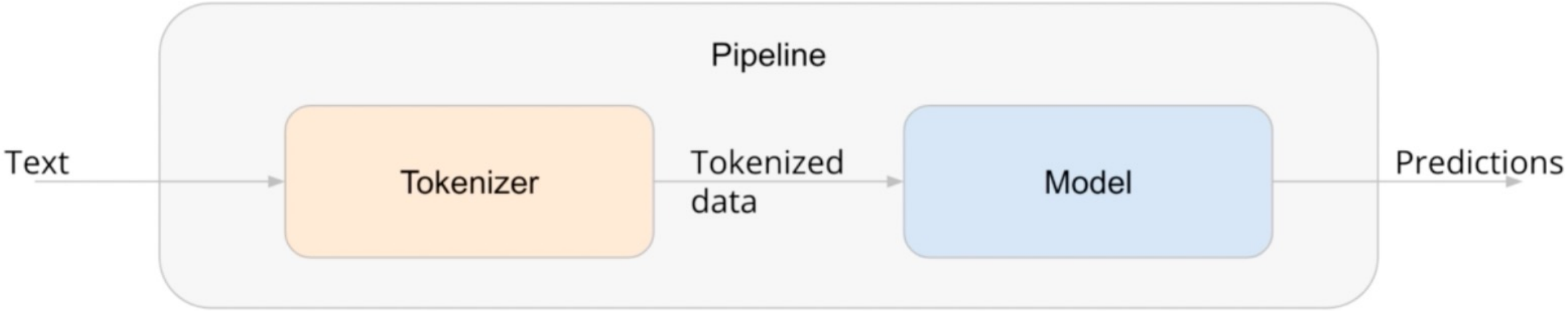
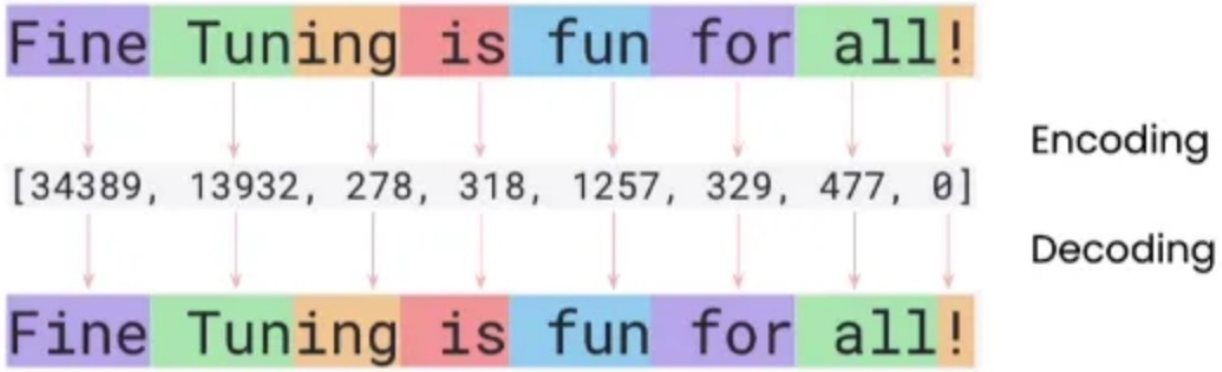
Tokenizers

Models & Tokenizers



- Tokenize the data



Models & Tokenizers

```
from transformers import AutoTokenizer

checkpoint = 'bert-base-cased'
tokenizer = AutoTokenizer.from_pretrained(checkpoint)
```

Models & Tokenizers

```
tokenizer("hello world")
```

```
OUTPUT:
{
   'input_ids': [101, 7592, 2088, 102],
   'token_type_ids': [0, 0, 0, 0]
   'attention_mask': [1, 1, 1, 1]
}
```

Models & Tokenizers

```
tokens = tokenizer.tokenize("hello world")
tokens
OUTPUT: ['hello', 'world']


ids = tokenizer.convert_tokens_to_ids(tokens)
ids
OUTPUT: [7592, 2088]
```

Models & Tokenizers

```
ids = tokenizer.encode("hello world")
ids
OUTPUT: [101, 7592, 2088, 102]
```

There are 4 ids because the input has been converted to:
[CLS] hello world [SEP]

```
tokenizer.convert_ids_to_tokens(ids)
OUTPUT: ['[CLS]', 'hello', 'world', '[SEP]']
```

```
tokenizer.decode(ids)
OUTPUT: '[CLS] hello world [SEP]'
```

Models & Tokenizers

```
tokenizer("hello world", return_tensors='pt')
```

```
OUTPUT:
{
   'input_ids': tensor([[ 101, 7592, 2088,  102]]),
   'token_type_ids': tensor([[0, 0, 0, 0]]),
   'attention_mask': tensor([[1, 1, 1, 1]])
}
```

Can also pass in 'np', 'tf'

Multiple Inputs

```
data = [
 "I like cats.",
 "Do you like cats too?",
]
model_inputs = tokenizer(data) # OK, but model won't accept
model_inputs = tokenizer(data, return_tensors='pt') # ERROR
```

Models & Tokenizers

```
model_inputs = tokenizer(
    data, padding=True, truncation=True, return_tensors='pt')
OUTPUT:
{
  'input_ids': tensor([[ 101, 1045, 8870,  102,    0,  0],
                       [ 101, 2079, 2017, 8870, 1029,102]]),
  'token_type_ids': tensor([[0, 0, 0, 0, 0, 0],
                            [0, 0, 0, 0, 0, 0]]),
  'attention_mask': tensor([[1, 1, 1, 1, 0, 0],
                            [1, 1, 1, 1, 1, 1]])
}
```

Double Asterisk

- Recall: converts a dictionary into named arguments

```
def my_function(name, email, password):
  # do some stuff

# normal function call
my_function("Alice", "alice@email.com", "12345")

# function call with explicit arguments
my_function(name="Alice", email="alice@email.com", password="12345")

# function call with dictionary - equivalent to above
d = {'name': 'Alice', 'email': 'alice@email.com', 'password'='12345'}
my_function(**d)
```

Reminder: PyTorch Model

```
class MyModel(torch.nn.Module):
  def __init__(self):
    super().__init__()
    # ...

  def forward(self, input_ids, attention_mask, ...):
    # do the computation
    return output


# usage
model = MyModel()
model(input_ids=some_data, attention_mask=other_data, ...)
```

Using the Model

```
from transformers import AutoModelForSequenceClassification

# should be the same checkpoint as tokenizer
model = AutoModelForSequenceClassification.from_pretrained(checkpoint)
```

Making Predictions

```
model_inputs = tokenizer(
    data, padding=True, truncation=True, return_tensors='pt')
outputs = model(**model_inputs)
```

Multiple Inputs

```
data = [
 "I like cats.",
 "Do you like cats too?",
]
model_inputs = tokenizer(data) # OK, but model won't accept
model_inputs = tokenizer(data, return_tensors='pt') # ERROR
```

About us

At Contoso, we empower organizations to foster collaborative thinking to further drive workplace innovation. By closing the loop and leveraging agile frameworks, we help businesses grow organically and foster a consumer-first mindset.

Company overview

Product overview

First beautifully designed product that's both stylish and functional

Problem

- Market gap: few, if any, products on the market help customers like we do
- Customers: 66% of US consumers spend money on multiple products that only partially resolves their issue
- Financials: millennials account for about a quarter of the $48 billion spent on other products in 2018
- Costs: loss of productivity costing consumers thousands of dollars
- Usability: customers want something easy to use that helps make their life easier

Product benefits

Online store and market swap

Our competition

- Our product is priced below that of other companies on the market
- Design is simple and easy to use, compared to the complex designs of the competitors
- Affordability is the main draw for our consumers to our product

- Company A product is more expensive
- Companies B & C product is expensive and inconvenient to use
- Companies D & E product is affordable, but inconvenient to use

Product overview

- Unique
- First to market
- Tested
- Authentic

- Only product specifically dedicated to this niche market
- First beautifully designed product that's both stylish and functional
- Conducted testing with college students in the area
- Designed with the help and input of experts in the field

Growth strategy

- Feb 20XX: roll out product to high profile or top-level participants to help establish the product
- May 20XX: release the product to the public and monitor press release and social media accounts
- Oct 20XX: gather feedback and adjust product design as necessary

Market overview

- Opportunity to build
- Fully inclusive market
- Total addressable market
- Freedom to invent
- Selectively inclusive market
- Serviceable available market

	Clients	Orders	Gross revenue	Net revenue
20XX	10	100	$10,000	$7,000
20XX	20	200	$20,000	$16,000
20XX	30	300	$30,000	$25,000
20XX	40	400	$40,000	$30,000

Solution

Our product makes consumer lives easier, and no other product on the market offers the same features

- Gen Z (18-25 years old)
- Reduce expenses for replacement products
- Simple design that gives customers the targeted information they need

- Close the gap
- Target audience
- Cost savings
- Easy to use

Financials

	Year 1	Year 2	Year 3
Income			
Users	50,000	400,000	1,600,000
Sales	500,000	4,000,000	16,000,000
Average price per sale	75	80	90
Revenue @ 15%	5,625,000	48,000,000	216,000,000
Gross profit	**5,625,000**	**48,000,000**	**216,000,000**

Thank you

Brita Tamm

502-555-0152

brita@firstupconsultants.com

www.firstupconsultants.com