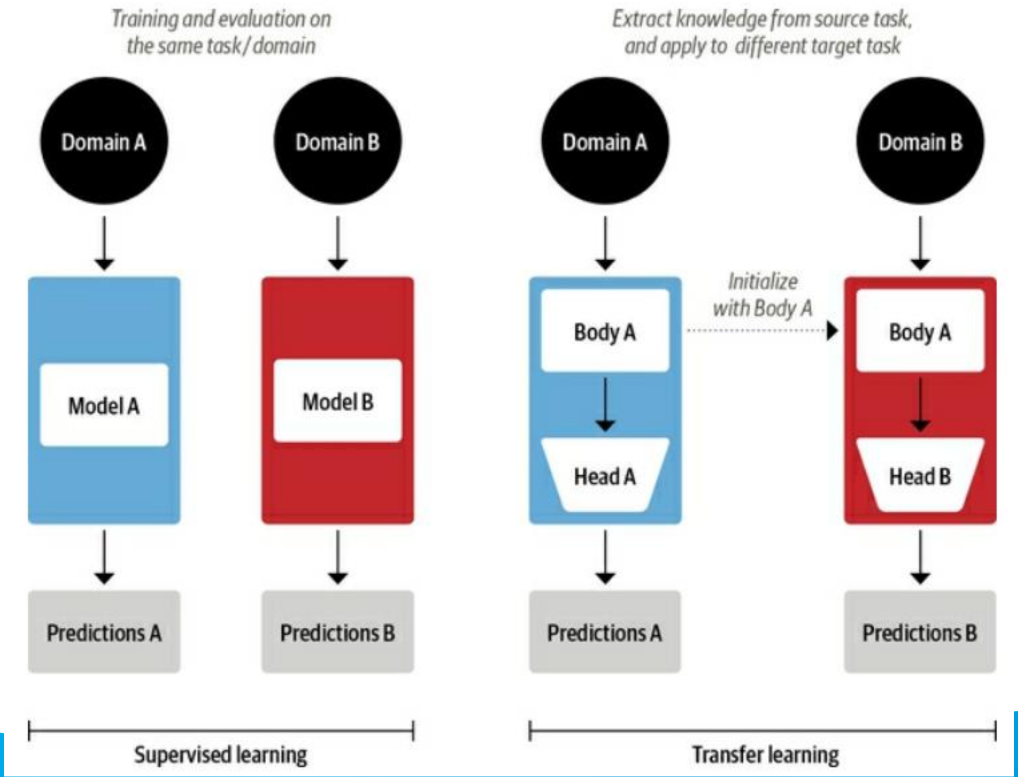
Exploring Transformers

Transfer Learning

It is common practice in computer vision to use transfer learning to train a convolutional neural network like ResNet on one task, and then adapt it to or fine-tune it on a new task. This allows the network to make use of the knowledge learned from the original task. Compared to traditional supervised learning, this approach typically produces high-quality models that can be trained much more efficiently on a variety of downstream tasks with much less labelled data. A comparison of the two approaches is shown in the figure (next slide).

Exploring Transformers

Transfer Learning



Comparison of traditional supervised learning (left) and transfer learning (right) (Tunstall et al, 2022, p. 30)

Tunstall, Lewis; Werra, Leandro von; Wolf, Thomas (2022). Natural Language Processing with Transformers. O'Reilly.

Exploring Transformers

Transfer Learning

In computer vision, the models are first trained on large-scale datasets such as ImageNet, which contain millions of images. This process is called pretraining and its main purpose is to teach the models the basic features of images, such as edges or colours.

These pretrained models can then be fine-tuned on a downstream task such as classifying flower species with a relatively small number of labelled examples (usually a few hundred per class). Fine-tuned models typically achieve a higher accuracy than supervised models trained from scratch on the same amount of labelled data.

Exploring Transformers

Transfer Learning

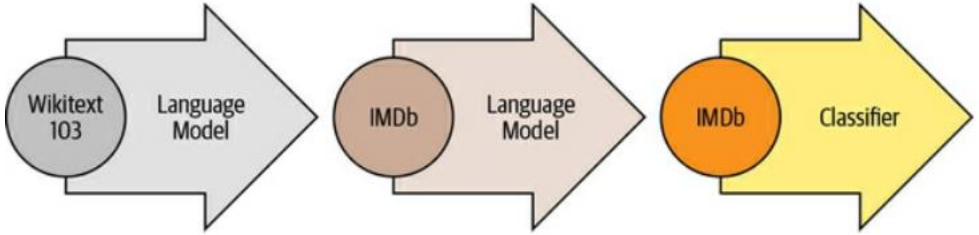
Although transfer learning became the standard approach in computer vision, for many years it was not clear what the analogous pretraining process was for NLP. As a result, NLP applications typically required large amounts of labelled data to achieve high performance which even then did not compare to what was achieved in the vision domain.

In 2017 and 2018, several research groups proposed new approaches that finally made transfer learning work for NLP. It started with an insight from researchers at OpenAI who obtained strong performance on a sentiment classification task by using features extracted from unsupervised pretraining. This was followed by ULMFiT which introduced a general framework to adapt pretrained LSTM models for various tasks.

,

Exploring Transformers

Transfer Learning



The ULMFiT process (Tunstall et al, 2022, p. 32)

The process involves 3 steps. Pretraining; Domain adaptation; and Fine-tuning.

Exploring Transformers

Transfer Learning

Pretraining Predict the next word based on the previous words. This task is referred to as language modelling. No labelled data is required and sources such as Wikipedia can be used.

Domain adaptation Once the language model is pretrained on a large-scale corpus, the next step is to adapt it to the in-domain corpus (e.g., IMDb corpus of movie reviews). This stage still uses language modelling, but now the model must predict the next word in the target corpus.

Fine-tuning The language model is fine-tuned with a classification layer for the target task (e.g., classifying the sentiment of movie reviews)

.

Exploring Transformers

Transfer Learning

Pretraining Predict the next word based on the previous words. This task is referred to as language modelling. No labelled data is required and sources such as Wikipedia can be used.

Domain adaptation Once the language model is pretrained on a large-scale corpus, the next step is to adapt it to the in-domain corpus (e.g., IMDb corpus of movie reviews). This stage still uses language modelling, but now the model must predict the next word in the target corpus.

Fine-tuning The language model is fine-tuned with a classification layer for the target task (e.g., classifying the sentiment of movie reviews)
.

Exploring Transformers

Transfer Learning

ULMFiT introduced a viable framework for pretraining and transfer learning in NLP.

In 2018, two transformers, GPT and BERT were released that combined self-attention with transfer learning.

GPT and BERT set a new state of the art across a variety of NLP benchmarks and ushered in the age of transformers.

Exploring Transformers

GPT

Uses only the decoder part of the Transformer architecture, and the same language modelling approach as ULMFiT.

GPT was pretrained on the BookCorpus which consists of 7,000 unpublished books from a variety of genres including Adventure, Fantasy, and Romance.

Exploring Transformers

BERT

Uses the encoder part of the Transformer architecture, and a special form of language modelling called masked language modelling.

The objective of masked language modelling is to predict randomly masked words in a text. For example, given a sentence like “I looked at my [MASK] and saw that [MASK] was late.” the model needs to predict the most likely candidates for the masked words that are denoted by [MASK].

BERT was pretrained on the BookCorpus and English Wikipedia.

Exploring Transformers

Hugging Face Transformers

Hugging Face Transformers provides a standardized interface to a wide range of transformer models as well as code and tools to adapt these models to new use cases.

The library currently supports three major deep learning frameworks (PyTorch, TensorFlow, and JAX) and allows you to easily switch between them.

In addition, it provides task-specific heads so you can easily fine-tune transformers on downstream tasks such as text classification, named entity recognition, and question answering. This significantly reduces the time to train and test models.

Exploring Transformers

Hugging Face Transformers

Hugging Face Transformers provides a standardized interface to a wide range of transformer models as well as code and tools to adapt these models to new use cases.

The library currently supports three major deep learning frameworks (PyTorch, TensorFlow, and JAX) and allows you to easily switch between them.

In addition, it provides task-specific heads so you can easily fine-tune transformers on downstream tasks such as text classification, named entity recognition, and question answering. This significantly reduces the time to train and test models.

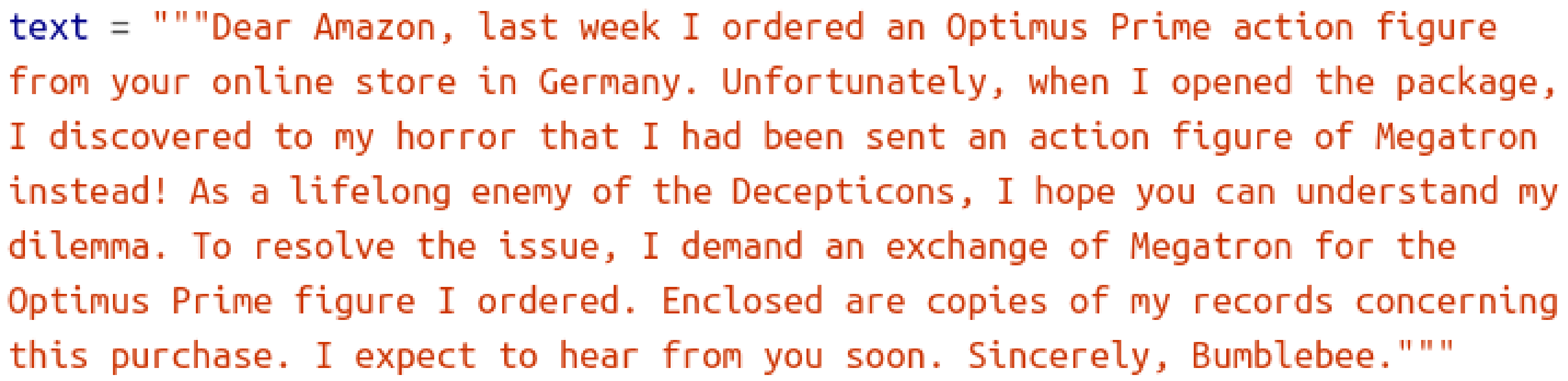
Exploring Transformers

Hugging Face NLP demonstration

```
text = """Dear Amazon, last week I ordered an Optimus Prime action figure
from your online store in Germany. Unfortunately, when I opened the package,
I discovered to my horror that I had been sent an action figure of Megatron
instead! As a lifelong enemy of the Decepticons, I hope you can understand my
dilemma. To resolve the issue, I demand an exchange of Megatron for the
Optimus Prime figure I ordered. Enclosed are copies of my records concerning
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
```

Exploring Transformers

Hugging Face NLP demonstration



```
text = """Dear Amazon, last week I ordered an Optimus Prime action figure
from your online store in Germany. Unfortunately, when I opened the package,
I discovered to my horror that I had been sent an action figure of Megatron
instead! As a lifelong enemy of the Decepticons, I hope you can understand my
dilemma. To resolve the issue, I demand an exchange of Megatron for the
Optimus Prime figure I ordered. Enclosed are copies of my records concerning
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
```

Tunstall, Lewis; Werra, Leandro von; Wolf, Thomas (2022). Natural Language Processing with Transformers. O'Reilly.

Exploring Transformers

HF NLP demonstration – Sentiment Classification

Hugging Face Transformers has a layered API that allows you to interact with the library at various levels of abstraction.

Pipelines abstract the steps needed to convert raw text into a set of predictions from a fine-tuned model. The first time you run the code a few progress bars appear because the pipeline automatically downloads the model weights from the Hugging Face Hub. The next time you instantiate the pipeline, the library will notice that you've already downloaded the weights and will use the cached version. By default, the **text-classification pipeline** uses a model that's designed for sentiment analysis, but it also supports multiclass classification.

Tunstall, Lewis; Werra, Leandro von; Wolf, Thomas (2022). Natural Language Processing with Transformers. O'Reilly.

Exploring Transformers

HF NLP demo – Sentiment Classification

Hugging Face Transformers has a layered API that allows you to interact with the library at various levels of abstraction.

Pipelines abstract the steps needed to convert raw text into a set of predictions from a fine-tuned model. The first time you run the code a few progress bars appear because the pipeline automatically downloads the model weights from the Hugging Face Hub. The next time you instantiate the pipeline, the library will notice that you've already downloaded the weights and will use the cached version. By default, the **text-classification pipeline** uses a model that's designed for sentiment analysis, but it also supports multiclass classification.

Exploring Transformers

HF NLP demo – Named Entity Recognition

Predicting the sentiment of customer feedback is a good first step, but you often want to know if the feedback was about a particular item or service. In NLP, real-world objects like products, places, and people are called named entities, and extracting them from text is called **named entity recognition (NER).**

We apply NER by loading the corresponding **NER pipeline** and feeding the customer review to it.

Exploring Transformers

HF NLP demo – Named Entity Recognition

The demo pipeline detects all the entities and assigns a category such as ORG (organization), LOC (location), or PER (person) to each of them.

The demo uses the aggregation_strategy argument to group the words according to the model’s predictions. For example, the entity “Optimus Prime” is composed of two words but is assigned a single category: MISC (miscellaneous).

The scores tell us how confident the model was about the entities it identified. We can see that it was least confident about “Decepticons.”

Exploring Transformers

HF NLP demo – Question Answering

In question answering, we provide the model with a passage of text called the context, along with a question whose answer we'd like to extract. The model then returns the text corresponding to the answer.

Exploring Transformers

HF NLP demo – Summarization

The goal of text summarization is to take a long text as input and generate a short version with all the relevant facts. This is a much more complicated task than the previous ones since it requires the model to generate coherent text.

We instantiate a **summarization pipeline** in the demo to summarize as below.

```
 Bumblebee ordered an Optimus Prime action figure from your online store in
Germany. Unfortunately, when I opened the package, I discovered to my horror
that I had been sent an action figure of Megatron instead.
```

Exploring Transformers

HF NLP demo – Translation

Like summarization, translation is a task where the output consists of generated text. Let’s use a translation pipeline to translate an English text to German:

We use a **translation pipeline** in the demo to translate the text into German that correctly uses German formal pronouns like “Ihrem” and “Sie.”

Exploring Transformers

HF NLP demo – Text Generation

Let’s say you would like to be able to provide faster replies to customer feedback by having access to an autocomplete function.

The **text generation pipeline can be used.**

```
generator = pipeline("text-generation")
response = "Dear Bumblebee, I am sorry to hear that your order was mixed up."
```

Exploring Transformers

HF NLP demo – Conclusion

The models demonstrated are publicly available and already fine-tuned for the task at hand. In general, however, you need to fine-tune models on your own data.

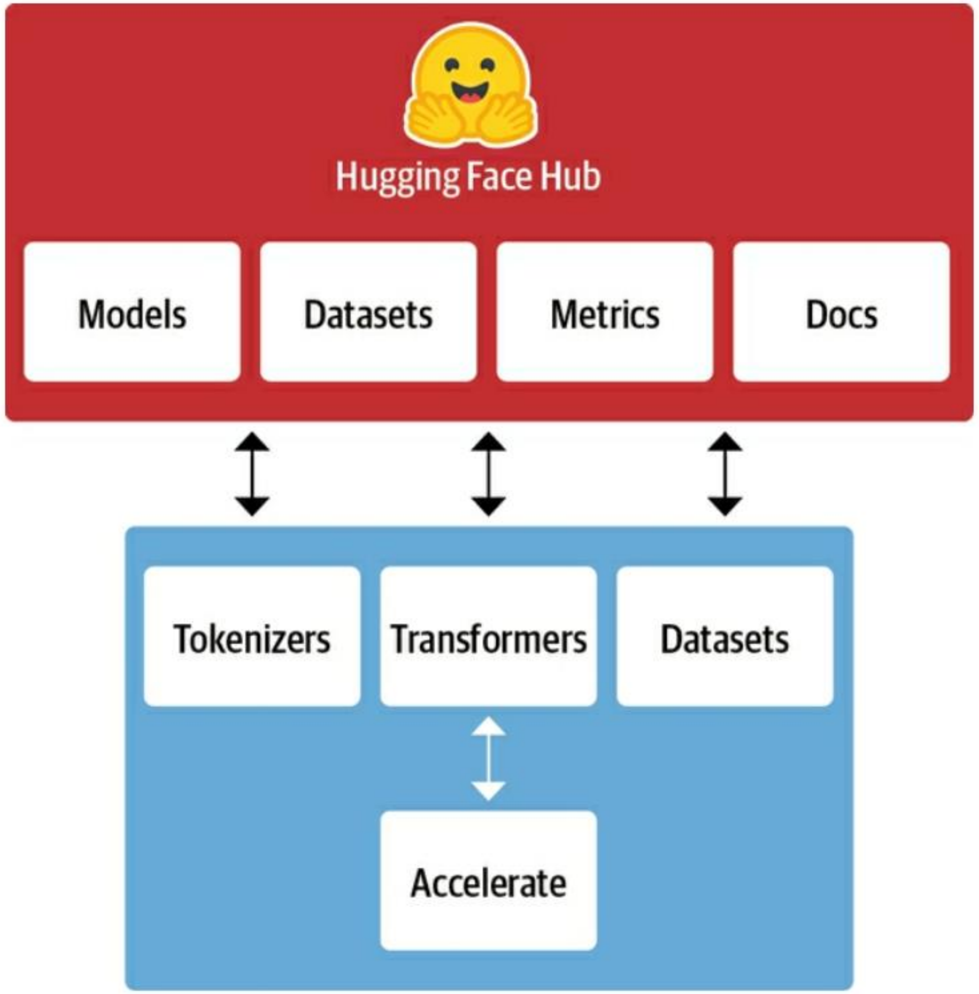
Exploring Transformers

The Hugging Face Ecosystem

The Hugging Face ecosystem consists mainly of two parts: a family of libraries and the Hub. The libraries provide the code while the Hub provides the pretrained model weights, datasets, scripts for the evaluation metrics, and more.

Exploring Transformers

Overview of the Hugging Face Ecosystem
(Tunstall et al, 2022, p. 44)



Tunstall, Lewis; Werra, Leandro von; Wolf, Thomas (2022). Natural Language Processing with Transformers. O'Reilly.

Exploring Transformers

The Hugging Face Hub

Transfer learning is one of the key factors driving the success of transformers because it makes it possible to reuse pretrained models for new tasks. Consequently, it is crucial to be able to load pretrained models quickly and run experiments with them.

The Hugging Face Hub hosts over 60,000 (2024) freely available models.

In addition to model weights, the Hub also hosts datasets and scripts for computing metrics, which let you reproduce published results or leverage additional data for your application.

Exploring Transformers

The Hugging Face Tokenizers

Hugging Face Tokenizers provides many tokenization strategies and is extremely fast at tokenizing text. It also takes care of all the pre- and postprocessing steps, such as normalizing the inputs and transforming the model outputs to the required format.

Exploring Transformers

The Hugging Face Datasets

Loading, processing, and storing datasets can be a cumbersome process, especially when the datasets get too large to fit in your laptop's RAM. In addition, you usually need to implement various scripts to download the data and transform it into a standard format.
Datasets simplifies this process by providing a standard interface for thousands of datasets that can be found on the Hub. It also provides smart caching (so you don't have to redo your preprocessing each time you run your code) and avoids RAM limitations by leveraging a special mechanism called memory mapping that stores the contents of a file in virtual memory and enables multiple processes to modify a file more efficiently. The library is also interoperable with popular data wrangling libraries like Pandas and NumPy.

Exploring Transformers

Main Challenges with Transformers

Despite their usefulness, transformers are far from being a silver bullet. Challenges include:

Language NLP research is dominated by the English language. There are several models for other languages, but it is harder to find pretrained models for rare languages.

Data availability Although transfer learning dramatically reduces the amount of labelled training data needed, the amount still exceeds how much a human needs to perform the task.

.

Main Challenges with Transformers

Working with long documents Self-attention works extremely well on paragraph-long texts, but it becomes very expensive when we move to longer texts like whole documents.

Opacity As with other deep learning models, transformers are “black box.” It is hard or impossible to unravel “why” a model made a certain prediction. This is an especially hard challenge when these models are deployed to make critical decisions.

Bias Transformer models are pretrained on text data from the internet with all the biases (racist, sexist etc.) that are present. This is a challenge.

Exploring Transformers

References

Devlin, J., Chang, M., Lee, K., Toutanova, K. (2018). Deep Bidirectional Transformers for Language Understanding. [1810.04805] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (arxiv.org)

Radford, A., Narasimhan, K., Salimans, T., Sutskever, R. (2018). Improving Language Understanding by Generative Pre-Training. language_understanding_paper.pdf (openai.com)

Tunstall, Lewis; Werra, Leandro von; Wolf, Thomas (2022). Natural Language Processing with Transformers. O'Reilly.

Exploring Transformers

References

Vaswani, A., Shazeer, N., Parmar, N., Uszkoeit, J., Jones, L., Gomez, A., Kaiser, L. (2017). Attention is All You Need. 31st Conference in Neural Information Processing Systems (NIPS 2017).