App Frameworks

#WWDC17

Natural Language Processing and your Apps

Session 208

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

Goal

Goal

Goal

Natural language input

typed text

recognized handwriting

transcribed speech

Goal

Natural language output

typed text

recognized handwriting

transcribed speech

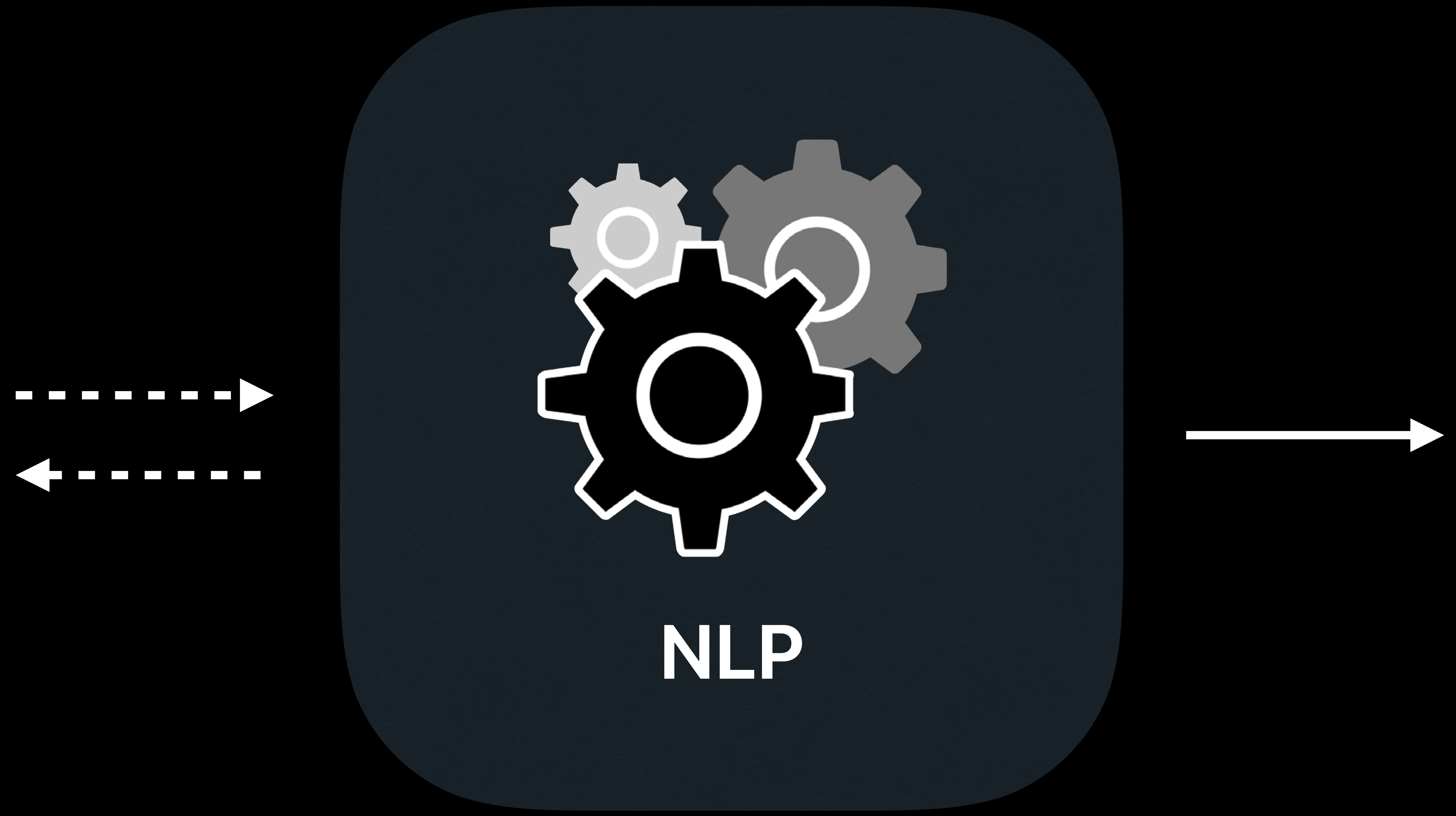
Goal

Natural language text

typed

recognized handwriting

transcribed speech

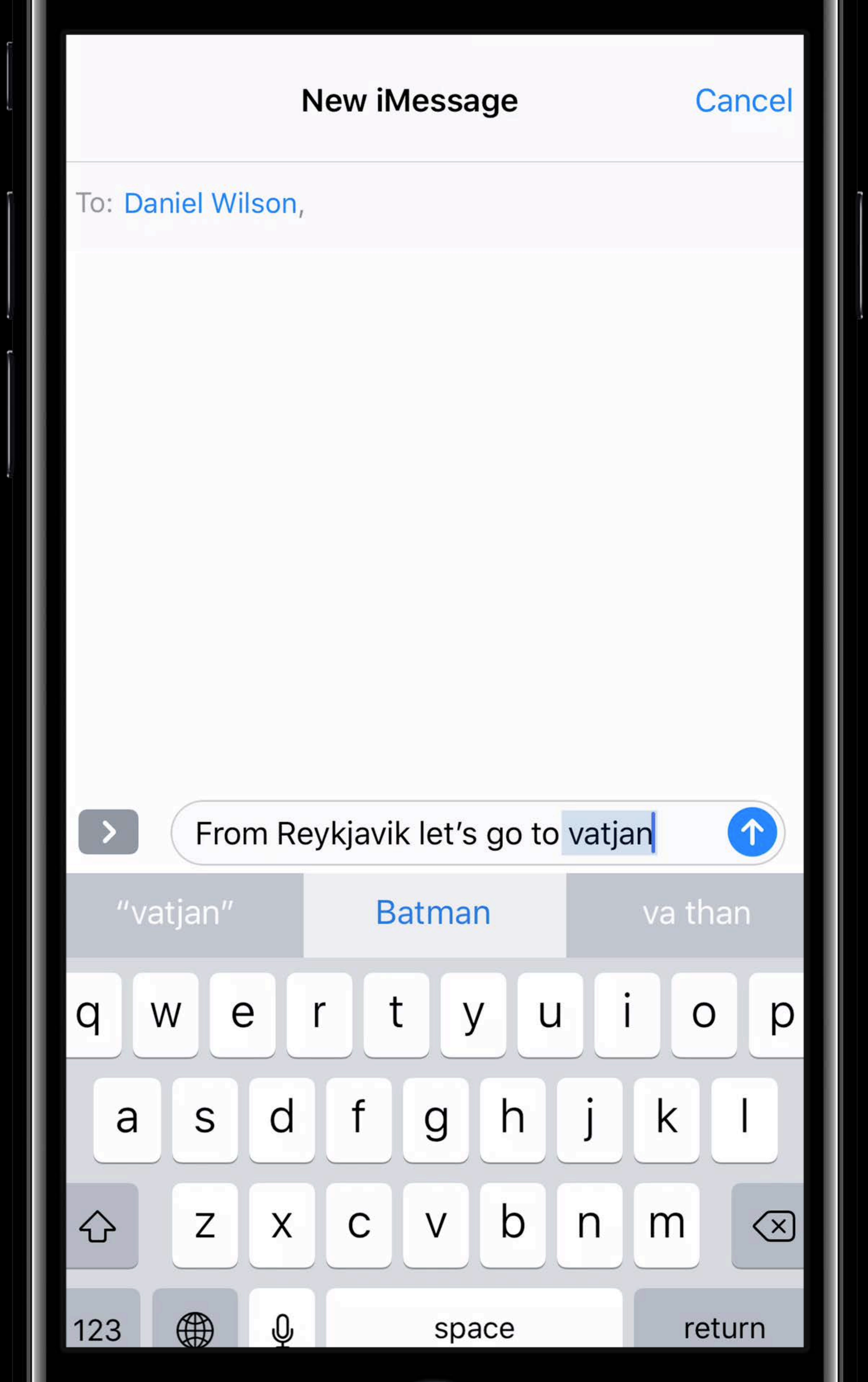


Confer intelligence

NLP APIs

NLP APIs

NLP APIs

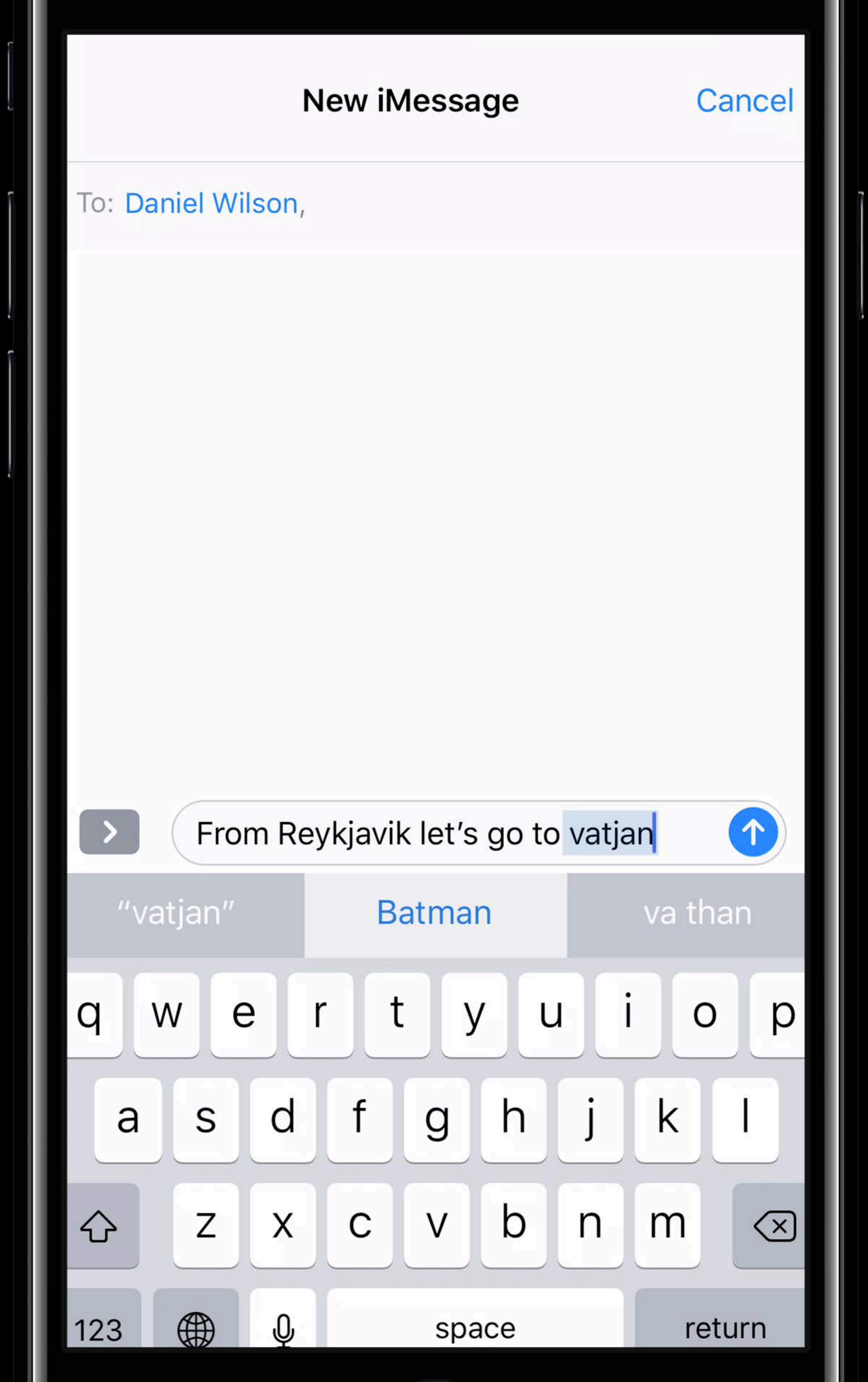

New iMessage

Cancel

To: Daniel Wilson,

From Reykjavik let's go to vatjan

NLP APIs


New iMessage

Cancel

To: Daniel Wilson,

From Reykjavik let's go to vatjan

NLP APIs

Natural Language Processing

Natural Language Processing

Natural language text

typed

recognized handwriting

transcribed speech

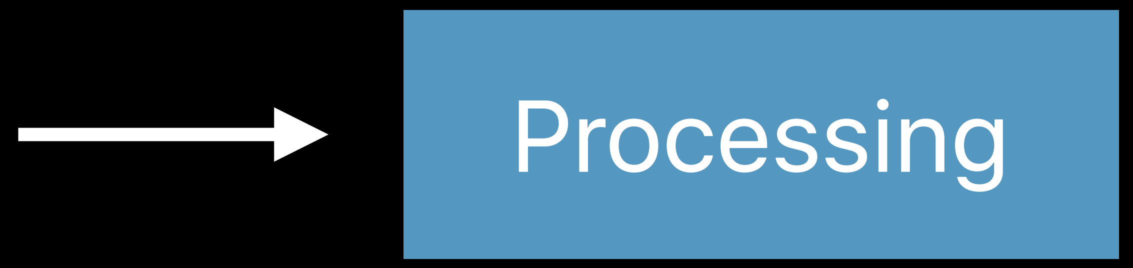
Natural Language Processing

Natural language text

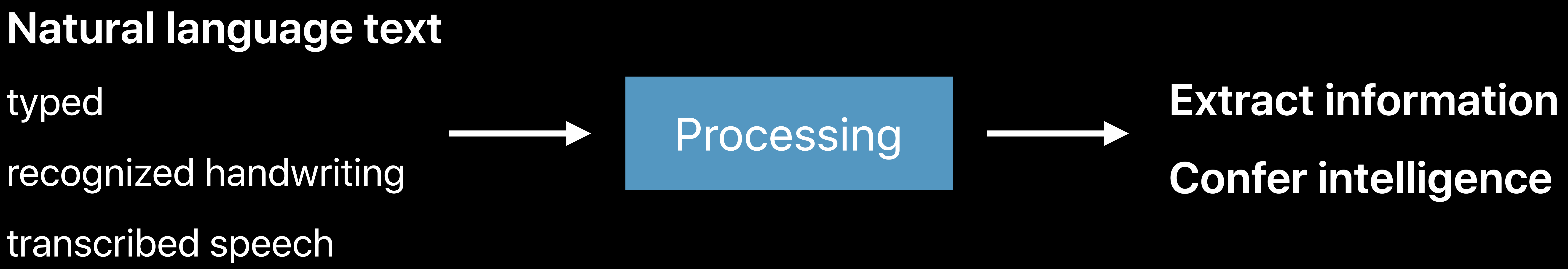
typed

recognized handwriting

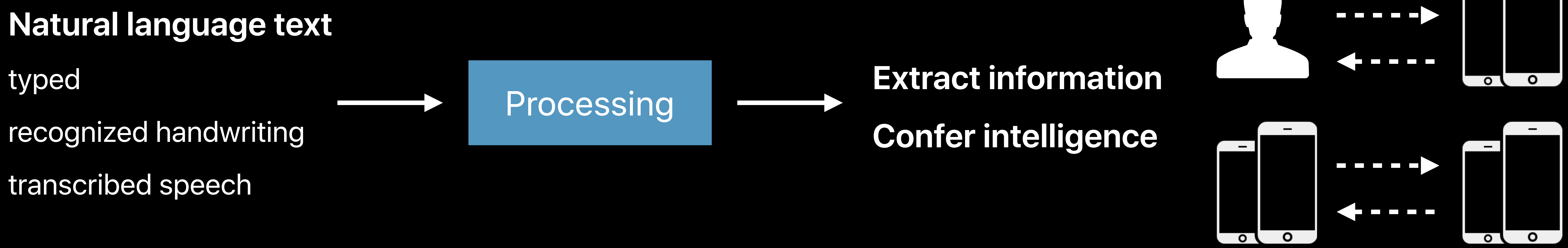
transcribed speech



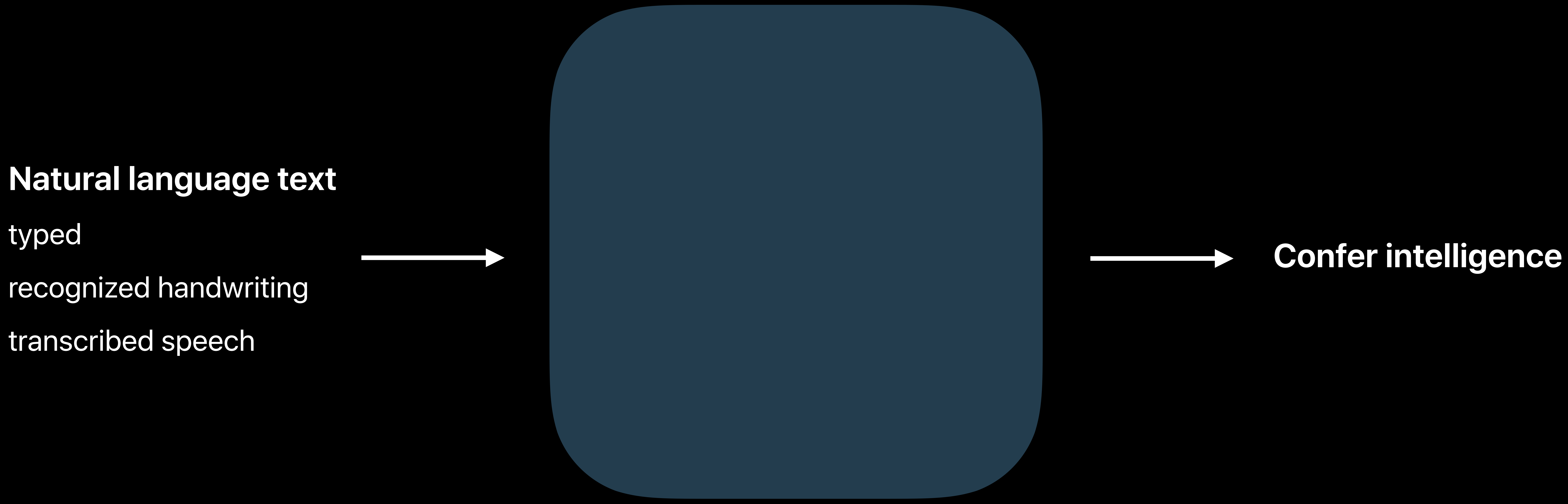
Natural Language Processing



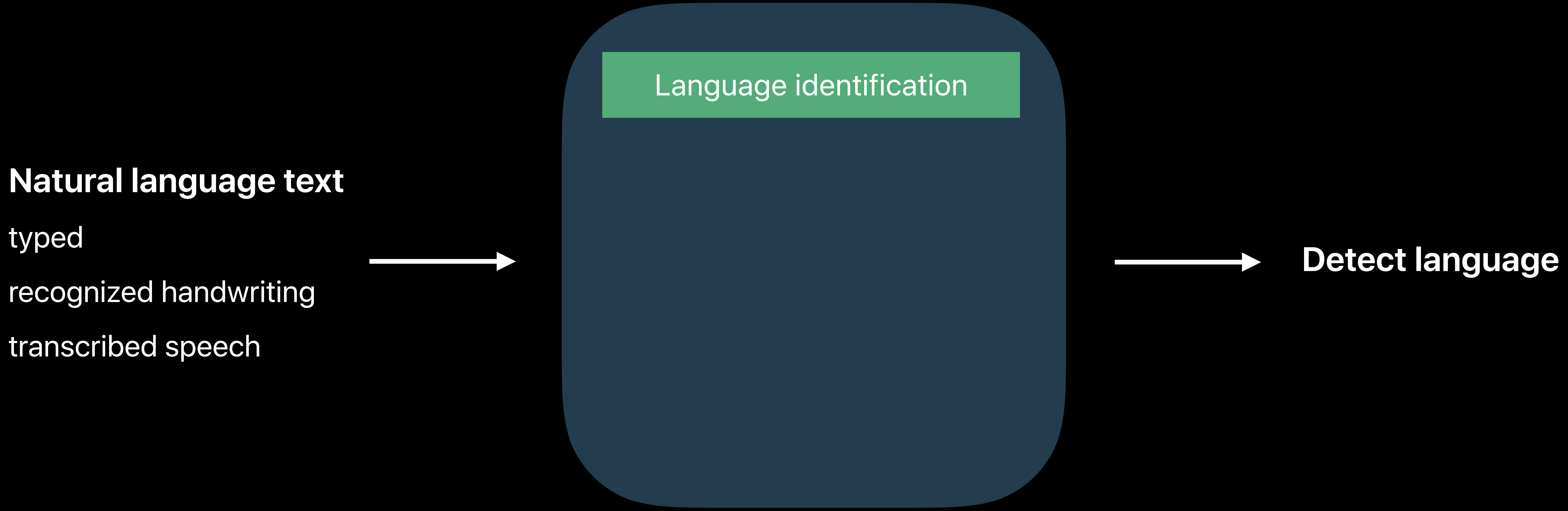
Natural Language Processing



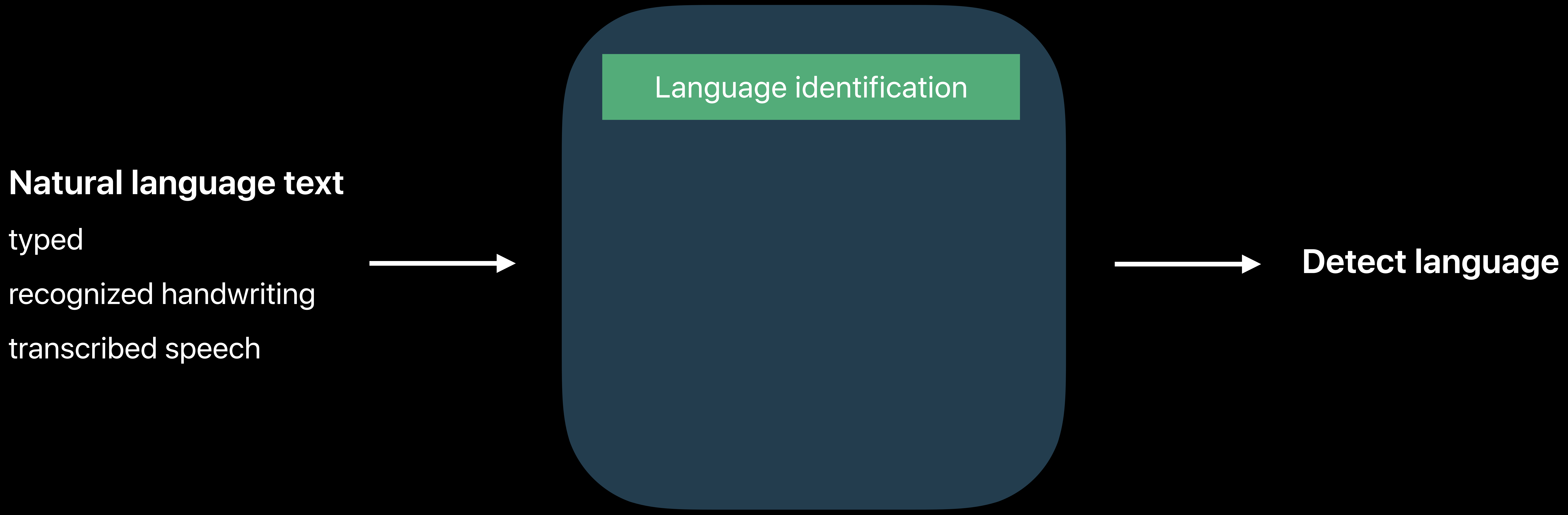
Natural Language Processing



Natural Language Processing



Natural Language Processing



Welcome to our talk on Natural Language Processing

欢迎参加我们关于自然语言处理的讲座 Bienvenidos a nuestra plática sobre Procesamiento de Lenguaje Natural

हमारी प्राकृतिक भाषा प्रसंस्करण भाषण में आपका स्वागत है Willkommen zu unserem Vortrag über Verarbeitung natürlicher Sprache

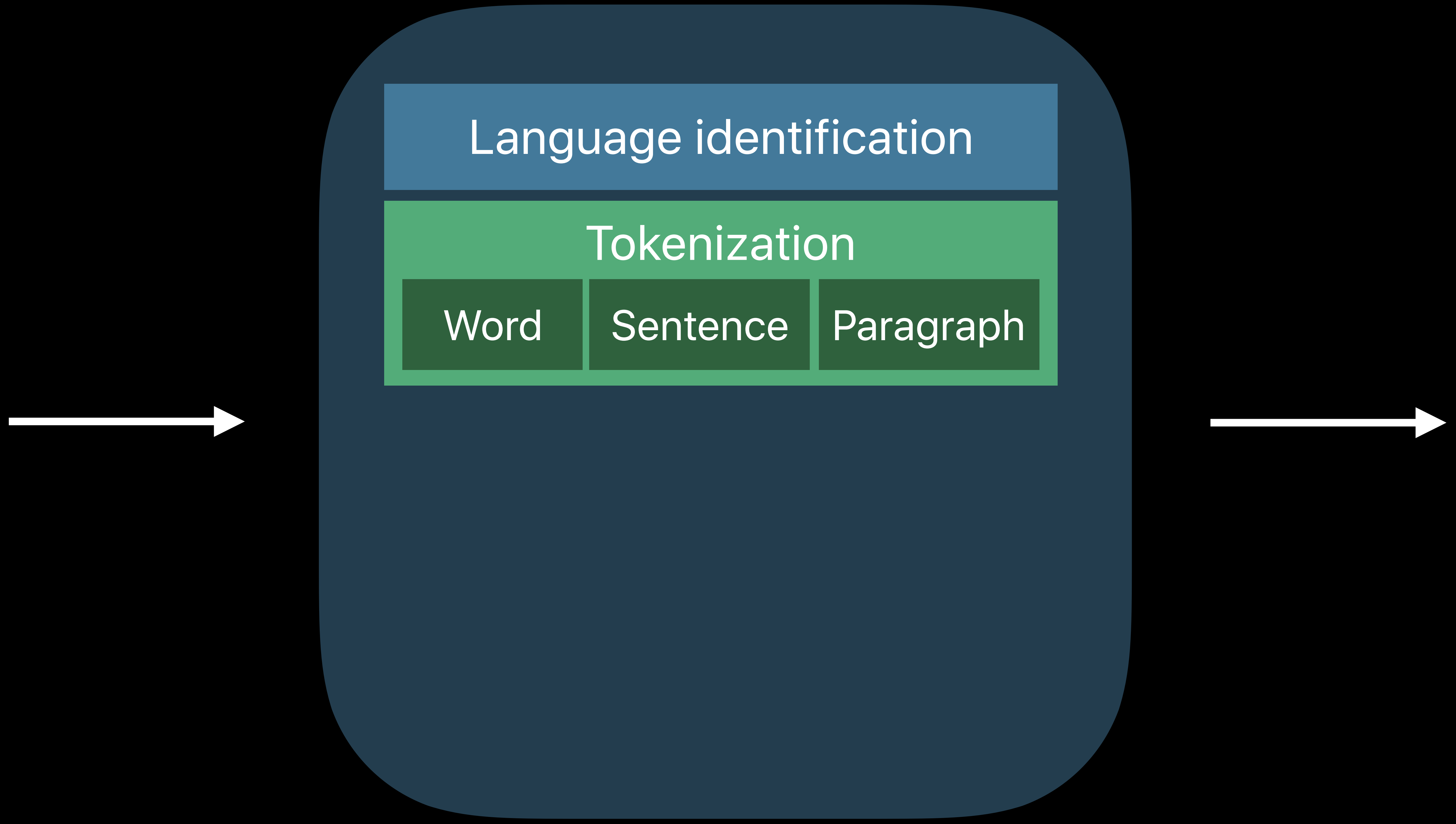
Natural Language Processing

Natural language text

typed

recognized handwriting

transcribed speech



Tokenize text

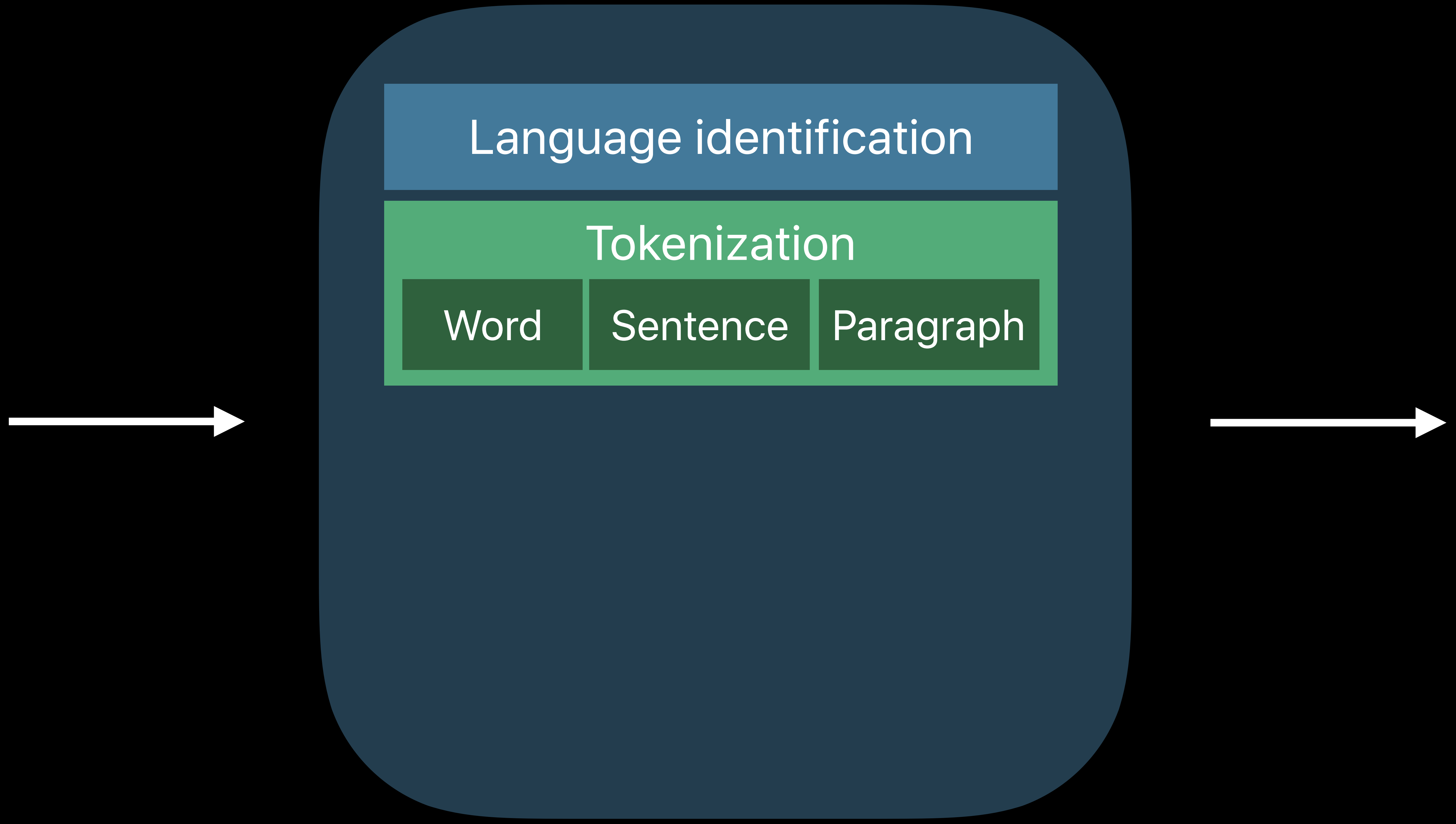
Natural Language Processing

Natural language text

typed

recognized handwriting

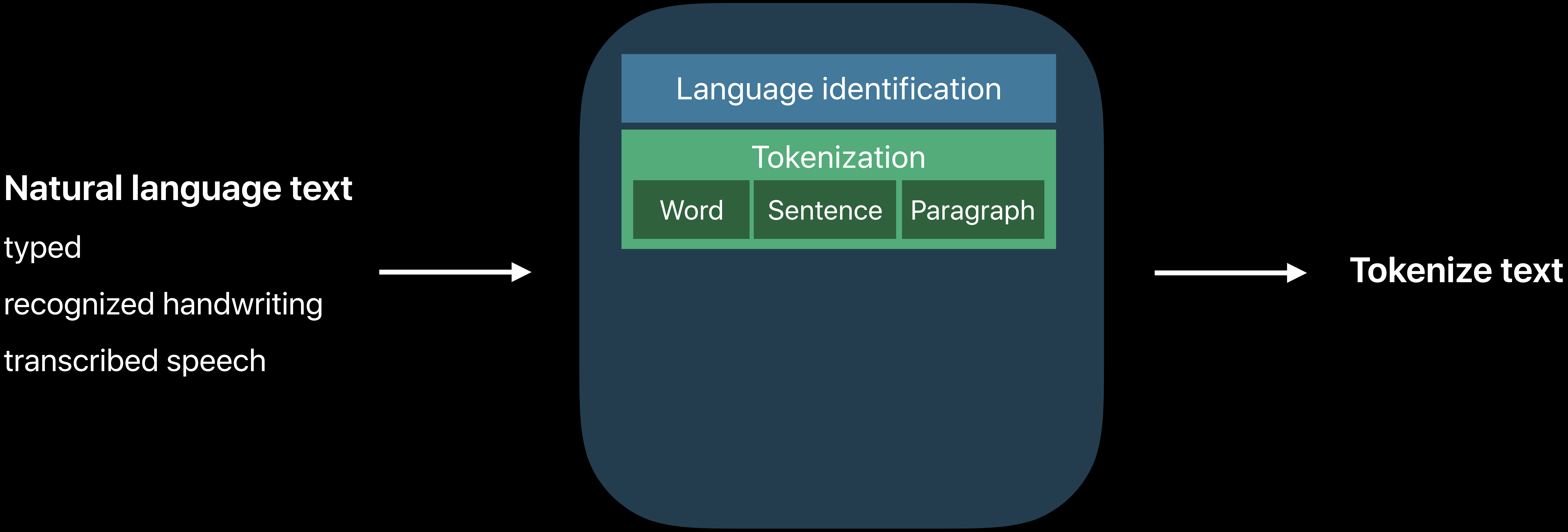
transcribed speech



Tokenize text

Mr. Tim Cook presided over the earnings report of Apple Inc. on Tuesday.

Natural Language Processing



星期二，蒂姆·库克先生主持了苹果公司的财报会议。

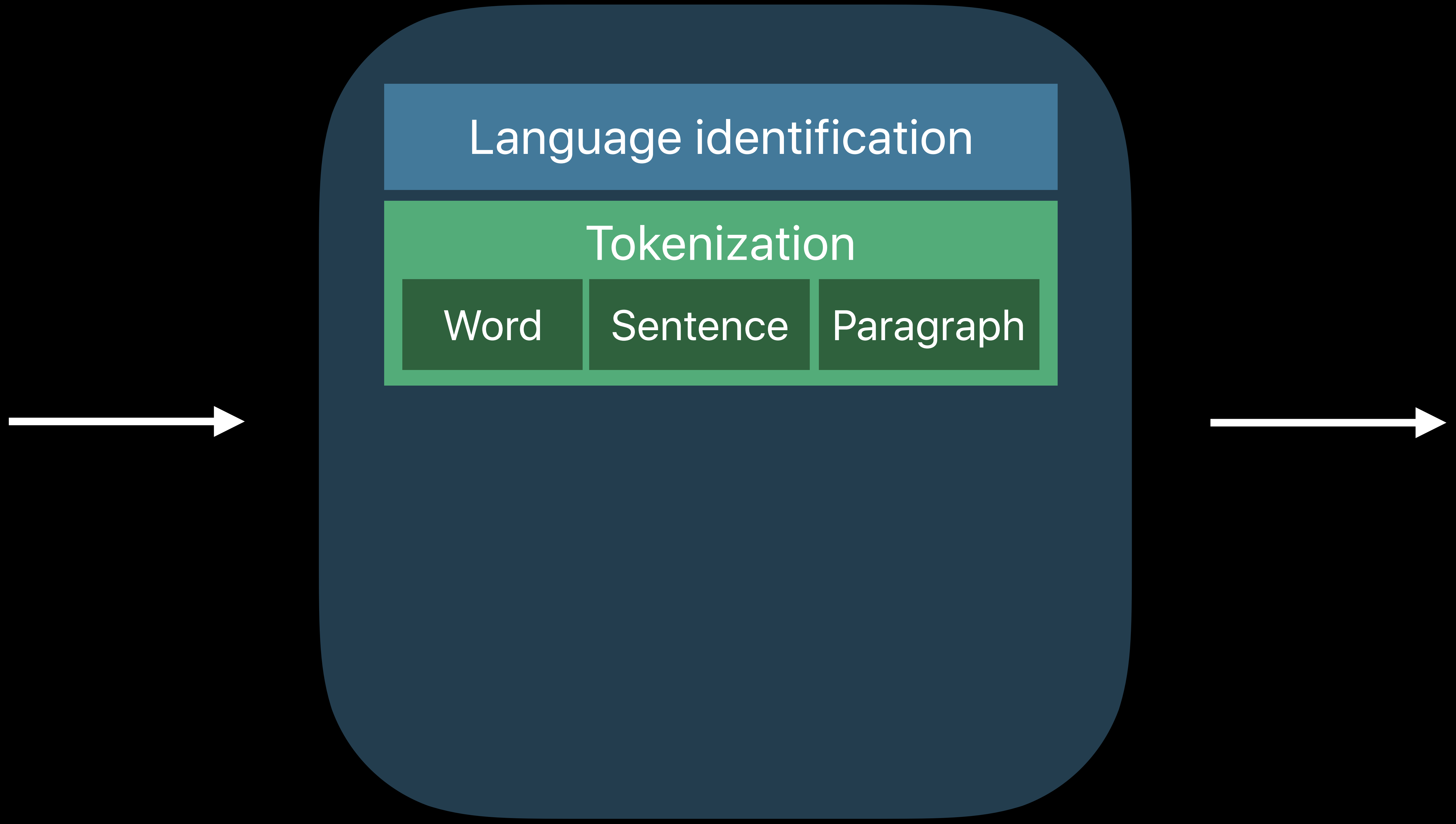
Natural Language Processing

Natural language text

typed

recognized handwriting

transcribed speech



Tokenize text

星期二，蒂姆·库克 先生 主持 了 苹果公司 的 财报 会议。

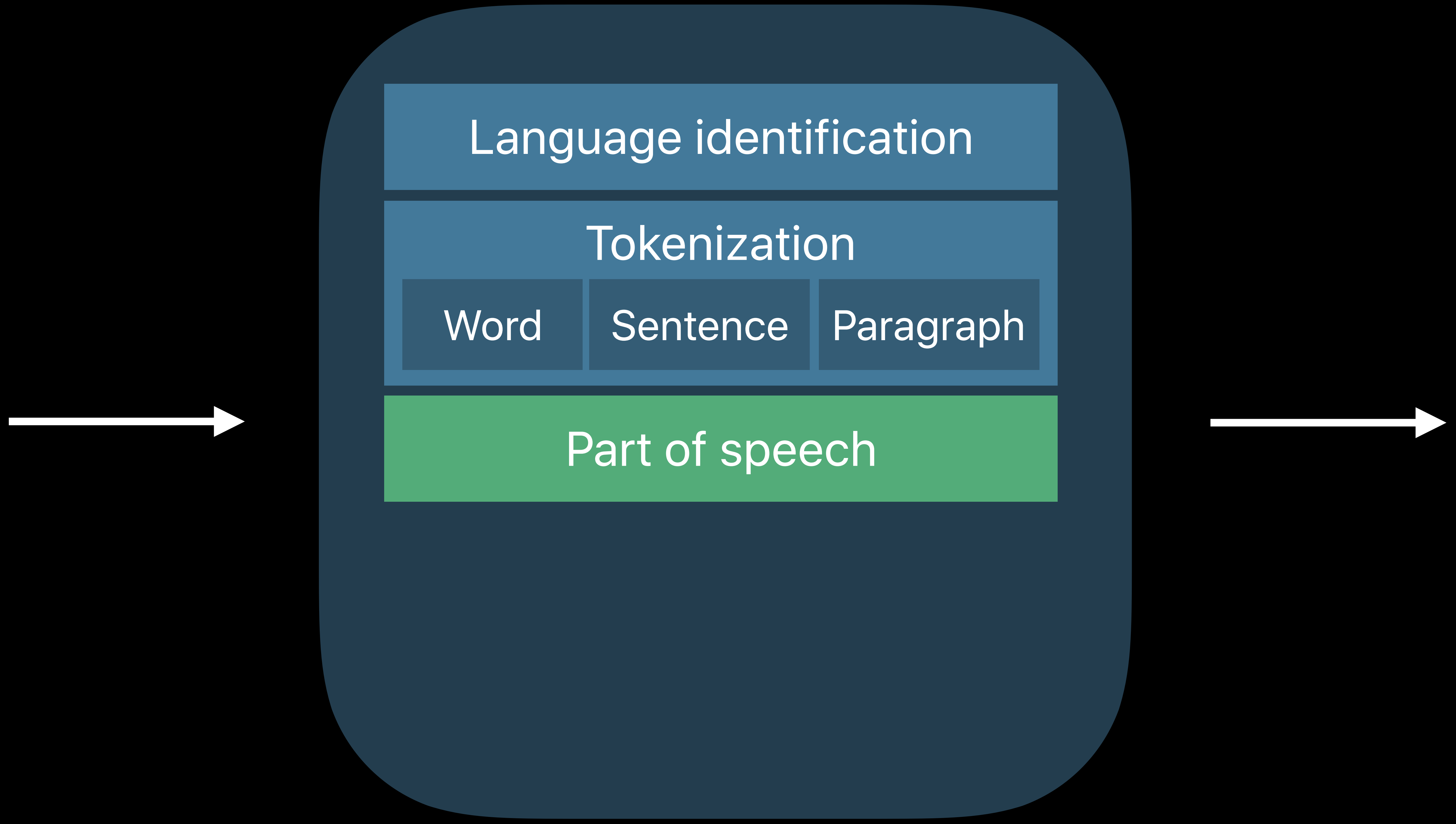
Natural Language Processing

Natural language text

typed

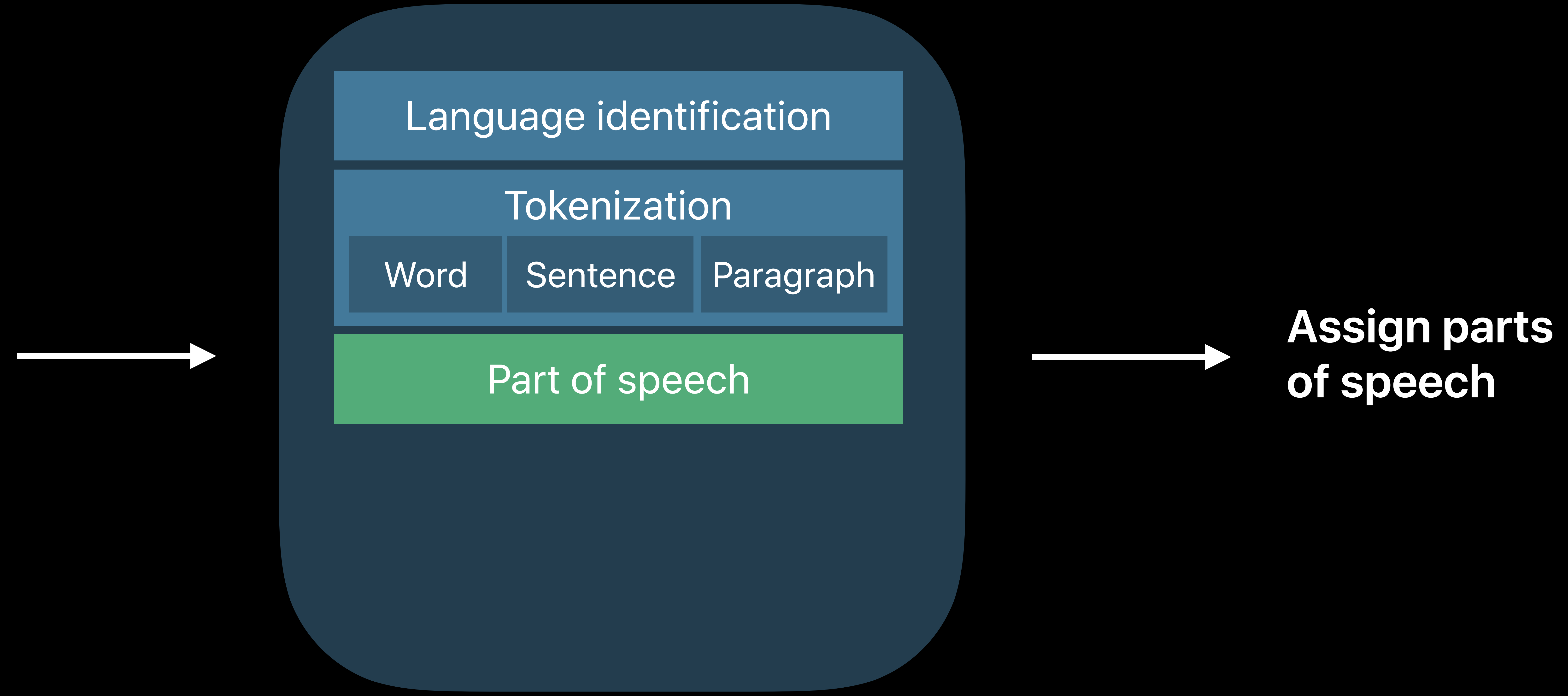
recognized handwriting

transcribed speech



Assign parts of speech

Natural Language Processing



Mr.	Tim	Cook	presided	over	the	earnings	report	of	Apple	Inc.	on	Tuesday	.
Noun	NNP	NNP	Verb	PP	DT	Noun	Noun	PP	Noun	Noun	PP	Noun	PUNCT

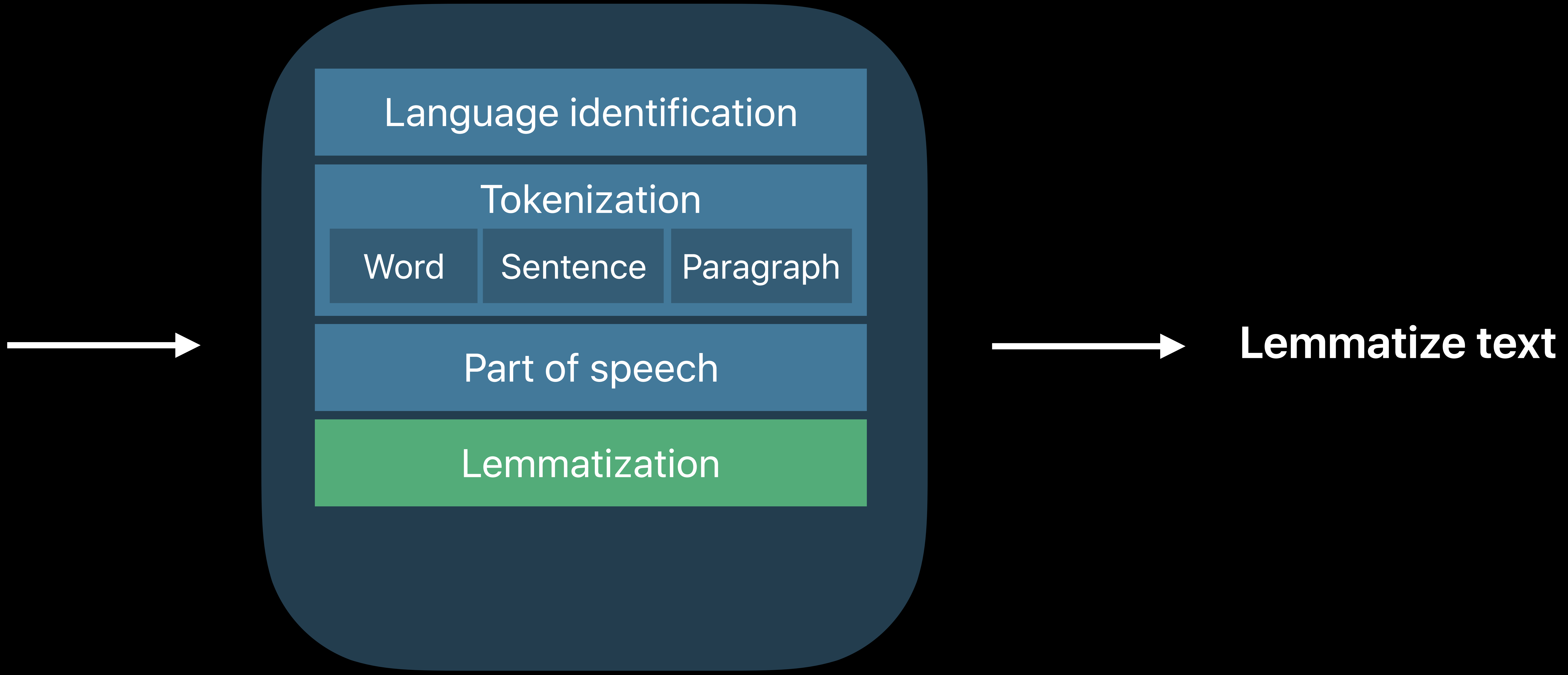
Natural Language Processing

Natural language text

typed

recognized handwriting

transcribed speech



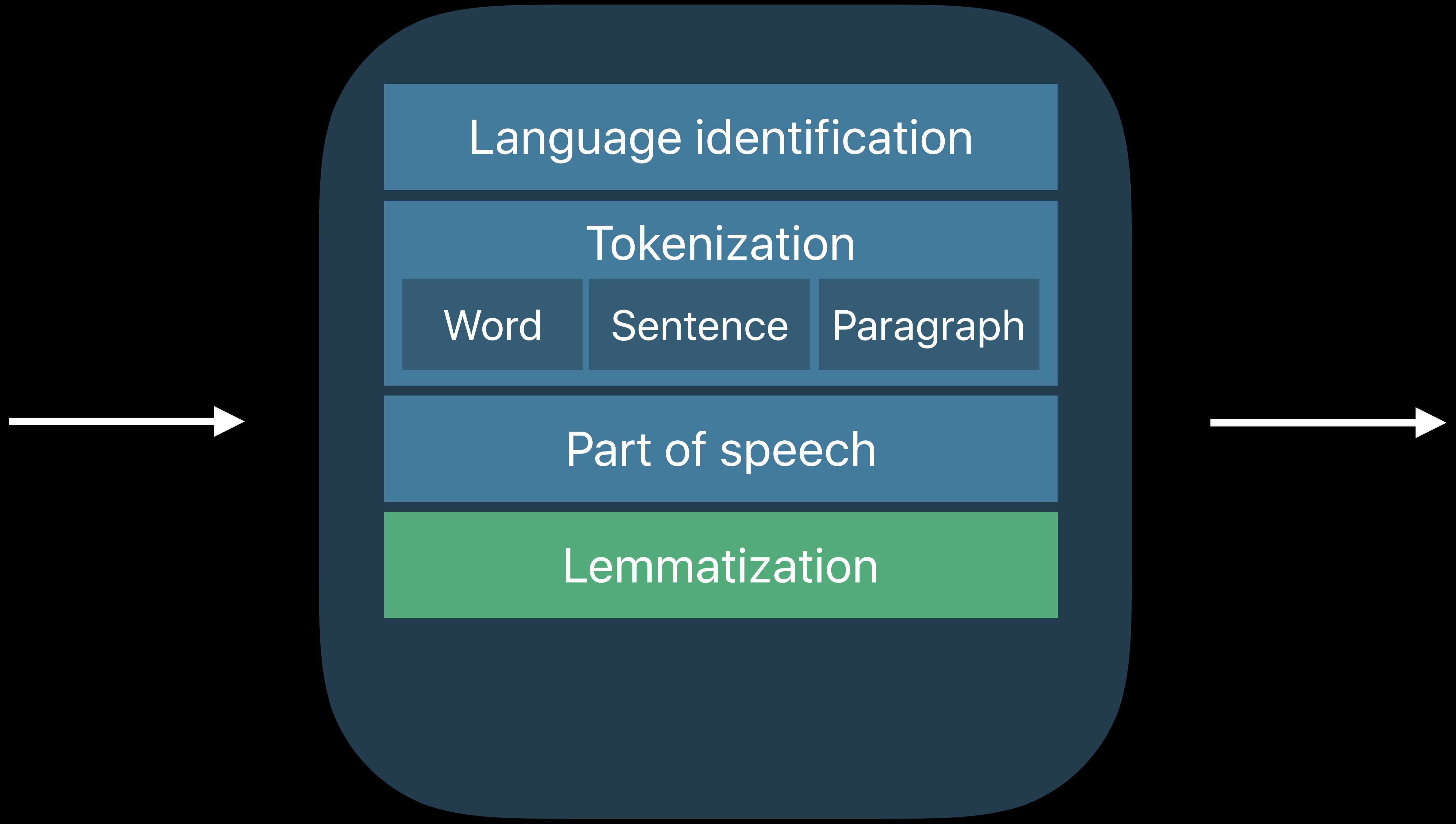
Natural Language Processing

Natural language text

typed

recognized handwriting

transcribed speech



Lemmatize text

Mr. Tim Cook presided over the earnings report of Apple. The stock was up 3% after hours.

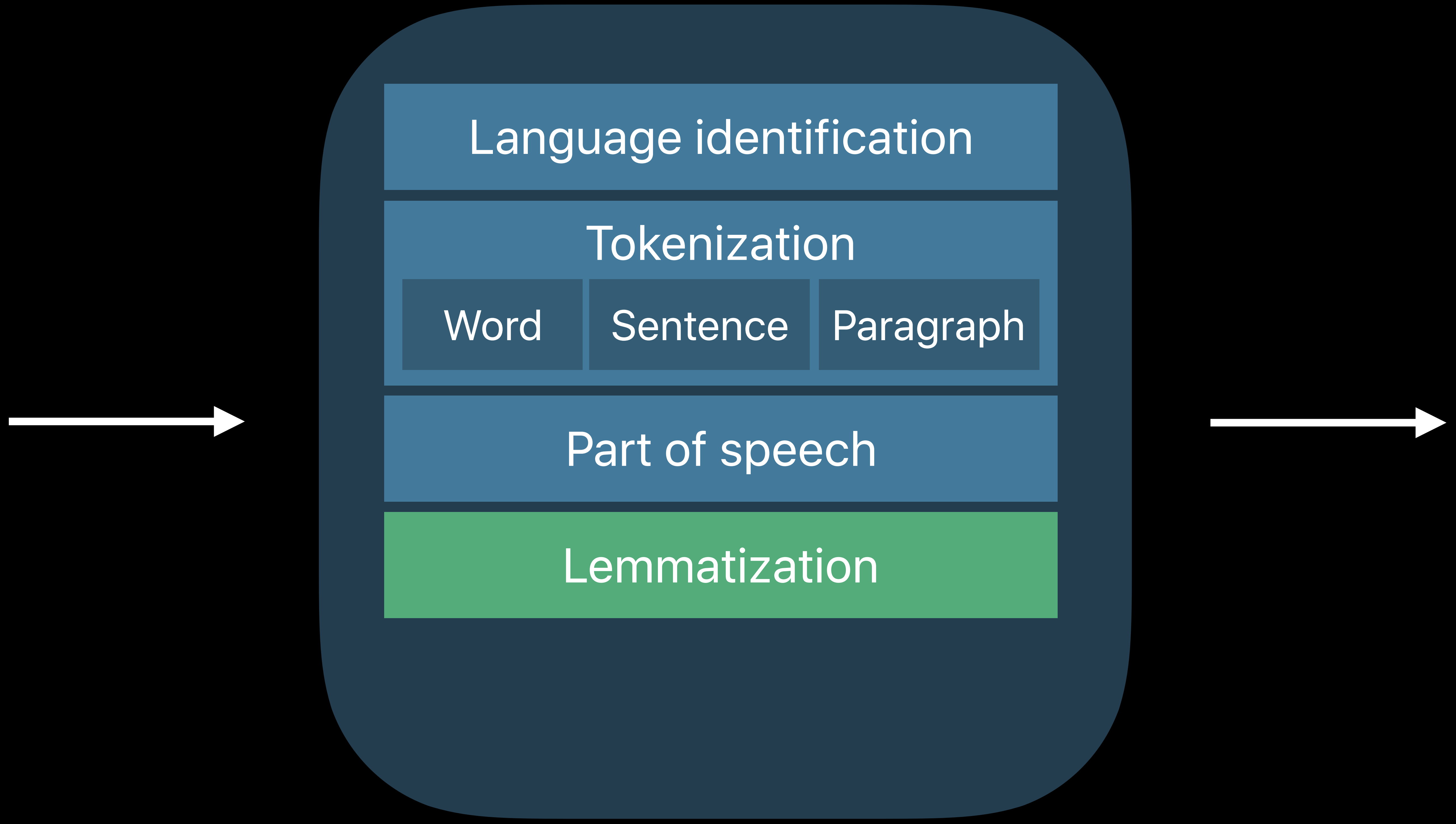
Natural Language Processing

Natural language text

typed

recognized handwriting

transcribed speech



Lemmatize text

	Verb									Verb		Noun
Mr. Tim Cook	presided	over	the	earnings	report	of	Apple.	The	stock	was	up 3% after	hours.
	preside									be		hour

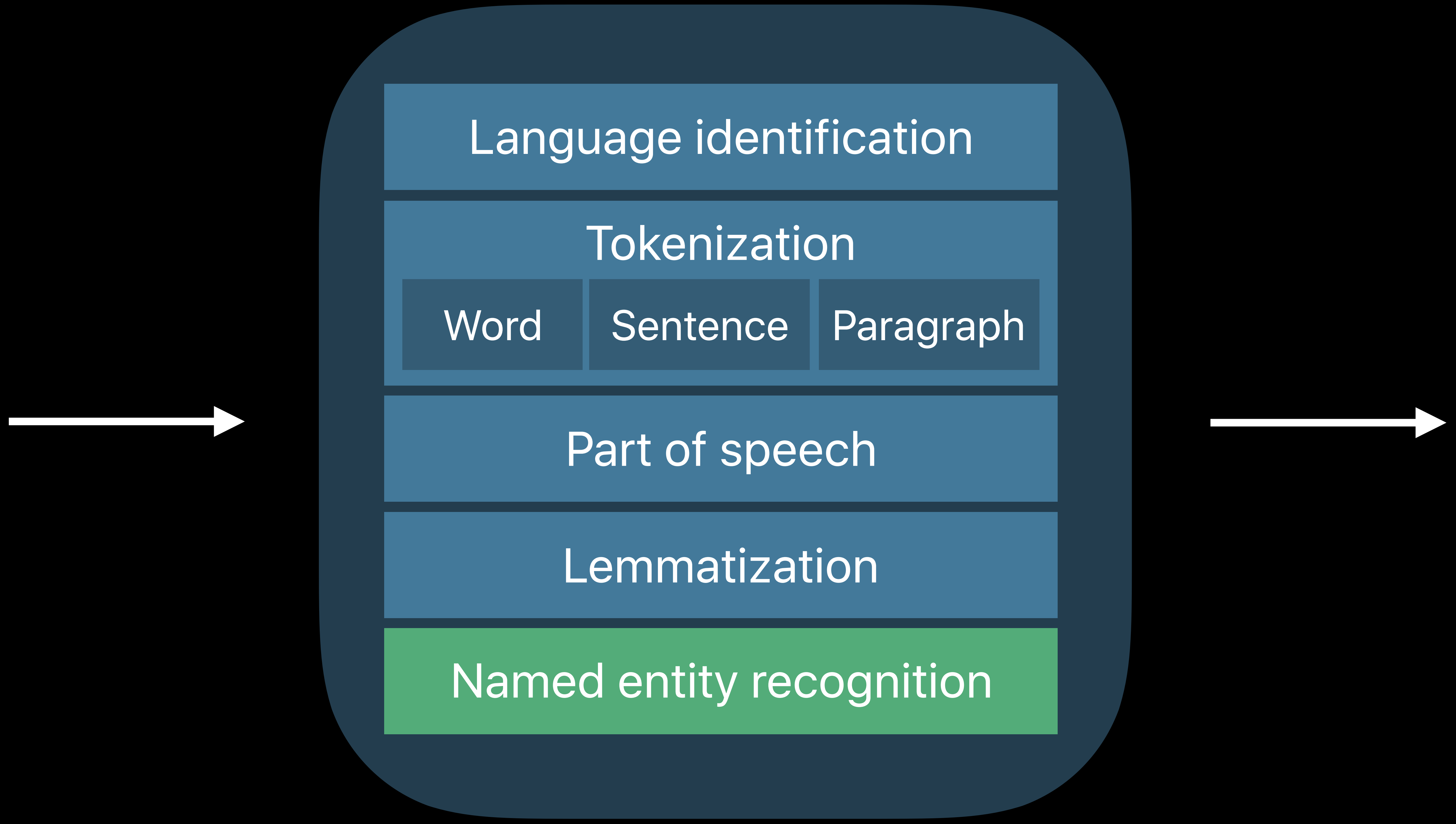
Natural Language Processing

Natural language text

typed

recognized handwriting

transcribed speech



Extract entities

Mr. Tim Cook presided over the earnings report of Apple Inc. on Tuesday.

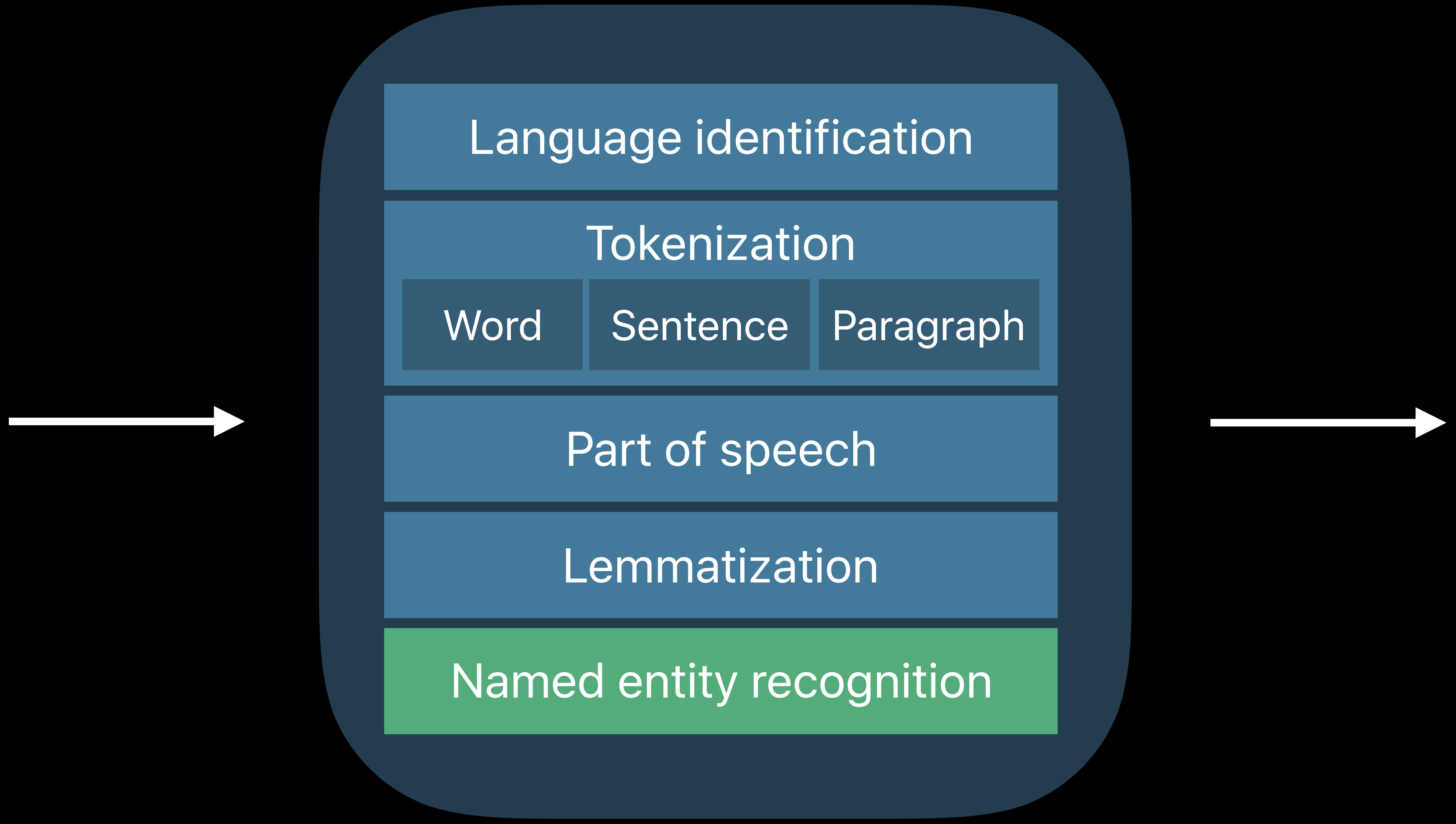
Natural Language Processing

Natural language text

typed

recognized handwriting

transcribed speech

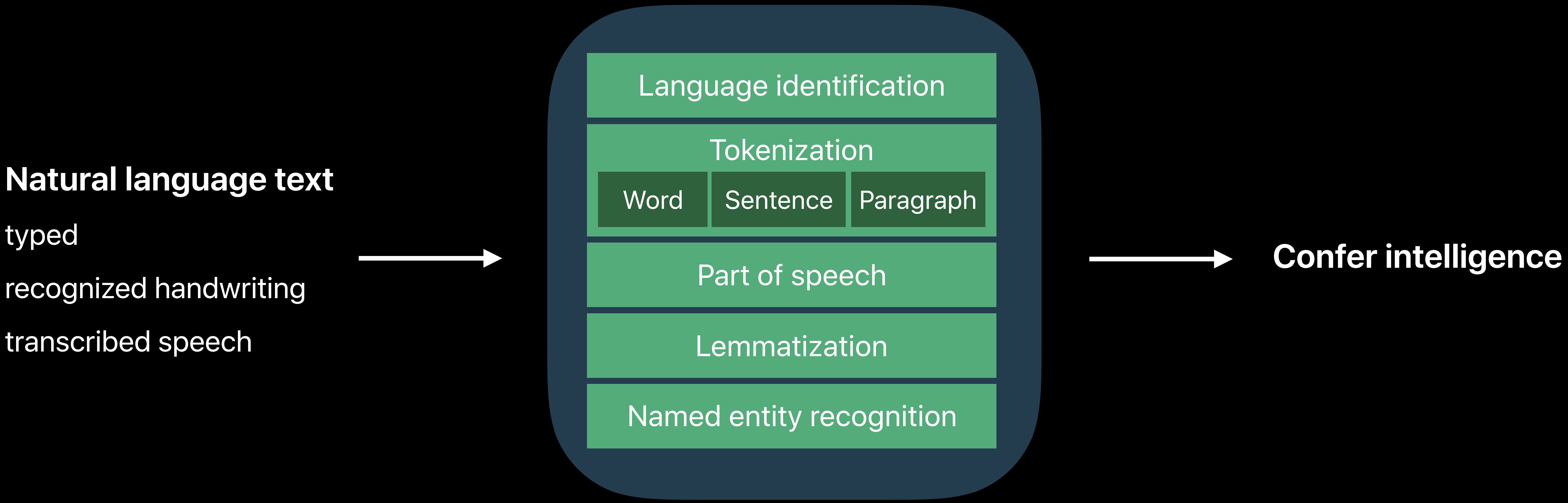


Extract entities

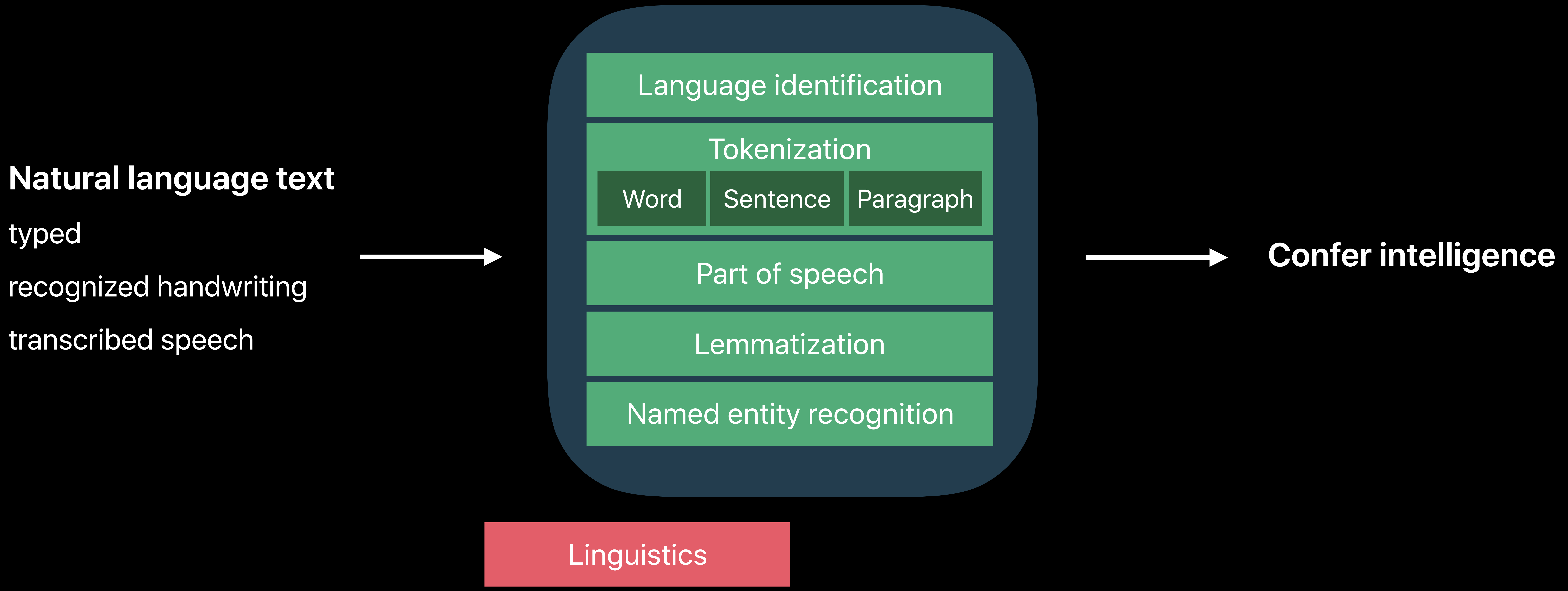
PER **ORG**

Mr. Tim Cook presided over the earnings report of Apple Inc. on Tuesday.

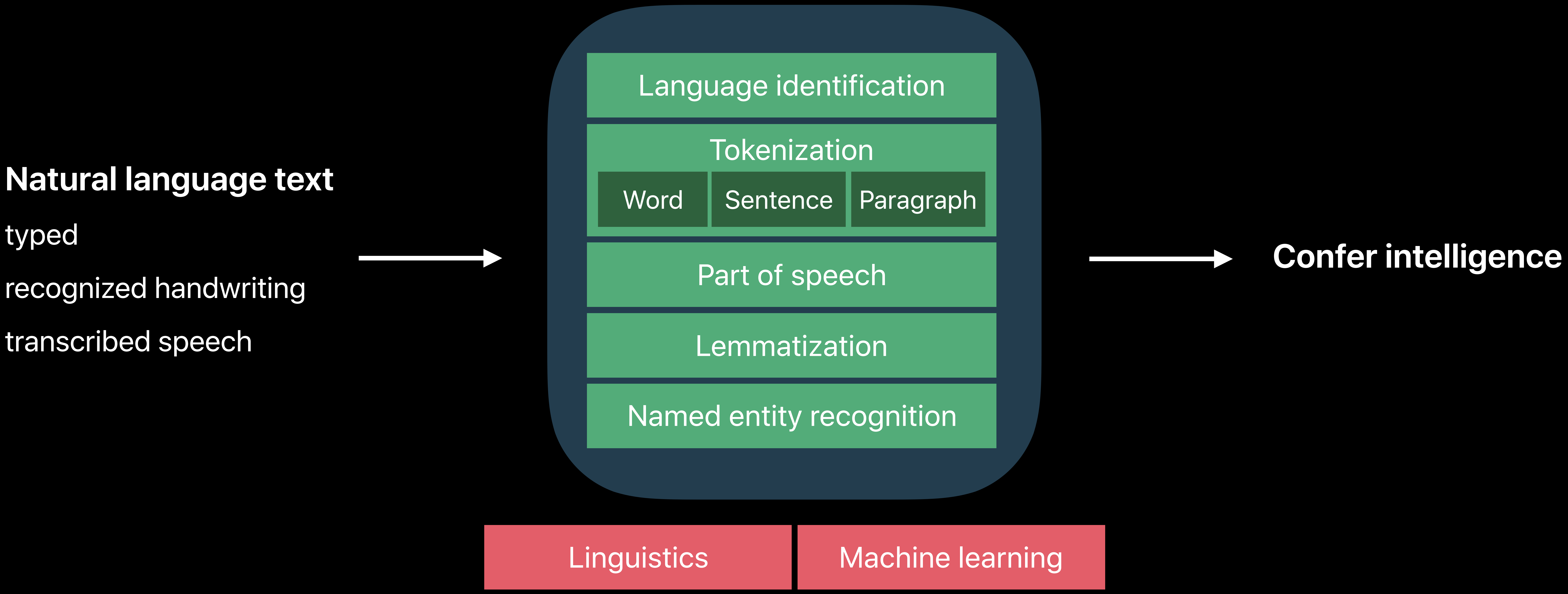
Natural Language Processing



Natural Language Processing



Natural Language Processing



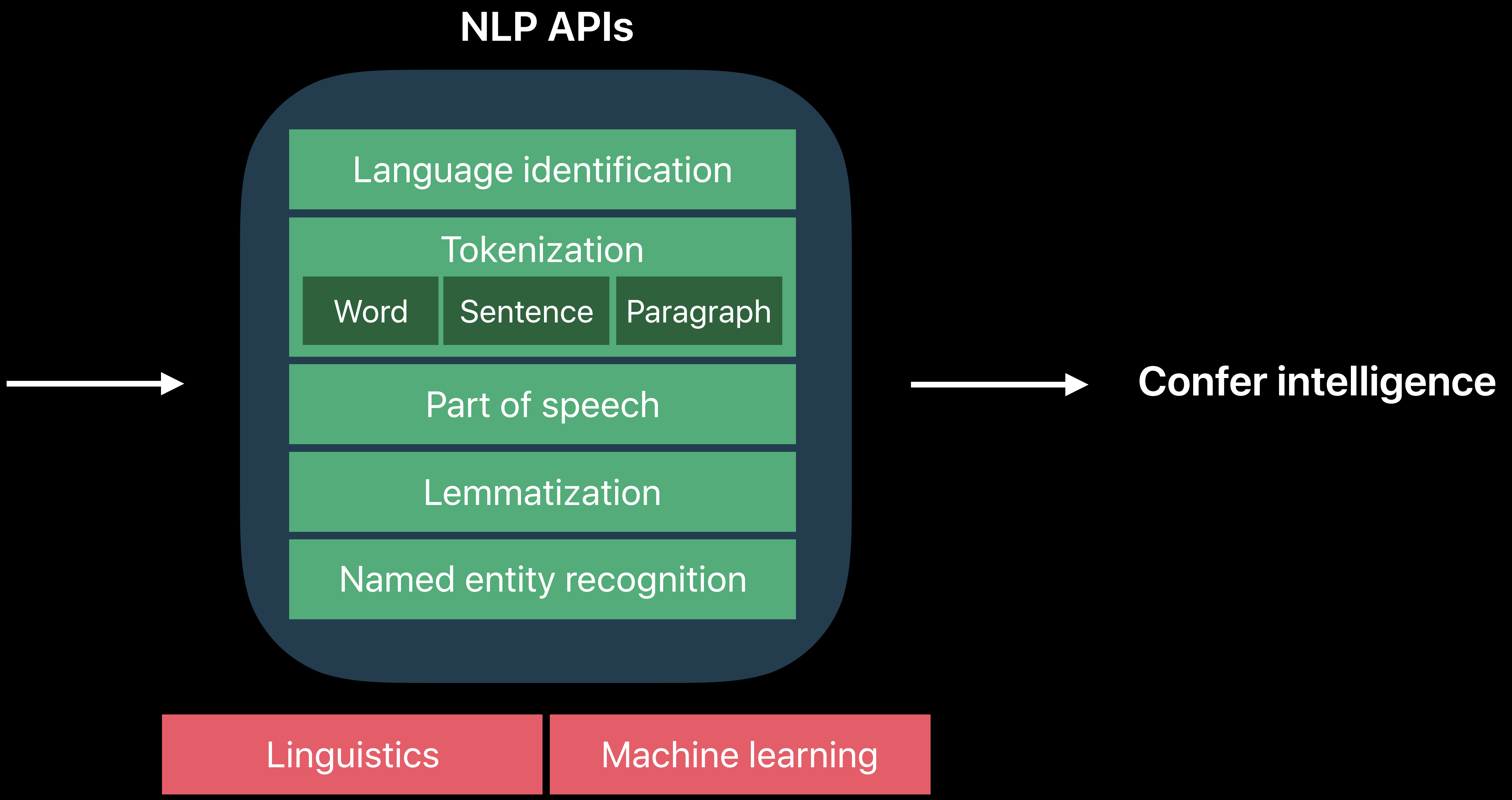
Natural Language Processing

Natural language text

typed

recognized handwriting

transcribed speech



Enough!
Tell me how to use them

NLP APIs

NLP APIs

NSLinguisticTagger

NSLinguisticTagger

Class in foundation

Segment and tag text

Linguistic tasks : tagSchemes

NSLinguisticTagger

Class in foundation

Segment and tag text

Linguistic tasks : tagSchemes

Developer

NSLinguisticTagger

NEW

NSLinguisticTagger

NEW

Tagging units

NSLinguisticTagger

NEW

Tagging units

```
public enum NSLinguisticTaggerUnit : Int {
    case word
    case sentence
    case paragraph
    case document
}
```

NSLinguisticTagger

NEW

Tagging units

```
public enum NSLinguisticTaggerUnit : Int {
    case word
    case sentence
    case paragraph
    case document
}
```

NSLinguisticTagger

NEW

Tagging units

```
public enum NSLinguisticTaggerUnit : Int {
    case word
    case sentence
    case paragraph
    case document
}
```

NSLinguisticTagger

NEW

Tagging units

```
public enum NSLinguisticTaggerUnit : Int {
    case word
    case sentence
    case paragraph
    case document
}
```

NSLinguisticTagger

NEW

Tagging units

```
public enum NSLinguisticTaggerUnit : Int {
    case word
    case sentence
    case paragraph
    case document
}
```

NSLinguisticTagger

NEW

Tagging units

```
public enum NSLinguisticTaggerUnit : Int {
    case word
    case sentence
    case paragraph
    case document
}
```

Units and schemes

NSLinguisticTagger

NEW

Tagging units

```
public enum NSLinguisticTaggerUnit : Int {
    case word
    case sentence
    case paragraph
    case document
}
```

Units and schemes

```
class func availableTagSchemes (for unit: NSLinguisticTaggerUnit, language: String) ->
                                [NSLinguisticTagScheme]
```

NSLinguisticTagger

NEW

Tagging units

```
public enum NSLinguisticTaggerUnit : Int {
    case word
    case sentence
    case paragraph
    case document
}
```

Units and schemes

```
class func availableTagSchemes(for unit: NSLinguisticTaggerUnit, language: String) ->
                                  [NSLinguisticTagScheme]
```

NSLinguisticTagger

NEW

NSLinguisticTagger

NEW

dominantLanguage

NSLinguisticTagger

NEW

dominantLanguage

Swift 4 : named types for tags and tagSchemes

NSLinguisticTagger

NEW

dominantLanguage

Swift 4 : named types for tags and tagSchemes

Improved performance

Higher accuracy

Additional language support

Winnow and Whisk

Winnow and Whisk

Winnow and Whisk

Winnow

Tag photos with descriptions

Winnow

Tag photos with descriptions

Winnow

Tag photos with descriptions

We hiked to the top and
then ran down. Lizzie won!

Kids partied
with all the goodies…

Die Kleinen haben friedlich
zusammen gespielt.

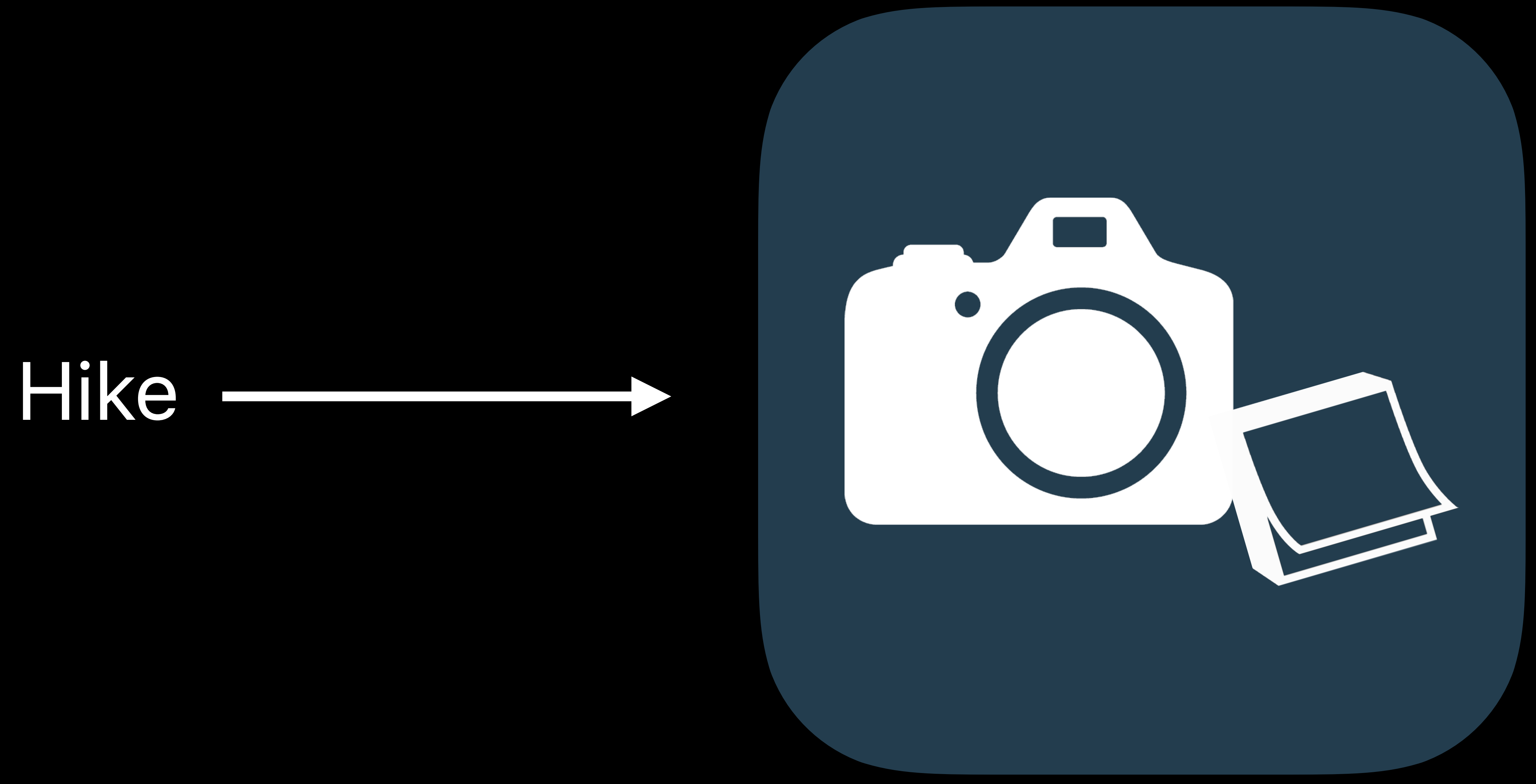
July 4th, 2016…
summer parties at the
Holden’s begin!

Winnow

Search for pictures

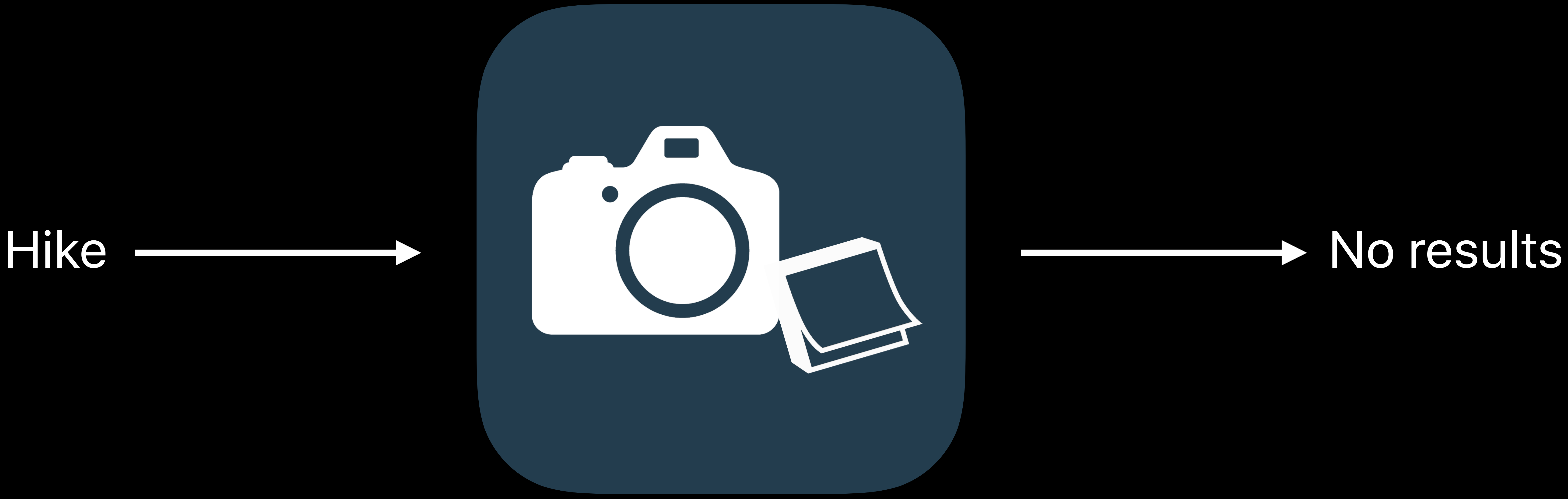
Winnow

Search for pictures



Winnow

Search for pictures

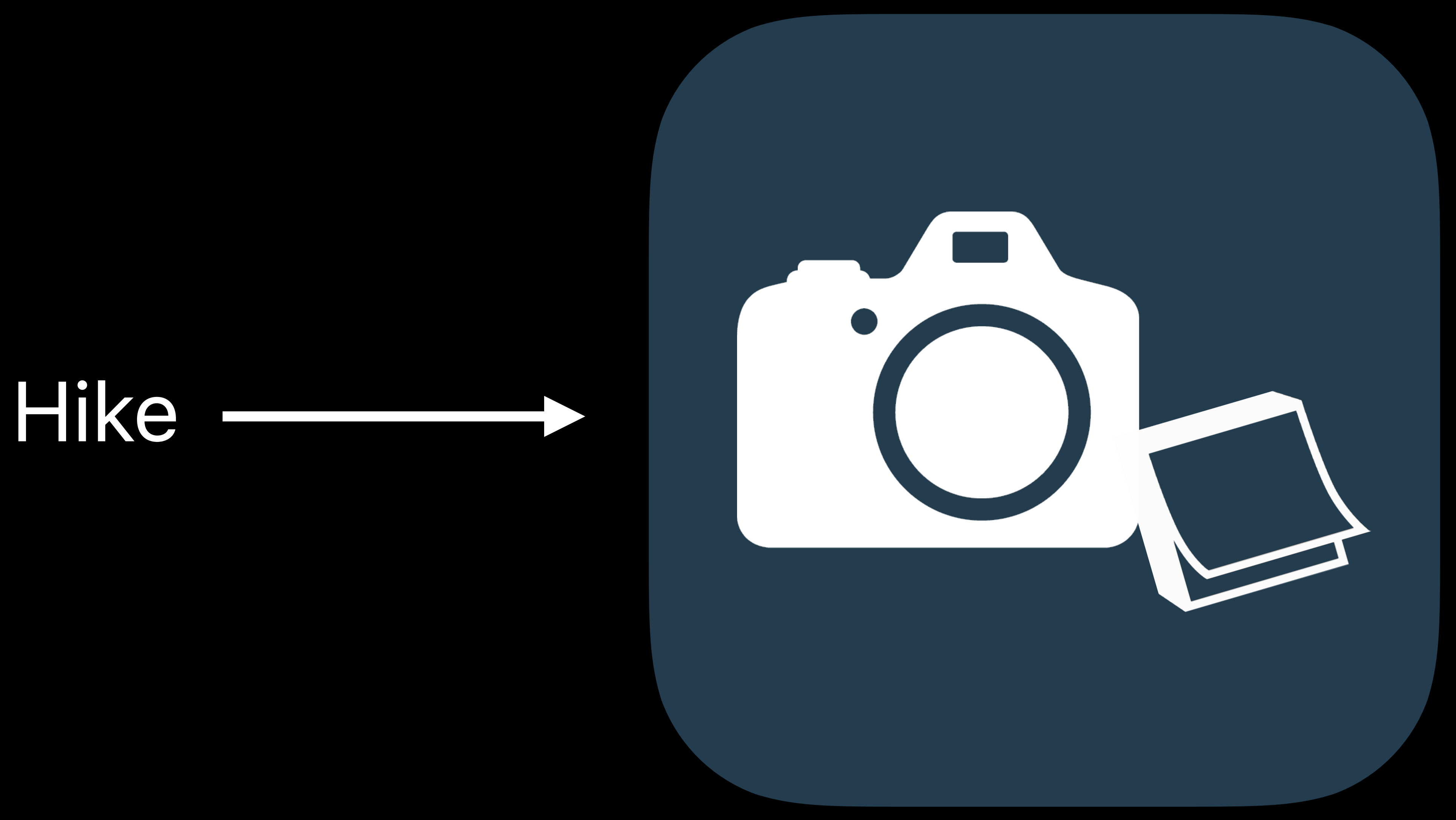


Winnow

Improve search experience using NLP

Winnow

Improve search experience using NLP



Winnow

Improve search experience using NLP

Hike ⟶ ⟶

We hiked to the top and then ran down. Lizzie won!

Great hikes make great pics!

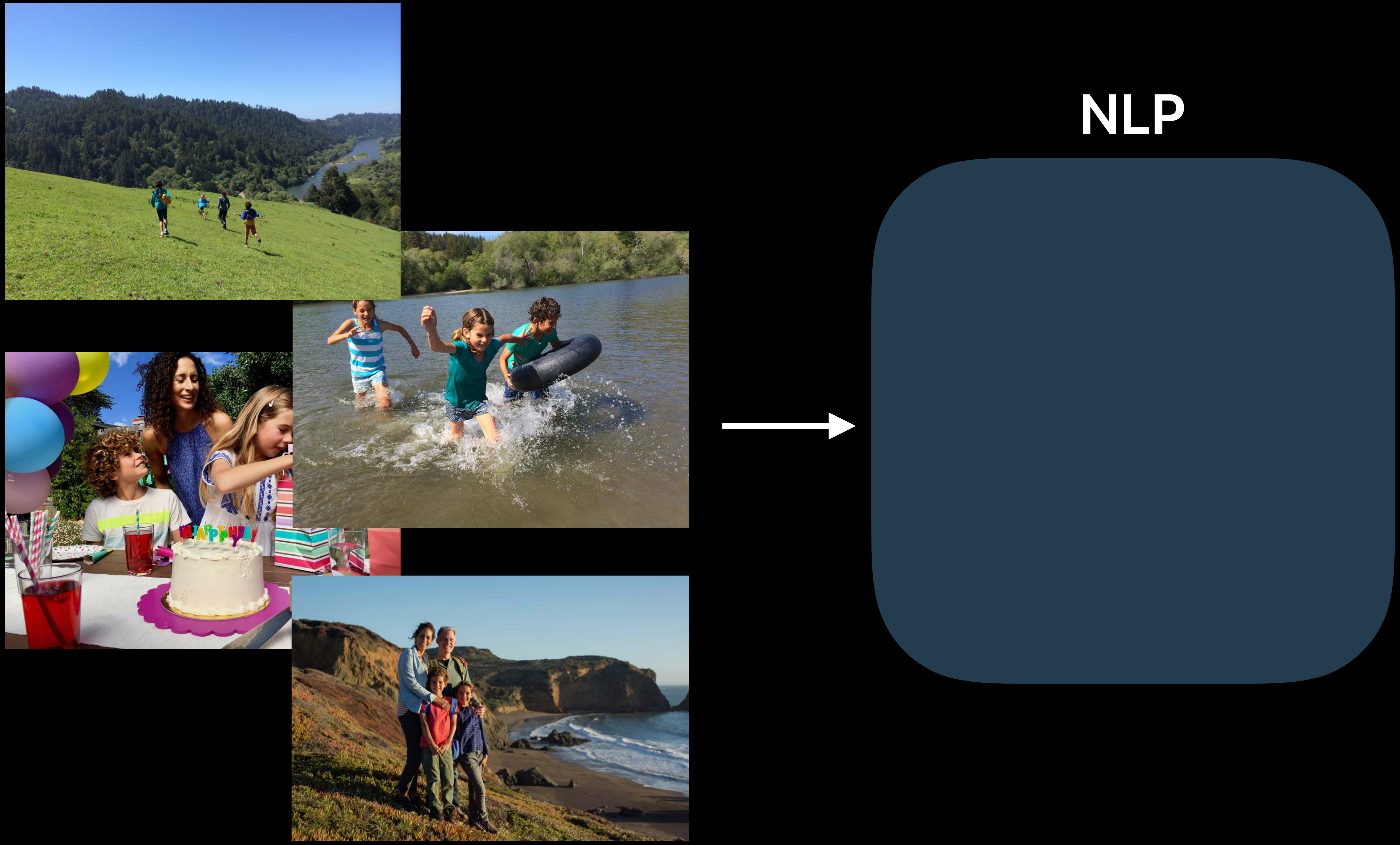
Holden family loves hiking and also posing for pictures

The hiking boys!

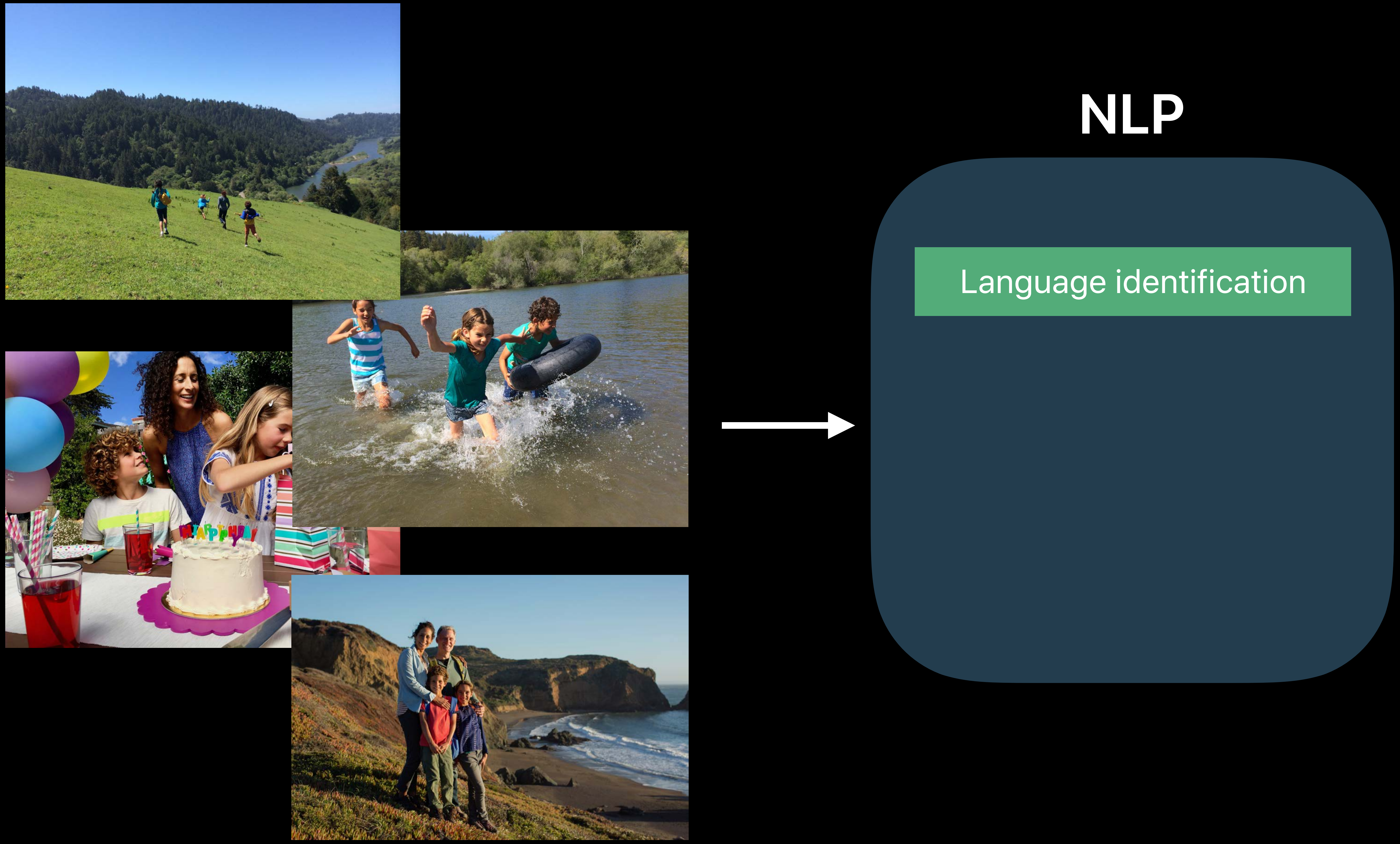
Winnow

Winnow

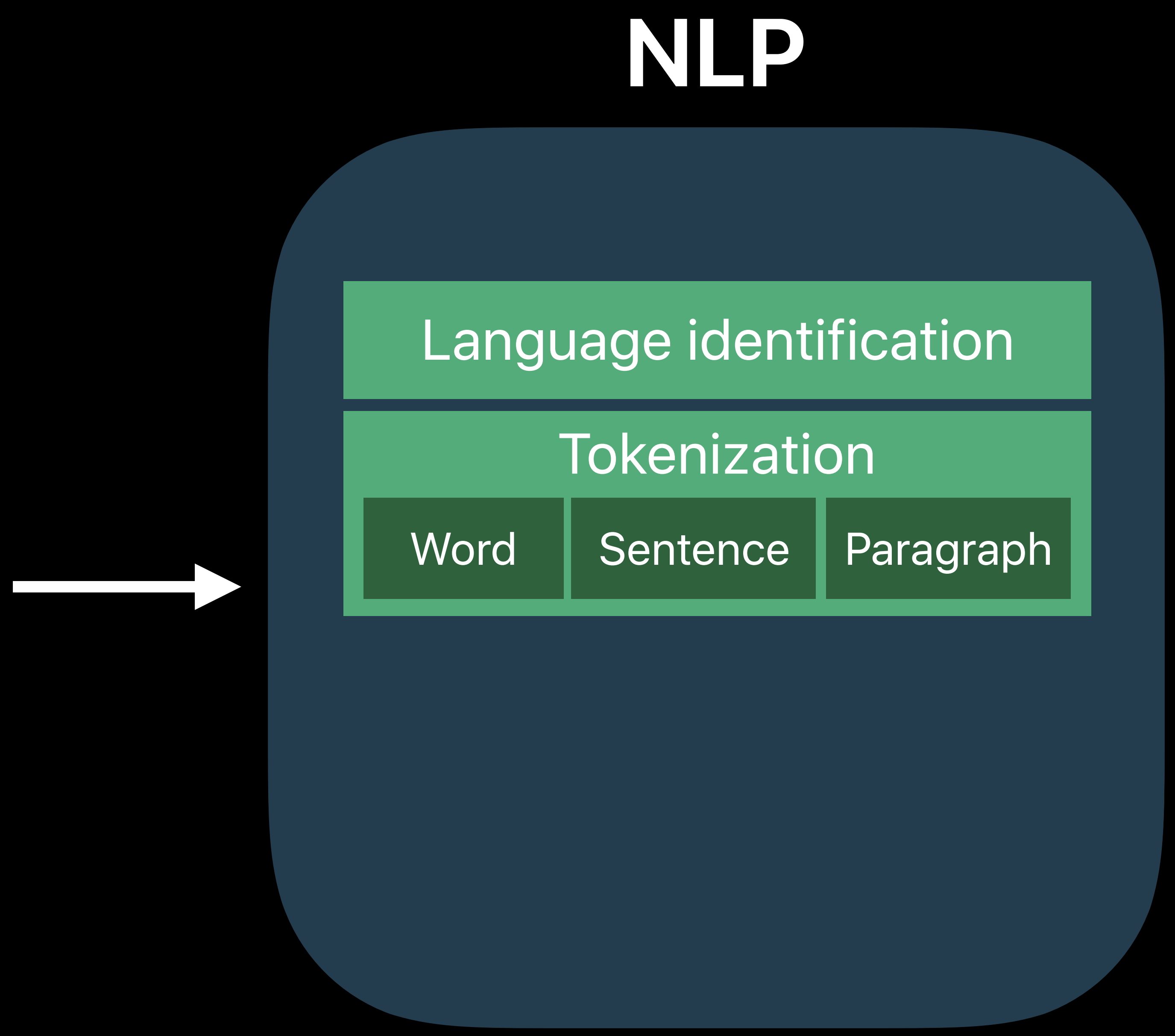
Winnow



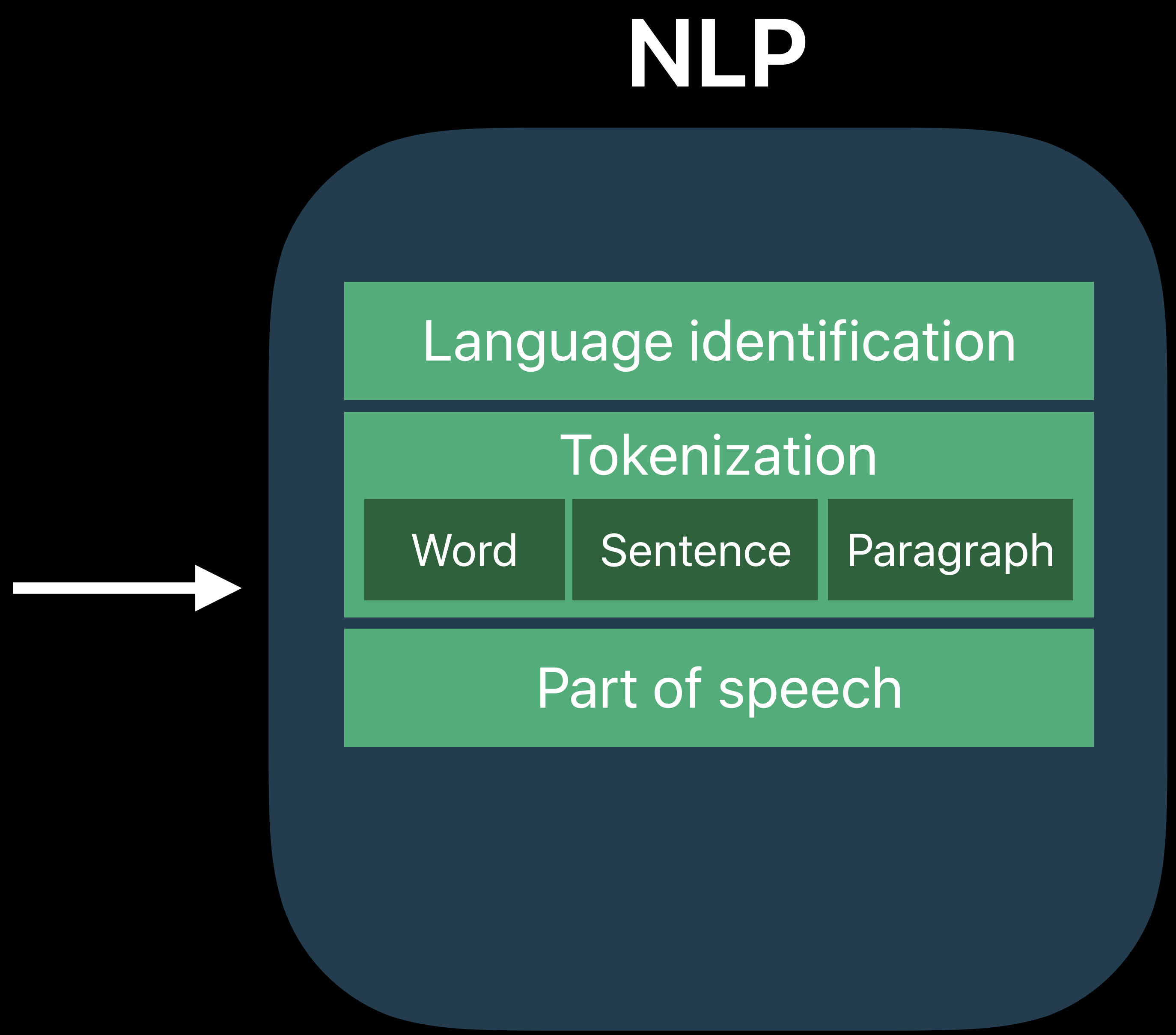
Winnow



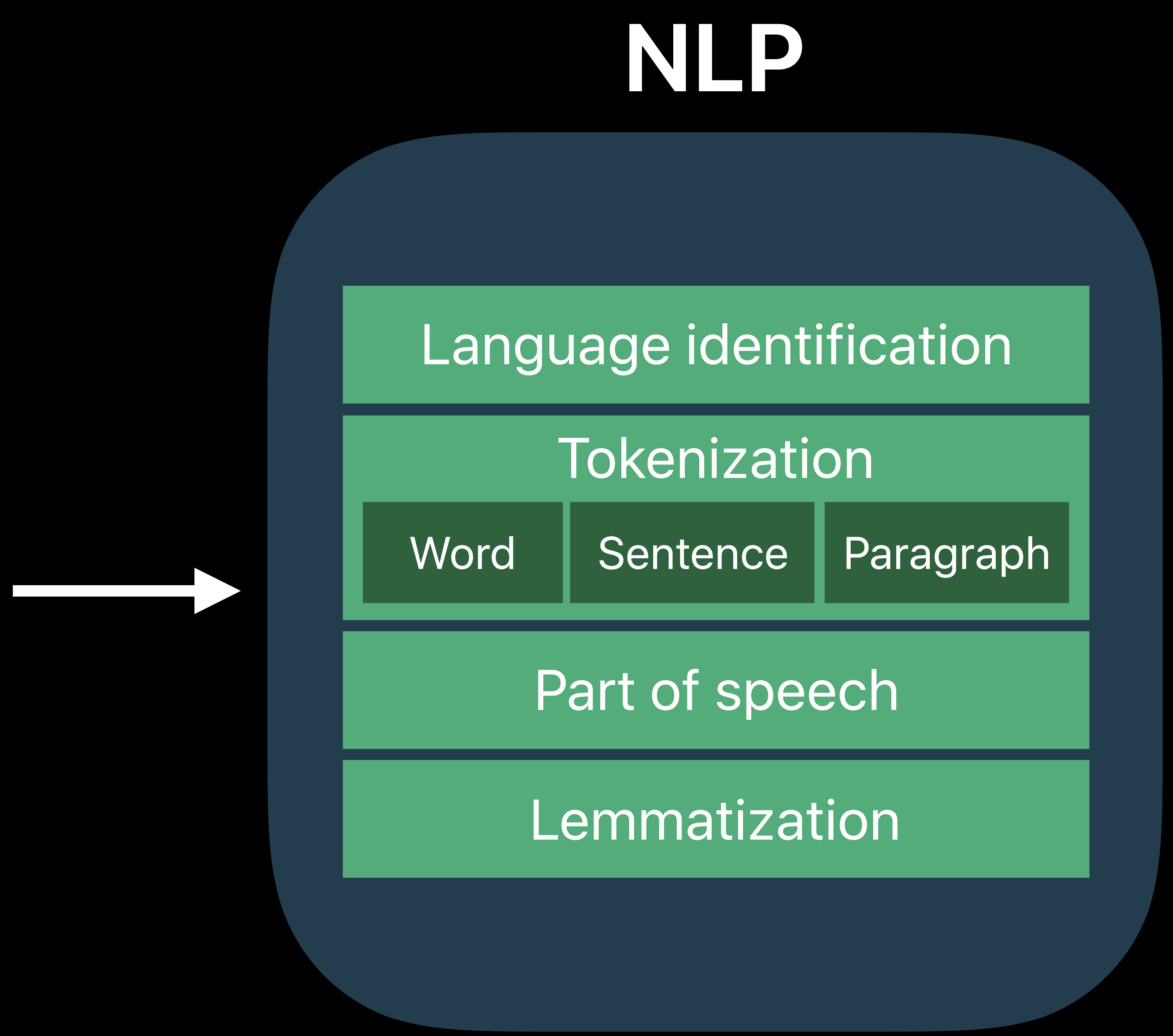
Winnow


NLP
Language identification
Tokenization
Word
Sentence
Paragraph

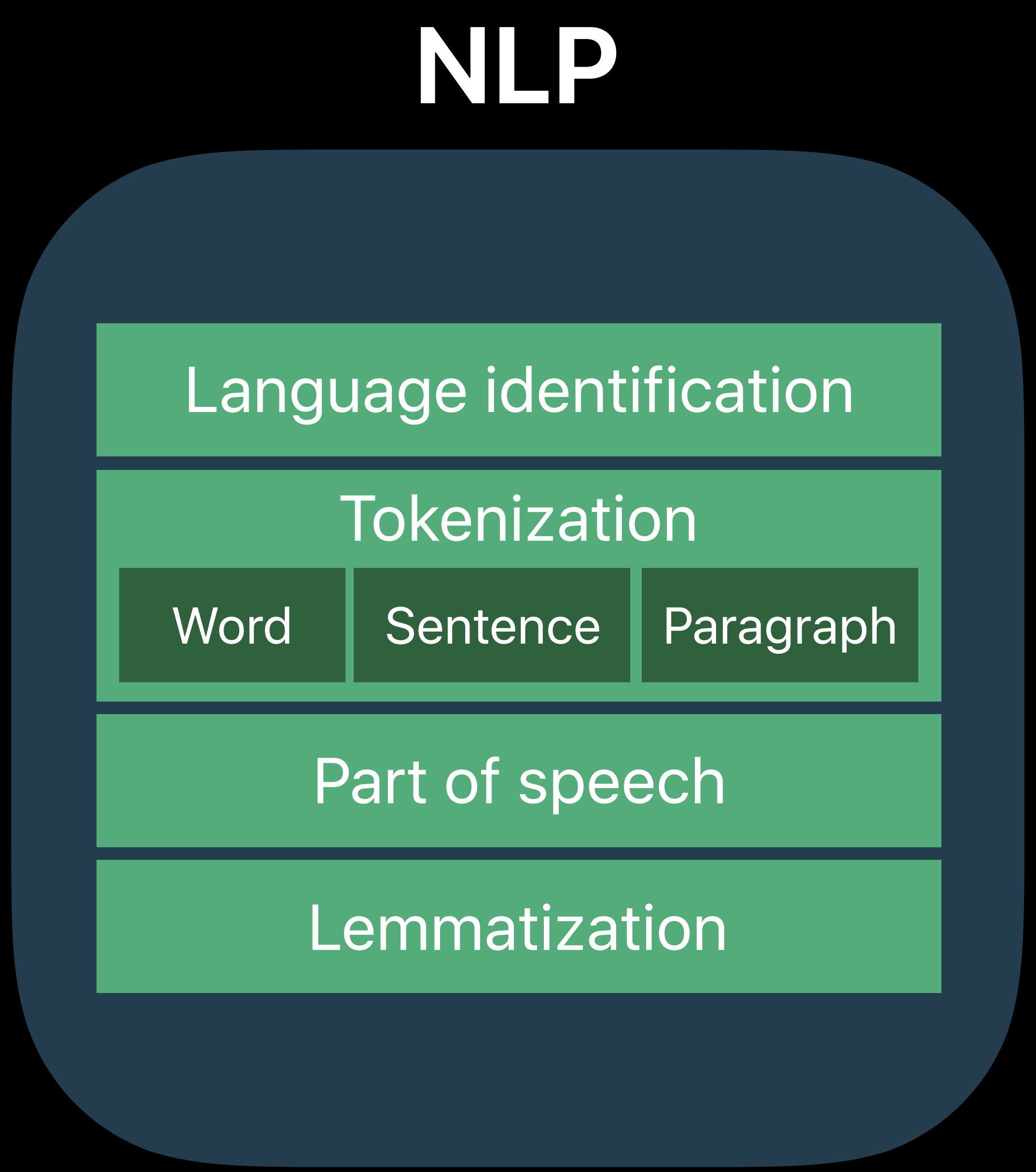
Winnow

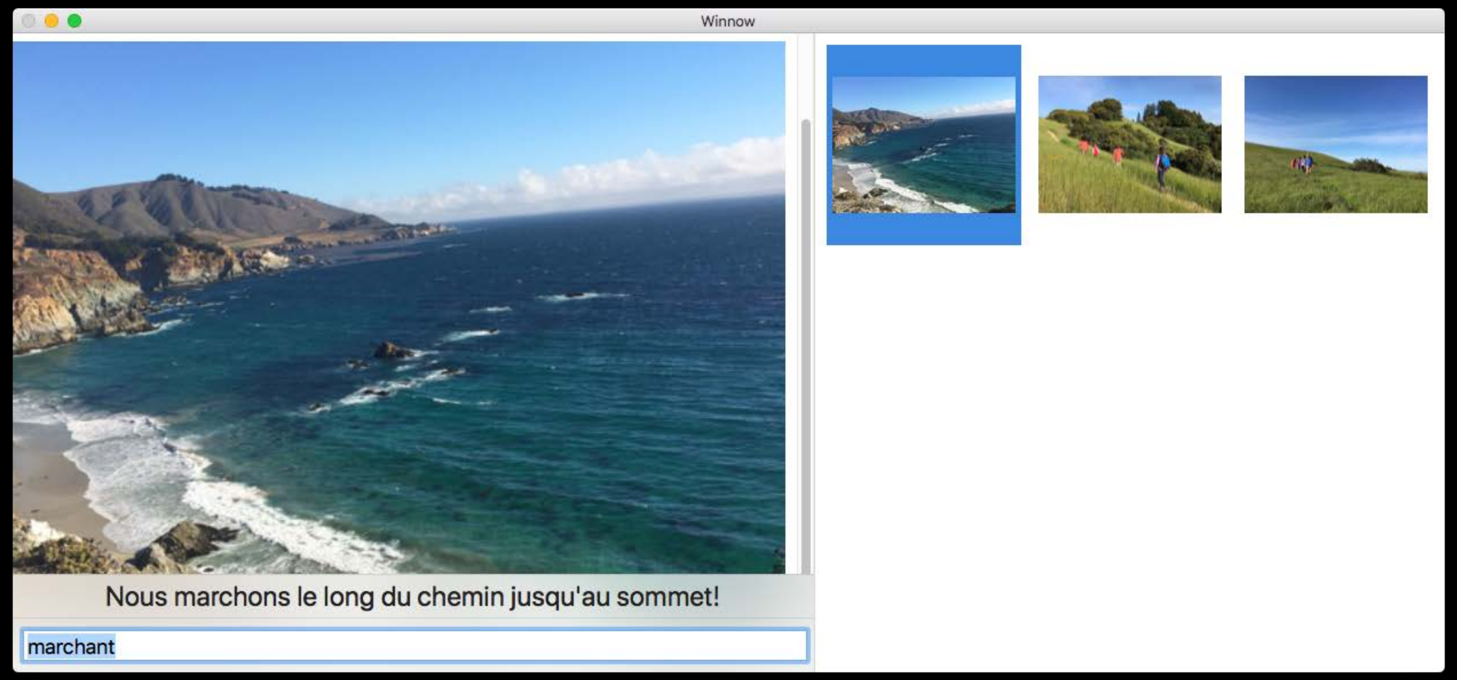

NLP
Language identification
Tokenization
Word
Sentence
Paragraph
Part of speech

Winnow


NLP
Language identification
Tokenization
Word
Sentence
Paragraph
Part of speech
Lemmatization

Winnow





Nous marchons le long du chemin jusqu'au sommet!

marchant

Language Identification

```
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.language], options: 0)

tagger.string = "Die Kleinen haben friedlich zusammen gespielt."

let language = tagger.dominantLanguage
```

Language Identification

```
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.language], options: 0)

tagger.string = "Die Kleinen haben friedlich zusammen gespielt."

let language = tagger.dominantLanguage
```

Language Identification

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.language], options: 0)

tagger.string = "Die Kleinen haben friedlich zusammen gespielt."

let language = tagger.dominantLanguage
```

Language Identification

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.language], options: 0)

tagger.string = "Die Kleinen haben friedlich zusammen gespielt."

let language = tagger.dominantLanguage
```

Language Identification

```
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.language], options: 0)

tagger.string = "Die Kleinen haben friedlich zusammen gespielt."

let language = tagger.dominantLanguage
```

Tokenization

```
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.tokenType], options: 0)

let text = "NSLinguisticTagger provides text processing APIs.\n NSLinguisticTagger 是苹果的文字处
理平台。"

tagger.string = text
let range = NSRange(location: 0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]


tagger.enumerateTags(in: range, unit: .word, scheme: .tokenType, options: options) {
tag, tokenRange, stop in
    let token = (text as NSString).substring(with: tokenRange)
    // Do something with each token
}
```

Tokenization

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.tokenType], options: 0)

let text = "NSLinguisticTagger provides text processing APIs.\n NSLinguisticTagger 是苹果的文字处
理平台。"

tagger.string = text
let range = NSRange(location: 0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]


tagger.enumerateTags(in: range, unit: .word, scheme: .tokenType, options: options) {
tag, tokenRange, stop in
    let token = (text as NSString).substring(with: tokenRange)
    // Do something with each token
}
```

Tokenization

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.tokenType], options: 0)

let text = "NSLinguisticTagger provides text processing APIs.\n NSLinguisticTagger 是苹果的文字处
理平台。"

tagger.string = text
let range = NSRange(location: 0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]


tagger.enumerateTags(in: range, unit: .word, scheme: .tokenType, options: options) {
tag, tokenRange, stop in
    let token = (text as NSString).substring(with: tokenRange)
    // Do something with each token
}
```

Tokenization

```
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.tokenType], options: 0)

let text = "NSLinguisticTagger provides text processing APIs.\n NSLinguisticTagger 是苹果的文字处
理平台。"

tagger.string = text
let range = NSRange(location: 0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]


tagger.enumerateTags(in: range, unit: .word, scheme: .tokenType, options: options) {
tag, tokenRange, stop in
    let token = (text as NSString).substring(with: tokenRange)
    // Do something with each token
}
```

Tokenization

```
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.tokenType], options: 0)

let text = "NSLinguisticTagger provides text processing APIs.\n NSLinguisticTagger 是苹果的文字处
理平台。"

tagger.string = text
let range = NSRange(location: 0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]


tagger.enumerateTags(in: range, unit: .word, scheme: .tokenType, options: options) {
tag, tokenRange, stop in
    let token = (text as NSString).substring(with: tokenRange)
    // Do something with each token
}
```

Tokenization

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.tokenType], options: 0)

let text = "NSLinguisticTagger provides text processing APIs.\n NSLinguisticTagger 是苹果的文字处
理平台。"

tagger.string = text
let range = NSRange(location: 0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]


tagger.enumerateTags(in: range, unit: .word, scheme: .tokenType, options: options) {
tag, tokenRange, stop in
    let token = (text as NSString).substring(with: tokenRange)
    // Do something with each token
}
```

Tokenization

```
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.tokenType], options: 0)

let text = "NSLinguisticTagger provides text processing APIs.\n NSLinguisticTagger 是苹果的文字处
理平台。"

tagger.string = text
let range = NSRange(location: 0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]


tagger.enumerateTags(in: range, unit: .word, scheme: .tokenType, options: options) {
tag, tokenRange, stop in
    let token = (text as NSString).substring(with: tokenRange)
    // Do something with each token
}
```

Lemmatization

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes:[.lemma], options: 0)

let text = "Great hikes make great pics! Wonderful afternoon in Marin County."

tagger.string = text
let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]

tagger.enumerateTags(in: range, unit: .word, scheme: .lemma, options: options) {
tag, tokenRange, stop in
    if let lemma = tag?.rawValue {
        // Do something with each lemma
    }
}
```

Lemmatization

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes:[.lemma], options: 0)

let text = "Great hikes make great pics! Wonderful afternoon in Marin County."

tagger.string = text
let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]

tagger.enumerateTags(in: range, unit: .word, scheme: .lemma, options: options) {
tag, tokenRange, stop in
    if let lemma = tag?.rawValue {
        // Do something with each lemma
    }
}
```

Lemmatization

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes:[.lemma], options: 0)

let text = "Great hikes make great pics! Wonderful afternoon in Marin County."

tagger.string = text
let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]

tagger.enumerateTags(in: range, unit: .word, scheme: .lemma, options: options) {
tag, tokenRange, stop in
    if let lemma = tag?.rawValue {
        // Do something with each lemma
    }
}
```

Lemmatization

```
import Foundation

let tagger = NSLinguisticTagger(tagSchemes:[.lemma], options: 0)

let text = "Great hikes make great pics! Wonderful afternoon in Marin County."

tagger.string = text
let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]

tagger.enumerateTags(in: range, unit: .word, scheme: .lemma, options: options) {
tag, tokenRange, stop in
    if let lemma = tag?.rawValue {
        // Do something with each lemma
    }
}
```

Lemmatization

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes:[.lemma], options: 0)

let text = "Great hikes make great pics! Wonderful afternoon in Marin County."

tagger.string = text
let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]

tagger.enumerateTags(in: range, unit: .word, scheme: .lemma, options: options) {
tag, tokenRange, stop in
    if let lemma = tag?.rawValue {
        // Do something with each lemma
    }
}
```

Lemmatization

```
import Foundation

let tagger = NSLinguisticTagger(tagSchemes:[.lemma], options: 0)

let text = "Great hikes make great pics! Wonderful afternoon in Marin County."

tagger.string = text
let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]

tagger.enumerateTags(in: range, unit: .word, scheme: .lemma, options: options) {
tag, tokenRange, stop in
    if let lemma = tag?.rawValue {
        // Do something with each lemma
    }
}
```

Lemmatization

```
import Foundation

let tagger = NSLinguisticTagger(tagSchemes:[.lemma], options: 0)

let text = "Great hikes make great pics! Wonderful afternoon in Marin County."

tagger.string = text
let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]

tagger.enumerateTags(in: range, unit: .word, scheme: .lemma, options: options) {
tag, tokenRange, stop in
    if let lemma = tag?.rawValue {
        // Do something with each lemma
    }
}
```

Lemmatization

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes:[.lemma], options: 0)

let text = "Great hikes make great pics! Wonderful afternoon in Marin County."

tagger.string = text
let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace]

tagger.enumerateTags(in: range, unit: .word, scheme: .lemma, options: options) {
tag, tokenRange, stop in
    if let lemma = tag?.rawValue {
        // Do something with each lemma
    }
}
```

Demo

Whisk

Collate social media feeds

Whisk

Collate social media feeds

Whisk

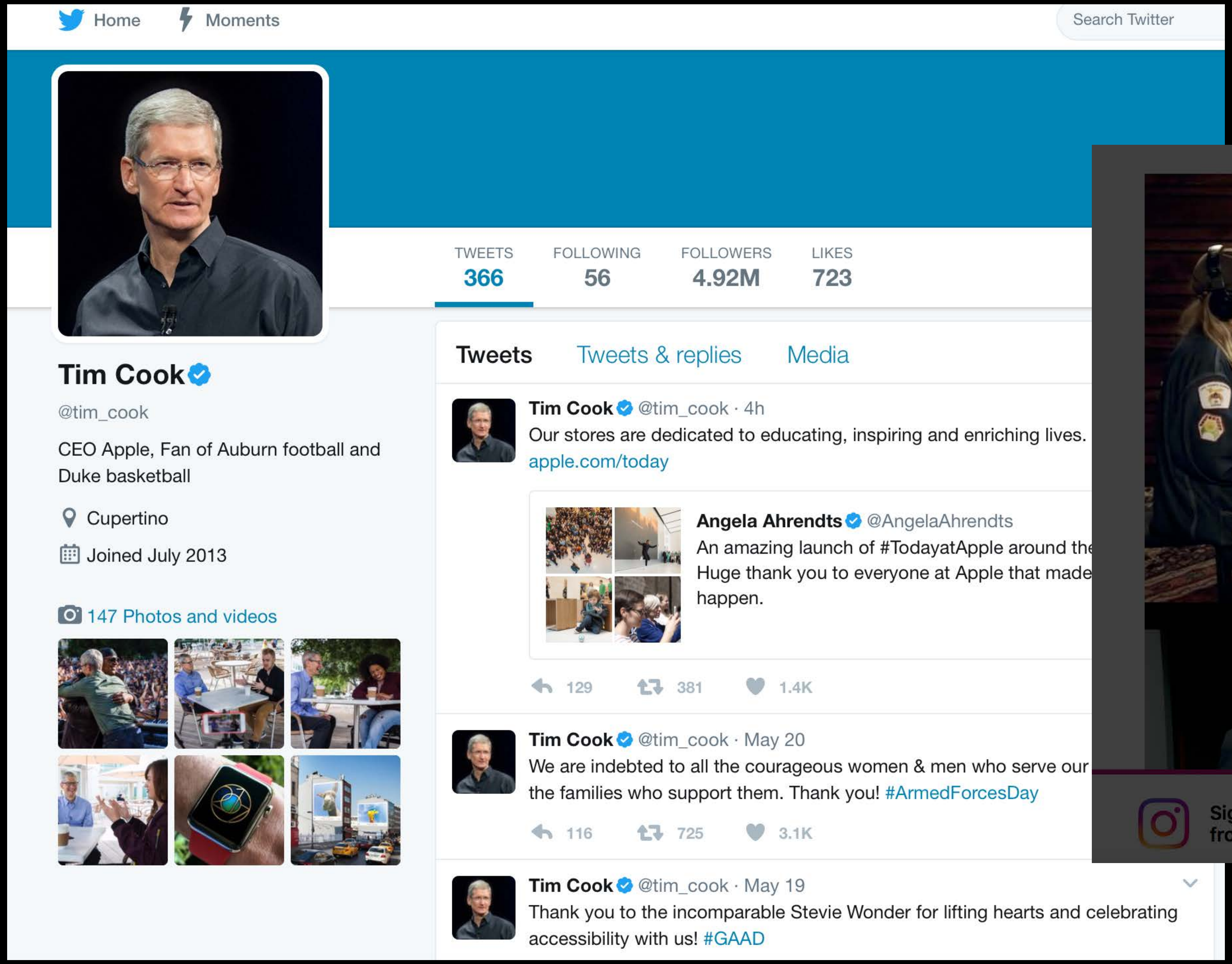
Collate social media feeds

Whisk

Organize feeds by People, Organization, and Location using NLP

Whisk

Organize feeds by People, Organization, and Location using NLP



Tim Cook
@tim_cook
CEO Apple, Fan of Auburn football and Duke basketball
Cupertino
Joined July 2013
147 Photos and videos

Tim Cook @tim_cook · 4h
Our stores are dedicated to educating, inspiring and enriching lives.
apple.com/today

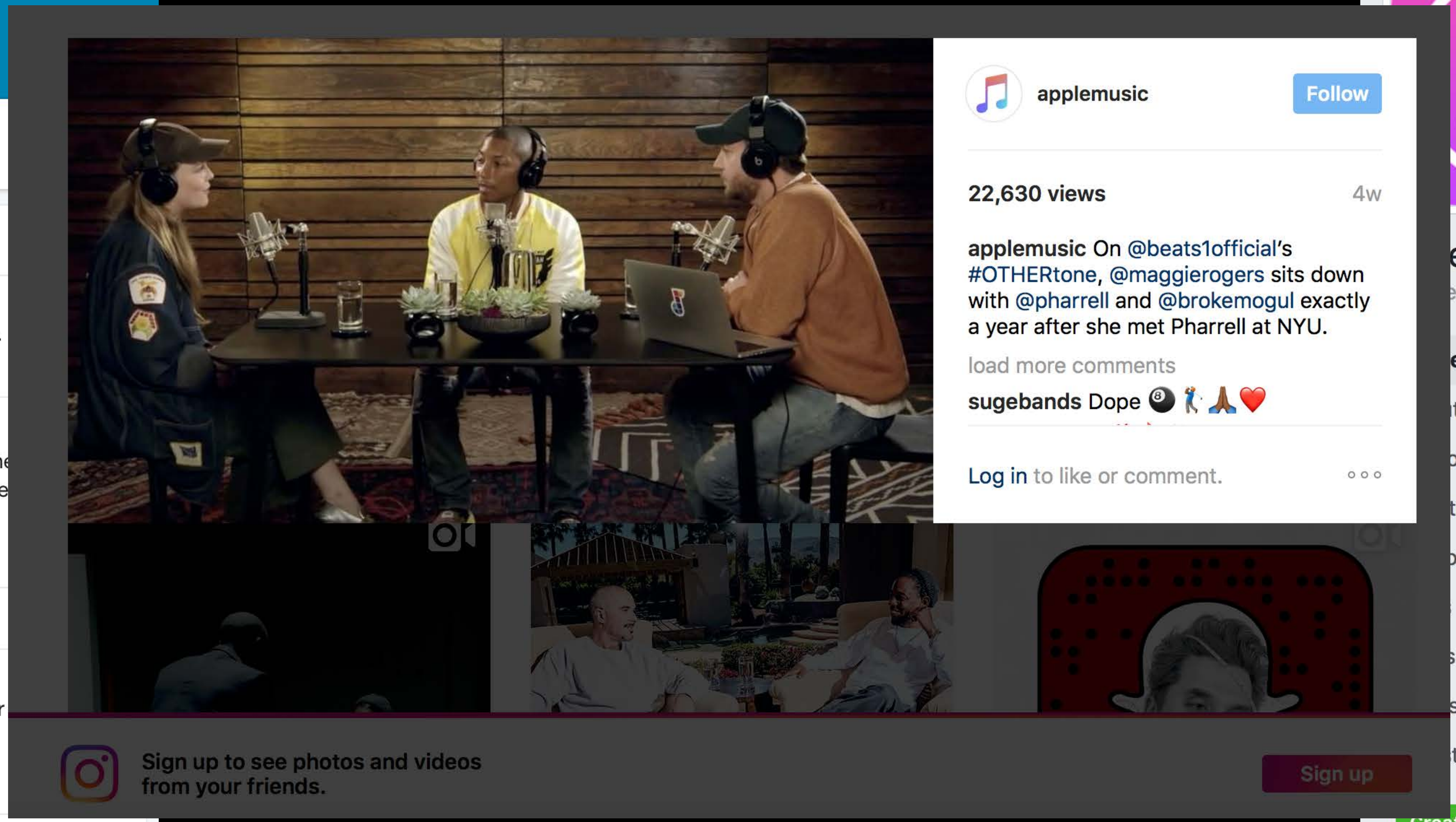
Angela Ahrendts @AngelaAhrendts
An amazing launch of #TodayatApple around the
Huge thank you to everyone at Apple that made
happen.

129 381 1.4K

Tim Cook @tim_cook · May 20
We are indebted to all the courageous women & men who serve our
the families who support them. Thank you! #ArmedForcesDay

116 725 3.1K

Tim Cook @tim_cook · May 19
Thank you to the incomparable Stevie Wonder for lifting hearts and celebrating accessibility with us! #GAAD



applemusic

22,630 views 4w

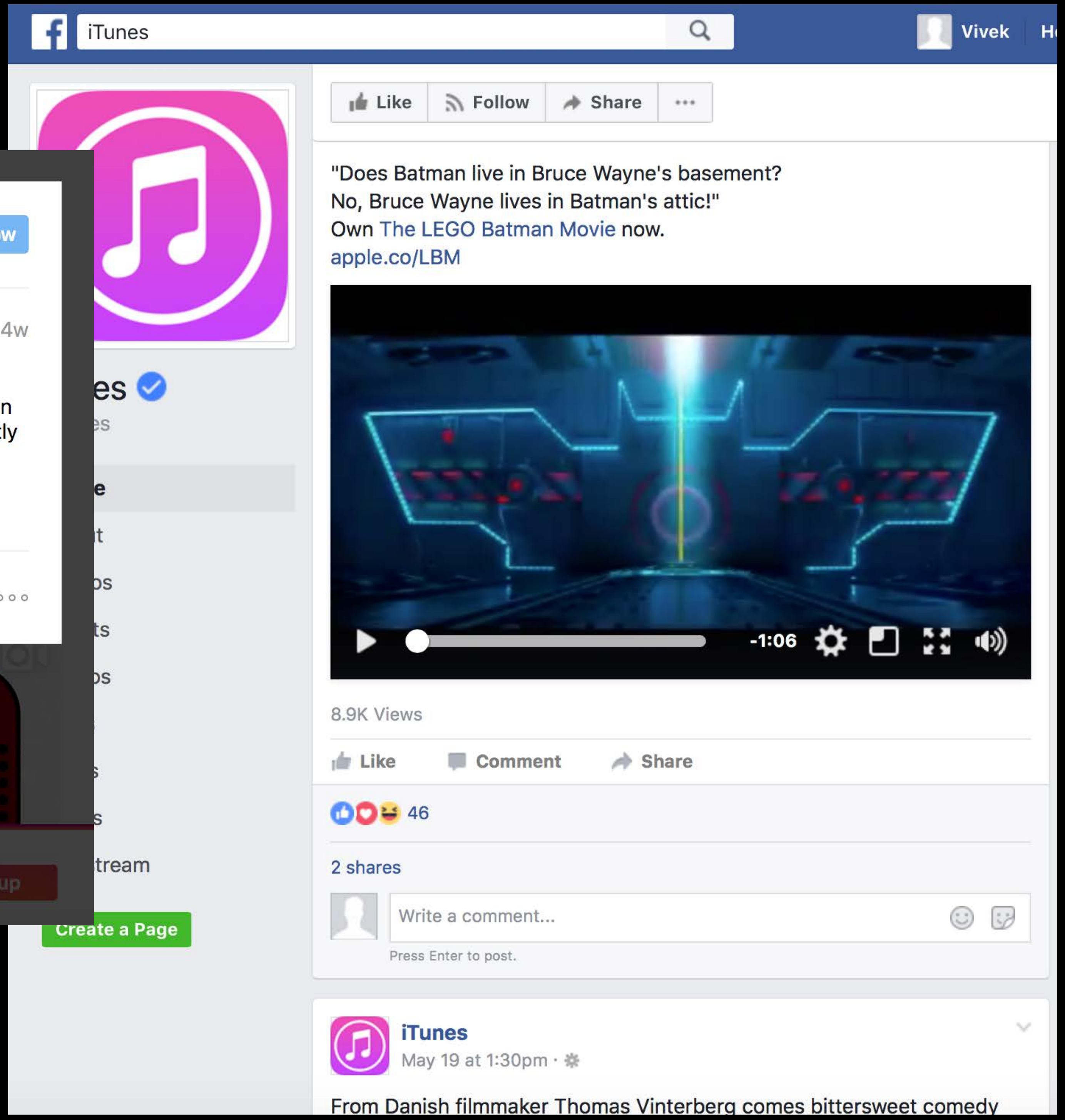
applemusic On @beats1official's #OTHERtone, @maggierogers sits down with @pharrell and @brokemogul exactly a year after she met Pharrell at NYU.

load more comments

sugebands Dope 🎱🏌🙏🏾❤️

Log in to like or comment.

Sign up to see photos and videos from your friends.



"Does Batman live in Bruce Wayne's basement?
No, Bruce Wayne lives in Batman's attic!"
Own The LEGO Batman Movie now.
apple.co/LBM

-1:06

8.9K Views

46

2 shares

Write a comment...

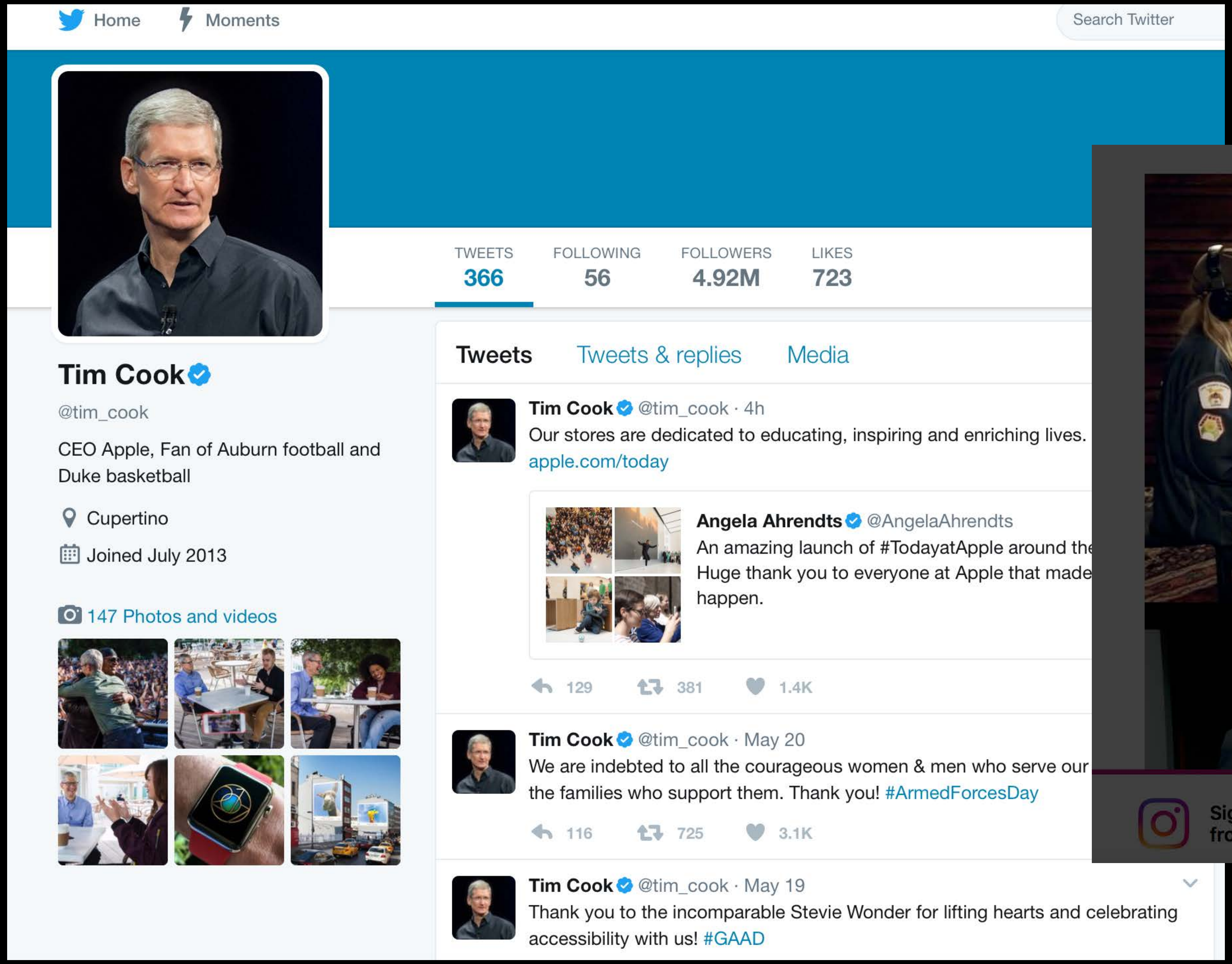
Press Enter to post.

iTunes
May 19 at 1:30pm ·

From Danish filmmaker Thomas Vinterberg comes bittersweet comedy

Whisk

Organize feeds by People, Organization, and Location using NLP



Home Moments Search Twitter

TWEETS 366 FOLLOWING 56 FOLLOWERS 4.92M LIKES 723

Tim Cook
@tim_cook
CEO Apple, Fan of Auburn football and Duke basketball
Cupertino
Joined July 2013
147 Photos and videos

Tweets Tweets & replies Media

Tim Cook @tim_cook · 4h
Our stores are dedicated to educating, inspiring and enriching lives.
apple.com/today

Angela Ahrendts @AngelaAhrendts
An amazing launch of #TodayatApple around the
Huge thank you to everyone at Apple that made
happen.

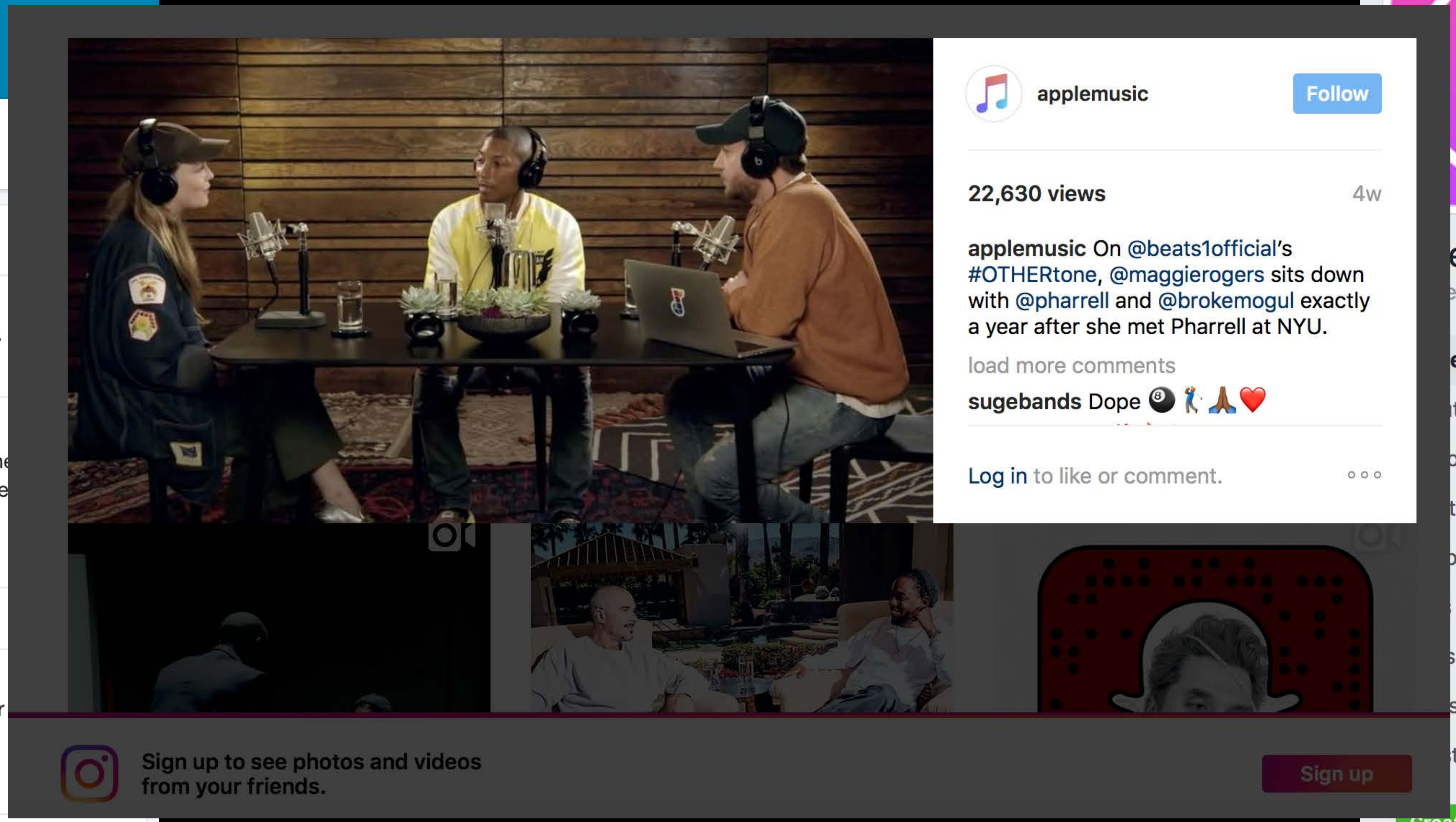
129 381 1.4K

Tim Cook @tim_cook · May 20
We are indebted to all the courageous women & men who serve our
the families who support them. Thank you! #ArmedForcesDay

116 725 3.1K

Tim Cook @tim_cook · May 19
Thank you to the incomparable Stevie Wonder for lifting hearts and celebrating accessibility with us! #GAAD

Tim Cook, Apple, Stevie Wonder, Cupertino, Angela Ahrendts



applemusic Follow

22,630 views 4w

applemusic On @beats1official's #OTHERtone, @maggierogers sits down with @pharrell and @brokemogul exactly a year after she met Pharrell at NYU.

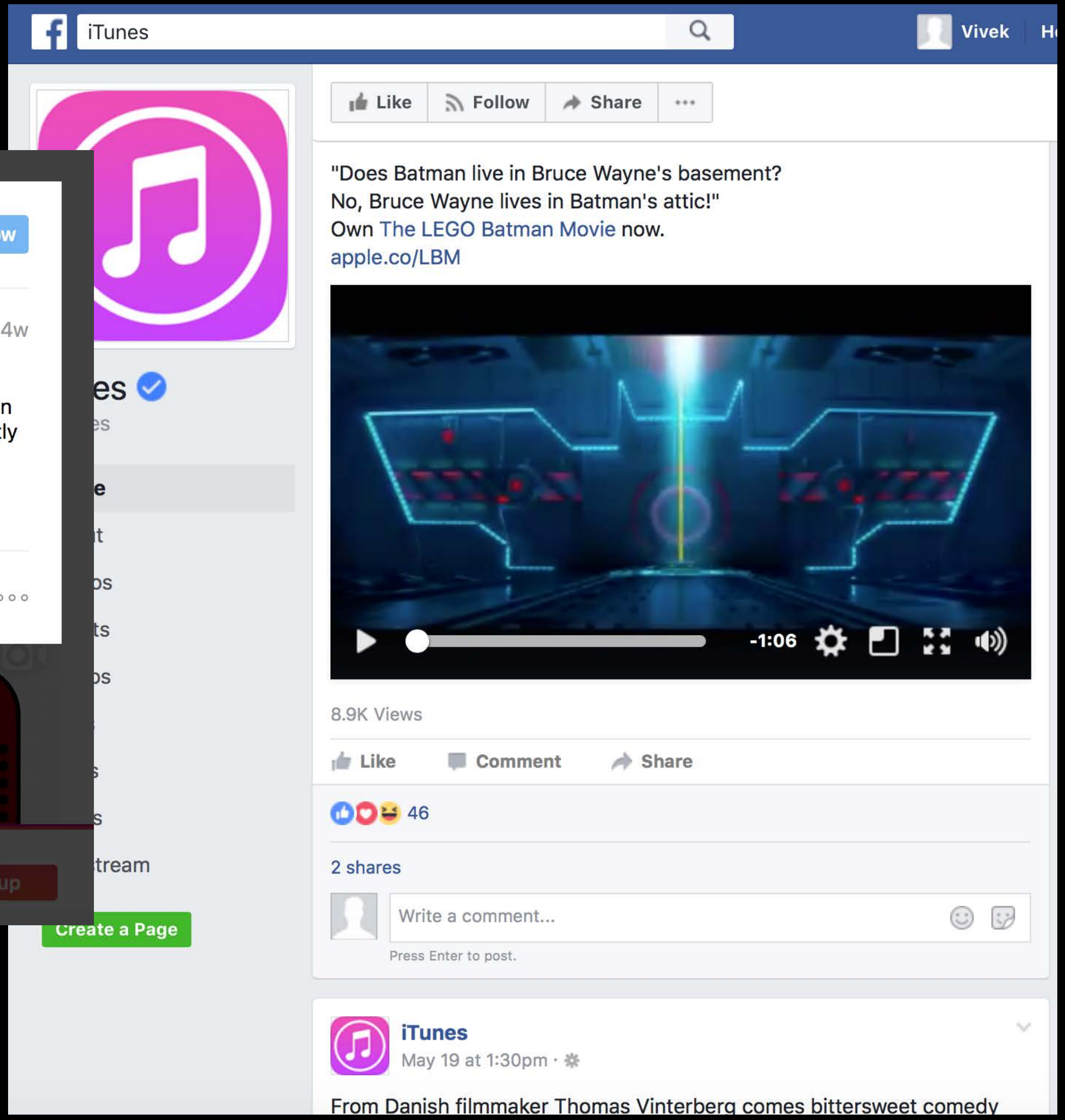
load more comments

sugebands Dope 🎱🕺🙏❤️

Log in to like or comment.

Sign up to see photos and videos from your friends. Sign up

Pharrell, NYU



iTunes Vivek

Like Follow Share

"Does Batman live in Bruce Wayne's basement?
No, Bruce Wayne lives in Batman's attic!"
Own The LEGO Batman Movie now.
apple.co/LBM

-1:06

8.9K Views

Like Comment Share

46

2 shares

Write a comment...

Press Enter to post.

Create a Page

iTunes
May 19 at 1:30pm

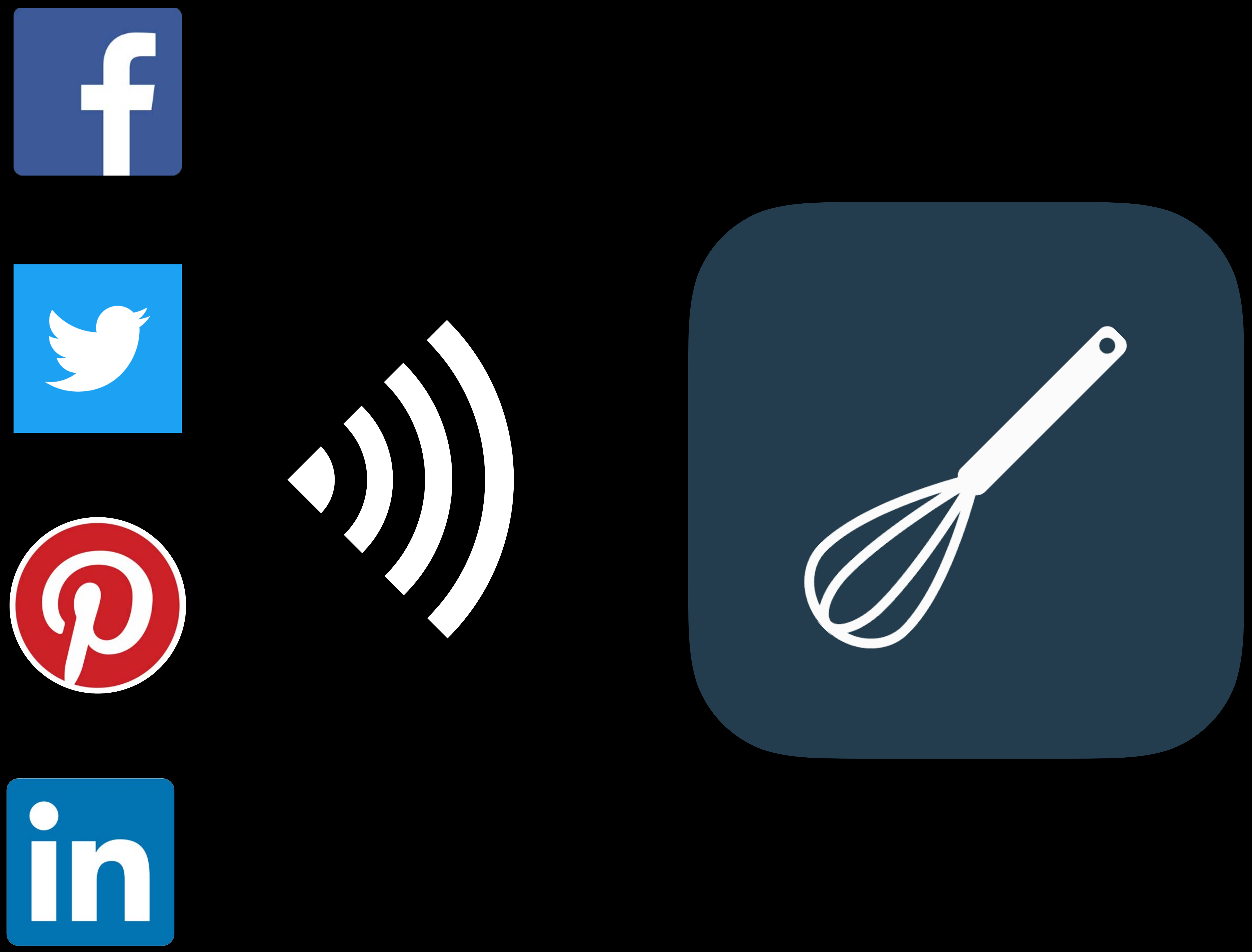
From Danish filmmaker Thomas Vinterberg comes bittersweet comedy

Bruce Wayne, Thomas Vintberg

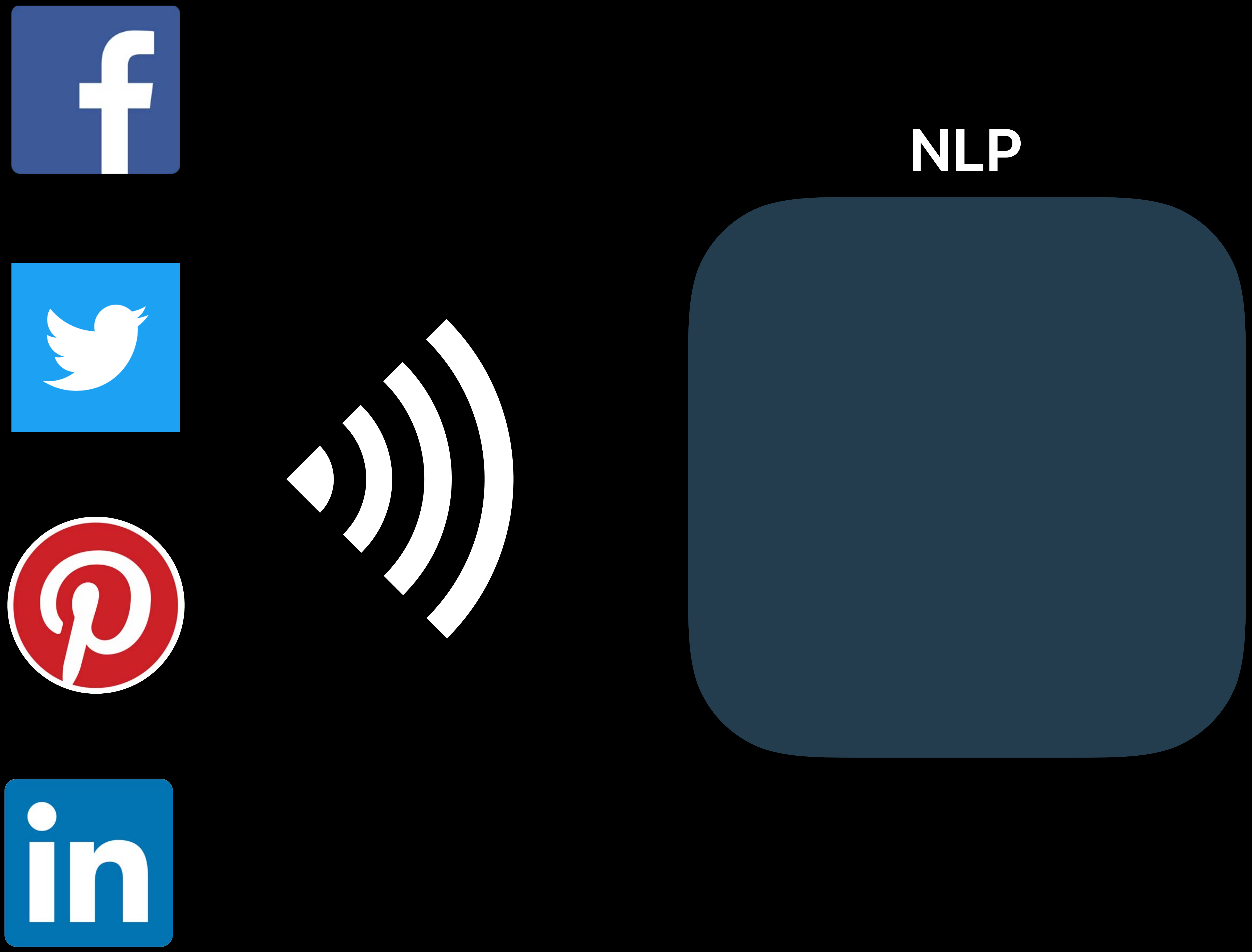
Whisk

Whisk

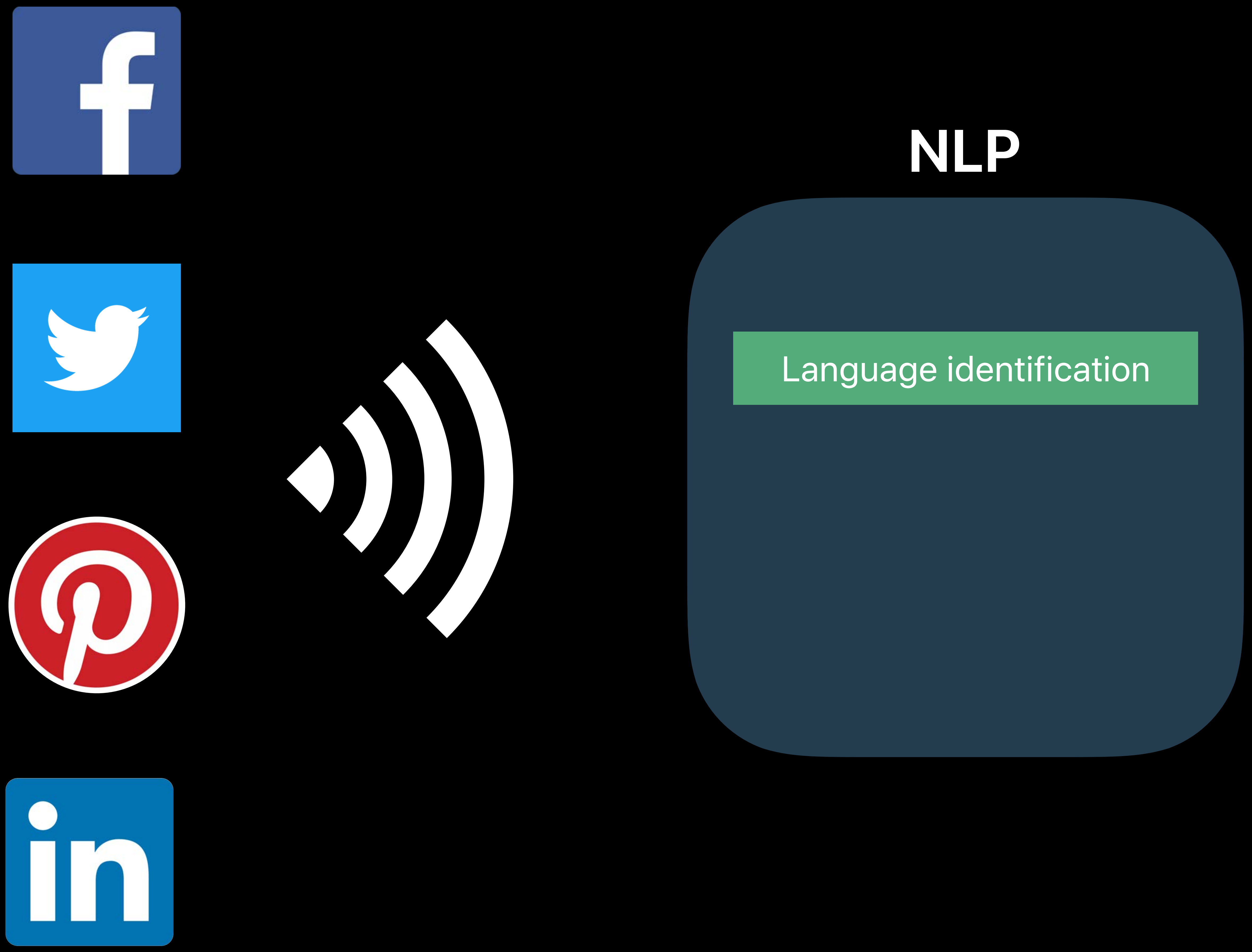
Whisk



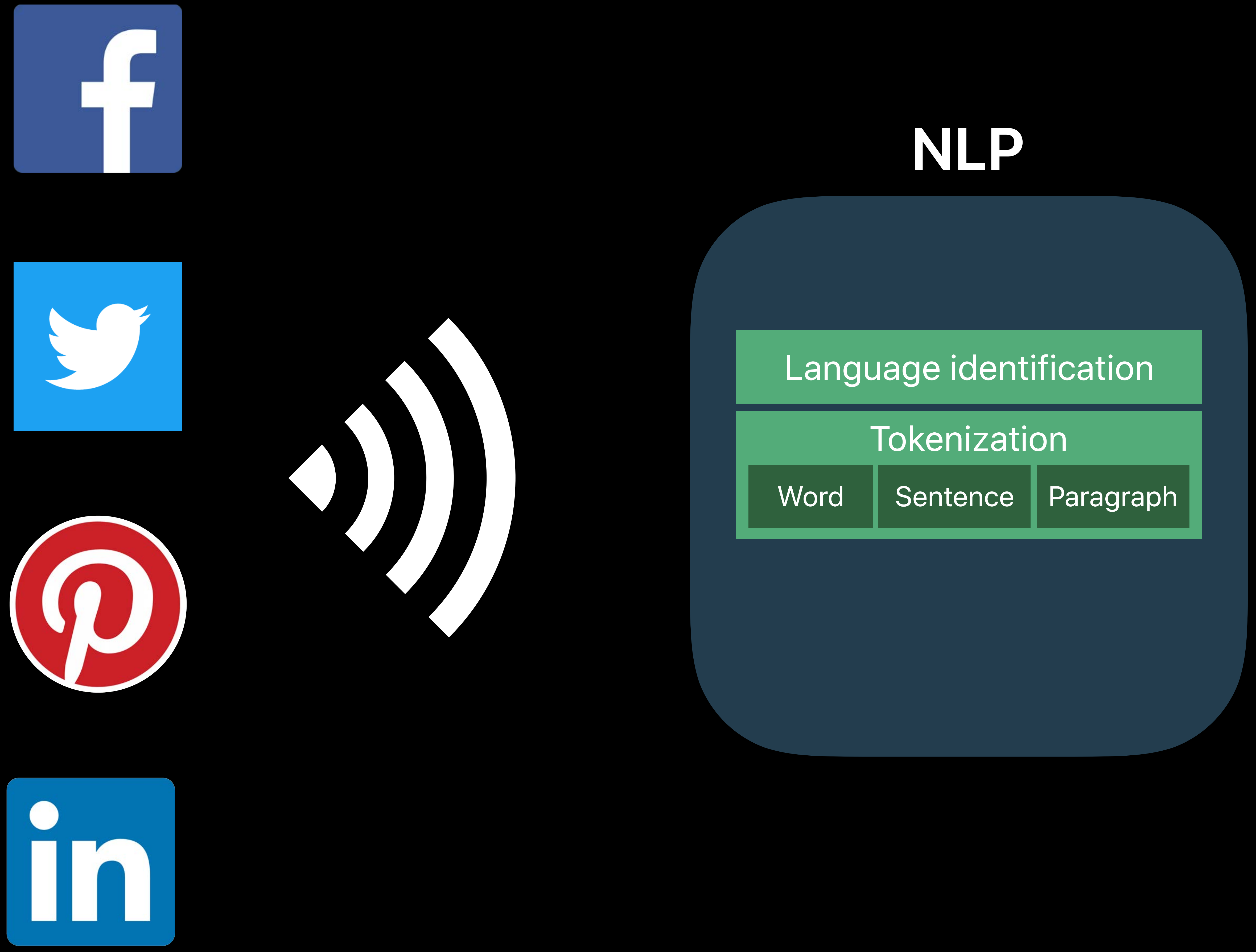
Whisk



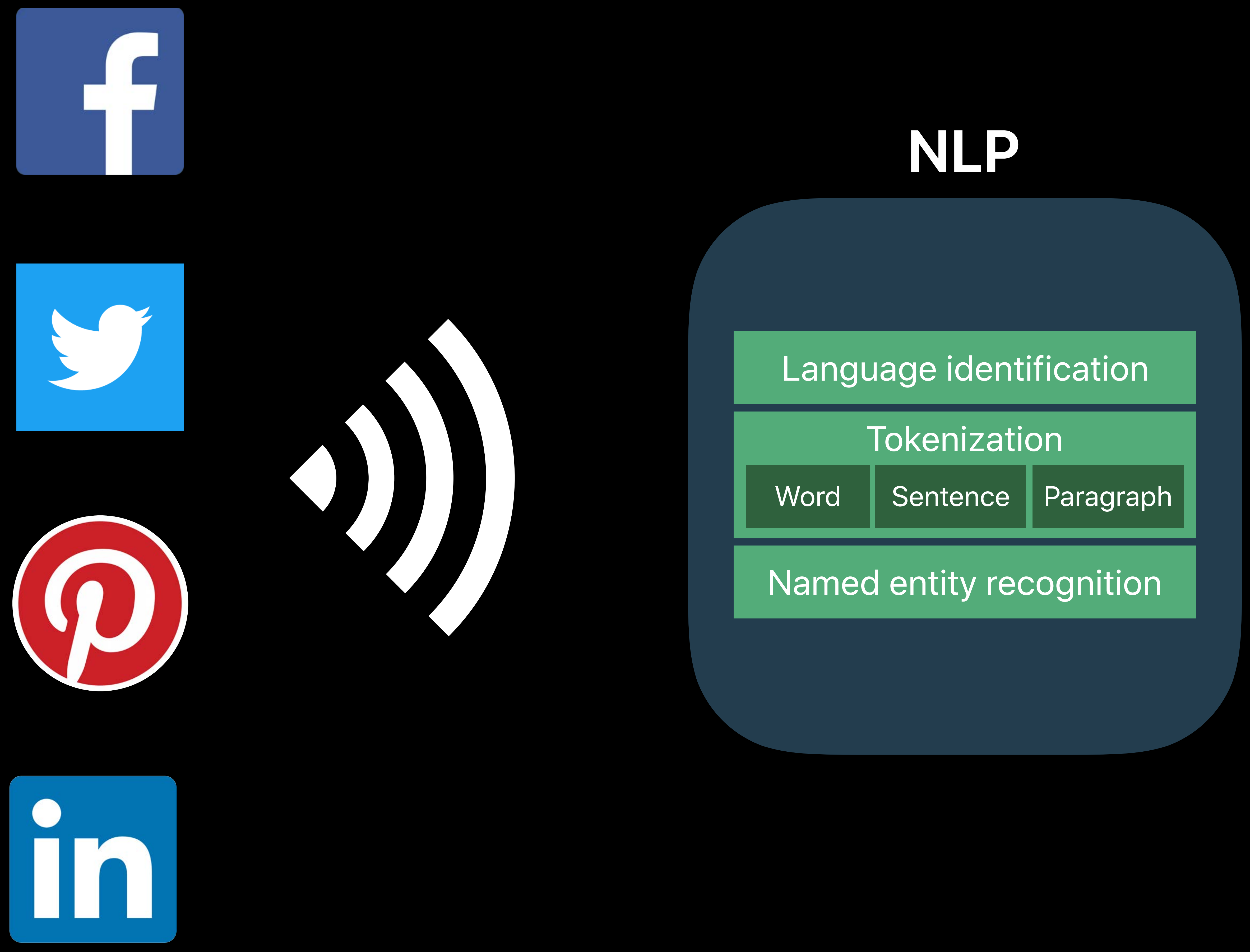
Whisk



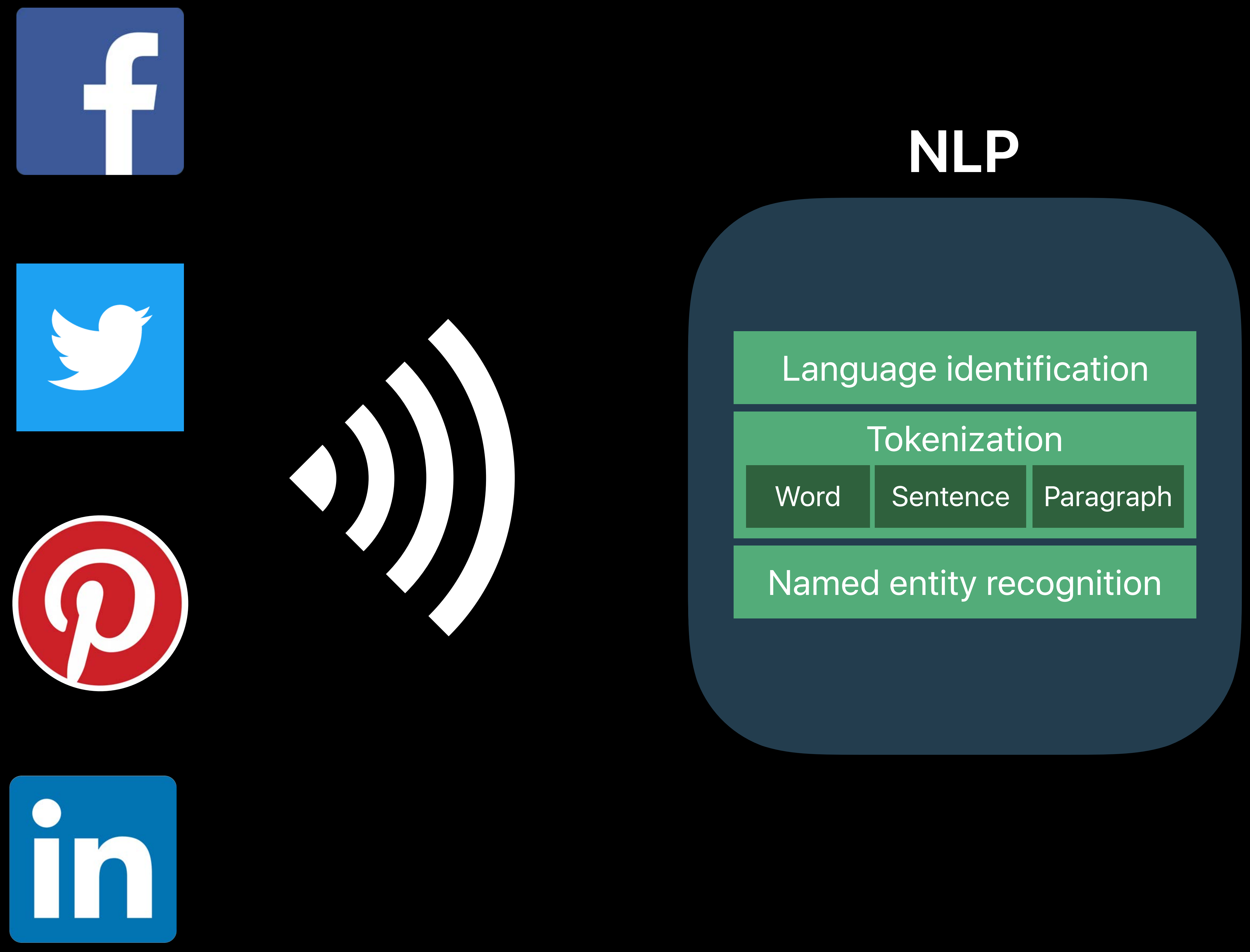
Whisk

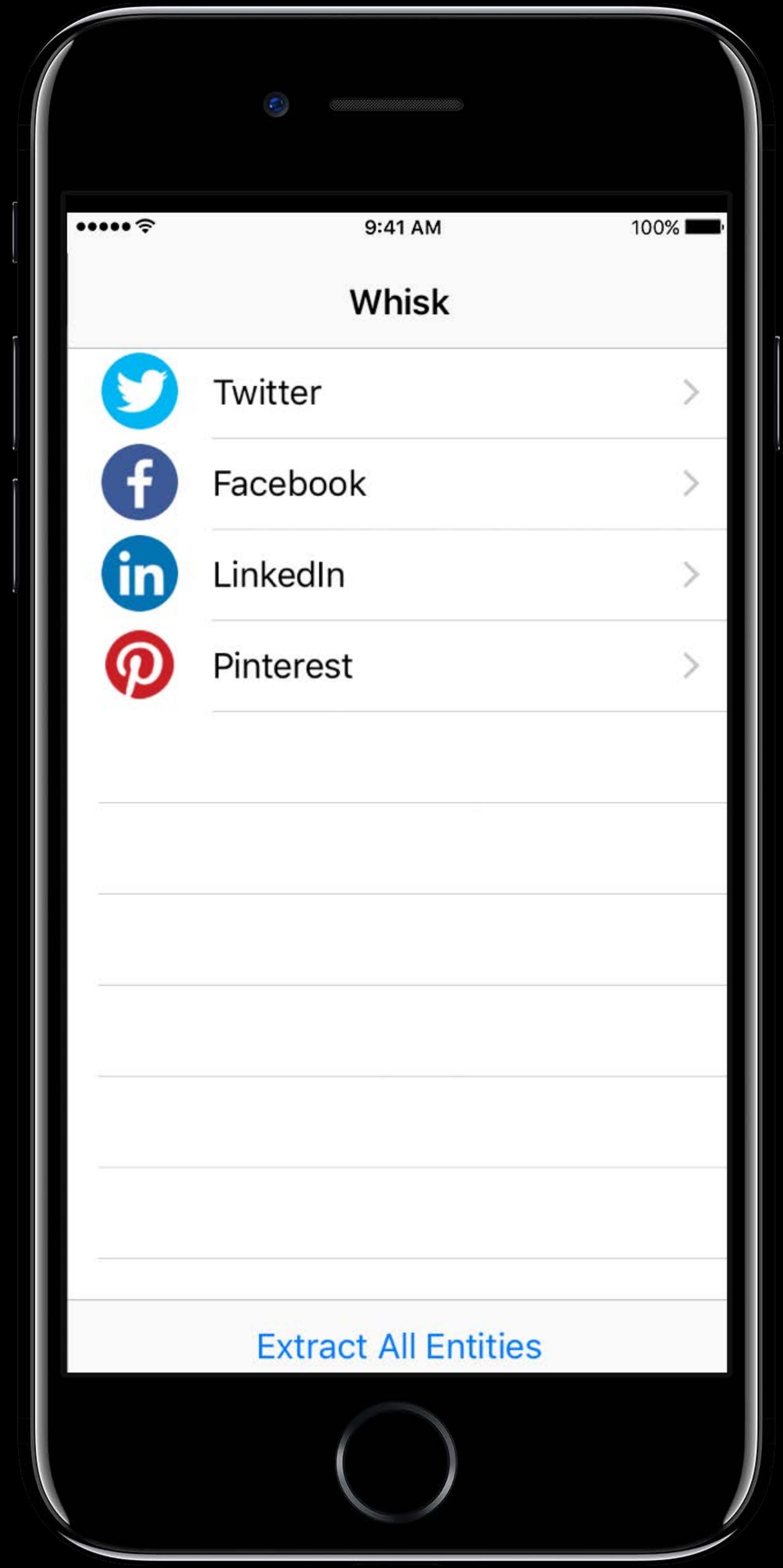

NLP
Language identification
Tokenization
Word
Sentence
Paragraph

Whisk


NLP
Language identification
Tokenization
Word
Sentence
Paragraph
Named entity recognition

Whisk


NLP
Language identification
Tokenization
Word
Sentence
Paragraph
Named entity recognition


9:41 AM
100%
Whisk
Twitter
Facebook
LinkedIn
Pinterest
Extract All Entities

Named Entity Recognition

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.nameType], options: 0)

let text = "Tim Cook is the CEO of Apple Inc. which is located in Cupertino, California"

tagger.string = text

let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace, .joinNames]
let tags: [NSLinguisticTag] = [.personalName, .placeName, .organizationName]

tagger.enumerateTags(in: range, unit: .word, scheme: .nameType, options: options) {
tag, tokenRange, stop in
    if let tag = tag, tags.contains(tag) {
        let name = (text as NSString).substring(with: tokenRange)
   }
}
```

Named Entity Recognition

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.nameType], options: 0)

let text = "Tim Cook is the CEO of Apple Inc. which is located in Cupertino, California"

tagger.string = text

let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace, .joinNames]
let tags: [NSLinguisticTag] = [.personalName, .placeName, .organizationName]

tagger.enumerateTags(in: range, unit: .word, scheme: .nameType, options: options) {
tag, tokenRange, stop in
    if let tag = tag, tags.contains(tag) {
        let name = (text as NSString).substring(with: tokenRange)
   }
}
```

Named Entity Recognition

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.nameType], options: 0)

let text = "Tim Cook is the CEO of Apple Inc. which is located in Cupertino, California"

tagger.string = text

let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace, .joinNames]
let tags: [NSLinguisticTag] = [.personalName, .placeName, .organizationName]

tagger.enumerateTags(in: range, unit: .word, scheme: .nameType, options: options) {
tag, tokenRange, stop in
    if let tag = tag, tags.contains(tag) {
        let name = (text as NSString).substring(with: tokenRange)
   }
}
```

Named Entity Recognition

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.nameType], options: 0)

let text = "Tim Cook is the CEO of Apple Inc. which is located in Cupertino, California"

tagger.string = text

let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace, .joinNames]
let tags: [NSLinguisticTag] = [.personalName, .placeName, .organizationName]

tagger.enumerateTags(in: range, unit: .word, scheme: .nameType, options: options) {
tag, tokenRange, stop in
    if let tag = tag, tags.contains(tag) {
        let name = (text as NSString).substring(with: tokenRange)
   }
}
```

Named Entity Recognition

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.nameType], options: 0)

let text = "Tim Cook is the CEO of Apple Inc. which is located in Cupertino, California"

tagger.string = text

let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace, .joinNames]
let tags: [NSLinguisticTag] = [.personalName, .placeName, .organizationName]

tagger.enumerateTags(in: range, unit: .word, scheme: .nameType, options: options) {
tag, tokenRange, stop in
    if let tag = tag, tags.contains(tag) {
        let name = (text as NSString).substring(with: tokenRange)
   }
}
```

Named Entity Recognition

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.nameType], options: 0)

let text = "Tim Cook is the CEO of Apple Inc. which is located in Cupertino, California"

tagger.string = text

let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace, .joinNames]
let tags: [NSLinguisticTag] = [.personalName, .placeName, .organizationName]

tagger.enumerateTags(in: range, unit: .word, scheme: .nameType, options: options) {
tag, tokenRange, stop in
    if let tag = tag, tags.contains(tag) {
        let name = (text as NSString).substring(with: tokenRange)
   }
}
```

Named Entity Recognition

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.nameType], options: 0)

let text = "Tim Cook is the CEO of Apple Inc. which is located in Cupertino, California"

tagger.string = text

let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace, .joinNames]
let tags: [NSLinguisticTag] = [.personalName, .placeName, .organizationName]

tagger.enumerateTags(in: range, unit: .word, scheme: .nameType, options: options) {
tag, tokenRange, stop in
    if let tag = tag, tags.contains(tag) {
        let name = (text as NSString).substring(with: tokenRange)
   }
}
```

Named Entity Recognition

```
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.nameType], options: 0)

let text = "Tim Cook is the CEO of Apple Inc. which is located in Cupertino, California"

tagger.string = text

let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace, .joinNames]
let tags: [NSLinguisticTag] = [.personalName, .placeName, .organizationName]

tagger.enumerateTags(in: range, unit: .word, scheme: .nameType, options: options) {
tag, tokenRange, stop in
    if let tag = tag, tags.contains(tag) {
        let name = (text as NSString).substring(with: tokenRange)
    }
}
```

Named Entity Recognition

```swift
import Foundation

let tagger = NSLinguisticTagger(tagSchemes: [.nameType], options: 0)

let text = "Tim Cook is the CEO of Apple Inc. which is located in Cupertino, California"

tagger.string = text

let range = NSRange(location:0, length: text.utf16.count)
let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace, .joinNames]
let tags: [NSLinguisticTag] = [.personalName, .placeName, .organizationName]

tagger.enumerateTags(in: range, unit: .word, scheme: .nameType, options: options) {
tag, tokenRange, stop in
    if let tag = tag, tags.contains(tag) {
        let name = (text as NSString).substring(with: tokenRange)
   }
}
```

Demo

Why should you use the NLP APIs?

Homogeneous Text Processing

Homogeneous Text Processing

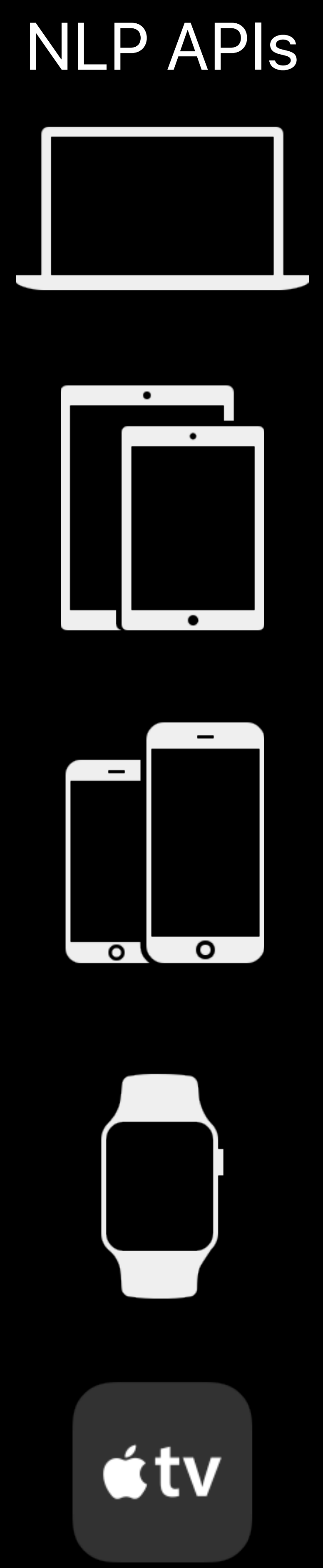
Natural language text

typed

recognized handwriting

transcribed speech

→



→

Consistent text processing

Consistent user experience

Privacy

Privacy

On-device machine learning

User data stays on-device

Performance

Performance

Highly optimized on-device

Multi-threaded

Existing clients—significant speedup

Performance

Highly optimized on-device

Multi-threaded

Existing clients—significant speedup

Chinese tokenization is 30% faster on iOS

Performance

Highly optimized on-device

Multi-threaded

Existing clients—significant speedup

Named Entity Recognition 80% faster on iOS

Performance

Performance

Part of Speech Tagging

●—● **50000**

tokens/sec

●—●
●—● **80000**

tokens/sec

●—● 1 thread

Performance

Part of Speech Tagging	Named Entity Recognition
— **50000** tokens/sec	— **40000** tokens/sec
= **80000** tokens/sec	= **65000** tokens/sec

— 1 thread

Language Support

Language Support

Language identification	29 scripts, 52 languages

Language Support

Language identification	29 scripts, 52 languages
Tokenization	All iOS/macOS system languages

Language Support

Language identification	29 scripts, 52 languages
Tokenization	All iOS/macOS system languages
Lemmatization	English French Italian German Spanish Portuguese Russian Turkish

Language Support

Language identification	29 scripts, 52 languages
Tokenization	All iOS/macOS system languages
Lemmatization	English French Italian German Spanish Portuguese Russian Turkish
Part of speech	

Language Support

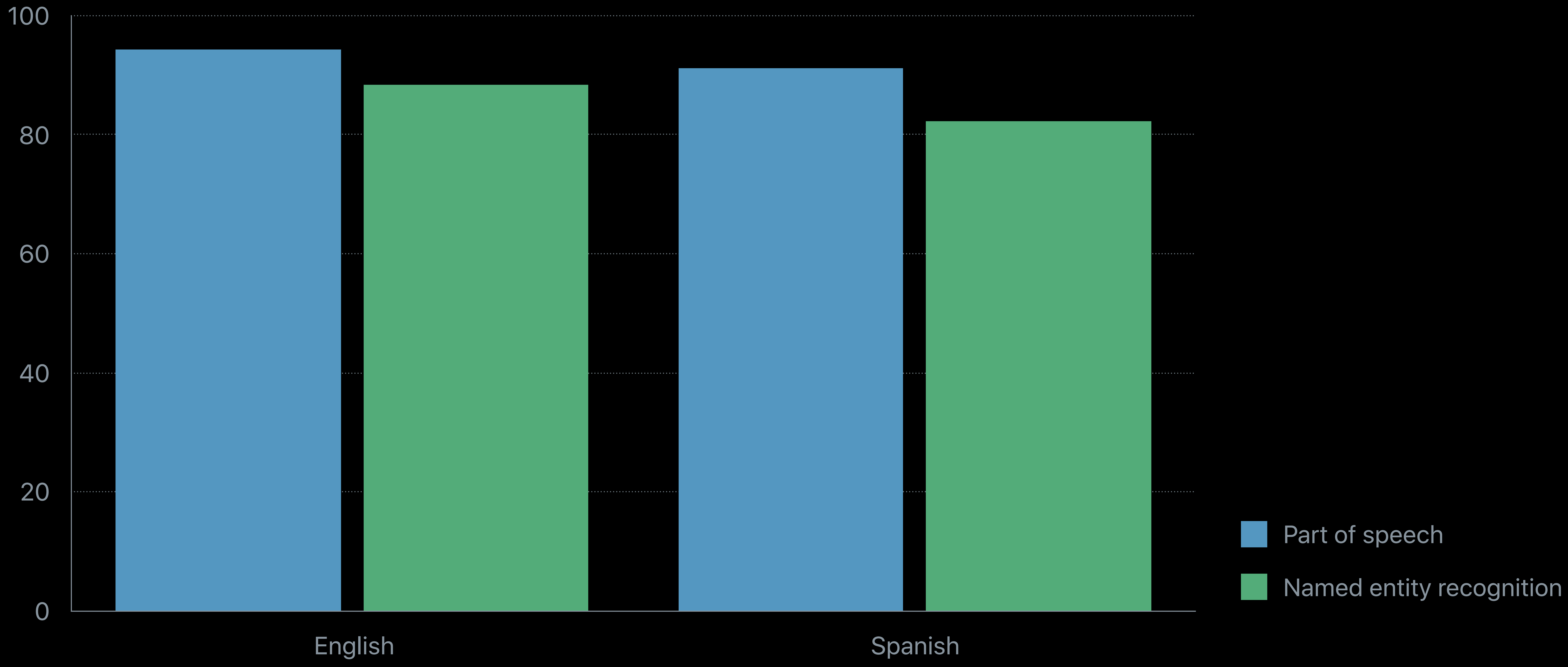
Language identification	29 scripts, 52 languages
Tokenization	All iOS/macOS system languages
Lemmatization	English French Italian German Spanish Portuguese Russian Turkish
Part of speech	
Named entity recognition	

Language Support

Language identification	29 scripts, 52 languages
Tokenization	All iOS/macOS system languages
Lemmatization	English, French, Italian, German, Spanish, Portuguese, Russian, Turkish
Part of speech	English, French, Italian, German, Spanish, Portuguese, Russian, Turkish
Named entity recognition	English, French, Italian, German, Spanish, Portuguese, Russian, Turkish

Accuracy

Accuracy


100
80
60
40
20
0
English
Spanish
Part of speech
Named entity recognition

Debugging Hints

Debugging Hints

Tags are NSLinguisticTagOtherWord

- Model not downloaded

Debugging Hints

Tags are NSLinguisticTagOtherWord

- Model not downloaded

Downloaded over-the-air

Debugging Hints

Tags are NSLinguisticTagOtherWord

- Model not downloaded

Downloaded over-the-air

Explicitly set language if known

Summary

NLP APIs—NSLinguisticTagger

- Support for new units
- Faster
- Higher accuracy
- More languages

More Information

https://developer.apple.com/wwdc17/208

Related Sessions

What's New in Cocoa Touch		WWDC 2017
Introducing Core ML		WWDC 2017
Vision Framework: Building on Core ML	Hall 2	Wednesday 3:10PM
Core ML in Depth	Hall 3	Thursday 9:00 AM
Accelerate and Sparse Solvers	Grand Ballroom A	Thursday 10:00 AM

Labs

Core ML & Natural Language Processing Lab	Technology Lab D	Thu 11:00AM-3:30PM
Vision Lab	Technology Lab A	Fri 1:50PM-4:00PM
Core ML & Natural Language Processing Lab	Technology Lab D	Fri 1:50PM-4:00PM
Cocoa Lab	Technology Lab B	Fri 1:50PM-3:20PM

WWDC17