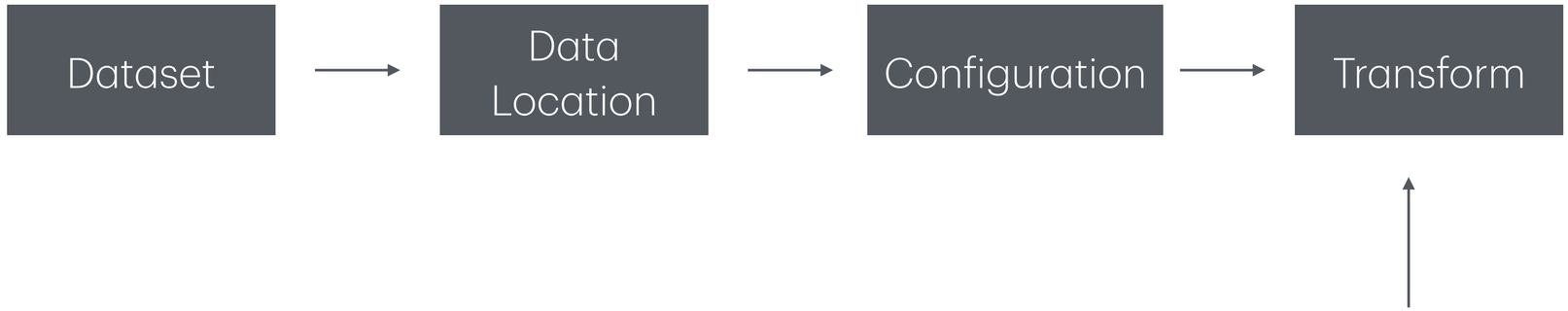
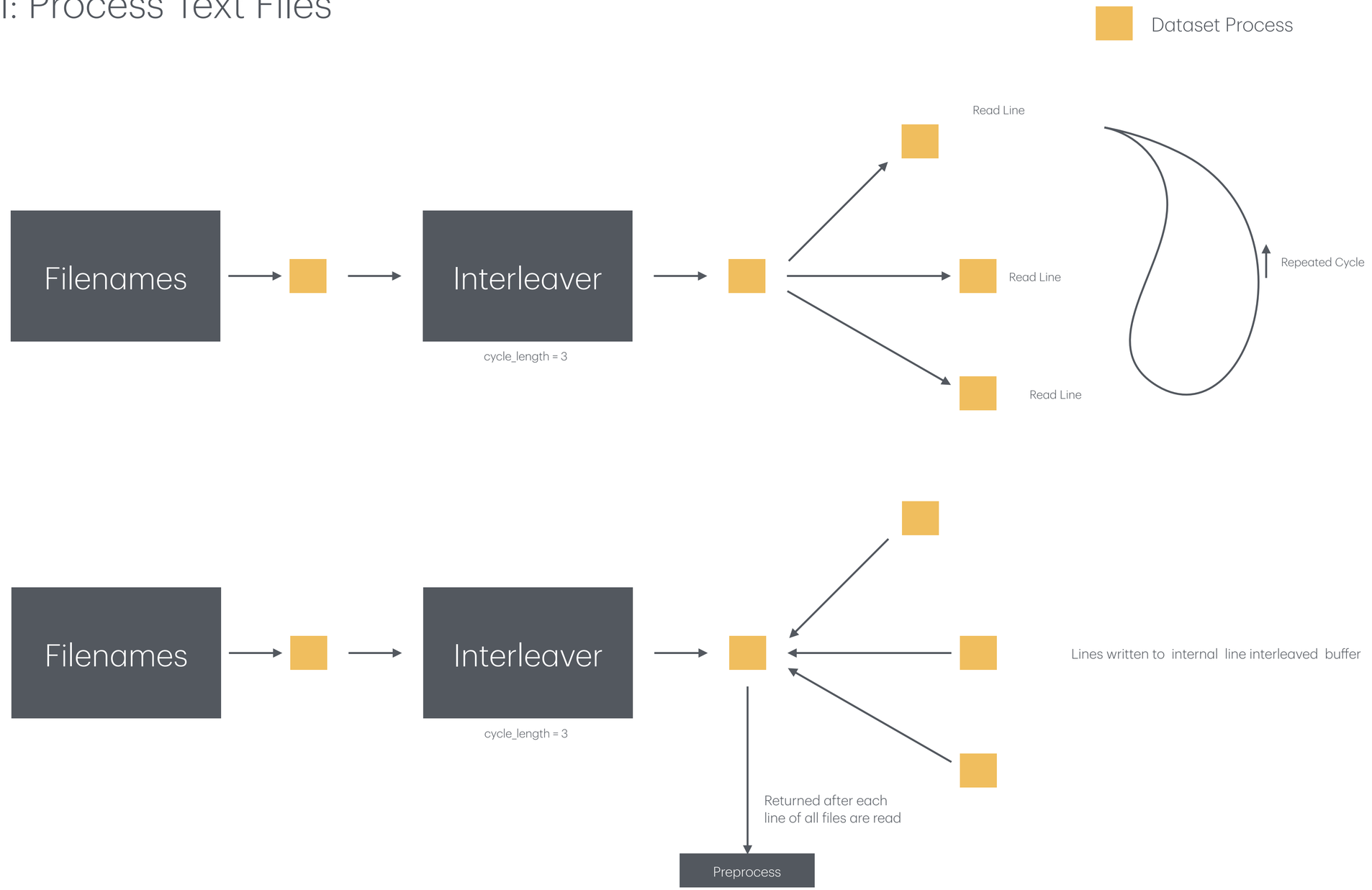
Loading Preprocessing

Data API

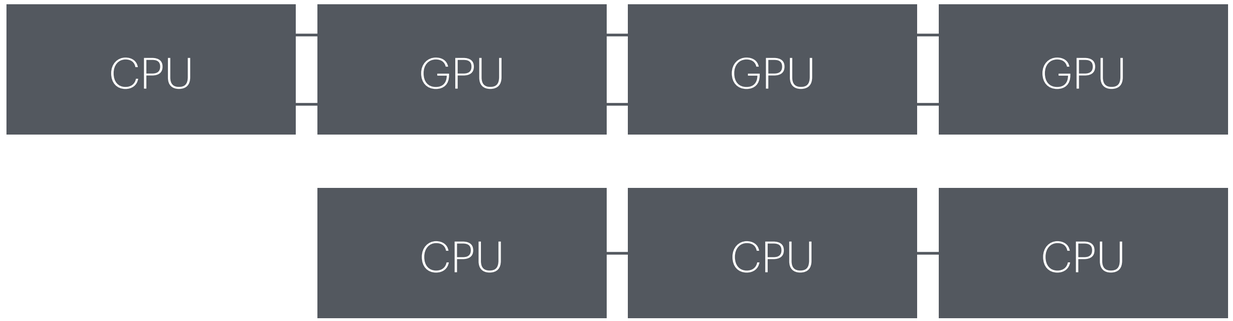

Dataset
Data Location
Configuration
Transform
Tensorflow handles and hides the processing details

Data API: Process Text Files



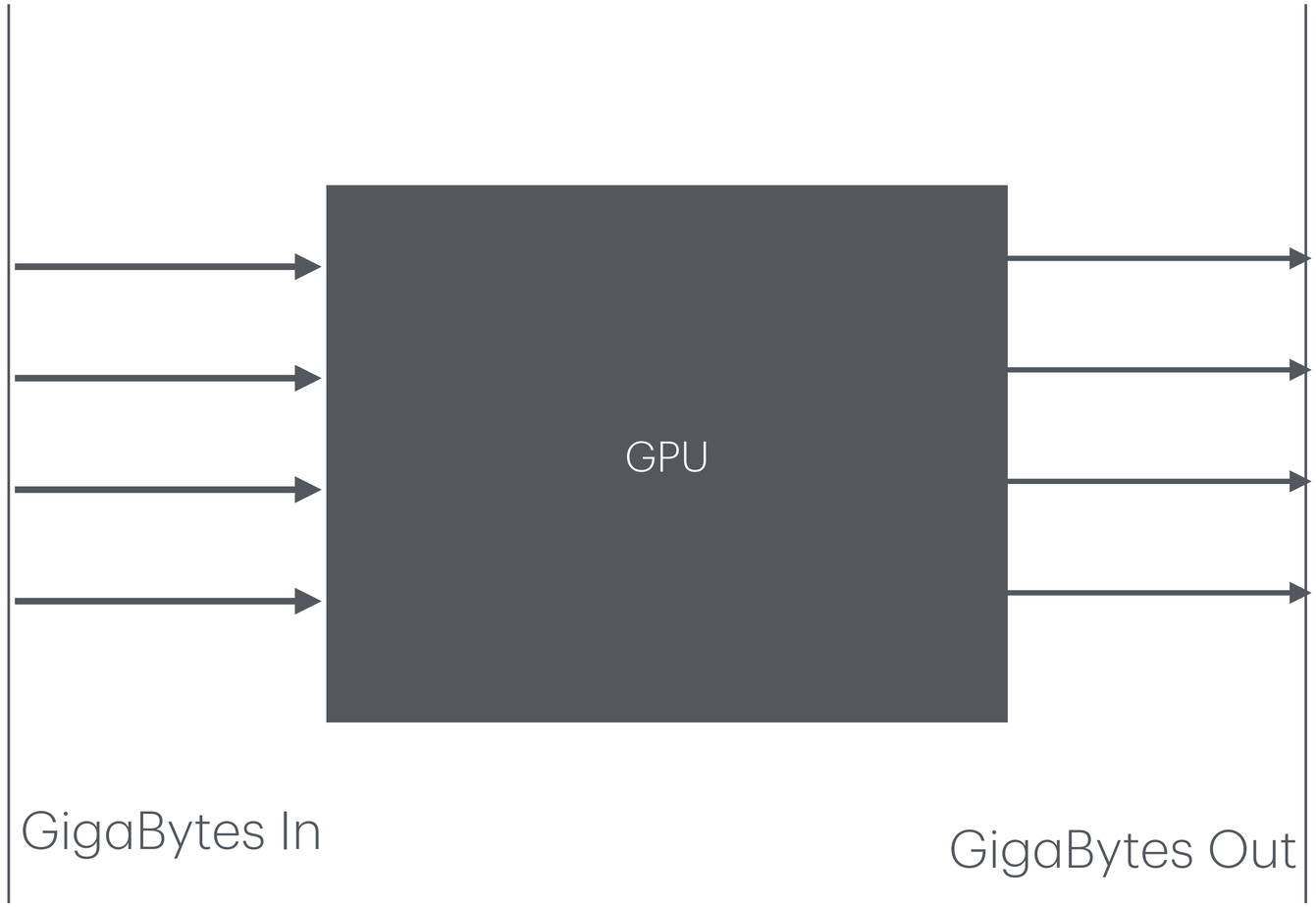
Prefetching

Without Prefetching



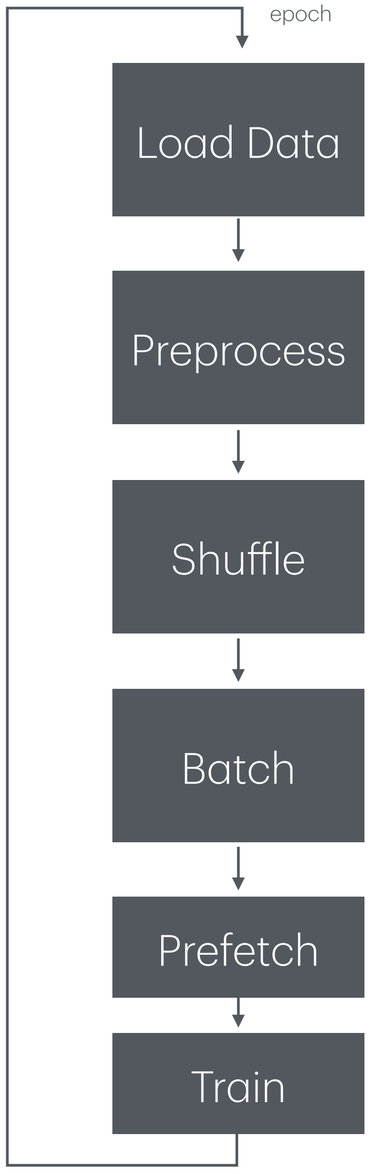
With Prefetching

GPU Card

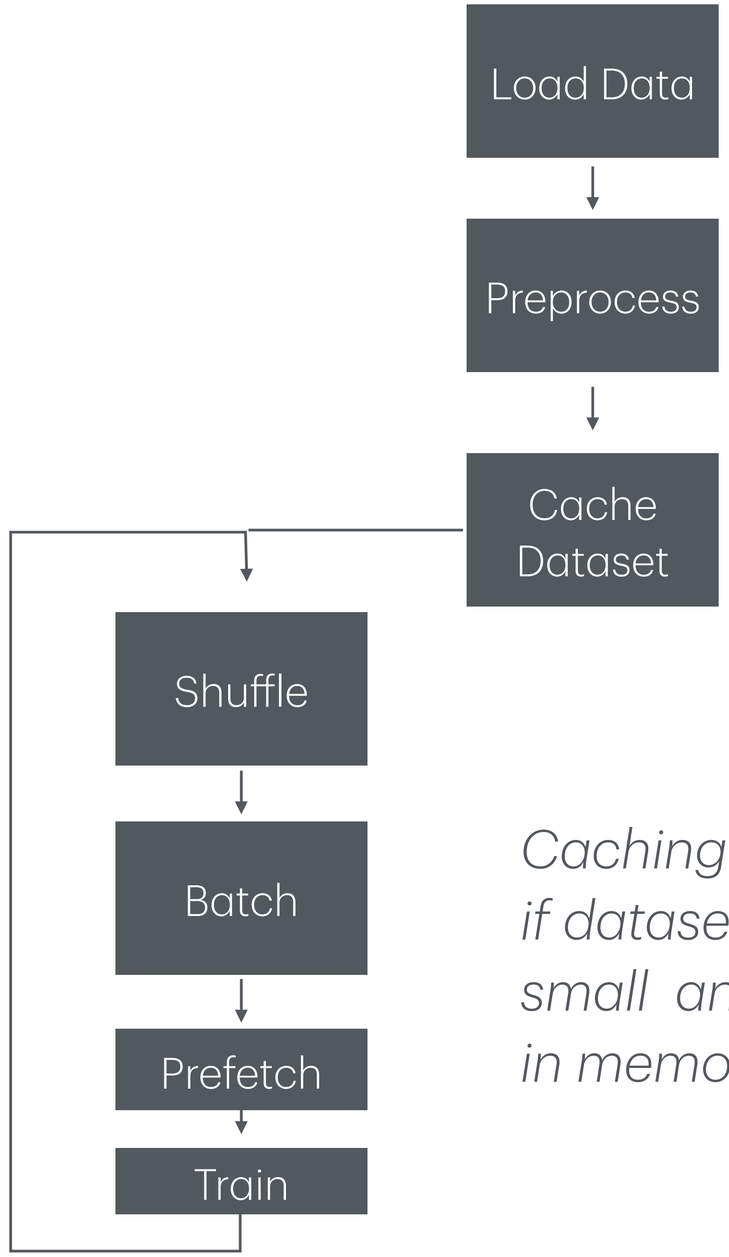


GigaBytes Per Second

Cache Data



Normal Operation



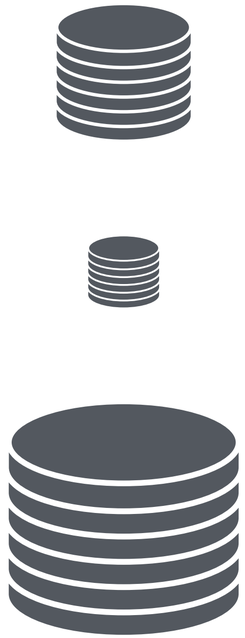
Caching used if dataset is small and fits in memory

Cached Operation

TF Record

If loading and preprocessing bottlenecks
develop use TF Record Format

TF Record



Sequence of binary records of varying sizes

Files typically contain **serialized protobufs** binary format

Format

- Length
- CRC Length Checksum
- Data
- CRC Data Checksum

TF Record

Python files generated with protobuf definitions(i.e. Person Class). The process is the same for other programming languages

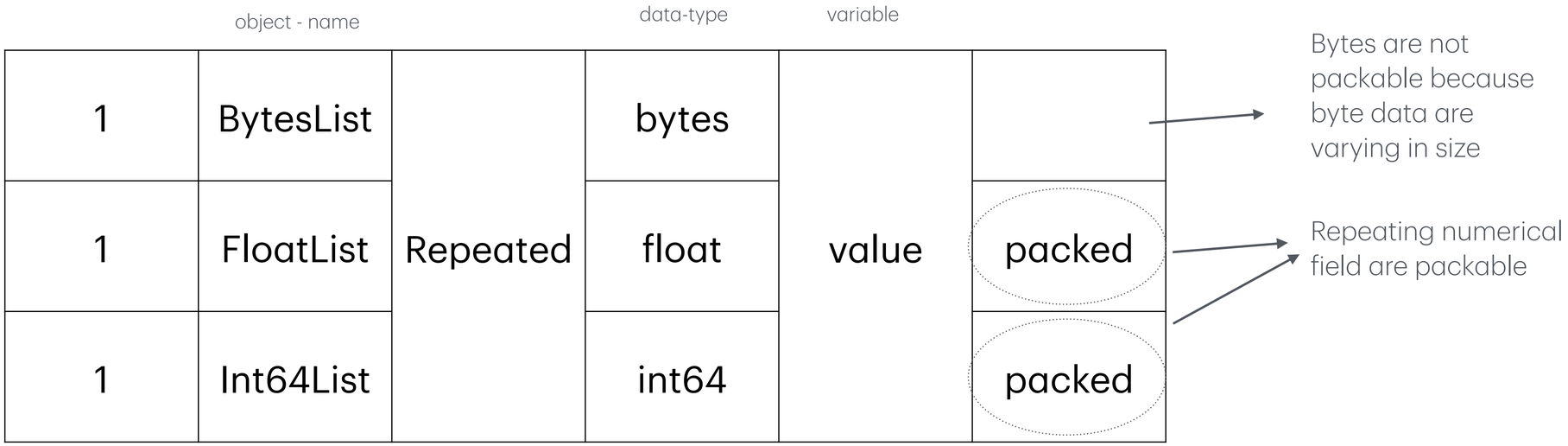
Requires protoc compiler

Defined in .proto file

```
syntax = "proto3"
message Person {
  string name = 1;
  int32 id = 2;
  repeated string email = 3;
}
```

Tensorflow has pre-compiled protobuf definitions. Most of the time Tensorflow has all thats needed for programming tasks

Tensorflow Protobufs



Tensorflow Protobufs

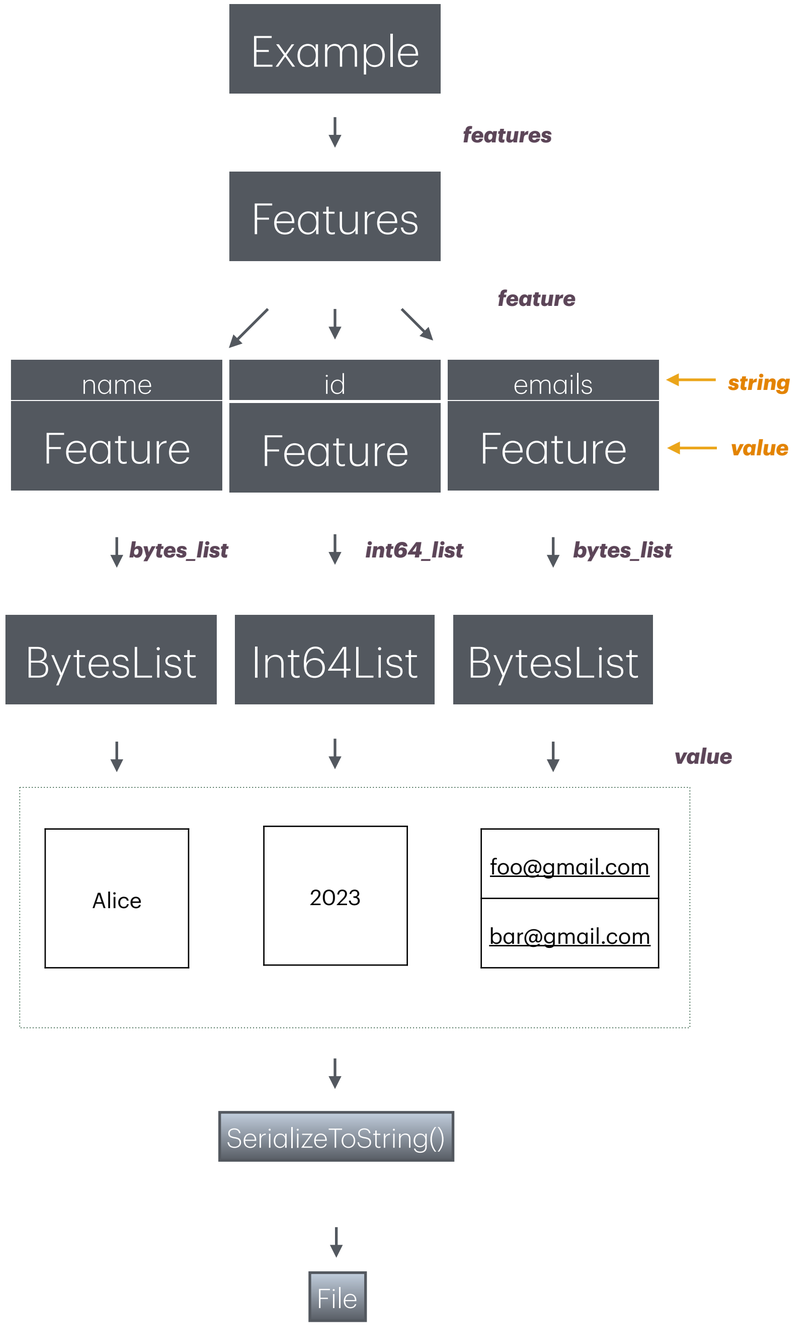
	object - name		data-type	variable	
1	Feature		BytesList	bytes_list	oneof
2	Feature		FloatList	float_list	oneof
3	Feature		Int64List	int64_list	oneof

Tensorflow Protobufs

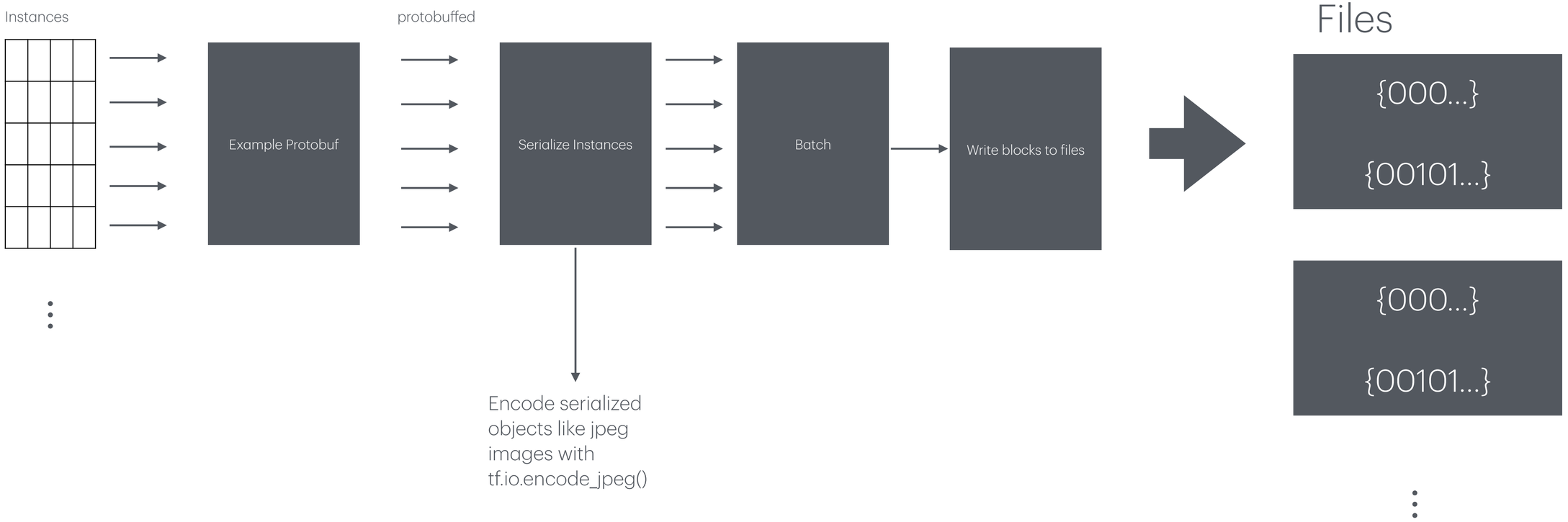
	object - name		data-type	variable
1	Features		<String,Feature>	feature

Tensorflow Protobufs

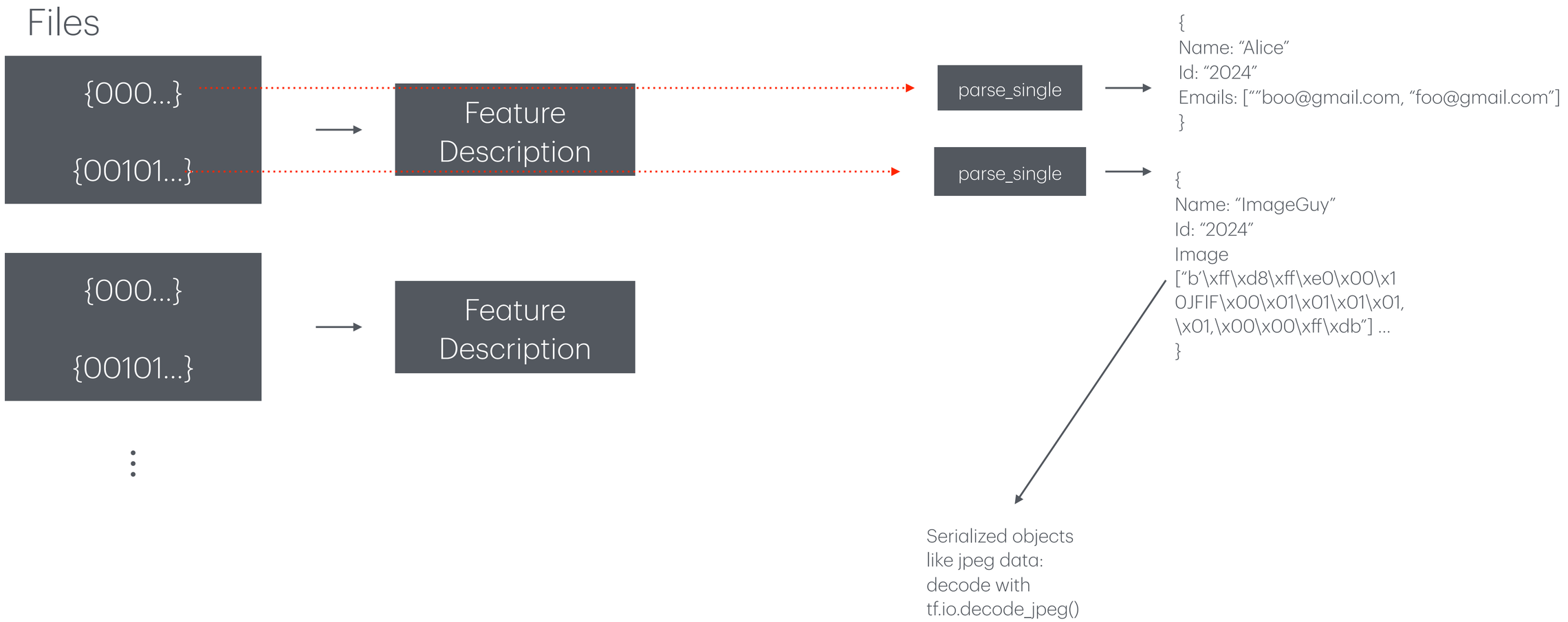
	object - name		data-type	variable
1	Example		Features	features


Example
features
Features
feature
name
id
emails
string
Feature
Feature
Feature
value
bytes_list
int64_list
bytes_list
BytesList
Int64List
BytesList
value
Alice
2023
foo@gmail.com
bar@gmail.com
SerializeToString()
File

Real World Examine

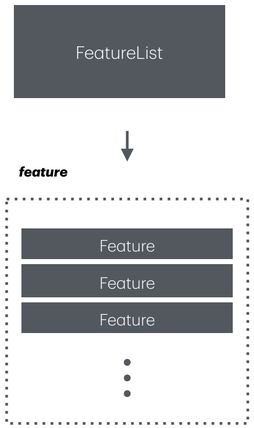



```
tf.io.encode_jpeg(b)
<tf.Tensor: shape=(), dtype=string,
numpy=b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x01\x01,\x01,\x00\x00\xff\xdb\x00C\x00\x02\x01
\x01\x01\x01\x02\x01\x01\x01\x02\x02\x02\x02\x02\x04\x03\x02\x02\x02\x02\x05\x04\x04\x03
\x04\x06\x05\x06\x06\x06\x05\x06\x06\x06\x07\t\x08\x06\x07\t\x07\x06\x06\x08\x0b\x08\t\n\n
\n\n\x06\x08\x0b\x0c\x0b\n\x0c\t\n\n\xff\xdb\x00C\x01\x02\x02\x02\x02\x02\x02\x05\x03\x03\x
05\n\x07\x06\x07\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n
\n\n\n\n\n\n\xff\xc0\x00\x11\x08\x00\x03\x00\x03\x03\x01"\x00\x02\x11\x01\x03\x11\x01\xff\xc4
\x00\x1f\x00\x00\x01\x05\x01\x01\x01\x01\x01\x01\x00\x00\x00\x00\x00\x00\x00\x00\x01\x02\x0
3\x04\x05\x06\x07\x08\t\n\x0b\xff\xc4\x00\xb5\x10\x00\x02\x01\x03\x03\x02\x04\x03\x05\x05\x0
4\x04\x00\x00\x01}\x01\x02\x03\x00\x04\x11\x05\x12!
```


Files
{000...}
{00101...}
Feature Description
{000...}
{00101...}
Feature Description
parse_single
parse_single
{
Name: “Alice”
Id: “2024”
Emails: [“”boo@gmail.com, “foo@gmail.com”]
}
{
Name: “ImageGuy”
Id: “2024”
Image
[“b’\xff\xd8\xff\xe0\x00\x1
0JFIF\x00\x01\x01\x01\x01,
\x01,\x00\x00\xff\xdb”] ...
}
Serialized objects
like jpeg data:
decode with
tf.io.decode_jpeg()

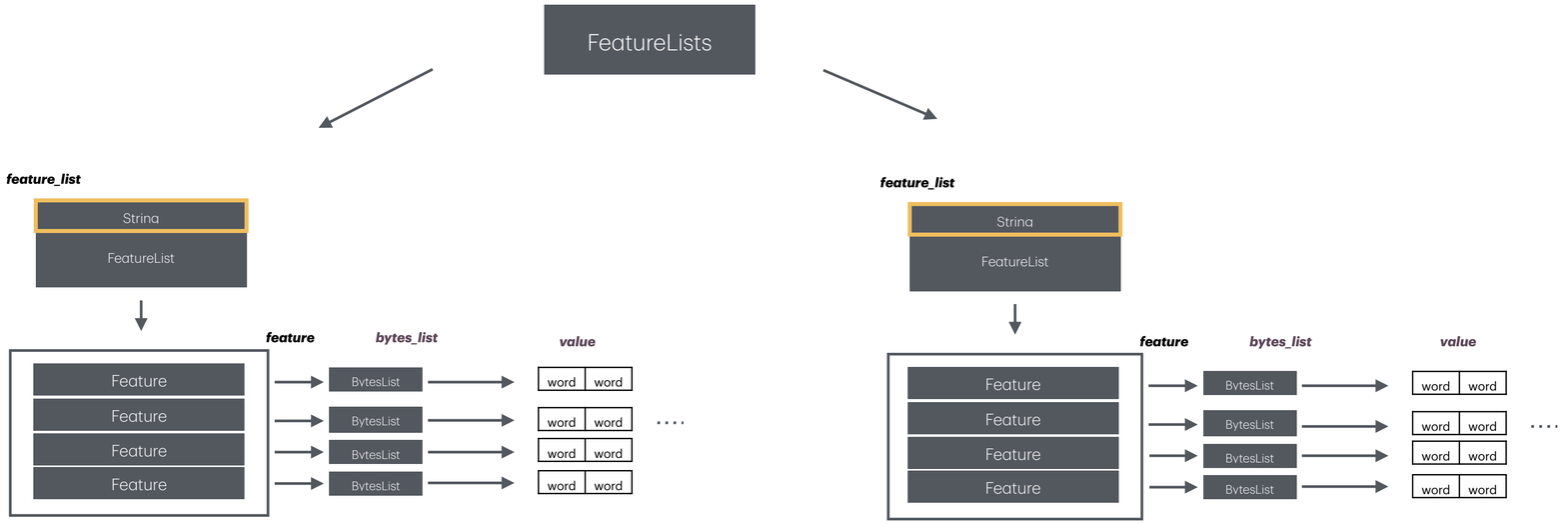
SequenceExample Protobuf

	object - name		data-type	variable	
1	FeatureList	Repeated	Feature	feature	



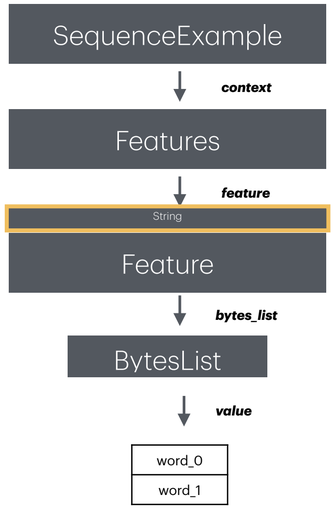
SequenceExample Protobuf

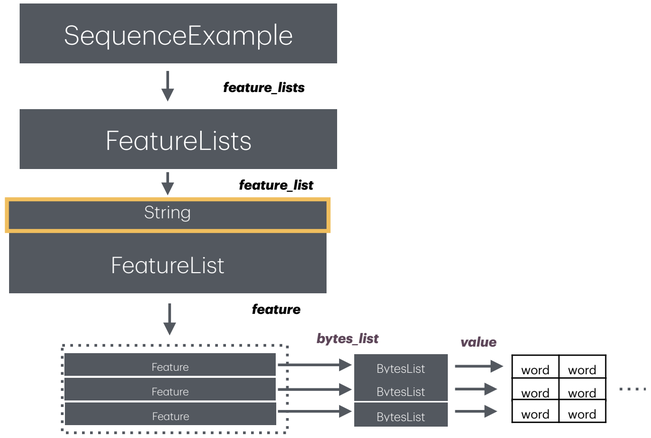
object - name		data-type		variable	
1	FeatureLists		map<string,FeatureList>	feature_list	



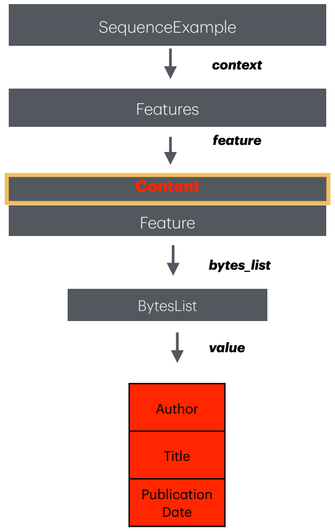
SequenceExample Protobuf

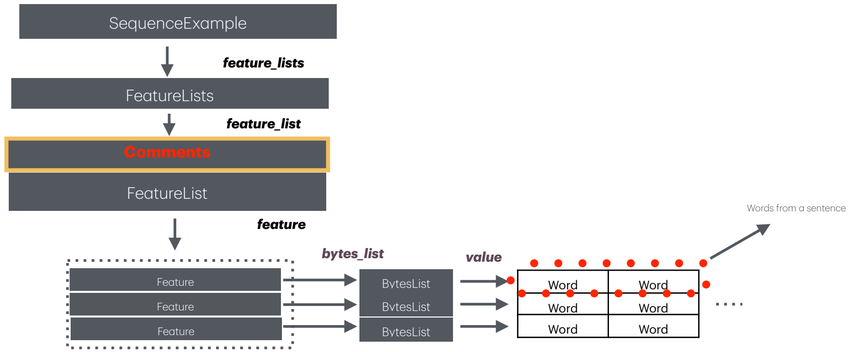
	object - name		data-type	variable	
1	SequenceExample		Features	context	
2	SequenceExample		FeatureLists	feature_lists	



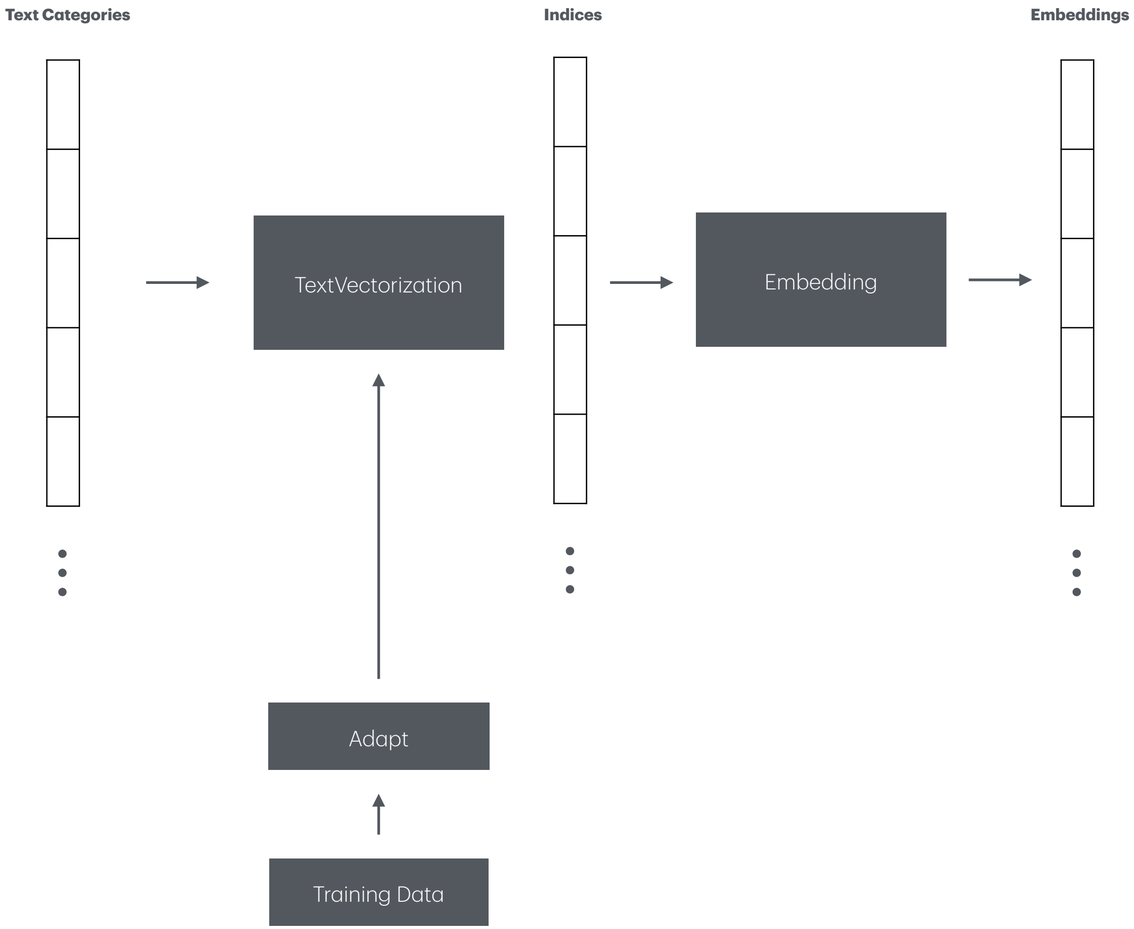


SequenceExample Protobuf

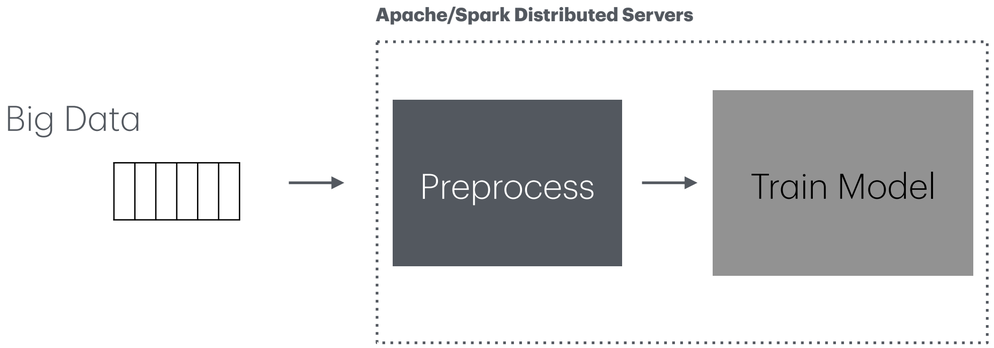


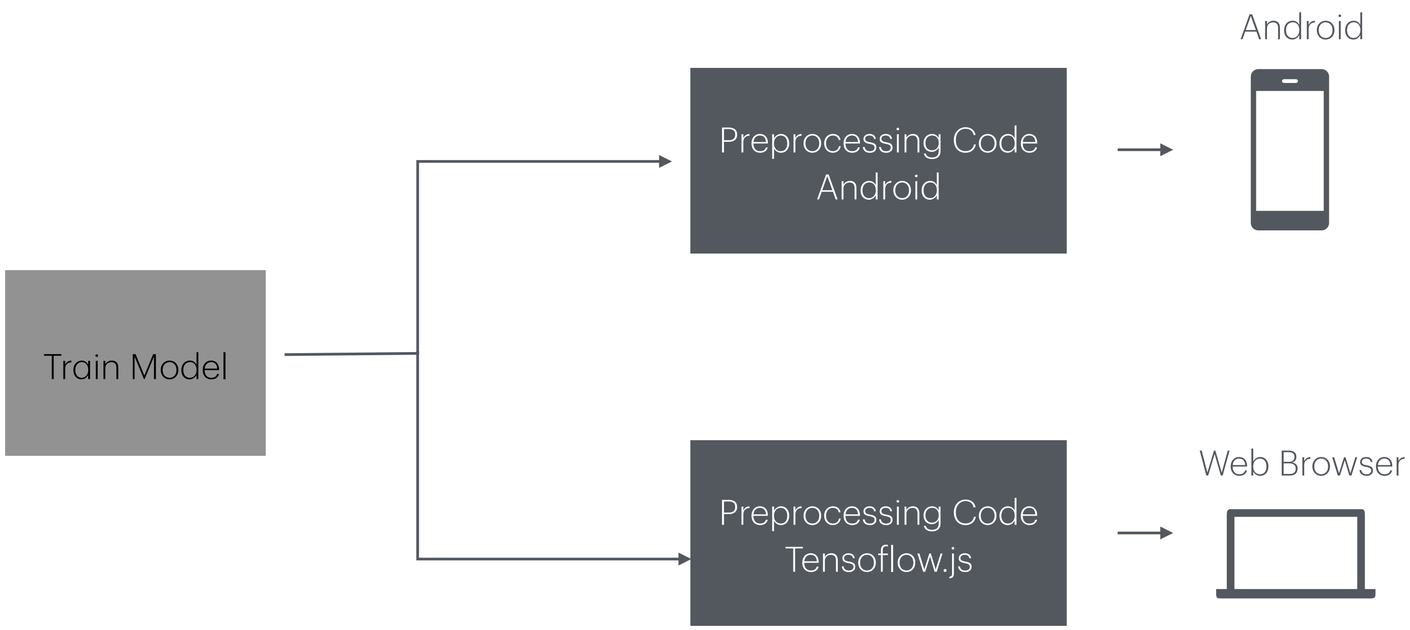


Embeddings


Text Categories
Indices
Embeddings
TextVectorization
Embedding
Adapt
Training Data

Deployment: Case 1





Deployment: Case 2

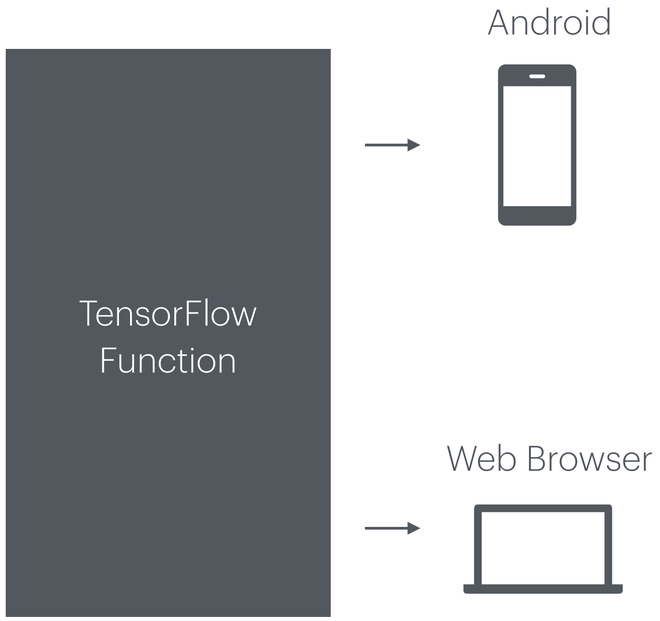
TF Transform

↓

TensorFlow Function

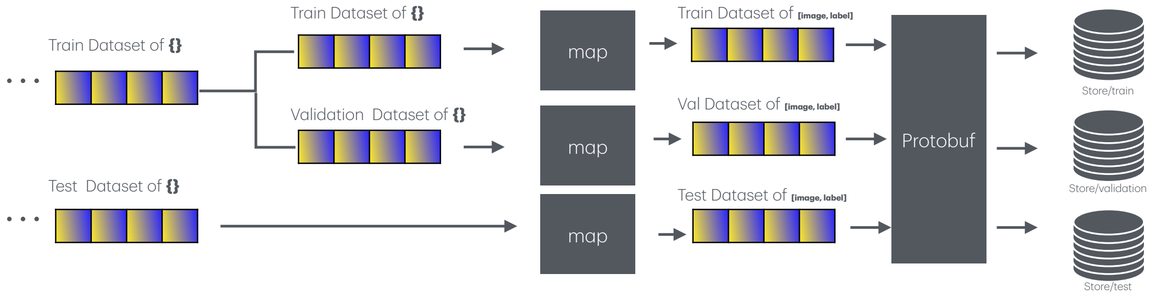
Define preprocess function for training data

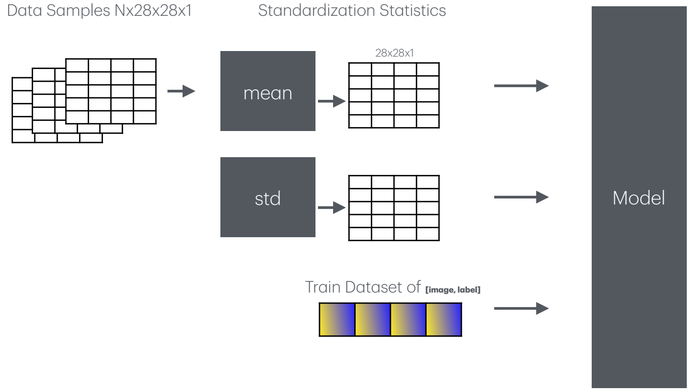
TF Transform generates equivalent Tensorflow function to plug into model



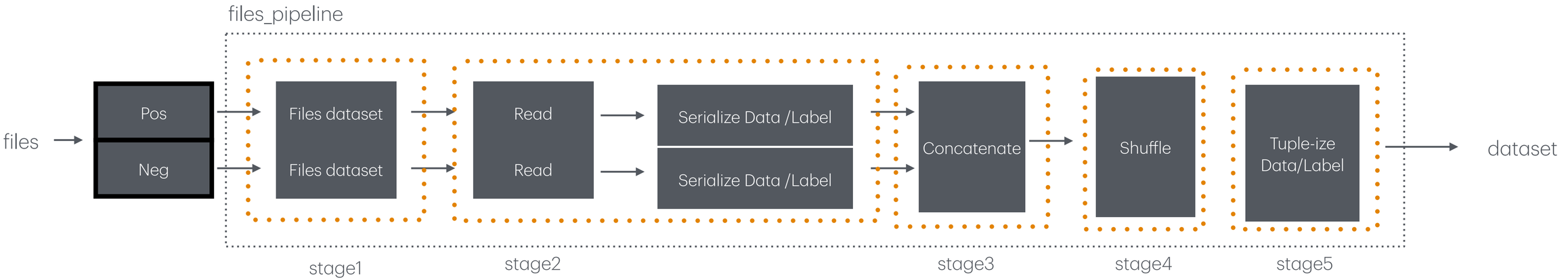
Single function can be dropped into any system using Tensorflow

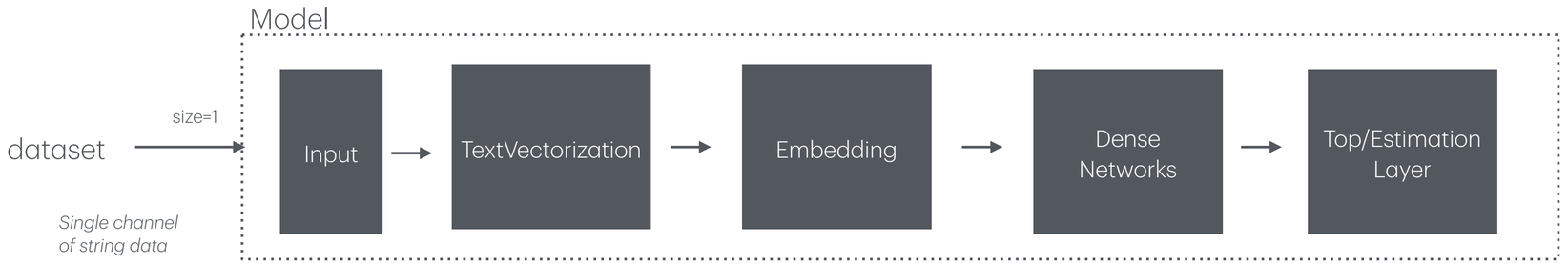
MNIST Problem



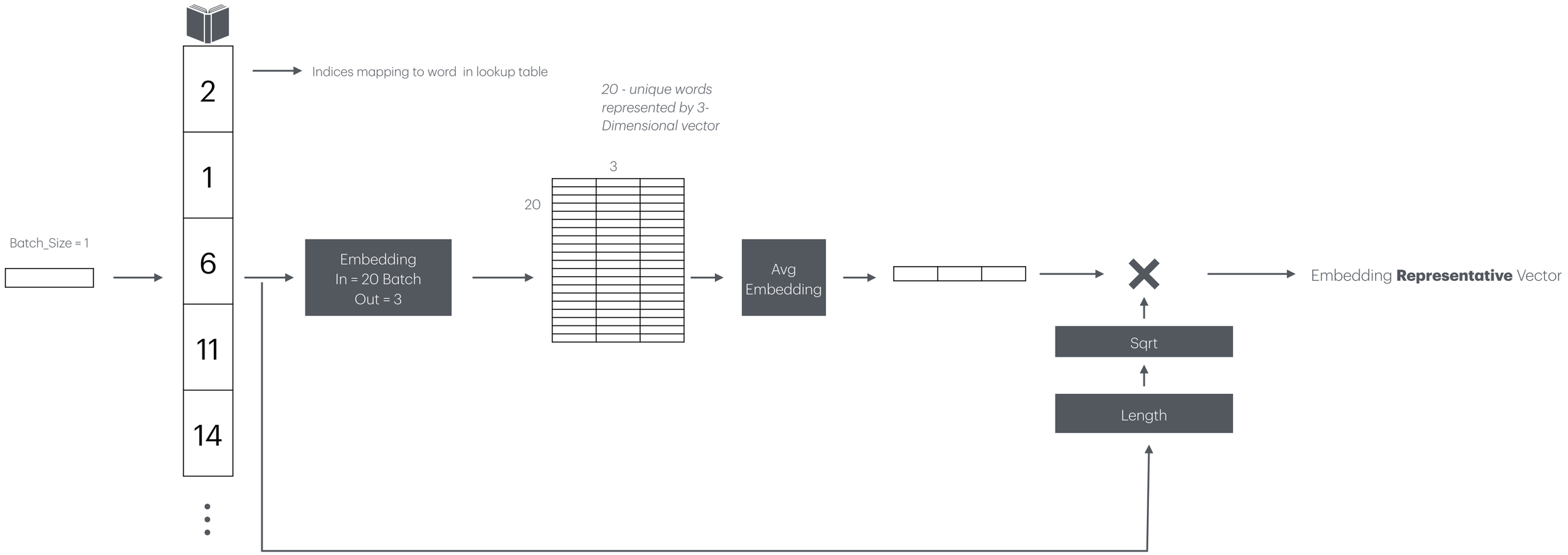


Movie Review Dataset

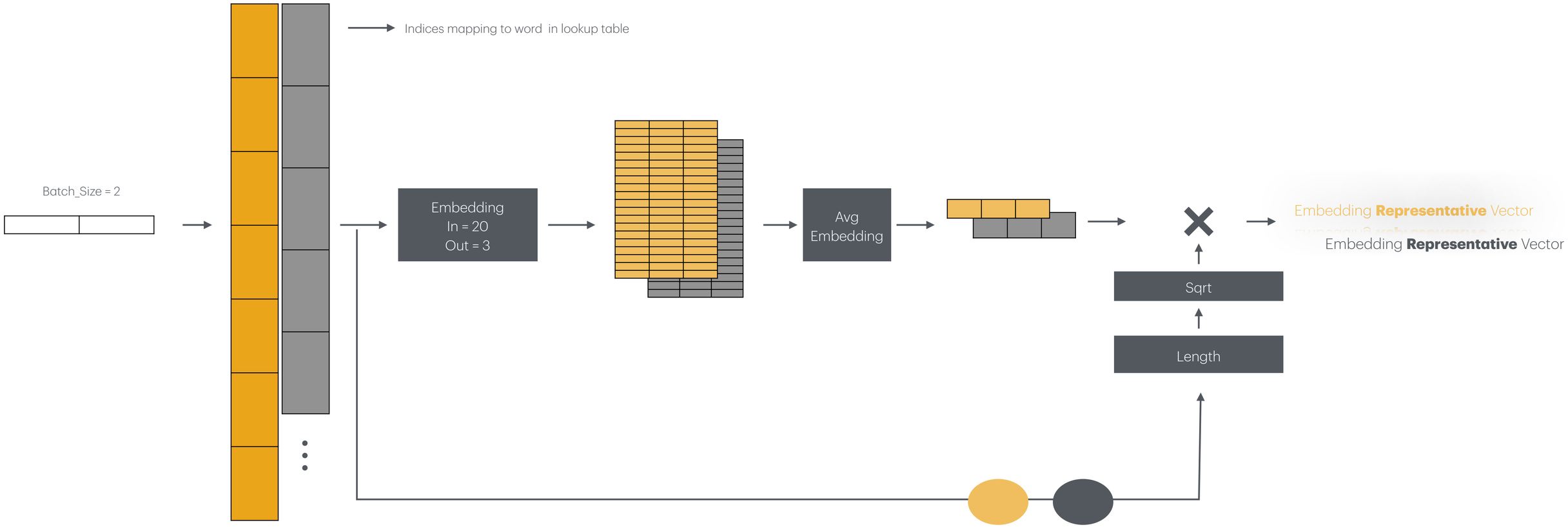
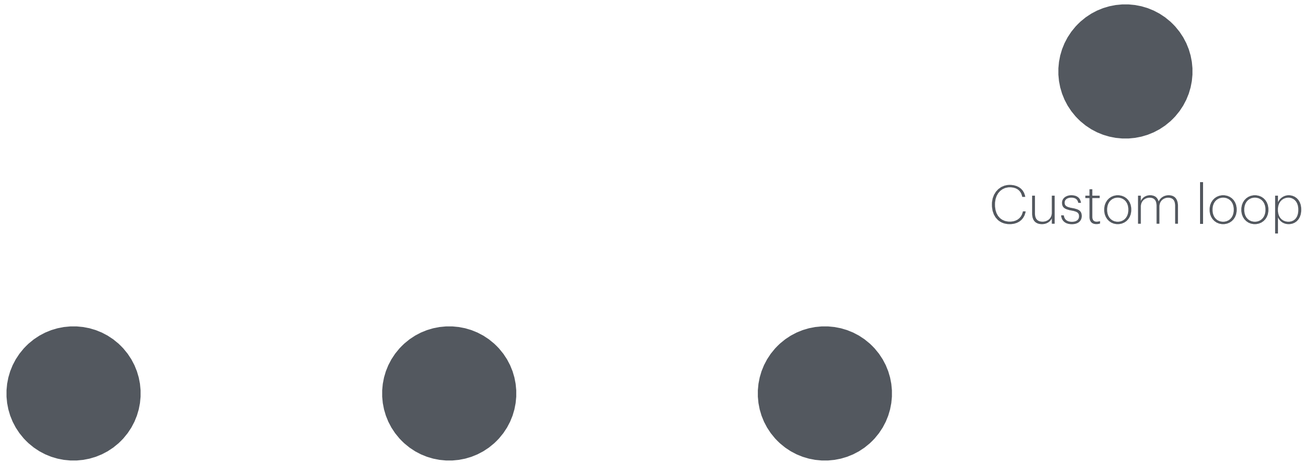




TextVectorization-Embedding



TextVectorization-Embedding



Branch