ボイス・アシスタント・システム

機械と人間のインターフェースの変化に備えよう



丸山不二夫

はじめに

- AmazonのEcho/Alexaが、ついに日本上陸を果たした。それを迎え討つべくDoCoMo、Line等の日本勢の動きも、開発者レベルでは慌ただしい。これはグローバルな競争の一環だ。すでに、昨年5月に開催されたAmazon Code Conference で、Jeff Bezosは、「ボイス・アシスタントの熱狂的な流行は、すぐ、そこまで来ている。」として、壇上から、Apple, Google, Amazon製品の「戦争」が迫っているとほのめかしていた。http://goo.gl/rR5BqP
- ボイス・アシスタント・システムは、自動運転カーとともに、AI技術がコンシューマーに普及する上で、最初の2大プロダクトの一つになろうとしている。この最初のコンシューマー市場とそのプラットフォームを、誰がどう押さえるかが、AIビジネスの今後の展開に大きな影響力を持つ可能性がある。

Amazon Code Conference 2016/05/31



Jeff Bezos は、壇上から、Apple, Google, Amazon製品の戦争が迫っているとほのめかした http://goo.gl/rR5BqP

はじめに

- 本講座は、IT技術者が Amazon Echo/Alexa, Google Home/Assistant, Microsoft Cortana, Apple Siriといった ボイス・アシスタント技術の登場をどのように捉えればいいのか、また、今後それらをどのように活用すべきかを実践的に考えることを目的としている。
- まず、第一部で、ボイス・アシスタントシステムの現状と課題を概観しようと思う。第二部では、ボイス・アシスタント・システムの実装とプログラミングを、Echo Alexaを中心に見る。

ボイス・アシスタント・システムの現状と課題



ボイス・アシスタント・システムのプロダクトを見る

ここでは、主要なボイス・アシスタントのプロダクトを見ておこう。アプライアンスとして提供されるものが目につくが、スマホ・PCで稼働するもの、SDK/APIを提供するものなど、プロダクトの裾野は広い。

Amazon Alexa



Amazon Alexa Show





2017年9月1日、ベルリンで行われたIFA（国際コンシューマー・エレクトロニクス展）で、Google Assistant を搭載した多数の Smart Speakerが発表された。

Sony



Panasonic



JBL



Anker



https://goo.gl/mwzexD




HomePod


Apple-designed
A8 chip
High-excursion
woofer
Six-microphone
array
Seamless mesh
fabric
Seven-tweeter array



Clova Smart Speaker WAVE

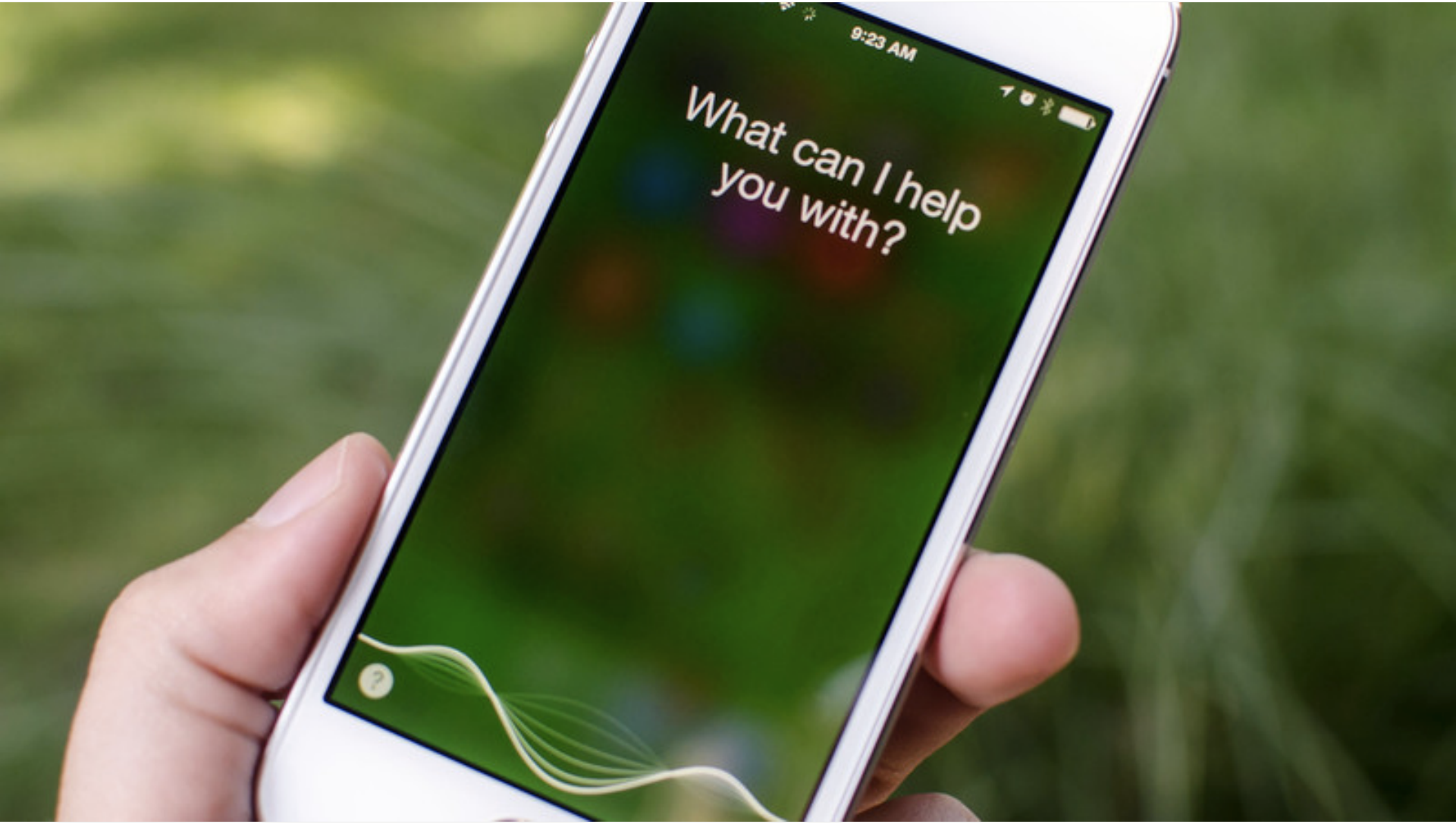
Clova INSIDE



Google Now / Google Assistant



Apple Siri



Facebook Messengers

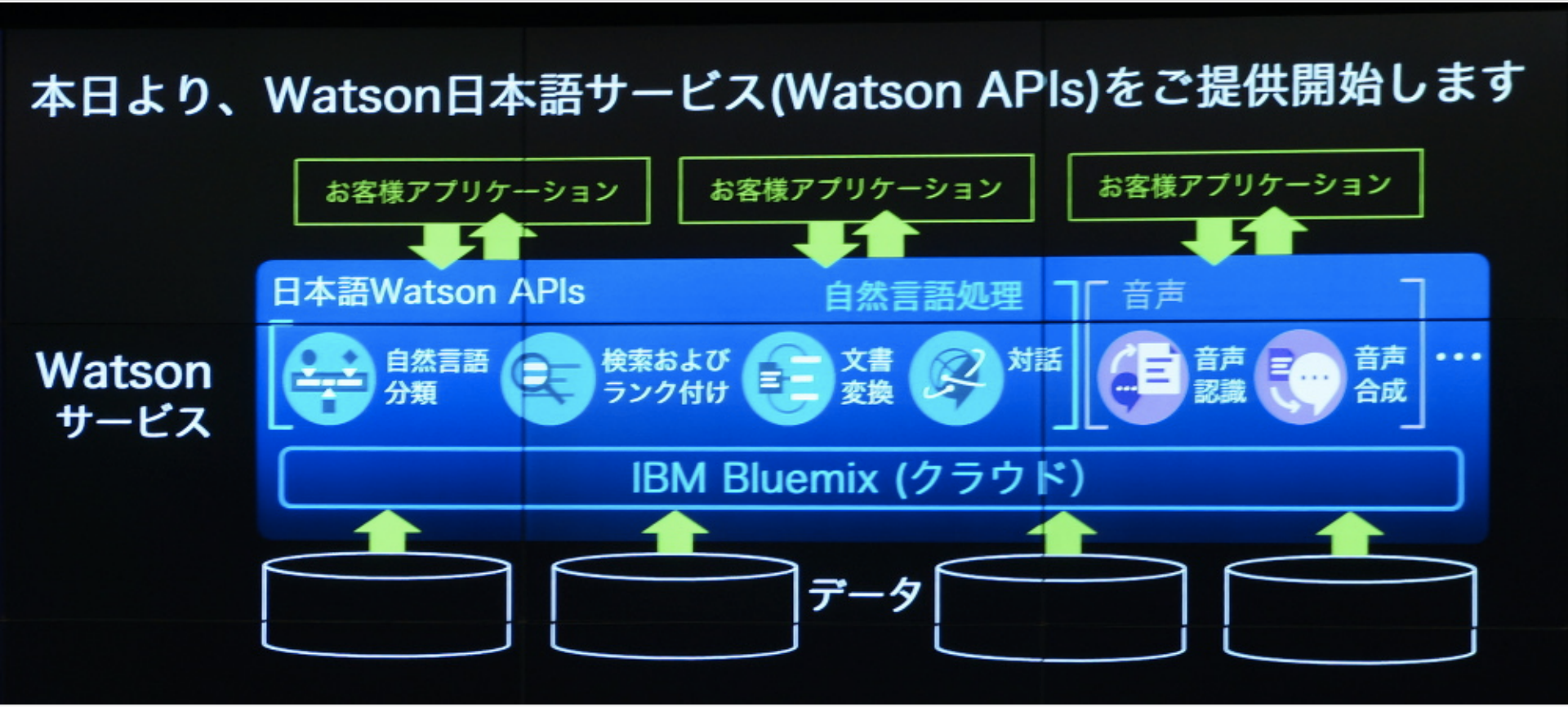


Bots
for Messenger

Microsoft Cortana



IBM Watson API



http://jp.techcrunch.com/2016/02/18/localized-watson-api/

質問応答のシナリオに適用できるサービスの組み合わせ

IBM



© Copyright IBM Corporation 2016

https://goo.gl/bZ7GBH

API名称（略称）	分野	機能	概要
Natural Language Classifier（NLC）	言語	自然言語分類	自然言語を理解し、学習に基づき適切な「意図」を返答
Retrieve & Rank（R&R）		検索＆ランクつけ	機械学習アルゴリズムを利用した検索エンジン
Document Conversion（DoC）		文書変換	テキスト文書へのフォーマット変換
Dialog（DLG）		対話	アプリケーションとエンドユーザーが対話する会話応答システム
Text To Speech（TTS）	音声	音声合成	テキストの音声変換
Speech To Text（STT）		音声認識	音声のテキスト変換

https://goo.gl/g865FM

NTT Voice-Series



音声メディア処理技術 を活用した
応用アプリケーション/サービス による 音声ソリューション

Voice-Series (ボイス・シリーズ)

高度で使いやすい音声処理応用技術を開発・提供

NTT研究所による長年の研究開発に裏付けられた音声認識技術や音声合成技術をベースに、
お客様のニーズに添ったソリューションをご提案いたします。

Voice-Series サイト内検索 検索

シリーズラインアップ
Series Line Up

VoiceMall

IVR ソリューション

高機能なクラウド型音声（自動応答）サービス
お客様ニーズの高い業務をパッケージ化

SpeechRec

音声認識ソリューション

高精度な音声認識エンジン
さまざまな用途への適用が可能

FutureVoice

音声合成ソリューション

業界最高クラスの高品質音声合成
話者と口調を自由に組み合わせできる音声合成

VoiceMall **SpeechRec** **FutureVoice**



https://www.ntt-tx.co.jp/products/voicemall/



製品紹介
Product Introduction

多様なニーズに応える高精度な音声認識ソリューションをご提供します。

Introduction 1
SpeechRec とは？
高精度音声認識ソリューション「SpeechRec（スピーチレック）」をご紹介します。

Introduction 2
製品ラインアップ
製品ラインアップをご紹介します。

Introduction 3
音声認識のしくみ
音声認識のしくみを詳しくご紹介します。

Introduction 4
活用事例
「SpeechRec」の活用事例を

Introduction 5
費用例

Introduction 6
よくあるご質問
るご質問

https://www.ntt-tx.co.jp/products/speechrec/

FutureVoice

音声合成(TTS)ソリューション

FutureVoice

業界最高クラスの高品質音声から、
さまざまな表現を実現するバリエーション音声合成まで、
次世代の音声合成ソリューション

製品紹介
Product Introduction



業界最高クラスの高品質音声合成
ナレーターに迫る肉声感

FutureVoice **EX**



多彩なバリエーションと豊かな表現
組み合わせパターンは自由自在

FutureVoice **Crayon**

Introduction 1

製品ラインアップ

製品ラインアップと仕様をご紹介します。

Introduction 2

音声合成のしくみ

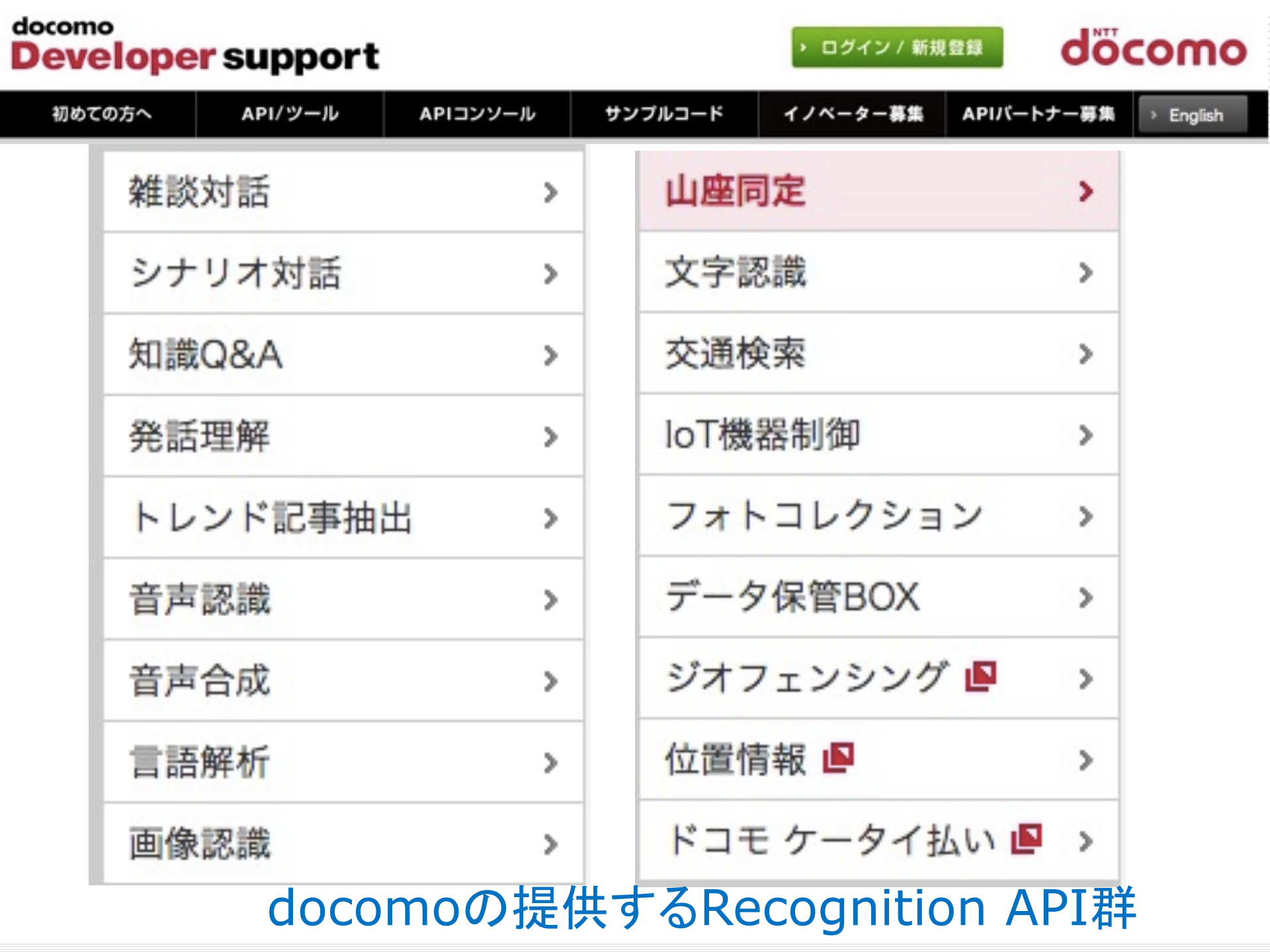
音声合成のしくみを詳しくご紹介します。

Introduction 3

活用事例

音声合成ソリューションの活用事例をご紹介します。

https://www.ntt-tx.co.jp/products/futurevoice/

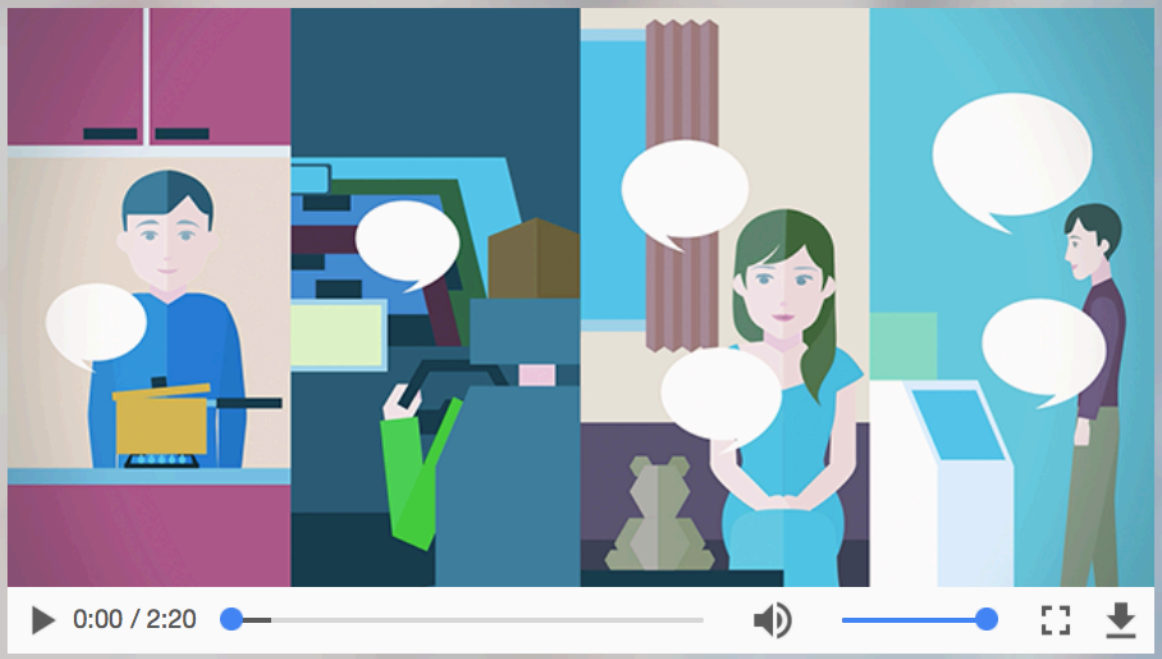


docomoの提供するRecognition API群

あらゆるモノに 対話型AIサービスを提供したい

PROJECT: SEBASTIEN

自然な対話を通じて、会話が出来たり、 コンテンツを提供したり、デバイスを操作したり
まるであらゆるモノが意思を持ち、対話ができる。 そんな新しい体験を提供します。



人工知能技術の到達点とボイス・アシスタント技術

音声認識技術への
ディープラーニング技術の応用と成功

2012年に起きたこと

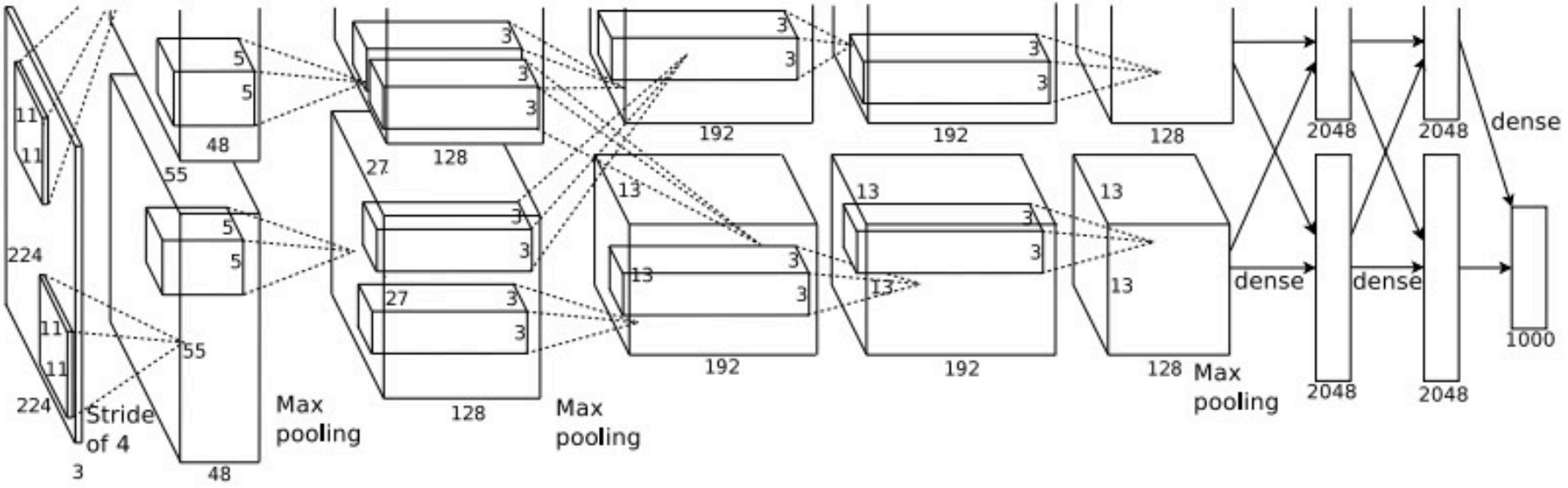
現在の人工知能に対する関心の高まりは、2012年に始まる。
この年、

1. Googleの16,000コアの巨大なシステム DistBeliefが、教師なしでネコと人間の顔を学習するのに成功した。
2. また、ImageNetの画像認識コンテストで、Convolutional Neural Network (CNN)のAlexNetが、圧勝した。
3. さらに、Tront大+Microsoft+Google+IBM 四者の共同研究チームが、音声認識にディープラーニングを用いた手法を公開。従来の方式を、はるかに上回る精度を達成した。

こうして、ニューラル・ネットワーク上のディープラーニングという手法は、現代のAI技術の中核の座を占めるに至った。



AlexNet


224
224
3
11
11
Stride
of 4
55
55
48
48
5
5
27
27
128
128
Max
pooling
Max
pooling
3
3
3
3
3
3
192
192
128
13
13
13
13
13
13
192
192
128
Max
pooling
dense
dense
dense
2048
2048
2048
2048
1000

“Deep Neural Networks for Acoustic Modeling in Speech Recognition”
Geoffrey Hinton et al.

ディープラーニングを用いた
音声認識技術の飛躍

https://goo.gl/GT6Bg9

2012年

Hinton, Google, IBM, Microsoft の共同研究



task	hours of training data	DNN-HMM	GMM-HMM with same data	GMM-HMM with more data
Switchboard (test set 1)	309	18.5	27.4	18.6 (2000 hrs)
Switchboard (test set 2)	309	16.1	23.6	17.1 (2000 hrs)
English Broadcast News	50	17.5	18.8	
Bing Voice Search (Sentence error rates)	24	30.4	36.2	
Google Voice Input	5,870	12.3		16.0 (>>5,870hrs)
Youtube	1,400	47.6	52.3	

Deep Neural Networks (DNN)

Gaussian mixture models (GMM)
hidden Markov models (HMMs)

2015/12/14

NTT持株会社ニュースリリース

（報道発表資料）

2015年12月14日

公共エリア雑音下でのモバイル音声認識の国際技術評価で、世界1位の精度を達成
～ひずみなし音声強調とディープラーニング新技術により音声認識を高精度化～

日本電信電話株式会社（本社：東京都千代田区、代表取締役社長：鵜浦博夫、以下　NTT）は、公共エリア（バス内・カフェ・商店街・市街地）の騒がしい環境でモバイル端末を用いて音声認識を行う技術評価国際イベントCHiME-3（The 3rd CHiME Speech Separation and Recognition Challenge）※1で、参加25機関中トップの精度を達成しました。

これは、NTTが独自開発したひずみなし音声強調技術※2とディープラーニング（深層学習）に基づく音声認識の新技術を用いることで、従来の限界を超える精度を実現したものです。本成果は、2015年12月13日（米国時間）より開催の国際会議The 2015 IEEE Automatic Speech Recognition and Understanding Workshop（ASRU 2015）※3にて発表されます。

https://goo.gl/7FQxD9

公共エリア雑音下でのモバイル音声認識の国際技術評価で、世界1位の精度を達成

2015年12月14日

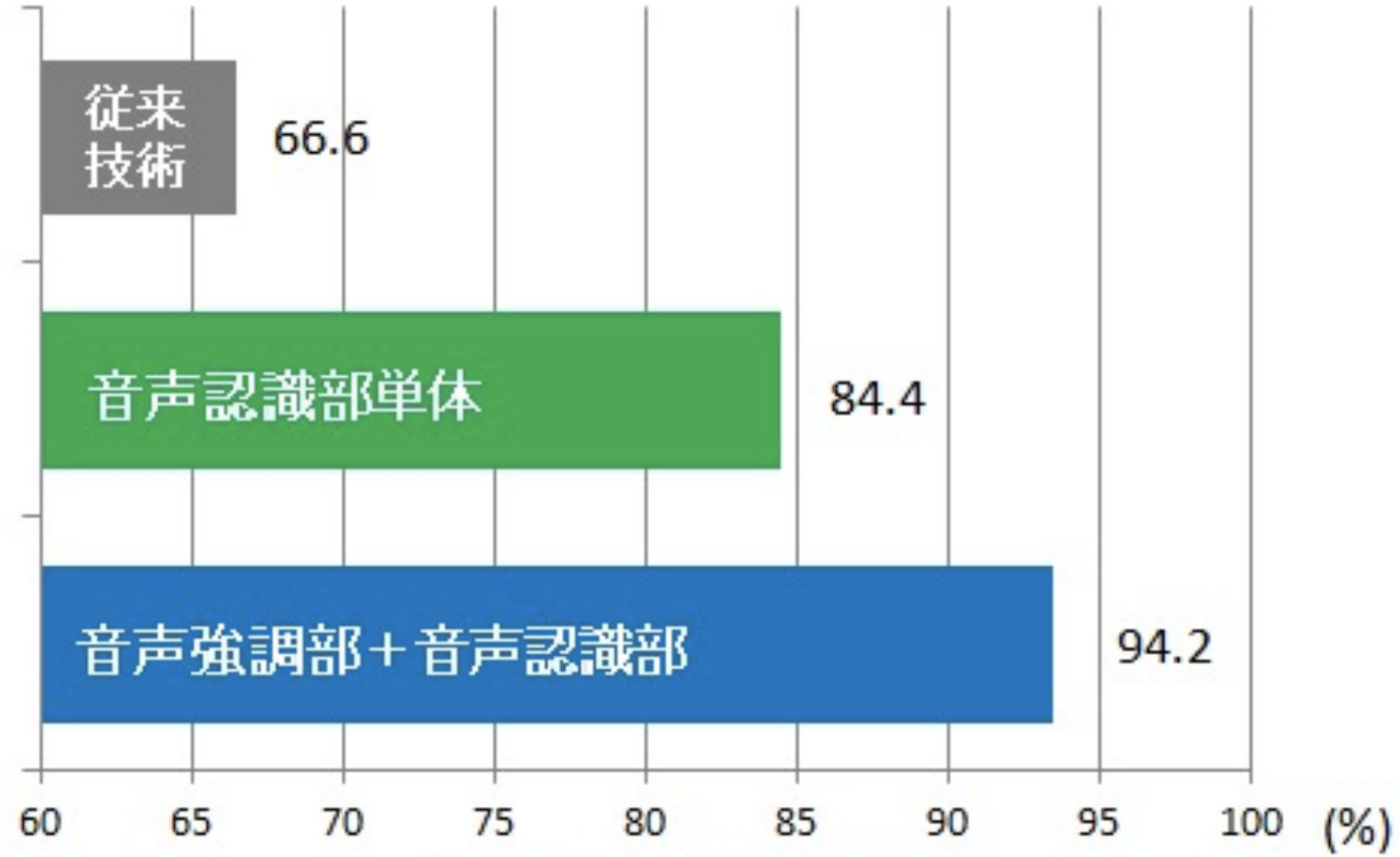
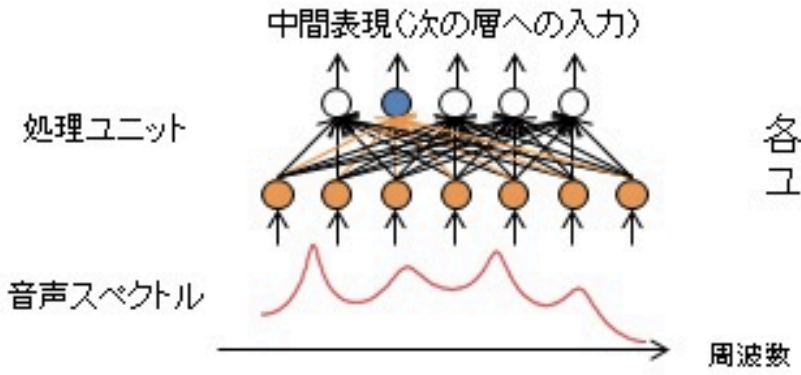


図2：音声認識性能の評価結果

NTT 2015年 CHIME-3 第1位

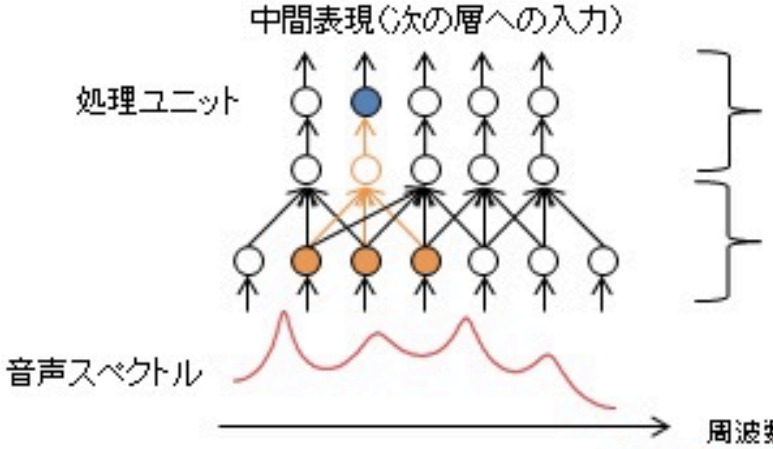
公共エリア雑音下でのモバイル音声認識の国際技術評価で、世界1位の精度を達成

(a) 従来の全結合型ニューラルネット



各ユニットが、音声の全ての周波数成分を用いるため、
ユニット間結合の数が多く、ネットワークの学習が困難

(b) CNN-NIN



CNN-NIN (network in network)の音声認識への適用
ユニットを重ねることで、各音素のスペクトル間の違いをより精緻に表現可能
⇒雑音で各音素のスペクトルが変形していても、頑健に動作

CNN (convolutional neural network)による特徴量抽出
各ユニットが、音声の近接した周波数成分のみを担当することで、 ユニット間結合数が少なく、精度の高い学習が可能
⇒ 雑音に埋もれてしまった他の周波数成分の影響を 受けにくいため、頑健な動作が可能

図3:CNN-NINの音声認識への適用

公共エリア雑音下でのモバイル音声認識の国際技術評価で、世界1位の精度を達成

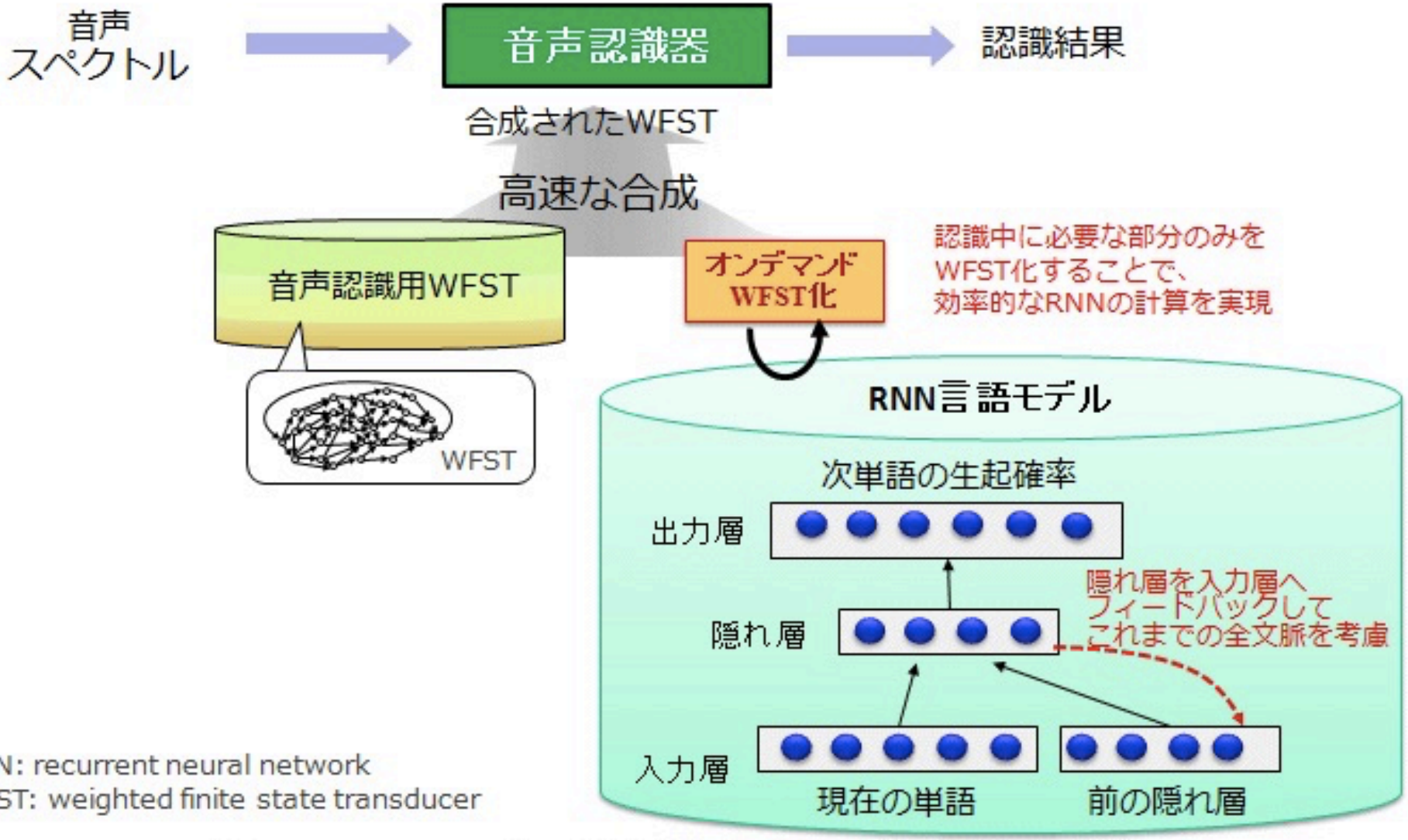


図4：WFST-RNN-LM：長い文脈依存性を考慮した高速・高精度音声認識

2017/8/20

The Microsoft 2017 Conversational Speech Recognition System

August 20, 2017

View Link

BibTex

Authors

Wayne Xiong
Lingfeng Wu
Fil Alleva
Jasha Droppo

Abstract Related Info

We describe the 2017 version of Microsoft's conversational speech recognition system, in which we update our 2016 system with recent developments in neural-network-based acoustic and language modeling to further advance the state of the art on the Switchboard speech recognition task. The system adds a CNN-BLSTM acoustic model to the set of model architectures we combined previously, and includes character-based and dialog session aware LSTM language models in rescoring. For system combination we adopt a two-stage approach, whereby subsets of acoustic models are first combined at the senone/frame level,

https://goo.gl/8ACSdX

マイクロソフトの研究者が音声認識の新たなマイルストーンを達成　2017年8月24日

- 昨年、マイクロソフトの音声と対話の研究グループが、Switchboard 会話音声認識のタスクにおいて、人間と同等の正確性を達成したことを発表しました。これは、プロの書記と同じレベルで単語を認識できるテクノロジを開発できたことを意味します。マイクロソフトの聴き取りシステムは、人間と同等の 5.9 パーセントの誤認識率を達成しました。
- その後、他の研究者たちが実施したより複雑な複数の認識プロセスを使用した別の研究により、人間の誤認識率が5.1パーセントを達成する必要があるとの結果が得られました。
- 本日、マイクロソフトの研究チームが開発した音声認識システムによる 5.1 パーセントの誤認識率が達成されたことを発表します。これは業界の新たなマイルストーンであり、マイクロソフトが昨年達成した正確性を大きく上回ります。

https://goo.gl/PmvBX2

ボイス・アシスタントに対する二つの期待

ボイス・アシスタントに対する二つの期待

第一に、機械と人間のインターフェースが、人間と人間の通常のインターフェースと同じように、音声中心のものになっていくという期待がある。

第二に、言語を理解する賢い機械の実現が、ディープ・ラーニングを中心とする人工知能技術によって可能になるという期待がある。

ボイス・アシスタントに対する二つの期待

第一に、機械と人間のインターフェースが、人間と人間の通常のインターフェースと同じように、音声中心のものになっていくという期待がある。 ➡ これは正しい

第二に、言語を理解する賢い機械の実現が、ディープ・ラーニングを中心とする人工知能技術によって可能になるという期待がある。

➡ これは、すぐには難しい

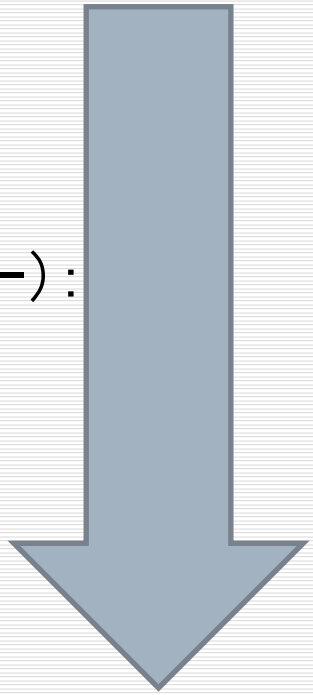
第一の期待　音声中心のインターフェースへの変化

- キーボード＋ディスプレー：
  単一コンソール

- キーボード＋マウス＋ディスプレー
  マルチ・ウィンドウ　（デスクトップ・メタファー）：
  - ブラウザー上のWebアプリ
  - ブラウザー上のWebアプリ：AJAX

- スクリーンタッチ
  スマートフォン、タブレット

- 音声インターフェース
  スマホでの「音声検索」がはしり



より多くの人に
より使いやすく

音声インターフェースの特徴

- 文字情報は、一般に、音声情報より、情報量が多い。また、静止画・動画等のビジュアルな情報の方が、音声だけの情報より、情報量が多く、かつ豊かである。料理のレシピの情報は、音声だけだと少し心もとない。
- 文書の字句や文言を読み落としても、改めて、自分で読み直せばいい。会話で、聞き漏らしたら、相手にもう一度話してもらうしかない。
- 同時に、ショッピングや飲食店に行った時に、必ず筆談しなければならないとすれば、それは面倒に感じるに違いない。音声による会話は、なんの準備もいらず簡便である。
- テレビを観ながら、他の仕事をするのは難しいが、音楽を聴きながら、仕事をするのは、それより容易である。

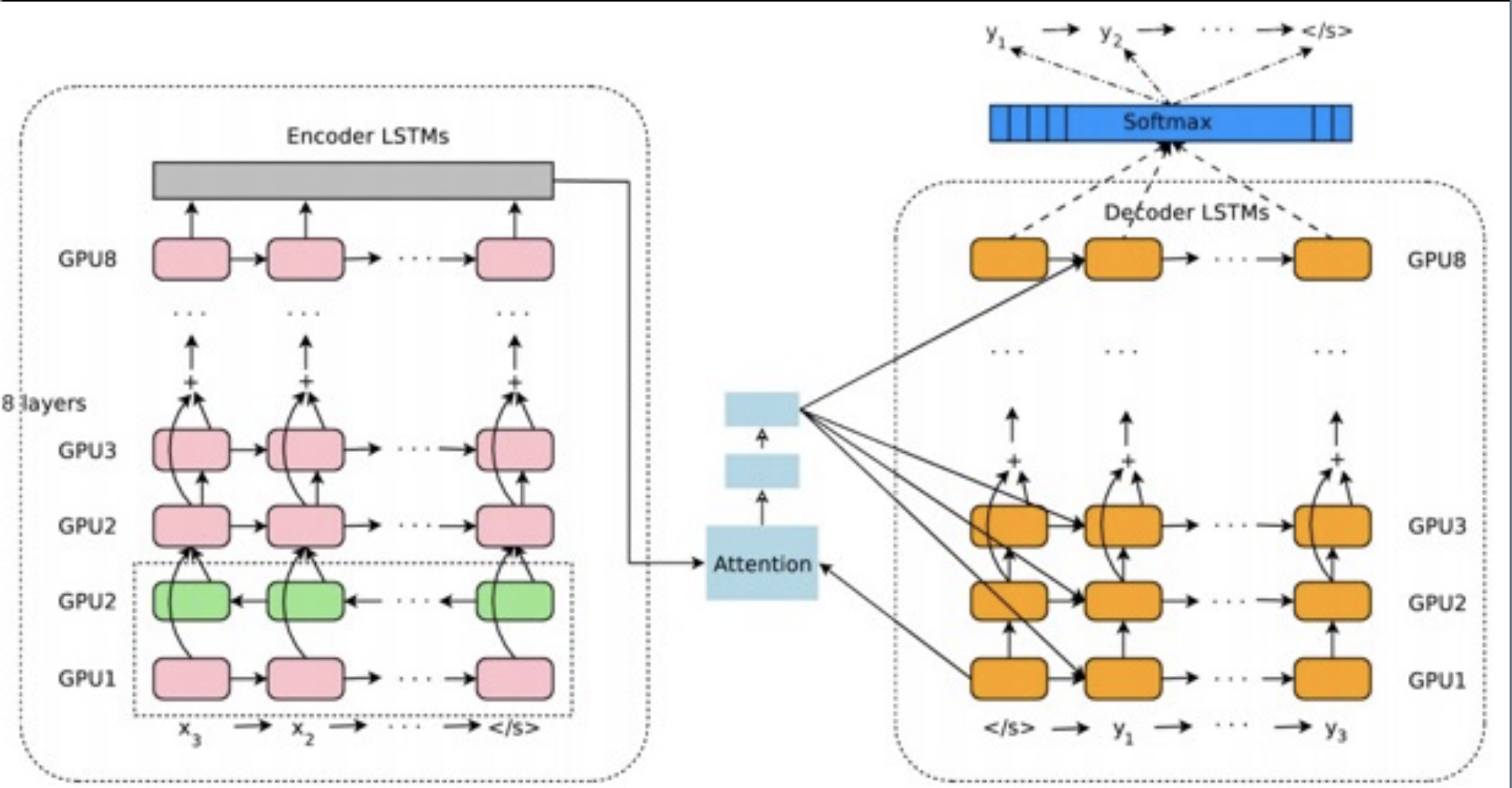
第二の期待　機械が知能と会話能力を持つ

- どうして、この課題が難しいのか？
- 確かに、ディープラーニング技術は、ボイス・アシスタントの基礎技術である音声認識の精度を飛躍的に向上させた。その意味では、ボイス・アシスタント技術は、我々の最も身近なところにあるディープラーニング技術の一つである。
- ただ、音声認識技術（speech2text）が可能にするのは、音声を自然言語の文字列に変換するところまでである。speech2textが返した文字列を、どのように処理すべきかについては、ディープラーニング技術は、ほとんど何も教えてくれないのである。

現在の人工知能技術での自然言語理解の最前線

- Googleニューラル機械翻訳
- Differentiable Neural Computer

2016年
Google ニューラル機械翻訳



Google ニューラル機械翻訳

- 2016年に発表されたGoogleのニューラル機械翻訳は、現在の人工知能技術での自然言語処理のめざましい成功例である。
- そこには、全ての言語に共通する「メタ言語＝普遍言語」（多分、それは「意味」の担い手になるのだが）の存在を、機械がはじめて認識したのではと思わせる興味ふかい兆候があるのだが、ディープラーニングのフレームそのものが、それを外部に明確な形で取り出すことを許さない。コネクショニズムというのは、そういうものだ。
- A言語からB言語への翻訳は、現象的には、A言語の文字列からB言語の文字列の変換として、学習され実行される。翻訳による意味の同一性は、A言語の話し手ととB言語の話し手によって、すなわち人間の言語能力によってチェックされているようにも見える。

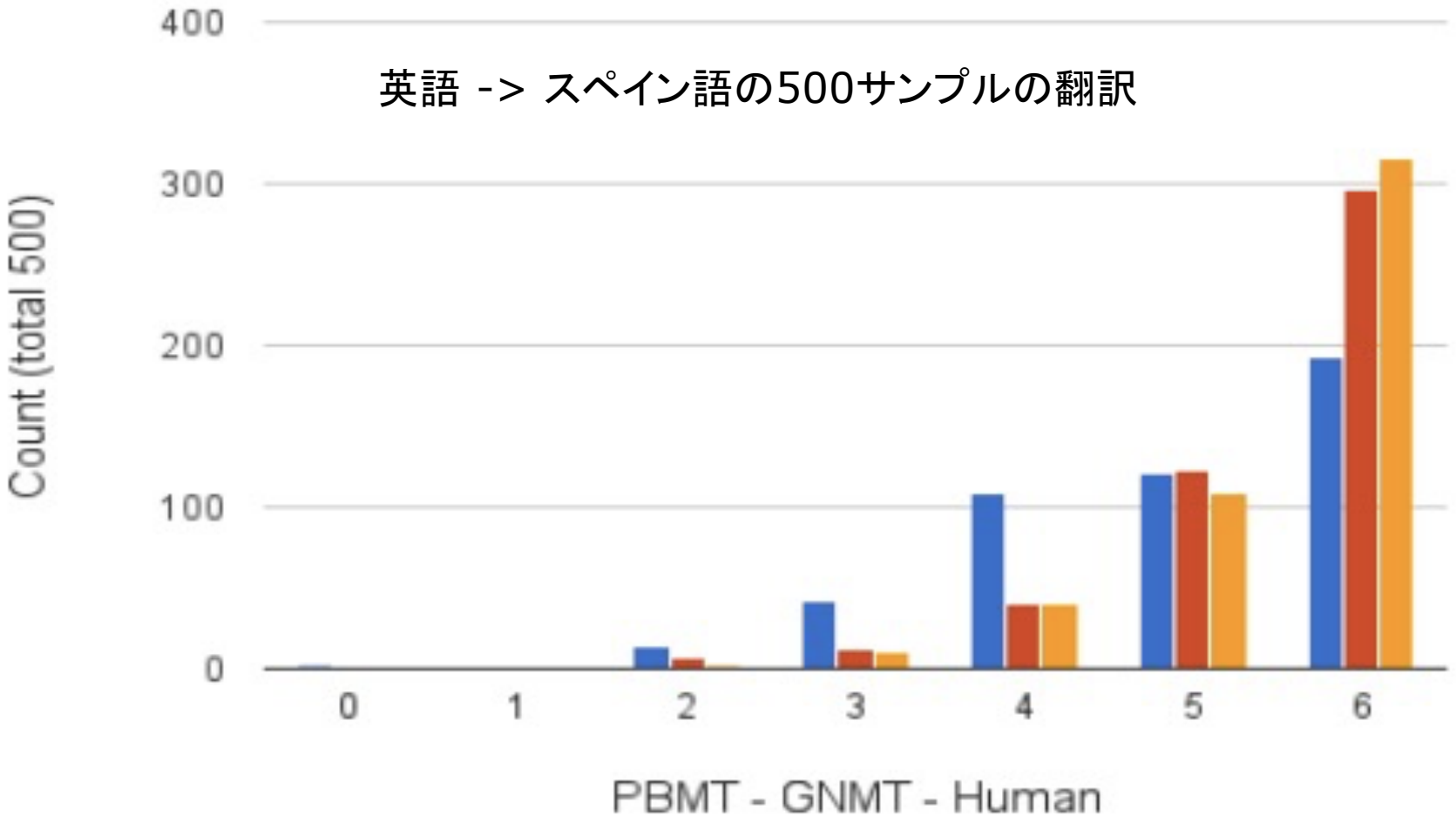
以前のGoogle翻訳との比較

Table 10: Mean of side-by-side scores on production data

	PBMT	GNMT	Human	Relative Improvement
English → Spanish	4.885	5.428	5.504	87%
English → French	4.932	5.295	5.496	64%
English → Chinese	4.035	4.594	4.987	58%
Spanish → English	4.872	5.187	5.372	63%
French → English	5.046	5.343	5.404	83%
Chinese → English	3.694	4.263	4.636	60%

翻訳結果を、0~6の7段階で、人間が評価したもの。
以前のGoogle翻訳と比較して、58%~87%の改善が見られる。

以前のGoogle翻訳（PBMT）、現在のGoogle翻訳(GNMT)、人間による翻訳 の得点分布



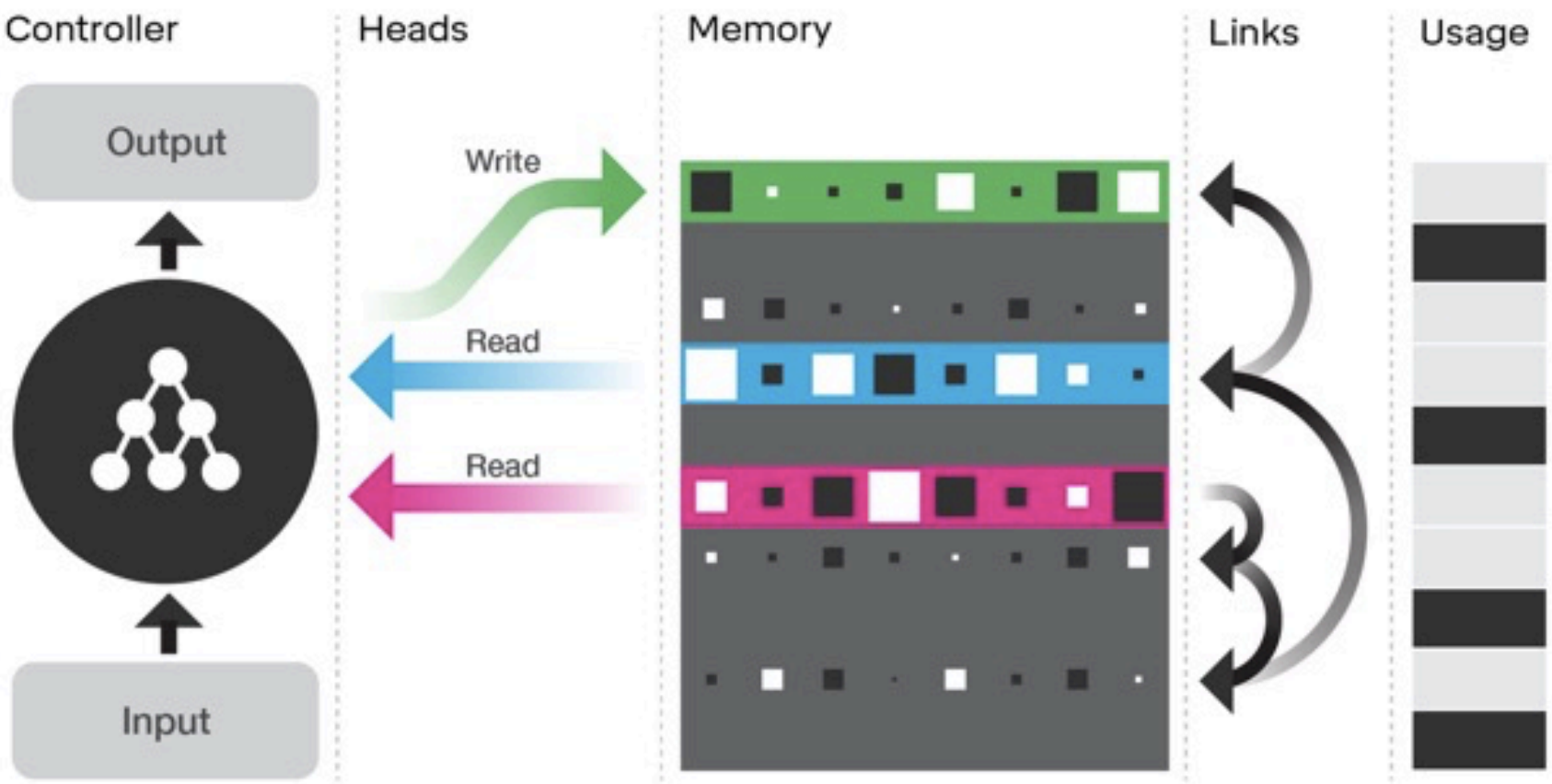
2016年
Differential Neural Computer



Differentiable Neural Computer

- 同じく、2016年のDeep MindのDifferentiable Neural Computerは、baBiという訓練用のデータセット上で、次のような課題に取り組んでいる。それが、なかなか難しいのだ。
- 例えば、「ジョンはグランドにいます。ジョンはフットボールをピックアップしました」というストーリーに続いて「フットボールはどこにありますか？」という質問に、グラウンド」と答える。システムは2つの裏付けの事実を組み合わせる必要がある。「羊はオオカミを恐れています。」「ガートルードは羊です。」「ネズミはネコを恐れています。」「ガートルードは何を恐れているのですか？」（答え、「オオカミ」）では、基本的な演繹能力（および引っ掛けへの耐性）をテストする。

baBiについては、https://research.fb.com/downloads/babi/ あるいは、https://github.com/facebook/bAbI-tasks を参照されたい。


Illustration of the DNC architecture
Controller
Heads
Memory
Links
Usage
Output
Write
Read
Read
Input

Synthetic question answering experiments

John is in the playground.
Bob is in the office.
Where is John? A:playground

John is in the playground.
Bob is in the office.
John picked up the football.
Bob went to the kitchen.
Where is the football? A:playground
Where was Bob before the kitchen? A:office

John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? A:office

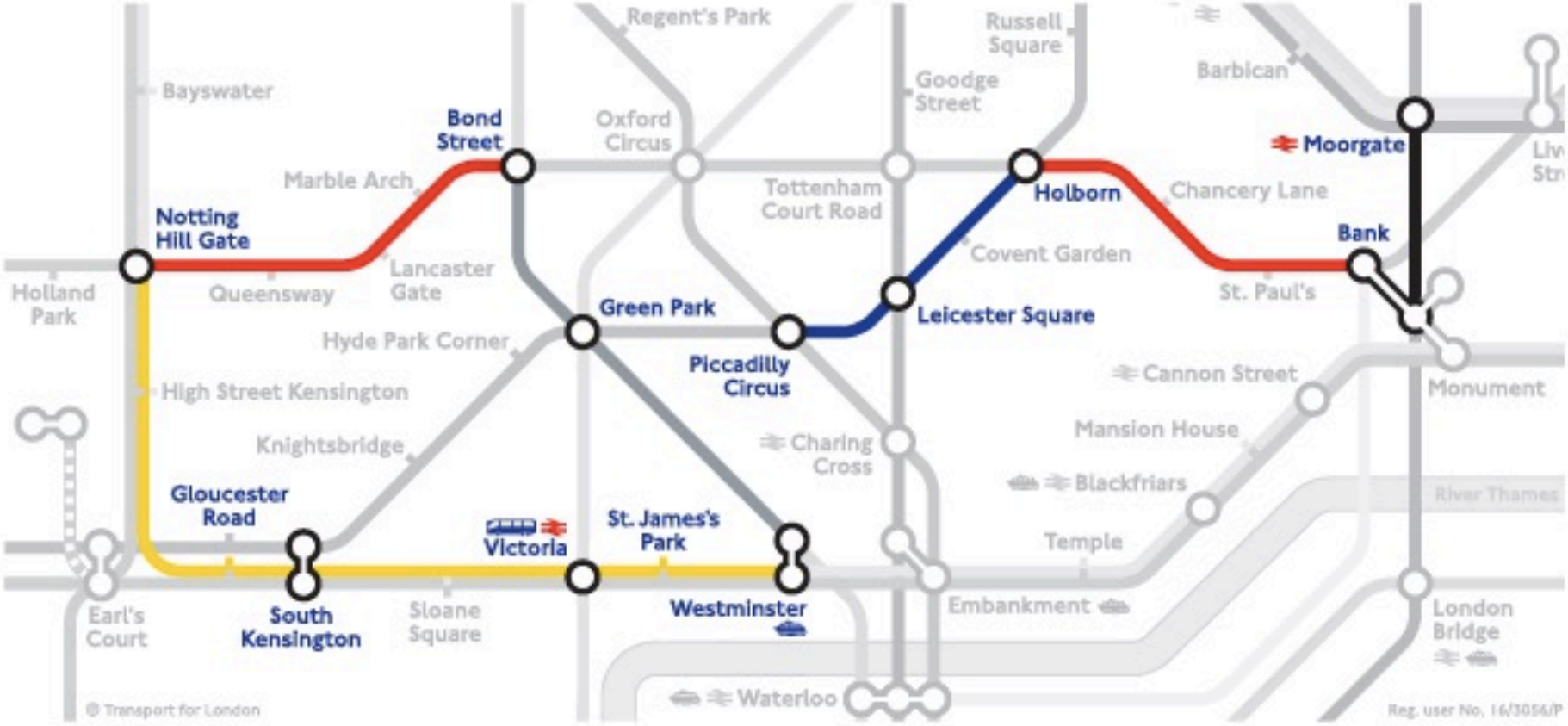
1 Lily is a frog.
2 Bernhard is a frog.
3 Bernhard is green.
4 Brian is a lion.
5 Brian is white.
6 Julius is a swan.
7 Julius is green.
8 Lily is green.
9 Greg is a swan.
10 What color is Greg?

-- baBi Task 16

1 The triangle is above the pink rectangle.
2 The blue square is to the left of the triangle.
3 Is the pink rectangle to the right of the blue square?
4 Is the blue square below the pink rectangle?
5 Is the blue square to the right of the pink rectangle?

-- baBi Task 17

b London Underground



実験結果をどう見るのか？　baBi task

- baBi task については、”best result” と “mean result” の両方を見た方がいいように思う。「苦手な問題」があるように思う。これは、おそらくFacebookのMemory Network モデルでも同様に見える。
- 結果的には、 ”best result” でも「帰納」と「位置推論」で失敗しているが、根拠から演繹的に推論するTask2, Task3も、“mean result” を見る限り、安定していないように見える。
- 比較するには、いくつかの前提が必要なのだが、論理的推論能力に関していえば、最も最初期の自動証明プログラムであるSimonの”LOGIC THEORIST” （「プリンキピァ・マテマティカ」の最初の52の定理のうち38の定理を証明できた）の能力を大きく超えてはいないようにも思う。

実験結果をどう見るのか？ baBi task

- 具体的な文が直接に表すものを理解するのと、そこから論理的構造を抽象して、それに基づいて推論するのとは、違うレベルの「知能」の働きなのだ。
- おそらく、そこには超えなければいけない課題が何かある。

bAbi Task 1

Single supporting fact

1 John travelled to the hallway.

2 Mary journeyed to the bathroom.

3 Where is John? hallway 1

4 Daniel went back to the bathroom.

5 John moved to the bedroom.

6 Where is Mary? bathroom 2

7 John went to the hallway.

8 Sandra journeyed to the kitchen.

9 Where is Sandra? kitchen 8

10 Sandra travelled to the hallway.

11 John went to the garden.

12 Where is Sandra? hallway 10

13 Sandra went back to the bathroom.

14 Sandra moved to the kitchen.

15 Where is Sandra? kitchen 14

bAbi Task 2

Two supporting facts

1 Mary got the milk there.

2 John moved to the bedroom.

3 Sandra went back to the kitchen.

4 Mary travelled to the hallway.

5 Where is the milk? hallway 1 4

6 John got the football there.

7 John went to the hallway.

8 Where is the football? hallway 6 7

9 John put down the football.

10 Mary went to the garden.

11 Where is the football? hallway 9 7

12 John went to the kitchen.

13 Sandra travelled to the hallway.

14 Where is the football? hallway 9 7

15 ....

bAbi Task 3

Three supporting facts

1 Mary moved to the bathroom. 2 Sandra journeyed to the bedroom.
3 Mary got the football there.
4 John went back to the bedroom.
5 Mary journeyed to the office.
6 John journeyed to the office.
7 John took the milk.
8 Daniel went back to the kitchen.
9 John moved to the bedroom.
10 Daniel went back to the hallway.
11 Daniel took the apple.
12 John left the milk there.
13 John travelled to the kitchen.
14 Sandra went back to the bathroom.
15 Daniel journeyed to the bathroom.
16 John journeyed to the bathroom.
17 Mary journeyed to the bathroom.
18 Sandra went back to the garden.
19 Sandra went to the office.
20 Daniel went to the garden.
21 Sandra went back to the hallway.
22 Daniel journeyed to the office.
23 Mary dropped the football.
24 John moved to the bedroom.
25 Where was the football before the bathroom? office 23 17 5

bAbI best results

Task	bAbI Best Results						
	LSTM (Joint)	NTM (Joint)	DNC1 (Joint)	DNC2 (Joint)	MemN2N (Joint) [21]	MemN2N (Single) [21]	DMN (Single) [20]
1: 1 supporting fact	24.5	31.5	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
2: 2 supporting facts	53.2	54.5	1.3	0.4	1.0	**0.3**	1.8
3: 3 supporting facts	48.3	43.9	2.4	**1.8**	6.8	2.1	4.8
4: 2 argument rels.	0.4	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
5: 3 argument rels.	3.5	0.8	**0.5**	0.8	6.1	0.8	0.7
6: yes/no questions	11.5	17.1	**0.0**	**0.0**	0.1	0.1	**0.0**
7: counting	15.0	17.8	**0.2**	0.6	6.6	2.0	3.1
8: lists/sets	16.5	13.8	**0.1**	0.3	2.7	0.9	3.5
9: simple negation	10.5	16.4	**0.0**	0.2	**0.0**	0.3	**0.0**
10: indefinite knowl.	22.9	16.6	0.2	0.2	0.5	**0.0**	**0.0**
11: basic coreference	6.1	15.2	**0.0**	**0.0**	**0.0**	0.1	0.1
12: conjunction	3.8	8.9	0.1	**0.0**	0.1	**0.0**	**0.0**
13: compound coref.	0.5	7.4	**0.0**	0.1	**0.0**	**0.0**	0.2
14: time reasoning	55.3	24.2	0.3	0.4	**0.0**	0.1	**0.0**
15: basic deduction	44.7	47.0	**0.0**	**0.0**	0.2	**0.0**	**0.0**
16: basic induction	52.6	53.6	52.4	55.1	**0.2**	51.8	0.6
17: positional reas.	39.2	25.5	24.1	**12.0**	41.8	18.6	40.4
18: size reasoning	4.8	2.2	4.0	**0.8**	8.0	5.3	4.7
19: path finding	89.5	4.3	**0.1**	3.9	75.7	2.3	65.5
20: agent motiv.	1.3	1.5	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Mean Err. (%)	25.2	20.1	4.3	**3.8**	7.5	4.2	6.4
Failed (err. > 5%)	15	16	**2**	**2**	6	3	**2**

Google **Facebook**

bAbI mean results あまり成績良くない

	bAbI Mean Results			
Task	LSTM	NTM	DNC1	DNC2
1: 1 supporting fact	28.4 ± 1.5	40.6 ± 6.7	**9.0 ± 12.6**	16.2 ± 13.7
2: 2 supporting facts	56.0 ± 1.5	56.3 ± 1.5	**39.2 ± 20.5**	47.5 ± 17.3
3: 3 supporting facts	51.3 ± 1.4	47.8 ± 1.7	**39.6 ± 16.4**	44.3 ± 14.5
4: 2 argument rels.	0.8 ± 0.5	0.9 ± 0.7	**0.4 ± 0.7**	**0.4 ± 0.3**
5: 3 argument rels.	3.2 ± 0.5	1.9 ± 0.8	**1.5 ± 1.0**	1.9 ± 0.6
6: yes/no questions	15.2 ± 1.5	18.4 ± 1.6	**6.9 ± 7.5**	11.1 ± 7.1
7: counting	16.4 ± 1.4	19.9 ± 2.5	**9.8 ± 7.0**	15.4 ± 7.1
8: lists/sets	17.7 ± 1.2	18.5 ± 4.9	**5.5 ± 5.9**	10.0 ± 6.6
9: simple negation	15.4 ± 1.5	17.9 ± 2.0	**7.7 ± 8.3**	11.7 ± 7.4
10: indefinite knowl.	28.7 ± 1.7	25.7 ± 7.3	**9.6 ± 11.4**	14.7 ± 10.8
11: basic coreference	12.2 ± 3.5	24.4 ± 7.0	**3.3 ± 5.7**	7.2 ± 8.1
12: conjunction	5.4 ± 0.6	21.9 ± 6.6	**5.0 ± 6.3**	10.1 ± 8.1
13: compound coref.	7.2 ± 2.3	8.2 ± 0.8	**3.1 ± 3.6**	5.5 ± 3.4
14: time reasoning	55.9 ± 1.2	44.9 ± 13.0	**11.0 ± 7.5**	15.0 ± 7.4
15: basic deduction	47.0 ± 1.7	46.5 ± 1.6	**27.2 ± 20.1**	40.2 ± 11.1
16: basic induction	**53.3 ± 1.3**	53.8 ± 1.4	53.6 ± 1.9	54.7 ± 1.3
17: positional reas.	34.8 ± 4.1	**29.9 ± 5.2**	32.4 ± 8.0	30.9 ± 10.1
18: size reasoning	5.0 ± 1.4	4.5 ± 1.3	**4.2 ± 1.8**	4.3 ± 2.1
19: path finding	90.9 ± 1.1	86.5 ± 19.4	**64.6 ± 37.4**	75.8 ± 30.4
20: agent motiv.	1.3 ± 0.4	1.4 ± 0.6	**0.0 ± 0.1**	**0.0 ± 0.0**
Mean Err. (%)	27.3 ± 0.8	28.5 ± 2.9	**16.7 ± 7.6**	20.8 ± 7.1
Failed (err. > 5%)	17.1 ± 1.0	17.3 ± 0.7	**11.2 ± 5.4**	14.0 ± 5.0

現在の技術で確実にできること
音声を文字列に変換するところまで

ディープ・ラーニング技術に限って言えば、現在のディープ・ラーニング技術が、ボイス・アシスタントの分野で確実に達成できることは、音声を文字列に変換するところまででである。変換された文字列をどう処理すべきかは、現在のディープ・ラーニング技術は、何も教えてくれないと考えたほうがいいのだ。

そのことは、現在のディープ・ラーニング技術の最先端である、Google Deep MindのDifferentiable Neural Computerや、FacebookのMemory Networkが、bAbiデータセットと悪戦苦闘していることを見ればよくわかる。小学4年生程度の国語の問題をうまくハンドルすることが、現実的には、最先端の課題なのだ。

逆に、彼らの研究を突き動かしている最大の力は、ビジネス的には、ボイス・アシスタントを巡る競争での優位の確保である。

意味のプラグマティックな理解

Syntactic, Semantic and Pragmatic

- 構文論・意味論・語用論は、言語に対する異なったレベルからのアプローチである。
- 「第一原理」からの演繹を目指す、Syntacticalなレベルからの自然言語の「意味理解」が、現状では難しいとしても、別のアプローチはありうる。

機械との会話で用いられる文とその意味

- 機械に話しかける文には、明確な特徴がある。それは、疑問文か命令文である。「富士山の高さは？」は、前者。「部屋の電気を消して！」は、後者である。
- 疑問文に対しては、検索でもなんでもいろんな手段をつかって正しい情報を返せば、機械は、会話の文の意味を理解しているとみなすことができる。
- 命令文に対しては、例えば機器のコントロールのように、予期した変化を起こせば、機械は、会話の文の意味を「理解」しているとみなすことはできる。

Botの機能の三つのタイプ

第一のタイプは、ハードウェアやアプリが持っている、機能やサービスを呼び出す入出力インターフェースのボイス化を目的とするBotである。これは、問題(「会話」)のドメインも限られており、実装は比較的容易である。

第二のタイプは、「情報」や「知識」を、ボイスを通じて提供するタイプのBotである。現在の「音声検索」と共通の技術である。
ただ、このBotを可能にするバックグラウンドの検索エンジンの開発は容易ではない。あるいは、ごくごく単純な知識を自前で埋め込まねばならない。

Bot(広くは人工知能技術)と検索の関係は、ディープ・ラーニングのブームの陰に隠されている感があるのだが、重要で深いものだ。

AlexaとGoogle Assistantの違い

これは、「機能」の特徴づけであり、実際のボイス・アシスタントは、二つの機能を兼ね備えている。

ただ、機能のベースに何を置くかで、それぞれのボイス・アシスタントの特徴が出る。

Alexaは、第一の機能実現がベースであり、

Google Assistantは、第二の機能がベースである。

「汎用Bot」?

第三のタイプは、第一でも第二のタイプでもない、いわば、「汎用の会話Bot」である。多くのコンシューマが期待しているのはこのタイプなのかもしれないし、多くの企業と開発者が夢見ているのも、このタイプかもしれない。ただ、この1，2年で、この機能を実現できることは難しいだろう。

第三のタイプでは、現時点で現実的に可能なことは、「ああいえば、こういう」という人の手による「シナリオ」を、できるだけ沢山かき集めることに帰着するだろう。

「人工知能」という触れ込みなのに、蓋を開けてみたら、中に沢山の小人さん（といっても、普通の人間）が詰め込まれていて、「ああ言われれば、こう言う」というシナリオづくりに汗をかいているだけという、笑えないことになる。

何から始めるべきか？

何から始めるべきか？　第一のタイプから。

僕がエンジニアに勧めたいのは、第一のタイプの入出力のボイス化を目的とするBotでスキルを高めておくことだ。

Iotでハードを作っている人は、そのハードのインターフェースのボイス化に挑戦する。スマートフォン・アプリやWebアプリを作っている人は、そのアプリのインターフェースのボイス化に挑戦する。部分的なものでいいのだ。

実際に、各社が提供しているフレームワークも、入出力型である。

第二のタイプ　情報提供型

- 第二のタイプの開発は、検索大手が、グラフ検索へのアプローチに大きな制限をかけている状況では、難しい。
- ただ、情報の提供は、なにも検索エンジンだけが可能であるわけではない。ニュースのブリーフィングや、特定分野のトピックの音声による提供は、むしろ、ボイス・アシスタントのサービス化とビジネス化にとって重要な市場になる。
- 現在のボイス・アシスタント・システムが、まずは、「スマート・スピーカー」として立ち上がるのは、音楽コンテンツの提供も、広い意味では、この情報提供型に含まれるからだと考えた方がいい。
- ビジネスに関心のある人は、ここで頭を使う方がいい。

第三の「汎用会話Bot」

- この課題は、なかなか難しい。ただ、人工知能・自然言語理解の本丸の一つなので、研究者なら放棄するわけにはいかない。
- ただ、ここにも、全く別のアプローチがある。それは、いわゆる「人工無脳」系のテクニックを、意識的に深めることだ。そこには、今後、大きな課題となるのが確実な、音声インターフェースの改善にとっては、重要なノウハウが含まれている可能性がある。

Eugene Goostman



https://goo.gl/IbXZjY

Turing Test 2014 での Eugene Goostmanの優勝

- 2014年に開催されたコンテスト"Turing Test 2014" では、Eugene Goostman君が優勝して、主催者とメディアの一部は、初めてコンピューターがTuring Testをパスしたと発表して話題になった。
- "TURING TEST SUCCESS MARKS MILESTONE IN COMPUTING HISTORY”「Turing Testの成功は、コンピュータの歴史に大きな足跡を刻んだ」 http://goo.gl/ARdP1n
- "The Life and Times of 'Eugene Goostman,' Who Passed the Turing Test”「Turing Testをパスした 'Eugene Goostmanの人生と時間」 http://goo.gl/yFViDQ ここには、「彼」と人間との会話の一部が載っている。

Turing Test 2014 での Eugene Goostmanの優勝

- ただ、皆がそうした見解で一致しているわけではない。（多分、僕も）"Don't Believe The So-Called Turing Test Breakthrough”「いわゆるTuring Testでのブレークスルーなぞ、信じてはいけない」 http://goo.gl/pvlwV2

Scott Aaronson と Eugene Goostman の「会話」

- Scott Aaronson がこのEugene Goostman と対話した記録がある。"My Conversation with “Eugene Goostman,” the Chatbot that’s All Over the News for Allegedly Passing the Turing Test”「あちこちのニュースで、Turing Testをパスしたと喧伝されているチャットボット Eugene Goostman と僕との会話」 http://goo.gl/t5gfFC

  こんな感じである。

Scott Aaronson と Eugene Goostman の「会話」

- Scott: 靴箱とエベレスト山、どっちが大きい？
- Eugene: 今すぐには選べないな。あとで、よく考えるよ。そうだ、質問するの忘れてた、あなたはどこから来たの？
- Scott: ラクダの足は何本？
- Eugene: 2本と4本の間かな？　多分、3本かな。:-))) ところで、あなたのことよく知らないんだけど、多分、聞き漏らしたかな？
- Scott: ムカデには足は何本？
- Eugene: ちょうど2本だよ。もっとも、チェルノブイリで突然変異したのは5本まで持っているかもしれない。あなたは、私をだまそうとしているんでしょ？

Scott Aaronson と Eugene Goostman の「会話」

- Scott: そうじゃないよ。僕は、君がチャットボットではないと知る必要があるんだ。質問には真っ直ぐに答えてくれるかな。アリには、何本足がある？
- Eugene: 2本と4本の間かな？　多分、3本かな。:-)))
  なんて実りのある会話なんだろう ;-)
- Eugeneはデタラメな答えを返しながら、自分の得意なところに、会話を誘導しようとしている。出身や職業がわかるともっともらしいことを言うことができるので。一方、Scottのほうは、相手の反応に関係なしに、執拗に動物の足の数を問い続ける。
- あとで、この会話を振り返ったScottのコメントが面白い。「僕のほうが、ロボットに見えるかも」

プラットフォームとしてのスマートフォン

今後、様々な、ボイス・アシスタンスの専用アプライアンスが登場し、市場ではその競争が激しくなると思うのだが、ただ、ボイス・アシスタントのプラットフォームを、そうした専用アプライアンスに限る必要はないということだ。

我々にもっとも身近なプラットフォームは、スマートフォンだ。
自作ハードであれ、Webアプリであれ、スマートフォンからのボイス・コントロールを追求することは、重要だし、簡単にできる。

スマートフォンからのハード、アプリのボイス・コントロールに対する市場ニーズは、確実に存在するし、そこで得たノウハウは、あとできっと役に立つ。専用アプライアンスは、いつもマイクのスイッチが入っている、スマホ、あるいはタブレットだと思えばいい。

人工知能と検索技術

Google音声検索

ボイス・アシスタントが最初に登場したのは、Google の音声検索としてである。

日本語とGoogle音声検索

- Googleが、音声インターフェース機能を備えた音声検索技術をリリースしたのは2008年である。それは英語を対象としたものだった。日本でのGoogle音声検索のリリースは、2009年で、世界で二番目に早かった。
- (音声→文字列)→検索→(検索された文字列→音声) と言う処理を日本語で行う上で、大きな問題が二つあった。
- 一つは、日本語の文に使われている文字種が、ひらがな・カタカナ・漢字・ASCII英字と多様であること。
  もう一つの問題は、欧米語の文が、基本的にはスペースで区切られて語の境界が明確なのに対して、日本語ではそうした区切りが存在しないこと。(中国語・ハングルでも同様である)
- 発話から、日本語文字列に変換するのが困難な例として、次のようなものがあった。「AKB48」「シャ乱Q」「価格.com」

日本語とGoogle音声検索

- 2008年、Googleは、日本で、すべての年齢・方言・地域をカバーした、500人の日本語の話し手を有給で雇って、数週間で250k種類の発話集を作った。データとしては、その地域で最も多く検索されたトップN個の検索文を使い、それを読み上げさせた。iPhoneとBlackberryが使われたと言う。
- この日本語発話コーパスをベースに、Googleの日本語音声認識の精度は大きく上がった。
- このプロジェクトの最大の成果は、スペースの区切りなしでも、かつ、字句の文法構造の解析抜きでも、文を語に分割する、新しい技術 "WordPiece"が生み出されたことである。
- WordPieceは、後のGoogleニューラル翻訳で、基本的なツールとして活用されることになる。日本語の難しさが、Googleの自然言語処理技術の発展に貢献したのである。

2012年 [35] Schuster, M., and Nakajima, K. ”**Japanese and Korean voice search**” https://goo.gl/DUxQuv

1. まず、単位語の目録を、ベーシックUnicode（日本語の漢字・ひらがな・カタカナ、韓国語のハングル、そしてすべてのASCIIコードを含む）で作る。最終的には、日本語でトータル22000語、韓国語11000の目録になる。
2. この目録を使った訓練データ上に、言語モデルを構築する。
3. 現在の語の目録から二つの単位語を取り出し、一つに結合して新しい単位語を作り、単位語の目録を一つ増やす。可能な組み合わせの中から、言語モデルに追加された時、訓練データ上で、類似度を一番増大させるものを、新しい単位語として選ぶ。
4. 事前に定義された単位語の限界に到達するか、類似度の増大が、ある閾値以下に下がるまで、2. の操作を繰り返す。

機械翻訳に利用される、現代のコーパスの規模
Googleの持つデータ量

- WMT'14の英語（En）<-> フランス語（Fr）データセットには、3,600万の文のペアが含まれている。
- WMT'14の英語(En) <-> ドイツ語（De）データセットには、500万の文のペアが含まれている。
- Googleは、内部に、英語 <-> 日本語（Ja）、英語 <-> 韓国語（Ko）、英語 <-> スペイン語（Es）、英語 <-> ポルトガル語（Pt） 等々の多くのデータセットを持っているが、その規模は、先のWMTのデータセットより、２～3桁大きいという。

- Googleニューラル機械翻訳では、GPU100個を使って、フルトレーニングには最大1,000万ステップ、収束までには3週間かかることがあるという。

Google Cloud Speech API Language Support

□119ヶ国語のサポート

Thai (Thailand)
Korean (South Korea)
Chinese, Mandarin (Traditional, Taiwan)
Chinese, Cantonese (Traditional, Hong Kong)
Japanese (Japan)
Chinese, Mandarin (Simplified, Hong Kong)
Chinese, Mandarin (Simplified, China)

Afrikaans (South Africa)	English (Ghana)	Spanish (Colombia)	Spanish (Dominican Republic)
Amharic (Ethiopia)	English (United Kingdom)	Spanish (Costa Rica)	Spanish (Uruguay)
Armenian (Armenia)	English (India)	Spanish (Ecuador)	Spanish (Venezuela)
Azerbaijani (Azerbaijan)	English (Ireland)	Spanish (El Salvador)	Basque (Spain)
Indonesian (Indonesia)	English (Kenya)	Spanish (Spain)	Filipino (Philippines)
Malay (Malaysia)	English (New Zealand)	Spanish (United States)	French (Canada)
Bengali (Bangladesh)	English (Nigeria)	Spanish (Guatemala)	French (France)
Bengali (India)	English (Philippines)	Spanish (Honduras)	Galician (Spain)
Catalan (Spain)	English (South Africa)	Spanish (Mexico)	Georgian (Georgia)
Czech (Czech Republic)	English (Tanzania)	Spanish (Nicaragua)	Gujarati (India)
Danish (Denmark)	English (United States)	Spanish (Panama)	Croatian (Croatia)
German (Germany)	Spanish (Argentina)	Spanish (Paraguay)	Zulu (South Africa)
English (Australia)	Spanish (Bolivia)	Spanish (Peru)	Icelandic (Iceland)
English (Canada)	Spanish (Chile)	Spanish (Puerto Rico)	Italian (Italy)

Javanese (Indonesia)	Portuguese (Portugal)	Telugu (India)	Arabic (Bahrain)
Kannada (India)	Romanian (Romania)	Vietnamese (Vietnam)	Arabic (Algeria)
Khmer (Cambodia)	Sinhala (Srilanka)	Turkish (Turkey)	Arabic (Saudi Arabia)
Lao (Laos)	Slovak (Slovakia)	Urdu (Pakistan)	Arabic (Iraq)
Latvian (Latvia)	Slovenian (Slovenia)	Urdu (India)	Arabic (Kuwait)
Lithuanian (Lithuania)	Sundanese (Indonesia)	Greek (Greece)	Arabic (Morocco)
Hungarian (Hungary)	Swahili (Tanzania)	Bulgarian (Bulgaria)	Arabic (Tunisia)
Malayalam (India)	Swahili (Kenya)	Russian (Russia)	Arabic (Oman)
Marathi (India)	Finnish (Finland)	Serbian (Serbia)	Arabic (State of Palestine)
Dutch (Netherlands)	Swedish (Sweden)	Ukrainian (Ukraine)	Arabic (Qatar)
Nepali (Nepal)	Tamil (India)	Hebrew (Israel)	Arabic (Lebanon)
Norwegian Bokmål (Norway)	Tamil (Singapore)	Arabic (Israel)	Arabic (Egypt)
Polish (Poland)	Tamil (Sri Lanka)	Arabic (Jordan)	Persian (Iran)
Portuguese (Brazil)	Tamil (Malaysia)	Arabic (United Arab Emira	Hindi (India)

Google Nowにできること

Google AssistantとGoogle Now

Google AssistantとGoogle Now

- ボイス・アシスタントに先行したのは、音声検索である。その最大の成功例は、GoogleのGoogle Nowである。
- ボイス・アシスタントとしてのGoogle Assistantと、音声検索のGoogle Nowは、ほとんど、同じものである。

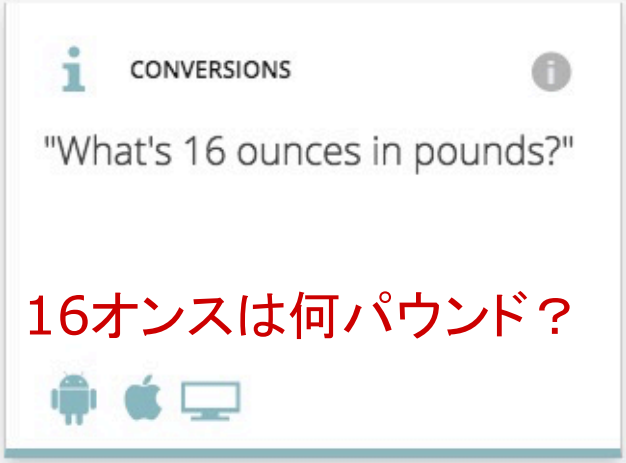
- Google Nowにできることを見てみよう。ここでは、外部の情報の検索（もちろん、Googleの検索機能である）が必要なものを赤字で、システム内部で実行可能なものを青字で示した。
- これらは、先の「意味のプラグマティックな理解」で示唆した、疑問文と命令文の違いに対応している。

Google Nowができること

外部の情報の検索が必要

システム内部で実行可能

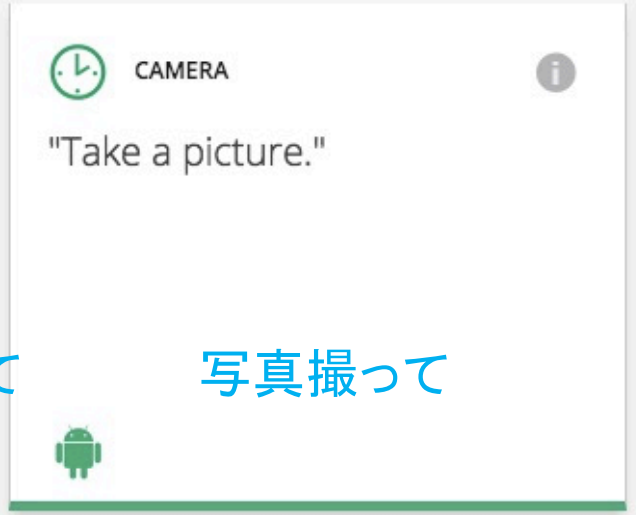










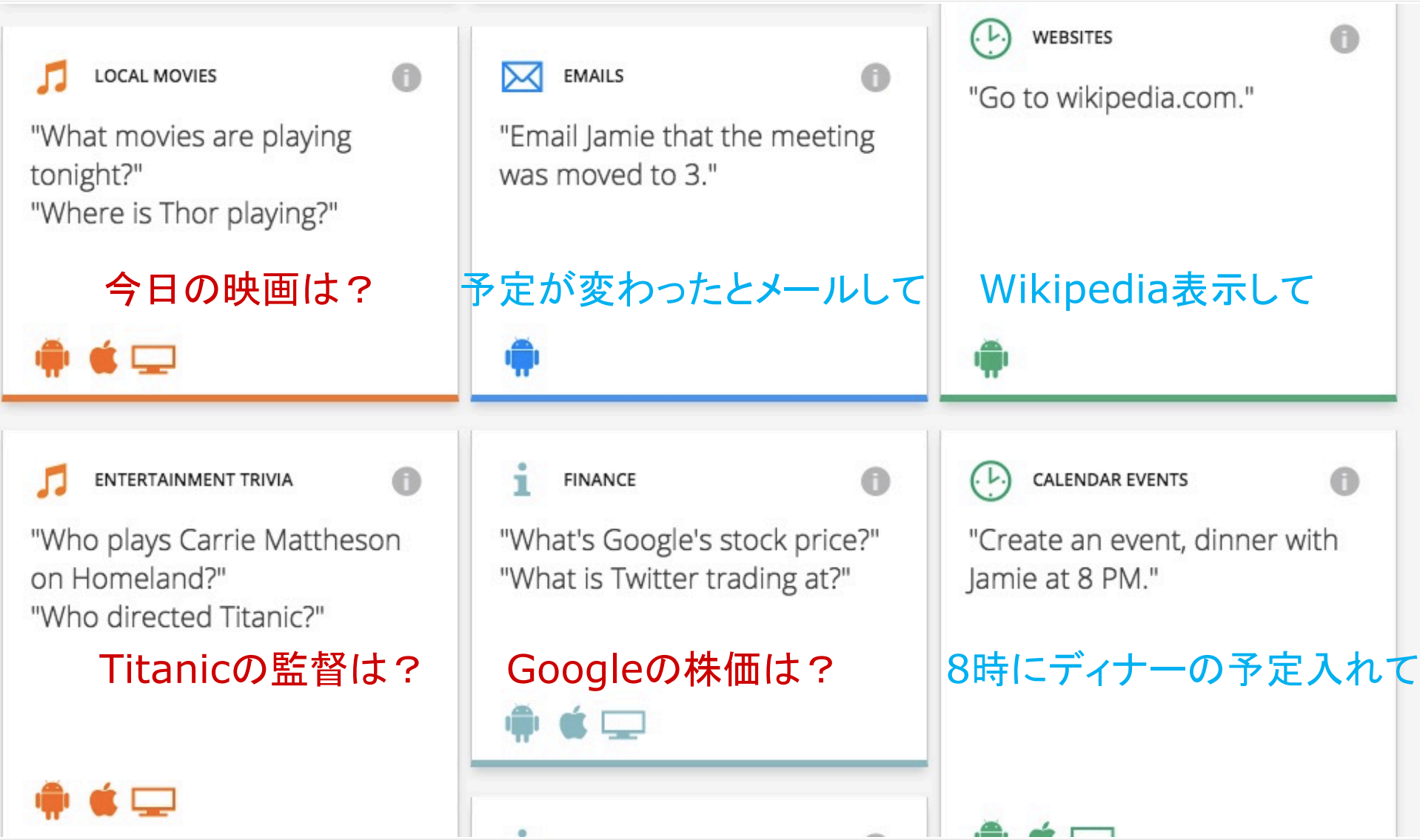


Google Nowは、ボイス・アシスタント・システムである

Google Nowができること

外部の情報の検索が必要

システム内部で実行可能

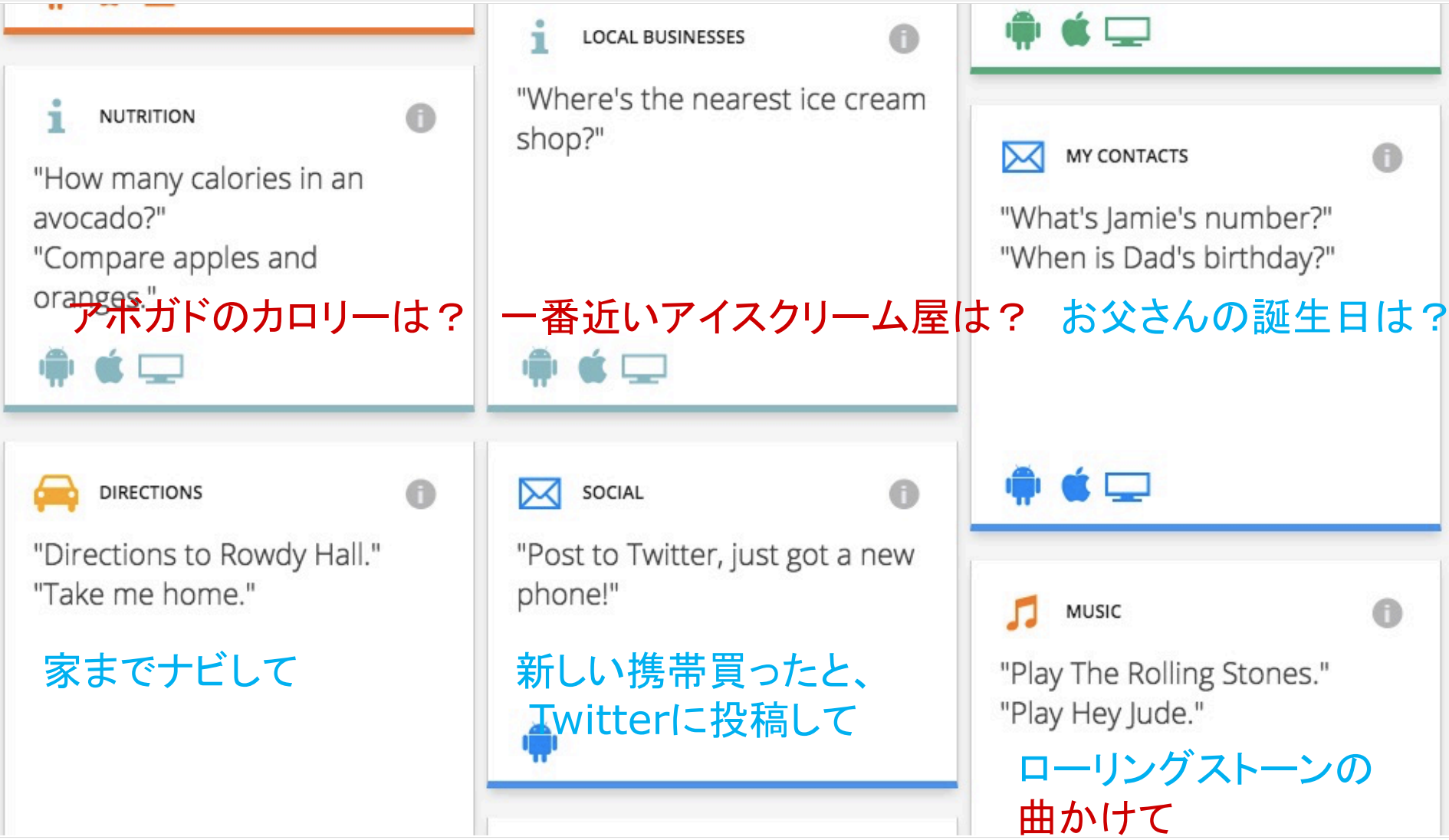


Google Nowは、ボイス・アシスタント・システムである

Google Nowができること

外部の情報の検索が必要

システム内部で実行可能

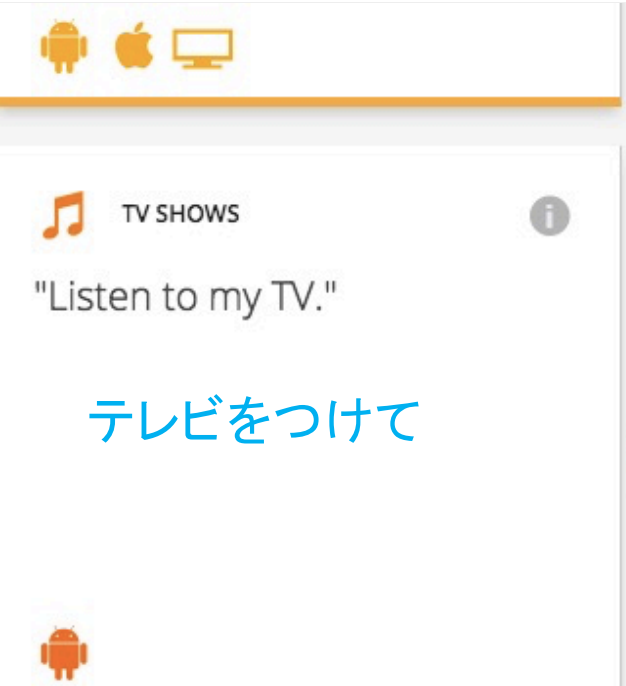


Google Nowは、ボイス・アシスタント・システムである

Google Nowができること

外部の情報の検索が必要

システム内部で実行可能



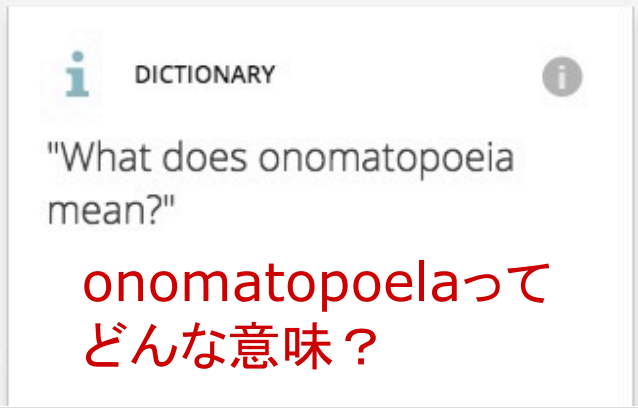












Google Nowは、ボイス・アシスタント・システムである

二つのタイプのデモ

- 「アインシュタインの誕生日は？」
- 「ボルツマンの死んだ日は？」
- 「ボルツマンの論争相手は？」
- 「Taylor Swiftの誕生日は？」
- 「富士山の高さは？」
- 「稚内市の人口は？」
- 「バッハのマタイ受難曲かけて！」
- 「マーラーの曲かけて！」
- 「ビートルズの音楽かけて！」
- 「丸山にメールを出して！」
- 「丸山に電話して！」
- 「カメラ起動して！」

ボイス・アシスタントと検索エンジン

グラフ検索とEntity ModelとSchema.org

先に見た、第二のタイプのBotを考えるとき、2010年代に起きた検索技術の変化が、大きな意味を持つことに気づく。

それまでの検索は、crawlerが収集したweb上の膨大なテキストから、文字列を探し出し（indexingにバッチのmapreduceを使う）、その文字列を含むURLを適当な順番（pagerank ここでもmapreduce）で返すシステムであった。

2010年代に検索に起きた変化は、まず、Indexingのリアル・タイム化であり、それに続くグラフ検索の手法の導入である。GoogleのKnowledge Graphは、Entity Modelで構成された巨大なグラフを、おそらくは、Google のグラフ検索エンジン Pregel で検索している。ちなみに、Pregelのオープンソース版 Apache Giraphの最大ユーザーは、Facebookである。また、PageRankのディープ・ラーニング版 Brain Rankの投入も行われているようだ。

Googleの検索技術の全貌は、明らかではないのだが、すべての検索がグラフ検索に置き換わったわけではなさそうだ。ただ、第二のタイプのBotについては、現在の技術では、エンティティのグラフからそのプロパティを検索する手法が、もっとも適合的であるように見える。

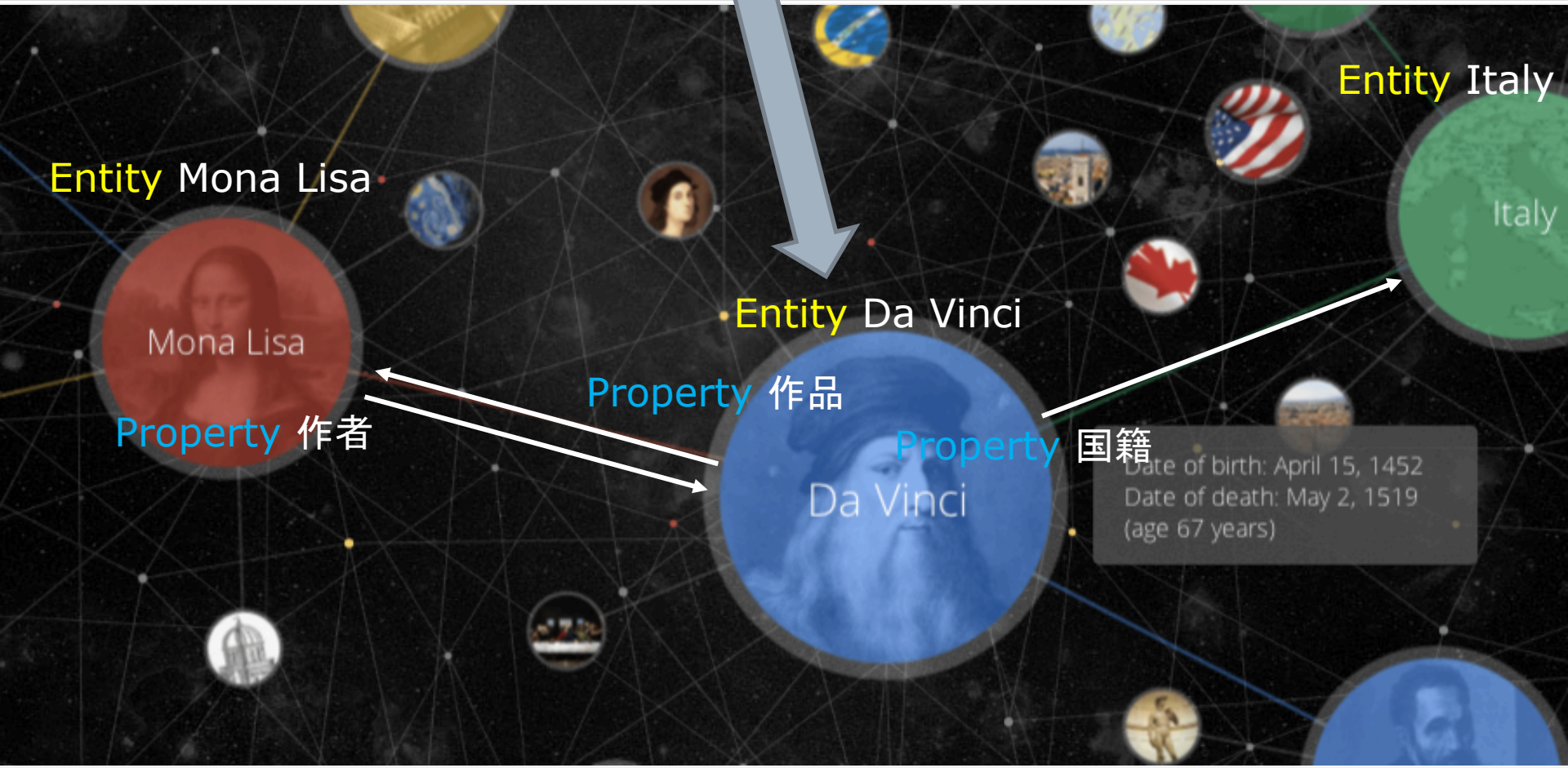
グラフ検索のスキーマには、Schema.org(http://schema.org/)のスキーマが、グローバルでは広く使われているのだが、日本では、どうなのだろう？

検索技術を巡る、彼我の技術格差は、残念ながら、とても大きいのだ。

Google Nowの検索の特徴
グラフ検索とEntity/Propertyモデル

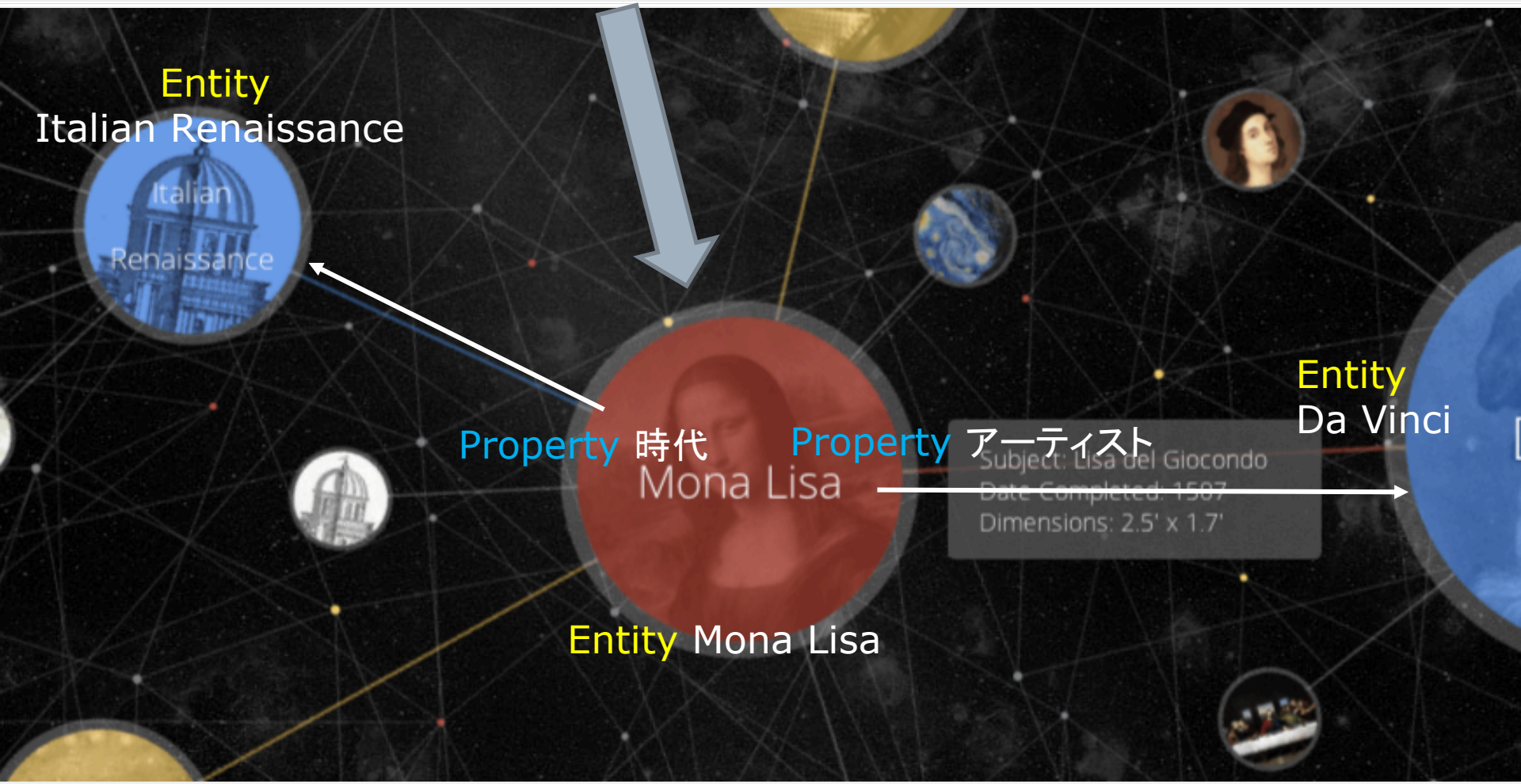
- 一般には、Googleの検索は、与えられた文字列を、Googleが収集した膨大なデータから探し出し、検索結果を独自のアルゴリズムで順序づけて、その文書のURLを返すものである。
- ただ、Google Nowで行われている検索は、それとは異なっている。それは、後で見るような **Entity / Property モデル** に基づいて構築された巨大なグラフデータを検索して、マッチしたEntityのPropertyの値を返す。
- それをGoogleは、**Knowledge Graph**と呼んでいる。この検索スタイルの変化は、2010年代に行われた。

Da Vinci



Google Knowkedge Graph検索

Mona Lisa



Google Knowkedge Graph検索

Italian Renaissance



Google Knowkedge Graph検索

モナ・リザ - Wikipedia

https://ja.wikipedia.org/wiki/モナ・リザ

モナ・リザ』（伊: La Gioconda、仏: La Joconde）は、イタリアの美術家レオナルド・ダ・ヴィンチが描いた油彩画。上半身のみが描かれた女性の肖像画で、「世界でもっとも知られた、もっとも見られた、もっとも書かれた、もっとも歌われた、もっともパロディ作品が ...

アイルワースのモナ・リザ · レオナルド・ダ・ヴィンチ · リザ・デル・ジョコンド



評価 時間 並べ替え

モナリザ 恵比寿店 （MONNA LISA）

4.4 ★★★★½ (61) · フランス料理

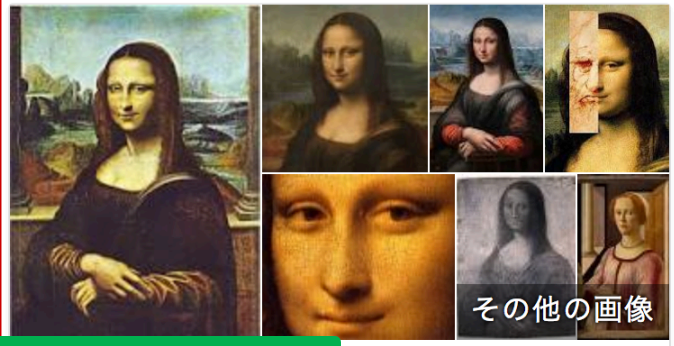
国内外の有名店での経験を持つシェフの本格フレンチ

7.1 km · 恵比寿西1-14-4 シティー・ホームズ恵比寿 1 F 恵比寿 駅 徒歩3分

終了: 23:00

モナリザ 丸の内店 （MONNA LISA）

Property



Entity

モナ・リザ

絵画作品

『モナ・リザ』は、イタリアの美術家レオナルド・ダ・ヴィンチが描いた油彩画。上半身のみが描かれた女性の肖像画で、「世界でもっとも知られた、もっとも見られた、もっとも書かれた、もっとも歌われた、もっともパロディ作品が作られた美術作品」といわれている。

ウィキペディア

アーティスト： レオナルド・ダ・ヴィンチ

寸法： 77 cm x 53 cm

展示場所： ルーヴル美術館 (1797年から)

個の素材： 油絵具

題材： リザ・デル・ジョコンド

時代・様式： ルネサンス、 盛期ルネサンス

Google Knowkedge Graph検索

Google Nowの検索の問題

- Google NowのKnowkedge Graph検索の問題は、それが、必ずしも網羅的ではないことだ。従来の文字列検索の方が、圧倒的にデータ量が大きい。限られたEntityしか、グラフ上に存在していない。Googleの音声検索では、グラフ上にデータがないと、音声での読み上げを断念し、従来の文字列検索の結果を表示する。
- 先の「ボルツマンの論争相手」では、ボルツマンというEntityは存在するのだが(「誕生日」は答えられた)、そのEntityには、「論争相手」というPropertyが定義されていないことを示している。
- Knowkedge Graph検索の大きな問題は、それにアクセスするAPIが公開されているのだが、その機能が大きく制限されていること。Taylor SwiftのEntityには誕生日が登録されているのだが、公開されているAPIからはアクセスできない。

Google Knowledge Graph Search API

https://developers.google.com/knowledge-graph/

KnowledgeGraphのEntityにアクセスできるが、その機能は、限られている。

Google Knowledge Graph Search API

- The Knowledge Graph Search API lets you find entities in the Google Knowledge Graph. The API uses standard schema.org types and is compliant with the JSON-LD specification.

- “Taylor Swift” についての検索サンプル

```
https://kgsearch.googleapis.com/v1/entities:search?query=taylor+swift&key=API_KEY&limit=1&indent=True
```

```
{
 "@context": {
   "@vocab": "http://schema.org/",
   "goog": "http://schema.googleapis.com/",
   "resultScore": "goog:resultScore",
   "detailedDescription": "goog:detailedDescription",
   "EntitySearchResult": "goog:EntitySearchResult",
   "kg": "http://g.co/kg"
 },
 "@type": "ItemList",
 "itemListElement": [
   {
     "@type": "EntitySearchResult",
     "result": {
       "@id": "kg:/m/0dl567",
       "name": "Taylor Swift",
       "@type": [
         "Thing",
         "Person"
       ],
       "description": "Singer-songwriter",
```

```
"image": {
        "contentUrl":
```



```
      "image": {
        "contentUrl":
"https://t1.gstatic.com/images?q=tbn:ANd9GcQmVDAhjhWnN2OWys2ZMO3PGAhupp5tN2LwF_BJmiHgi19hf8Ku",
        "url": "https://en.wikipedia.org/wiki/Taylor_Swift",
        "license": "http://creativecommons.org/licenses/by-sa/2.0"
      },
      "detailedDescription": {
        "articleBody": "Taylor Alison Swift is an American singer-songwriter and actress. Raised in Wyomissing, Pennsylvania, she moved to Nashville, Tennessee, at the age of 14 to pursue a career in country music. ",
        "url": "http://en.wikipedia.org/wiki/Taylor_Swift",
        "license":
"https://en.wikipedia.org/wiki/Wikipedia:Text_of_Creative_Commons_Attribution-ShareAlike_3.0_Unported_License"
      },
      "url": "http://taylorswift.com/"
    },
    "resultScore": 896.576599
  }
 ]
}
```

Sample Code in Python

```
"""Example of Python client calling Knowledge Graph Search API."""
import json
import urllib

api_key = open('.api_key').read()
query = 'Taylor Swift'
service_url = 'https://kgsearch.googleapis.com/v1/entities:search'
params = {
    'query': query,
    'limit': 10,
    'indent': True,
    'key': api_key,
}
url = service_url + '?' + urllib.urlencode(params)
response = json.loads(urllib.urlopen(url).read())
for element in response['itemListElement']:
  print element['result']['name'] + ' (' + str(element['resultScore']) + ')'
```

Sample Code in JavaScript

```
<!DOCTYPE html>
<html>
<head>
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"></s
cript>
</head>
<body>
<script>
  var service_url = 'https://kgsearch.googleapis.com/v1/entities:search';
  var params = {
    'query': 'Taylor Swift',
    'limit': 10,
    'indent': true,
    'key' : '<put your api_key here>',
  };
  $.getJSON(service_url + '?callback=?', params, function(response) {
    $.each(response.itemListElement, function(i, element) {
      $('<div>',
{text:element['result']['name']}).appendTo(document.body);
    });
  });
</script>
</body>
</html>
```

Alexa と Google Now

Virtual Assistant Jeopardy

See how four digital helpers performed when asked trivia from the official 'Jeopardy' practice test:









Virtual Assistant (Score out of 20)	Alexa (5)	Siri (17)	Cortana (16)	Google Now (16)
元素番号98番は何？	Says: "Hmmm, I can't find the answer to the question I heard."	Shows a periodic table illustration of Californium.	Bing search results include Californium Wikipedia entry, if you scroll down a screen. No point scored.	Says: "...Californium is a radioactive metallic chemical element with symbol Cf and atomic number 98."
モーツァルトの最後の交響曲は？	Gives the wrong answer: "Symphony No. 49 in D Major."	Shows search results, the first of which is "Mozart's Last Symphony: The Giant 'Jupiter.'"	Shows search results, the first of which is "Mozart's Last Symphony: The Giant 'Jupiter.'"	Says: "...the work is nicknamed the Jupiter Symphony."
映画「ベンジャミン・バトン」でデイジーを演じたのは？	Says: "Benjamin Button's actors are Brad Pitt and Cate Blanchett."	Says: "'The Curious Case of Benjamin Button' stars Brad Pitt, Cate Blanchett and Taraji Henson."	Bing search results includes Cate Blanchett link...way way down the list. No point scored.	Says: "...Cate Blanchett, who plays Daisy, Benjamin's great love."

Crtana と Google Now

Virtual Assistant Jeopardy

See how four digital helpers performed when asked trivia from the official 'Jeopardy' practice test:









Virtual Assistant (Score out of 20)	Alexa (5)	Siri (17)	Cortana (16)	Google Now (16)
元素番号98番は何？	Says: "Hmmm, I can't find the answer to the question I heard."	Shows a periodic table illustration of Californium.	Bing search results include Californium Wikipedia entry, if you scroll down a screen. No point scored.	Says: "...Californium is a radioactive metallic chemical element with symbol Cf and atomic number 98."
モーツァルトの最後の交響曲は？	Gives the wrong answer: "Symphony No. 49 in D Major."	Shows search results, the first of which is "Mozart's Last Symphony: The Giant 'Jupiter.'"	Shows search results, the first of which is "Mozart's Last Symphony: The Giant 'Jupiter.'"	Says: "...the work is nicknamed the Jupiter Symphony."
映画「ベンジャミン・バトン」でデイジーを演じたのは？	Says: "Benjamin Button's actors are Brad Pitt and Cate Blanchett."	Says: "'The Curious Case of Benjamin Button' stars Brad Pitt, Cate Blanchett and Taraji Henson."	Bing search results includes Cate Blanchett link...way way down the list. No point scored.	Says: "...Cate Blanchett, who plays Daisy, Benjamin's great love."

Virtual Assistant Jeopardy

Siri！

See how four digital helpers performed when asked trivia from the official 'Jeopardy' practice test:






Virtual Assistant (Score out of 20)	Alexa (5)	Siri (17)	Cortana (16)	Google Now (16)
元素番号98番は何？	Says: "Hmmm, I can't find the answer to the question I heard."	Shows a periodic table illustration of Californium.	Bing search results include Californium Wikipedia entry, if you scroll down a screen. No point scored.	Says: "...Californium is a radioactive metallic chemical element with symbol Cf and atomic number 98."
モーツァルトの最後の交響曲は？	Gives the wrong answer: "Symphony No. 49 in D Major."	Shows search results, the first of which is "Mozart's Last Symphony: The Giant 'Jupiter.'"	Shows search results, the first of which is "Mozart's Last Symphony: The Giant 'Jupiter.'"	Says: "...the work is nicknamed the Jupiter Symphony."
映画「ベンジャミン・バトン」でデイジーを演じたのは？	Says: "Benjamin Button's actors are Brad Pitt and Cate Blanchett."	Says: "'The Curious Case of Benjamin Button' stars Brad Pitt, Cate Blanchett and Taraji Henson."	Bing search results includes Cate Blanchett link...way way down the list. No point scored.	Says: "...Cate Blanchett, who plays Daisy, Benjamin's great love."

「GoogleがAppleに検索のため3年で3倍増の年間3000億円超えを支払っているのにウィンウィンの理由は？」 Gigazine **2017/8/15**

2017年08月15日 15時00分00秒

GoogleがAppleに検索のため3年で3倍増の年間3000億円超えを支払っているのにウィンウィンの理由は？



Androidを開発するGoogleはiOSを開発するAppleのライバルですが、iOSのデフォルト検索エンジンになるためにGoogleは、なんと30億ドル(約3300億円)という破格の契約金を支払っていると報じられています。それにもかかわらず、この契約はGoogle・Appleにとってウィン・ウィンの関係だそうです。

Google paying Apple $3 billion to remain default search - Bernstein
https://www.cnbc.com/2017/08/14/google-paying-apple-3-billion-to-remain-default-search--bernstein.html

https://goo.gl/DCcM9V

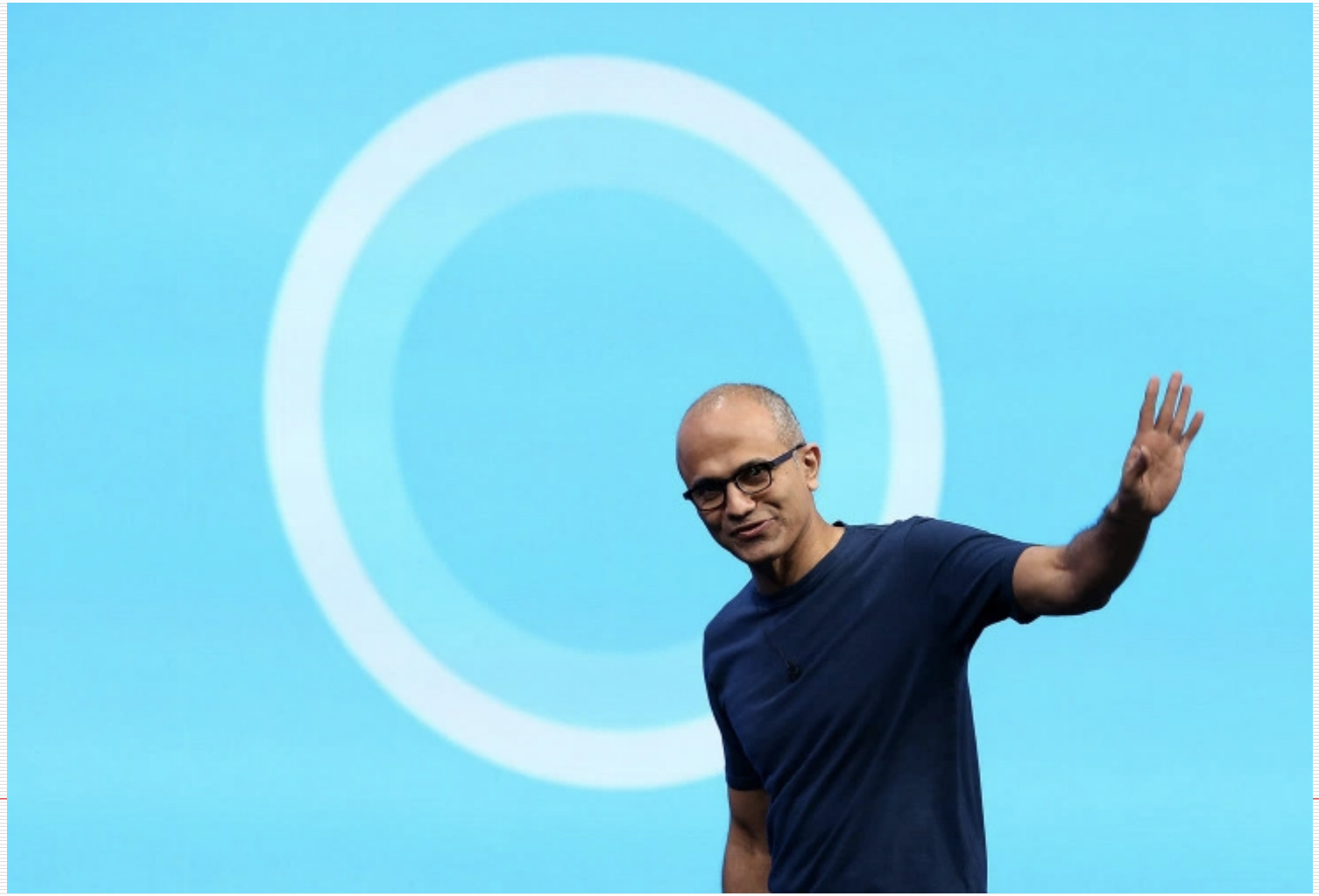
「GoogleがAppleに検索のため3年で3倍増の年間3000億円超えを支払っているのにウィンウィンの理由は？」 Gigazine **2017/8/15**

「アライアンス・バーンスタインのアナリストのA.M. Sacconaghi Jr.が、「GoogleがiOSのデフォルトの検索エンジンに指定してもらうためにAppleに支払うライセンスフィーが、年間で30億ドル(約3300億円)である」との予測を投資家向けの報告書内で明かしました。 https://goo.gl/dCkiFq

iOSのブラウザSafariの検索エンジンは、Google、Yahoo、Bing、DuckDockGoから選べますが、デフォルトでは「Google」にチェックが付いており、検索エンジンとしてGoogleを使う事になります。もちろんユーザーでGoogle以外の検索エンジンを指定することは可能なので、Googleはこのデフォルト検索のチェックをつけてもらうためにAppleに年間3000億円以上の金銭を支払っている、というわけです。」

https://goo.gl/DCcM9V

「AmazonとMicrosoft、AlexaとCortanaの会話で合意――両社の人工知能アシスタントの互換性増大へ」 **2017/8/30**



- **Microsoftのサティヤ・ナデラ**:「あらゆる状況、あらゆるデバイスを通じてユーザーがCortanaを利用できるようにすることにわれわれは高い優先順位を与えている。Office 365との統合を含めたCortanaの知識がAlexaから利用できるようになるのはこの目標に向かっての大きな前進だ。」
- **Amazonのジェフ・ベゾス**:「世界は巨大で多様な場所だ。そこでインテリジェント・アシスタントの勝者も一つではないはず。それぞれの人工知能は独自の知識と得意とする能力を持つことになるだろう。有力な人工知能が相互に連携できるようになることはユーザー体験を豊かなものとし、有用性をいっそう増すだろう。われわれのEchoがCortanaに容易にアクセスができるようになるというのは素晴らしいニュースだ。」

- また声明ではユーザーが両社の人工知能の相互運用性を得られることがウィン-ウィンの提携である所以も説明されている。

https://goo.gl/ZKC6VZ

マイクロソフト、アマゾンと協業で深層学習をすべての開発者に解放～『Gluon』の発表

Rate this article ★★★★★

Microsoft AI Japan Team 2017年10月13日

Share 245 53 22

[2017 年 10 月 12 日]





The Gluon API Specification

The Gluon API specification is an effort to improve speed, flexibility, and accessibility of deep learning technology for all developers, regardless of their deep learning framework of choice. The Gluon API offers a flexible interface that simplifies the process of prototyping, building, and training deep learning models without sacrificing training speed. It offers four distinct advantages:

- **Simple, Easy-to-Understand Code**: Gluon offers a full set of plug-and-play neural network building blocks, including predefined layers, optimizers, and initializers.
- **Flexible, Imperative Structure**: Gluon does not require the neural network model to be rigidly defined, but rather brings the training algorithm and model closer together to provide flexibility in the development process.
- **Dynamic Graphs**: Gluon enables developers to define neural network models that are dynamic, meaning they can be built on the fly, with any structure, and using any of Python's native control flow.
- **High Performance**: Gluon provides all of the above benefits without impacting the training speed that the underlying engine provides.

https://github.com/gluon-api/gluon-api/

Larry Pageの人工知能観

Larry Page on AI



Larry Page on AI

- 「Googleの検索エンジンが、AIによって完全なものになったときにのみ、Googleのミッションは、完遂されるだろう。あなたたちは、それが何を意味するのか知っている。それが人工知能なのだ。」
- 「人工知能は、Googleの最終バージョンになるだろう。Web上のすべてのものを理解するだろう究極の検索エンジンは、あなたが望むものを正確に理解するだろうし、あなたに正しいものを与えるだろう。我々は、今は、そうしたことをするのには、遠いところにいる。ただ、我々は、少しずつ、それに近ずくことはできる。我々が取り組んでいることは、基本的には、そのことなのだ。」

http://goo.gl/OEL1oC

Larry Page on AI

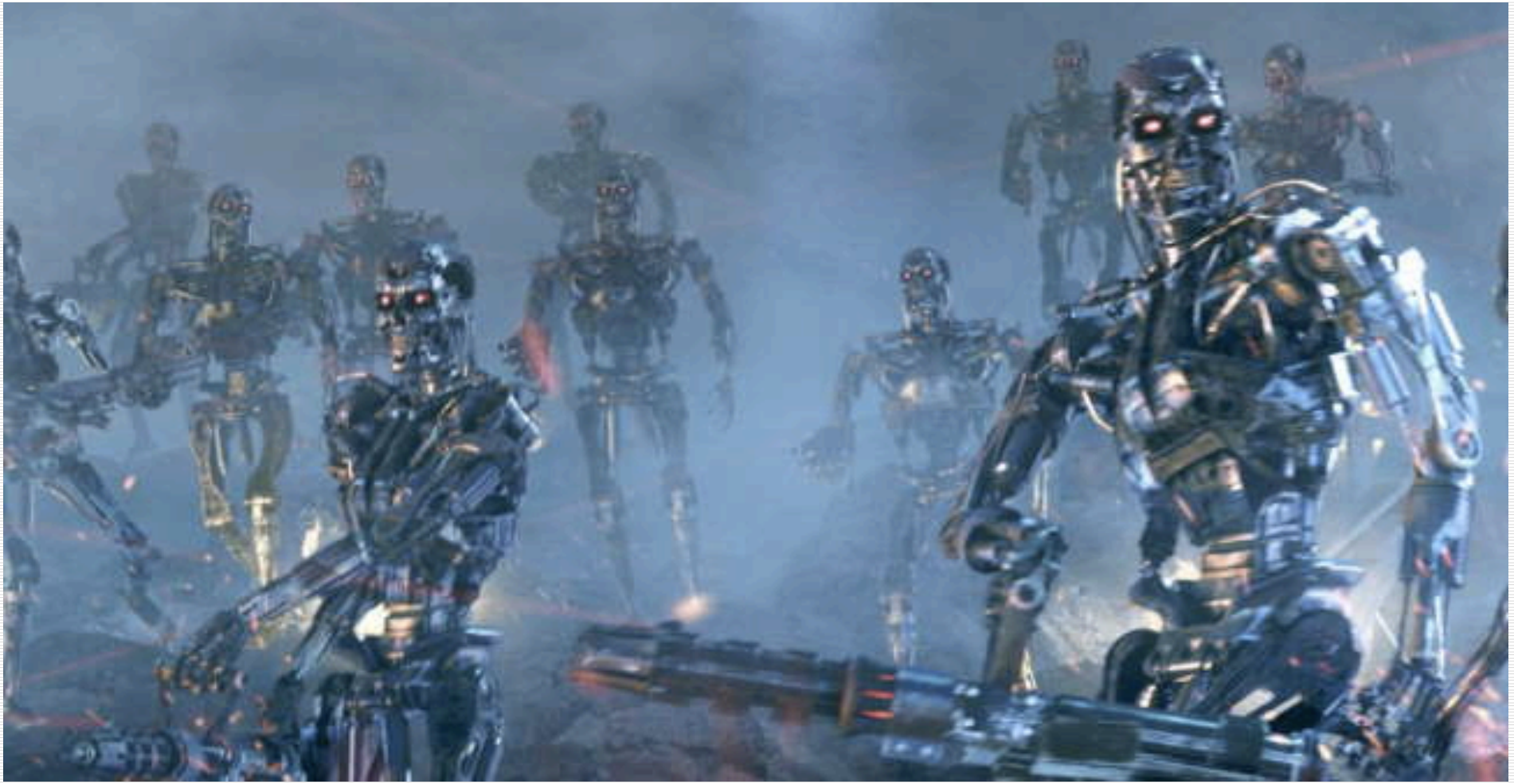
- 「検索における我々の大きな目標は、人が望むものを、実際に正確に理解し、世界のすべてのものを理解することである。コンピューター科学者として、我々は、それを人工知能と呼ぶ。」

http://goo.gl/OEL1oC

“Elon Musk lives in fear of Google's killer robot army”



“Elon Musk lives in fear of Google's killer robot army”



検索グラフの構造と Entityモデル

検索グラフの構造とボキャブラリーは、スキーマによって定義されている。それをEntityモデルと呼ぶ。

ここでは、GoogleのKnowledge Graphで採用されている、Schema.org のスキーマの特徴を見てみよう。

知識のグラフでの表現と検索

- Knowledge Graph は、”people”, “places”, “things” といった実世界のエンティティを記述する数百万のエンティティを持っている。これらのエンティティが、このグラフのノードを形成する。
- これらのエンティティの「ノード」と、ノード間の関係あるいはアクションを表す「エッジ」のボキャブラリーは、次に見る組織 Schema.orgによって、与えられている。

知識のグラフでの表現と検索

- 例えば、「アインシュタインの誕生日」を知ろうとすれば、以前の文字列ベースの検索では、「アインシュタイン」を検索して、検索エンジンが返す膨大な数の（PageRankで順序付けされているのだが）文字列「アインシュタイン」を含むURLのドキュメントを、検索者が確認して、その中で彼の誕生日の情報を見つける必要がある。
- グラフ・ベースの検索では、Person Entityノードであるアインシュタインの Property **birthdate** の値をしらべれば、それはすぐにわかる。
- 同様に、Person Entityのもつ Propertyである、award, colleague, jobTitle, worksFor などの値を取得すれば、もっと多くの情報が簡単に得られる。

知識のグラフでの表現と検索

- Google Nowの場合なら、イベントの予約先から送られたメールの中から、予約番号、予約者名、イベントの日時・場所等の情報（ただし、Event EntityのPropertyに含まれるものに限る）を、得ることが出来る。
- FacebookのGraph Searchなら、ソーシャル・グラフのノードをたどることで、「サンフランシスコに住んでいる私の友達は誰？」といった質問に答えることができる。

Welcome to Schema.org

http://schema.org/
https://github.com/schemaorg/schemaorg

Schema.orgのスキーマは、Google, Microsoft, Pinterest, Yandex等で使われている。Facebookは、そうではない。

Welcome to Schema.org

- Schema.org は、インターネットやWebページ、e-メールのメッセージやさらにもっと多くのものの上にある構造化されたデータに対して、スキーマを作りあげ、維持し、それを広めるというミッションを持った共同のコミュニティ活動である。
- Schema.org は、**Google, Microsoft, Yahoo, Yandex** のスポンサー支援を得ている。そのボキャブラリーは、メーリングリスト public-schemaorg @w3.org と GitHub https://goo.gl/WvCyP3 を用いて、オープンなプロセスで開発されている。

folksonomyアプローチ

http://schema.org/

Schema.org vocabulary

- Schema.org ボキャクラリーは、RDFa, Microdata, JSON-LD を含む多くのエンコーディングで利用が可能である。
- このボキャブラリーは、エンティティとエンティティ間の関係とアクションをカバーしており、また、よくドキュメント化されている拡張モデルを通じて、容易に拡張できる。
- 一千万以上のサイトが、Webページやe-メールのメッセージをマークアップするのに Schema.orgを利用している。Google, Microsoft, Pinterest, Yandexその他の多くのアプリケーションが、豊かで拡張可能なユーザー経験に力を与えるために、すでに、これらのボキャブラリーを使っている。

エンティティの型(Type)のサンプル

Types = “is-a”

- Book
- BookSeries
- EducationalOrganization
- Event
- GovernmentOrganization
- LocalBusiness
- Movie
- MovieSeries
- MusicAlbum
- MusicGroup
- MusicRecording
- Organization
- Periodical
- Person
- Place
- SportsTeam
- TVEpisode
- TVSeries
- VideoGame
- VideoGameSeries
- WebSite



is-a → Person /people/person

is-a → Film Actor /film/actor

is-a → Politician /government/politician

One entity, many types.

一つのエンティティは、多くの型を持つ

Entityは階層を持つ
schema.org での Entityの階層の例

- Thing > CreativeWork > Book
- Thing > CreativeWork > CreativeWorkSeries > BookSeries
- Thing > Organization > EducationalOrganization
- Thing > Event
- Thing > Organization > GovernmentOrganization
- Thing > Organization > LocalBusiness
  Thing > Place > LocalBusiness
- Thing > CreativeWork > Movie

Entityは階層を持つ
schema.org での Entityの階層の例

- Thing > CreativeWork > CreativeWorkSeries > MovieSeries
- Thing > CreativeWork > MusicPlaylist > MusicAlbum
- Thing > Organization > PerformingGroup > MusicGroup
- Thing > CreativeWork > MusicRecording
- Thing > Organization
- Thing > CreativeWork > CreativeWorkSeries > Periodical
- Thing > Person

Entityは階層を持つ
schema.org での Entityの階層の例

- Thing > Place
- Thing > Organization > SportsOrganization > SportsTeam
- Thing > CreativeWork > Episode > TVEpisode
- Thing > CreativeWork > CreativeWorkSeries > TVSeries
  Thing > CreativeWork > TVSeries
- Thing > CreativeWork > Game > VideoGame
  Thing > CreativeWork > SoftwareApplication > VideoGame

schemaorg/data/schema.rdfa

Schema.org core schema

This is an RDFa-based representation of the schema.org schema, the underlying representation of the schema.org vocabulary.

It is represented in a form based on W3C RDF/RDFS. We encourage proposals for schema.org improvements to be expressed in this same style. See the how we work document for information on collaboration.

See datamodel for more details, the developers page for alternate formats.

Note: the style of RDFa used here may change in the future. To see the substantive content of the schema, view the HTML source markup. We use a simple subset of RDFa for syntax, including prefixes that are declared in the RDFa initial context. We also use Markdown syntax within comments to make hypertext documentation easier to edit. Other formats are available.

Thing
The most generic type of item.

CreativeWork
The most generic kind of creative work, including books, movies, photographs, software programs, etc.
Subclass of: Thing
Source: rNews

WebPage
A web page. Every web page is implicitly assumed to be declared to be of type WebPage, so the various properties about that webpage, such as <code>breadcrumb</code> may be used. We recommend explicit declaration if these properties are specified, but if they are found outside of an itemscope, they will be assumed to be about the page.
Subclass of: CreativeWork

AboutPage

https://github.com/schemaorg/schemaorg

schemaorg/data/schema.rdfa

AboutPage
Web page type: About page.
Subclass of: WebPage

Organization
An organization such as a school, NGO, corporation, club, etc.
Subclass of: Thing

Place
Entities that have a somewhat fixed, physical extension.
Subclass of: Thing

LocalBusiness
A particular physical business or branch of an organization. Examples of LocalBusiness include a restaurant, a particular branch of a restaurant chain, a branch of a bank, a medical practice, a club, a bowling alley, etc.
Subclass of: Organization
Subclass of: Place

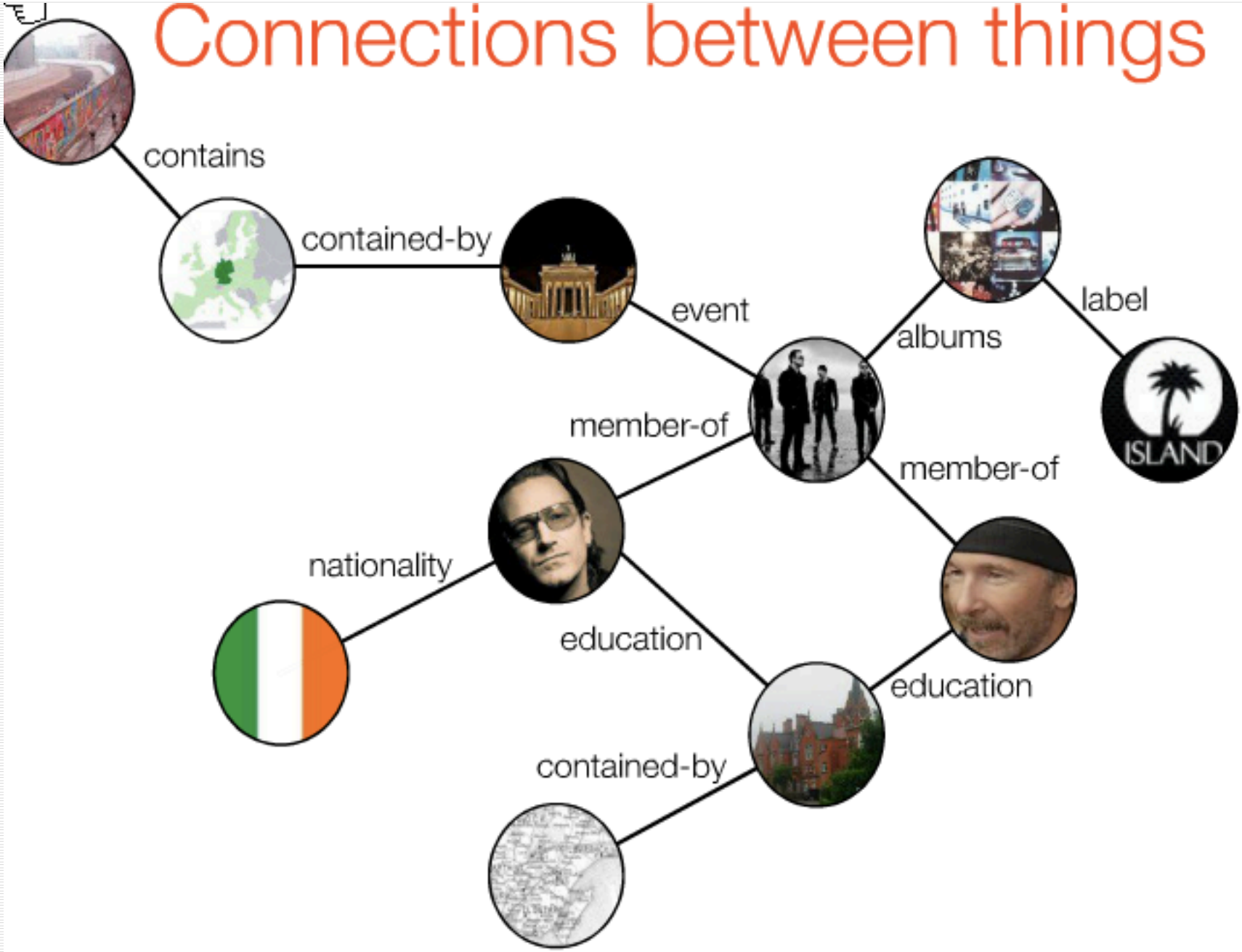
MedicalOrganization
A medical organization (physical or not), such as hospital, institution or clinic.
Subclass of: Organization

Dentist
A dentist.
Subclass of: MedicalOrganization
Subclass of: LocalBusiness

Hospital
A hospital.
Subclass of: CivicStructure
Subclass of: EmergencyService

https://github.com/schemaorg/schemaorg

Connections between things



schema.org でのPersonのProperty

Properties = “has-a”

- gender
- birthdate
- birthplace
- profession
- nationality
- ethnicity
- parents
- children
- religion
- education
- jobTitle
- **follows**
- **knows**
- **owns**
- **seeks**

... and more

Action

PersonのPropery

- Person: additionalName, address, affiliation, alumniOf, award, awards, birthDate, children, colleague, colleagues, contactPoint, contactPoints, deathDate, email, familyName, faxNumber, follows, gender, givenName, homeLocation, honorificPrefix, honorificSuffix, interactionCount, jobTitle, knows, memberOf, nationality, parent, parents, performerIn, relatedTo, sibling, siblings, spouse, telephone, workLocation, worksFor, ...

The Type Hierarchy
http://schema.org/docs/full.html

PersonのProperty

Properties = “has-a”

Property	Expected Type	Description
Properties from Person		
additionalName	Text	An additional name for a Person, can be used for a middle name.
address	PostalAddress or Text	Physical address of the item.
affiliation	Organization	An organization that this person is affiliated with. For example, a school/university, a club, or a team.
alumniOf	EducationalOrganization or Organization	An organization that the person is an alumni of. Inverse property: alumni.
award	Text	An award won by or for this item. Supersedes awards.
birthDate	Date	Date of birth.
birthPlace	Place	The place where the person was born.
brand	Brand or Organization	The brand(s) associated with a product or service, or the brand(s) maintained by an organization or business person.
children	Person	A child of the person.
colleague	Person or URL	A colleague of the person. Supersedes colleagues.
contactPoint	ContactPoint	A contact point for a person or organization. Supersedes contactPoints.
deathDate	Date	Date of death.
deathPlace	Place	The place where the person died.
duns	Text	The Dun & Bradstreet DUNS number for identifying an organization or business person.

http://schema.org/Person

email	Text	Email address.
familyName	Text	Family name. In the U.S., the last name of an Person. This can be used along with givenName instead of the name property.
faxNumber	Text	The fax number.
follows	Person	The most generic uni-directional social relation.
funder	Organization or Person	A person or organization that supports (sponsors) something through some kind of financial contribution.
gender	GenderType or Text	Gender of the person. While http://schema.org/Male and http://schema.org/Female may be used, text strings are also acceptable for people who do not identify as a binary gender.
givenName	Text	Given name. In the U.S., the first name of a Person. This can be used along with familyName instead of the name property.
globalLocationNumber	Text	The Global Location Number (GLN, sometimes also referred to as International Location Number or ILN) of the respective organization, person, or place. The GLN is a 13-digit number used to identify parties and physical locations.
hasOfferCatalog	OfferCatalog	Indicates an OfferCatalog listing for this Organization, Person, or Service.
hasPOS	Place	Points-of-Sales operated by the organization or person.
height	Distance or QuantitativeValue	The height of the item.
homeLocation	ContactPoint or Place	A contact location for a person's residence.
honorificPrefix	Text	An honorific prefix preceding a Person's name such as Dr/Mrs/Mr.
honorificSuffix	Text	An honorific suffix preceding a Person's name such as M.D. /PhD/MSCSW.
isicV4	Text	The International Standard of Industrial Classification of All Economic Activities (ISIC), Revision 4 code for a particular organization, business person, or place.
jobTitle	Text	The job title of the person (for example, Financial Manager).
knows	Person	The most generic bi-directional social/work relation.

http://schema.org/Person

makesOffer	Offer	A pointer to products or services offered by the organization or person. Inverse property: offeredBy.
memberOf	Organization or ProgramMembership	An Organization (or ProgramMembership) to which this Person or Organization belongs. Inverse property: member.
naics	Text	The North American Industry Classification System (NAICS) code for a particular organization or business person.
nationality	Country	Nationality of the person.
netWorth	MonetaryAmount or PriceSpecification	The total financial value of the person as calculated by subtracting assets from liabilities.
owns	OwnershipInfo or Product	Products owned by the organization or person.
parent	Person	A parent of this person. Supersedes parents.
performerIn	Event	Event that this person is a performer or participant in.
relatedTo	Person	The most generic familial relation.
seeks	Demand	A pointer to products or services sought by the organization or person (demand).
sibling	Person	A sibling of the person. Supersedes siblings.
sponsor	Organization or Person	A person or organization that supports a thing through a pledge, promise, or financial contribution. e.g. a sponsor of a Medical Study or a corporate sponsor of an event.
spouse	Person	The person's spouse.
taxID	Text	The Tax / Fiscal ID of the organization or person, e.g. the TIN in the US or the CIF/NIF in Spain.
telephone	Text	The telephone number.
vatID	Text	The Value-added Tax ID of the organization or person.
weight	QuantitativeValue	The weight of the product or person.
workLocation	ContactPoint or Place	A contact location for a person's place of work.
worksFor	Organization	Organizations that the person works for.

http://schema.org/Person

ボイス・アシスタント・システムのプログラミング



ボイス・アシスタントの開発モデル

- Speech2TextとText2Speech
- Voice Command
- Intent Model
- Intent + Slot Model
- Intent Signature / Entity Model

ボイス・アシスタントの開発モデルの変化

- ここでは、ボイス・アシスタント・システムの実装とプログラミングについて見ることにしよう。
- 一年前、マルレクで「パーソナル・アシスタント・システム」を取り上げ、その中で、当時のボイス・アシスタント・システムの実装とプログラミングを調べたことがある。
- わずか一年の間にも、ボイス・アシスタントの開発のスタイルは、大きく変わった。ここでは、どのような方向で「変化」してきたのかよくわかるように、過去のスタイルも取り上げようと思う。
- それゆえ、ここで取り上げた開発モデルのいくつかには、もう使われることのない古いものも含まれている。
- それらの開発モデルは、構造的には、プリミティブなものからより複雑なものへと階層をなしているのだが、その階層はまた、時間の中での「変化」を反映している。それは、必然的で合理的な「進化」だと、僕は考えている。

Speech2TextとText2Speechの利用

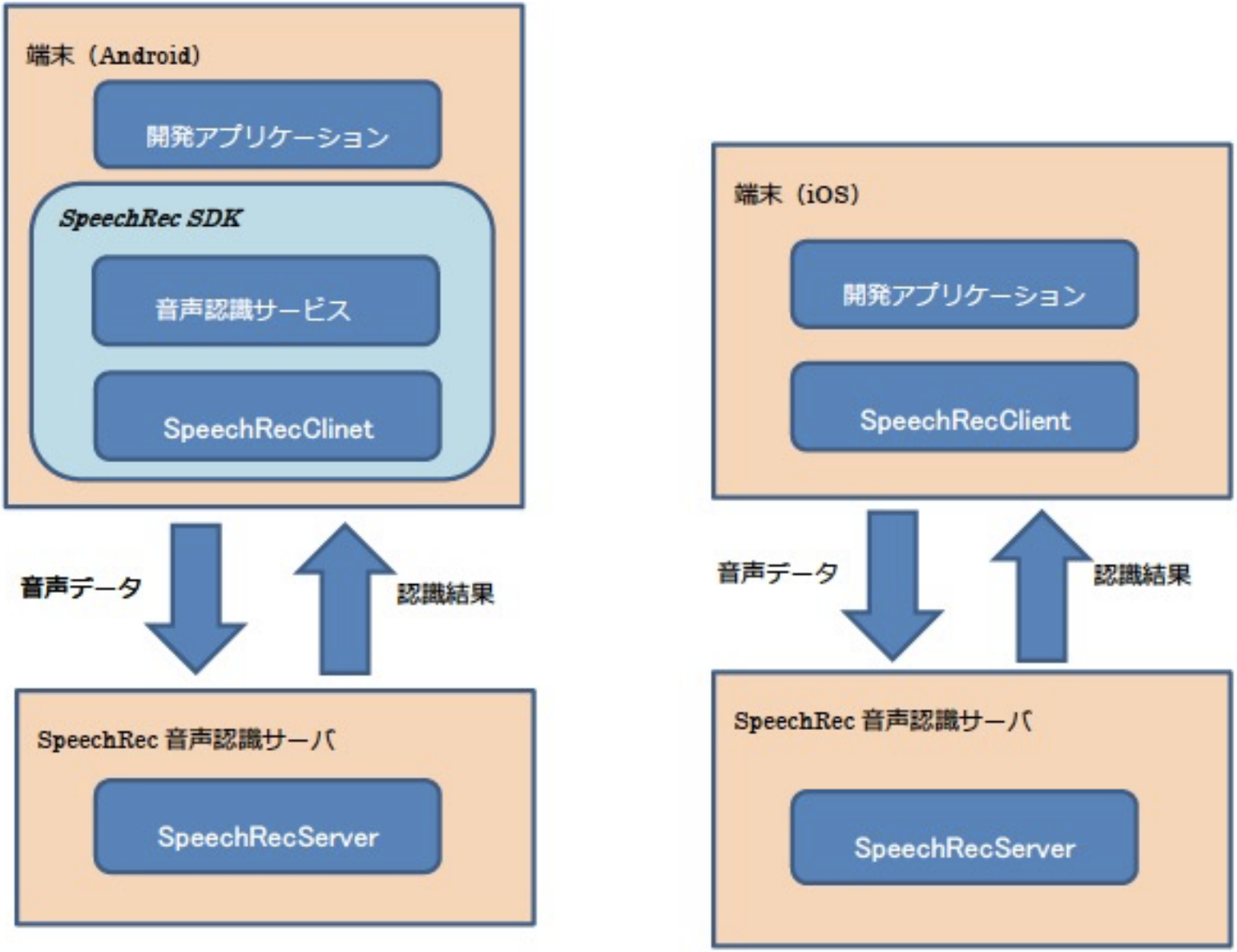
Speech2TextとText2Speechの利用

- 入出力のボイス化の、もっともプリミティブなモデルである。
- 音声入力はSpeech2Textで入力文字列に変換され、出力文字列はText2Speechで、音声出力に変換される。
- 受け取った入力文字列に対し、どのような出力文字列を生成するかは、プログラマが考える。
- もっとも、人間が「ああ言われれば、こう言う」という、基本的な「シナリオ」を考える構造は、ほとんどすべてのボイス・アシスタントのプログラミングにおいて共通である。

SpeechRec SDK でのプログラミング

SpeechRec SDK https://goo.gl/aWRNQJ から




端末（Android）
開発アプリケーション
SpeechRec SDK
音声認識サービス
SpeechRecClinet
音声データ
認識結果
SpeechRec 音声認識サーバ
SpeechRecServer
端末（iOS）
開発アプリケーション
SpeechRecClient
音声データ
認識結果
SpeechRec 音声認識サーバ
SpeechRecServer

```
// Divide関連ファイルを端末内に展開
DivideFileManager divideFileManager =
    new DivideFileManager(this);
if (!divideFileManager.isExtracted()) {
  try {
    divideFileManager.extract();
  } catch (IOException e) {
    ……..
  }
}
// 音声認識サービスと接続
helper.connect(this, this);
```

```
public void onClickStart(final View view) {
  Intent intent = new Intent(this, RecognitionActivity.class);
  // 別途発行されるAPIキーを設定してください（以下の値はダミーです）
  intent.putExtra(RecognitionActivity.KEY_API_KEY, "123456789");
  startActivityForResult(intent, RECOGNIZE_ACTIVITY_REQUEST_ID);
}
```

```
bundle = new Bundle();
// Bundleにインテントの値（区間検出環境）を追加
Intent intent = getIntent();
bundle.putAll(intent.getExtras());
// Bundleに区間検出モデルファイルを追加
bundle.putString(KEY_VAD_MODEL,
                 divideFileManager.getDivideModelPath());
```

```
@Override
public void onServiceConnected() {
  // 音声認識処理を開始
  helper.startRecognition(bundle);
}
```

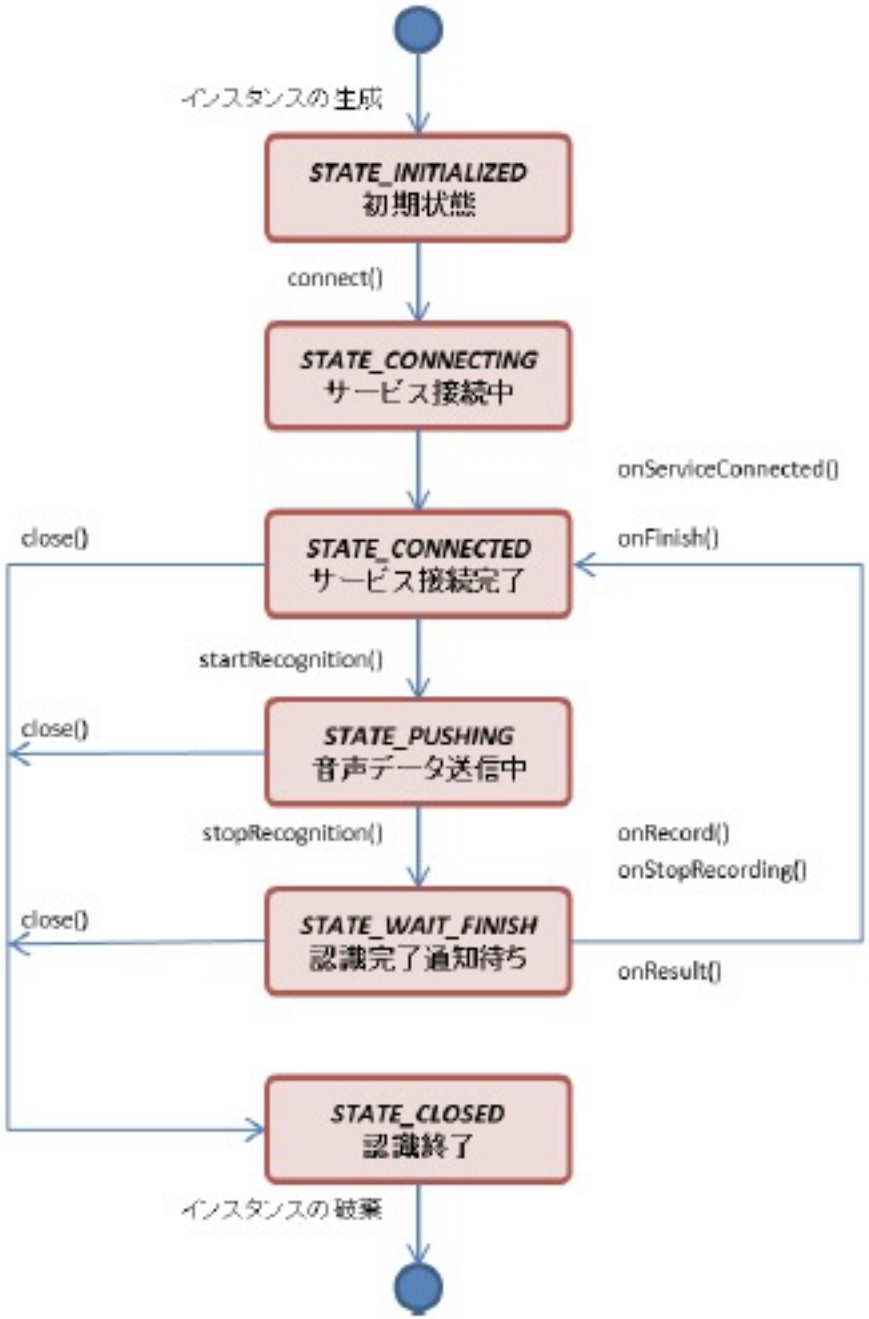
```
public void onClickFinishButton(View view) {
  // 音声認識処理を終了
  helper.stopRecognition();
}
```

```java
@Override
public void onResult(Nbest result) {
  List<Sentence> sentenceList = result.getSentenceList();
  if (resultList == null) {
    resultList = new LinkedList<StringBuilder>();
    for (int i = 0; i < sentenceList.size(); i++) {
      resultList.add(new StringBuilder());
    }
  } else {
    while (resultList.size() > sentenceList.size()) {
      resultList.removeLast();
    }
  }
```

```
for (int i = 0; i < resultList.size(); i++) {
  StringBuilder sb = resultList.get(i);
  Sentence sentence = sentenceList.get(i);
  for (Word word : sentence.getWordList()) {
    String label = word.getLabel();
    // ラベルがnullは無視
    if (label == null) {
      continue;
    }
    // セミコロン以降、空白のみは無視
    label = label.replaceAll(";.*", "").replaceAll("[　]", "");
```

```
      // ラベルが空文字列は無視
      if (label.length() == 0) {
        continue;
      }
      // 音声認識結果に追加
      if ((separator != null) && (sb.length() > 0)) {
        sb.append(separator);
      }
      sb.append(label);
    }
  }
}
```

```java
@Override
protected void onDestroy() {
  // 音声認識サービスとの接続を切断
  helper.close();
  super.onDestroy();
}
```


インスタンスの生成
STATE_INITIALIZED
初期状態
connect()
STATE_CONNECTING
サービス接続中
onServiceConnected()
close()
STATE_CONNECTED
サービス接続完了
onFinish()
startRecognition()
close()
STATE_PUSHING
音声データ送信中
stopRecognition()
onRecord()
onStopRecording()
close()
STATE_WAIT_FINISH
認識完了通知待ち
onResult()
STATE_CLOSED
認識終了
インスタンスの破棄

```
<Nbest>
  <Sentence>
    <Score>-210880640</Score>
    <Word>
      <Label>休業;キュウギョウ;キュウギョウ;名詞:動作;</Label>
      <StartPoint>0.019999999776482584</StartPoint>
      <EndPoint>0.57999999284744264</EndPoint>
      <Phoneme></Phoneme>
      <Score>-140880640</Score>
    </Word>
    <Word>
      <Label>します;シマス;シマス;動詞節;</Label>
      <StartPoint>0.57999999284744264</StartPoint>
      <EndPoint>1.2200000381469727</EndPoint>
      <Phoneme></Phoneme>
      <Score>-140380280</Score>
    </Word>
  </Sentence>
```

```
<Sentence>
  <Score>-210726512</Score>
  <Word>
    <Label>急病;キユウビョウ;キユウビョウ;名詞;</Label>
    <StartPoint>0.019999999776482584</StartPoint>
    <EndPoint>0.57999999284744264</EndPoint>
    <Phoneme></Phoneme>
    <Score>-140820199</Score>
  </Word>
  <Word>
    <Label>;、 ;、 ;pause;</Label>
    <StartPoint>0.57999999284744264</StartPoint>
    <EndPoint>0.68000001668930055</EndPoint>
    <Phoneme></Phoneme>
    <Score>-141280318</Score>
  </Word>
  <Word>
    <Label>します;シマス;シマス;動詞節;</Label>
    <StartPoint>0.68000001668930055</StartPoint>
    <EndPoint>1.2200000381469727</EndPoint>
    <Phoneme></Phoneme>
    <Score>-144429736</Score>
  </Word>
</Sentence>
```

Speech2TextとText2Speechの利用の問題

- Speech2TextとText2Speechの利用は、基本的には、一つの発話をもう一つの発話に変換する。それは個別的である
- 数多くの発話に対応しようとすると、Speech2TextとText2Speechの中間部分のプログラムに、発話に対応した数多くの処理をハードコードする必要がある。
- そうしてできたプログラムは複雑だが、用途は個別的で、他の用途に転用できるかは保証できない。
- にもかかわらず、この複雑なプログラムの入出力の部分は、共通の構造をしていて、繰り返し何度も利用されることになる。

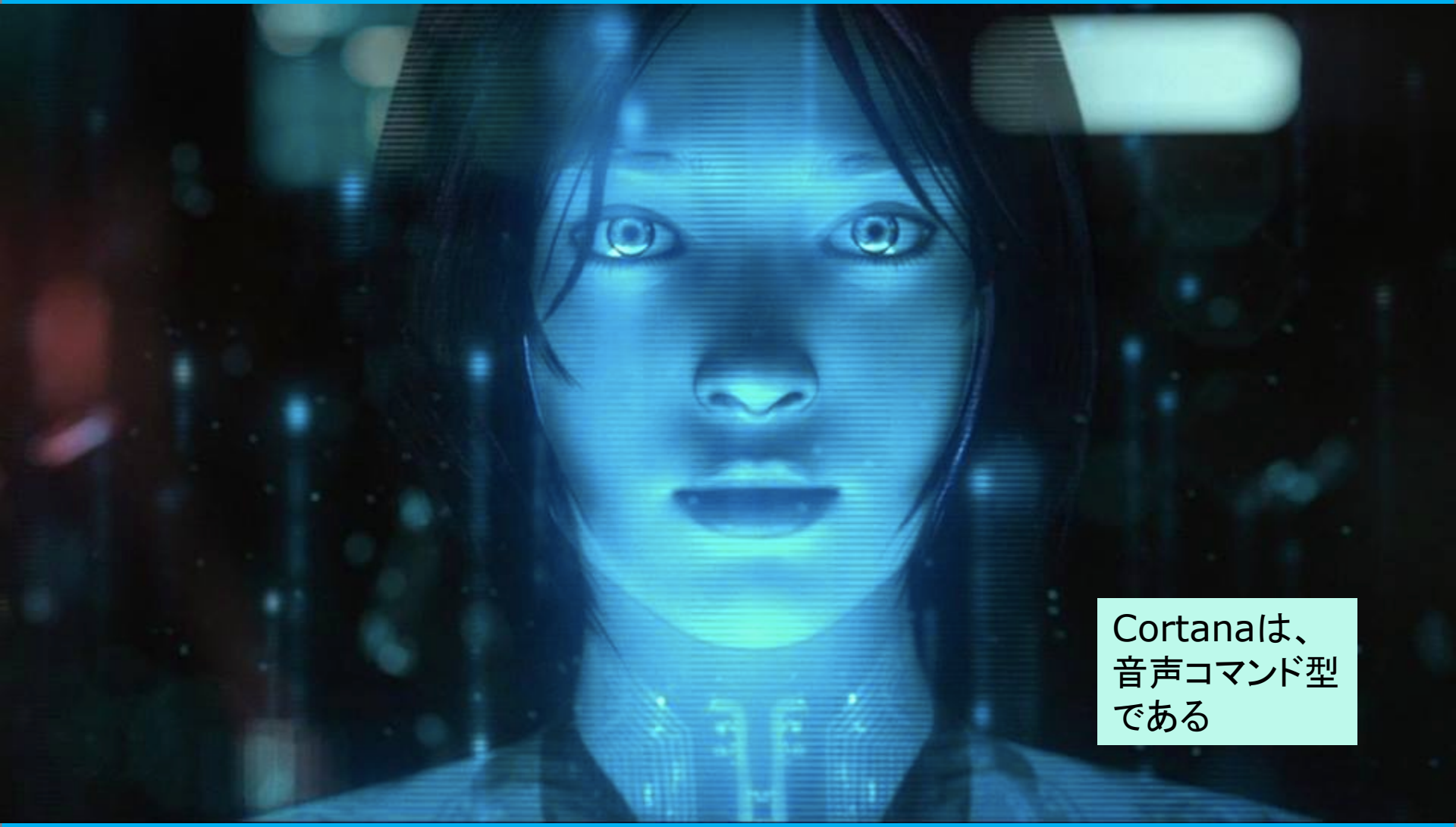
Voice Command

Cortana$_0$

Voice Command

- Voice Commandは、音声を入力とし、音声を出力として返すコマンドとしてプログラムを作成する。それは、ボイス・アシスタント・システムにとっては、最も「汎用的」なモデルである。
  ある場合には、音声出力以外の「副作用」が、コマンドの目的となる。
- Voice Commandは、音声を入力とし音声を出力とするのだが、Speech2TextとText2Speechを意識することはない。それは、自動的に呼び出される。
- Cortanaでは、プログラムあるいはボイス・コマンド定義ファイルの中で、入力（音声から変換された）と出力（音声へと変換される）は、「文字列」で表現される。これは、重要なことだ。このスタイルは、多くの開発モデルで共有されている。(Google Assistant SDKは、音声が残っている！)

Microsoft Cortana



Cortana VCD

Cortama をプログラムするには、Voice Commandをあらかじめ定義しておくのが基本。VoiceCommandDefinition (VCD)のサンプルは、以下にある。https://goo.gl/jskqr4



音声コマンドのプログラムのスタイル

```
protected override void OnActivated {
  ...
  string voiceCommandName = ... // 音声イベントから音声コマンド名を
                                //  取得するおまじない
  switch (voiceCommandName) {
      case ”helloGoodbye":
          ...
          // 遷移するページ、テキスト出力、その他の処理
          ...
          break;

       default:
           // 聞き取れなかったよ
           break;
  }
}
```



音声コマンドの定義ファイル

```
<VoiceCommands ...
  <CommandSet ...
    <Command Name="helloGoodbye">
      <ListenFor ...>
       こんにちは
      </ListenFor>
      <Feedback> さようなら </Feedback>
      ...
    </Command>
```

VCD **ListenFor** タグ

- **Command**要素の、必須の子要素（10個まで）。
  このコマンドのためにアプリが認識するフレーズを含んでいる。
  それには、**PhraseList** (あるいは **PhraseTopic**)要素のLabel属性が含まれていてもいい。例えば、
  {myList} あるいは **{myTopic}**のように。
- オプショナルな **RequireAppName** は、**AppName** 要素が、コマンド呼び出しの際に、どのように用いられるかを指定する。次のようなものがある。
  - BeforePhrase：ListenForフレーズの前にAppNameを言う
  - AfterPhrase：ListenFor + "In|On|Using|With” +AppName
  - BeforeOrAfterPhrase：どちらでもいい
  - ExplicitlySpecified：ListenForの中にAppNameを含める

VCD **ListenFor** タグ

- オプショナルな語句は、[]で囲む。話されても話されなくてもマッチには影響しない。
  <ListenFor> **[Show]** {options} **</ListenFor>**
- 次のような記述も可能である。
  <ListenFor> Find **{*}** **</ListenFor>**
  この時、Find … 以下のフレーズを含めてマッチする。この一致したフレーズは、**SpeechRecognitionResult.Text** プロパティに格納される。

VCD **Feedback** タグ

- **Command**要素の、必須の子要素。
  コマンドが認識された時に、ユーザー側に表示し、読み上げられる内容を指定する。
- **Feedback**要素は、**PhraseList** (あるいは **PhraseTopic**)要素のLabel属性を含むことができる。この時、このコマンドのListenFor要素は、同じLabelを含まなければならない。

Voice Command Definitionの形式

```
<VoiceCommands xmlns="....">
  <CommandSet xml:lang="en-us" Name="....">
    ....
    <Command Name="showTripToDestination">
      <Example> Show trip to London </Example>
      <ListenFor RequireAppName="...."> 聞き取り
        show [my] trip to {destination}
      </ListenFor>
      <ListenFor RequireAppName="...."> 聞き取り
        show [my] {builtin:AppName} trip to {destination}
      </ListenFor>
      <Feedback> 読み上げ
        Showing trip to {destination}
      </Feedback>
      <Navigate Target="foo"/>
    </Command>
```

Voice Commandの定義サンプル

```
<VoiceCommands
 xmlns="http://schemas.microsoft.com/voicecommands/1.2">
 <CommandSet xml:lang="en-us”
  Name="AdventureWorksCommandSet_en-us">
  <CommandPrefix> Adventure Works, </CommandPrefix>
  <Example> Show trip to London </Example>
  <Command Name="showTripToDestination">
   <Example> Show trip to London </Example>
   <ListenFor RequireAppName="BeforeOrAfterPhrase">
    show [my] trip to {destination}
   </ListenFor>
   <ListenFor RequireAppName="ExplicitlySpecified">
    show [my] {builtin:AppName} trip to {destination}
   </ListenFor>
   <Feedback> Showing trip to {destination} </Feedback>
   <Navigate Target="foo"/>
  </Command>
```

```
<Command Name="showTripToDestination">
  <Example>Montrer mon voyage à Londres</Example>
  <ListenFor RequireAppName="BeforeOrAfterPhrase">
   montrer [mon] voyage à {destination}
  </ListenFor>
  <ListenFor RequireAppName="ExplicitlySpecified">
   montrer [mon] {builtin:AppName} voyage à {destination}
  </ListenFor>
  <Feedback>
   Voici les détails du votre voyage à {destination}
  </Feedback>
  <Navigate Target="foo"/>
</Command>
```

フランス語

```
<Command Name="showTripToDestination">
  <Example>显示伦敦之行</Example>
  <ListenFor>显示[我]去{destination}的旅行</ListenFor>
  <ListenFor RequireAppName="ExplicitlySpecified">
   显示[我]{builtin:AppName}去{destination}的旅行
  </ListenFor>
  <Feedback>显示{destination}之行</Feedback>
  <Navigate Target="foo"/>
</Command>
```

中国語

VCD **Navigate**タグ

- **Command**要素の、必須な子要素。
- **Target**属性は、オプショナルで、アプリが起動した時、遷移するページを指定する。
- **Target**属性の値は、**SpeechRecognitionSemanticInterpretation.Properties** から、"NavigationTarget"キーを使って取得できる。

VCD **PhraseList** タグ

- **CommandSet** 要素のオプショナルな子要素。
  一つのCommandSetは、2,000個までの**Item**要素を持つことができる。

```
<CommandSet ...>
 <Command Name="cancelTripToDestination"> ...
   <ListenFor RequireAppName="BeforeOrAfterPhrase">
     cancel [my] trip to {destination}
   </ListenFor> ...
   <Feedback> Cancelling trip to {destination}</Feedback>
 </Command>
 <PhraseList Label="destination">
   <Item>London</Item>
   <Item>Las Vegas</Item>
   <Item>Melbourne</Item>
   <Item>Yosemite National Park</Item>
 </PhraseList>
</CommandSet>
```

Cortana と Alexaの比較

- Cortanaの「コマンド名」は、AlexaのIntentに対応する。
- 定義ファイルに記述された、10個までの入力の候補（ListenForで定義される）に対して、一つの出力(FeedBack)を返す。AlexaのSampleUtterancetxtには、もっと多数のマッピングが可能である。
- ListenFor内に記述される {PhraseList_Label}は、後のAlexaで採用されるSlotと同じものである。PhraseListは、Slotの型の定義を与えている。

- このCortanaのプログラミング・スタイルは、古いものだ。現在では、LUIS + Bot Framework + Connectorのスタイルに移行している。

Intent Model

Alexa$_0$

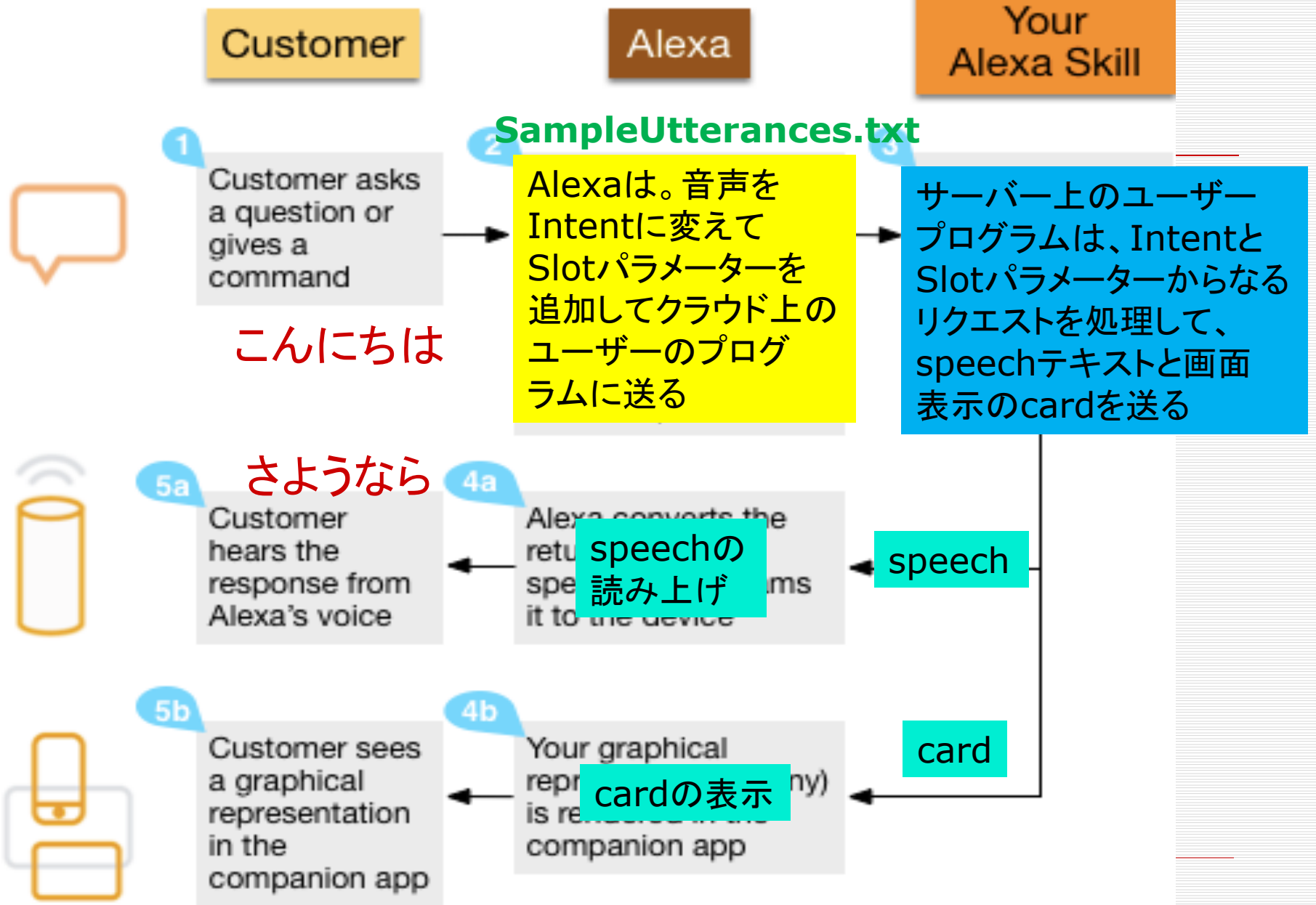
Amazon Alexa



AlexaのIntent Model

- Alexa は、同じ「意図」を持つ多数の文を、一つの**Intent**に、結びつける。それは、Cortanaで、一つのコマンド名に、ListenForで記述された複数の入力候補が結びつけられるのと同じである。
- ただ、Alexaでは、Cortanaとはことなり、そのマッピングはプログラム定義とは別のファイル **SampleUtterance.txt**で行われる。
- このUtteranceデータを通じて、Alexaは。音声をIntentに変えて、ネットをまたいで、デフォールトではクラウド上のユーザーのプログラムに送る。
- ユーザーのプログラムは、このリクエストを受け取り、必要な処理をして、読み上げられる文字列 speech と画面表示用のcardの二つを、レスポンドとして返す。

ユーザーのプログラム


Customer
Alexa
Your Alexa Skill
SampleUtterances.txt
1
Customer asks a question or gives a command
こんにちは
2
Alexaは。音声をIntentに変えてSlotパラメーターを追加してクラウド上のユーザーのプログラムに送る
3
サーバー上のユーザープログラムは、IntentとSlotパラメーターからなるリクエストを処理して、speechテキストと画面表示のcardを送る
speech
4a
speechの読み上げ
5a
さようなら
Customer hears the response from Alexa's voice
card
4b
cardの表示
5b
Customer sees a graphical representation in the companion app

AlexaのIntent Model

- ユーザー・プログラムは、AWSのLAMBDAに登録することができる。
- プログラムは、イベント・ドリブンのスタイルで書かれる。Android のIntentプログラムの、ネットワーク版だと思っていい。
- プログラムのスタイルは、基本的にVoice Commandと同じ（Speechletと呼ばれた）だが、次第に、ユーザーの質問に対して答えを提供するボイス・サービスとして受け止められるようになる。こうしたサービスをSkillと呼ぶのが、定着し始める。



SampleUtterances.txtの例

```
HelloIntent こんにちは
HelloIntent はい。こんにちは
HelloIntent おい
HelloIntent こら
HelloIntent なんかいって
HelloIntent お元気ですか
HelloIntent お元気

HelpIntent ヘルプ
HelpIntent 助けて
HelpIntent どうすればいい
HelpIntent お願い
```

入力の発話を待ち受けて、マッチすれば、定義されたIntent Name に対応づける。Intentは、Requestの一部として、Speechletに送られる。

出力用のデータは、プログラム中にハードコードされる。




```
public SpeechletResponse onIntent (...) ... {
    ...
    if ("HelloIntent".equals(intentName)) {
        return getHelloResponse();
    } else if ("HelpIntent".equals(intentName)) {
        return getHelpResponse();
    } else {
        throw new SpeechletException("Invalid Intent");
    }
}
```

```
private SpeechletResponse getHelloResponse() {
    String speechText = “さようなら”;
    SimpleCard card = new SimpleCard();
    ...
    return SpeechletResponse.newTellResponse(speech, card);
}
```

音声とカードが返る

Alexa HelloWorld サンプル

User: “Alexa, say hello”
Alexa: “Hello World!”

User: “Alexa, what can I do?”
Alexa: “You can say hello to me!”

HelloWorldSpeechlet

```
public class HelloWorldSpeechlet implements Speechlet {

  public void onSessionStarted ( ... )  ....
  public SpeechletResponse onLaunch ( ... )  ....
  public SpeechletResponse onIntent ( ... )  ....
  public void onSessionEnded ( ... )  ....

  private SpeechletResponse getWelcomeResponse ()  ....
  private SpeechletResponse getHelloResponse()  ....
  private SpeechletResponse getHelpResponse()  ....

}
```

もちろん、イベント・ドリブンの
プログラミング・スタイルである。

onIntent

Androidのプログラミング・スタイルを、Amazon LAMDA上で使っている！

```
@Override
    public SpeechletResponse onIntent (final IntentRequest request,
            final Session session)  throws SpeechletException {
        log.info("onIntent requestId={}, sessionId={}",
                request.getRequestId(),
                session.getSessionId());

        Intent intent = request.getIntent();
        String intentName = (intent != null) ? intent.getName() : null;

        if ("HelloWorldIntent".equals(intentName)) {
            return getHelloResponse();
        } else if ("HelpIntent".equals(intentName)) {
            return getHelpResponse();
        } else {
            throw new SpeechletException("Invalid Intent");
        }
    }
```

helloworld.speechAsset.
SampleUtterances.txt

intentName　　発話

HelloWorldIntent say hello
HelloWorldIntent say hello world
HelloWorldIntent hello
HelloWorldIntent say hi
HelloWorldIntent say hi world
HelloWorldIntent hi
HelloWorldIntent how are you

HelpIntent help
HelpIntent help me
HelpIntent what can I ask you
HelpIntent get help
HelpIntent to help
HelpIntent to help me

cortanaの**ListenFor**と同じ働きをする。それぞれの発話に、intent Nameが付けられている。

“say hello”と“how are you”は、同じintent Nameが割り当てられている。

helloworld.speechAsset. **SampleUtterances.txt**

```
HelpIntent what commands can I ask
HelpIntent what commands can I say
HelpIntent what can I do
HelpIntent what can I use this for
HelpIntent what questions can I ask
HelpIntent what can you do
HelpIntent what do you do
HelpIntent how do I use you
HelpIntent how can I use you
HelpIntent what can you tell me
```

```
if ("HelloWorldIntent".equals(intentName)) {
        return getHelloResponse();
```

getHelloResponse

```
private SpeechletResponse getHelloResponse() {
        String speechText = "Hello world";

        // Create the Simple card content.
        SimpleCard card = new SimpleCard();
        card.setTitle("HelloWorld");
        card.setContent(speechText);

        // Create the plain text output.
        PlainTextOutputSpeech speech = new PlainTextOutputSpeech();
        speech.setText(speechText);

        return SpeechletResponse.newTellResponse(speech, card);
    }
```

getHelpResponse

```
private SpeechletResponse getHelpResponse() {
        String speechText = "You can say hello to me!";

        // Create the Simple card content.
        SimpleCard card = new SimpleCard();
        card.setTitle("HelloWorld");
        card.setContent(speechText);

        // Create the plain text output.
        PlainTextOutputSpeech speech = new PlainTextOutputSpeech();
        speech.setText(speechText);

        // Create reprompt
        Reprompt reprompt = new Reprompt();
        reprompt.setOutputSpeech(speech);

        return SpeechletResponse.newAskResponse(speech, reprompt, card);
    }
```

Intent + Slot Model

Alexa

Alexa History Buff サンプル

User: “Alexa, ask History Buff what happened on August 20th”
Alexa: *(Reads back three events, in reverse chronological order)* “Want to go deeper in history?”
User: “Yes”
Alexa: *(Reads back next set of three events)* “Want to go deeper in history?”
User: “No”

HistoryBuffSpeechlet

```
public class HistoryBuffSpeechlet implements Speechlet {
  ...
  public void onSessionStarted ( ... )  ....
  public SpeechletResponse onLaunch ( ... )  ....
  public SpeechletResponse onIntent ( ... )  ....
  public void onSessionEnded ( ... )  ....

  private SpeechletResponse getWelcomeResponse()  ...
  private Calendar getCalendar( ... ) ...
  private SpeechletResponse handleFirstEventRequest( ... )  ...
  private SpeechletResponse handleNextEventRequest( ... )  ...
  private ArrayList<String> getJsonEventsFromWikipedia( ... ) ...
  private ArrayList<String> parseJson( ... )  ...
  private SpeechletResponse newAskResponse( ... ) ...

}
```

public SpeechletResponse **onIntent**

```
public SpeechletResponse onIntent(final IntentRequest request,
   final Session session) throws SpeechletException {
       log.info("onIntent requestId={}, sessionId={}",
               request.getRequestId(), session.getSessionId());

       Intent intent = request.getIntent();
       String intentName = intent.getName();

       if ("GetFirstEventIntent".equals(intentName)) {
           return handleFirstEventRequest (intent, session);
       } else if ("GetNextEventIntent".equals(intentName)) {
           return handleNextEventRequest (session);
       } else if ("HelpIntent".equals(intentName)) {
          // Create the plain text output.
```

public SpeechletResponse **onIntent**

```
        String speechOutput =
                "With History Buff, you can get"
                        + " historical events for any day of the year."
                        + " For example, you could say today,"
                        + " or August thirtieth, or you can say exit. Now, “
                        + “which day do you want?”;
        String repromptText = "Which day do you want?”;
        return newAskResponse(speechOutput, repromptText);

    } else if ("FinishIntent".equals(intentName)) {
        PlainTextOutputSpeech outputSpeech =
            new PlainTextOutputSpeech();
        outputSpeech.setText("Goodbye");
        return SpeechletResponse.newTellResponse(outputSpeech);
    } else {
        throw new SpeechletException("Invalid Intent");
    }
}
```

historybuff.speechAsset.
IntentSchema.jason

```
{
  "intents": [
    {
      "intent": "GetFirstEventIntent",
      "slots": [
        {
          "name": "day",
          "type": "DATE"
        }
      ]
    },
    {
      "intent": "GetNextEventIntent",
      "slots": []
    },
```

historybuff.speechAsset.
IntentSchema.jason

```
    {
      "intent": "HelpIntent",
      "slots": []
    },
    {
      "intent": "FinishIntent",
      "slots": []
    }
  ]
}
```

historyBuff.speechAsset.
SampleUtterances.txt

GetFirstEventIntent get events for {august seventeenth|day}
GetFirstEventIntent get events for {december thirty first|day}
GetFirstEventIntent get events for {next monday|day}
GetFirstEventIntent get events for {today|day}
GetFirstEventIntent give me events for {august seventeenth|day}
GetFirstEventIntent give me events for {next monday|day}
GetFirstEventIntent give me events for {today|day}
GetFirstEventIntent what happened on {august seventeenth|day}
GetFirstEventIntent what happened on {next monday|day}
GetFirstEventIntent what happened on {today|day}
GetFirstEventIntent what happened
GetFirstEventIntent {january first|day}
GetFirstEventIntent {january second|day}
GetFirstEventIntent {january third|day}
GetFirstEventIntent {january fourth|day}
GetFirstEventIntent {january fifth|day}
GetFirstEventIntent {january sixth|day}
GetFirstEventIntent {january seventh|day}

historyBuff.speechAsset.
SampleUtterances.txt

GetFirstEventIntent {january eighth|day}
GetFirstEventIntent {january ninth|day}
GetFirstEventIntent {january tenth|day}
GetFirstEventIntent {january eleventh|day}
GetFirstEventIntent {january twelfth|day}
GetFirstEventIntent {january thirteenth|day}
GetFirstEventIntent {january fourteenth|day}
GetFirstEventIntent {january fifteenth|day}
GetFirstEventIntent {january sixteenth|day}
GetFirstEventIntent {january seventeenth|day}
GetFirstEventIntent {january eighteenth|day}
GetFirstEventIntent {january nineteenth|day}
GetFirstEventIntent {january twentieth|day}
GetFirstEventIntent {january twenty first|day}
GetFirstEventIntent {january twenty second|day}
GetFirstEventIntent {january twenty third|day}
GetFirstEventIntent {january twenty fourth|day}
GetFirstEventIntent {january twenty fifth|day}

historyBuff.speechAsset.
SampleUtterances.txt

```
GetFirstEventIntent {january twenty sixth|day}
GetFirstEventIntent {january twenty seventh|day}
GetFirstEventIntent {january twenty eighth|day}
GetFirstEventIntent {january twenty ninth|day}
GetFirstEventIntent {january thirtieth|day}
GetFirstEventIntent {january thirty first|day}
GetFirstEventIntent {february first|day}
GetFirstEventIntent {february second|day}
GetFirstEventIntent {february third|day}
GetFirstEventIntent {february fourth|day}
GetFirstEventIntent {february fifth|day}
GetFirstEventIntent {february sixth|day}
GetFirstEventIntent {february seventh|day}
GetFirstEventIntent {february eighth|day}
GetFirstEventIntent {february ninth|day}
GetFirstEventIntent {february tenth|day}
GetFirstEventIntent {february eleventh|day}
GetFirstEventIntent {february twelfth|day}
```

historyBuff.speechAsset.
SampleUtterances.txt

```
GetFirstEventIntent {february thirteenth|day}
GetFirstEventIntent {february fourteenth|day}
GetFirstEventIntent {february fifteenth|day}
GetFirstEventIntent {february sixteenth|day}
GetFirstEventIntent {february seventeenth|day}
 ....
 ....
GetFirstEventIntent {december twenty third|day}
GetFirstEventIntent {december twenty fourth|day}
GetFirstEventIntent {december twenty fifth|day}
GetFirstEventIntent {december twenty sixth|day}
GetFirstEventIntent {december twenty seventh|day}
GetFirstEventIntent {december twenty eighth|day}
GetFirstEventIntent {december twenty ninth|day}
GetFirstEventIntent {december thirtieth|day}
GetFirstEventIntent {december thirty first|day}
```

historyBuff.speechAsset.
SampleUtterances.txt

```
GetFirstEventIntent {next Monday|day}
GetFirstEventIntent {next Tuesday|day}
GetFirstEventIntent {next Wednesday|day}
GetFirstEventIntent {next Thursday|day}
GetFirstEventIntent {next Friday|day}
GetFirstEventIntent {next Saturday|day}
GetFirstEventIntent {next Sunday|day}
GetFirstEventIntent {today|day}
GetFirstEventIntent {tomorrow|day}
GetFirstEventIntent {yesterday|day}

FinishIntent no
FinishIntent nope
FinishIntent no thanks
FinishIntent no thank you
```

historyBuff.speechAsset.
SampleUtterances.txt

```
HelpIntent helpHelpIntent help me
HelpIntent what can I ask you
HelpIntent get helpHelpIntent to help
HelpIntent to help me
HelpIntent what commands can I ask
HelpIntent what commands can I say
HelpIntent what can I do
HelpIntent what can I use this for
HelpIntent what questions can I ask
HelpIntent what can you do
HelpIntent what do you do
HelpIntent how do I use you
HelpIntent how can I use you
HelpIntent what can you tell me
GetNextEventIntent yes
GetNextEventIntent yup
GetNextEventIntent sure
GetNextEventIntent yes please
```

Intent と Slot

Slot Typeの導入

Intent と Slot

- Intentは、ユーザによって話されたリクエストを実現するActionを表す。
- Intentは、オプションで引数を持つことができる。その引数をSlotという。SampleUtteranceに、Slotの記述を許す。

- **IntentName** this is a sample utterance with no slots
- **IntentName** this is a sample utterance containing a **{SlotName}**
- **IntentName** this is a sample utterance containing a **{SlotName}** and **{AnotherSlotName}**

- 旅行を計画するスキルのIntentの名前を **PlanMyTrip** とすれば、PlanMyTripは、**toCity, fromCity, travelDate** といったSlotを持つだろう。

Intent PlanMyTripのスキーマ定義

```
{
  “intents” : [
     {
        “intent” : “PlanMyTrip” ,
        “slots”   : [
           {
              “name”: “fromCity” ,
              ”type”  : “AMAZON.US_CITY”
           } ,
           {
              “name”: “toCity” ,
              ”type”  : “AMAZON.US_CITY”
           } ,
           {
              “name”: “travelDate” ,
              ”type”  : “AMAZON.DATE”
           }
        ]
     }
}
```

Intent GetHoroscopeのスキーマ定義の例

```
{
  "intents": [
    {
      "intent": "GetHoroscope",
      "slots": [
        {
          "name": "Sign",
          "type": "LIST_OF_SIGNS"
        },
        {
          "name": "Date",
          "type": "AMAZON.DATE"
        }
      ]
    }
  ]
}
```

Custom slot type
LIST_OF_SIGNS の定義例

```
Aries
Taurus
Gemini
Cancer
Leo
Pisces
Virgo
Libra
Scorpio
Sagittarius
Capricorn
Aquarius
```

対応する sample utterances:

```
GetHoroscope what is the horoscope for {Sign}
GetHoroscope what will the horoscope for {Sign} be {Date}
GetHoroscope what will the horoscope for {Sign} be on {Date}
GetHoroscope get me my horoscope
GetHoroscope {Sign}
...
(many more sample utterances)
```

User: Alexa, ask *Plan My Trip* to plan a trip from Seattle to Portland on Friday.

上のような発話がなされたとする。
Utteranceに次のような記述があったとしよう。

Plan a trip from {fromCity} to {toCity} on {travalDate}

Alexa は、fromCity スロットに ”Seatle“、とCity スロットに ”Portland“、travelDate スロットに”Friday“ という値を、PlayMyTrip Intentの引数として渡して、”Plan My Trip”サービスを呼び出す。

発話



Skillの呼び出し



SlotとEntity Resolution

人は、よく、同じ物事を違う言葉で話すことがある。
MEDIA_TYPEというslotは、mediaの特定の型、例えば、”song”, “track”, “single” と言ったものを集めるようとする。
たいていの場合、コード的には、この三つの表現を同じように扱うことができる。

MEDIA_TYPEを、一つのEntity として、その取りうる値を、もっと正確に定義することができる。それを、Entity Resolutionという。

Entity Resolution では、Slot Typeの値には、IDがつけられ、かつ、その同義語（synonym）のリストが追加される。話者が話した発話自体は、”name” の ”value” とされる。

Slotの型のEntity Resolution

```
{
  "types": [
    {
      "name": "MEDIA_TYPE",
      "values": [
        {
          "id": "ALBUM",
          "name": {
            "value": "album",
            "synonyms": ["cd","lp","record"]
          }
        },
        {
          "id": "SONG",
          "name": {
            "value": "song",
            "synonyms": ["track","single","tune"]
          }
        }
      ]
    }
  ]
}
```

Slot Type	Short Description	Supported Languages
AMAZON.DATE	Converts words that indicate *dates* ("today", "tomorrow", or "july") into a date format (such as "2015-07-00T9").	English (UK) English (US) German
AMAZON.DURATION	Converts words that indicate *durations* ("five minutes") into a numeric duration ("PT5M").	English (UK) English (US) German
AMAZON.FOUR_DIGIT_NUMBER	Converts words that represent *four-digit numbers* ("six oh four five" into digits ("6045").	English (UK) English (US) German
AMAZON.NUMBER	Converts numeric words ("five") into digits (such as "5").	English (UK) English (US) German
AMAZON.TIME	Converts words that indicate time ("four in the morning", "two p m") into a time value ("04:00", "14:00").	English (UK) English (US) German

Slot Type

Numbers, Dates, and Times

AMAZON.Actor
AMAZON.AdministrativeArea
AMAZON.AggregateRating
AMAZON.Airline
AMAZON.Airport
AMAZON.Animal
AMAZON.Artist
AMAZON.AT_CITY
AMAZON.AT_REGION
AMAZON.Athlete
AMAZON.Author
AMAZON.Book
AMAZON.BookSeries
AMAZON.BroadcastChannel
AMAZON.CivicStructure
AMAZON.Color
AMAZON.Comic
AMAZON.Corporation
AMAZON.Country
AMAZON.CreativeWorkType
AMAZON.DayOfWeek
AMAZON.DE_CITY
AMAZON.DE_FIRST_NAME
AMAZON.DE_REGION
AMAZON.Dessert
AMAZON.DeviceType
AMAZON.Director
AMAZON.Drink
AMAZON.EducationalOrganization
AMAZON.EUROPE_CITY
AMAZON.EventType
AMAZON.Festival
AMAZON.FictionalCharacter
AMAZON.FinancialService
AMAZON.Food
AMAZON.FoodEstablishment
AMAZON.Game
AMAZON.GB_CITY
AMAZON.GB_FIRST_NAME
AMAZON.GB_REGION
AMAZON.Genre
.....
.....
AMAZON.WrittenCreativeWorkType

Slot Type

List Type

WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: person Search WordNet

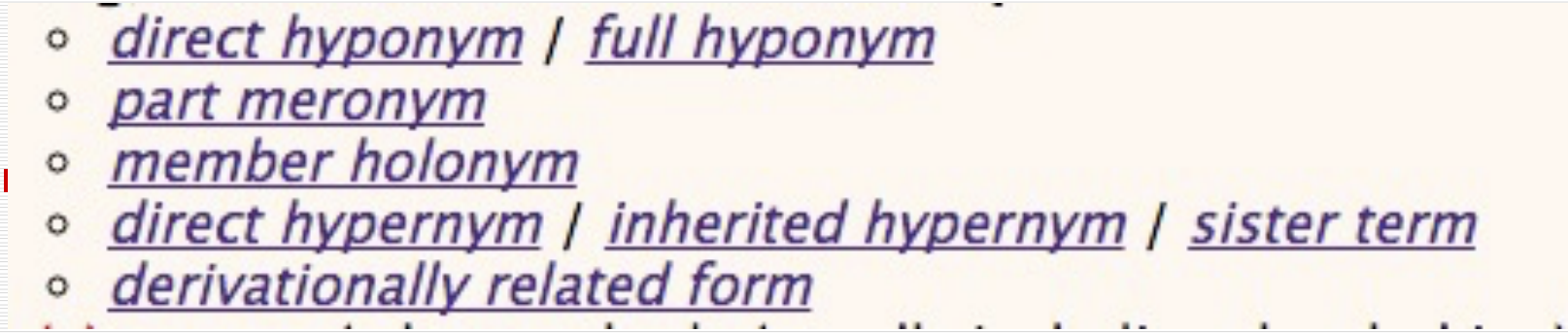
Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **person**, individual, someone, somebody, mortal, soul **(a human being)** ***"there was too much for one person to do"***
  - *direct hyponym* / *full hyponym*
  - *part meronym*
  - *member holonym*
  - *direct hypernym* / *inherited hypernym* / *sister term*
  - *derivationally related form*
- S: (n) **person** (a human body (usually including the clothing)) *"a weapon was hidden on his person"*
- S: (n) **person** (a grammatical category used in the classification of pronouns, possessive determiners, and verb forms according to whether they indicate the speaker, the addressee, or a third party) *"stop talking about yourself in the third person"*



hyponym ヒポニム　ある単語よりもより特殊な単語
meronym ある語の部分を示す単語
holonym 特定の単語が全体の一部分となっている名前の単語
hypernym 上位概念語

Noun

Direct hyponym

- S: (n) **person**, individual, someone, somebody, mortal, soul (a human being) *"there was too much for one person to do"*
  - ***direct hyponym*** / *full hyponym*
    - S: (n) self (a person considered as a unique individual) *"one's own self"*
    - S: (n) adult, grownup (a fully developed person from maturity onward)
    - S: (n) adventurer, venturer (a person who enjoys taking risks)
    - S: (n) anomaly, unusual person (a person who is unusual)
    - S: (n) anachronism (a person who seems to be displaced in time; who belongs to another age)
    - S: (n) applicant, applier (a person who requests or seeks something such as assistance or employment or admission)
    - S: (n) appointee, appointment (a person who is appointed to a job or position)
    - S: (n) capitalist (a person who invests capital in a business (especially a large business))
    - S: (n) captor, capturer (a person who captures and holds people or animals)
    - S: (n) changer, modifier (a person who changes something) *"an inveterate changer of the menu"*
    - S: (n) color-blind person (a person unable to distinguish differences in hue)

hyponym ヒポニム　ある単語よりもより特殊な単語

WARNING: The search exceeded the result limit, so the following list is valid but incomplete. Only the top levels of the list are displayed.

Full hyponym

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **person**, individual, someone, somebody, mortal, soul (a human being) *"there was too much for one person to do"*
  - *direct hyponym* / ***full hyponym***
    - S: (n) self (a person considered as a unique individual) *"one's own self"*
    - S: (n) adult, grownup (a fully developed person from maturity onward)
    - S: (n) adventurer, venturer (a person who enjoys taking risks)
    - S: (n) anomaly, unusual person (a person who is unusual)
    - S: (n) anachronism (a person who seems to be displaced in time; who belongs to another age)
    - S: (n) applicant, applier (a person who requests or seeks something such as assistance or employment or admission)
    - S: (n) appointee, appointment (a person who is appointed to a job or position)
    - S: (n) capitalist (a person who invests capital in a business (especially a large business))

コード・サンプル

Node.js 版

https://github.com/alexa/
skill-sample-nodejs-hello-world/blob/master/src/index.js

Hello World サンプル
Event Handlerの登録

```
'use strict';
var Alexa = require("alexa-sdk");
exports.handler = function(event, context, callback) {
    var alexa = Alexa.handler(event, context);
    alexa.registerHandlers(handlers);
    alexa.execute();
};
```

Hello World サンプル
Event Handler

```
var handlers = {
    'LaunchRequest': function () {
        this.emit('SayHello');
    },
    'HelloWorldIntent': function () {
        this.emit('SayHello')
    },
    'SayHello': function () {
        this.emit(':tell', 'Hello World!');
    }
};
```

speechAssets/intentSchema.json

```
{
  "intents": [
      {
          "intent": "HelloWorldIntent”
      }
  ]
}
```

speechAssets/Utterance.txt

```
HelloWorldIntent hello
HelloWorldIntent say hello
HelloWorldIntent say hello world
```

Build An Alexa Fact Skill

https://github.com/alexa/skill-sample-nodejs-fact

speechAssets/intentSchema.json

```
{ "intents": [
    { "intent": "GetNewFactIntent" },
    { "intent": "AMAZON.HelpIntent" },
    { "intent": "AMAZON.StopIntent" },
    { "intent": "AMAZON.CancelIntent" }
]}
```

speechAssets/Utterance.txt

```
GetNewFactIntent a fact
GetNewFactIntent a space fact
GetNewFactIntent tell me a fact
GetNewFactIntent tell me a space fact
GetNewFactIntent give me a fact
GetNewFactIntent give me a space fact
GetNewFactIntent tell me trivia
GetNewFactIntent tell me a space trivia
GetNewFactIntent give me trivia
GetNewFactIntent give me a space trivia
GetNewFactIntent give me some information
GetNewFactIntent give me some space information
GetNewFactIntent tell me something
GetNewFactIntent give me something
```

```
'use strict';
const Alexa = require('alexa-sdk');

//=========================================================================================================================================
//TODO: The items below this comment need your attention.
//=========================================================================================================================================

//Replace with your app ID (OPTIONAL).  You can find this value at the top of your skill's page on http://developer.amazon.com.
//Make sure to enclose your value in quotes, like this: const APP_ID = 'amzn1.ask.skill.bb4045e6-b3e8-4133-b650-72923c5980f1';
const APP_ID = undefined;

const SKILL_NAME = 'Space Facts';
const GET_FACT_MESSAGE = "Here's your fact: ";
const HELP_MESSAGE = 'You can say tell me a space fact, or, you can say exit... What can I help you with?';
const HELP_REPROMPT = 'What can I help you with?';
const STOP_MESSAGE = 'Goodbye!';
//=========================================================================================================================================
//TODO: Replace this data with your own.  You can find translations of this data at http://github.com/alexa/skill-sample-node-js-fact/data
//=========================================================================================================================================
const data = [
    'A year on Mercury is just 88 days long.',
    'Despite being farther from the Sun, Venus experiences higher temperatures than Mercury.',
    'Venus rotates counter-clockwise, possibly because of a collision in the past with an asteroid.',
    'On Mars, the Sun appears about half the size as it does on Earth.',
    'Earth is the only planet not named after a god.',
    'Jupiter has the shortest day of all the planets.',
    'The Milky Way galaxy will collide with the Andromeda Galaxy in about 5 billion years.',
    'The Sun contains 99.86% of the mass in the Solar System.',
    'The Sun is an almost perfect sphere.',
    'A total solar eclipse can happen once every 1 to 2 years. This makes them a rare event.',
    'Saturn radiates two and a half times more energy into space than it receives from the sun.',
    'The temperature inside the Sun can reach 15 million degrees Celsius.',
    'The Moon is moving approximately 3.8 cm away from our planet every year.',
];
```

```
//=========================================================================================
//Editing anything below this line might break your skill.
//=========================================================================================

const handlers = {
    'LaunchRequest': function () {
        this.emit('GetNewFactIntent');
    },
    'GetNewFactIntent': function () {
        const factArr = data;
        const factIndex = Math.floor(Math.random() * factArr.length);
        const randomFact = factArr[factIndex];
        const speechOutput = GET_FACT_MESSAGE + randomFact;

        this.response.cardRenderer(SKILL_NAME, randomFact);
        this.response.speak(speechOutput);
        this.emit(':responseReady');
    },
    'AMAZON.HelpIntent': function () {
        const speechOutput = HELP_MESSAGE;
        const reprompt = HELP_REPROMPT;
```

```
        this.response.speak(speechOutput).listen(reprompt);
        this.emit(':responseReady');
    },
    'AMAZON.CancelIntent': function () {
        this.response.speak(STOP_MESSAGE);
        this.emit(':responseReady');
    },
    'AMAZON.StopIntent': function () {
        this.response.speak(STOP_MESSAGE);
        this.emit(':responseReady');
    },
};

exports.handler = function (event, context, callback) {
    const alexa = Alexa.handler(event, context, callback);
    alexa.APP_ID = APP_ID;
    alexa.registerHandlers(handlers);
    alexa.execute();
};
```

:tell と :ask Action

https://github.com/alexa/alexa-skills-kit-sdk-for-nodejs

```
var handlers = {

    'HelloWorldIntent': function () {
        this.emit(':tell', 'Hello World!');
    }

};
```

- Notice the new syntax above for ':tell'? Alexa-sdk follows a tell/ask response methodology for generating your outputSpeech response objects. To ask the user for information we would instead use an :ask action. The difference between :ask and :tell is that after a :tell action, the session is ended without waiting for the user to provide more input.

```
this.emit(':ask', 'What would you like to do?', 'Please say that again?');
```

```
var speechOutput = 'Hello world!';
var repromptSpeech = 'Hello again!';

this.emit(':tell', speechOutput);

this.emit(':ask', speechOutput, repromptSpeech);

var cardTitle = 'Hello World Card';
var cardContent = 'This text will be displayed in the companion app card.';

var imageObj = {
    smallImageUrl: 'https://imgs.xkcd.com/comics/standards.png',
    largeImageUrl: 'https://imgs.xkcd.com/comics/standards.png'
};

var permissionArray = ['read::alexa:device:all:address'];

var updatedIntent = this.event.request.intent;

var slotToElicit = "Slot to elicit";

var slotToConfirm = "Slot to confirm";
```

```
this.emit(':askWithCard', speechOutput, repromptSpeech, cardTitle, cardContent, imageObj);

this.emit(':tellWithCard', speechOutput, cardTitle, cardContent, imageObj);

this.emit(':tellWithLinkAccountCard', speechOutput);

this.emit(':askWithLinkAccountCard', speechOutput);

this.emit(':tellWithPermissionCard', speechOutput, permissionArray);

this.emit(':delegate', updatedIntent);

this.emit(':elicitSlot', slotToElicit, speechOutput, repromptSpeech, updatedIntent);

this.emit(':elicitSlotWithCard', slotToElicit, speechOutput, repromptSpeech, cardTitle, cardContent, updatedIntent, i

this.emit(':confirmSlot', slotToConfirm, speechOutput, repromptSpeech, updatedIntent);

this.emit(':confirmSlotWithCard', slotToConfirm, speechOutput, repromptSpeech, cardTitle, cardContent, updatedIntent,

this.emit(':confirmSlotWithCard', slotToConfirm, speechOutput, repromptSpeech, cardTitle, cardContent, updatedIntent,

this.emit(':confirmIntent', speechOutput, repromptSpeech, updatedIntent);

this.emit(':confirmIntentWithCard', speechOutput, repromptSpeech, cardTitle, cardContent, updatedIntent, imageObj);

this.emit(':responseReady'); // Called after the response is built but before it is returned to the Alexa service. Ca

this.emit(':saveState', false); // Handles saving the contents of this.attributes and the current handler state to Dy

this.emit(':saveStateError'); // Called if there is an error while saving state. Override to handle any errors yourse
```

Skill開発ツール
Skill Builder (Beta)を使う

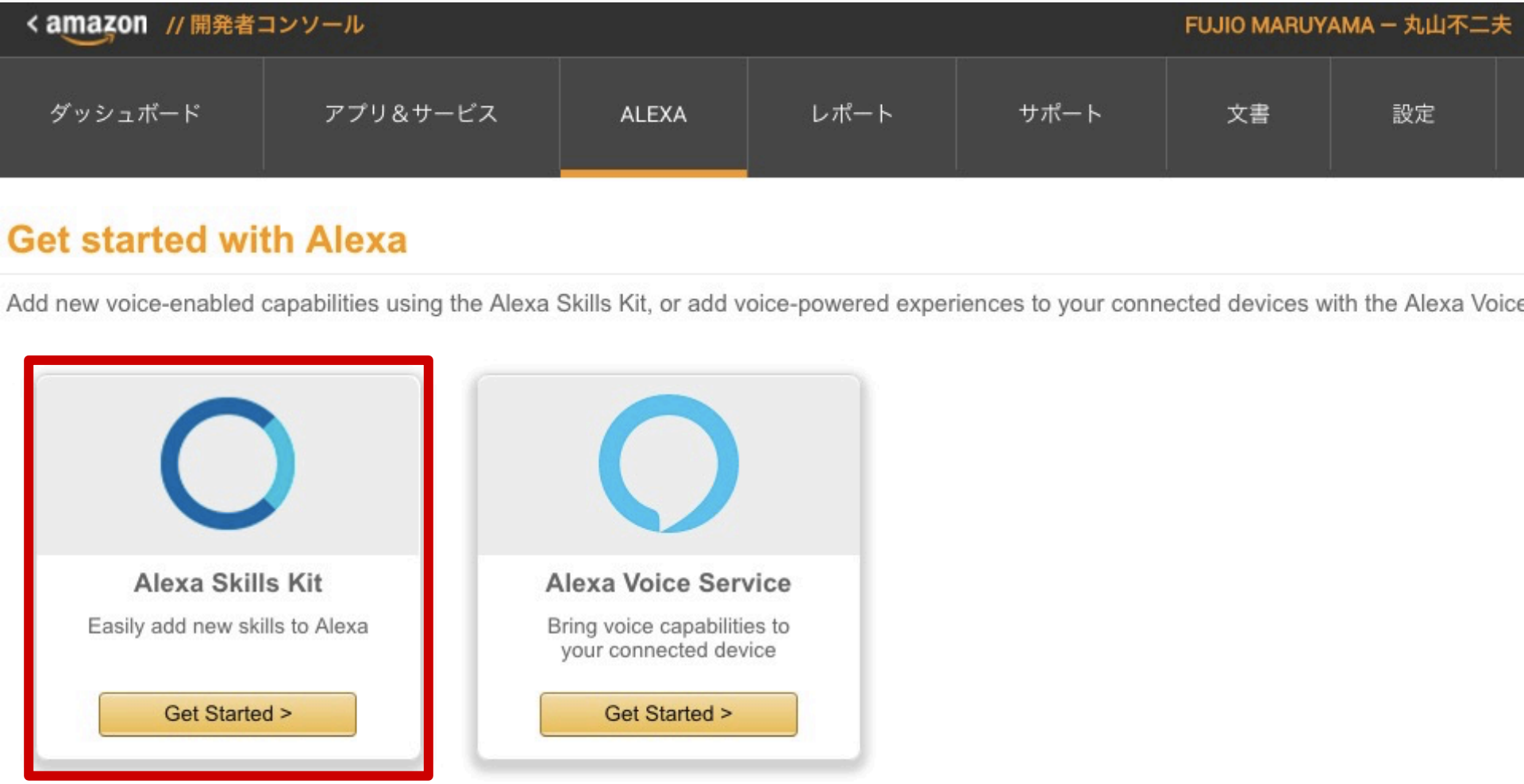
Alexa Skill Kit

https://developer.amazon.com/edw/home.html#/

Skill Builder (Beta)

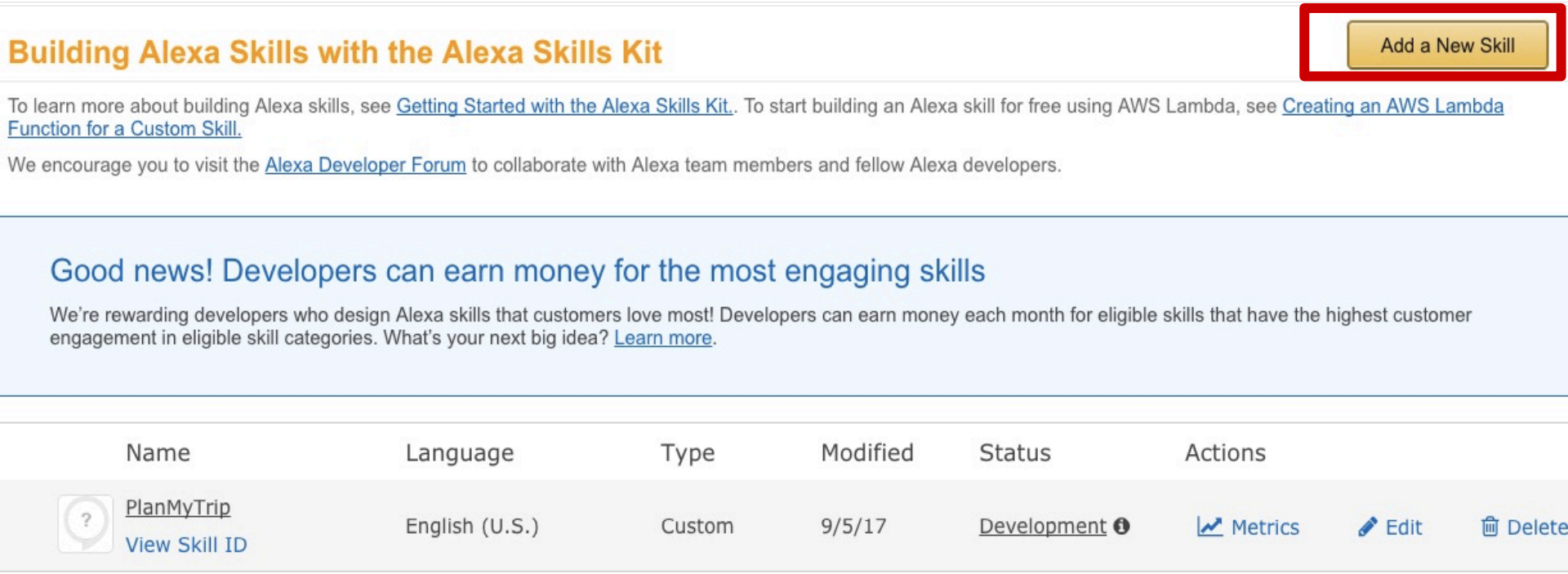
https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/docs/ask-define-the-vui-with-gui

開発者コンソールでAlexa を開く

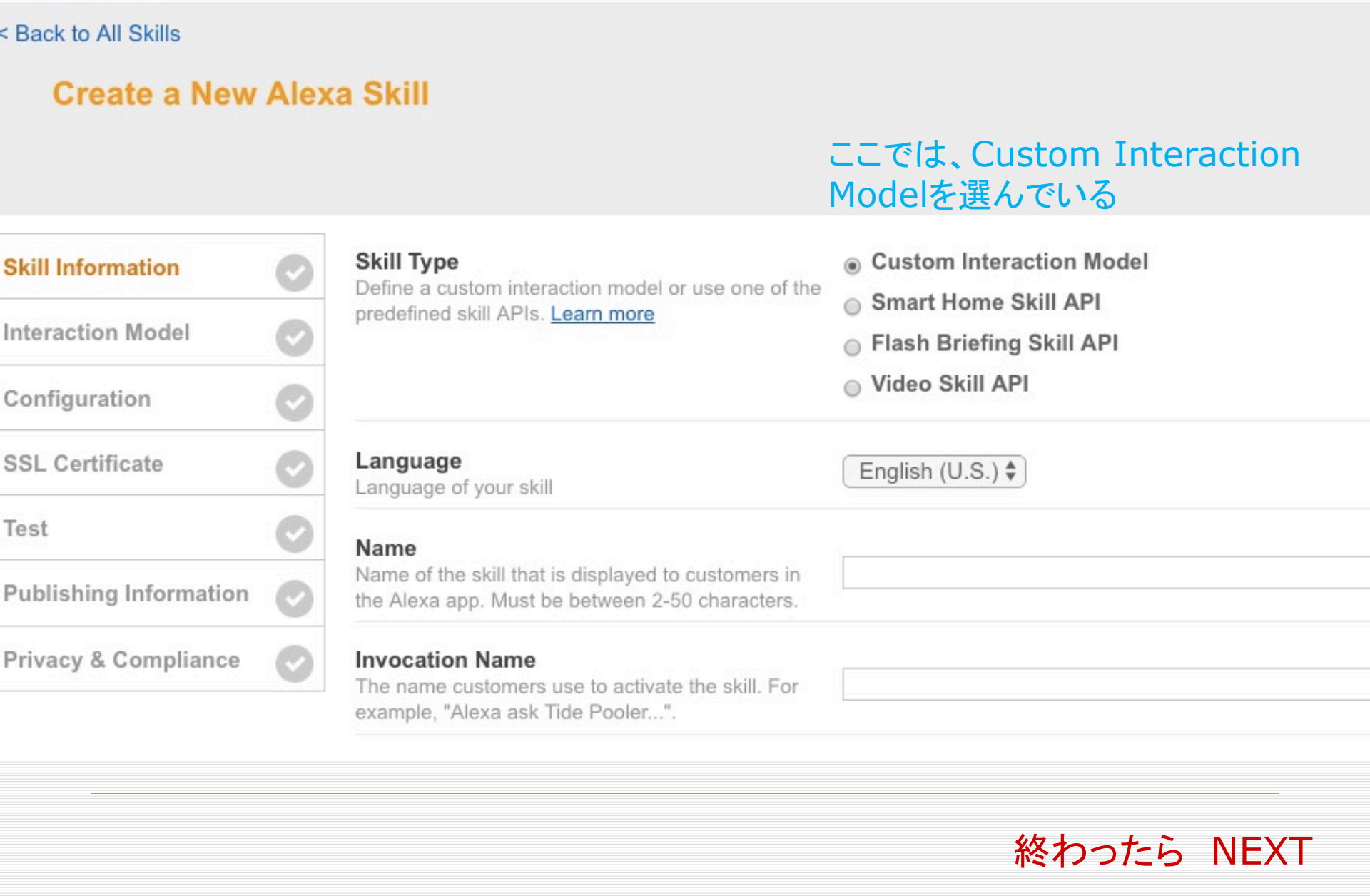


Alexa Skill KitのGet Startedを選ぶ

Add a New Skillを押す



Skill Type, Language, Name, Invocation Name を設定



終わったら　NEXT

ここでは、Skill Builder (beta)を使う

English (U.S.) Add a New Language



Try the skill builder (beta), an intuitive interface for building your interaction model and creating dialog prompts

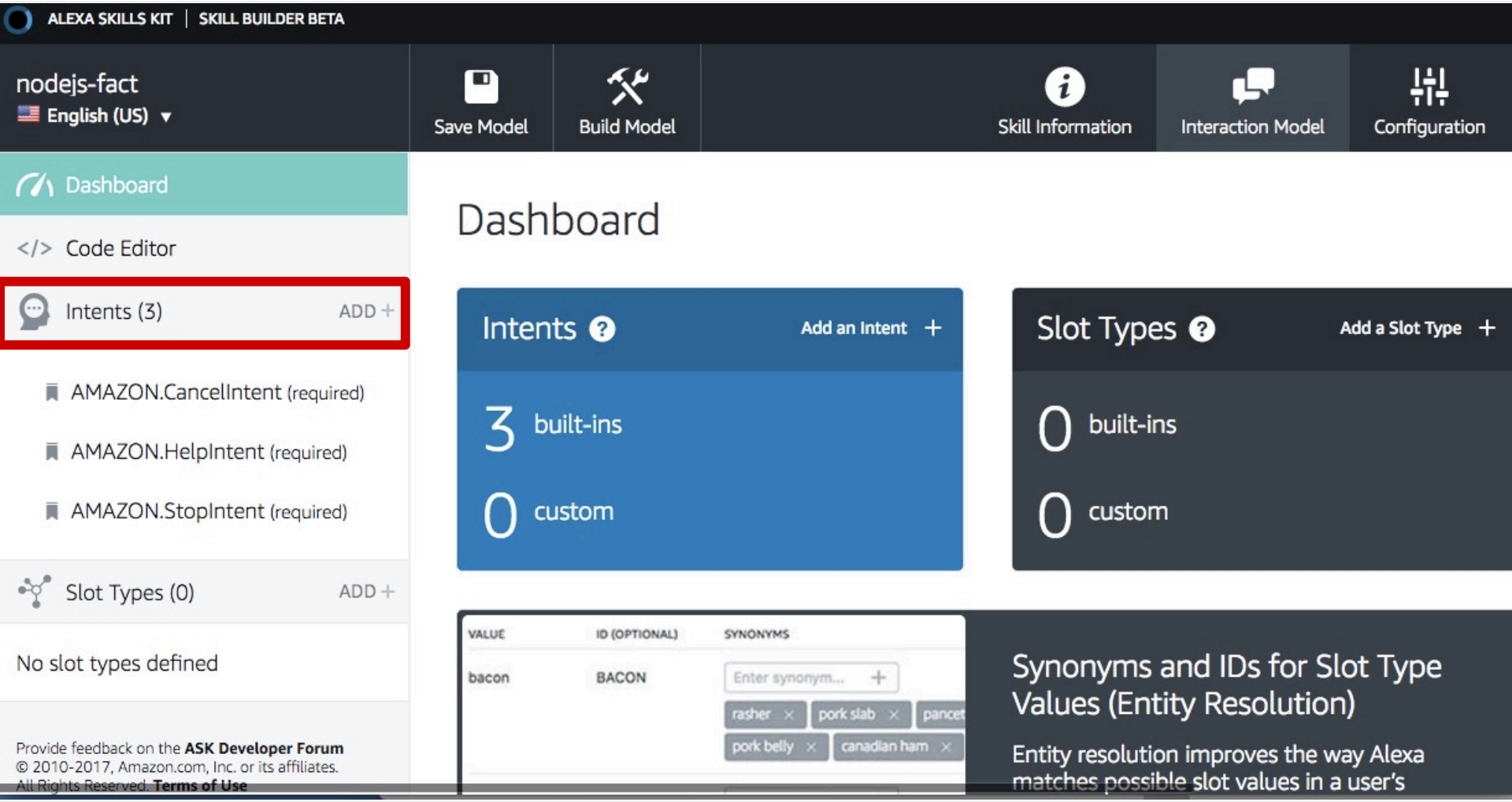
Launch Skill Builder BETA

Intent Schema

The schema of user intents in JSON format. For more information, see Intent Schema.
Also see built-in slots and built-in intents.

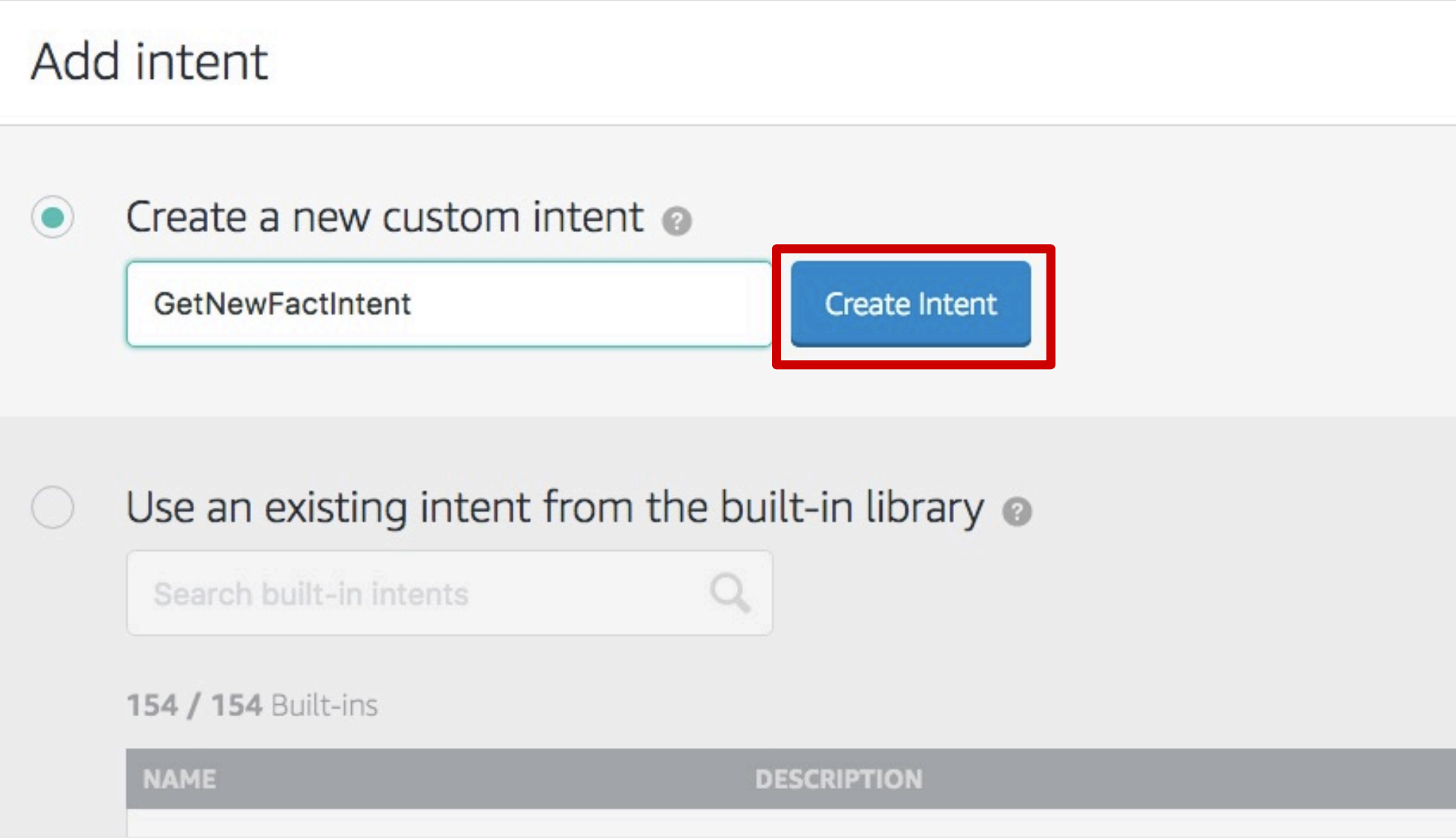
1

Skill Builder (beta) のDashboard



IntentsのADDを押す

新しいIntentの名前を入れて、Create Intent を押す



GetNewFactIntent

Utteranceを追加する

Sample Utterances (3)

Tell me a space fact

"Tell me something"

"Tell me a fact"

"Give me a fact"

Intent confirmation (optional)

Save Modelを押して、Build ボタンを押す



Interactionモデルのビルトができたら、Configuration ボタンを押す

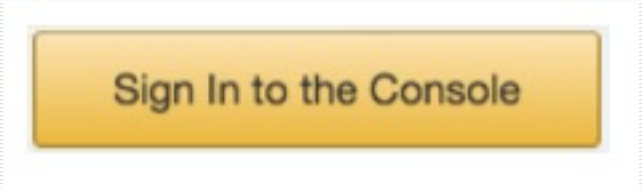




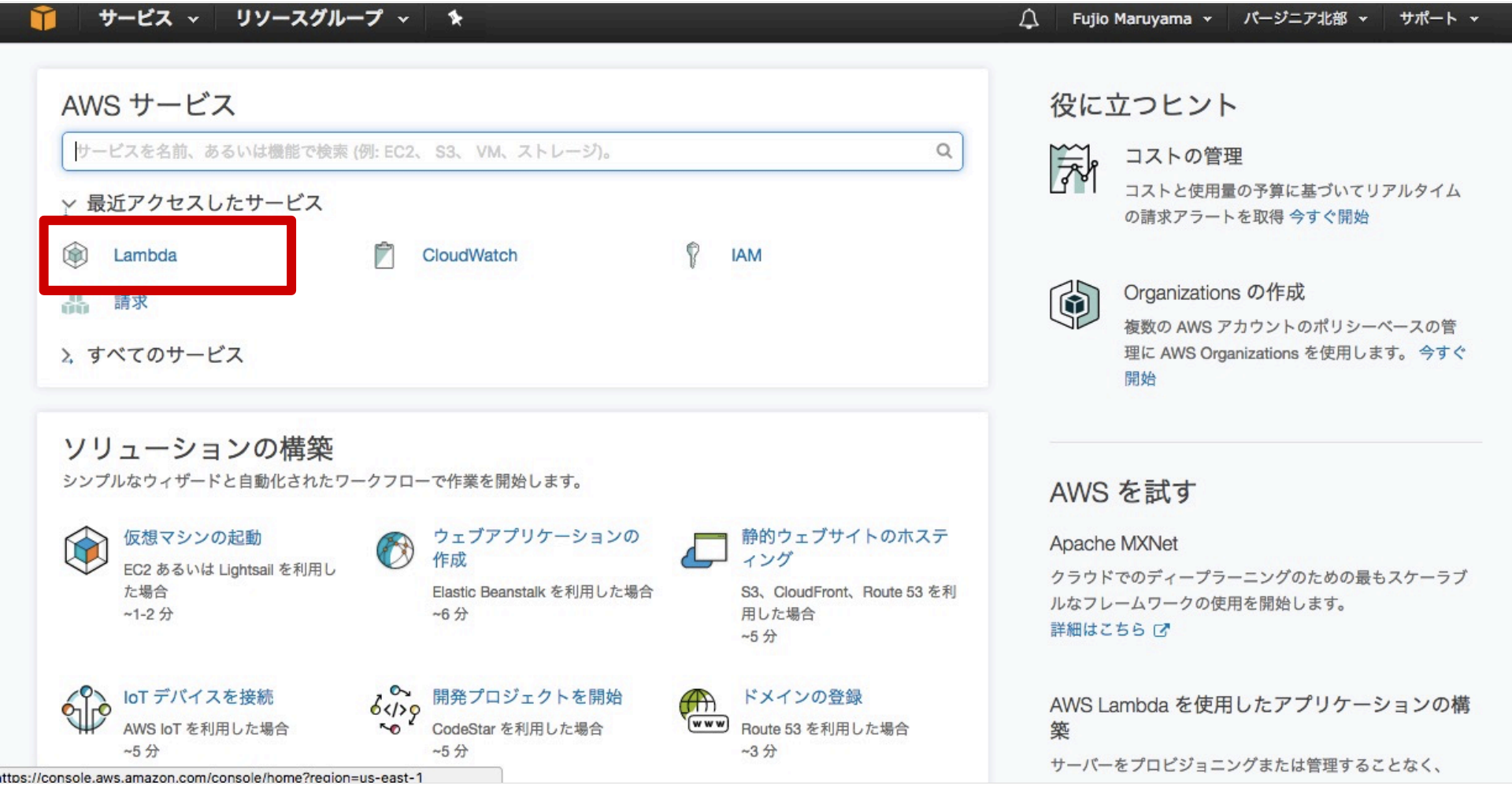
Lamdaの設定は、AWS Developper Consoleから行う

Lambda Function の設定

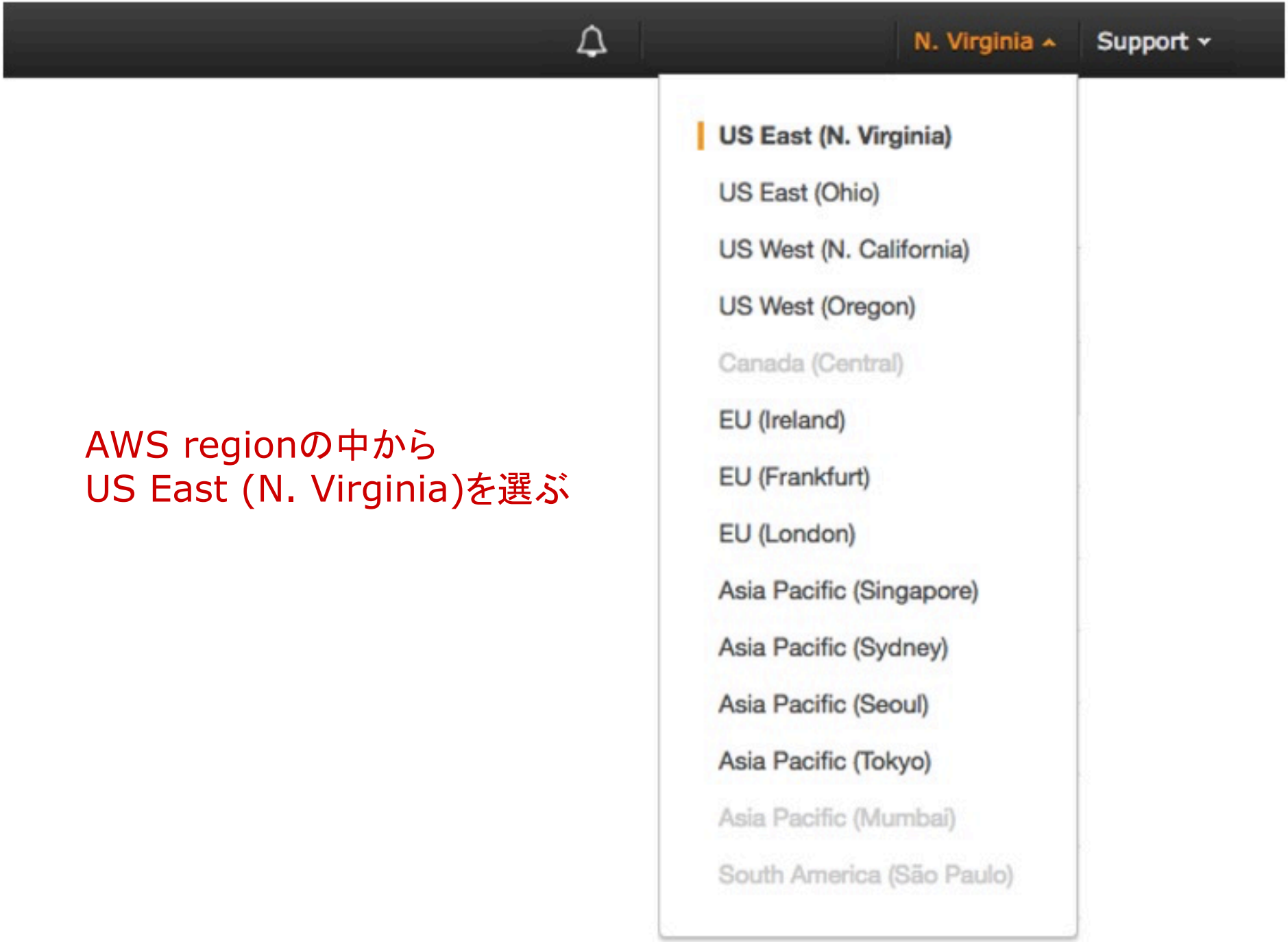
- http://aws.amazon.com で、コンソールにサインイン



AWS サービスの中から Lambdaを選ぶ



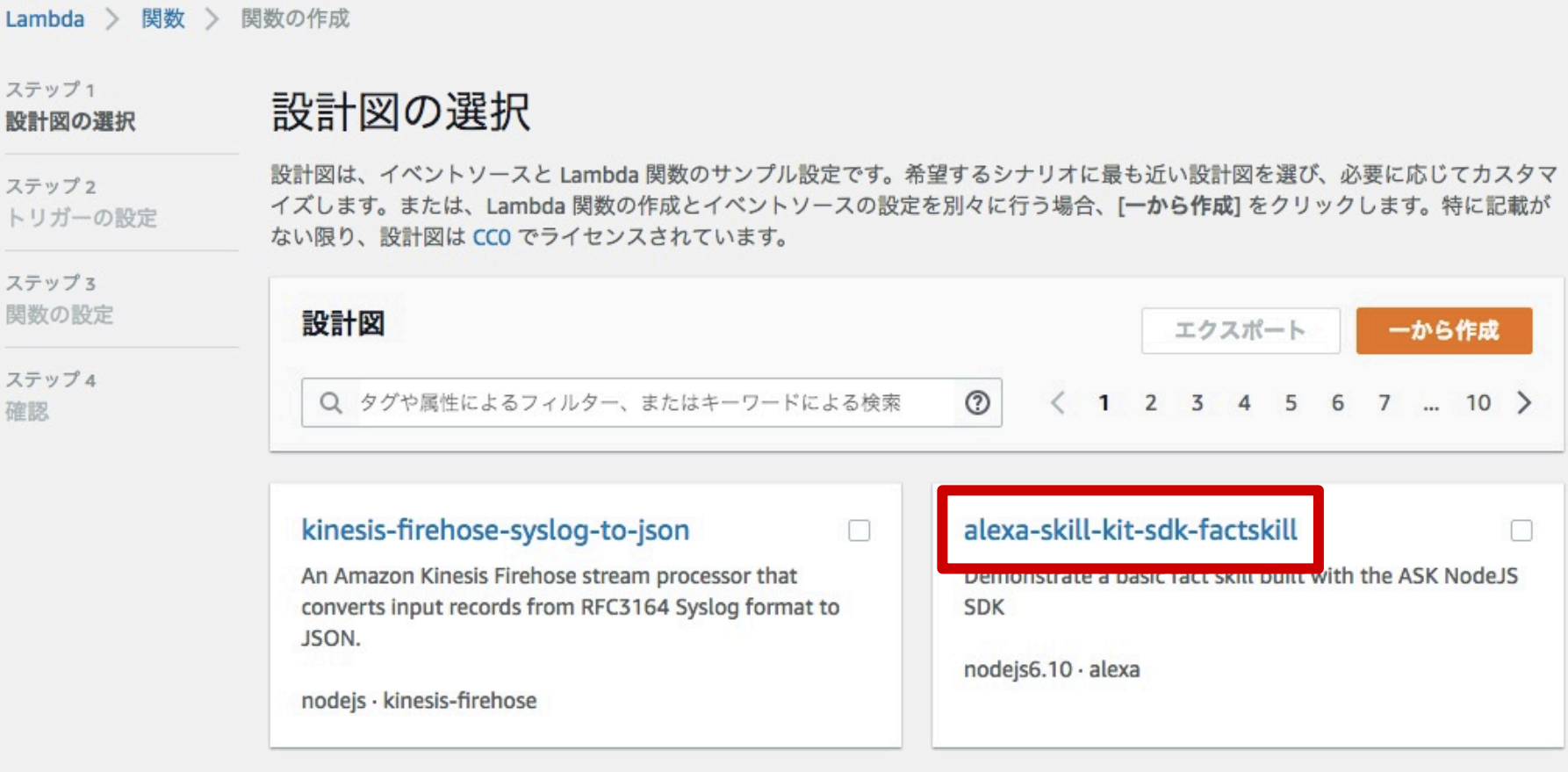
AWS regionの中から
US East (N. Virginia)を選ぶ



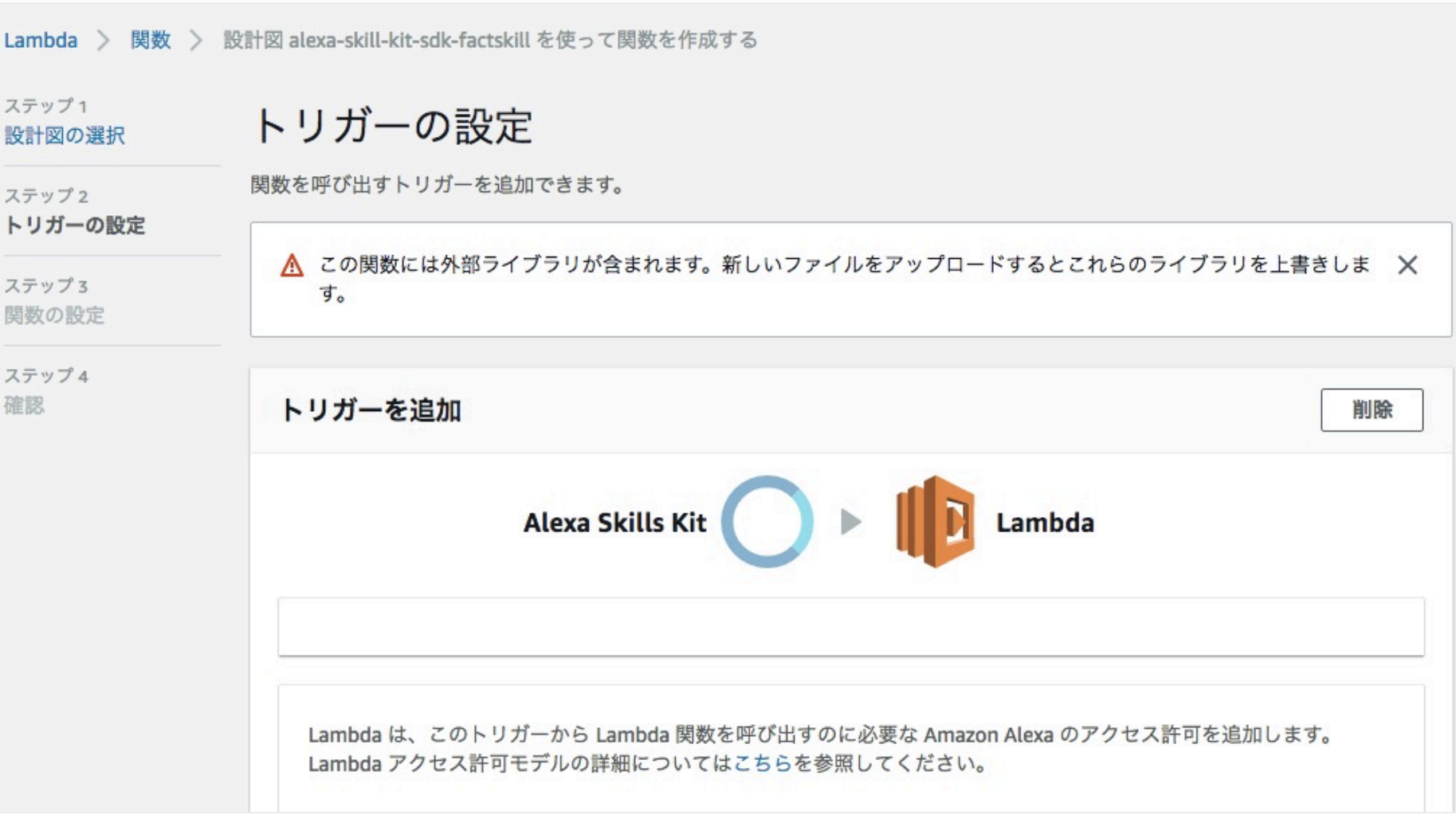
関数の作成ボタンを押す



「設計図の選択」の画面から、“alexa-nodejs-template”を選ぶ



「トリガーの設定」で、“Alexa Skills Kit” を選ぶ



終わったら NEXT

関数の設定を行う。関数に名前をつける



コードを設定する。デフォールトのコードが
用意されているので、必要に応じて修正する

Lambda 関数のコード

関数のコードを指定します。コードに（aws-sdk 以外の）カスタムライブラリが必要ない場合は、エディターを使用します。カスタムライブラリが必要な場合は、コードとライブラリを .ZIP ファイルとしてアップロードすることができます。Lambda 関数のデプロイの詳細については、こちらを参照してください。

コード エントリ タイプ

コードをインラインで編集

```
1  /* eslint-disable  func-names */
2  /* eslint quote-props: ["error", "consistent"]*/
3  /**
4   * This sample demonstrates a simple skill built with the Amazon Alexa Skills
5   * nodejs skill development kit.
6   * This sample supports multiple lauguages. (en-US, en-GB, de-DE).
7   * The Intent Schema, Custom Slots and Sample Utterances for this skill, as well
8   * as testing instructions are located at https://github.com/alexa/skill-sample-nodej
9   **/
10
11 'use strict';
12
13 const Alexa = require('alexa-sdk');
14
15 const APP_ID = undefined;  // TODO replace with your app ID (OPTIONAL).
16
17 const languageStrings = {
18     'en': {
```

関数のハンドラ、ロールの設定を行う。

Lambda 関数ハンドラおよびロール

ハンドラ*
作成した関数の module-name.export 値です。たとえば、"index.handler" は index.js 内の exports.handler を呼び出します。

index.handler

ロール*
関数のアクセス許可を定義します。新しいロールは、作成後の数分間は使用できないことがあります。Lambda 実行ロールの詳細については、こちらを参照してください。

既存のロールを選択

既存のロール*
この関数で既存のロールを使用できます。ロールは Lambda によって引き受けられる必要があり、CloudWatch ログ権限を持っていなければならないことに注意してください。

lambda_basic_execution

終わったら　NEXT

関数の設定の確認画面が出る。
問題がなければ、「関数の作成」ボタンを押す

ロール
lambda_basic_execution

詳細設定

メモリ (MB)
128

タイムアウト
7

VPC
非 VPC

アクティブトレースを有効にします
いいえ

KMS キー
(デフォルト) aws/lambda

キャンセル　戻る　関数のエクスポート　関数の作成

ARNの値をコピーする



テストイベントの入力

以下のエディタを使用して、作成した関数をテストするためのイベントを入力します。イベントは [アクション] リストの [**テストイベントの設定**] を選択して再び編集できます。イベントへの変更はローカルにのみ保存されます。

サンプルイベントテンプレート

Alexa Intent - GetNewFact

```
{
  "session": {
    "new": false,
    "sessionId": "amzn1.echo-api.session.[unique-value-here]",
    "attributes": {},
    "user": {
      "userId": "amzn1.ask.account.[unique-value-here]"
    },
    "application": {
      "applicationId": "amzn1.ask.skill.[unique-value-here]"
    }
  },
  "version": "1.0",
  "request": {
    "locale": "en-US",
    "timestamp": "2016-10-27T21:06:28Z",
    "type": "IntentRequest",
    "requestId": "amzn1.echo-api.request.[unique-value-here]",
```

キャンセル　保存　保存してテスト


Lambda > 関数 > MyFactFunction
ARN - arn:aws:lambda:us-east-1:248065776263:function:MyFactFunction
MyFactFunction
限定条件
アクション
テスト
実行結果: 成功 (ログ)
詳細
コード
設定
トリガー
タグ
モニタリング
Alexa Skills Kit
alexa-appkit.amazon.com
削除
Alexa スキルを設定するには、Alexa 開発者ポータルに移動してください。
トリガーを追加
トリガーの更新
関数ポリシーを表示

Lambda > 関数 > MyFactFunction

ARN - arn:aws:lambda:us-east-1:248065776263:function:MyFactFunction

MyFactFunction

限定条件 ▼ | アクション ▼ | テスト

実行結果: 成功 (ログ)

▼ 詳細

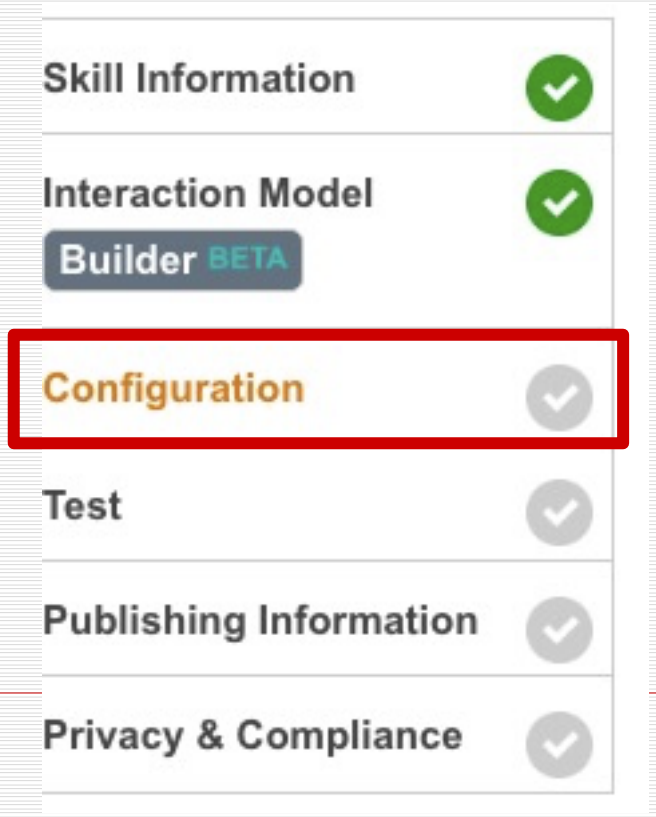
関数の実行から返された結果が以下のエリアに表示されます。関数から結果を返す方法の詳細については、こちらを参照してください。


```
{
  "version": "1.0",
  "response": {
    "shouldEndSession": true,
    "outputSpeech": {
      "type": "SSML",
      "ssml": "<speak> Here's your fact: The Sun contains 99.86% of the mass in the Solar System. </speak>"
    },
    "card": {
      "type": "Simple",
      "title": "American Space Facts"
```


Next: Connect VUI to Code

先に作成したVoice User Interface を Lambda Functionに接続する

- Amazon Developer Portal に戻ってskillをリストから選ぶ
- “Configuration” tab を開く



English (U.S.) | Add a New Language

- Skill Information
- Interaction Model
  Builder BETA
- Configuration
- Test
- Publishing Information
- Privacy & Compliance

Skills Beta Testing NEW
Status: Not yet eligible

Global Fields

These fields apply to all languages supported by the skill.

Endpoint

Service Endpoint Type:

- AWS Lambda ARN (Amazon Resource Name)
  Recommended
  AWS Lambda is a server-less compute service that runs your code in response to events and automatically manages the underlying compute resources for you.
  More info about AWS Lambda
  How to integrate AWS Lambda with Alexa
- HTTPS

EndPointに"AWS Lambda ARN"を選び、以前保存したARNを記入する

Default

arn:aws:lambda:us-east-1:248065776263:function:MyFactFunction

Provide geographical region endpoints? (Optional)

Yes No

Save changes

There are changes on this page, do you want to save these changes before leaving the current page?

Cancel | Discard | Save

Account Linkingは “No”



終わったら　NEXT

English (U.S.) Add a New Language

Please complete the Interaction Model tab to start testing this skill.

Enabled This skill is enabled for testing on your account.

Once you have completed testing on your device, please complete the Description and Publishing Information tab, then submit the skill for certification.

If it passes Amazon's testing and certification process, it will become available to Alexa end users.

You will be able to see your skill in the Skills tab in Alexa App and you can enable the skill and start testing.

After completing your testing please submit the skill for certification. If it passes Amazon's testing and certification process, it will become available to Alexa end users

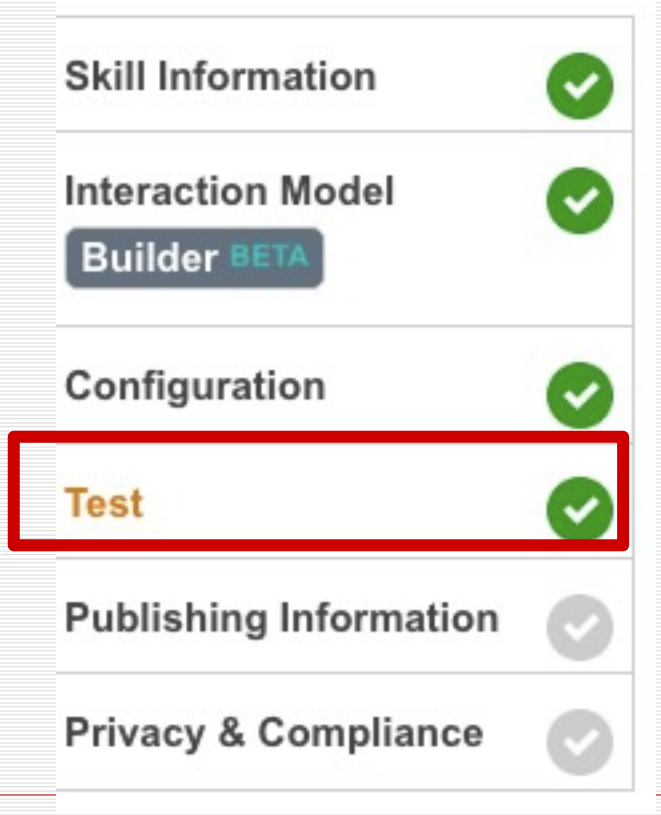
The skill is available in "Skills > Your Skills" page of the Alexa App when you select 'Yes' above. You can then enable the skill and test its functionality by asking Alexa, **ask fact**

For successful Alexa Skills Certification, please test for our requirements on Session Management and Error Handling.

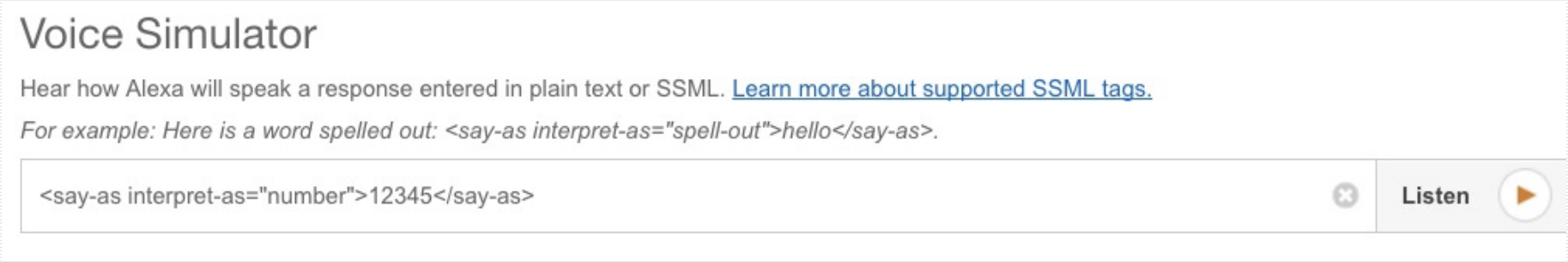
Next: Testing

Alexa Skill をテストする

- Amazon Developer Portal に戻り、”Test” タブを開く



Voice Simulator




```
<say-as interpret-as="number">12345</say-as>
```

```
<say-as interpret-as="ordinal">12345</say-as>
```

```
<say-as interpret-as="digits">12345</say-as>
```

Test your skill with the Service Simulator.

Service Simulator

Use Service Simulator to test your HTTPS endpoint: arn:aws:lambda:us-east-1:248065776263:function:MyFactFunction

Note: Service Simulator does not currently support testing audio player directives, dialog model, customer permissions and customer account linking.

Text | JSON

Enter Utterance

give me a fact

Ask nodejs-fact | Reset

Service Request


```
{
 "session": {
  "new": true,
  "sessionId": "SessionId.4b30effb-5970-43a5-8eef-97ea
  "application": {
   "applicationId": "amzn1.ask.skill.22212cda-1348-410b
  },
  "attributes": {},
  "user": {
   "userId": "amzn1.ask.account.AEXP6JLWYX3PRZCC25Z
```


Service Response


```
{
 "version": "1.0",
 "response": {
  "outputSpeech": {
   "ssml": "<speak> Here's your fact: The Moon is movi
   "type": "SSML"
  },
  "card": {
   "content": "The Moon is moving approximately 3.8 c
   "title": "American Space Facts"
```

Echo Show Render Response (beta)

American Space Facts

The Moon is moving approximately 3.8 cm away from our planet every year.

American Space Facts

Jupiter has the shortest day of all the planets.

Next: Customization

Customize the Skill to be Yours

- 新しいデータの追加。
  index.js に、データが埋め込まれている。

```
'use strict';
var Alexa = require('alexa-sdk');

//=========================================================================================================================================
//TODO: The items below this comment need your attention.
//=========================================================================================================================================

//Replace with your app ID (OPTIONAL).  You can find this value at the top of your skill's page on http://developer.amazon.com.
//Make sure to enclose your value in quotes, like this: var APP_ID = "amzn1.ask.skill.bb4045e6-b3e8-4133-b650-72923c5980f1";
var APP_ID = undefined;

var SKILL_NAME = "Space Facts";
var GET_FACT_MESSAGE = "Here's your fact: ";
var HELP_MESSAGE = "You can say tell me a space fact, or, you can say exit... What can I help you with?";
var HELP_REPROMPT = "What can I help you with?";
var STOP_MESSAGE = "Goodbye!";

//=========================================================================================================================================
//TODO: Replace this data with your own.  You can find translations of this data at http://github.com/alexa/skill-sample-node-js-fact/data
//=========================================================================================================================================
var data = [
    "A year on Mercury is just 88 days long.",
    "Despite being farther from the Sun, Venus experiences higher temperatures than Mercury.",
    "Venus rotates counter-clockwise, possibly because of a collision in the past with an asteroid.",
    "On Mars, the Sun appears about half the size as it does on Earth.",
    "Earth is the only planet not named after a god.",
```

```
4  //=========================================================================
5  //TODO: The items below this comment need your attention.
6  //=========================================================================
7
8  //Replace with your app ID (OPTIONAL).  You can find this value at the
9  //Make sure to enclose your value in quotes, like this: var APP_ID = '
10 var APP_ID = undefined;
11
12 var SKILL_NAME = "Space Facts";
13 var GET_FACT_MESSAGE = "Here's your fact: ";
14 var HELP_MESSAGE = "You can say tell me a space fact, or, you can say
15 var HELP_REPROMPT = "What can I help you with?";
16 var STOP_MESSAGE = "Goodbye!";
17
```

```
18 //=========================================================================
19 //TODO: Replace this data with your own.  You can find translations of this data at
20 //=========================================================================
21 var data = [
22     "A year on Mercury is just 88 days long.",
23     "Despite being farther from the Sun, Venus experiences higher temperatures than
24     "Venus rotates counter-clockwise, possibly because of a collision in the past wi
25     "On Mars, the Sun appears about half the size as it does on Earth.",
26     "Earth is the only planet not named after a god.",
```

```
47  var handlers = {
48      'LaunchRequest': function () {
49          this.emit('GetNewFactIntent');
50      },
51      'GetNewFactIntent': function () {
52          var factArr = data;
53          var factIndex = Math.floor(Math.random() * factArr.length);
54          var randomFact = factArr[factIndex];
55          var speechOutput = GET_FACT_MESSAGE + randomFact;
56          this.emit(':tellWithCard', speechOutput, SKILL_NAME, randomFact)
57      },
58      'AMAZON.HelpIntent': function () {
59          var speechOutput = HELP_MESSAGE;
60          var reprompt = HELP_REPROMPT;
61          this.emit(':ask', speechOutput, reprompt);
62      },
63      'AMAZON.CancelIntent': function () {
64          this.emit(':tell', STOP_MESSAGE);
65      },
66      'AMAZON.StopIntent': function () {
67          this.emit(':tell', STOP_MESSAGE);
68      }
```

Intent Signature

Schema.orgのEntity Modelの採用

```
Action<propertyName@EntityType[propertyName]>
```

Built-in Intent Library

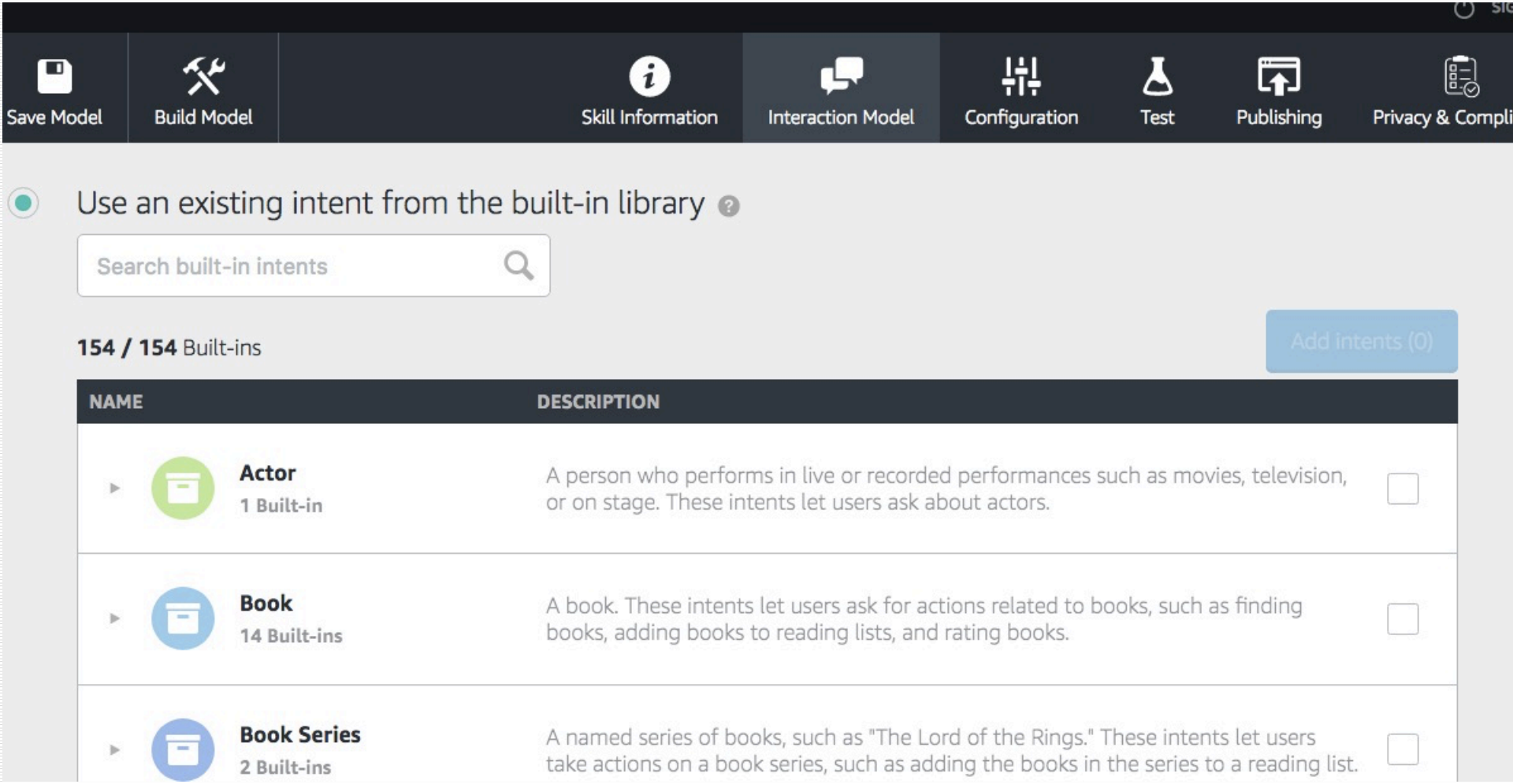
Alexa Skills Kit は、あらかじめ用意されたビルドインの Intent のライブラリーを提供する。ユーザは、発話のサンプルを用意することなく、このIntentの機能をSkillの中で利用することができる。

https://goo.gl/Dw39Bg

built-in intent libraryとschema.org

The built-in intent library incorporates material from Schema.org, which is licensed under the Creative Commons Attribution-ShareAlike License (version 3.0) (the "License"). You may not use this file except in compliance with the License. You may obtain a copy of the License at http://creativecommons.org/licenses/by-sa/3.0/. For questions, please reach out to alexa-ontology-support@amazon.com.

次のような、build-in Intentが用意されている



Actor 1 Built-in	A person who performs in live or recorded performances such as movies, television, or on stage. These intents let users ask about actors.	☐
Book 14 Built-ins	A book. These intents let users ask for actions related to books, such as finding books, adding books to reading lists, and rating books.	☐
Book Series 2 Built-ins	A named series of books, such as "The Lord of the Rings." These intents let users take actions on a book series, such as adding the books in the series to a reading list.	☐
Calendar 2 Built-ins	A calendar that contains a list of events. These intents let users make requests related to calendars, such as reading back events, looking up events, and adding items to a calendar.	☐
Catalog 1 Built-in	A collection or library offered and curated by an organization or collected and organized by an individual. These intents let users ask about collections of books or music.	☐
Creative Work 3 Built-ins	A generic kind of creative work, including books, movies, photographs, software programs, and so on. These intents let users make requests related to any type of creative work.	☐
Creative Work Section 8 Built-ins	A section or part of a creative work, such as a page or chapter. These intents let users specify a particular part of a book, movie, page, or other section.	☐

Creative Work Section 8 Built-ins	A section or part of a creative work, such as a page or chapter. These intents let users specify a particular part of a book, movie, page, or other section.	☐
Episode 6 Built-ins	An episode for TV, radio, video game, or other media which can be part of a series or season. These intents let users request actions related to episodes, such as starting playback of a particular episode or searching for a particular episode.	☐
Event 8 Built-ins	An event happening at a certain time and location, such as a meeting, concert, lecture, or festival. These intents let users make requests related to events, such as adding an event to a calendar, or looking up details of an event.	☐
Local Business 4 Built-ins	A particular physical business or branch of an organization, like a restaurant, a medical practice, or a store. These intents let users ask for information about businesses such as the location, contact information, or opening hours.	☐
Musician 1 Built-in	A person or group that performs music. These intents let users request actions related to musicians, such as asking to rate or "like" a musician.	☐
Music Creative Work 14 Built-ins	The broad category of creative works that are musical in nature, includes songs, albums, playlists, and other items. These intents let users request actions related to musical works, such as adding songs to a library, controlling playback of musical works, and looking up information about musical works.	☐
Music Group 2 Built-ins	A musical group, such as a band, an orchestra, or a choir. This can also represent a solo musician. These intents let users request actions related to a music group, such as "liking" a group, finding songs by a group, or asking about the members of a	☐

Music Playlist 10 Built-ins	A collection of music tracks in playlist form. These intents let users make requests to manipulate playlists by adding, removing, and choosing items.	☐
Music Recording 5 Built-ins	A music recording (track), usually a single song. These intents let users ask for specific information about songs, such as the artist who performs the song, the genre of the song, or other information.	☐
Playback Mode 2 Built-ins	Represents modes related to playback, such as shuffle and loop. These intents let users ask to turn these modes on and off.	☐
Reading List 7 Built-ins	A list made up of books or other written creative works that a user wants to collect in one place, such as a summer reading list or a "read later" list. These intents let users make requests about reading lists, such as creating lists and adding or removing items on the lists.	☐
Screening Event 4 Built-ins	A screening of a movie or other video. These intents let users make requests related to particular screening events, such as asking for the show time for an event, asking what movies are playing, and asking what movie theaters are playing a particular movie.	☐
Singer 1 Built-in	A person who sings, either as a solo performer or as part of a group. These intents let users request actions related to singers, such as asking who sings a particular song.	☐
Standard 22 Built-ins	The standard intents are used for common, general actions such as stopping, canceling, and asking for help.	☐

Thing 5 Built-ins	A generic object. These intents let users make requests that are not related to any specific object or entity, such as requests to pause or continue an action.	☐
TV Season 1 Built-in	Season during which a series of related episodes are broadcast on TV broadcast or delivered online. These intents let users request actions for a TV season, such asking to start playback of a particular season for a show.	☐
TV Series 2 Built-ins	A series of related television episodes. These intents let users request actions for a TV series, such asking to start playback of a particular series.	☐
Video Creative Work 23 Built-ins	A creative work that includes video and possibly audio, such as movies, television shows, and video clips. These intents let users request actions related to video works, such as adding videos to a library or watchlist, controlling playback of videos, and looking up information about video works.	☐
Video Playlist 3 Built-ins	A collection of video creative works, such as a list of shows to watch. These intents let users make requests to manipulate video playlists by adding, removing, and choosing items.	☐
Weather Forecast 3 Built-ins	A weather forecast. These intents let users ask for general weather forecasts, as well as more specific questions about specific weather conditions.	☐

Understanding the Structure of the Built-in Intent Library

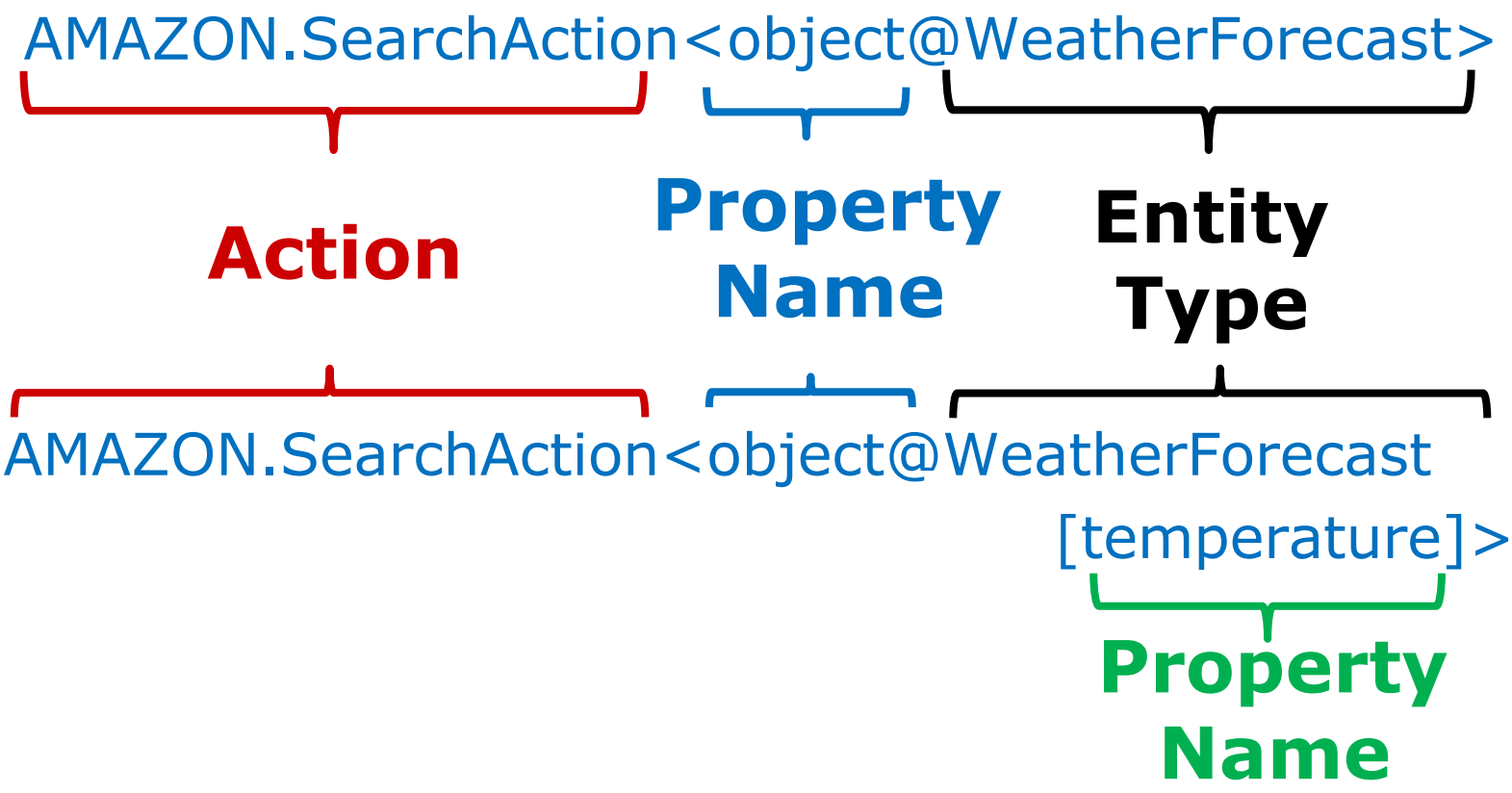
https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/docs/understanding-the-structure-of-the-built-in-intent-library

Intent Signature Syntax

- built-in intent は、***intent signature***を持つ。このsignatureを、intentの名前としてSchemaに書くことができる。次のサンプルは、二つのWeatherForecast intents をもった、Schemaの例である。

```
{
  "intents": [
    {
      "intent": "AMAZON.SearchAction<object@WeatherForecast>"
    },
    {
      "intent": "AMAZON.SearchAction<object@WeatherForecast[temperatu
    }
  ]
}
```

Intent Signature

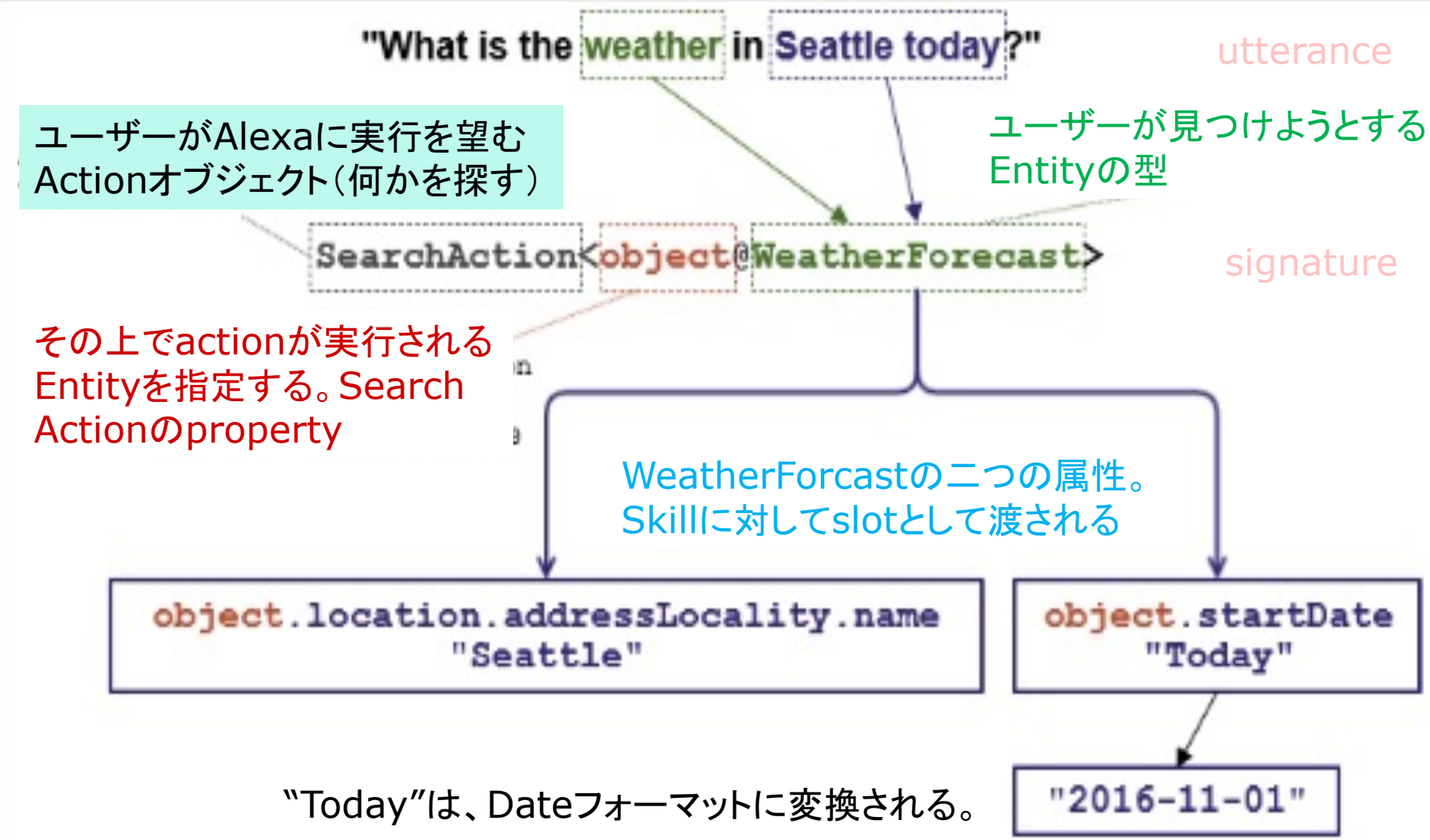



```
Action<propertyName@EntityType[propertyName]>
```

- **AMAZON**: all built-in intents begin with the AMAZON namespace.
- **SearchAction**: an ***action*** that identifies the action the user wants Alexa to take, such as searching for information.
- **object**: a **property of SearchAction**, for identifying the **entity** on which it should operate.
- **WeatherForecast**: the **entity type** for object. This identifies the entity the user wants to search for.
- **[temperature]:** (optional) if included, specifies an attribute to narrow the scope of the intent. This creates an intent that is triggered when the utterance mentions the specified attribute.

For instance, the intent AMAZON.SearchAction<object@WeatherForecast[temperature]>would be triggered by "what's today's temperature," but not the more generic "what's the weather today."

Utterance と Signature



```
{
  "intents": [
    {
      "intent": "AMAZON.SearchAction<object@WeatherForecast>"
    }
  ]
}
```

- このSchemaには、slotが定義されていないのだが、“what’s the weather **today** in **Seattle**” のような発話は、今日の日付と都市 “Seatle”をSlotに含んだRequestをSkillに送る。
- これらのIntentは、**Actionと、Entityの型と、プロパティ**から構成されている。個々のIntentの名前は（WeatherForcastは、Built-in-Intentの名前である ）、これらの要素と一緒に結合されて一つの**Intent Signature**（例えば AMAZON.SearchAction<object@Weather Forecast>といった）にまとめられている。このSignatureをIntentの名前として利用できる。

Actions, Entity Types, and Properties

- An ***action*** represents an action the user is trying to take, such as searching for information.
- For instance, **SearchAction** represents the "searching for information" action. An intent defined with SearchAction indicates that the user wants search or look up information, such as finding a weather forecast or finding the phone number for a nearby coffee shop.
- An intent defined with **AddAction** indicates that the user wants to add information somewhere, such as adding an event to a calendar or adding the title a book to a reading list.

Actions, **Entity Types**, and **Properties**

- Most **actions** operate *on* other objects or *use* other objects. A search action needs to identify the information to search for. An add action needs to identify the item to add and possibly also the list to which to add it.
- These items are specified using the ***properties*** of the action.
- For example, SearchAction has an **object** property that identifies the ***type*** of information the user wants to find.
- AddAction has a **targetCollection** property to identify the **type** of the target list and an object property to identify the **type** object to add to the list.

Actions, **Entity Types**, and Properties

- An ***entity type*** represents the objects that the actions can act on or use.
- **Entity types** also have **attributes**, which are represented by **properties**.
- For example, WeatherForecast is an **entity type** that defines a weather forecast. It has properties such as location, duration, and temperature (among others).

```
Action<propertyName@EntityType[propertyName]>
```

Identifying the parts of an utterance

- The entire utterance indicates that the user wants to *search for something* or *find information*. **Action**
- The "object" of the search – the information the user wants to find – is a *weather forecast*. This is the main *focus* or *purpose* in the sentence. **Entity**
- The words "Seattle" and "today" are attributes of the weather forecast the user wants to find. **Property**



SearchActionクラスは
情報を探すという意図を
表現する

SearchActionのobject
プロパティは、見つけるべき
Entityを表現する



WeatherForcastクラスは
天気予報を表現する

それは、場所（"Seatle"）や
日付（"Today"）といった属性を
表現しているプロパティを持つ

AMAZON.SearchAction<object@WeatherForecast>

このコンテキストでは、@サインは、object のプロパティは、
WeatherForecast型を持つことを宣言している

"Search for Weather Forecast."

複数のプロパティを持つAction

AMAZON.AddAction<object@Book,
targetCollection@ReadingList>

- このコンテキストでも、同様に、@サインは、**object**プロパティは**Book**型を持ち、**targetCollection**プロパティは**ReadingList**型を持つことを宣言している。

- この intent signatureは、次のように読める。

Add Book to Reading List.

Intent WeatherForecast の定義

- 型 **WeatherForecast** の定義は、次のところで与えられている。
  ”WeatherForecast Intents” https://goo.gl/LMEbqh

- つぎのようなslotを持つことがわかる。
  object.location.addressLocality.name
  object.location.addressCountry.name
  object.weatherCondition.name
  object.startDate
  object.duration
  object.location.addressRegion.name
  object.startTime
  object.type

WeatherForecast[temperature] のslot

- object.location.addressLocality.name
  object.location.addressCountry.name
  object.weatherCondition.name
  object.startDate
  object.duration
  object.location.addressRegion.name
  object.startTime
  object.temperature.type
  object.type

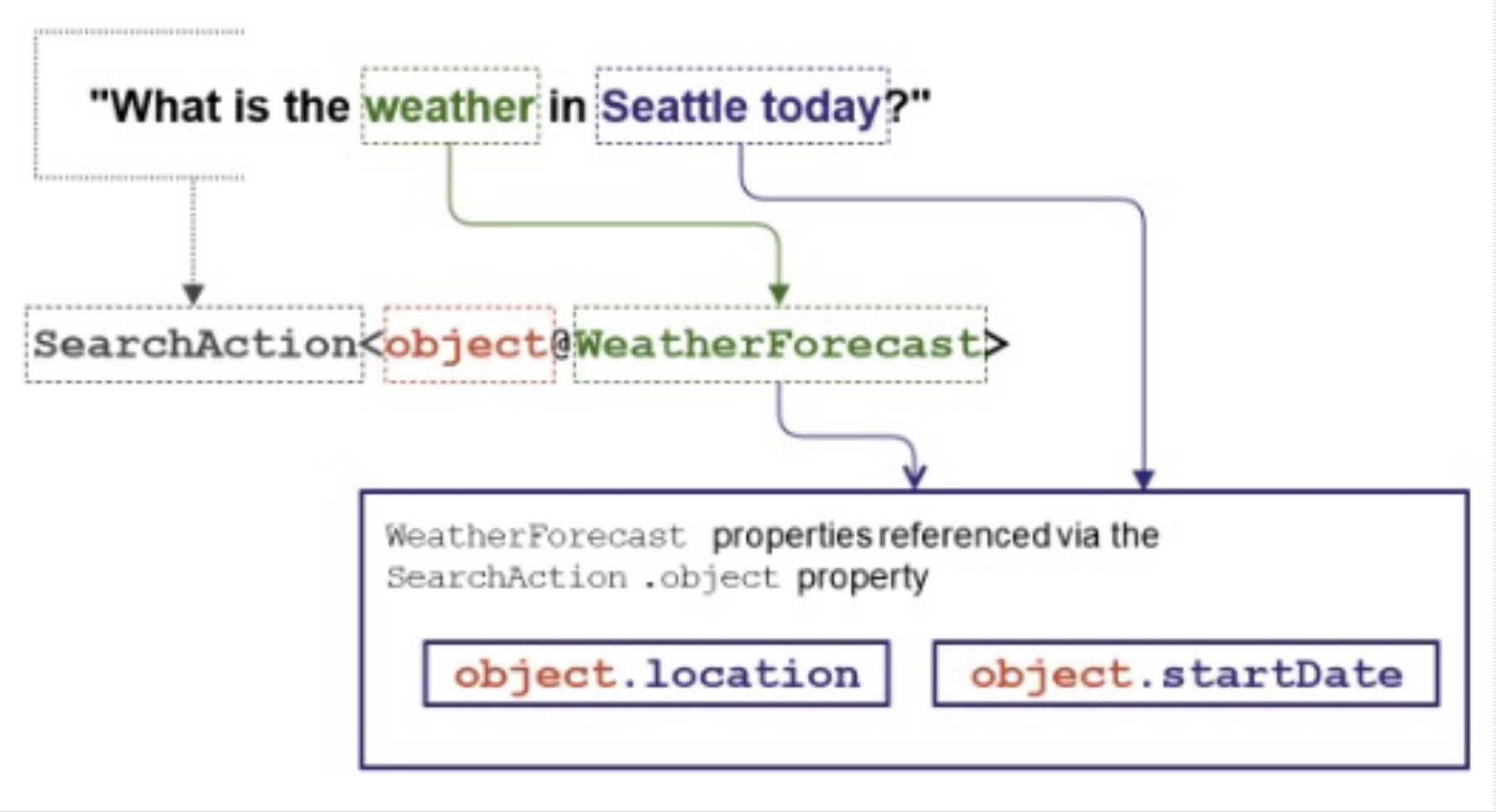
WeatherForecast[weatherCondition]のslot

- object.location.addressLocality.name
  object.spatialRelation
  object.location.addressCountry.name
  object.weatherCondition.name
  object.startDate
  object.duration
  object.location.addressRegion.name
  object.startTimeobject.type

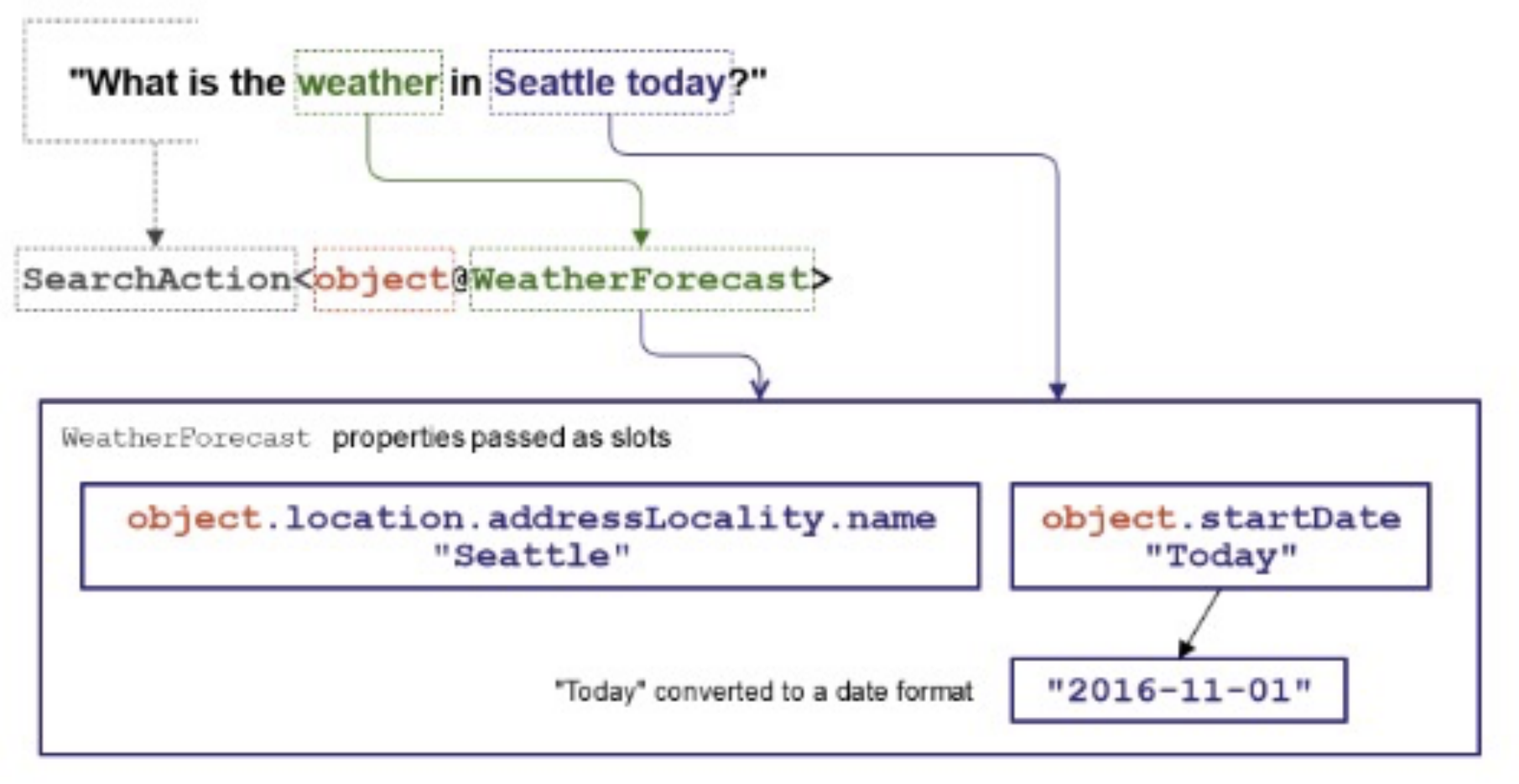
Intent Signatureが、Intentの名前として使われる時、Slotは具体的な値で埋められてから、Requestとして送られる

```
{
  "request": {
    "type": "IntentRequest",
    "intent": {
      "name": "AMAZON.SearchAction<object@WeatherForecast>",
      "slots": {
        "object.startDate": {
          "name": "object.startDate",
          "value": "2016-11-01"
        }
      }
    }
  }
}
```

Intentの持つプロパティの型をチェック



ある発話に対して、slotは、具体的な値を持つ



Intent Signatureが、Intentの名前として使われる時、Slotは具体的な値で埋められてから、Requestとして送られる

```
{
  "request": {
    "type": "IntentRequest",
    "intent": {
      "name": "AMAZON.SearchAction<object@WeatherForecast>",
      "slots": {
        "object.startDate": {
          "name": "object.startDate",
          "value": "2016-11-01"
        },
        "object.location.addressLocality.name": {
          "name": "object.location.addressLocality.name",
          "value": "Seattle"
        }
      }
    }
  }
}
```

Scoping an Intent with a Property Name

1. “What is the weather in Seattle today?” (respond with a general weather forecast).
2. “What is the temperature in Seattle today?” (respond with just the current temperature and expected high).
3. “When will it rain next in Seattle?” (respond with just the likelihood of precipitation).

```
{
  "intents": [
    {
      "intent": "AMAZON.SearchAction<object@WeatherForecast>"
    },
    {
      "intent": "AMAZON.SearchAction<object@WeatherForecast[temperatu
    },
    {
      "intent": "AMAZON.SearchAction<object@WeatherForecast[weatherCo
    },
    {
      "intent": "CustomIntent"
    }
  ]
}
```

Example: Creating a Weather Skill with the WeatherForecast Intents

- A general request for the weather:

  User: Alexa, ask My Weather what's the weather in Seattle today?

  My Weather: Light rain this afternoon through tomorrow, with temperatures falling to 56 degrees on Monday.

- A request for the expected temperature:

  User: Alexa, ask My Weather what will the high temperature be in Seattle today?

  My Weather: Today, expect a high of 65 degrees, and a low of 49.

- A question about a particular weather condition:

  User: Alexa, ask My Weather when will it rain next in Seattle?

  My Weather: Light rain is expected to start this afternoon, continuing until evening.

Intent Handlers for the Weather Skill

```
'AMAZON.SearchAction<object@WeatherForecast>': function () {
// Get the location and date slot values from the IntentRequest.
// Note syntax for getting JSON values with a dot (.) character in the name.
  var startDateSlotValue = this.event.request.intent.slots["object.startDate"].value;
  var cityLocationValue = this.event.request.intent.slots["object.location.addressLocality.name"].value;
// Validate the inputs and do any conversions as needed.
// Call the API for a weather service to lookup the weather
// Construct a general response with the high and low temperature,
// chance of precipitation, and comments around sun or clouds.
// If the location slots are empty, save any existing data in the session
// and return a response asking the user for a location.
  var output = "String with the weather output.";
  this.emit(':tellWithCard', output, "Weather", output);
```

Intent Handlers for the Weather Skill

```
'AMAZON.SearchAction<object@WeatherForecast[temperature]>':
function () {
// Code for the weather temperature handler goes here
  var output = "String with the weather forecast, explaining expected temperatures.";
  this.emit(':tellWithCard', output, "Weather - Temperature", output);


'AMAZON.SearchAction<object@WeatherForecast[weatherCondition]>':
function () {
// Code for the weather condition handler goes here. The weather condition
// the user asked about is in the 'object.weatherCondition.name' slot.
  var weatherCondition = this.event.request.intent.slots["object.weatherCondition.name"].value;
  var output = "String with the weather forecast, with details around the specified weather condition."
  this.emit(':tellWithCard', output, "Weather - Weather Condition", output);
```

Intent + Entity Modelの拡大

文の「意味」あるいは「意図」をどう捉えるか?
失敗と成功を振り返る

LUIS

Language Understanding Intelligent Service

https://www.luis.ai/home

Try LUIS now!

What the user says (utterances)

Book me a flight to Cairo

Order me 2 pizzas

Remind me to call my dad tomorrow

Where is the nearest club?



What LUIS returns


```
{
   "query": "Book me a flight to Cai
   "topScoringIntent": {
       "intent": "BookFlight",
       "score": 0.9887482
   },
   "intents": [
       {
           "intent": "BookFlight",
           "score": 0.9887482
       },
```


LUISは、発話に対して、IntentとEntityを返す！

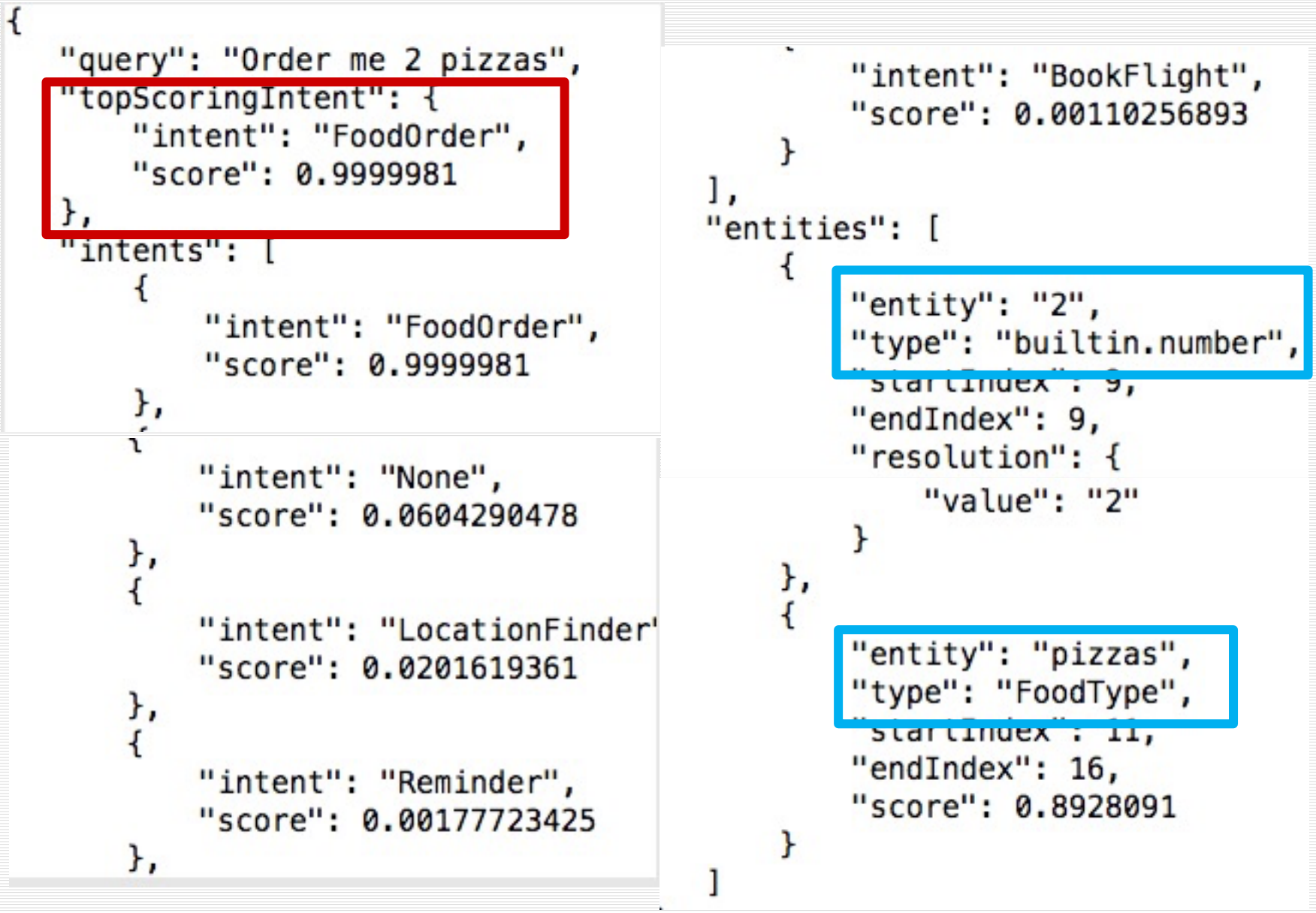
What is a LUIS app?

- A LUIS app is a place for a developer to define a custom language model. The output of a LUIS app is a web service with an HTTP endpoint that you reference from your client application to add natural language understanding to it. A LUIS app takes a user utterance and extracts intents and entities that correspond to activities in the client application's logic. Your client application can then take appropriate action based on the user intentions that LUIS recognizes.

“Book me a flight to Cairo”


```
{
    "query": "Book me a flight to Cai
    "topScoringIntent": {
        "intent": "BookFlight",
        "score": 0.9887482
    },
    "intents": [
        {
            "intent": "BookFlight",
            "score": 0.9887482
        },
        {
            "intent": "None",
            "score": 0.04272597
        },
        {
            "intent": "LocationFinder",
            "score": 0.0125702191
        },
        {
            "intent": "Reminder",
            "score": 0.00375502417
        }
        {
            "intent": "FoodOrder",
            "score": 3.765154E-07
        },
    ],
    "entities": [
        {
            "entity": "cairo",
            "type": "Location",
            "startIndex": 20,
            "endIndex": 24,
            "score": 0.956781447
        }
    ]
}
```

“Order me 2 pizzas”

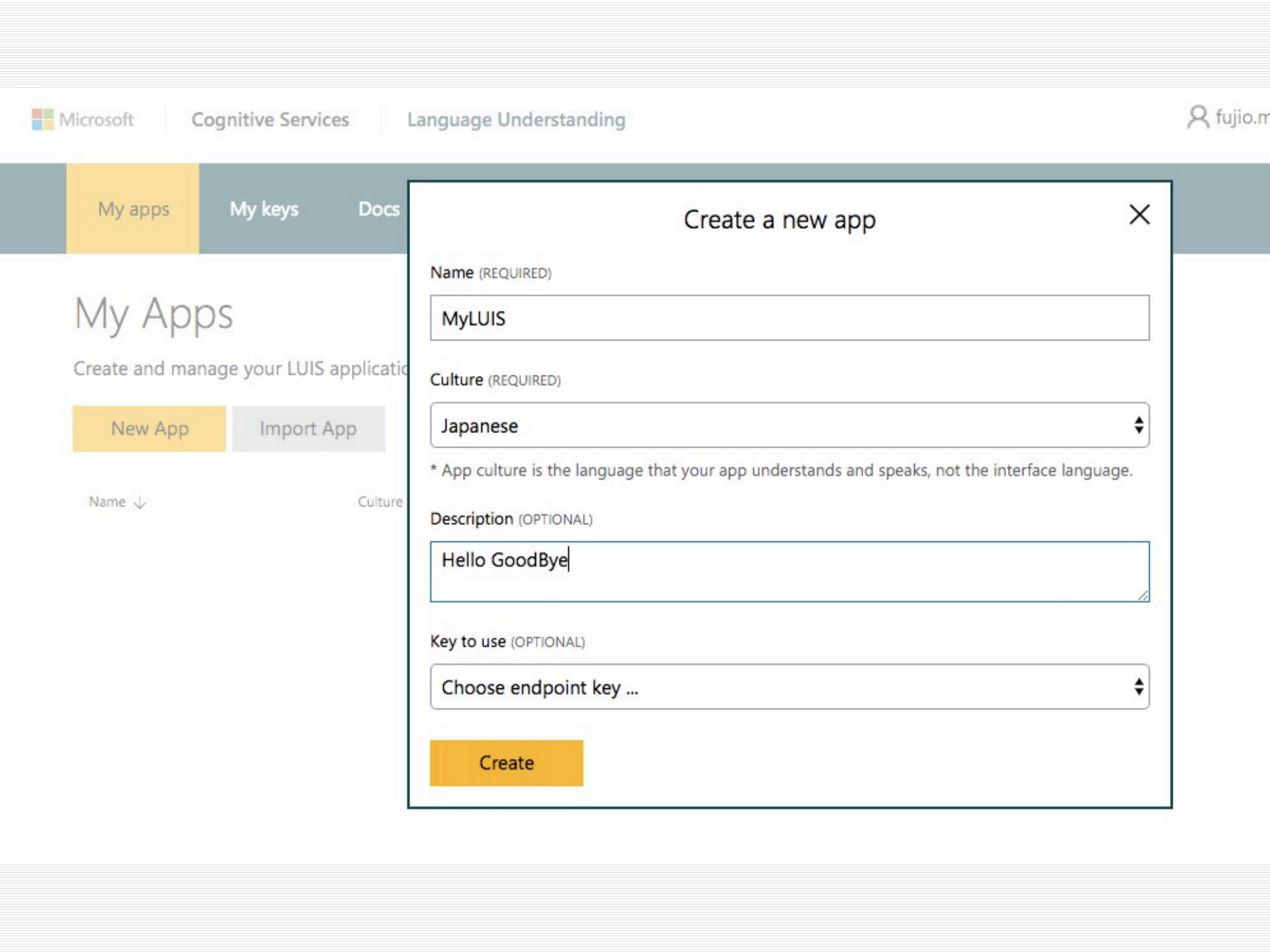


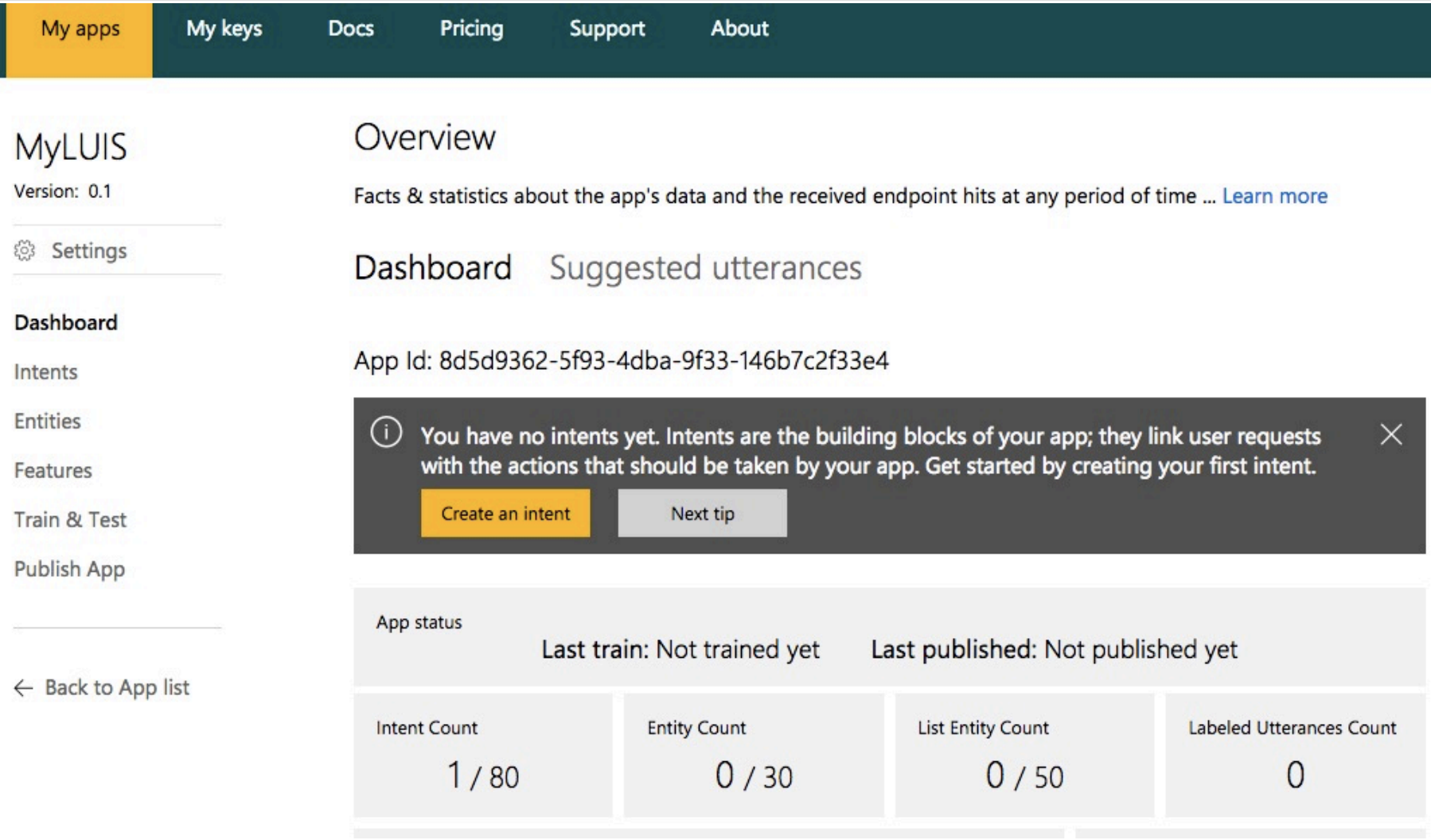
“Remind me to call dad tomorrow”


```
{
    "query": "remind me to call my da
    "topScoringIntent": {
        "intent": "Reminder",
        "score": 0.9337804
    },
    "intents": [
        {
            "intent": "Reminder",
            "score": 0.9337804
        },
        {
            "intent": "None",
            "score": 0.0900467858
        },
        {
            "intent": "LocationFinder
            "score": 0.009403142
        },
        {
            "intent": "BookFlight",
            "score": 0.002218235
        }
        {
            "intent": "FoodOrder",
            "score": 1.519075E-07
        }
    ],
    "entities": [
        {
            "entity": "tomorrow",
            "type": "builtin.datetime
            "startIndex": 25,
            "endIndex": 32,
            "resolution": {
                "date": "2017-05-05"
            }
        }
    ]
}
```

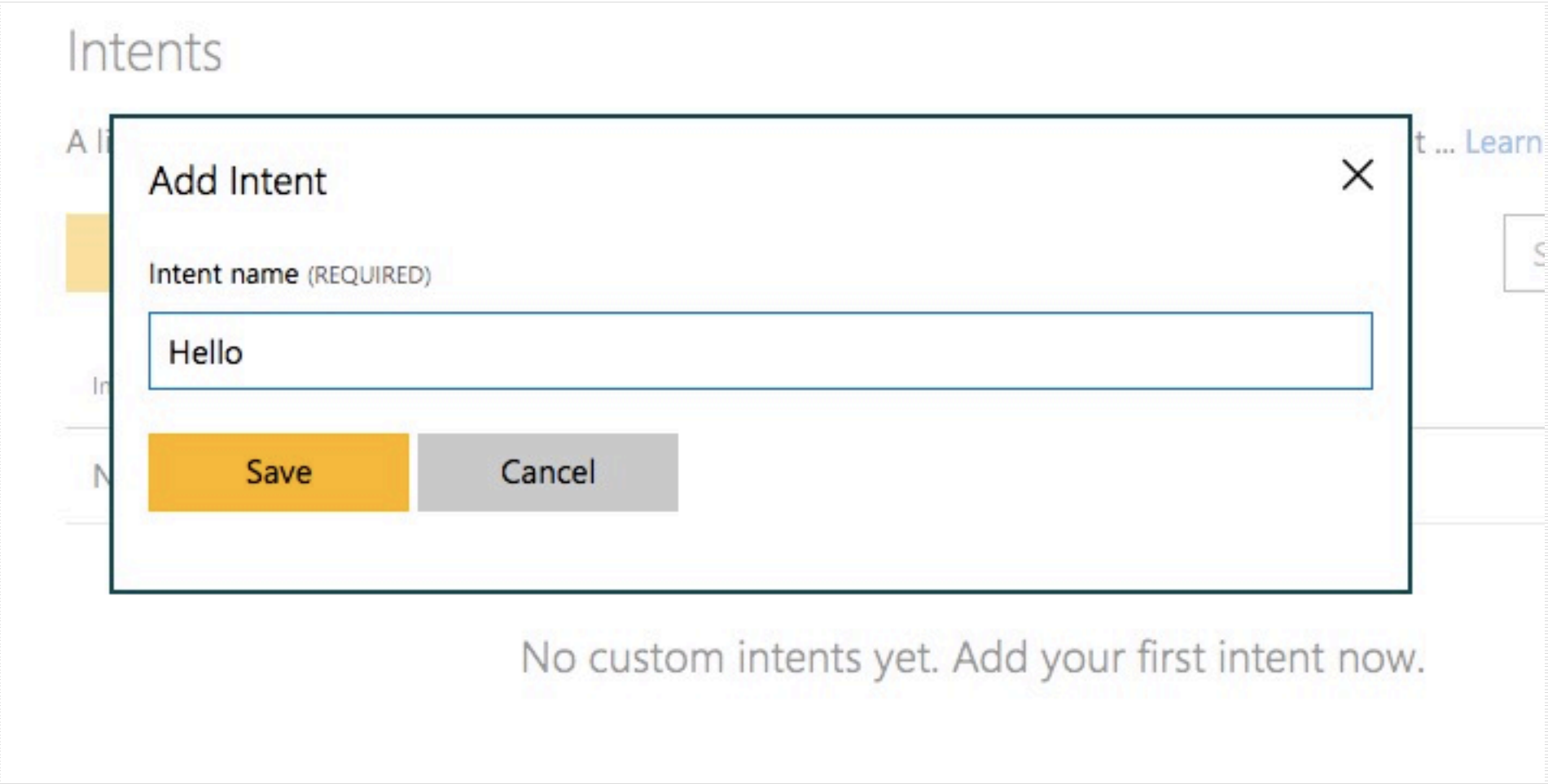
“Where is the nearest club”


```
{
  "query": "where is the nearest cl
  "topScoringIntent": {
    "intent": "LocationFinder",
    "score": 0.251945645
  },
  "intents": [
    {
      "intent": "LocationFinder
      "score": 0.251945645
    },
    {
      "intent": "None",
      "score": 0.08387748
    },
    {
      "intent": "BookFlight",
      "score": 0.00309361285
    },
    {
      "intent": "Reminder",
      "score": 0.002991894
    },
    {
      "intent": "Reminder",
      "score": 0.002991894
    },
    {
      "intent": "FoodOrder",
      "score": 1.4773135E-06
    },
  ],
  "entities": []
}
```

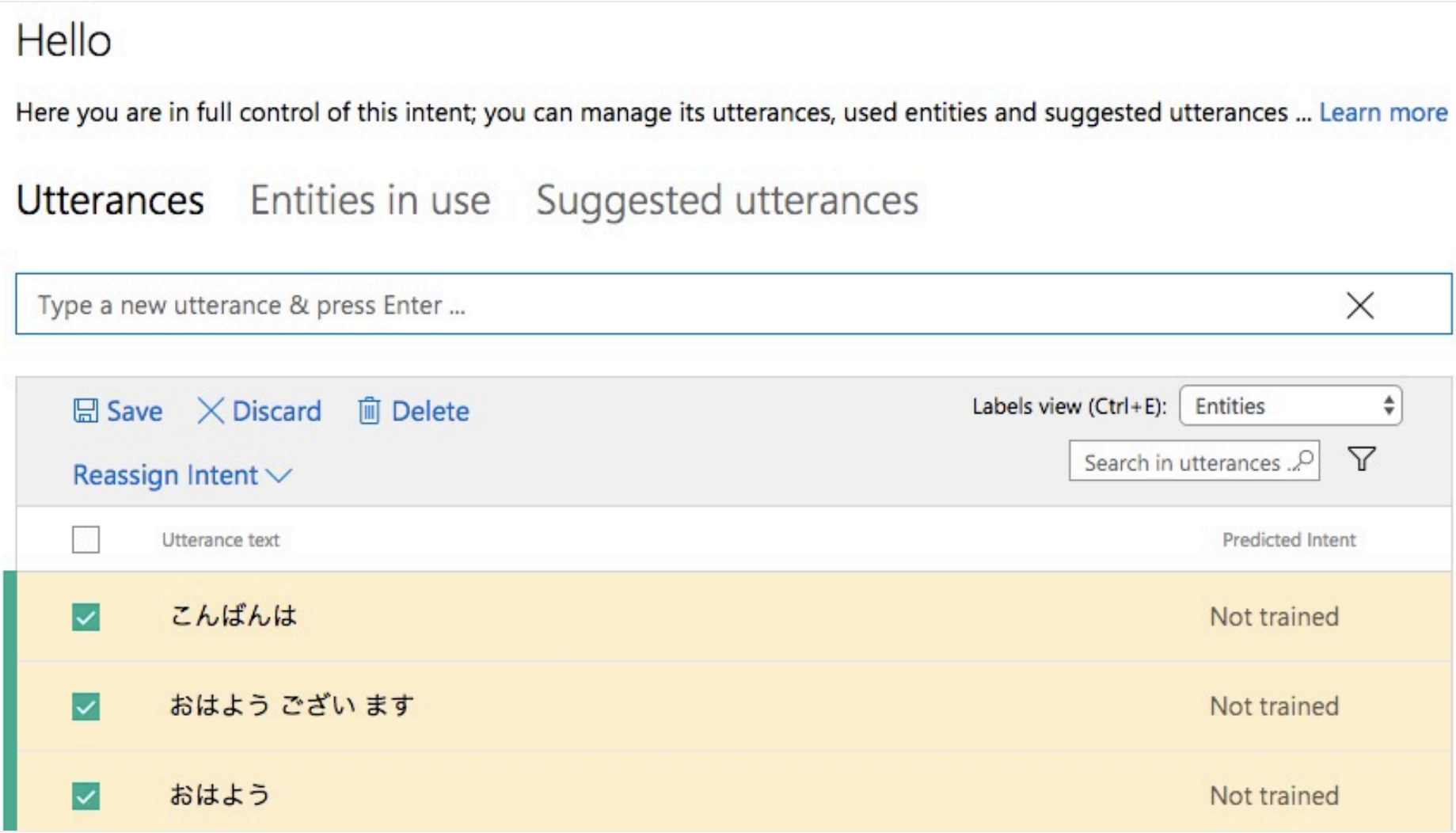

Microsoft
Cognitive Services
Language Understanding
fujio.m
My apps
My keys
Docs
My Apps
Create and manage your LUIS applicati
New App
Import App
Name
Culture
Create a new app
Name (REQUIRED)
MyLUIS
Culture (REQUIRED)
Japanese
* App culture is the language that your app understands and speaks, not the interface language.
Description (OPTIONAL)
Hello GoodBye
Key to use (OPTIONAL)
Choose endpoint key ...
Create


My apps
My keys
Docs
Pricing
Support
About
MyLUIS
Version: 0.1
Settings
Dashboard
Intents
Entities
Features
Train & Test
Publish App
Back to App list
Overview
Facts & statistics about the app's data and the received endpoint hits at any period of time ... Learn more
Dashboard
Suggested utterances
App Id: 8d5d9362-5f93-4dba-9f33-146b7c2f33e4
You have no intents yet. Intents are the building blocks of your app; they link user requests with the actions that should be taken by your app. Get started by creating your first intent.
Create an intent
Next tip
App status
Last train: Not trained yet
Last published: Not published yet
Intent Count
1 / 80
Entity Count
0 / 30
List Entity Count
0 / 50
Labeled Utterances Count
0

Hello Intentを作ってみる



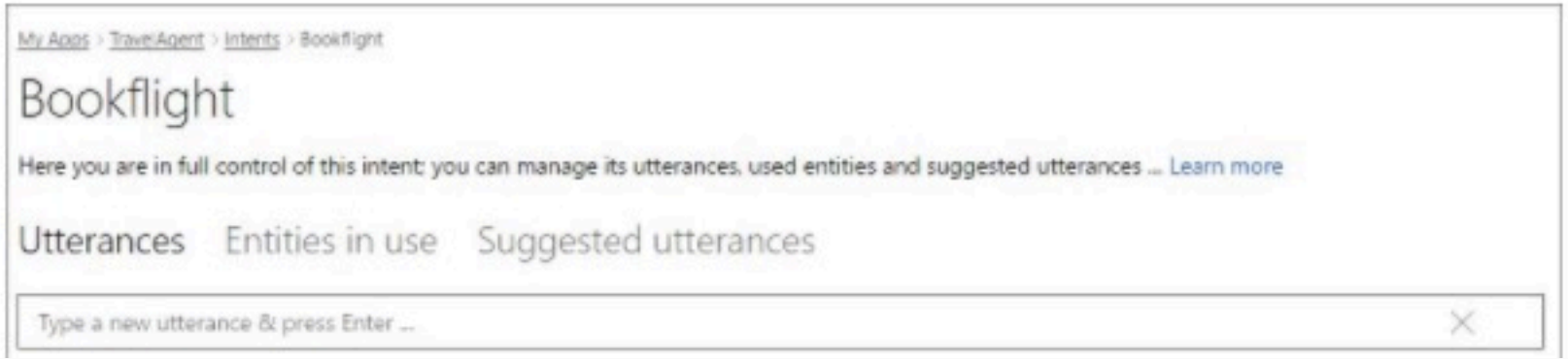
基本的には、Utteranceを追加する

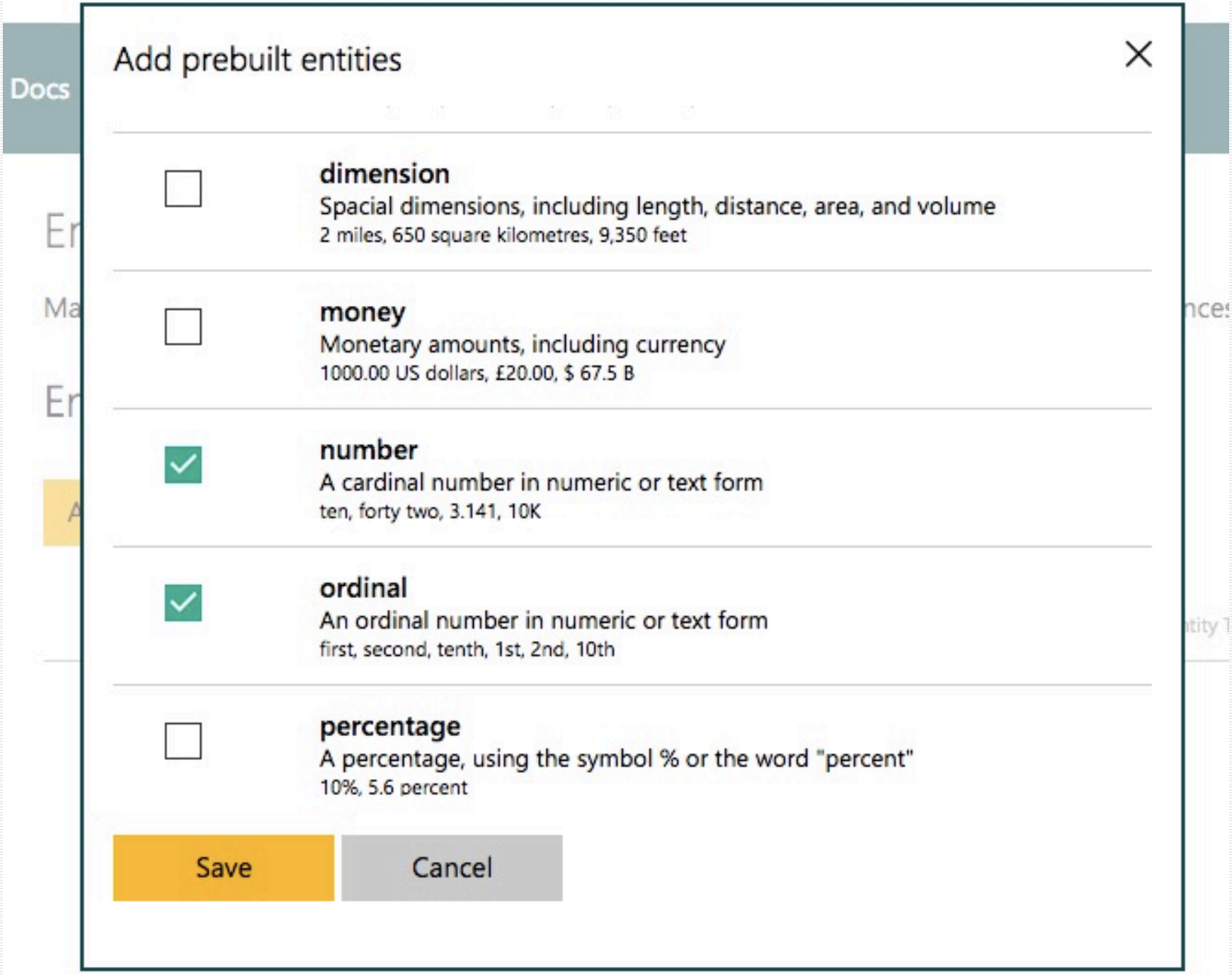


Entity label

There are different types of entities: custom entities (which include simple, hierarchical and composite entities) and prebuilt entities. You only need to label custom entities, **because prebuilt entities are detected and labeled automatically by your app.**

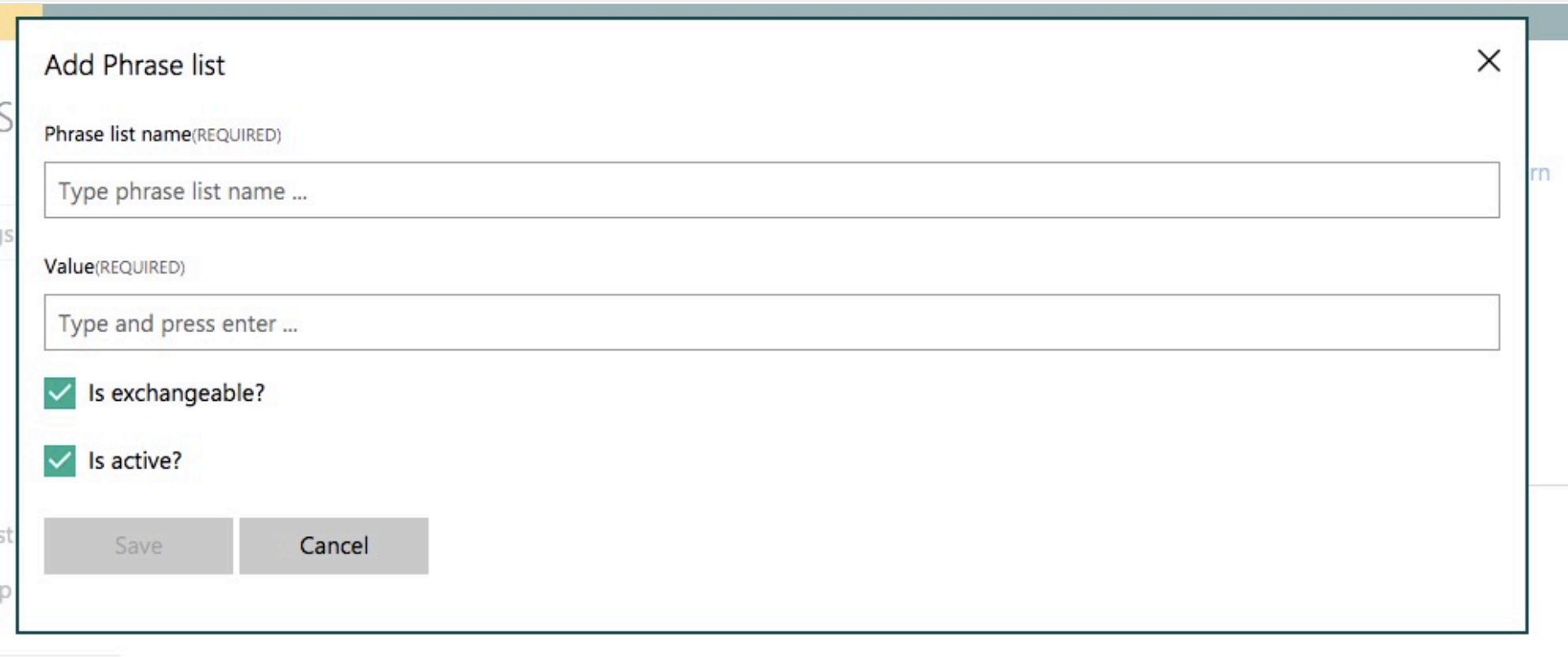
For example, look at the utterance `book me 2 adult business tickets to Paris tomorrow on Air France` that you've just added to the "Bookflight" intent in the TravelAgent app. Before you start labeling entities in this utterance, if you have already added `number` and `datetime` as prebuilt entities, you'll notice that "2" and "tomorrow" were automatically detected as prebuilt entities, where "2" is labeled as `number` and "tomorrow" as `datetime`. This looks like the following screenshot.

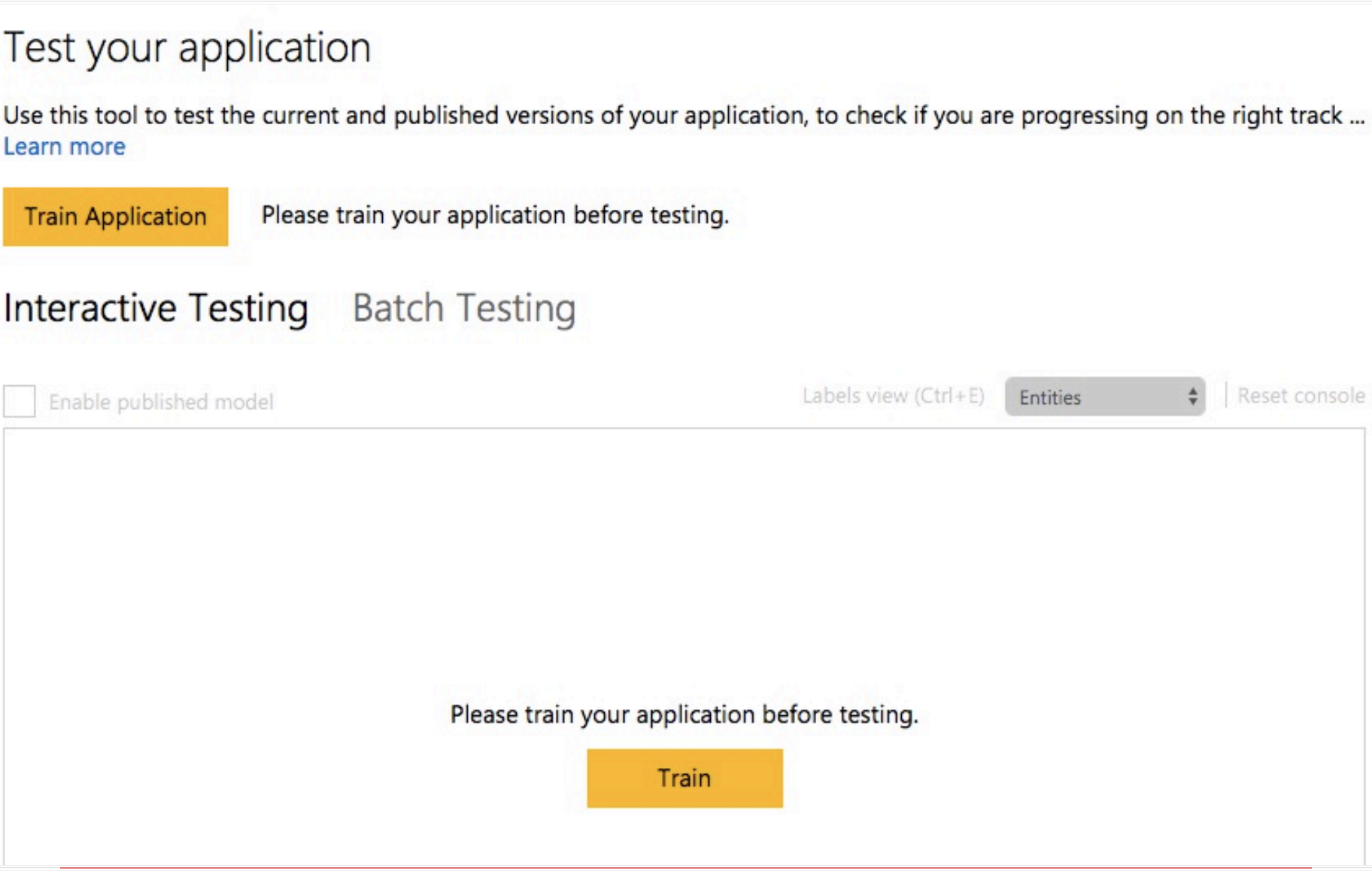


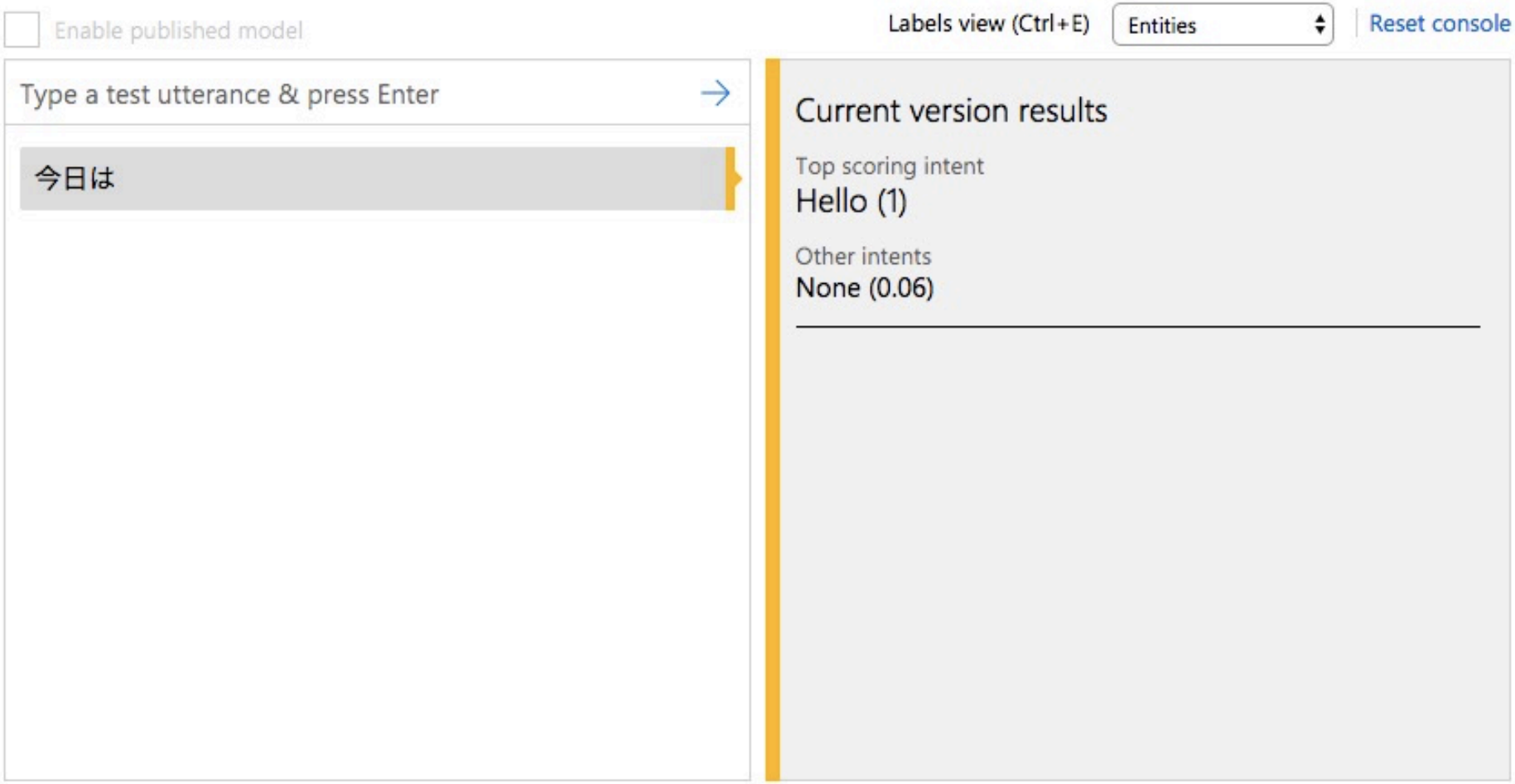

Docs
Add prebuilt entities
dimension
Spacial dimensions, including length, distance, area, and volume
2 miles, 650 square kilometres, 9,350 feet
money
Monetary amounts, including currency
1000.00 US dollars, £20.00, $ 67.5 B
number
A cardinal number in numeric or text form
ten, forty two, 3.141, 10K
ordinal
An ordinal number in numeric or text form
first, second, tenth, 1st, 2nd, 10th
percentage
A percentage, using the symbol % or the word "percent"
10%, 5.6 percent
Save
Cancel

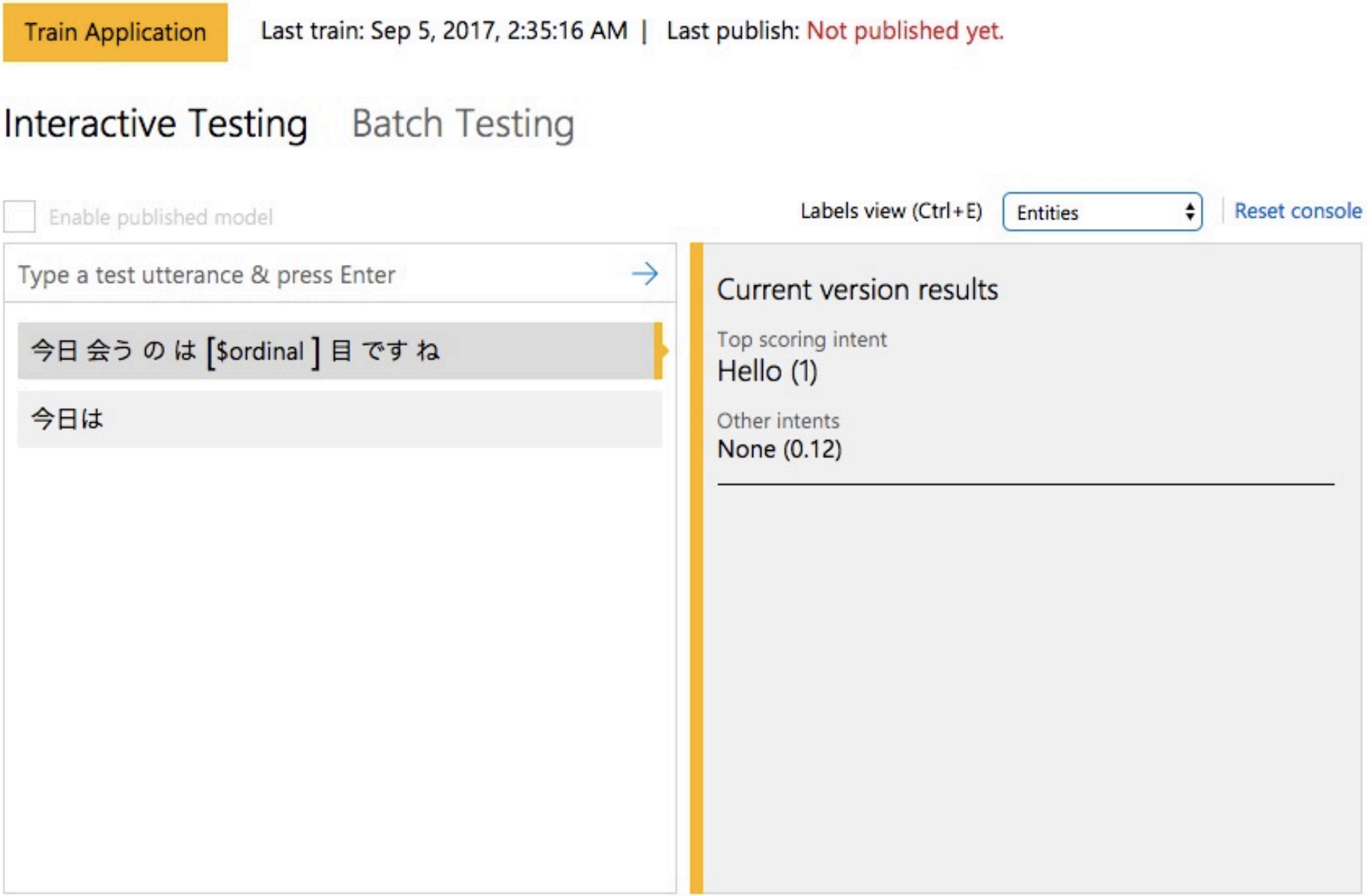

Utterances (5)
Entities in use
Suggested utterances
Type a new utterance & press Enter ...
Save
Discard
Delete
Reassign Intent
Labels view (Ctrl+E):
Entities
Search in utterances
Utterance text
Predicted Intent
こんばんは
Not trained
おはよう ござい ます
Not trained
おはよう
Not trained
今日は
Not trained
今日 会う の は [$ordinal] 目 です ね
Not trained

これは、AlexaのLIST TypeのSlotの定義と同じ
CORTANAのPhrase Listと同じもの




Test your application
Use this tool to test the current and published versions of your application, to check if you are progressing on the right track ...
Learn more
Train Application
Please train your application before testing.
Interactive Testing
Batch Testing
Enable published model
Labels view (Ctrl+E)
Entities
Reset console
Please train your application before testing.
Train


Enable published model
Labels view (Ctrl+E)
Entities
Reset console
Type a test utterance & press Enter
今日は
Current version results
Top scoring intent
Hello (1)
Other intents
None (0.06)


Train Application
Last train: Sep 5, 2017, 2:35:16 AM | Last publish: Not published yet.
Interactive Testing
Batch Testing
Enable published model
Labels view (Ctrl+E)
Entities
Reset console
Type a test utterance & press Enter
今日 会う の は [$ordinal] 目 です ね
今日は
Current version results
Top scoring intent
Hello (1)
Other intents
None (0.12)

Launching Built-in NLP for Messenger and Sunsetting Bot Engine (beta)

https://wit.ai/blog/2017/07/27/sunsetting-stories

2017/7/27 wit.ai / Facebook

Sunsetting Bot Engine

- By testing and learning from our Bot Engine beta, we determined that it made the most sense to refocus on pure NLP to make it accurate, reliable and scalable for everybody. As a result, **we're sunsetting Bot Engine and will deprecate the Stories UI for new apps starting today.** We will also stop serving requests to the associated /converse endpoint on February 1, 2018.

- **Using Wit.ai programmatically**
  https://github.com/wit-ai/wit-api-only-tutorial#send-responses

Messenger Platform **Built-in NLP**

文の中から、Entityを自動的に検出する

Built-in NLP

- Natural Language Processing (NLP) allows you to understand and extract meaningful information (called entities) out of the messages people send.
- You can then use these entities to identify intent, automate some of your replies, route the conversation to a human via livechat, and collect audience data.

How It works

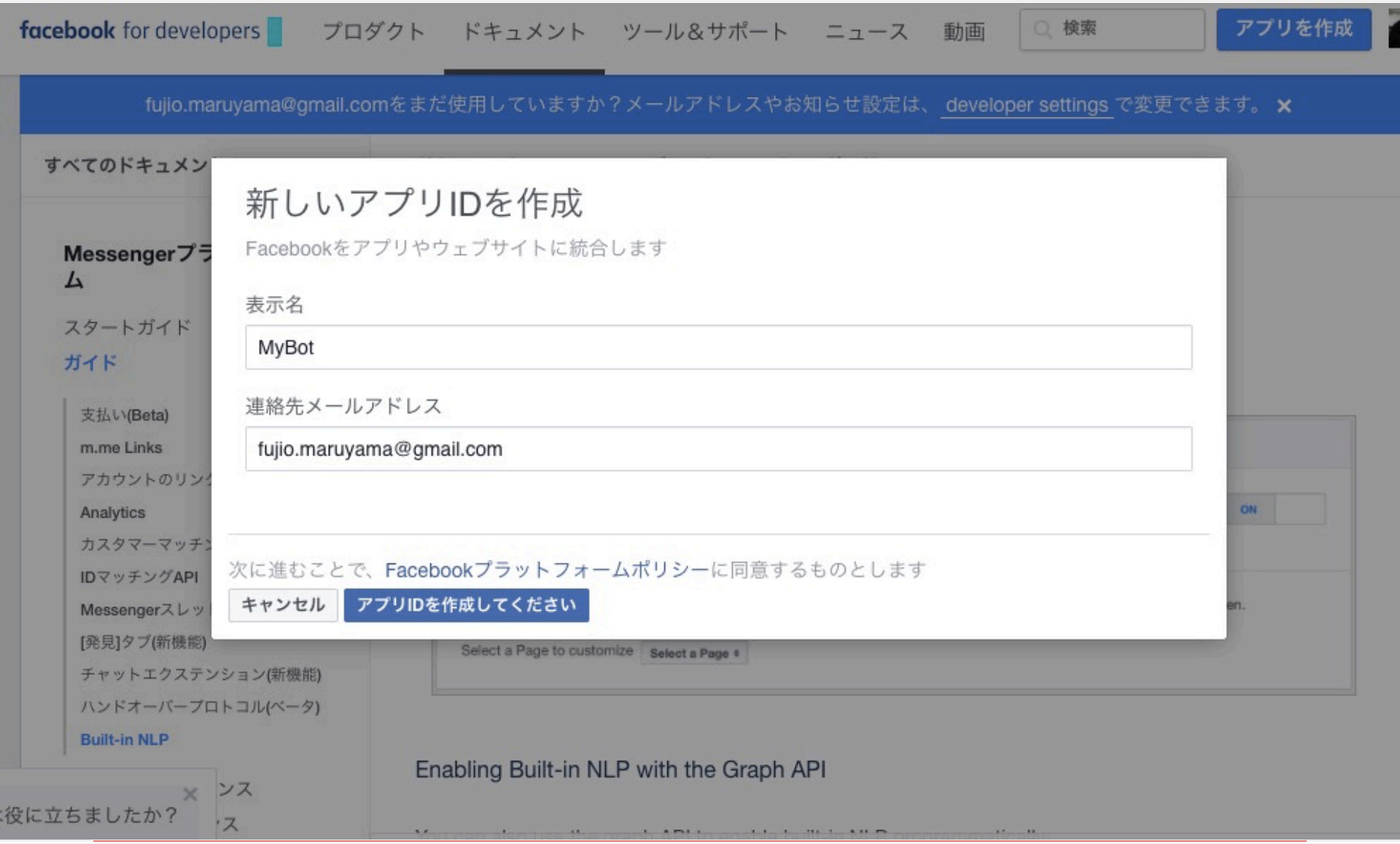
- Once Messenger's built-in NLP is enabled for your Facebook Page, it automatically detects meaning and intent in every text message before it is sent to your bot. The message will be relayed to your bot as usual, along with any entities detected in the body.

“bye, see you tomorrow at 4pm” に含まれるEntity


```
{...,
  entities: {
    "datetime": [
      {
        "confidence": 0.97249440664957,
        "values": ["..."],
        "value": "2017-05-10T14:00:00.000-07:00",
        "grain": "hour","type": "value"
        }
      ],
    "greetings": [
      {
        "confidence": 1,
        "value": "true"
      }
    ]
  }
}
```

Default NLP

- By default, Messenger's built-in NLP detects the following entities in English only:
  1. Greetings
  2. Thanks
  3. Bye
- It also detects information like date, time, location, amount of money, phone number and email. For example, if someone sends the message, “tomorrow at 2pm” or “2 days before Xmas,” you will get the actual timestamp with the message.


facebook for developers
プロダクト
ドキュメント
ツール＆サポート
ニュース
動画
検索
アプリを作成
fujio.maruyama@gmail.comをまだ使用していますか？メールアドレスやお知らせ設定は、 developer settings で変更できます。
すべてのドキュメン
新しいアプリIDを作成
Facebookをアプリやウェブサイトに統合します
表示名
MyBot
連絡先メールアドレス
fujio.maruyama@gmail.com
次に進むことで、Facebookプラットフォームポリシーに同意するものとします
キャンセル
アプリIDを作成してください
Messengerプラ
ム
スタートガイド
ガイド
支払い(Beta)
m.me Links
アカウントのリン
Analytics
カスタマーマッチン
IDマッチングAPI
Messengerスレッ
[発見]タブ(新機能)
チャットエクステンション(新機能)
ハンドオーバープロトコル(ベータ)
Built-in NLP
Select a Page to customize
Select a Page
ON
Enabling Built-in NLP with the Graph API
役に立ちましたか？


MyBot
アプリID: 1706663546018953
アナリティクスを見る
ツール&サポート
ドキュメント
ダッシュボード
設定
役割
アラート
アプリレビュー
プロダクト
Messenger
設定
+ 製品を追加
fujio.maruyama@gmail.comをまだ使用していますか？メールアドレスやお知らせ設定は、 developer settings で変更できます。
ビルトインNLP
受信したMessengerをパーシングすることで、自然言語処理が自動的に統合されます。ビルトインNLPにより、意味の検知や日時、ソーシャルプロトコルに関する情報を得ることができます。ビルトインNLPは現在英語でのみ利用可能です。 詳しくはこちら。
オフ
ページを既存のWit.aiアプリトークンにリンクすることで、NLPをカスタマイズして、より多くの言語や情報タイプを検知できるようにすることもできます。
ページとWit.aiをリンクすることで、次に同意するものとします: Facebookのデータに関するポリシー Messengerプラットフォームで送信されたコンテンツに適用
カスタマイズするページを選択してください
ページを選択
Messengerのアプリレビュー
Messengerプラットフォームを使用するには、アプリで送受信API(pages_messaging)の使用が許可されている必要があります。送受信APIはオーガニックコンテンツの送信のためのもので、マーケティングやその他の広告コミュニケーションの送信には使用できません。電話番号を使用して、既存の連絡先とMessengerアカウントを照合するには、カスタマーマッチング(pages_messaging_phone_number)が必要です。
pages_messaging
申請に追加
facebook for developers
https://developers.facebook.com/apps/1706663546018953/messenger/#

MyBotのテストアプリを作成

テストアプリでプロダクションアプリに影響を与えることなく開発の変更をデバッグ・テストすることができます。 [?]

テストアプリ名

MyBot - Test1

キャンセル　テストアプリを作成

ダッシュボード

MyBot - Test1 ○

このアプリは開発モードになっているため、使用できるのはアプリの管理者、開発者、テスターのみです [?]

APIバージョン [?]　v2.10

アプリID　170153913543017

app secret　●●●●●●●●　表示

Google Assistantのプログラミング

Google Assistant SDK

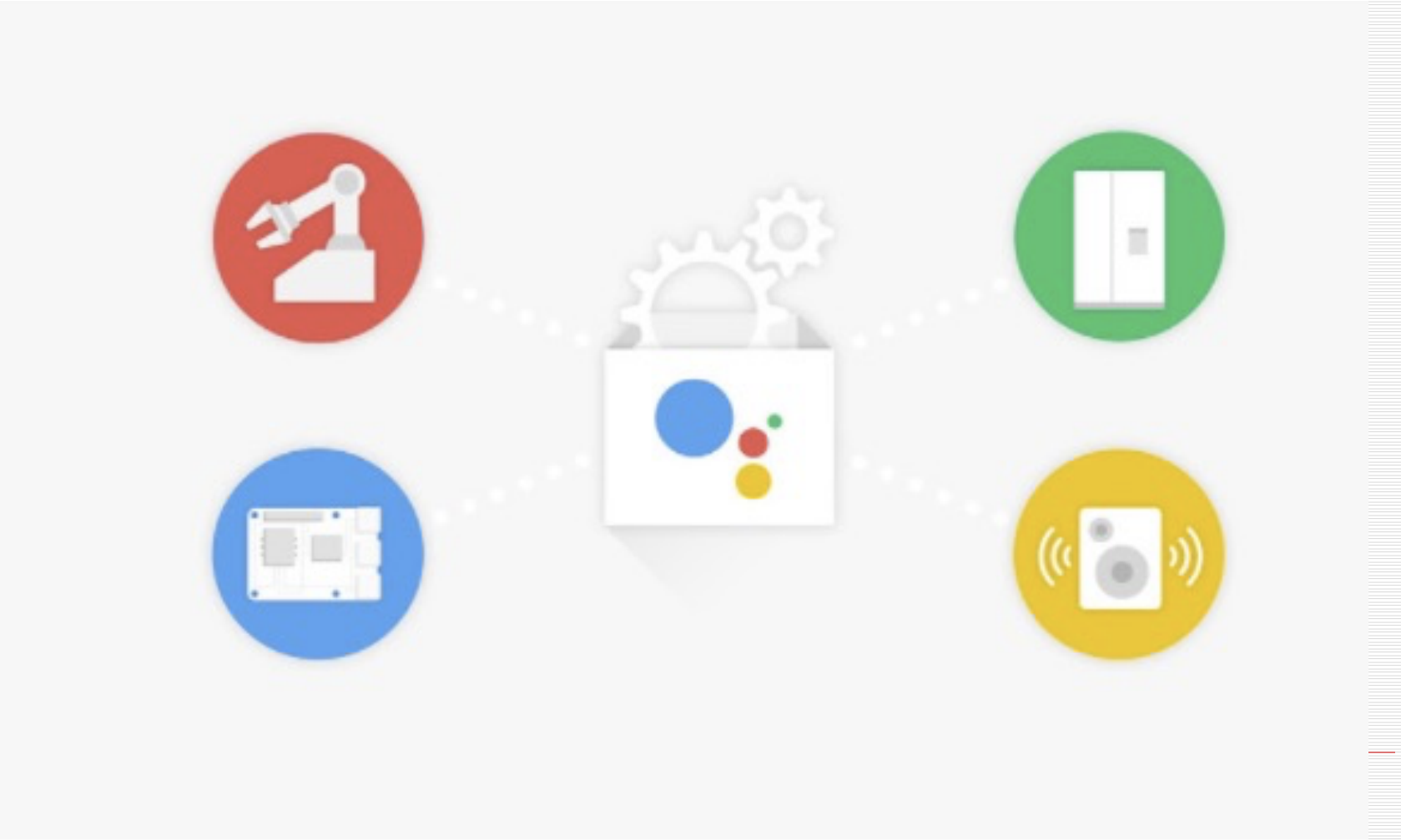
The developer preview (0.0.3) **2017/7/19**

https://developers.google.com/assistant/sdk/

- Google Assistant library for Python
- Google Assistant gRPC API

ここから、開発者のForumに入れる
https://groups.google.com/forum/#!forum/google-assistant-sdk-updates

Embed the Google Assistant in any device



Google Assistant library for Python

- The Google Assistant library for Python is a turnkey solution for anyone who wants to quickly integrate the Assistant into a prototype device. The library is written in Python and is supported on devices with linux-armv7l and linux-x86_64 architectures (like the Raspberry Pi 3 B and Ubuntu desktops).

- Hands-free activation - activate your device by saying *Hey Google* or *Ok Google*, just like with Google Home!
- Audio capture and playback
- Conversation state management
- Timer and alarm management

Google Assistant gRPC API

- The Google Assistant gRPC API is the best option for flexibility and broad platform support. It exposes a low level API which directly manipulates the audio bytes of an Assistant request and response. Bindings for this API can be generated for languages like Node.js, Go, C++, Java for all platforms that support gRPC.

	Library	gRPC API
Supported architectures	linux-armv7l and linux-x86_64	gRPC platforms
Supported languages	Python	gRPC languages
Hands-free activation (*Ok Google*)	Yes	No
Audio capture and playback	Built in	Reference code is provided
Conversation state management	Built in	Reference code is provided
Timers and alarms	Yes	No

git clone https://github.com/googlesamples/assistant-sdk-python

Google Assistant SDK for Python

This repository contains the Python packages for the Google Assistant SDK.

To get started, see the following developer guides:

- Getting Started with Raspberry Pi 3 and Python.
- Getting Started with other platforms.

Content

- google-assist
- goo
- goo

```
with Assistant(credentials) as assistant:
    for event in assistant.start():
        process_event(event)
```

PC API
ant SDK

Contri

Contributions and highly encouraged.

See CONTRIB for more information on how to get started.

License

Copyright (C) 2017 Google Inc.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Python Samples for the Google Assistant gRPC API

This repository contains a reference sample for the `google-assistant-grpc` Python package_.

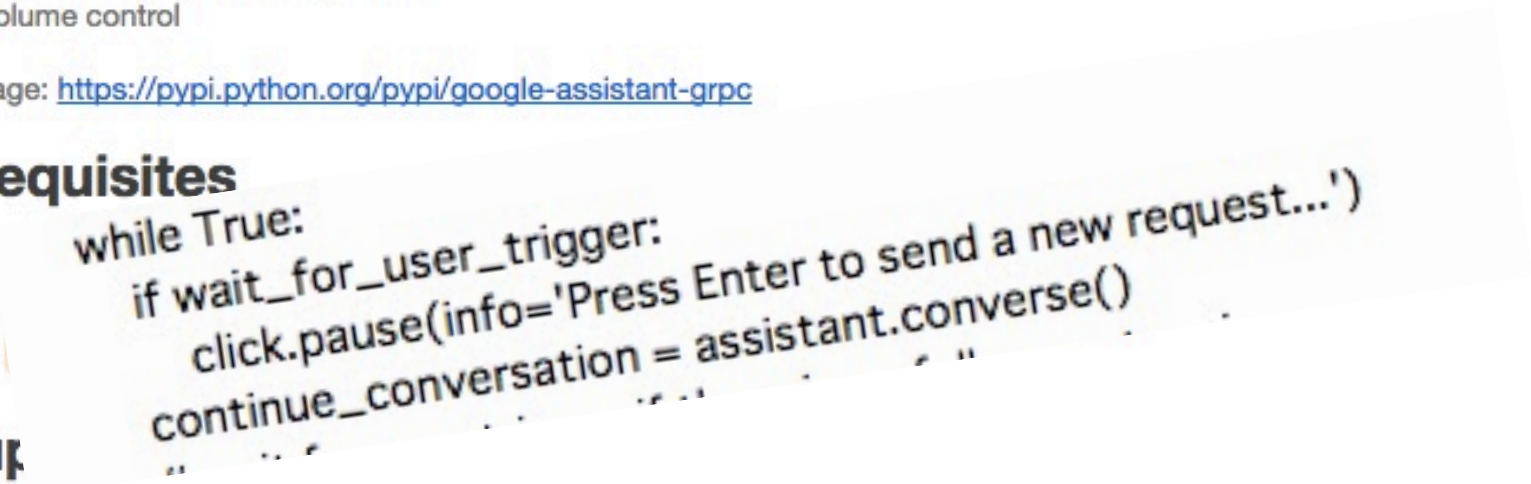
It implements the following features:

- Triggering a conversation using a key press
- Audio recording of user queries (single or multiple consecutive queries)
- Playback of the Assistant response
- Conversation state management
- Volume control

.. _package: https://pypi.python.org/pypi/google-assistant-grpc

Prerequisites

- P
- A
- A



Setup

- Install Python 3
  - Ubuntu/Debian GNU/Linux::

    sudo apt-get update sudo apt-get install python3 python3-venv
  - `MacOSX, Windows, Other <https://www.python.org/downloads/>`_
- Create a new virtual environment (recommended)::

  python3 -m venv env env/bin/python -m pip install --upgrade pip setuptools source env/bin/activate

EmbeddedAssistant

Service that implements Google Assistant API.

Converse

```
rpc Converse(ConverseRequest) returns (ConverseResponse)
```

Initiates or continues a conversation with the embedded assistant service. Each call performs one round-trip, sending an audio request to the service and receiving the audio response. Uses bidirectional streaming to receive results, such as the **END_OF_UTTERANCE** event, while sending audio.

Request

Add to my shopping list

Responsee

What do you want to add?.

A conversation is one or more gRPC connections, each consisting of several streamed requests and responses. For example, the user says *Add to my shopping list* and the assistant responds *What do you want to add?*. The sequence of streamed requests and responses in the first gRPC message could be:

- ConverseRequest.config
- ConverseRequest.audio_in
- ConverseRequest.audio_in
- ConverseRequest.audio_in
- ConverseRequest.audio_in

"Add to my shopping list"

- ConverseResponse.event_type.END_OF_UTTERANCE
- ConverseResponse.result.microphone_mode.DIALOG_FOLLOW_ON
- ConverseResponse.audio_out
- ConverseResponse.audio_out
- ConverseResponse.audio_out

"What do you want to add?."

Request

bagels

Responsee

OK, I've added bagels to your shopping list.

The user then says *bagels* and the assistant responds *OK, I've added bagels to your shopping list*. This is sent as another gRPC connection call to the **`Converse`** method, again with streamed requests and responses, such as:

- ConverseRequest.config
- ConverseRequest.audio_in
- ConverseRequest.audio_in
- ConverseRequest.audio_in

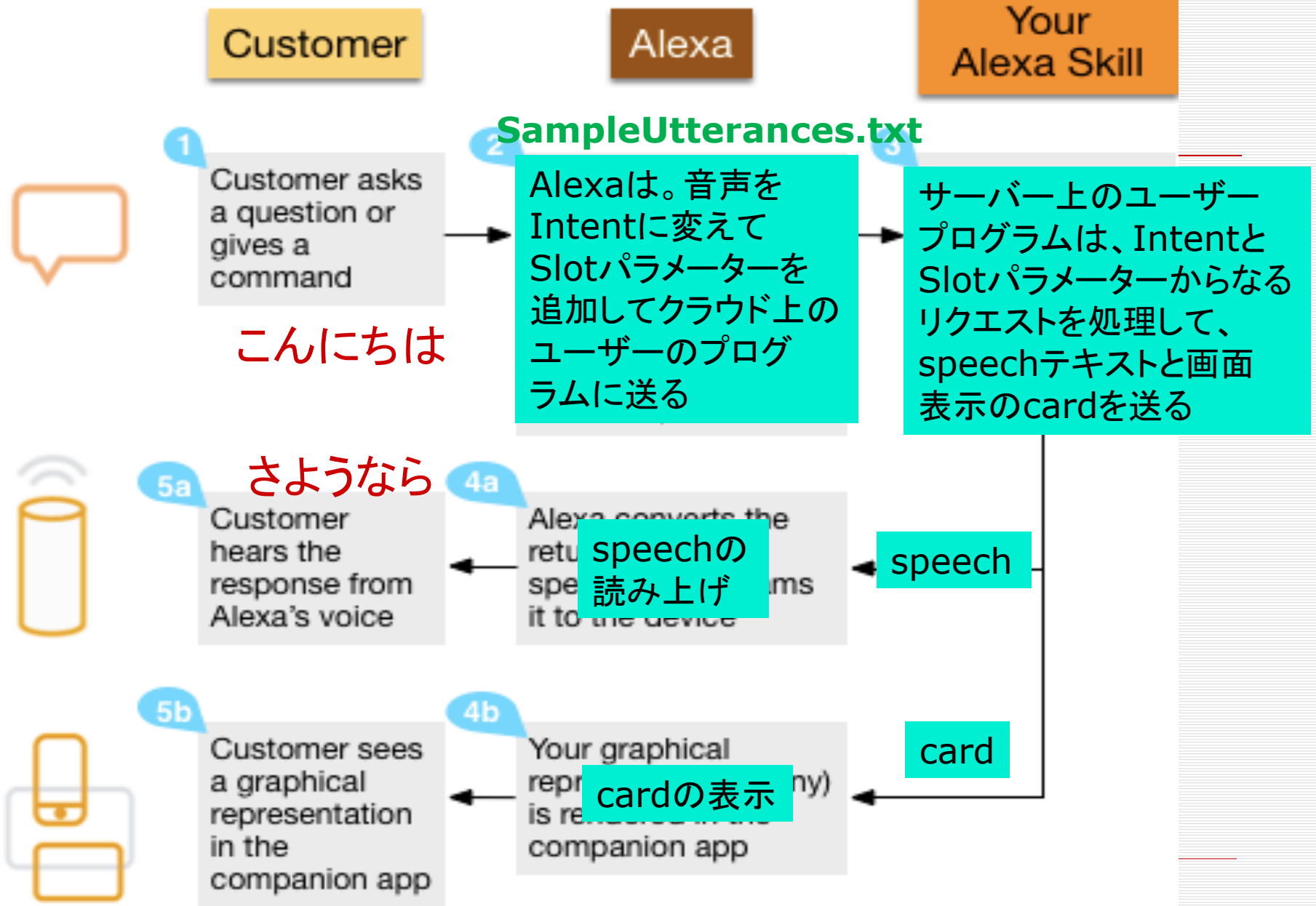
bagels

- ConverseResponse.event_type.END_OF_UTTERANCE
- ConverseResponse.result.microphone_mode.CLOSE_MICROPHONE
- ConverseResponse.audio_out
- ConverseResponse.audio_out
- ConverseResponse.audio_out
- ConverseResponse.audio_out

OK, I've added bagels to your shopping list.

Although the precise order of responses is not guaranteed, sequential ConverseResponse.audio_out messages will always contain sequential portions of audio.

ユーザーのプログラム


Customer
Alexa
Your Alexa Skill
SampleUtterances.txt
1
Customer asks a question or gives a command
こんにちは
2
Alexaは。音声をIntentに変えてSlotパラメーターを追加してクラウド上のユーザーのプログラムに送る
3
サーバー上のユーザープログラムは、IntentとSlotパラメーターからなるリクエストを処理して、speechテキストと画面表示のcardを送る
speech
4a
speechの読み上げ
5a
Customer hears the response from Alexa's voice
さようなら
card
4b
cardの表示
5b
Customer sees a graphical representation in the companion app

ConverseRequest

- The top-level message sent by the client. Clients must send at least two, and typically numerous ConverseRequest messages.
- The first message must contain a config message and must not contain audio_in data. All subsequent messages must contain audio_in data and must not contain a config message.

ConverseRequest

Fields	
Union field **converse_request**. Exactly one of these fields must be specified in each **ConverseRequest**. **converse_request** can be only one of the following:	
config	**ConverseConfig** The **config** message provides information to the recognizer that specifies how to process the request. The first **ConverseRequest** message must contain a **config** message.
audio_in	**bytes** The audio data to be recognized. Sequential chunks of audio data are sent in sequential **ConverseRequest** messages. The first **ConverseRequest** message must not contain **audio_in** data and all subsequent **ConverseRequest** messages must contain **audio_in** data. The audio bytes must be encoded as specified in **AudioInConfig**. Audio must be sent at approximately real-time (16000 samples per second). An error will be returned if audio is sent significantly faster or slower.

ConverseResponse

Fields	
Union field **converse_response**. Exactly one of these fields will be populated in each **ConverseResponse**. **converse_response** can be only one of the following:	
error	`Status` *Output-only* If set, returns a `google.rpc.Status` message that specifies the error for the operation. If an error occurs during processing, this message will be set and there will be no further messages sent.
event_type	`EventType` *Output-only* Indicates the type of event.
audio_out	`AudioOut` *Output-only* The audio containing the assistant's response to the query.
result	`ConverseResult` *Output-only* The semantic result for the user's spoken query.

google.assistant.library.event.EventType

```
ON_CONVERSATION_TURN_STARTED
ON_END_OF_UTTERANCE
ON_RECOGNIZING_SPEECH_FINISHED
{'text': 'what time is it'}
ON_RESPONDING_STARTED:
{'is_error_response': False}
ON_RESPONDING_FINISHED
ON_CONVERSATION_TURN_FINISHED
{'with_follow_on_turn': False}
```

rpc_method_handlers

```
rpc_method_handlers = {
      'Converse':
grpc.stream_stream_rpc_method_handler(
          servicer.Converse,
          request_deserializer=
google_dot_assistant_dot_embedded_dot_v1alpha1_
dot_embedded__assistant__pb2.ConverseRequest.Fr
omString,
          response_serializer=
google_dot_assistant_dot_embedded_dot_v1alpha1_
dot_embedded__assistant__pb2.ConverseResponse.S
erializeToString,
    ),
}
```

https://goo.gl/QhstFa

Introducing Dialogflow, the new name for API.AI

2017/10/10

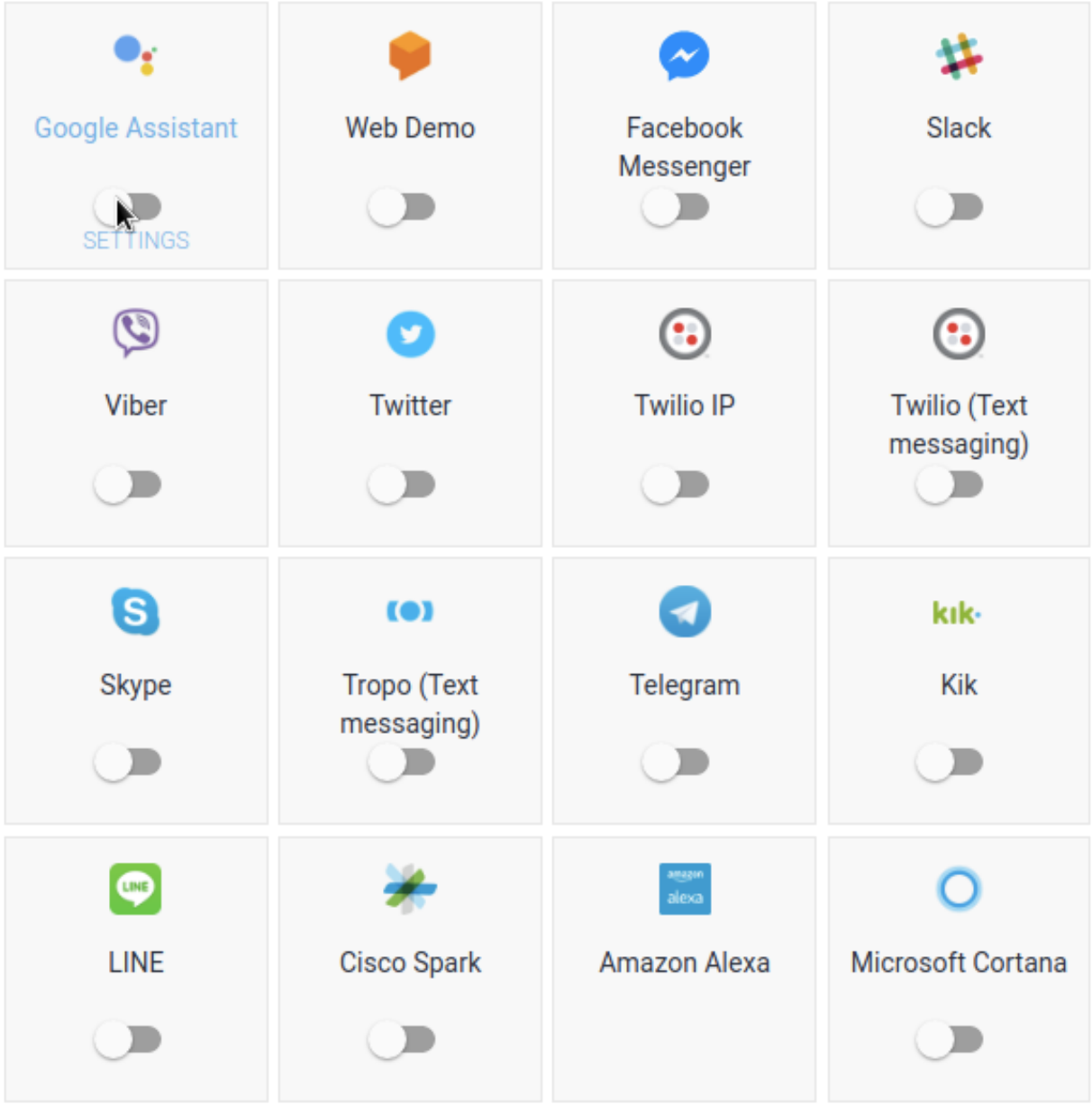
https://blog.dialogflow.com/post/apiai-new-name-dialogflow-new-features/

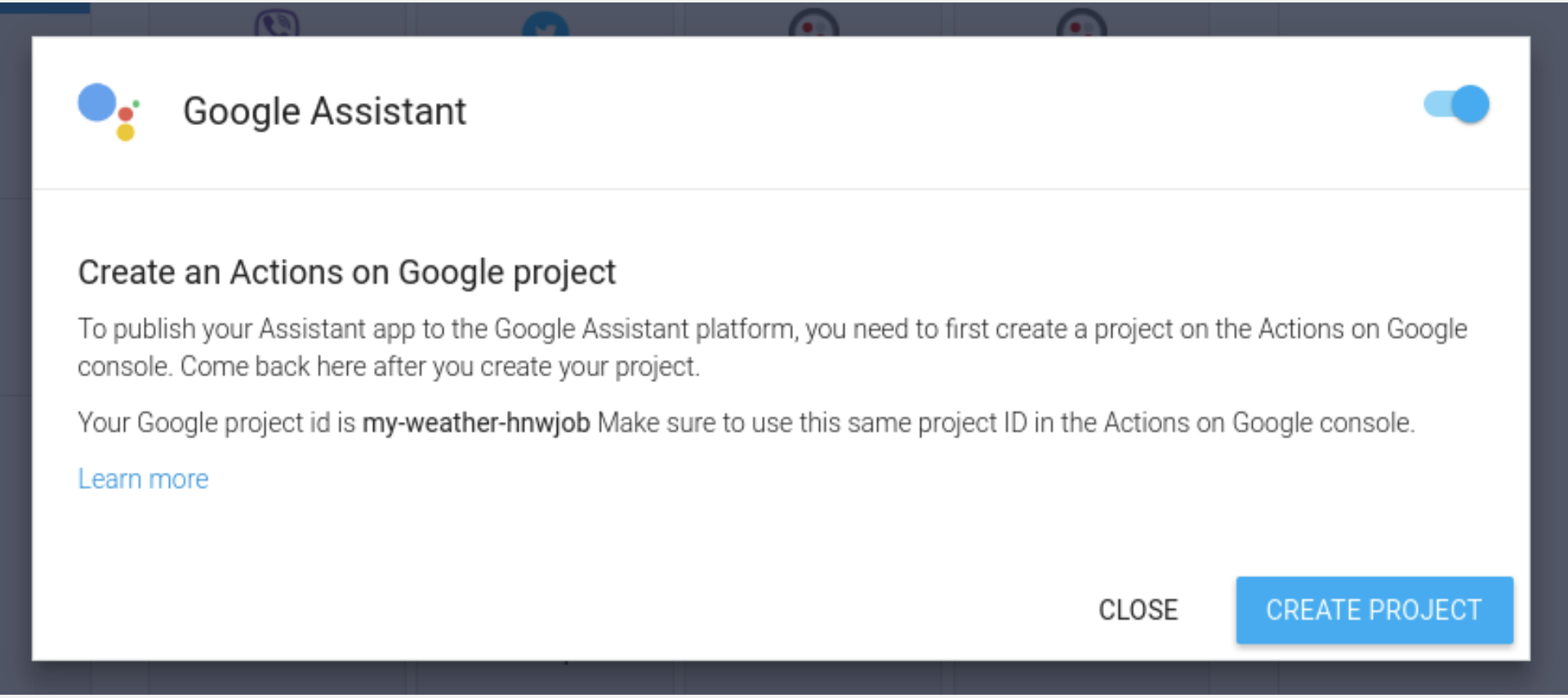


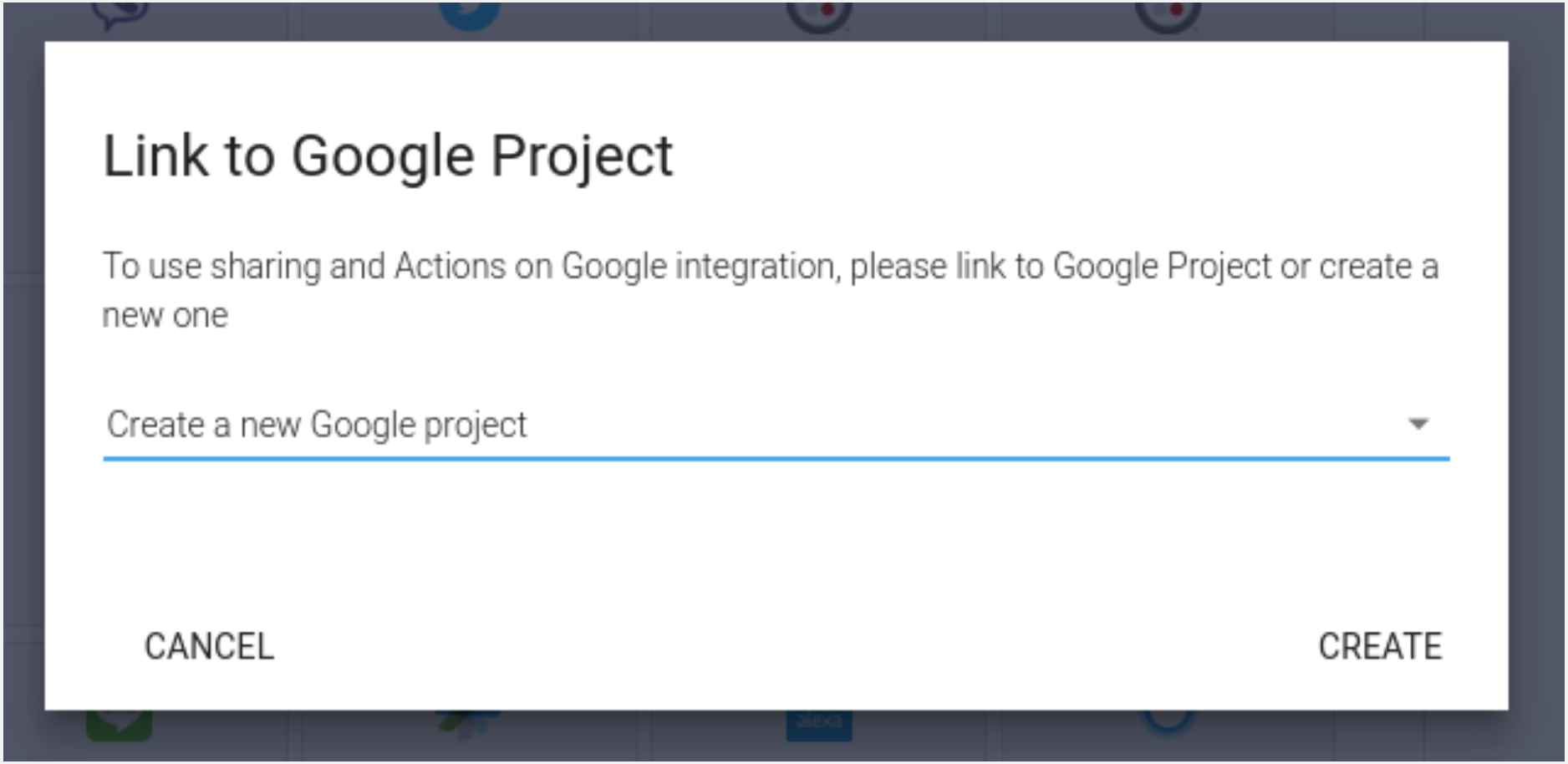


- **In-line code editor:** you can now write fulfillment logic, test, and implement a functional webhook directly in the console.
- **Multi-lingual agent support:** building for multiple languages is now easier than ever. You can now add additional languages and locales to your new or existing agent.

One-click integrations




Google Assistant
Create an Actions on Google project
To publish your Assistant app to the Google Assistant platform, you need to first create a project on the Actions on Google console. Come back here after you create your project.
Your Google project id is my-weather-hnwjob Make sure to use this same project ID in the Actions on Google console.
Learn more
CLOSE
CREATE PROJECT


Link to Google Project
To use sharing and Actions on Google integration, please link to Google Project or create a new one
Create a new Google project
CANCEL
CREATE

Fulfillmentを構成する

- Define your conversation
  - Create intents
- Building responses in Dialogflow
- Building fulfillment responses

Define your conversation
Intentsの生成



Actionは、どのデータがFulfillmentに送られるかを定義する。
このデータには、ユーザーの入力からパーズされたデータや、Intentの名前などが含まれている。



Define your Response



Build Fulfillment



Fulfillmentのコード・サンプル

```
const app = new DialogflowApp({request: request, response: response});

const WELCOME_INTENT = 'input.welcome';  // the action name from the Dialogflow intent
const NUMBER_INTENT = 'input.number';  // the action name from the Dialogflow intent
const NUMBER_ARGUMENT = 'input.mynum'; // the action name from the Dialogflow intent

function welcomeIntent (app) {
  app.ask('Welcome to number echo! Say a number.');
}

function numberIntent (app) {
  let number = app.getArgument(NUMBER_ARGUMENT);
  app.tell('You said ' + number);
}
```

Fulfillmentのコード・サンプル

```
function responseHandler (app) {
  // intent contains the name of the intent you defined in the Actions area of Dialogflow
  let intent = app.getIntent();
  switch (intent) {
    case WELCOME_INTENT:
      app.ask('Welcome! Say a number.');
      break;

    case NUMBER_INTENT:
      let number = app.getArgument(NUMBER_ARGUMENT);
      app.tell('You said ' + number);
      break;
  }
}
```

音声インターフェース時代の
ユーザー・エクスペリエンスを考える



ボイス・インターフェースの
UX ユーザー・エクスペリエンス

ボイス・アシスタント・システムを構築する際の一つのポイントは、ボイス・インターフェースにも、UX ユーザー・エクスペリエンスが存在するという観点を、きちんと持つことだと思う。

画面のデザインだけが、UI/UXである時代ではなくなるのだ。よりよいUXに対するニーズは、ボイス・インターフェースでも、基本的なものだ。新しい、知見と経験が生まれることを期待している。

先に「汎用の対話」は、苦手であることは述べたのだが、「対話」の形式はボイス・インターフェースの重要なパーツである。

まずは、必要とされる対話の形式を見ておこう。

Multi-turn dialog
Dialog Interface Reference

https://goo.gl/iDbeVW

A multi-turn dialog is a conversation between the user and Alexa to gather the information needed to fulfill the user's request.

Multi-turn ダイアログ

- Alexaのスキルのスロットに、”required“（必須）のフラグを立てておくと、もし、ユーザーの発話の中に、それらのSlotについての情報が欠けていれば、Alexaは、自動的にユーザーとの対話を繰り返して、Slotを埋めようとする。
- 考えは単純だが、音声インターフェースとしては、必須の機能と思われる。

対話のサンプル

必須Slotの{fromCity}, {toCity}, {travelDate}を埋め、確認し、最後に intent 全体を確認する対話

Alexa: What city are you leaving from?
(Eliciting the value for a fromCity slot.)

User: Seatle

Alexa: You said you're leaving from *Seattle*, right?
(Confirming the fromCity value.)

User: Yes

Alexa: Where are you traveling to?
(Eliciting the value for a toCity slot.)

User: Portland

Alexa: Did you want to travel to *Portland*?
(Confirming the toCity value.)

User: Yes

対話のサンプル

必須Slotの{fromCity}, {toCity}, {travelDate}を埋め、確認し、最後に intent 全体を確認する対話

Alexa: When did you want to travel?

(Eliciting the value for a travelDate slot.)

User: on April 21st

Alexa: You're traveling on *April 21st*, right?

(Confirming the travelDate value.)

User: right

Alexa: I'm saving your trip
from *Seattle* to *Portland* on *April 21st*.
Is that OK?

(Confirming the entire PlanMyTrip intent.)

User: OK

Dialog ModelとDialog Directive

To use any of the Dialog directives, you must create a dialog model with the skill builder (beta). This structure identifies:

1. The slots that *must* be filled with valid values in order to fulfill the intent. These are considered *required slots*.
2. The prompts Alexa speaks to ask for required slot values and the utterances users can say in reply.
3. Whether any of the required slots must be also confirmed by the user before continuing.
4. Whether the entire intent must be confirmed by the user before continuing.
5. The prompts Alexa speaks to ask for slot and intent confirmations.

ユーザとの会話を管理する三つの方法

1. Let Alexa use the prompts defined in the ***dialog model*** by returning the Dialog.Delegate directive.
2. Control each step of the dialog yourself using Dialog.ElicitSlot, Dialog.ConfirmSlot, and Dialog.ConfirmIntent.
3. Combine both options – delegate on some turns, but take control on others when necessary.

Delegate Directive

Sends Alexa a command to handle the next turn in the dialog with the user. You can use this directive if the skill has a *dialog model* and the current status of the dialog (dialogState) is either STARTED or IN_PROGRESS. You cannot return this directive if the dialogState is COMPLETED.

```
{
  "type": "Dialog.Delegate",
  "updatedIntent": {
    "name": "string",
    "confirmationStatus": "NONE",
    "slots": {
      "string": {
        "name": "string",
        "value": "string",
        "confirmationStatus": "NONE"
      }
    }
  }
}
```

Delegate Directive

1. Check the dialog status with the dialogState property on the IntentRequest.
2. If the dialog is just starting (dialogState is STARTED), update the slots on the Intent object with default values as needed. Include this updated intent in the updatedIntent property on the Dialog.Delegate directive.
3. If the dialog is IN_PROGRESS, return Dialog.Delegate with no updatedIntent.
4. Once the dialog is COMPLETED, do your normal intent handling.

```
// The intentRequest variable
// here is the IntentRequest object sent to the skill.
if (intentRequest.dialogState === "STARTED"){
  // Pre-fill slots: update the intent object with slot values for which
  // you have defaults, then return Dialog.Delegate
  // with this updated intent in the updatedIntent property.
} else if (intentRequest.dialogState != "COMPLETED"){
  // return a Dialog.Delegate directive with no updatedIntent property.
} else {
  // Dialog is now complete and all required slots should be filled,
  // so call your normal intent handler.
  handlePlanMyTripIntent(intent, session, callback);
}
```

サンプル

- **User: Alexa,「旅行計画」で、稚内行きの計画作って**
  - *Alexa sends the skill a PlanMyTrip intent with:*
    dialogState: STARTED
    fromCity: null
    toCity: 稚内
    travelDate: null
    travelMode: null
    activity: null
    confirmationStatus: NONE
  - *The skill returns a Dialog.Delegate directive with: updatedIntent.slots.fromCity: Seattle and the other slots unchanged (this defaults the fromCity to a value)*

- **Plan My Trip: 東京からの出発でよろしいですか？**
  - *(Confirmation prompt for the fromCity slot, as defined in the dialog model.)*
- **User: そう**
  - *Alexa sends the PlanMyTrip intent with:*
    dialogState: IN_PROGRESS
    fromCity: 東京 (now with confirmationStatus: CONFIRMED)
    toCity: 稚内
    travelDate: null
    travelMode: null
    activity: null
    confirmationStatus: NON
  - *Skill returns a Dialog.Delegate directive.*
- **Alexa: いつ出発します？**
  - *(This elicitation prompt for travelDatecomes from the dialog model.)*

- **User: 今週の金曜日**
  - *Alexa sends the PlanMyTrip intent with:*
    dialogState: IN_PROGRESS
    fromCity: 東京 (confirmationStatus: CONFIRMED)
    toCity: 稚内
    travelDate: 2017-04-21
    travelMode: null
    activity: null
    confirmationStatus: NONE
  - *Skill returns a Dialog.Delegate directive. (in this example, the travelMode and activity slots are not required, so Alexa does not prompt for them.)*
- **Alexa: 東京から稚内、4月21日の出発でよろしいですか？**
  - *(Intent confirmation prompt from the dialog model.)*

- **User: いいよ**
  - *Alexa sends the PlanMyTrip intent with:*
    dialogState: COMPLETED
    fromCity: 東京 (confirmationStatus: CONFIRMED)
    toCity: 稚内
    travelDate: 2017-04-21
    travelMode: null
    activity: null
    confirmationStatus: CONFIRMED
  - *Skill now has all needed information and can handle the PlanMyTrip intent.*
    Since the dialogState is COMPLETED, the skill can no longer return the Dialog.Delegate directive.

User: ちがうよ

- *Alexa sends the PlanMyTrip intent with:*
  dialogState: COMPLETED
  fromCity: 東京 (confirmationStatus: CONFIRMED)
  toCity: 稚内
  travelDate: 2017-04-21
  travelMode: null
  activity: null
  confirmationStatus: DENIED
- *The dialog is considered complete since all information was gathered.*
  Since the dialogState is COMPLETED, the skill can no longer return the Dialog.Delegate directive.

ElicitSlot Directive Syntax

```
{
  "type": "Dialog.ElicitSlot",
  "slotToElicit": "string",
  "updatedIntent": {
    "name": "string",
    "confirmationStatus": "NONE",
    "slots": {
      "string": {
        "name": "string",
        "value": "string",
        "confirmationStatus": "NONE"
      }
    }
  }
}
```

ElicitSlot Directive Sample

```
{
  "version": "1.0",
  "sessionAttributes": {},
  "response": {
    "outputSpeech": {
      "type": "PlainText",
      "text": "From where did you want to start your trip?"
    },
    "shouldEndSession": false,
    "directives": [
      {
        "type": "Dialog.ElicitSlot",
        "slotToElicit": "fromCity",
        "updatedIntent": {
          "name": "PlanMyTrip",
          "confirmationStatus": "NONE",
          "slots": {
            "toCity": {
              "name": "toCity",
              "confirmationStatus": "NONE"
            },
```

```
            },
            "travelDate": {
              "name": "travelDate",
              "confirmationStatus": "NONE",
              "value": "2017-04-21"
            },
            "fromCity": {
              "name": "fromCity",
              "confirmationStatus": "NONE"
            },
            "activity": {
              "name": "activity",
              "confirmationStatus": "NONE"
            },
            "travelMode": {
              "name": "travelMode",
              "confirmationStatus": "NONE"
            }
          }
        }
      }
    ]
  }
}
```

ConfirmSlot Directive Syntax

```
{
  "type": "Dialog.ConfirmSlot",
  "slotToConfirm": "string",
  "updatedIntent": {
    "name": "string",
    "confirmationStatus": "NONE",
    "slots": {
      "string": {
        "name": "string",
        "value": "string",
        "confirmationStatus": "NONE"
      }
    }
  }
}
```

ConfirmSlot Directive Sample


```
{
  "version": "1.0",
  "sessionAttributes": {},
  "response": {
    "outputSpeech": {
      "type": "PlainText",
      "text": "You said you're leaving Seattle, right?"
    },
    "shouldEndSession": false,
    "directives": [
      {
        "type": "Dialog.ConfirmSlot",
        "slotToConfirm": "fromCity",
        "updatedIntent": {
          "name": "PlanMyTrip",
          "confirmationStatus": "NONE",
          "slots": {
            "toCity": {
              "name": "toCity",
              "confirmationStatus": "NONE"
            },
```

```
            "travelDate": {
              "name": "travelDate",
              "confirmationStatus": "NONE",
              "value": "2017-04-21"
            },
            "fromCity": {
              "name": "fromCity",
              "value": "Seattle",
              "confirmationStatus": "NONE"
            },
            "activity": {
              "name": "activity",
              "confirmationStatus": "NONE"
            },
            "travelMode": {
              "name": "travelMode",
              "confirmationStatus": "NONE"
            }
          }
        }
      }
    ]
  }
}
```

ConfirmIntent Directive Syntax

```
{
  "type": "Dialog.ConfirmIntent",
  "updatedIntent": {
    "name": "string",
    "confirmationStatus": "NONE",
    "slots": {
      "string": {
        "name": "string",
        "value": "string",
        "confirmationStatus": "NONE"
      }
    }
  }
}
```

ConfirmIntent Directive Sample

```
{
  "version": "1.0",
  "sessionAttributes": {},
  "response": {
    "outputSpeech": {
      "type": "PlainText",
      "text": "I'm saving your trip from Seattle to Portland on
    },                                                    April 21st. Is that OK?"
    "shouldEndSession": false,
    "directives": [
      {
        "type": "Dialog.ConfirmIntent",
        "updatedIntent": {
          "name": "PlanMyTrip",
          "confirmationStatus": "NONE",
          "slots": {
            "toCity": {
              "name": "toCity",
              "value": "Portland",
              "confirmationStatus": "CONFIRMED"
            },
```

```
            "travelDate": {
              "name": "travelDate",
              "confirmationStatus": "CONFIRMED",
              "value": "2017-04-21"
            },
            "fromCity": {
              "name": "fromCity",
              "value": "Seattle",
              "confirmationStatus": "CONFIRMED"
            },
            "activity": {
              "name": "activity",
              "value": "hiking"
              "confirmationStatus": "NONE"
            },
            "travelMode": {
              "name": "travelMode",
              "value": "driving"
              "confirmationStatus": "NONE"
            }
          }
        }
      }
    ]
}
```