Introduction to Hugging Face

Juan David Martínez Vargas
jdmartinev@eafit.edu.co

2023

UNIVERSIDAD EAFIT®

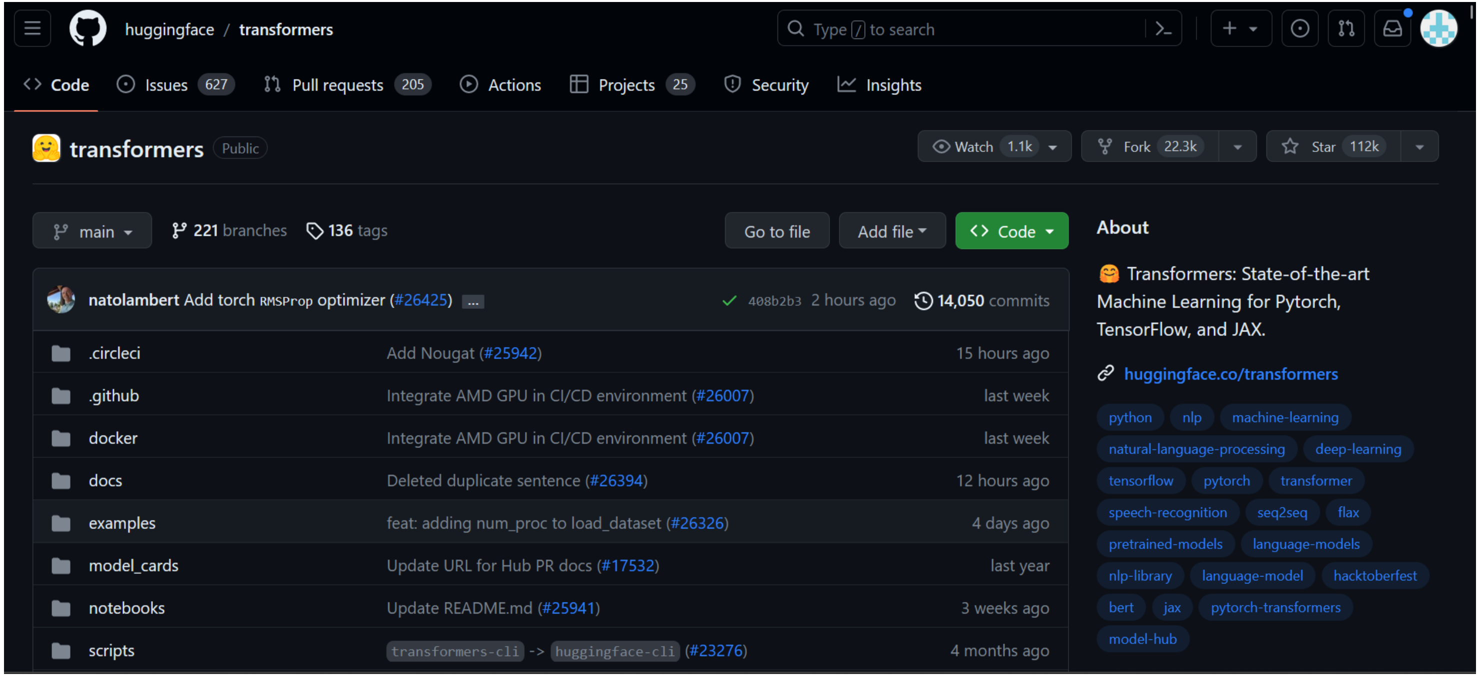
What is Hugging Face?

🤗

1. **NLP & Machine Learning Company**: Leading innovator in the field of natural language processing.
2. **Transformers Library**: Offers a user-friendly interface to state-of-the-art models like BERT, GPT-2, and T5.
3. **Model Hub**: A platform for sharing and discovering pre-trained NLP models.
4. **Datasets Library**: Centralized access to diverse NLP datasets.
5. **Community-Driven**: Active forums, collaborations, and open-source contributions.

Bridging the gap between research and real-world applications in NLP

What is Hugging Face?



natolambert Add torch `RMSProp` optimizer (#26425) ... ✓ 408b2b3 2 hours ago 14,050 commits

```
.circleci      Add Nougat (#25942)                                    15 hours ago
.github        Integrate AMD GPU in CI/CD environment (#26007)        last week
docker         Integrate AMD GPU in CI/CD environment (#26007)        last week
docs           Deleted duplicate sentence (#26394)                    12 hours ago
examples       feat: adding num_proc to load_dataset (#26326)         4 days ago
model_cards    Update URL for Hub PR docs (#17532)                    last year
notebooks      Update README.md (#25941)                              3 weeks ago
scripts        transformers-cli -> huggingface-cli (#23276)           4 months ago
```

About

🤗 Transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow, and JAX.

huggingface.co/transformers

What is Natural Language Processing

1. Definition: NLP stands for Natural Language Processing.

2. Bridge Between Humans & Computers: Enables machines to understand, interpret, and generate human language.

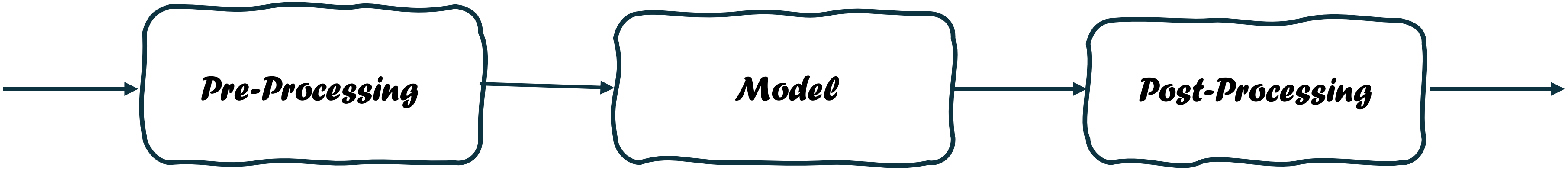
3. Applications:

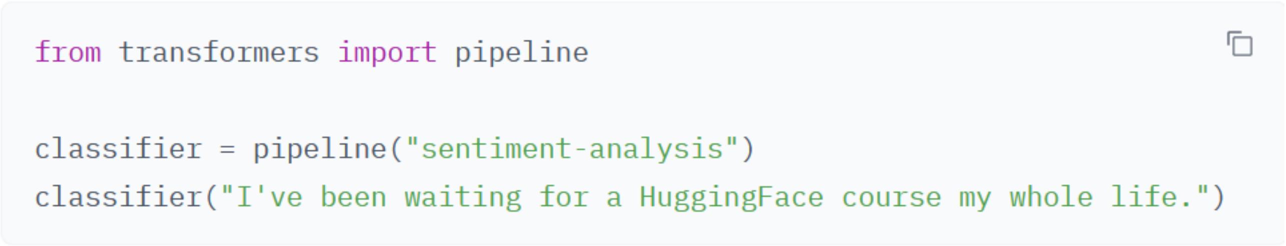
1. Text and speech recognition
2. Machine translation
3. Sentiment analysis
4. Chatbots and virtual assistants

4. Combines: Linguistics, computer science, and artificial intelligence to simulate human language abilities in machines.

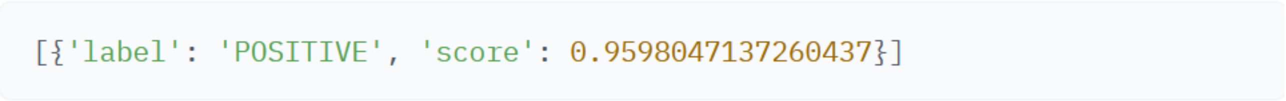
Making machines understand and respond to us more naturally

Pipeline





```
from transformers import pipeline

classifier = pipeline("sentiment-analysis")
classifier("I've been waiting for a HuggingFace course my whole life.")
```

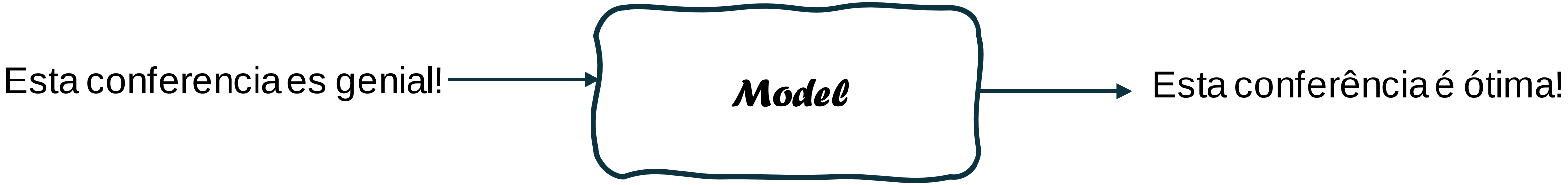


```
[{'label': 'POSITIVE', 'score': 0.9598047137260437}]
```

The pipeline function returns an end-to-end object that performs an NLP task on one or several texts

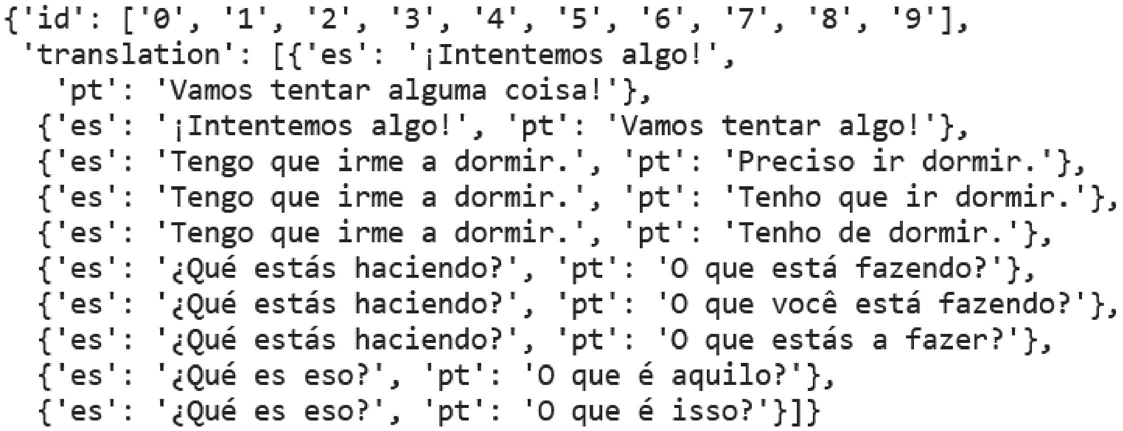
https://huggingface.co/learn/nlp-course/chapter1/3?fw=pt

seq2seq models



Esta conferencia es genial!

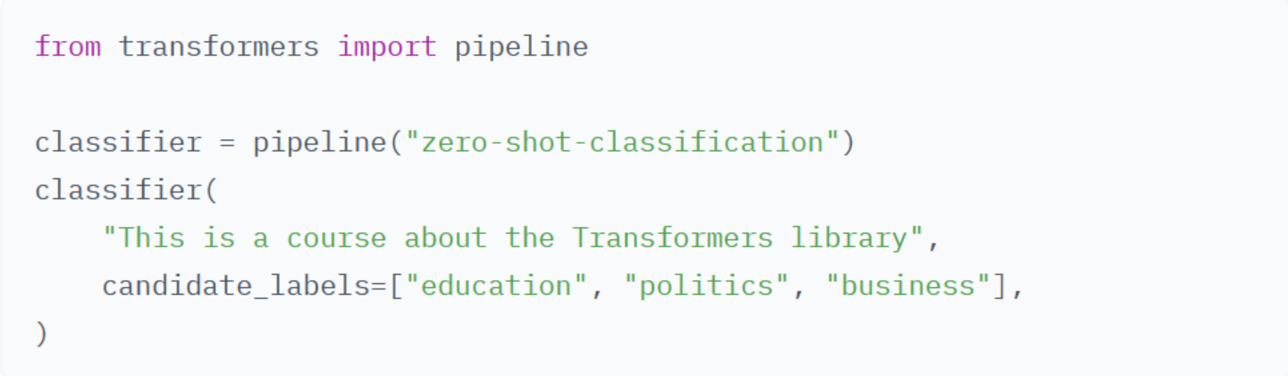
Esta conferência é ótima!



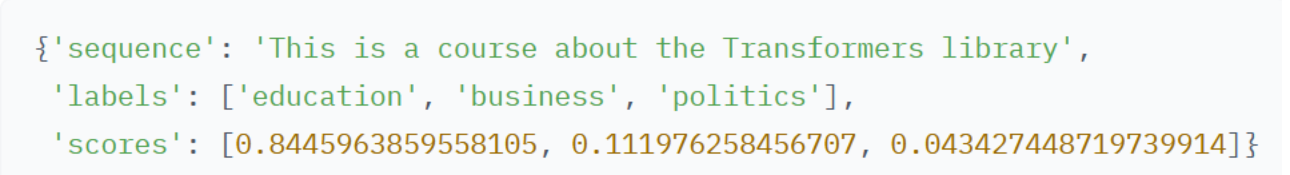
```
{'id': ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
 'translation': [{'es': '¡Intentemos algo!',
   'pt': 'Vamos tentar alguma coisa!'},
  {'es': '¡Intentemos algo!', 'pt': 'Vamos tentar algo!'},
  {'es': 'Tengo que irme a dormir.', 'pt': 'Preciso ir dormir.'},
  {'es': 'Tengo que irme a dormir.', 'pt': 'Tenho que ir dormir.'},
  {'es': 'Tengo que irme a dormir.', 'pt': 'Tenho de dormir.'},
  {'es': '¿Qué estás haciendo?', 'pt': 'O que está fazendo?'},
  {'es': '¿Qué estás haciendo?', 'pt': 'O que você está fazendo?'},
  {'es': '¿Qué estás haciendo?', 'pt': 'O que estás a fazer?'},
  {'es': '¿Qué es eso?', 'pt': 'O que é aquilo?'},
  {'es': '¿Qué es eso?', 'pt': 'O que é isso?'}]}
```

Available pipelines

- feature-extraction (get the vector representation of a text)
- fill-mask
- ner (named entity recognition)
- question-answering
- sentiment-analysis
- summarization
- text-generation
- translation
- zero-shot-classification



```python
from transformers import pipeline

classifier = pipeline("zero-shot-classification")
classifier(
    "This is a course about the Transformers library",
    candidate_labels=["education", "politics", "business"],
)
```



```
{'sequence': 'This is a course about the Transformers library',
 'labels': ['education', 'business', 'politics'],
 'scores': [0.8445963859558105, 0.111976258456707, 0.043427448719739914]}
```

Pre-processing - tokenizers

Word-based

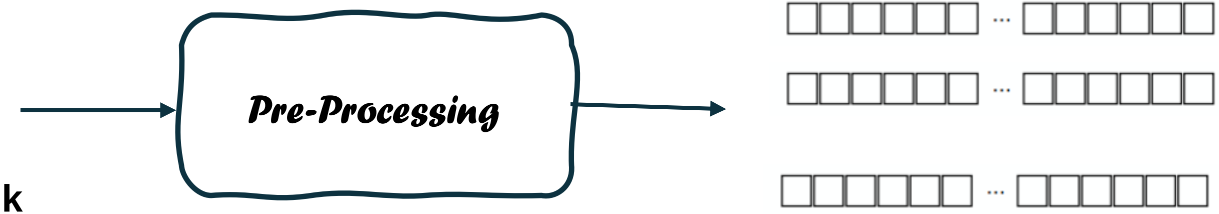
Character-based

Subword-based

Pre-processing - tokenizers

Raw text

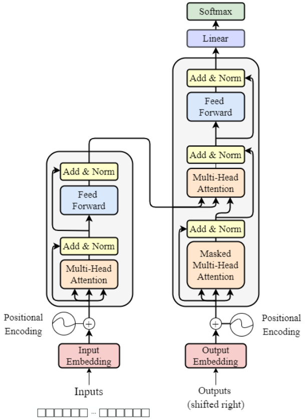
This is an amazing talk



Pre-Processing

Neural networks can’t process raw text

Convert text into numbers



1. Split text into words, subwords or symbols (punctuation) – tokens

2. Map each token to an integer

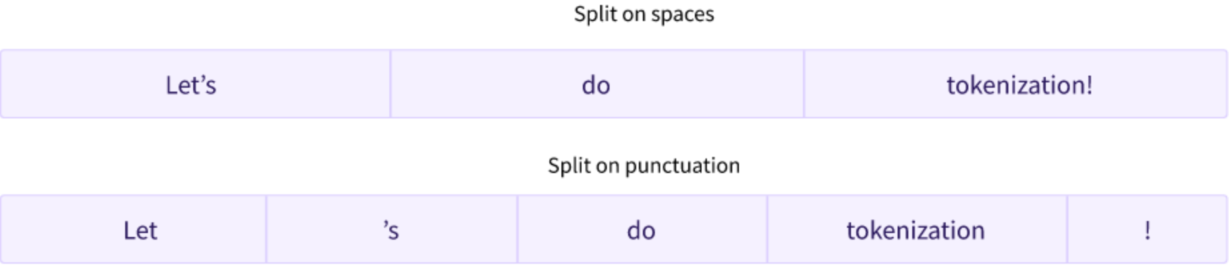
3. Add additional inputs

Pre-processing - tokenizers

Word-based

Each word has a token (integer) that will turn into a word embedding

A lot of embeddings!!!

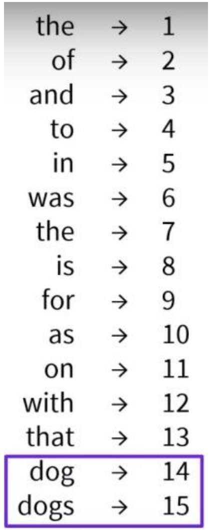


Split on spaces

Let’s	do	tokenization!

Split on punctuation

Let	’s	do	tokenization	!



the	→	1
of	→	2
and	→	3
to	→	4
in	→	5
was	→	6
the	→	7
is	→	8
for	→	9
as	→	10
on	→	11
with	→	12
that	→	13
dog	→	14
dogs	→	15

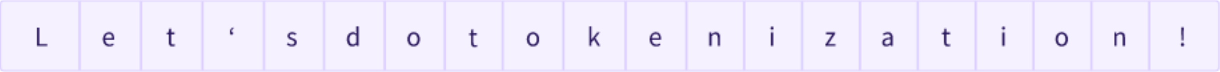
Pre-processing - tokenizers

Character-based

Each character has a token (integer) that will turn into a word embedding

Less embeddings 😋

Less meaningful 😔



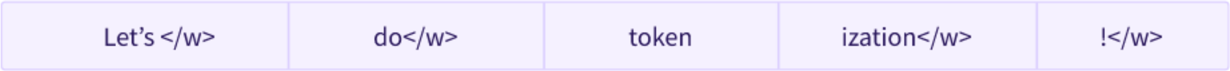
L	e	t	‘	s	d	o	t	o	k	e	n	i	z	a	t	i	o	n	!

Pre-processing - tokenizers

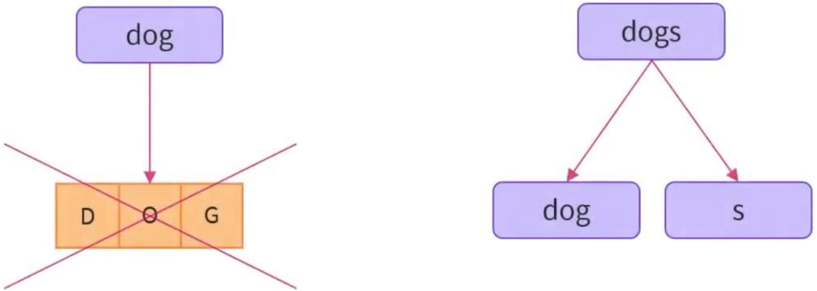
Subword-based

Frequently used words are should not be split into smaller subwords

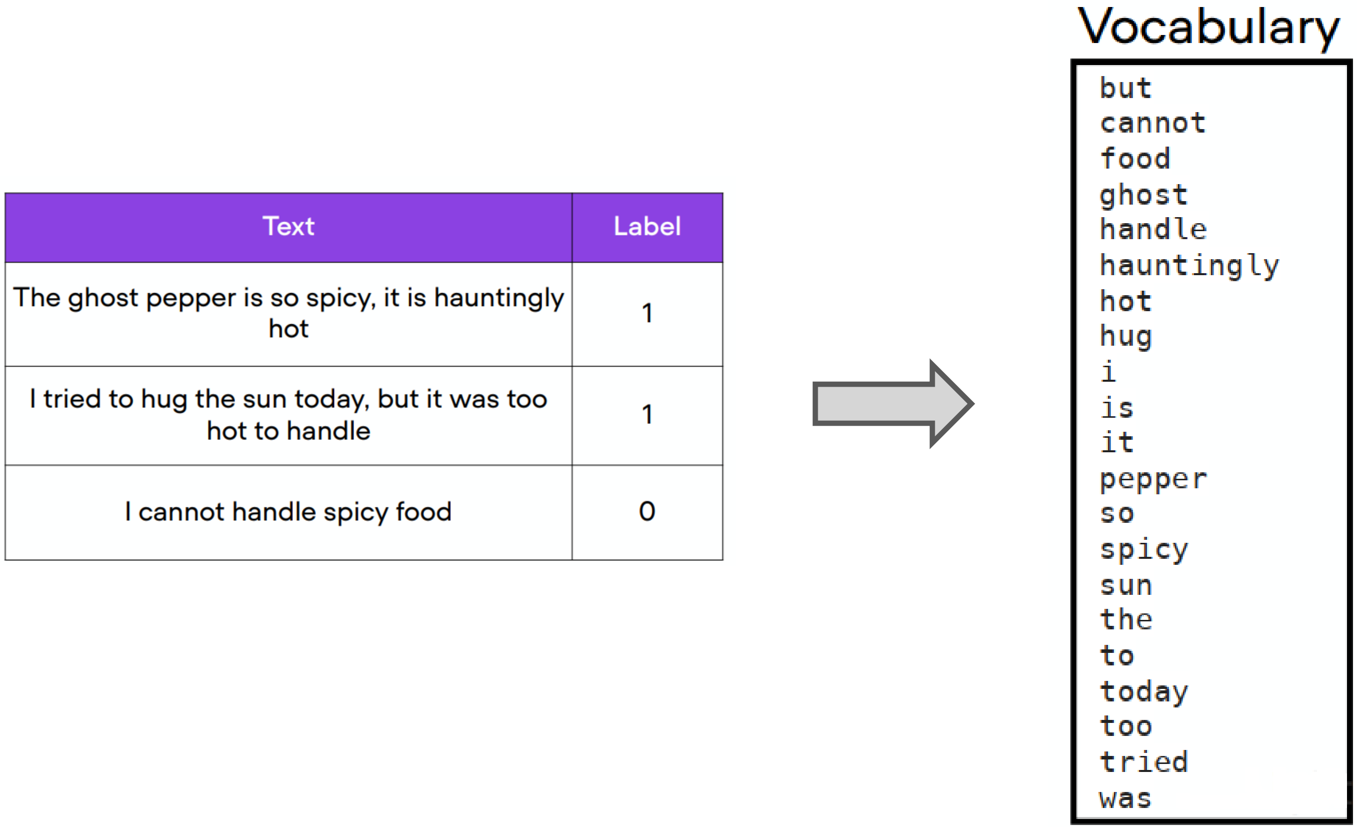
Rare words should be decomposed into meaningful subwords



Let’s </w>	do</w>	token	ization</w>	!</w>



Vocabulary

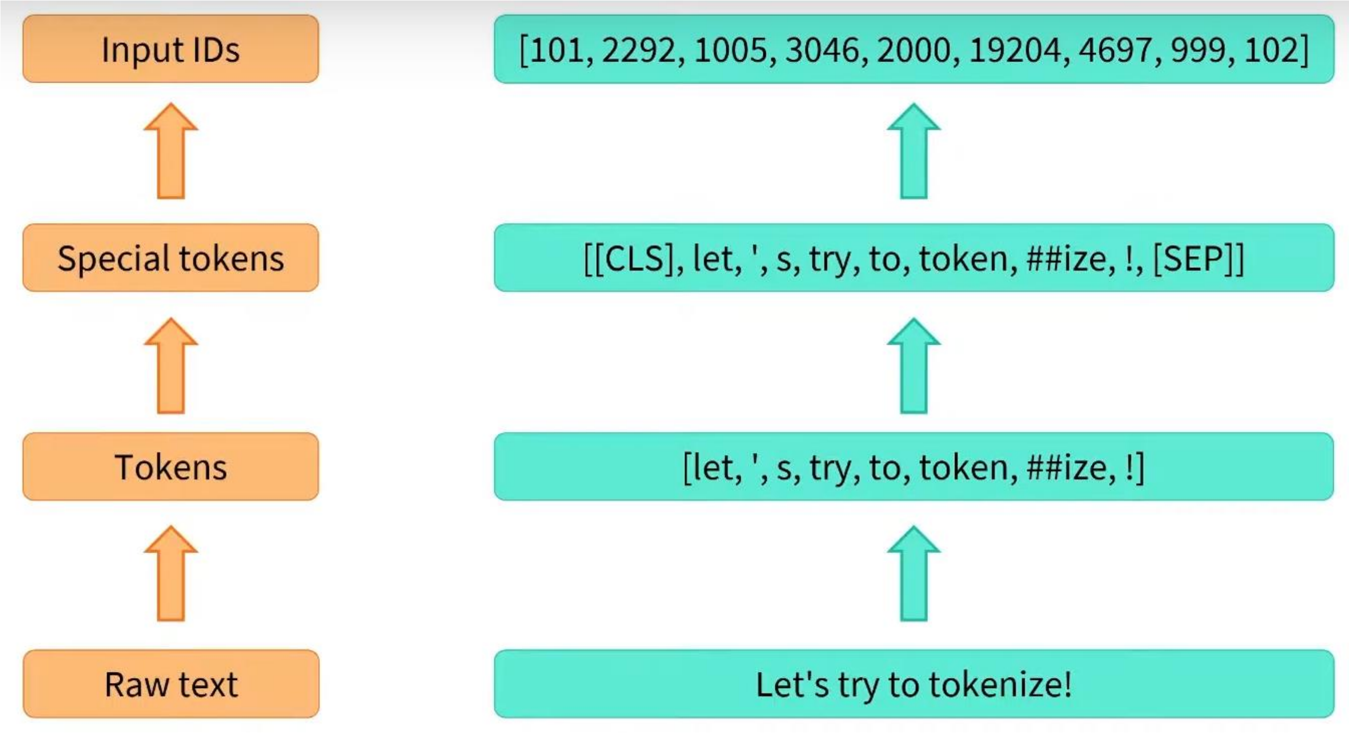


Text	Label
The ghost pepper is so spicy, it is hauntingly hot	1
I tried to hug the sun today, but it was too hot to handle	1
I cannot handle spicy food	0

Vocabulary

```
but
cannot
food
ghost
handle
hauntingly
hot
hug
i
is
it
pepper
so
spicy
sun
the
to
today
too
tried
was
```

Tokenization pipeline



Input IDs

[101, 2292, 1005, 3046, 2000, 19204, 4697, 999, 102]

Special tokens

[[CLS], let, ', s, try, to, token, ##ize, !, [SEP]]

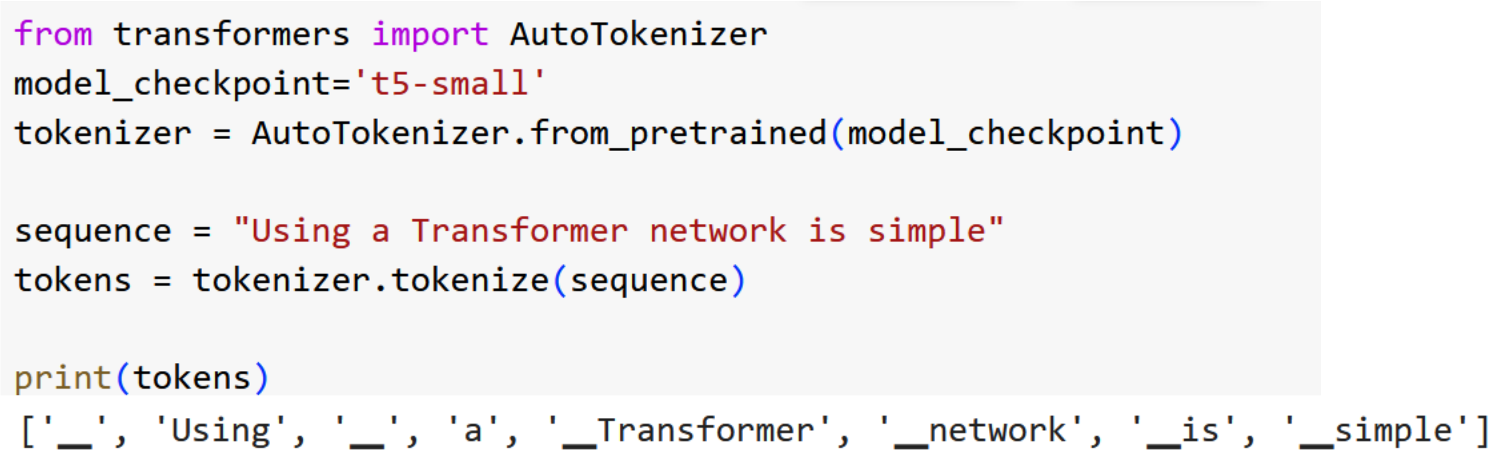
Tokens

[let, ', s, try, to, token, ##ize, !]

Raw text

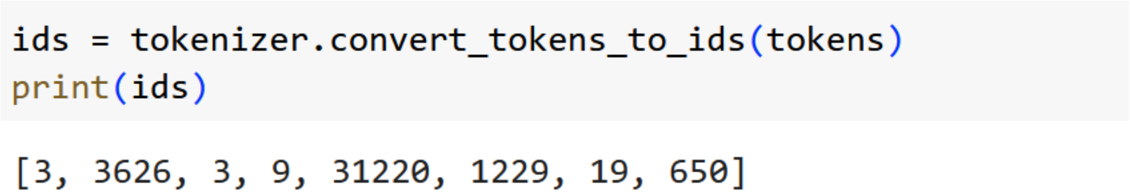
Let's try to tokenize!

Hugging Face Tokenizer



```
from transformers import AutoTokenizer
model_checkpoint='t5-small'
tokenizer = AutoTokenizer.from_pretrained(model_checkpoint)

sequence = "Using a Transformer network is simple"
tokens = tokenizer.tokenize(sequence)

print(tokens)
```

```
['▁', 'Using', '▁', 'a', '▁Transformer', '▁network', '▁is', '▁simple']
```

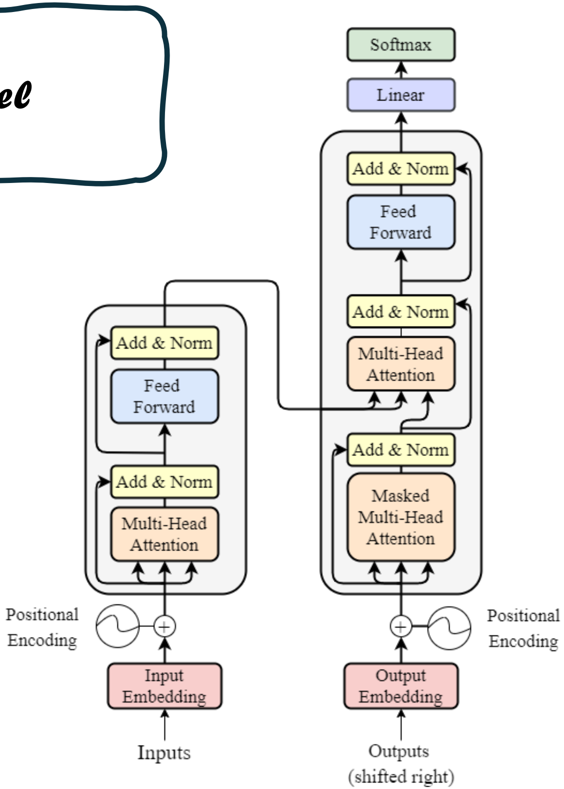


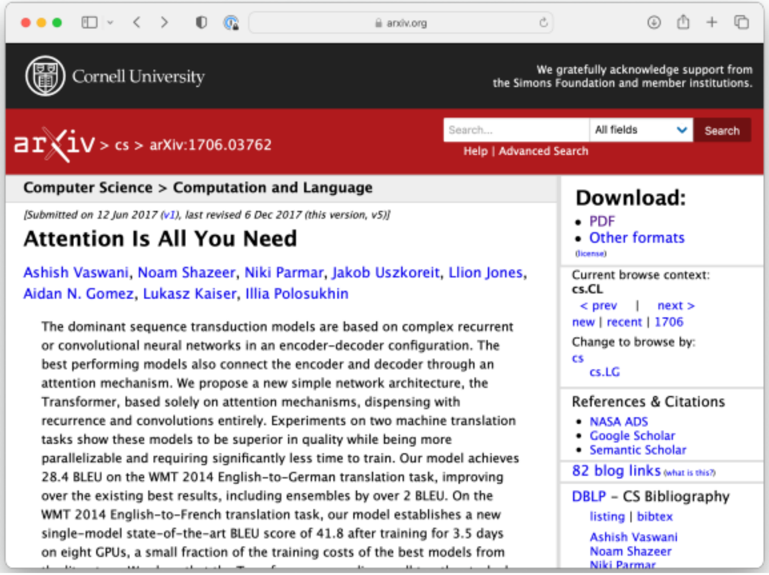
```
ids = tokenizer.convert_tokens_to_ids(tokens)
print(ids)
```

```
[3, 3626, 3, 9, 31220, 1229, 19, 650]
```

Model - Transformer

Model





We gratefully acknowledge support from the Simons Foundation and member institutions.

arXiv > cs > arXiv:1706.03762

Computer Science > Computation and Language

[Submitted on 12 Jun 2017 (v1), last revised 6 Dec 2017 (this version, v5)]

Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks in an encoder-decoder configuration. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from

Download:

- PDF
- Other formats

(license)

Current browse context:
cs.CL
< prev | next >
new | recent | 1706

Change to browse by:
cs
cs.LG

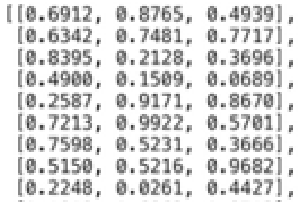
References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

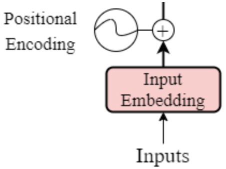
82 blog links (what is this?)

DBLP - CS Bibliography
listing | bibtex
Ashish Vaswani
Noam Shazeer
Niki Parmar

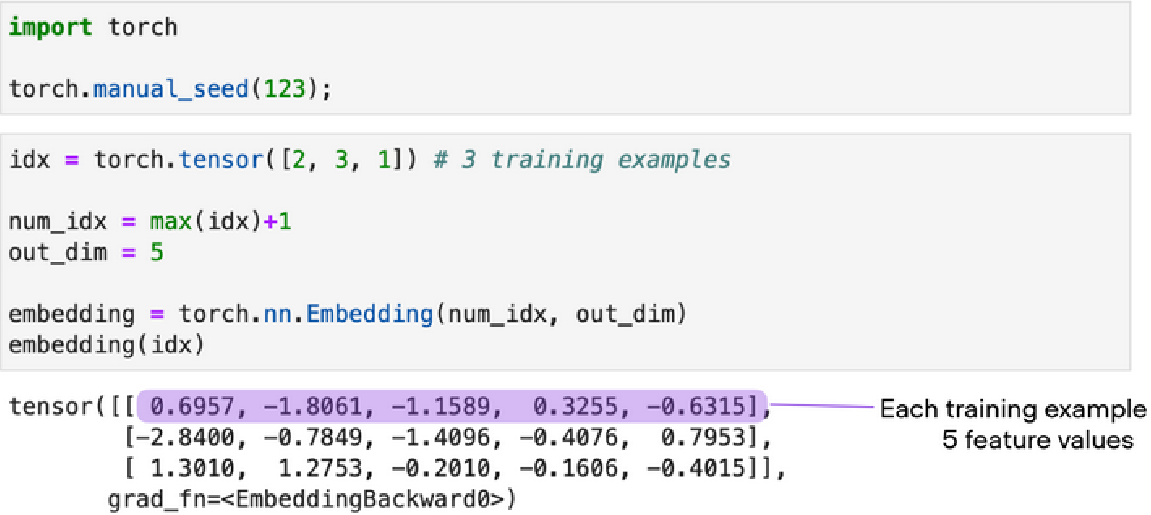
Embeddings



```
[[0.6912, 0.8765, 0.4939],
 [0.6342, 0.7481, 0.7717],
 [0.8395, 0.2128, 0.3696],
 [0.4900, 0.1509, 0.0689],
 [0.2587, 0.9171, 0.8670],
 [0.7213, 0.9922, 0.5701],
 [0.7598, 0.5231, 0.3666],
 [0.5150, 0.5216, 0.9682],
 [0.2248, 0.0261, 0.4427],
```



[101, 2292, 1005, 3046, 2000, 19204, 4697, 999, 102]



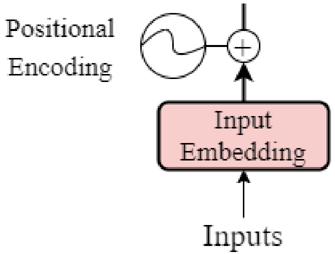
```
import torch

torch.manual_seed(123);
```

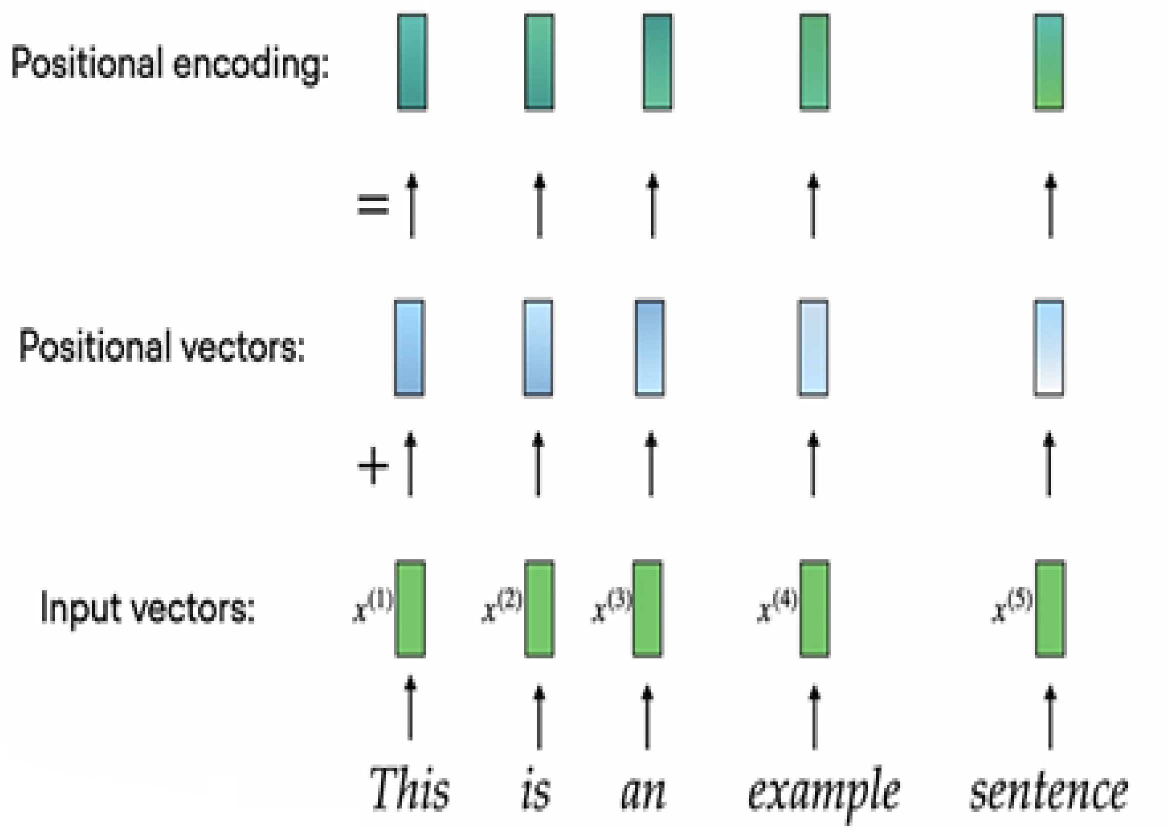
```
idx = torch.tensor([2, 3, 1]) # 3 training examples

num_idx = max(idx)+1
out_dim = 5

embedding = torch.nn.Embedding(num_idx, out_dim)
embedding(idx)
```

```
tensor([[ 0.6957, -1.8061, -1.1589,  0.3255, -0.6315],
        [-2.8400, -0.7849, -1.4096, -0.4076,  0.7953],
        [ 1.3010,  1.2753, -0.2010, -0.1606, -0.4015]],
       grad_fn=<EmbeddingBackward0>)
```

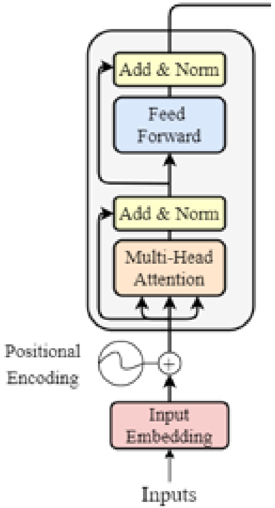
Each training example has 5 feature values

Positional encoding

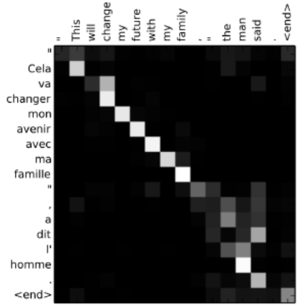




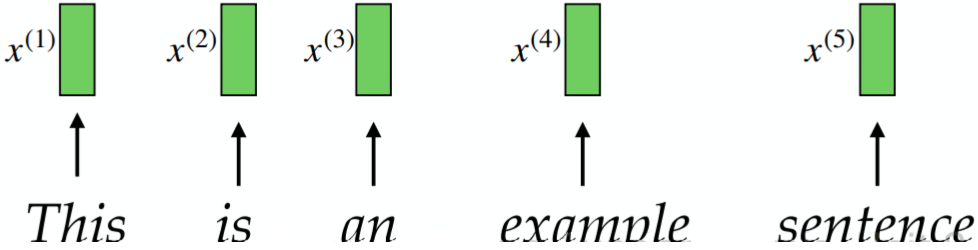
Encoder – Attention - Training



Idea: Create a context vector that encodes the meaning of the Word within the text



$$z^{(i)} = \sum_{j=1}^{T} \alpha_{ij} \cdot x^{(j)}$$

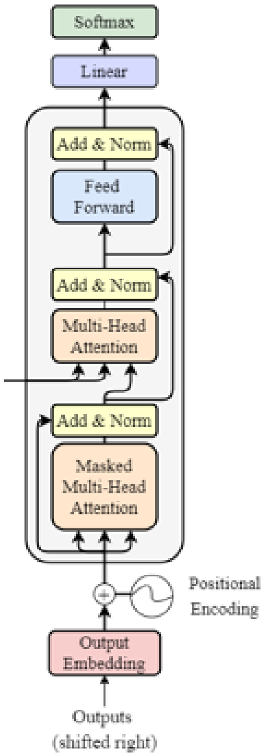


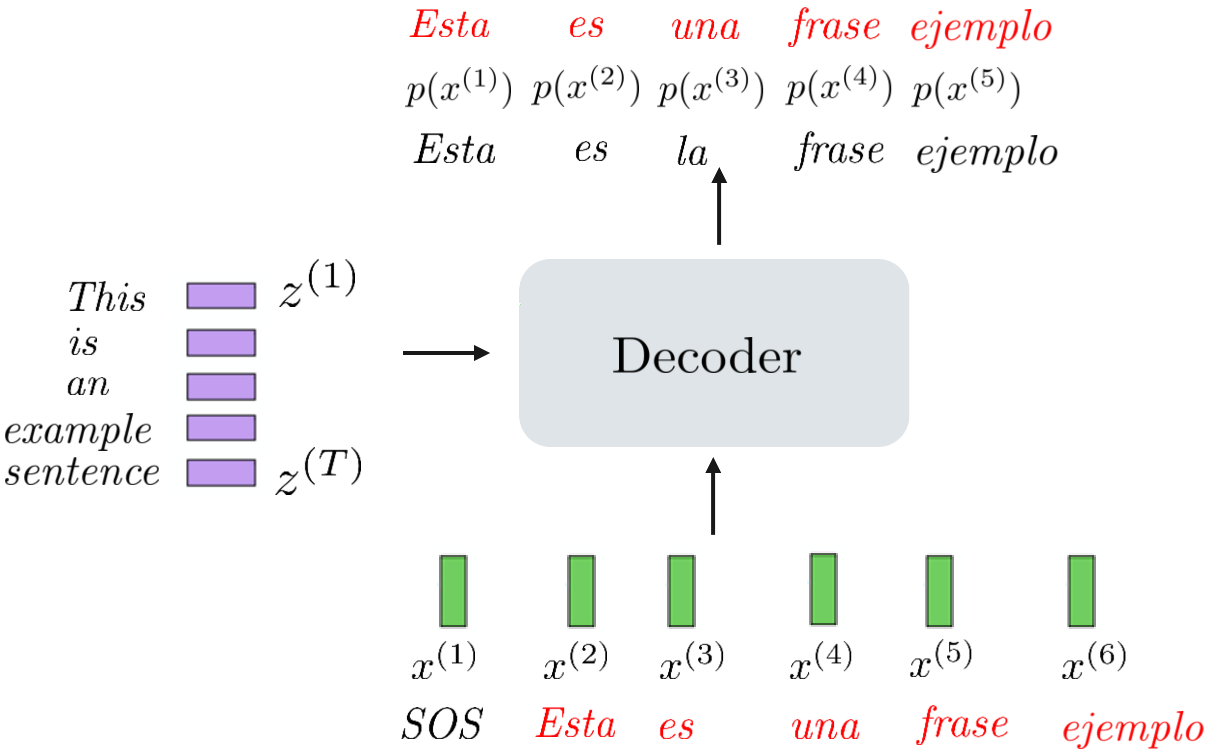
$T = 5$

Vigilada Mineducación

UNIVERSIDAD EAFIT

Decoder – Masked Attention - Training





Esta *es* *una* *frase* *ejemplo*

$p(x^{(1)})$ $p(x^{(2)})$ $p(x^{(3)})$ $p(x^{(4)})$ $p(x^{(5)})$

Esta *es* *la* *frase* *ejemplo*

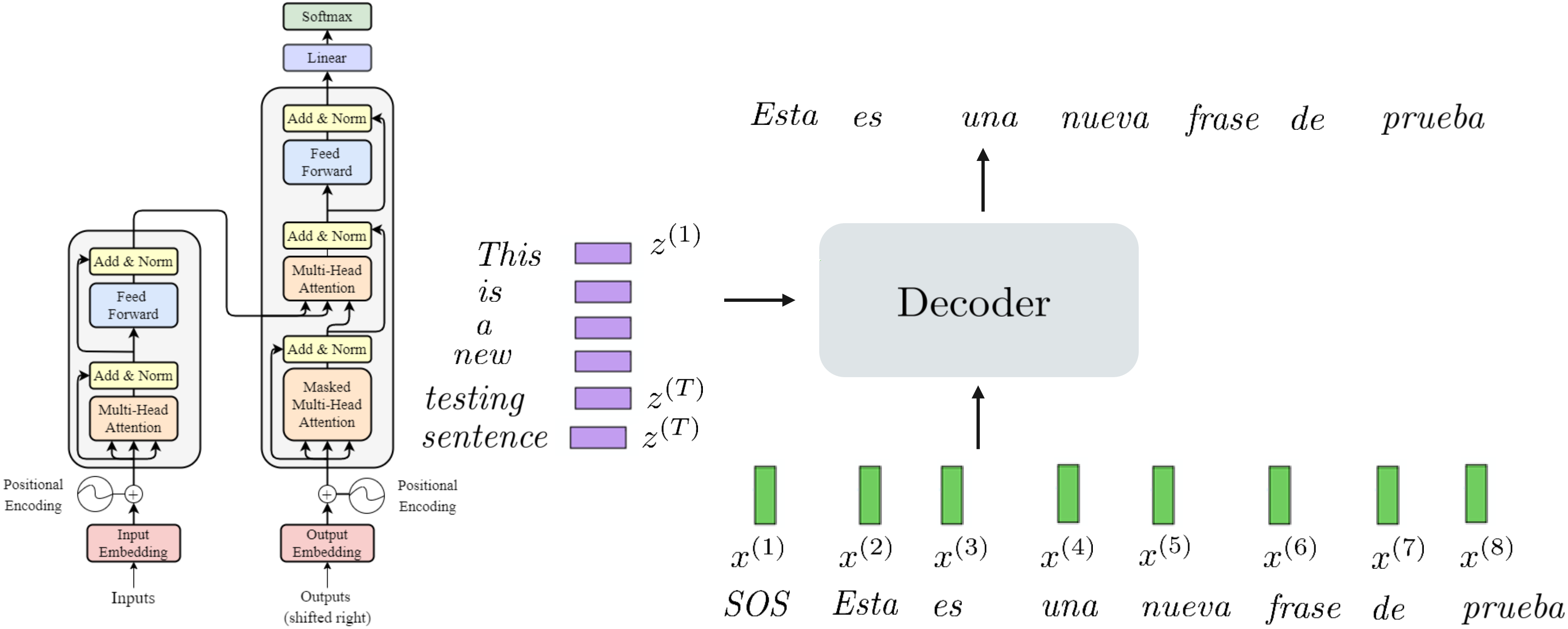
This $z^{(1)}$
is
an
example
sentence $z^{(T)}$

Decoder

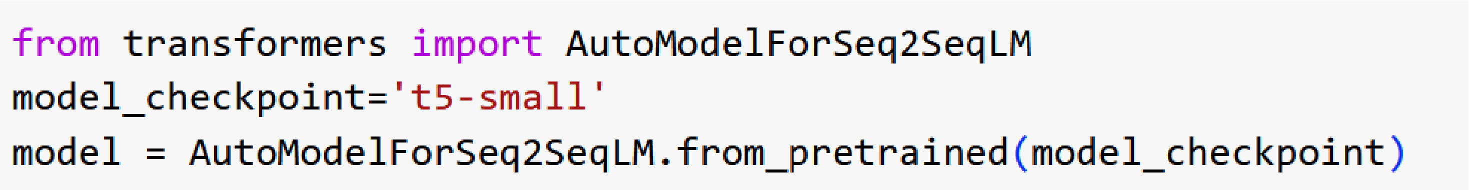
$x^{(1)}$ $x^{(2)}$ $x^{(3)}$ $x^{(4)}$ $x^{(5)}$ $x^{(6)}$

SOS *Esta* *es* *una* *frase* *ejemplo*

Transformer - Deploy



Transformer – Hugging Face

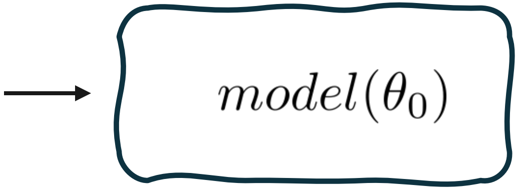


```
from transformers import AutoModelForSeq2SeqLM
model_checkpoint='t5-small'
model = AutoModelForSeq2SeqLM.from_pretrained(model_checkpoint)
```

Transfer Learning

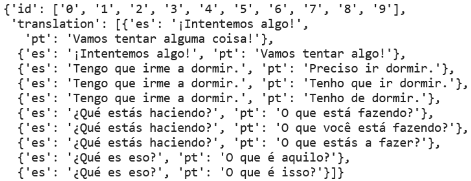
Pre-training

Supervised and self supervised learning

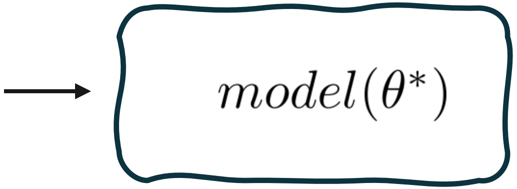


$model(\theta_0)$

Fine-tuning

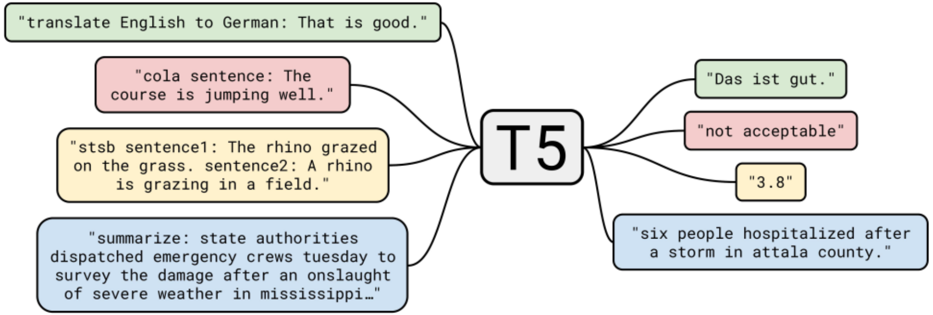


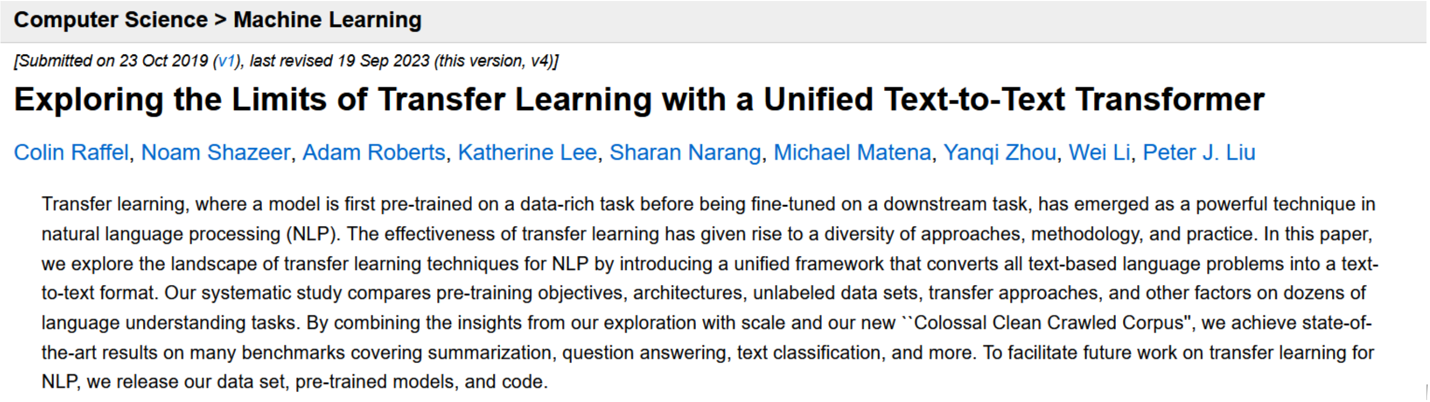
```
{'id': ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'],
 'translation': [{'es': '¡Intentemos algo!',
   'pt': 'Vamos tentar alguma coisa!'},
  {'es': '¡Intentemos algo!', 'pt': 'Vamos tentar algo!'},
  {'es': 'Tengo que irme a dormir.', 'pt': 'Preciso ir dormir.'},
  {'es': 'Tengo que irme a dormir.', 'pt': 'Tenho que ir dormir.'},
  {'es': 'Tengo que irme a dormir.', 'pt': 'Tenho de dormir.'},
  {'es': '¿Qué estás haciendo?', 'pt': 'O que está fazendo?'},
  {'es': '¿Qué estás haciendo?', 'pt': 'O que você está fazendo?'},
  {'es': '¿Qué estás haciendo?', 'pt': 'O que estás a fazer?'},
  {'es': '¿Qué es eso?', 'pt': 'O que é aquilo?'},
  {'es': '¿Qué es eso?', 'pt': 'O que é isso?'}]}
```



$model(\theta^*)$

Transfer Learning – Transformer T5


"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."



Computer Science > Machine Learning

[Submitted on 23 Oct 2019 (v1), last revised 19 Sep 2023 (this version, v4)]

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu

Transfer learning, where a model is first pre-trained on a data-rich task before being fine-tuned on a downstream task, has emerged as a powerful technique in natural language processing (NLP). The effectiveness of transfer learning has given rise to a diversity of approaches, methodology, and practice. In this paper, we explore the landscape of transfer learning techniques for NLP by introducing a unified framework that converts all text-based language problems into a text-to-text format. Our systematic study compares pre-training objectives, architectures, unlabeled data sets, transfer approaches, and other factors on dozens of language understanding tasks. By combining the insights from our exploration with scale and our new ``Colossal Clean Crawled Corpus", we achieve state-of-the-art results on many benchmarks covering summarization, question answering, text classification, and more. To facilitate future work on transfer learning for NLP, we release our data set, pre-trained models, and code.

Questions

Thank you 🤗