#WWDC18

Introducing Natural Language

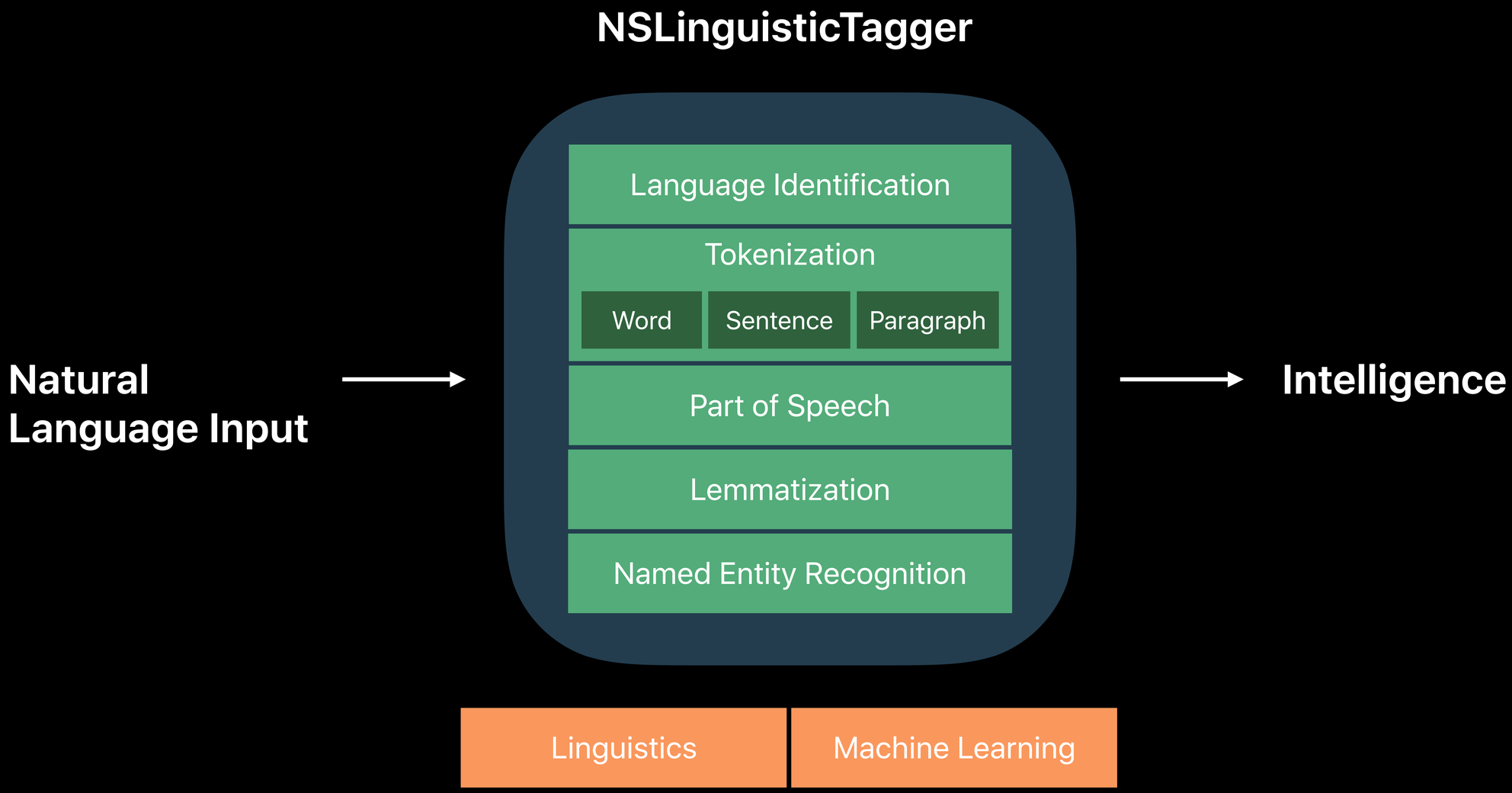
Session 713

Doug Davidson, Senior Software Engineer
Vivek Kumar Rangarajan Sridhar, Software Engineering Manager

Natural Language Input

Typed text
Transcribed speech
Recognized handwriting

Intelligence


NSLinguisticTagger
Natural Language Input
Language Identification
Tokenization
Word
Sentence
Paragraph
Part of Speech
Lemmatization
Named Entity Recognition
Intelligence
Linguistics
Machine Learning

What’s new in NLP ?

Natural Language Framework

Swift APIs for NLP

Swift APIs for NLP

Custom NLP models

Swift APIs for NLP

Custom NLP models

Performance

Swift APIs for NLP

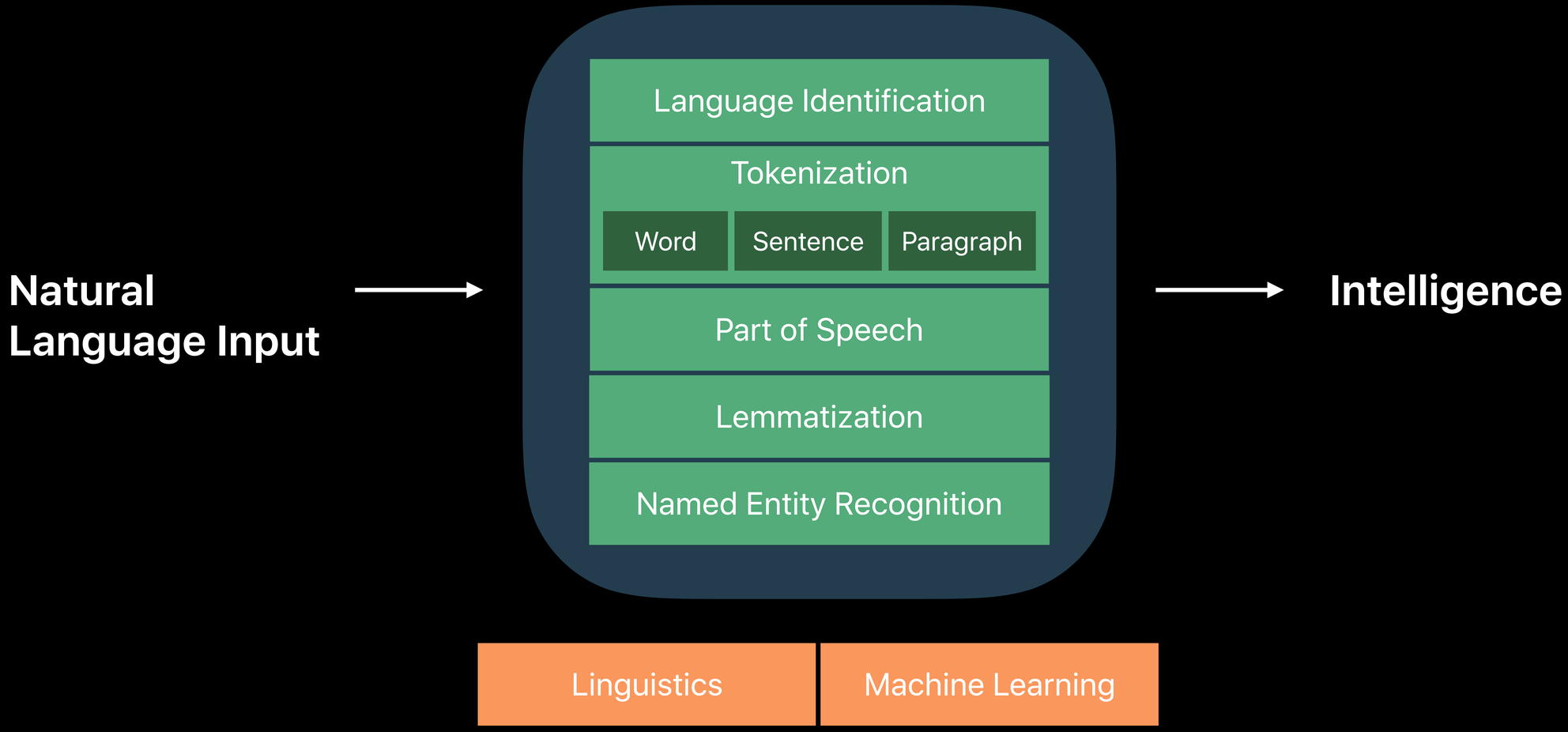
Custom NLP models

Performance

Privacy

Natural Language

Natural Language




9:41 AM
100%
<
John M
Design is coming together but needs some final tweaks
It’s getting late and I’m tired. Will pick it up tomorrow morning...good night!

9:41 AM 100%

< John M

Design is coming together but needs some final tweaks

It’s getting late and I’m tired. Will pick it up tomorrow morning...good night!

9:41 AM 100%

< John M

Design is coming together but needs some final tweaks

It's getting late and I'm tired. Will pick it up tomorrow morning...good night!

9:41 AM 100%

< John M

Design is coming together but needs some final tweaks

困死啦睡觉去了

9:41 AM 100%

< John M

Design is coming together but needs some final tweaks

It's getting late and I'm tired. Will pick it up tomorrow morning...good night!

9:41 AM 100%

< John M

Design is coming together but needs some final tweaks

困死啦睡觉去了

Language Identification

```
import NaturalLanguage

let recognizer = NLLanguageRecognizer()

recognizer.processString("困死啦睡觉去了")

let lang = recognizer.dominantLanguage

let hypotheses  = recognizer.languageHypotheses(withMaximum:2)
```

Language Identification

```
import NaturalLanguage

let recognizer = NLLanguageRecognizer()

recognizer.processString("困死啦睡觉去了")

let lang = recognizer.dominantLanguage

let hypotheses  = recognizer.languageHypotheses(withMaximum:2)
```

Language Identification

```
import NaturalLanguage

let recognizer = NLLanguageRecognizer()

recognizer.processString("困死啦睡觉去了")

let lang = recognizer.dominantLanguage

let hypotheses  = recognizer.languageHypotheses(withMaximum:2)
```

Language Identification

```
import NaturalLanguage

let recognizer = NLLanguageRecognizer()

recognizer.processString("困死啦睡觉去了")

let lang = recognizer.dominantLanguage

let hypotheses  = recognizer.languageHypotheses(withMaximum:2)
```

Language Identification

```
import NaturalLanguage

let recognizer = NLLanguageRecognizer()

recognizer.processString("困死啦睡觉去了")

let lang = recognizer.dominantLanguage

let hypotheses  = recognizer.languageHypotheses(withMaximum:2)
```

zh-Hans: Simplified Chinese

Language Identification

```
import NaturalLanguage

let recognizer = NLLanguageRecognizer()

recognizer.processString("困死啦睡觉去了")

let lang = recognizer.dominantLanguage

let hypotheses  = recognizer.languageHypotheses(withMaximum:2)
```

zh-Hans: Simplified Chinese

Tokenization

```
import NaturalLanguage

let tokenizer = NLTokenizer(unit: .word)

let str = “困死啦睡觉去了”

let strRange = str.startIndex ..< str.endIndex

tokenizer.string = str

let tokenArray = tokenizer.tokens(for: strRange)
```

Tokenization

```
import NaturalLanguage

let tokenizer = NLTokenizer(unit: .word)

let str = “困死啦睡觉去了”

let strRange = str.startIndex ..< str.endIndex

tokenizer.string = str

let tokenArray = tokenizer.tokens(for: strRange)
```

Tokenization

```
import NaturalLanguage

let tokenizer = NLTokenizer(unit: .word)

let str = “困死啦睡觉去了”

let strRange = str.startIndex ..< str.endIndex

tokenizer.string = str

let tokenArray = tokenizer.tokens(for: strRange)
```

Tokenization

```
import NaturalLanguage

let tokenizer = NLTokenizer(unit: .word)

let str = “困死啦睡觉去了”

let strRange = str.startIndex ..< str.endIndex

tokenizer.string = str

let tokenArray = tokenizer.tokens(for: strRange)
```

Tokenization

```
import NaturalLanguage

let tokenizer = NLTokenizer(unit: .word)

let str = “困死啦睡觉去了”

let strRange = str.startIndex ..< str.endIndex

tokenizer.string = str

let tokenArray = tokenizer.tokens(for: strRange)
```

Tokenization

```
import NaturalLanguage

let tokenizer = NLTokenizer(unit: .word)

let str = “困死啦睡觉去了”

let strRange = str.startIndex ..< str.endIndex

tokenizer.string = str

let tokenArray = tokenizer.tokens(for: strRange)
```

困　死　啦　睡觉　去　了

Tokenization

```
import NaturalLanguage

let tokenizer = NLTokenizer(unit: .word)

let str = “困死啦睡觉去了”

let strRange = str.startIndex ..< str.endIndex

tokenizer.string = str

let tokenArray = tokenizer.tokens(for: strRange)
```

困　死　啦　睡觉　去　了

9:41 AM 100%

< John M

Design is coming together but needs some final tweaks

困死啦睡觉去了

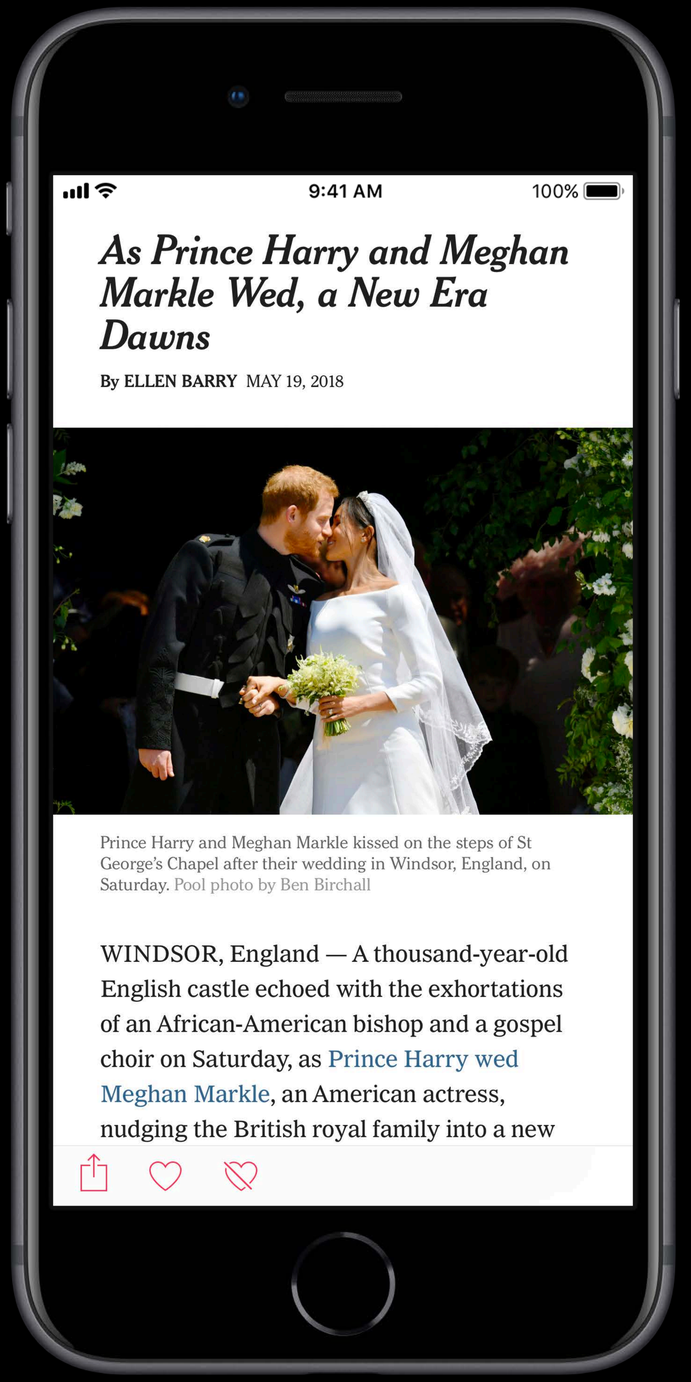
9:41 AM 100%

As Prince Harry and Meghan Markle Wed, a New Era Dawns

By ELLEN BARRY MAY 19, 2018

Prince Harry and Meghan Markle kissed on the steps of St George's Chapel after their wedding in Windsor, England, on Saturday. Pool photo by Ben Birchall

WINDSOR, England — A thousand-year-old English castle echoed with the exhortations of an African-American bishop and a gospel choir on Saturday, as Prince Harry wed Meghan Markle, an American actress, nudging the British royal family into a new



Harry

Harry Potter

Harry Styles

Harry Connick Jr.

Named Entity Recognition

```
import NaturalLanguage

let tagger = NLTagger(tagSchemes: [.nameType])

let str = “Prince Harry and Meghan Markle had their wedding ceremony in Windsor”

let strRange =  str.startIndex ..< str.endIndex

tagger.string = str

tagger.setLanguage(.english, range: strRange)

let tags = tagger.tags(in: strRange, unit: .word,

            scheme: .nameType, options: .omitWhitespace)
```

Named Entity Recognition

```
import NaturalLanguage

let tagger = NLTagger(tagSchemes: [.nameType])

let str = “Prince Harry and Meghan Markle had their wedding ceremony in Windsor”

let strRange =  str.startIndex ..< str.endIndex

tagger.string = str

tagger.setLanguage(.english, range: strRange)

let tags = tagger.tags(in: strRange, unit: .word,

            scheme: .nameType, options: .omitWhitespace)
```

Named Entity Recognition

```
import NaturalLanguage

let tagger = NLTagger(tagSchemes: [.nameType])

let str = “Prince Harry and Meghan Markle had their wedding ceremony in Windsor”

let strRange =  str.startIndex ..< str.endIndex

tagger.string = str

tagger.setLanguage(.english, range: strRange)

let tags = tagger.tags(in: strRange, unit: .word,

            scheme: .nameType, options: .omitWhitespace)
```

Named Entity Recognition

```
import NaturalLanguage

let tagger = NLTagger(tagSchemes: [.nameType])

let str = “Prince Harry and Meghan Markle had their wedding ceremony in Windsor”

let strRange =  str.startIndex ..< str.endIndex

tagger.string = str

tagger.setLanguage(.english, range: strRange)

let tags = tagger.tags(in: strRange, unit: .word,

            scheme: .nameType, options: .omitWhitespace)
```

Named Entity Recognition

```
import NaturalLanguage

let tagger = NLTagger(tagSchemes: [.nameType])

let str = “Prince Harry and Meghan Markle had their wedding ceremony in Windsor”

let strRange =  str.startIndex ..< str.endIndex

tagger.string = str

tagger.setLanguage(.english, range: strRange)

let tags = tagger.tags(in: strRange, unit: .word,

            scheme: .nameType, options: .omitWhitespace)
```

Named Entity Recognition

```
import NaturalLanguage

let tagger = NLTagger(tagSchemes: [.nameType])

let str = “Prince Harry and Meghan Markle had their wedding ceremony in Windsor”

let strRange =  str.startIndex ..< str.endIndex

tagger.string = str

tagger.setLanguage(.english, range: strRange)

let tags = tagger.tags(in: strRange, unit: .word,

            scheme: .nameType, options: .omitWhitespace)
```

PER: Prince Harry
Meghan Markle

LOC: Windsor

Natural Language

Developer documentation

Future support for new NLP APIs

Natural
Language Input

Intelligence

Swift APIs for NLP

Swift APIs for NLP

Custom NLP models

Custom NLP Models

Doug Davidson, Senior Software Engineer

Concept Learning

Concept Learning

Examples

Concept Learning

Examples

Concept Learning

Examples →  → Classify and Understand

Machine Learning

Training Data →  → Classify and Understand

Types of Custom Model

Types of Custom Model

Text Classification

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Label

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Label

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Label

Types of Custom Model

Text Classification

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Label

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Label

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Label

Word Tagging

Lorem	ipsum	dolor	sit	amet
Label	**Label**	**Label**	**Label**	**Label**

Text Classification

Sentiment classification

I am really excited, would definitely recommend it highly!	POSITIVE
This was terrible, much worse than I expected.	NEGATIVE
It was OK, something I could live with for now.	NEUTRAL

Text Classification

Topic classification

Foles’ late TD pass leads Eagles to 1st Super Bowl title	SPORTS
Apple introduces new 9.7 inch iPad with Apple Pencil support	TECHNOLOGY
Apple Reports Second Quarter Results	FINANCE

Text Classification

Domain classification

I’m looking for a place to stay in Barcelona.	HOTELS
Where can I get good Mexican food on a Sunday?	RESTAURANTS
Find me an inexpensive round-trip flight to London.	FLIGHTS

Word Tagging

Part of speech

Το	παιδί	διάβασε	το	βιβλίο
DETERMINER	NOUN	VERB	DETERMINER	NOUN

Word Tagging

Named entity

The	iPad	is	popular	in	Singapore
NONE	PRODUCT	NONE	NONE	NONE	LOCATION

Word Tagging

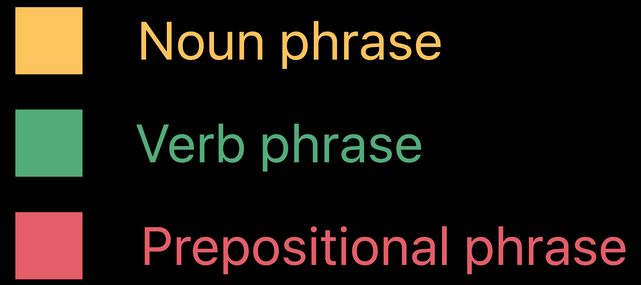
Slot parsing

Round trip	fares	from	Pittsburgh	to	Philadelphia
ROUND_TRIP	NONE	NONE	FROM_LOCATION	NONE	TO_LOCATION

Word Tagging

Chunking

Tim Cook	is	the CEO	of	Apple Inc
NP NP	VP	NP NP	PP	NP NP

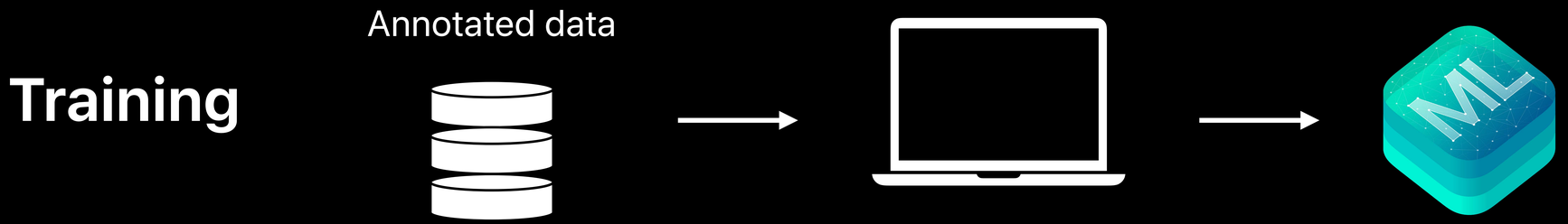


Supervised Machine Learning

Training and inference

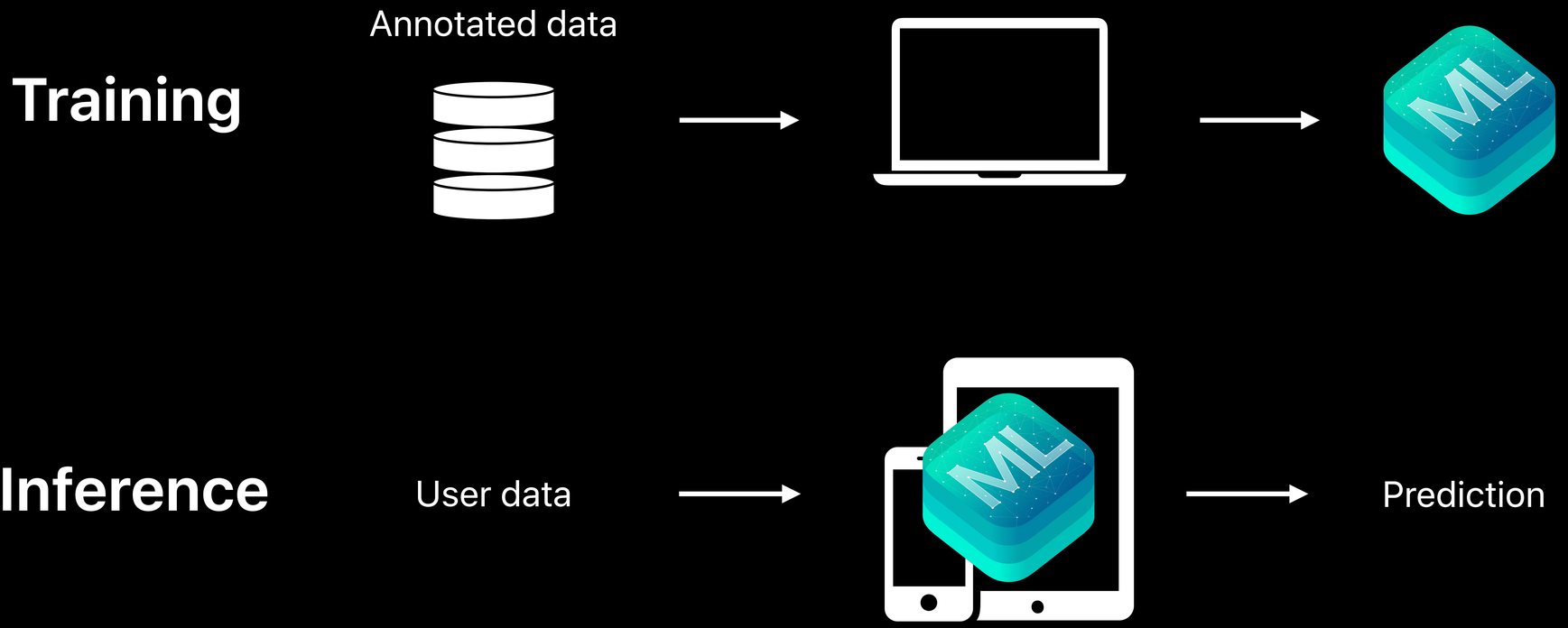
Supervised Machine Learning

Training and inference



Supervised Machine Learning

Training and inference



Custom NLP Models

Training

Custom NLP Models

Training

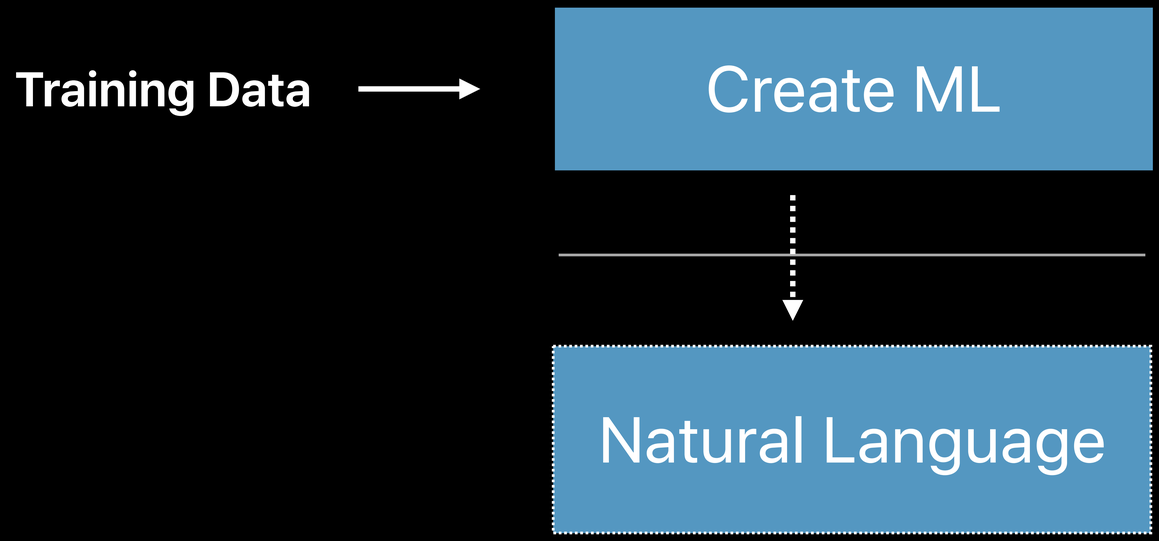
Training Data

Custom NLP Models

Training

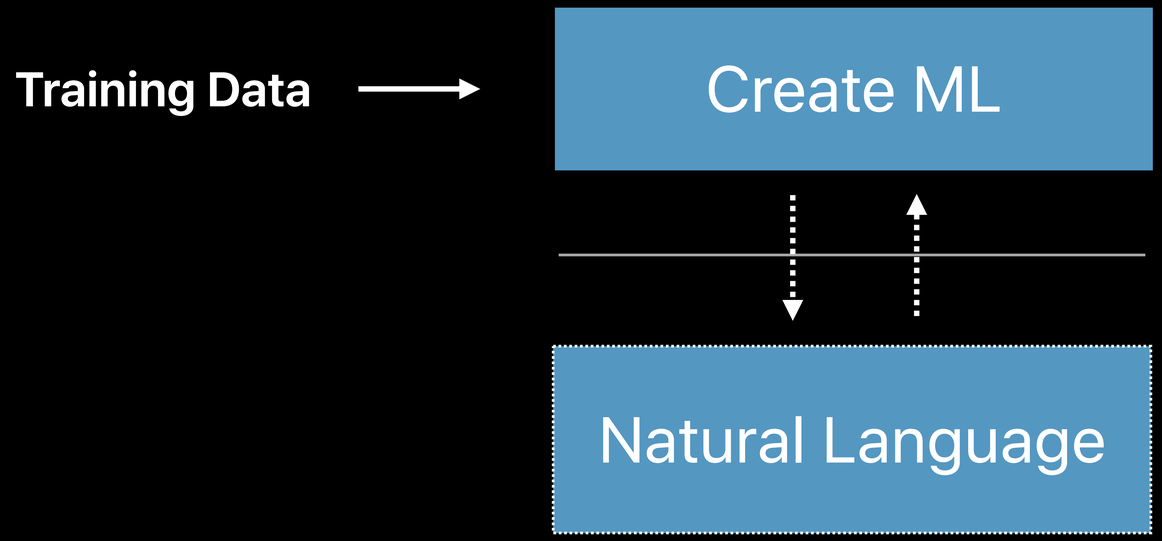
Custom NLP Models

Training



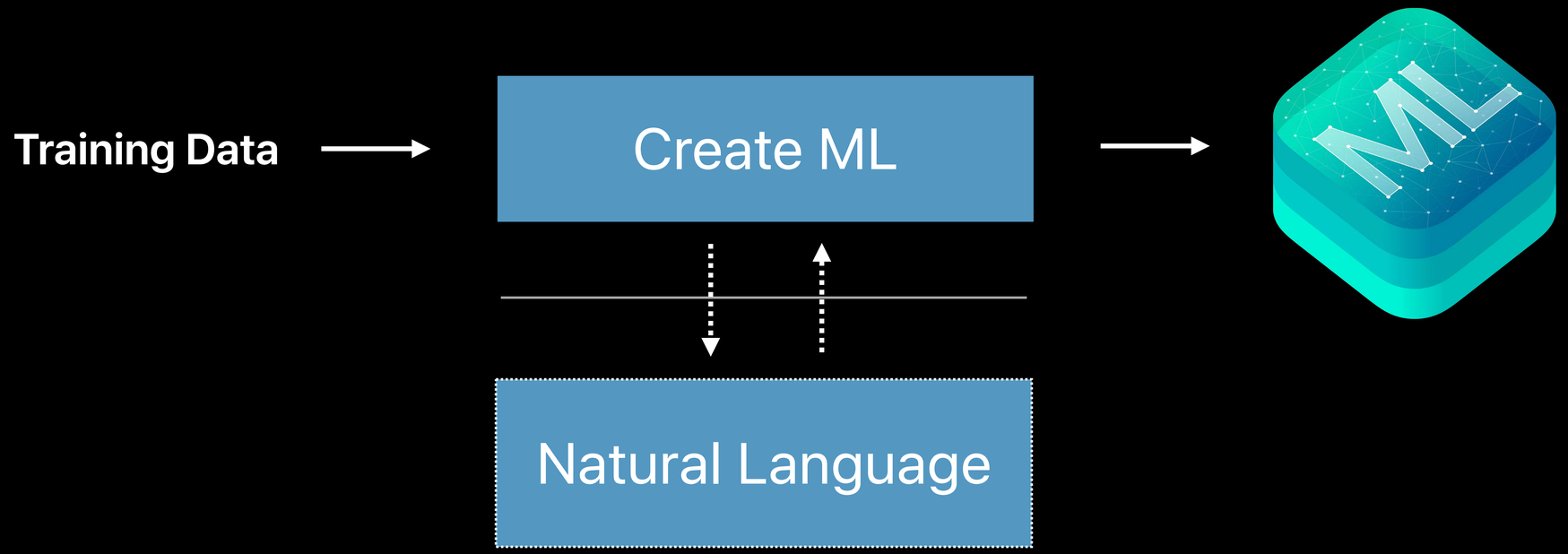
Custom NLP Models

Training



Custom NLP Models

Training



Text Classifier

Annotated data

```
[
  {
    “text”: “I am really excited, would definitely recommend it highly!”,
    “label”: “Positive”
  },
  {
    “text”: “It was OK, something I could live with for now.“,
    “label”: “Neutral”
  },
  {
    “text”: “This was terrible, much worse than I expected.“,
    “label”: “Negative”
  },
 ]
```

Text Classifier

Annotated data

```
[
  {
    "text": "I am really excited, would definitely recommend it highly!",
    "label": "Positive"
  },
  {
    "text": "It was OK, something I could live with for now.",
    "label": "Neutral"
  },
  {
    "text": "This was terrible, much worse than I expected.",
    "label": "Negative"
  },
]
```

Text Classifier

Training

```
import CreateML

import Foundation

let trainingData = try MLDataTable(contentsOfFile: Bundle.main.url(forResource: “news”,

                                   withExtension: “json”)!)

let model = try MLTextClassifier(trainingWith: trainingData,

                                 textColumn: “text”, labelColumn: “label”)

try model.write(to: URL(fileURLWithPath: “/Users/me/Desktop/textclassifier.mlmodel”)
```

Text Classifier

Training

```
import CreateML

import Foundation

let trainingData = try MLDataTable(contentsOfFile: Bundle.main.url(forResource: “news”,

                                   withExtension: “json”)!)

let model = try MLTextClassifier(trainingWith: trainingData,

                                 textColumn: “text”, labelColumn: “label”)

try model.write(to: URL(fileURLWithPath: “/Users/me/Desktop/textclassifier.mlmodel”)
```

Text Classifier

Training

```
import CreateML

import Foundation

let trainingData = try MLDataTable(contentsOfFile: Bundle.main.url(forResource: “news”,

                                   withExtension: “json”)!)

let model = try MLTextClassifier(trainingWith: trainingData,

                                 textColumn: “text”, labelColumn: “label”)

try model.write(to: URL(fileURLWithPath: “/Users/me/Desktop/textclassifier.mlmodel”)
```

Text Classifier

Training

```
import CreateML

import Foundation

let trainingData = try MLDataTable(contentsOfFile: Bundle.main.url(forResource: “news”,

                                   withExtension: “json”)!)

let model = try MLTextClassifier(trainingWith: trainingData,

                                 textColumn: “text”, labelColumn: “label”)

try model.write(to: URL(fileURLWithPath: “/Users/me/Desktop/textclassifier.mlmodel”)
```

Word Tagger

Annotated data

```
[
  {
    "tokens": ["AirPods", "are", "a", "fantastic", "product", "from", "Apple", "."],
    "labels":  ["PROD", "NONE", "NONE", "NONE", "NONE", "NONE", "ORG", "NONE"]
    },
  {
    "tokens": ["Apple", "and", "Tim", "Cook", "have", "another", "hit", "."],
    "labels":  ["ORG", "NONE", "PER", "PER", "NONE", "NONE", "NONE", "NONE"]
    }
 ]
```

Word Tagger

Annotated data

```
[
  {
    "tokens": ["AirPods", "are", "a", "fantastic", "product", "from", "Apple", "."],
    "labels":  ["PROD", "NONE", "NONE", "NONE", "NONE", "NONE", "ORG", "NONE"]
    },
  {
    "tokens": ["Apple", "and", "Tim", "Cook", "have", "another", "hit", "."],
    "labels":  ["ORG", "NONE", "PER", "PER", "NONE", "NONE", "NONE", "NONE"]
    }
 ]
```

Word Tagger

Training

```
import CreateML

import Foundation

let trainingData = try MLDataTable(contentsOfFile:

                    Bundle.main.url(forResource: “products”, withExtension: “json”)!)

let model = try MLWordTagger(trainingWith: trainingData,

                                tokenHeader: “tokens”, labelColumn: “labels”)

try model.write(to: URL(fileURLWithPath: “/Users/me/Desktop/wordtagger.mlmodel”)
```

Introducing Create ML | Hall 2 | Tuesday 2:00PM

Word Tagger

Training

```
import CreateML

import Foundation

let trainingData = try MLDataTable(contentsOfFile:

                    Bundle.main.url(forResource: “products”, withExtension: “json”)!)

let model = try MLWordTagger(trainingWith: trainingData,

                                tokenHeader: “tokens”, labelColumn: “labels”)

try model.write(to: URL(fileURLWithPath: “/Users/me/Desktop/wordtagger.mlmodel”)
```

Introducing Create ML Hall 2 Tuesday 2:00PM

Word Tagger

Training

```
import CreateML

import Foundation

let trainingData = try MLDataTable(contentsOfFile:

                    Bundle.main.url(forResource: “products”, withExtension: “json”)!)

let model = try MLWordTagger(trainingWith: trainingData,

                                tokenHeader: “tokens”, labelColumn: “labels”)

try model.write(to: URL(fileURLWithPath: “/Users/me/Desktop/wordtagger.mlmodel”)
```

Introducing Create ML | Hall 2 | Tuesday 2:00PM

Word Tagger

Training

```
import CreateML

import Foundation

let trainingData = try MLDataTable(contentsOfFile:

                    Bundle.main.url(forResource: “products”, withExtension: “json”)!)

let model = try MLWordTagger(trainingWith: trainingData,

                                tokenHeader: “tokens”, labelColumn: “labels”)

try model.write(to: URL(fileURLWithPath: “/Users/me/Desktop/wordtagger.mlmodel”)
```

Introducing Create ML | Hall 2 | Tuesday 2:00PM

Natural Language

Inference

What’s New in Core ML, Part 1	Hall 1	Wednesday 9:00AM
What’s New in Core ML, Part 2	Hall 1	Wednesday 10:00AM

Natural Language

Inference

User data

What’s New in Core ML, Part 1	Hall 1	Wednesday 9:00AM
What’s New in Core ML, Part 2	Hall 1	Wednesday 10:00AM

Natural Language

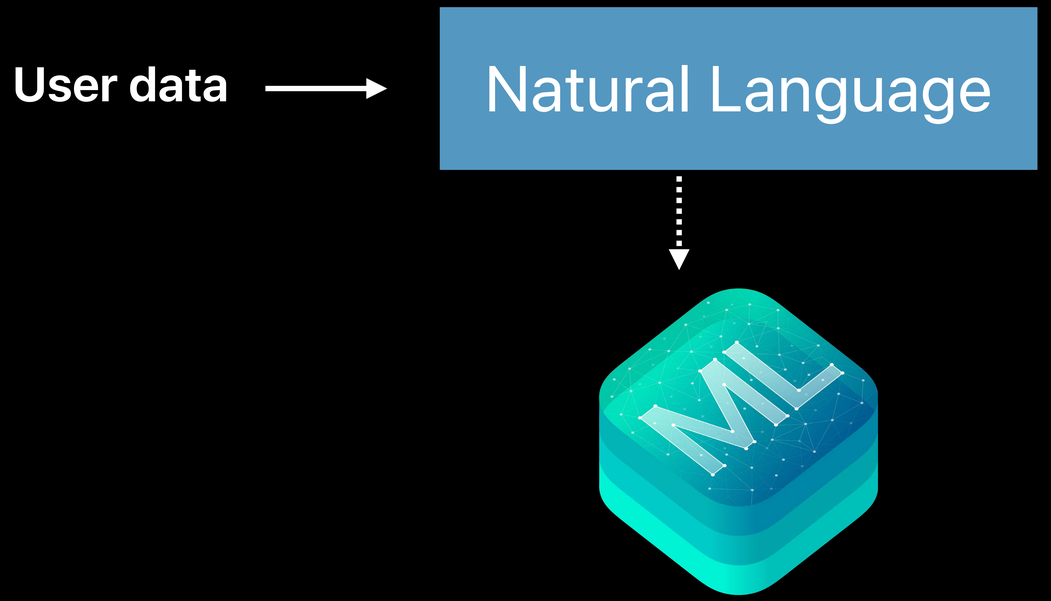
Inference

User data → Natural Language

What’s New in Core ML, Part 1	Hall 1	Wednesday 9:00AM
What’s New in Core ML, Part 2	Hall 1	Wednesday 10:00AM

Natural Language

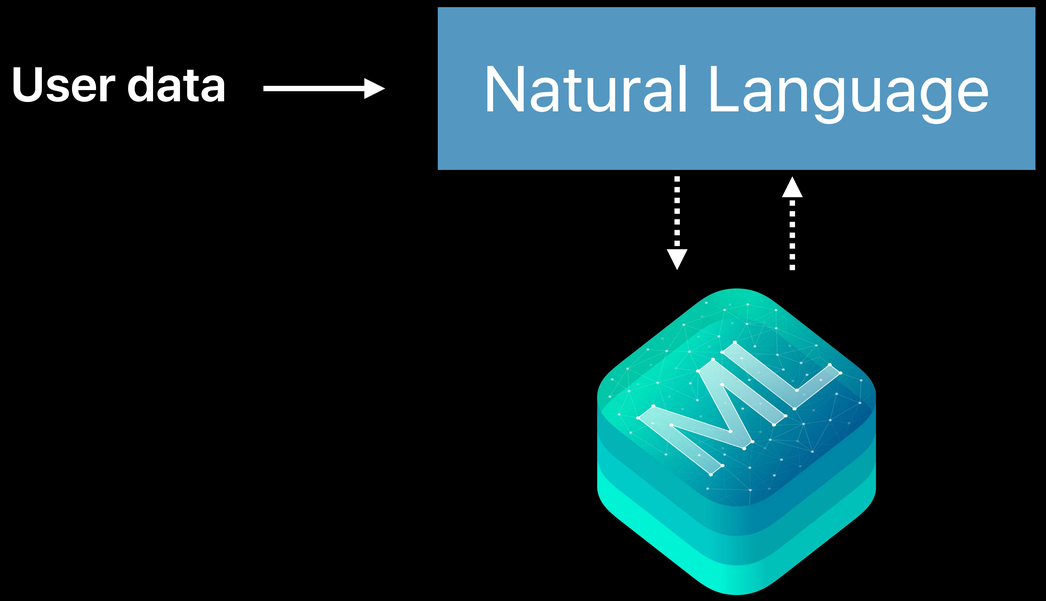
Inference



What’s New in Core ML, Part 1	Hall 1	Wednesday 9:00AM
What’s New in Core ML, Part 2	Hall 1	Wednesday 10:00AM

Natural Language

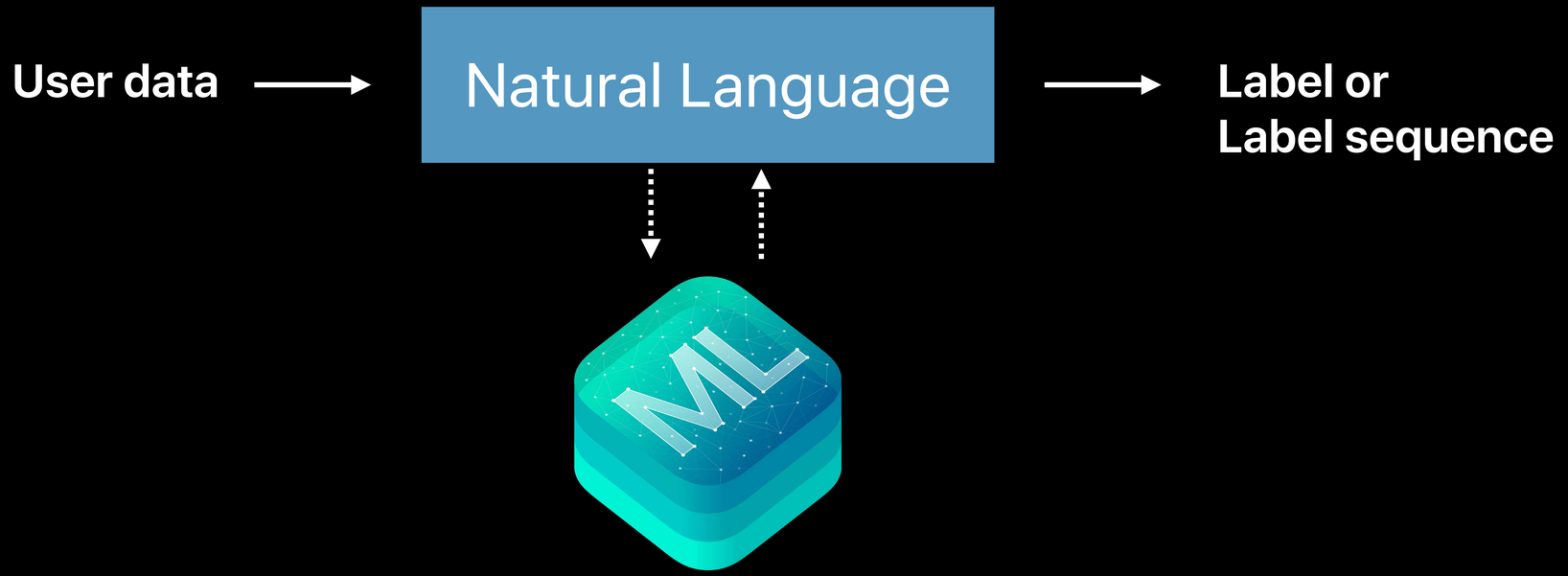
Inference



What’s New in Core ML, Part 1	Hall 1	Wednesday 9:00AM
What’s New in Core ML, Part 2	Hall 1	Wednesday 10:00AM

Natural Language

Inference



What’s New in Core ML, Part 1	Hall 1	Wednesday 9:00AM
What’s New in Core ML, Part 2	Hall 1	Wednesday 10:00AM

Using a Custom Model

```
import NaturalLanguage

if let modelURL = Bundle.main.url(forResource: “classifier”,
                                  withExtension: “mlmodelc”) {
        let model = try NLModel(contentsOf: modelURL)
        let label = model.predictedLabel(for: “I really loved it!”)
}
```

Using a Custom Model

```
import NaturalLanguage

if let modelURL = Bundle.main.url(forResource: “classifier”,
                                  withExtension: “mlmodelc”) {
        let model = try NLModel(contentsOf: modelURL)
        let label = model.predictedLabel(for: “I really loved it!”)
}
```

Using a Custom Model

```
import NaturalLanguage

if let modelURL = Bundle.main.url(forResource: “classifier”,
                                  withExtension: “mlmodelc”) {
        let model = try NLModel(contentsOf: modelURL)
        let label = model.predictedLabel(for: “I really loved it!”)
}
```

Using a Custom Model

```
import NaturalLanguage

if let modelURL = Bundle.main.url(forResource: “classifier”,
                                  withExtension: “mlmodelc”) {
        let model = try NLModel(contentsOf: modelURL)
        let label = model.predictedLabel(for: “I really loved it!”)
}
```

Using Custom Models with NLTagger

```
let myTagScheme = NLTagScheme(“MyTagScheme”)

let tagger = NLTagger(tagSchemes: [myTagScheme])

tagger.setModels([model], forTagScheme: myTagScheme)

tagger.string = myString

tagger.enumerateTags(in: range, unit: .sentence,

                     scheme: myTagScheme, options: []) { (tag, tokenRange) -> Bool in

   // use the generated tags
   return true

 }
```

Using Custom Models with NLTagger

```
let myTagScheme = NLTagScheme(“MyTagScheme”)

let tagger = NLTagger(tagSchemes: [myTagScheme])

tagger.setModels([model], forTagScheme: myTagScheme)

tagger.string = myString

tagger.enumerateTags(in: range, unit: .sentence,

                     scheme: myTagScheme, options: []) { (tag, tokenRange) -> Bool in

   // use the generated tags
   return true

 }
```

Using Custom Models with NLTagger

```
let myTagScheme = NLTagScheme(“MyTagScheme”)

let tagger = NLTagger(tagSchemes: [myTagScheme])

tagger.setModels([model], forTagScheme: myTagScheme)

tagger.string = myString

tagger.enumerateTags(in: range, unit: .sentence,

                     scheme: myTagScheme, options: []) { (tag, tokenRange) -> Bool in

   // use the generated tags
   return true

 }
```

Using Custom Models with NLTagger

```
let myTagScheme = NLTagScheme(“MyTagScheme”)

let tagger = NLTagger(tagSchemes: [myTagScheme])

tagger.setModels([model], forTagScheme: myTagScheme)

tagger.string = myString

tagger.enumerateTags(in: range, unit: .sentence,

                     scheme: myTagScheme, options: []) { (tag, tokenRange) -> Bool in

   // use the generated tags
   return true

 }
```

Using Custom Models with NLTagger

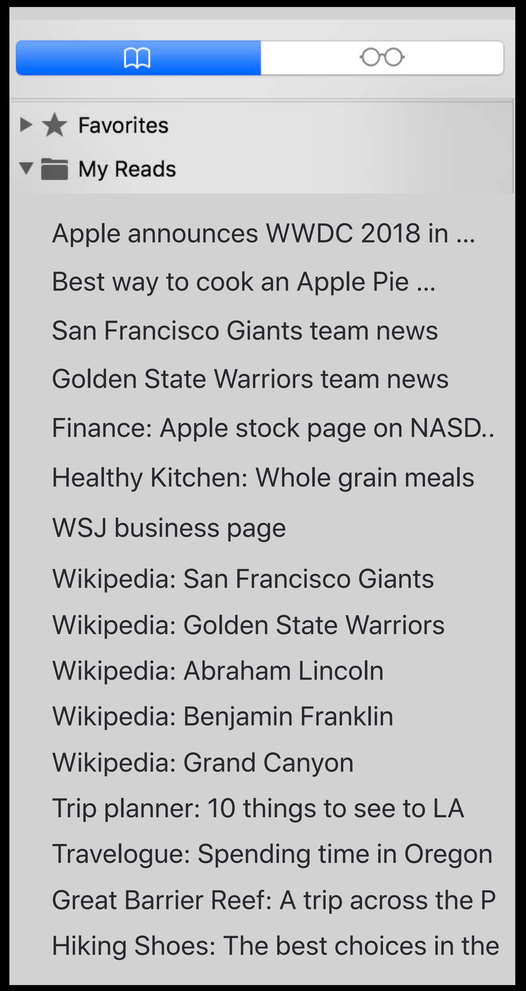
```
let myTagScheme = NLTagScheme(“MyTagScheme”)

let tagger = NLTagger(tagSchemes: [myTagScheme])

tagger.setModels([model], forTagScheme: myTagScheme)

tagger.string = myString

tagger.enumerateTags(in: range, unit: .sentence,

                     scheme: myTagScheme, options: []) { (tag, tokenRange) -> Bool in

   // use the generated tags
   return true

 }
```

Using Custom Models with NLTagger

```
let myTagScheme = NLTagScheme(“MyTagScheme”)

let tagger = NLTagger(tagSchemes: [myTagScheme])

tagger.setModels([model], forTagScheme: myTagScheme)

tagger.string = myString

tagger.enumerateTags(in: range, unit: .sentence,

                     scheme: myTagScheme, options: []) { (tag, tokenRange) -> Bool in

    // use the generated tags
    return true

 }
```

Wade

Bookmark organization



Organize by topics, entities

Demo

Swift APIs for NLP

Custom NLP models

Swift APIs for NLP

Custom NLP models

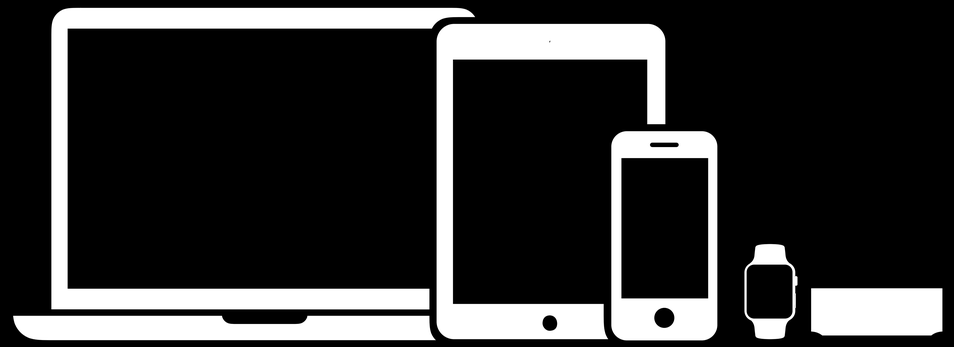
Performance

Natural Language

Available on all Apple platforms

Natural Language

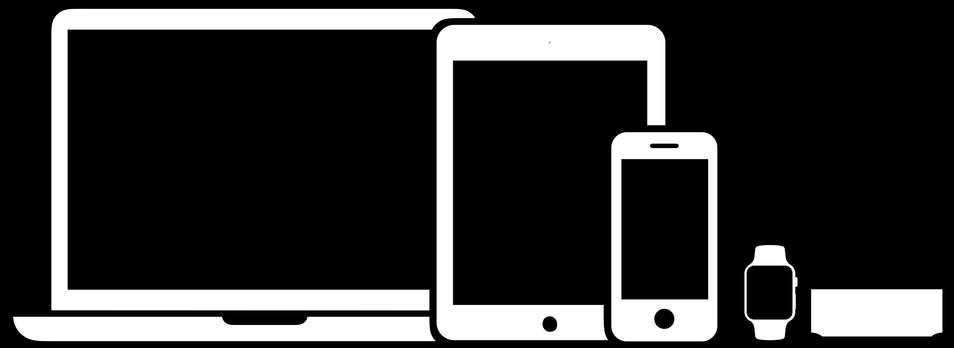
Available on all Apple platforms



Natural Language

Available on all Apple platforms

Standardized text processing



Text Processing

Typical ML training

Text Processing

Typical ML inference

User Data →  → Output

Text Processing

Typical ML inference

User Data →  → Output

Text Processing

Typical ML inference

Standarized Text Processing

Natural Language

Standarized Text Processing

Natural Language

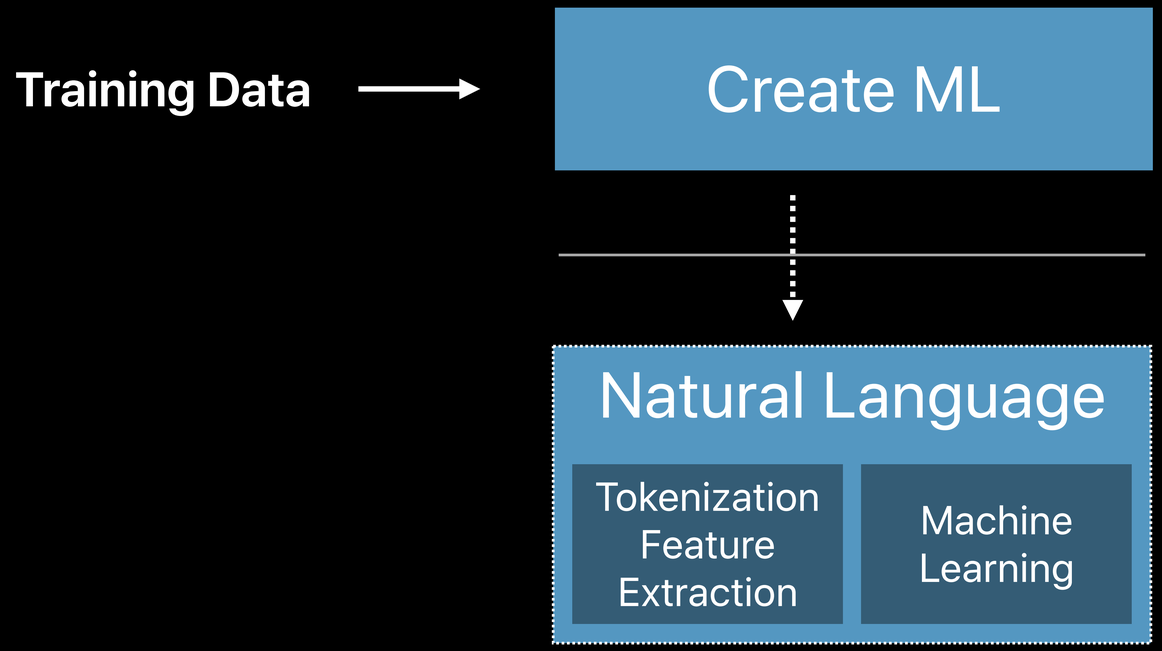
Training Data

Standarized Text Processing

Natural Language

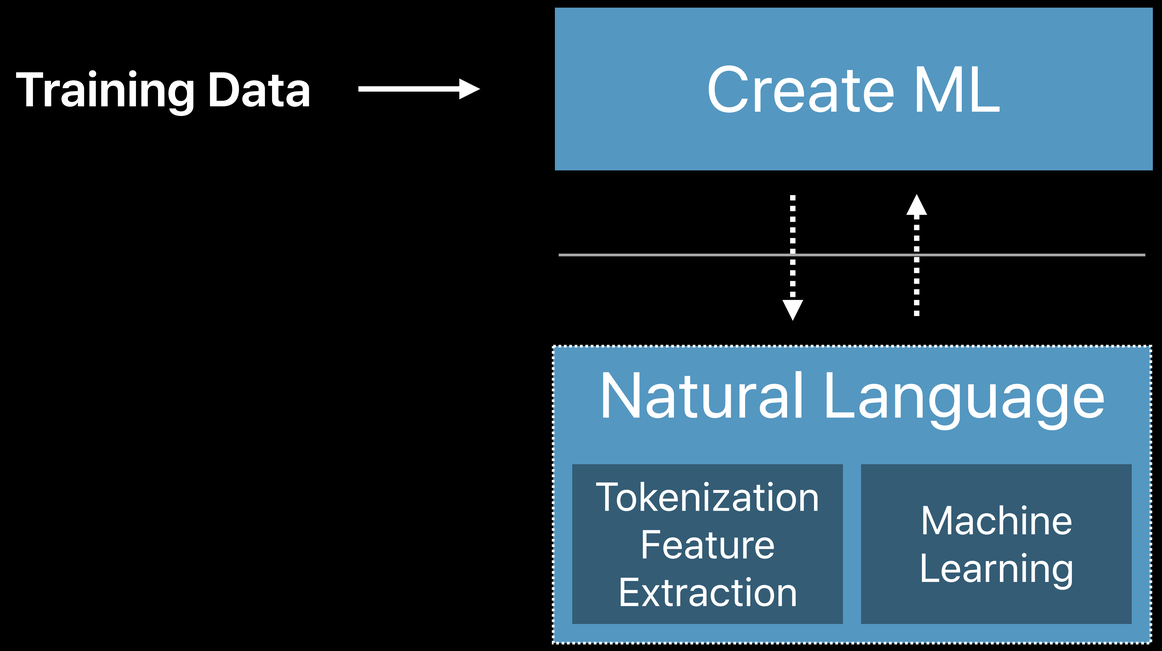
Standarized Text Processing

Natural Language



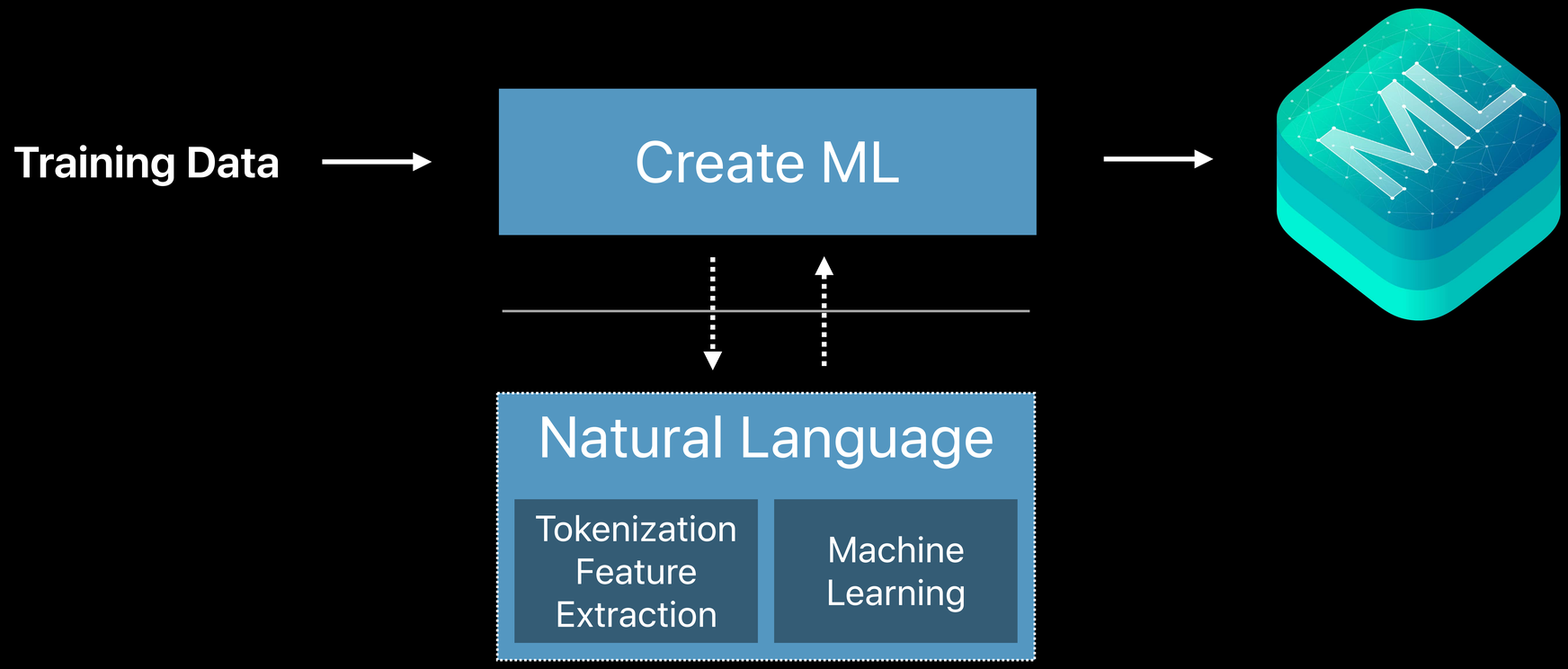
Standarized Text Processing

Natural Language



Standarized Text Processing

Natural Language



Standarized Text Processing

Natural Language

User Data →  → Output

Standarized Text Processing

Natural Language

User Data →  → Output

Standarized Text Processing

Natural Language

Optimized Models

Low latency

Optimized for Apple hardware

Optimized Models

Low latency

Optimized for Apple hardware

	Open Source CRFSuite	Natural Language
Named Entity Recognition	70 MB	1.4 MB
Chunking	30 MB	1.8 MB

Results on CoNLL training and test data

ML Algorithms

Text classification

```
let modelParameters = MLTextClassifier.ModelParameters(algorithm: .maxEnt(version: 1))

let modelParameters = MLTextClassifier.ModelParameters(algorithm: .CRF(version: 1))
```

ML Algorithms

Text classification

```
let modelParameters = MLTextClassifier.ModelParameters(algorithm: .maxEnt(version: 1))

let modelParameters = MLTextClassifier.ModelParameters(algorithm: .CRF(version: 1))
```

ML Algorithms

Text classification

```
let modelParameters = MLTextClassifier.ModelParameters(algorithm: .maxEnt(version: 1))

let modelParameters = MLTextClassifier.ModelParameters(algorithm: .CRF(version: 1))
```

Maximum Entropy

ML Algorithms

Text classification

```
let modelParameters = MLTextClassifier.ModelParameters(algorithm: .maxEnt(version: 1))

let modelParameters = MLTextClassifier.ModelParameters(algorithm: .CRF(version: 1))
```

Maximum Entropy

Conditional Random Field

ML Algorithms

Text classification

```
let modelParameters = MLTextClassifier.ModelParameters(algorithm: .maxEnt(version: 1))

let modelParameters = MLTextClassifier.ModelParameters(algorithm: .CRF(version: 1))
```

Maximum Entropy

Conditional Random Field

ML Algorithms

Word tagging

```
let model = try MLWordTagger(trainingWith: trainingData,
                             tokenHeader: “tokens”, labelColumn: “labels”)
```

Conditional Random Field

ML Algorithms

Word tagging

```
let model = try MLWordTagger(trainingWith: trainingData,
                             tokenHeader: “tokens”, labelColumn: “labels”)
```

Conditional Random Field

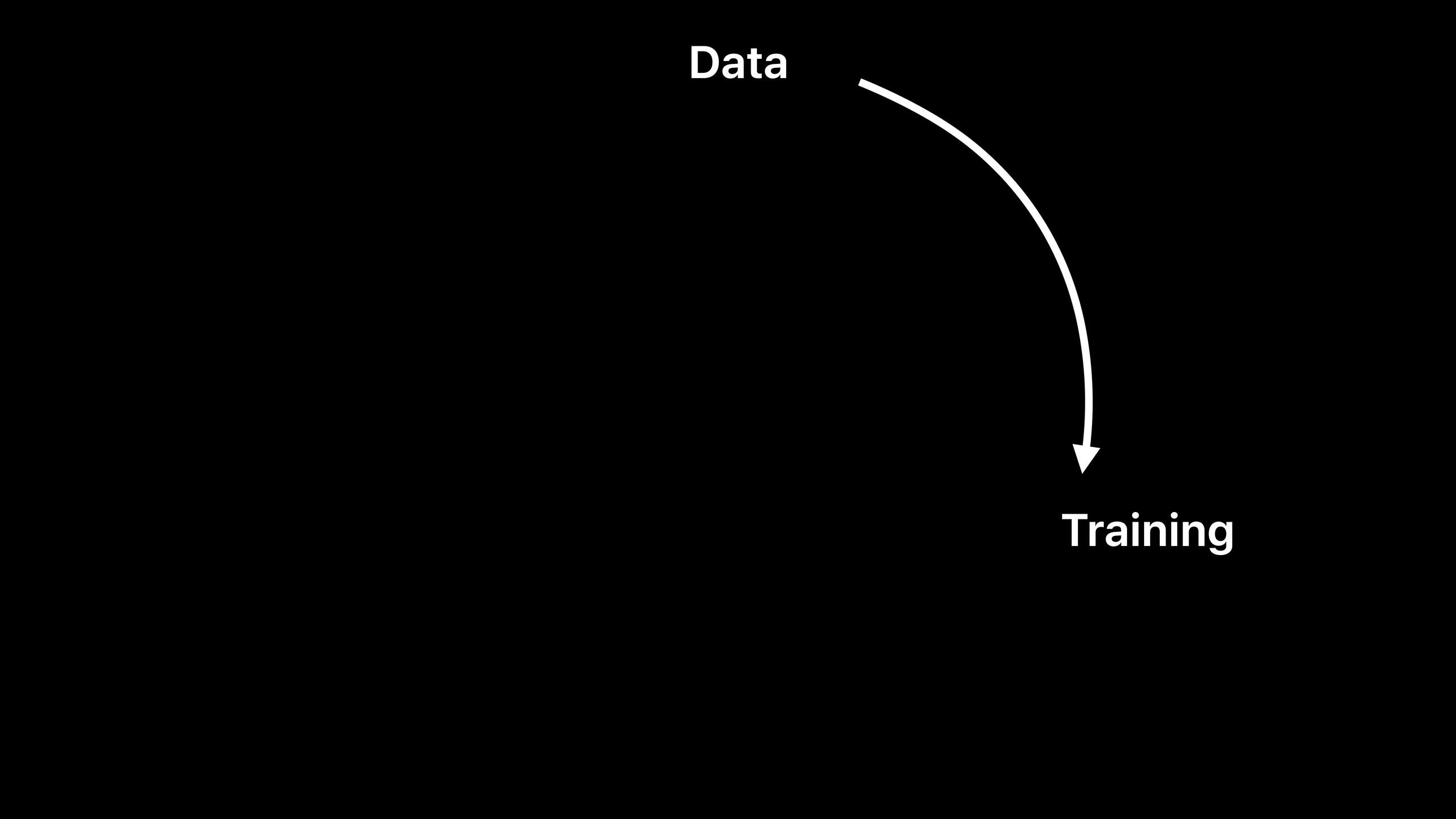
Data

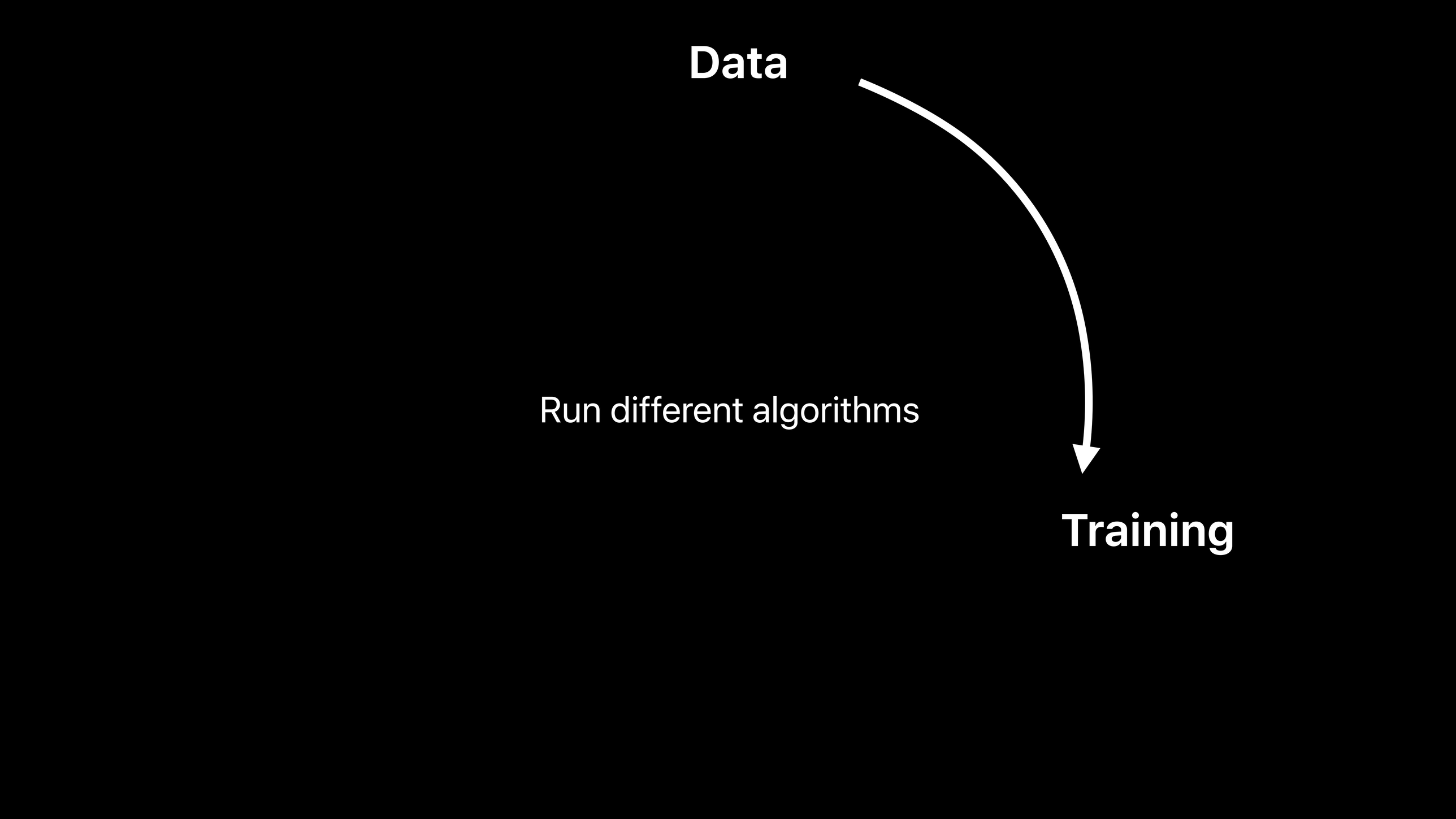
Data

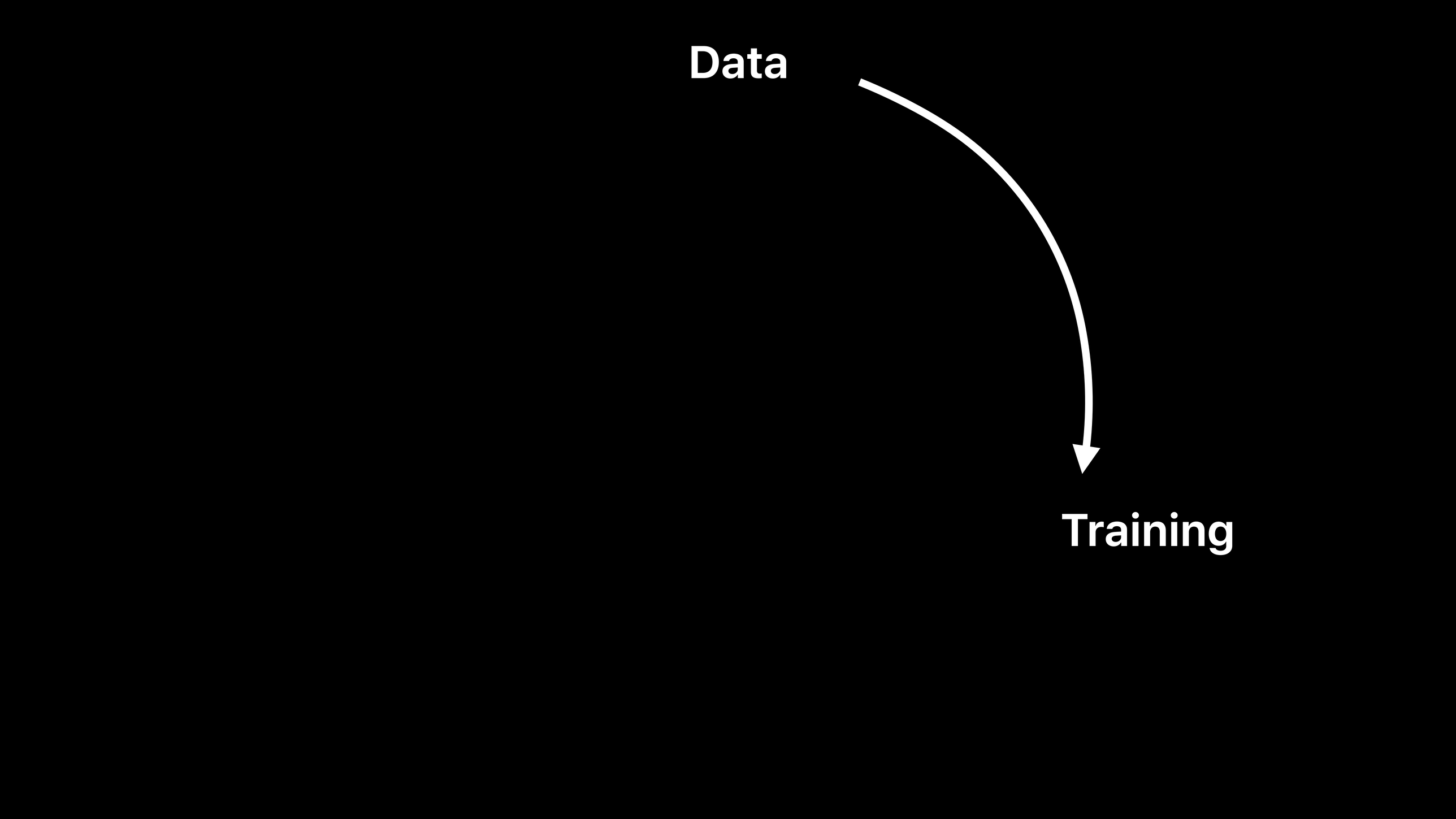
Validate training data

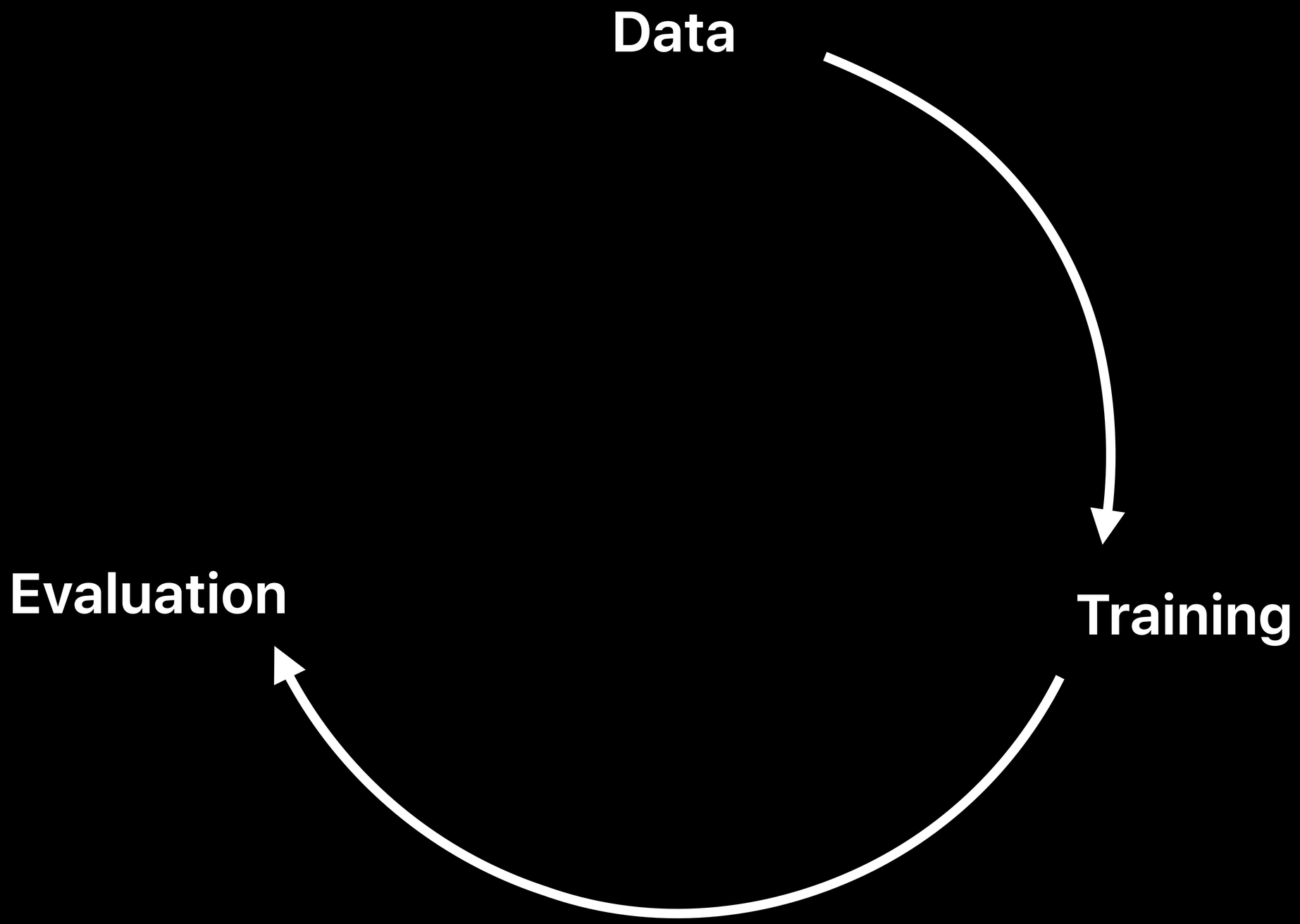
Inspect training instances per class

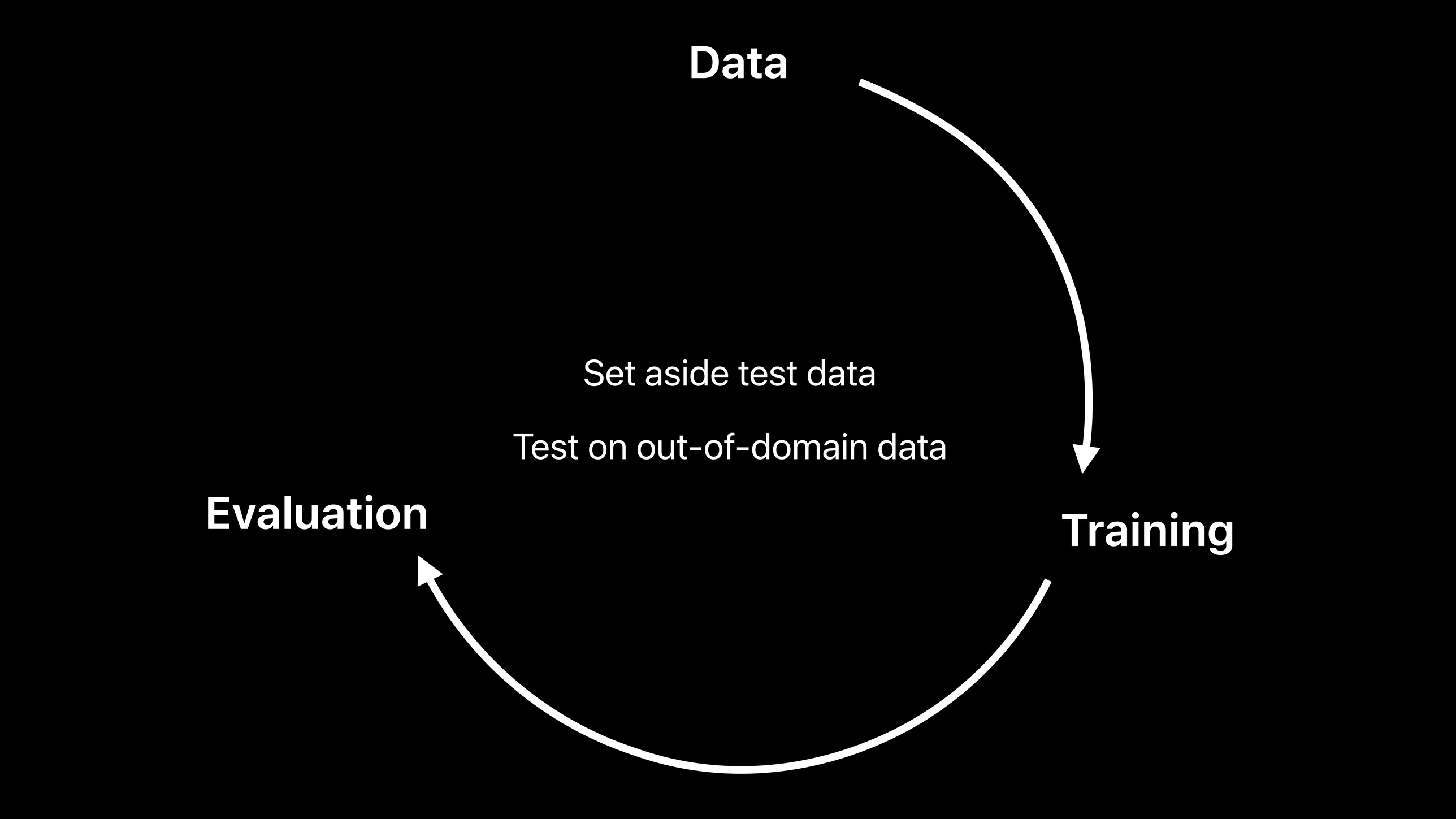
Data

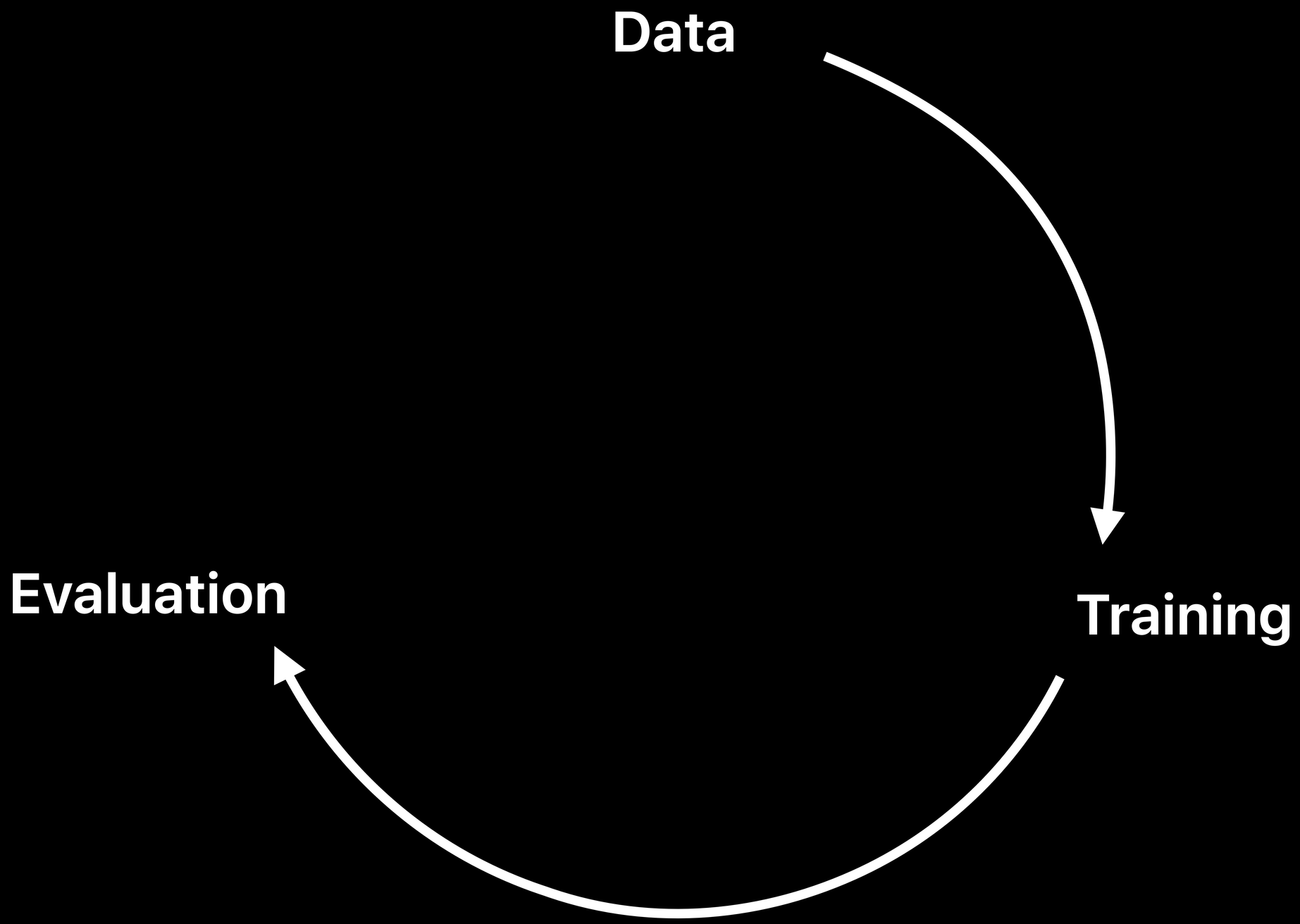

Data
Training

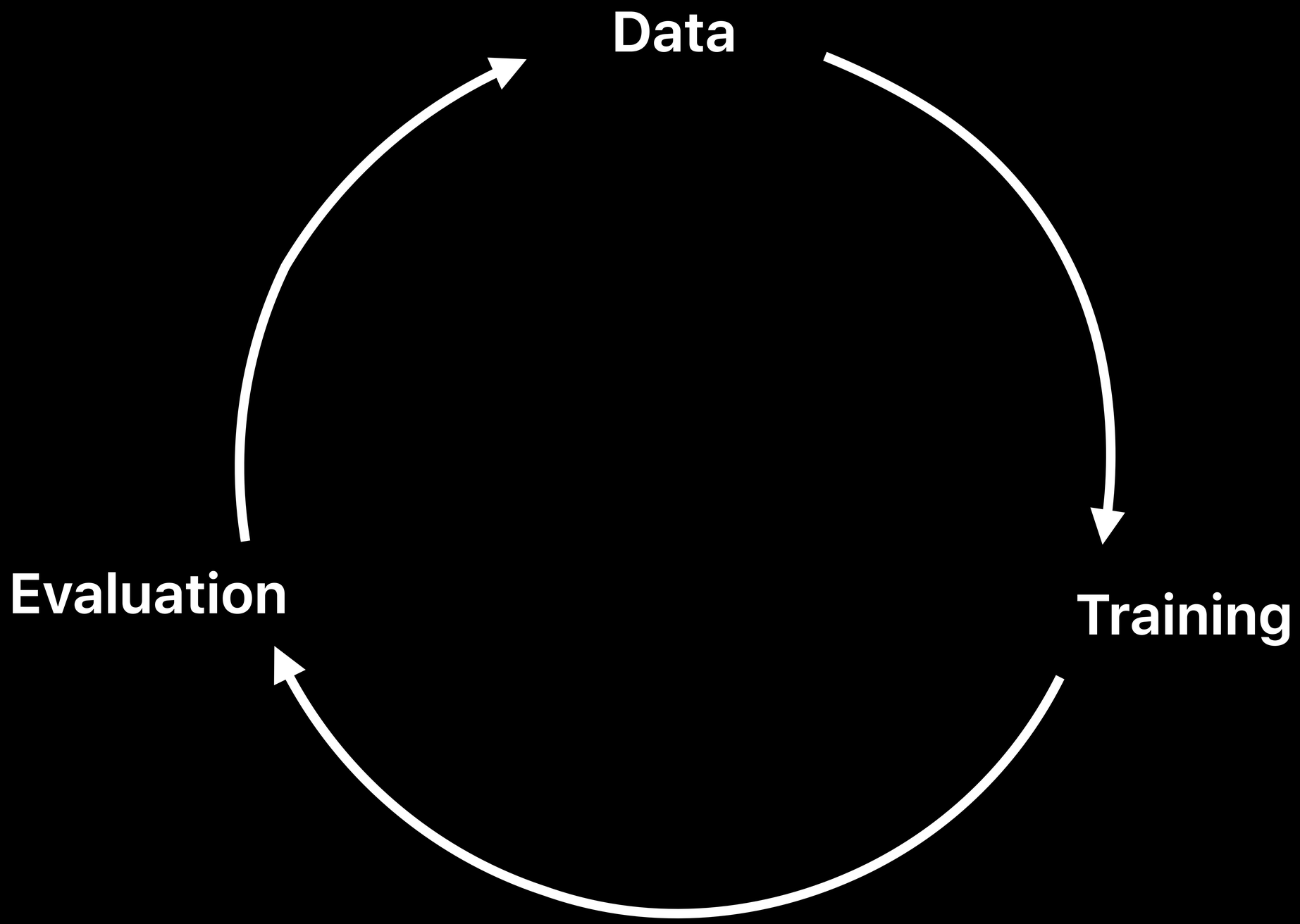

Data
Run different algorithms
Training

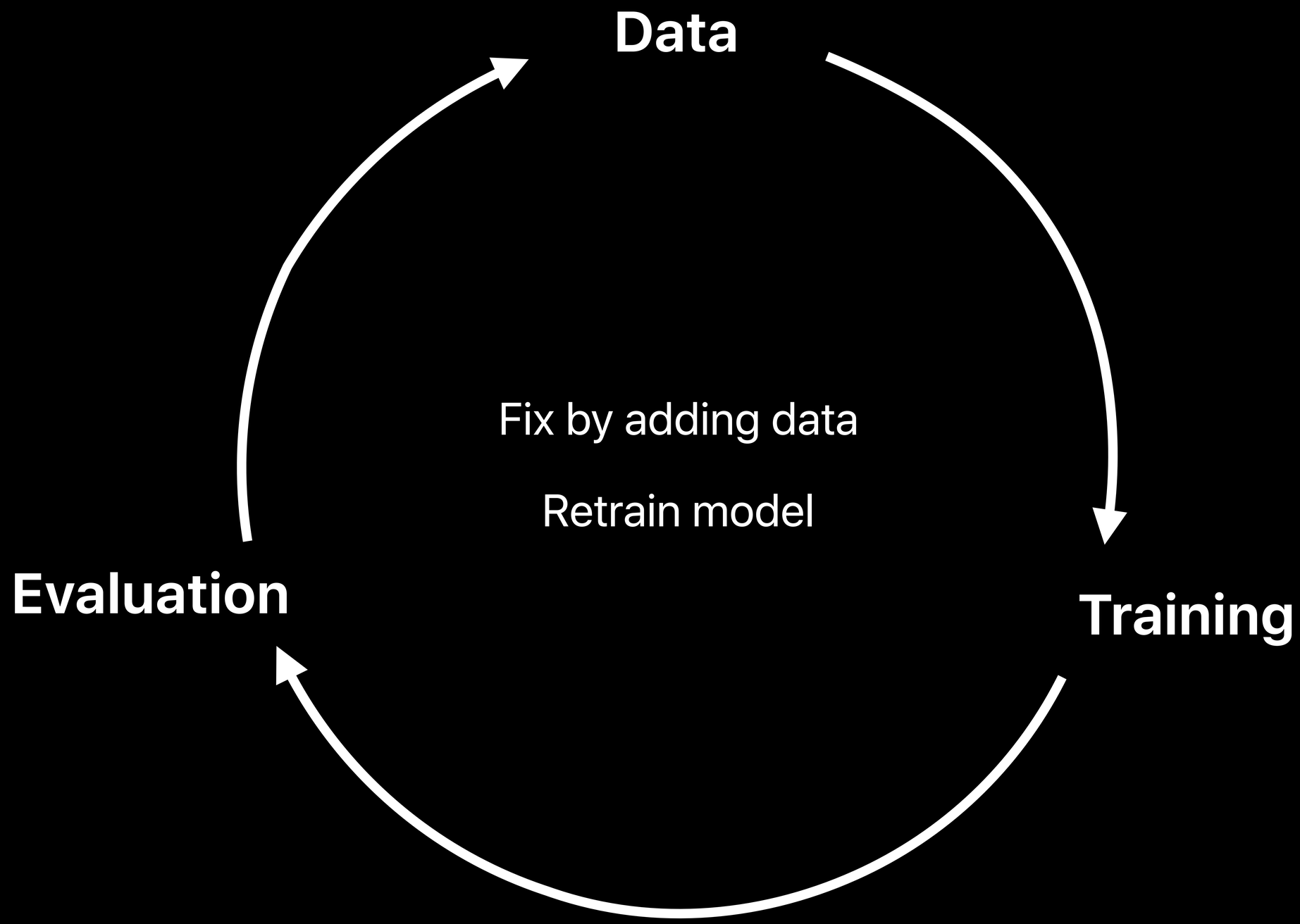

Data
Training

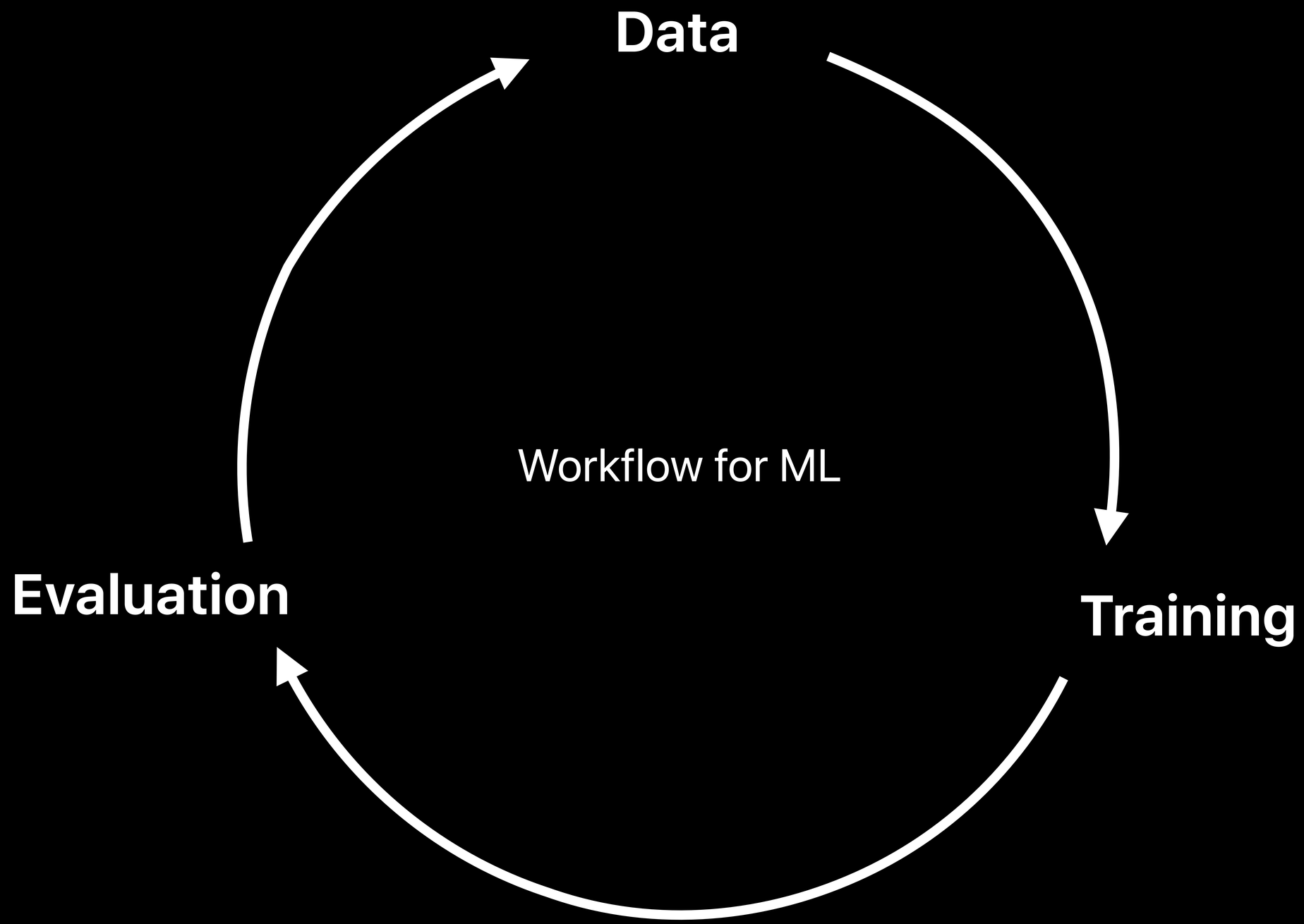

Data
Training
Evaluation


Data
Set aside test data
Test on out-of-domain data
Training
Evaluation


Data
Training
Evaluation


Data
Training
Evaluation


Data
Fix by adding data
Retrain model
Evaluation
Training


Data
Workflow for ML
Training
Evaluation

Swift APIs for NLP

Custom NLP models

Performance

Swift APIs for NLP

Custom NLP models

Performance

Privacy

Privacy Preserving ML Applied to NLP

Natural Language Framework

Summary

New: Natural Language Framework

Custom NLP models

Performance

Privacy

More Information

https://developer.apple.com/wwdc18/713

Machine Learning Get-Together	Café Lounge	Wednesday 6:15PM
Natural Language Lab	Technology Lab 4	Thursday 1:00PM


WWDC18