

ボイス・アシスタントから見る AIの未来



丸山不二夫

The most striking aspect of linguistic competence is what we may call the 'creativity of language,' that is, the speaker's ability to produce new sentences, sentences that are immediately UNDERSTOOD by other speakers although they bear no physical resemblance to sentences which are 'familiar.'

-- Noam Chomsky

Google will fulfill its mission only when its search engine is AI-complete. You guys know what that means? That's artificial intelligence.

-- Larry Page

The ultimate search engine would understand exactly what you wanted when you typed in a query, and it would give you the exact right thing back, in computer science we call that artificial intelligence. That means it would be smart, and we're a long way from having smart computers.

-- Larry Page

はじめに

- 筆者は、コンシューマ向けの初めてのAIアプライアンスとしてのボイス・アシスタントの普及は、AIの未来を展望する上で、とても重要なステップになるだろうと考えている。そのことは、現在のボイス・アシスタント技術が完成したAI技術であることを、少しも意味しない。現状は、むしろ、その逆である。
- 現在、市場に出ているスキル/アプリの大部分は、比較的単純なプログラムの入出力をボイス化したもの、あるいは、単純な機器コントロールのボイス・コマンド化と言っているものである。また、この分野での有力なサービスは、音楽・ニュース・情報提供等の既存のコンテンツをそのまま利用したものである。
- そうした現状は、音声インターフェース技術は立ち上がったものの、「人工知能」という名にふさわしい「知的」なパーソナル・アシスタントシステムの登場は、これからの課題だということを示している。

はじめに

- それも無理がない。現在のボイス・アシスタントは、ディープ・ラーニング技術による、Speech2Text, Text2Speechの技術に基づいているのだが、それが可能にするのは、音声を文字列に、また、文字列を音声に変換することだけだからである。
- それではなぜ、ボイス・アシスタント技術を、未来の人工知能技術との関係で注目する必要があるのだろうか？
- それは、機械と人間の日常的なインターフェースが、自然言語に変わること、我々人間の側からの、機械の「知能」に対する関心と要求が、コンシューマ・ベースで、質と量の両面で、飛躍的に高まる可能性があるからである。
- それはまた、我々人間の「知能」の中核の一つである、我々人間の言語能力に対する、理論的かつ実践的な関心を大きく高めるだろう。

はじめに

- より良いボイス・アシスタント技術を目指す、コンシューマ市場での激しい競争が、自然言語処理技術に止まらず、現在の人工知能技術のいくつかの問題の解決に向けた進化の、大きな「淘汰圧」として作用するだろうことを期待している。
- 小論では、Amazon EchoとGoogle Homeを取り上げた。両者の開発モデルは、ともに、Alexa流のIntentモデルに収斂していくようにも見えるが、実際は、両者の関係は複雑である。Alexaが知識検索の能力を手に入れるか、あるいは、Googleがそうした能力を公開した時、ボイス・アシスタントをめぐる競争の第一幕は終わり、次の段階に進むだろう。
- Appendixに、「質問応答システム」としては、これまでの最良の達成であるIBM Watsonが考えていたことをまとめておいた。ここには、学ぶべきことが多くあるように考えている。

Agenda

Part I ボイス・アシスタントの現状

1. ボイス・アシスタント・システムの登場
2. 多くのアプライアンスの登場
3. 利用可能なアプリの拡大
4. Smart Homeへの応用
5. 多言語対応
6. Third Party 開発ツール

Part II Amazon EchoとGoogle Homeの 開発モデル／開発環境

1. Amazon Alexa の開発モデル
 2. Google Homeの開発環境
-

Agenda

Part III ボイス・アシスタントから見るAIの課題

1. ボイス・アシスタントは「知識」を持つか？
2. 「知識」の検索へ
 - 21世紀の大規模データ処理の基本的な流れを振り返る
3. Schema.orgとEntityモデルの問題
4. 知識データベース構築の試み
5. ボイス・アシスタントは「文」の意味を理解しているか？

Appendix

1. IBM Watson(旧)の取り組みから学ぶ
 2. ボイス・アシスタントのSmart Home / IoTへの応用
-

ボイス・アシスタントの現状



ボイス・アシスタントの現状

1. ボイス・アシスタント・システムの登場
2. 多くのアプライアンスの登場
3. 利用可能なアプリの拡大
4. Smart Homeへの応用
5. 多言語対応
6. Third Party 開発ツール

ボイス・アシスタント・システムの登場

Amazon EchoとGoogle Home

Amazon Echo	Google Home
2016/07/23 アメリカで発売開始	
	2016/11/04 アメリカで発売開始
	
	2017/10/06 日本で発売開始
2017/11/15 日本で発売開始(招待制)	

Amazon Echo

- 2015/11/06 Amazon Primeと招待客
- 2016/07/23 アメリカで発売開始
- 2016年中頃、カナダで利用可能に
- 2016/09/28 イギリスで発売開始
- 2016/10/26 ドイツで発売開始
- 2017/11/15 日本での発売開始(招待制)
 - 「Amazon Echo」(税込1万1980円)
 - 「Amazon Echo Dot」(同5980円)
 - 「Amazon Echo Plus」(同1万7980円)。



Google Home

- 2016/05/18 発表(Google I/O)
- 2016/11/04 アメリカで発売開始
- 2017/04/06 イギリスで発売開始
- 2017/06/04 カナダで発売開始
- 2017/07/20 オーストラリアで発売開始
- 2017/08/03 フランスで発売開始
- 2017/08/08 ドイツで発売開始
- 2017/10/06 日本で発売開始
 - Google Home (1万1400円)
 - Google Home Mini (5400円)



多くのアプライアンスの登場



echo dot



echo



echo plus

Amazon Alexa Show



Alexa対応製品の日本での事例



Onkyo P3



Harman Kardon Allure



The Eufy Genie



ecobee4

Alexa Voice Service (AVS) と遠隔音声制御機構を内蔵したスマートサーモスタットです。[詳細情報](#)

»



HTC U11

ハンズフリーのAmazon Alexaが搭載された世界初のスマートフォンです。[詳細情報](#)



The Eufy Genie

Anker製スピーカーがあれば、自宅のすべての部屋にAlexaの遠隔音声制御機能を設置できます。[詳細情報](#)



Sony



Panasonic



JBL



Anker



<https://goo.gl/mwzexD>



JBL Link View



Lenovo Smart Display

利用可能なアプリの拡大

Amazon Alexa Skill

様々な種類のスキル



スマートホームスキル

照明の明るさや色の調節などをAlexaに学ばせることで、ユーザーの日常に違和感なく溶け込むだけでなく、生活をより快適に導くハンズフリーの操作性を実現しましょう。 [詳細情報](#)



ゲームとトリビアのスキル

米国で、Alexaの人気上位スキルの20%はゲームです。トリビアスキル、インタラクティブなアドベンチャーゲーム、まったく新しい仮想現実の世界など、多くの人々を魅了する次なるスキルを開発しましょう。 [詳細情報](#)



カスタムスキル

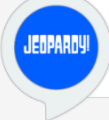
皆様のアイデアや既存のサービスに音声技術を統合することができます。独自の対話モデルを構築すれば、タクシーの呼び出しや出前の注文、情報検索など、様々な場面で手を使うことなく、声で操作するという体験をユーザーにご提供できます。 [詳細情報](#)



フラッシュブリーフィングスキル

事前に録音されたオーディオクリップや音声合成 (TTS) によって、最新のコンテンツをユーザーへ届けることができます。人気の放送局やブログからの最新情報、最新のトップニュース、天気予報など、ユーザーが求める事柄に応じて情報を提供できるようになります。 [詳細情報](#)

<https://developer.amazon.com/ja/alexa-skills-kit/learn>



"Alexa, play Jeopardy!"



"Alexa, show me recipes with chicken and potatoes"



"Alexa, turn kitchen on"

Jeopardy!J6: 自宅で楽しめるクイズ番組

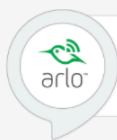
米国で人気のクイズ番組から誕生したゲーム。司会者のAlex Trebekが幅広いジャンルから毎日6問のクイズを出題します。

Allrecipes: 料理アプリ

家族での料理づくりをサポートするスキルです。ユーザーは声を出してAlexaに質問するだけで、自分好みの朝食の提案や食材、作り方など必要なことを教えてもらえます。さらに、Echo Showを使えば、画像や動画も同時に見られるので、よりわかりやすいレシピと出会うこともできます。

Bethesda: 真のスマートホーム

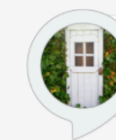
テレビや照明、天井ファン、サーモスタットからブラインドに至るまで、Alexaがコントロールするスマートホームを構築します。ユーザーの気分や活動に合わせて、あらゆるスマート家電を操作してくれるので、より快適な暮らしを楽しめるようになります



"Alexa, show me the baby's room"



"Alexa, chase thief"



"Alexa, open the Magic Door"

Echo ShowによるArloカメラ制御

Arloのユーザーは電話やアプリを使うことなく、ただEcho Showに「アレクサ、玄関を見せて」や「Alexa、赤ちゃんの部屋を見せて」などと話かけるだけで、セキュリティカメラのライブフィードを見ることができます。

The Wayne Investigation : アドベンチャーゲーム

バットマンになりきって遊ぶことができる探偵もののゲームです。Alexaを頼りに、事件の犯人を探していきます。

The Magic Door: 別世界へと続く魔法の扉

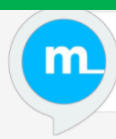
「Alexaが何でも願いをかなえてくれるとしたら、みんな何を願ひするだろうか」という、Andy Huntwork とLaura Huntwork夫妻のアイデアから始まったゲームです。森、庭、古代の寺院をナビゲートしながら、隠されたアイテムを見つけたり、謎を解決したりすることがこのゲームの醍醐味です。



"Alexa, tell drone to take off"



"Alexa, ask Smart Cap what's in front of me"



"Alexa, tell Scryb I'm going over to Sally's"

Alexa制御によるドローン

Joel Evansは、音声コマンドで飛び立つドローンをたったの500ドルで開発しました。しかもAlexaは、「キミたちの基地はすべて我々がイタダキタ」という片言の言葉で楽しませてくれます。

Smart Cap: 目が不自由な方のためのスマートキャップ

このスキルには、カメラを備えた帽子とAlexaが必要となります。誰でも帽子をかぶって「Alexa、目の前に何があるか教えて」と頼むことができます。

Scryb: ハンズフリーのテキストメッセージ入力

ソフトウェア設計会社のMacadamianによって開発されたスキルです。米国内だけで1日60億回行われていると言われるテキストメッセージ。その入力に手を使わなくて済むようになります。

ボイス・アシスタントで利用可能なアプリの拡大 Amazon Echo



Alexa Skills 

New Year New You

Departments  Browsing History  Fujio's Amazon.com Today's Deals Gift Cards  Hello, Fujio  Account & Lists  Orders Prime   Cart

Alexa Skills For Your Smart Home Games and Trivia Lifestyle Your Skills Getting Started Help

Show results for

New Arrivals

Last 7 Days

Last 30 Days

Last 90 Days

Alexa Skills

Business & Finance

Communication

Connected Car

Education & Reference

Food & Drink

Games, Trivia & Accessories

Health & Fitness

Home Services

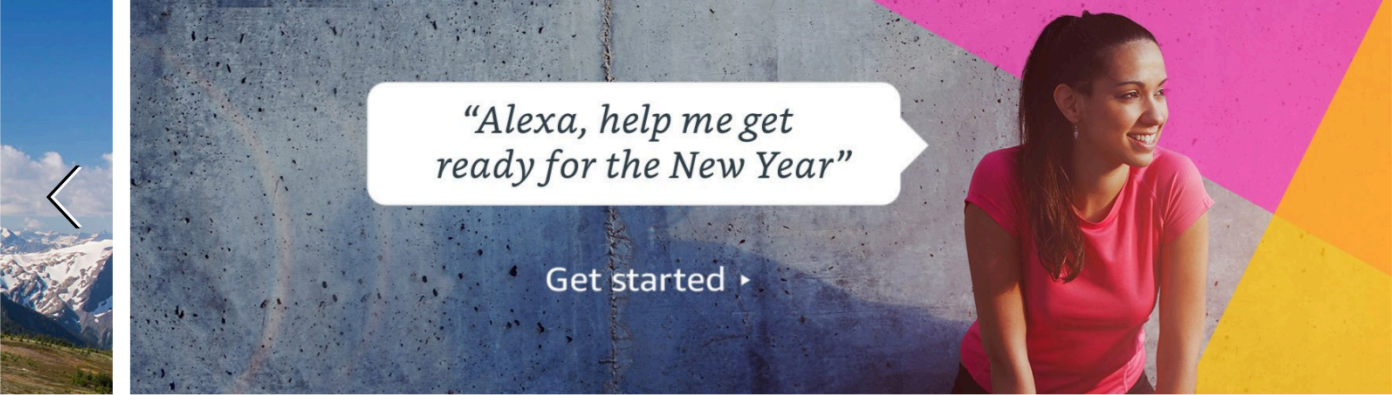
Lifestyle

Local

Movies & TV

Music & Audio

News



Get started 

"Alexa, what are your top skills?"



"Alexa, Play Jeopardy!"

Jeopardy!

★★★★☆ 1,757



"Alexa, open the box of cats"

Box of Cats

★★★★☆ 181



"Alexa, ask Jurassic Bark to bark"

Jurassic Bark

★★★★☆ 27



"Alexa, ask Jurassic Bark to bark"

Twenty Quest

★★★★☆

<https://www.amazon.com/b?node=13727921011>

"Alexa, recommend some skills."



"Alexa, what's my Flash Briefing?"

BBC

★★★★☆ 167



"Alexa, what's my Flash Briefing?"

NPR News Now

★★★★☆ 169



"Alexa, what's my Flash Briefing?"

The Daily from The New York Times

★★★★☆ 116



"Alexa, what's my Flash Briefing?"

The Washington Post

★★★★☆

"Alexa, what are your popular skills?"



"Alexa, start Song Quiz"

Song Quiz

★★★★★ 5,871



"Alexa open Sleep Sounds"

Sleep and Relaxation Sounds

★★★★★ 6,982



"Alexa, open Rain Sounds"

Ambient Sounds: Rain Sounds

★★★★★ 4,741



"Alexa, open Rain Sounds"

Millionaire Quiz

★★★★★

"Alexa, what are your games skills?"



"Alexa, Play Jeopardy!"

Jeopardy!

★★★★☆ 1,757



"Alexa, play Twenty Questions."

Twenty Questions

★★★★☆ 441



"Alexa, ask Question of the Day."

Question of the Day

★★★★★ 1,232



"Alexa, ask Question of the Day."

The Magic Door

★★★★★

"Alexa, what are your latest skills?"



"Alexa, what's my Flash Briefing?"

The Indicator from Planet Money

★★★★★ 2



"Alexa, open Twitch"

Twitch

★★★★★ 18



"Alexa, tell FordPass to unlock my car with PIN 1234."

FordPass

★★★★★ 3



"Alexa, tell IMDb's What's Hot to show me the top movies."

IMDb's What's Hot

★★★★★

"Alexa, help me relax."



"Alexa, open Guided Meditation"

Guided Meditation: Meditation of the Mind

★★★★★ 74



"Alexa, start Peaceful Habit"

Peaceful Habit

★★★★☆ 63



"Alexa open Healing Sounds"

Zen Sounds: Healing Sounds

★★★★★ 1,347



"Alexa, open Meditation Studio"

Meditation Studio

★★★★★

"Alexa, what are your Echo Button skills?"



"Alexa, launch Trivial Pursuit Tap"

Trivial Pursuit Tap

★★★★☆ 54



"Alexa, launch Party Foul"

Party Foul

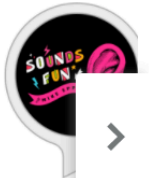
★★★★☆ 12



"Alexa, open Beat The Intro"

Beat The Intro

★★★★☆ 25



"Alexa, open Sounds Fun with Alexa"

Sounds Fun with Alexa

★★★★☆

"Alexa, what are your smart home skills?"



"Alexa, tell Vivint to lock my doors"

Vivint.SmartHome - Complete

★★★★☆ 146



"Alexa, ask Roomba to start cleaning."

iRobot Home

★★★★☆ 262



"Alexa, ask ecobee to set the bedroom to 72 degrees."

ecobee plus

★★★★☆ 51



>

Tile

★★★★☆

"Alexa, what are your news skills?"



"Alexa, open ESPN."

ESPN

★★★★☆ 9



"Alexa, ask The Wall Street Journal for a news summary"

The Wall Street Journal

★★★★☆ 18



"Alexa, open NBA"

NBA

★★★★☆ 18



>

'We Are' Chee

★★★★☆

Alexa でレシピやレ
ストランを探したり、
食事を頼ん
だりしてみよう

アレクサ、クックパッドを開いて

使いたい材料を教えてください



アレクサ、スシローの
持ち帰りを始めて



アレクサ、ぐるなびで
新宿の焼肉を調べて



アレクサ、すき家の
牛丼を開いて



アレクサ、ニュースを教えて

フラッシュニュースです

Alexa で毎日の
ニュースを聞こう



アレクサ、ニュースを教えて



アレクサ、ニュースは何？



アレクサ、
Yahoo!ニュースを開いて

アレクサ、寝室の電気を消して

はい

Alexaで家電を
操作してみよう



アレクサ、ルンバを使って
掃除して



アレクサ、家電リモコンを使って
テレビをつけて



アレクサ、リビングルームの
電気をつけて

Alexa でお出かけ
を便利にしよう

アレクサ、ナビタイム乗換案内で
渋谷から新大阪を調べて

渋谷から新大阪の経路です。所要時間は
3時間です。料金は1万3620円です。

LEAF

アレクサ、リーフの
バッテリー残量を確認して



全国タクシー

アレクサ、全国タクシーで
タクシーを探して



アレクサ、じゃらんを開いて

Amazon Alexaスキル一覧 (265スキル)

2017年11月8日現在

- | | | |
|--|--|--|
|  @cosme ビューティニュース
株式会社アイスタイル |  @cosme 化粧品ランキング
株式会社アイスタイル |  528究極の癒しミュージック
株式会社エクシング |
|  ABC関西ニュース
朝日放送株式会社 |  AbemaTimes
株式会社AbemaTV |  AbemaTV
株式会社AbemaTV |
|  ADASTRIA ファッション素材
知識 ・株式会社アグストリア |  AKIBA PC Hotline!ニュース
株式会社インプレス |  ALSOKあんしん英会話
総合警備保障株式会社 |
|  ANA
全日本空輸株式会社 |  AV Watchニュース
株式会社インプレス |  B. League News
公益社団法人ジャパン・プロフェッショナル・バスケットボールリーグ |
|  Business Insider Japan
株式会社メディアジーン |  BuzzFeed Japan News
BuzzFeed Japan株式会社 |  cafe globeニュース
株式会社メディアジーン |
|  Car Watchニュース
株式会社インプレス |  CNET Japan News
朝日インタラクティブ株式会社 |  CNN.co.jp News
朝日インタラクティブ株式会社 |
|  COCORO KITCHEN
シャープ株式会社 |  Conteホーム さあ、はじめよう!
かんたんスマート生活。
株式会社ピクセラ |  DELISH KITCHENのおすすめ
レシピ特集 ・株式会社エブリー |
|  DIGIDAYニュース
株式会社メディアジーン |  d Hits
株式会社NTTドコモ |  E・レシピの今日の献立
エキサイト株式会社 |
|  ENERTALK
エンコアードジャパン株式会社 |  From Aしよ!!
株式会社リクルートホールディングス |  FUZEニュース
株式会社メディアジーン |
|  GAME Watchニュース
株式会社インプレス |  GLITTYニュース
株式会社メディアジーン |  GooMAGAZINE 中古車の
情報ならグーネット
株式会社プロトコーポレーション |
|  Hey MIKU!
クリプトン・フューチャー・メディア株式会社 |  Hue
Philips Lighting B.V. |  Impress Watch Headline
ニュース ・株式会社インプレス |
|  INTERNET Watchニュース
株式会社インプレス |  ippin
株式会社ぐるなび |  iRemocon Smart Home
Skill ・株式会社グラモ |



INTERNET Watchニュース
株式会社インプレス



iRobot Home
アイロボットジャパン合同会社



ITmedia NEWS
アイティメディア株式会社



J-WAVE
株式会社J-WAVE



JAL
日本航空株式会社



JTBチケット検索
株式会社ジェイティービー



kurashiru クラシル
dely株式会社



LIFX Optimised for Smart Home Japan • LiFi Labs, Inc.



MOTTAINAI SOUND
エキサイト株式会社



NAVITIMEフライト情報 - ナビタイム
なら飛行機の運航状況も検索できる
株式会社ナビタイムジャパン



NEWSポストセブン
株式会社小学館



ippin
株式会社ぐるなび



ITmedia LifeStyle
アイティメディア株式会社



ITmedia PC USER
アイティメディア株式会社



J-WAVE CHECK THE TOKIO HOT 100 • 株式会社J-WAVE



JEANASIS music skill
株式会社アグストリア



JTBホテル検索
株式会社ジェイティービー



LEAF
日産自動車株式会社



LinkJapan
株式会社リンクジャパン



Music Shower
電通アイソバー株式会社



NAVITIME乗換案内 - ナビタイムの乗
り換え案内で、電車の移動もおまかせ!
株式会社ナビタイムジャパン



NHKラジオニュース
日本放送協会(NHK)



iRemocon Smart Home Skill • 株式会社グラモ



ITmedia Mobile
アイティメディア株式会社



ITmedia ビジネスオンライン
アイティメディア株式会社



J-WAVE HEADLINE NEWS
株式会社J-WAVE



JR東日本 列車運行情報案内
東日本旅客鉄道株式会社



Kufuraー仕事も家庭も、私らしく!
を叶えるヒント • 株式会社小学館



Lifehackerニュース
株式会社メディアジーン



LIVE JAPAN
株式会社ぐるなび



MYLOHASニュース
株式会社メディアジーン



NAVITIME運行情報 - ナビタイムなら
電車の運行状況も検索できる
株式会社ナビタイムジャパン

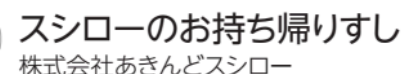
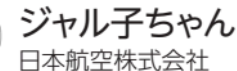
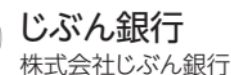
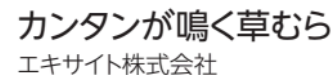
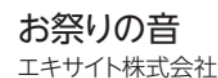
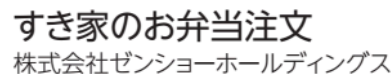
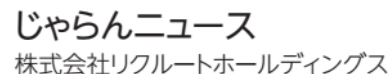
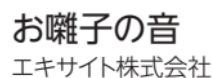
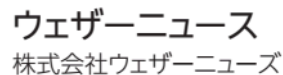
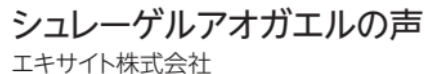
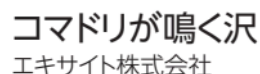
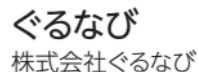
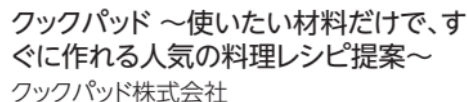
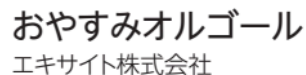


nocria ノクリア
株式会社富士通ゼネラル

<https://goo.gl/eQEXx6>

- | | | |
|--|---|--|
|  PARCO
株式会社パルコ |  PC Watch ニュース
株式会社インプレス |  Qiita 人気の投稿
Increments株式会社 |
|  Qrio Smart Lock
Qrio株式会社 |  QUICK
株式会社QUICK |  radiko.jp
株式会社radiko |
|  RBBTODAY News
株式会社イード |  Response.jp News
株式会社イード |  ROOMIEニュース
株式会社メディアジーン |
|  Secual
株式会社Secual |  SOMPO
SOMPOホールディングス株式会社 |  SUUMOジャーナル
株式会社リクルートホールディングス |
|  TABROOM NEWS
株式会社リクルートホールディングス |  TATERU Kit
株式会社インベスターズクラウド |  TBSニュースランキング
株式会社TBSテレビ |
|  TBSラジオニュース
株式会社TBSラジオ |  TechCrunch Japan
Oath Japan株式会社 |  TechRepublic Japan News
朝日インタラクティブ株式会社 |
|  TEPCOホーム
東京電力エナジーパートナー株式会社 |  TOKAIうるのん
株式会社TOKAI |  TOKAIおいしい水の宅配便
株式会社TOKAI |
|  TOKYO MX NEWS
東京メトロポリタンテレビジョン株式会社 |  Voicy
株式会社Voicy |  Watch Videoニュース
株式会社インプレス |
|  WIRED
合同会社コンデナスト・ジャパン |  Yahoo! ニュース
ヤフー株式会社 |  Yahoo! ニュース forフラッシュ
ニュース・ヤフー株式会社 |
|  Yahoo!天気・災害
ヤフー株式会社 |  Yomiuri Digital News
株式会社読売新聞東京本社 |  ZDNet Japan News
朝日インタラクティブ株式会社 |
|  アカショウビンが鳴く山奥の池
エキサイト株式会社 |  あすけん
株式会社ウィット |  アダストリアファッショントレンド
トーク・株式会社アダストリア |
|  アニメ情報 by TOKYO MX
東京メトロポリタンテレビジョン株式会社 |  アルフの英語クイズ
株式会社アルフ |  アルフの英語入門
株式会社アルフ |

<https://goo.gl/eQEXxC>



<https://goo.gl/eQEXxC>

 **スタディサプリ進路**
株式会社リクルートホールディングス

 **スマート家電コントローラ**
ラトックシステム株式会社

 **ゼクシキッチン**
株式会社リクルートホールディングス

 **タンチョウの求愛ダンス**
エキサイト株式会社

 **テレビ東京ニュース**
株式会社テレビ東京コミュニケーションズ

 **ニッカンスポーツニュース**
株式会社日刊スポーツ新聞社

 **ハイウェイバスドットコム**
京王電鉄バス株式会社

 **ぴあ首都圏イベント情報**
ぴあ株式会社

 **ヒグラシの声**
エキサイト株式会社

 **フジテレビスキル**
株式会社フジテレビジョン

 **ホットペッパーグルメ**
株式会社リクルートホールディングス

 **マネパレート**
株式会社マネーパートナーズ

 **スポーツブルニュース**
株式会社運動通信社

 **スマート留守電**
ソースネクスト株式会社

 **ダイヤモンド・オンライン**
株式会社サイバード

 **チカの元気が出るひとこと**
エキサイト株式会社

 **テレ朝news**
株式会社テレビ朝日

 **ニッポン放送**
株式会社ニッポン放送

 **はぴeみる電**
関西電力株式会社

 **ピカチュウトーク**
株式会社ポケモン

 **ヒバリが鳴く大草原**
エキサイト株式会社

 **ベースマガジンチューナー**
株式会社リットーミュージック

 **ホットペッパービューティー**
株式会社リクルートホールディングス

 **みずほ銀行**
株式会社みずほ銀行

 **スマートフォンを探す**
TrackR, Inc.

 **セキララゼクシィ**
株式会社リクルートホールディングス

 **タウンワークマガジン**
株式会社リクルートホールディングス

 **デジカメ Watchニュース**
株式会社インプレス

 **トラベル Watchニュース**
株式会社インプレス

 **ニホンアマガエルの合唱**
エキサイト株式会社

 **ハローウイスキー**
サントリーコミュニケーションズ株式会社

 **ヒガラとミソサザイが鳴く山**
エキサイト株式会社

 **フクロウが鳴く森**
エキサイト株式会社

 **ヘルシーシェフ お料理検索**
株式会社日立製作所

 **ホンネスト**
株式会社リクルートホールディングス

 **ミニマル教えてサポート**
株式会社日立製作所

<https://goo.gl/eQEXxC>

- | | | | | | |
|--|---|---|--|---|---|
|  | メシ通
株式会社リクルートホールディングス |  | モーニングCROSS
東京メトロポリタンテレビジョン株式会社 |  | モデルプレス・エンタメ
株式会社ネットネイティブ |
|  | モデルプレス・コラム
株式会社ネットネイティブ |  | ラジオクラウド
株式会社博報堂DYメディアパートナーズ |  | リクナビNEXTジャーナル
株式会社リクルートホールディングス |
|  | リクナビNEXT転職成功
ノウハウ
株式会社リクルートホールディングス |  | るるぶNEWS関東版
株式会社ジェイティービー |  | るるぶNEWS関西版
株式会社ジェイティービー |
|  | レタスクラブの本日のレシピ
株式会社KADOKAWA |  | ロボスタニュース
ロボットスタート株式会社 |  | 赤すぐ みんなの体験記
株式会社リクルートホールディングス |
|  | 秋の夜更け
エキサイト株式会社 |  | 朝日新聞アルキキ
株式会社朝日新聞社 |  | 委員長・椎名乙葉 タイマー
ですよ ・株式会社サイバード |
|  | 委員長・椎名乙葉 時間ですよ
株式会社サイバード |  | 岩場に打ち付ける荒波
エキサイト株式会社 |  | 海の音
エキサイト株式会社 |
|  | 運行情報 by 駅すぱあと
株式会社ヴァル研究所 |  | 映画.comニュース
株式会社エイガ・ドット・コム |  | 英語で泣けるちょっといい話
株式会社アルク |
|  | 駅しりとり powered by 駅す
ぱあと ・株式会社ヴァル研究所 |  | 小川のせせらぎ
エキサイト株式会社 |  | 教えて!ぐりこっち
江崎グリコ株式会社 |
|  | 海外旅行保険見積もり
SOMPOホールディングス株式会社 |  | 海岸の岩場
エキサイト株式会社 |  | 海岸の波打ち際
エキサイト株式会社 |
|  | 鏡リュウジ占星術 今日の
恋愛運 ・株式会社サイバード |  | 風が吹く草原
エキサイト株式会社 |  | 家電 Watchニュース
株式会社インプレス |



空返事さん

株式会社よしもとロボット研究所



今日のうお座の恋愛運

エキサイト株式会社



今日のおひつじ座の恋愛運

エキサイト株式会社



今日のしし座の恋愛運

エキサイト株式会社



今日のやぎ座の恋愛運

エキサイト株式会社



健康飲料ナビ

サントリーコミュニケーションズ株式会社



四国新聞社

株式会社四国新聞社



集中タイム

エキサイト株式会社



新刊JPニュース

株式会社オトバンク



鈴虫の鳴き声

エキサイト株式会社



全国タクシー

JapanTaxi株式会社



食ベログ

株式会社カカクコム



今日のみずがめ座の恋愛運

エキサイト株式会社



今日のおうし座の恋愛運

エキサイト株式会社



今日のかに座の恋愛運

エキサイト株式会社



今日のてんびん座の恋愛運

エキサイト株式会社



京王グループ: 運行情報・おすすめ・クーポン

京王電鉄株式会社



木の葉に落ちる雨

エキサイト株式会社



鹿威し

エキサイト株式会社



女子旅プレス

株式会社ネットネイティブ



水琴窟の音

エキサイト株式会社



住信SBIネット銀行

住信SBIネット銀行株式会社



早朝の野鳥コーラス

エキサイト株式会社



知恵ラジオ

ヤフー株式会社



今日のいて座の恋愛運

エキサイト株式会社



今日のおとめ座の恋愛運

エキサイト株式会社



今日のさそり座の恋愛運

エキサイト株式会社



今日のふたご座の恋愛運

エキサイト株式会社



溪流で歌うカジカガエル

エキサイト株式会社



災害情報

東京海上日動火災保険株式会社



資生堂

資生堂ジャパン株式会社



白糸の滝

エキサイト株式会社



水滴の音

エキサイト株式会社



**請求額・ポイント残高かんたん
確認アプリ for Alexa**

三菱UFJニコス株式会社



 天気予報トピックスのウェザー
ニュース
株式会社ウェザーニュース

 東京インフォメーション
東京メトロポリタンテレビジョン株式会社

 鳥のさえずり
エキサイト株式会社

 鳴り続ける雷
エキサイト株式会社

 日テレNEWS24
日本テレビ放送網株式会社

 野村アセットマネジメント
野村アセットマネジメント株式会社

 東池袋52
株式会社クレディセゾン

 窓の杜ニュース
株式会社インプレス

 三菱UFJモルガン・スタンレー
証券 投資情報
三菱UFJモルガン・スタンレー証券株式会社

 読むテレ東
株式会社テレビ東京コミュニケーションズ

 東急線アプリ
東京急行電鉄株式会社

 東京海上日動
東京海上日動火災保険株式会社

 夏の音
エキサイト株式会社

 日刊カーセンサー
株式会社リクルートホールディングス

 寝たまんまヨガ〜ピースフル・
ヨガ〜・エキサイト株式会社

 野村証券
野村証券株式会社

 光BOX+
西日本電信電話株式会社

 豆しば
株式会社電通

 三菱東京UFJ銀行 Voice
Banking for Alexa β版
株式会社三菱東京UFJ銀行

 東急ハンズ
株式会社東急ハンズ

 遠くの雷鳴
エキサイト株式会社

 波の音
エキサイト株式会社

 日本経済新聞 電子版
株式会社日本経済新聞社

 寝たまんまヨガ〜リラックス・
ヨガ〜・エキサイト株式会社

 春風の吹く棚田
エキサイト株式会社

 毎日新聞ニュース
株式会社毎日新聞社

 三井住友銀行
株式会社三井住友銀行

 山に降る雨
エキサイト株式会社

Google Assistant Agent

Featured Integrations



Google Assistant

Build apps for the Google Assistant users on voice-activated speakers like Google Home, eligible Android phones and iPhones.

[One-click integration](#)



Facebook Messenger

Build Messenger bots to connect with users on Facebook's Messenger Platform.

[One-click integration](#)
[Integration toolkit on Github](#)



Slack

Build chatbots for teams and companies to use on Slack.

[One-click integration](#)
[Integration toolkit on Github](#)
[Botkit-based SDK](#)



Dialogflow Web Demo

[Web Demo instructions](#)



Kik

[One-click integration](#)
[Integration toolkit on Github](#)



Line

[One-click integration](#)
[Integration toolkit on Github](#)



Skype

[One-click integration](#)
[Integration toolkit on Github](#)



Cisco Spark

[One-click integration](#)
[Integration toolkit on Github](#)



Telegram

[One-click integration](#)
[Integration toolkit on Github](#)



Cisco Tropo

[One-click integration](#)
[Integration toolkit on Github](#)



Twilio

[One-click integration](#)
[Integration toolkit on Github](#)



Twilio Programmable Chat

[One-click integration](#)
[Integration toolkit on Github](#)



Twitter

[One-click integration](#)



Viber

[One-click integration](#)

Prebuilt agents

Name	Description	Support
Agent Name	Conversations to change your agents name	en
Alarm	Set and manage alarms	en
App Management	Manage your applications, user controlled settings, and preferences	en
Banking	Banking operations	en
Car	Interact with a vehicle's systems	en
Coffee Shop	Order drinks, snacks and gift cards from a coffee shop	en
Currency Converter	Get exchange rates for major currencies	en
Date	Date calculator and holiday lookup	en
Device	Control your device settings	en
Dining Out	Search restaurants, cafes and bars	en
Easter Eggs	Agent answers fun questions	en
Event Search	Search for tickets to events	en
FAQ	A very simple FAQ agent	en
Flights	Find, create, and manage reservations for flights	en

Food Delivery	Create and manage food and drink orders	en
Formats	Control default units of measurement	en
Hotel Booking	Find, create and manage reservations for hotels	en
Job Interview	Sample job interview	en
Jokes	The agent tells jokes	en
Language Settings	Set language preferences	en
Local Services	Search local services and shops	en
Maps	Search maps for a location	en
Mobile Account	Information about mobile accounts	en
Music	Play and control your music and playlists	en
Navigation	Ask for directions	en
News	Get news stories and manage news feed	en
Online Shopping	Search items and manage orders for online shopping	en
Radio	Control playing radio stations	en
Reminders	Schedule, edit, view and remove reminders	en
Restaurant Booking	Find create and manage reservations for restaurants	en

Small Talk	Small talk topics to help make your bot more personable	en, dzh-t
Smart Home	Control your IoT connected devices	en
Support	Support and CRM related conversations	en, dzh-t
Time	Time calculator and converter commands	en
Tips Calculator	Calculate tip percentages	en
Tourism	Search for sights, entertainment and transportation	en
Traffic	Get traffic details	en
Translate	Translate phrases to and from languages	en, dzh-t
TV	Control connected TVs	en
Unit Converter	Convert to and from different units of measure	en
User Name	Conversations to change a users name	en
Video	Search for and play videos	en
Volume	Control volume of your device	en
Weather	Weather and forecast related conversations	en, dzh-t
Web Search	Perform a web search	en

ボイス・アシスタントの Smart Home / IoTへの応用

Alexa Compatible

Philips Hue White LED Starter Kit

Best Alexa-compatible smart bulbs

Philips Hue's lighting platform is terrific even without Alexa involved, and this white-light-only starter kit is your most affordable way in.

[Read full review](#)

\$69.95

at Apple

[See It](#)

[See all prices](#)

CNET'S RATING

8

9

8

9



\$59.99

MSRP

Not Available

CNET'S RATING

8

9

8

9

Features

Usability

Design

Performance

Lixt Color 1000

Best Alexa-compatible color-changer

It's not Hue you think. Alexa still can't change the colors of Philips bulbs, so go with Lixt, which offers its own custom Alexa skill for voice-activated color changes.

[Read full review](#)

Lutron Caseta In-Wall Wireless Smart Lighting Kit

Best Alexa-compatible smart switches

WeMo Switches are cheaper, but Lutron's switches actually let you dim the lights. They also work with three-way switch setups (WeMo Switches won't), and they offer a much better slate of features, too.

[Read full review](#)

\$159.90

at Amazon.com

[See It](#)

[See all prices](#)

CNET'S RATING

9

Features

9

Usability

8

Design

9

Performance



\$149.00

at Amazon.com

[See It](#)

[See all prices](#)

CNET'S RATING

7.5

Features

8

Usability

9

Design

9

Performance

Haiku Home Haiku Light

Best high-end smart lights for Alexa

They're expensive, but Haiku Home's overhead smart fixtures work flawlessly with Alexa, and look terrific doing it. They also pack built-in presence detectors.

[Read full review](#)

Emerson Sensi

Best Alexa-compatible thermostat

We wouldn't blame you for splurging on something from Nest or Ecobee, but this thermostat from Emerson will do the job just as well at a much lower cost.

[Read full review](#)

\$93.11

at Amazon.com

See It

See all prices

CNET'S RATING

7

9

6

8

Features

Usability

Design

Performance



\$604.99

at Jet.com

See It

See all prices

CNET'S RATING

8

7

8

10

Features

Usability

Design

Performance

Neato Botvac Connected Robot Vacuum

Best Alexa showcase

Admit it. You want to tell an artificial intelligence to tell another artificial intelligence to clean your carpets. With Alexa and Neato, you can -- plus, it's one of the best robot vacuums we've tested.

[Read full review](#)



CNET Smart Home

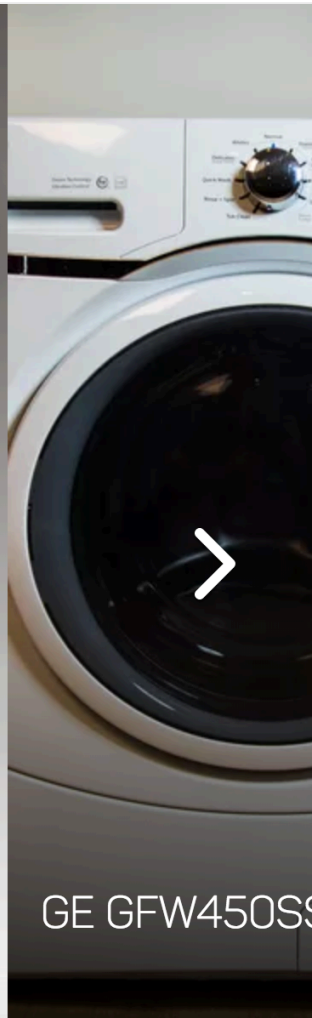
<https://www.cnet.com/smart-home/house/>



CNET Smart
Apartment



Bonavita Connoisseur

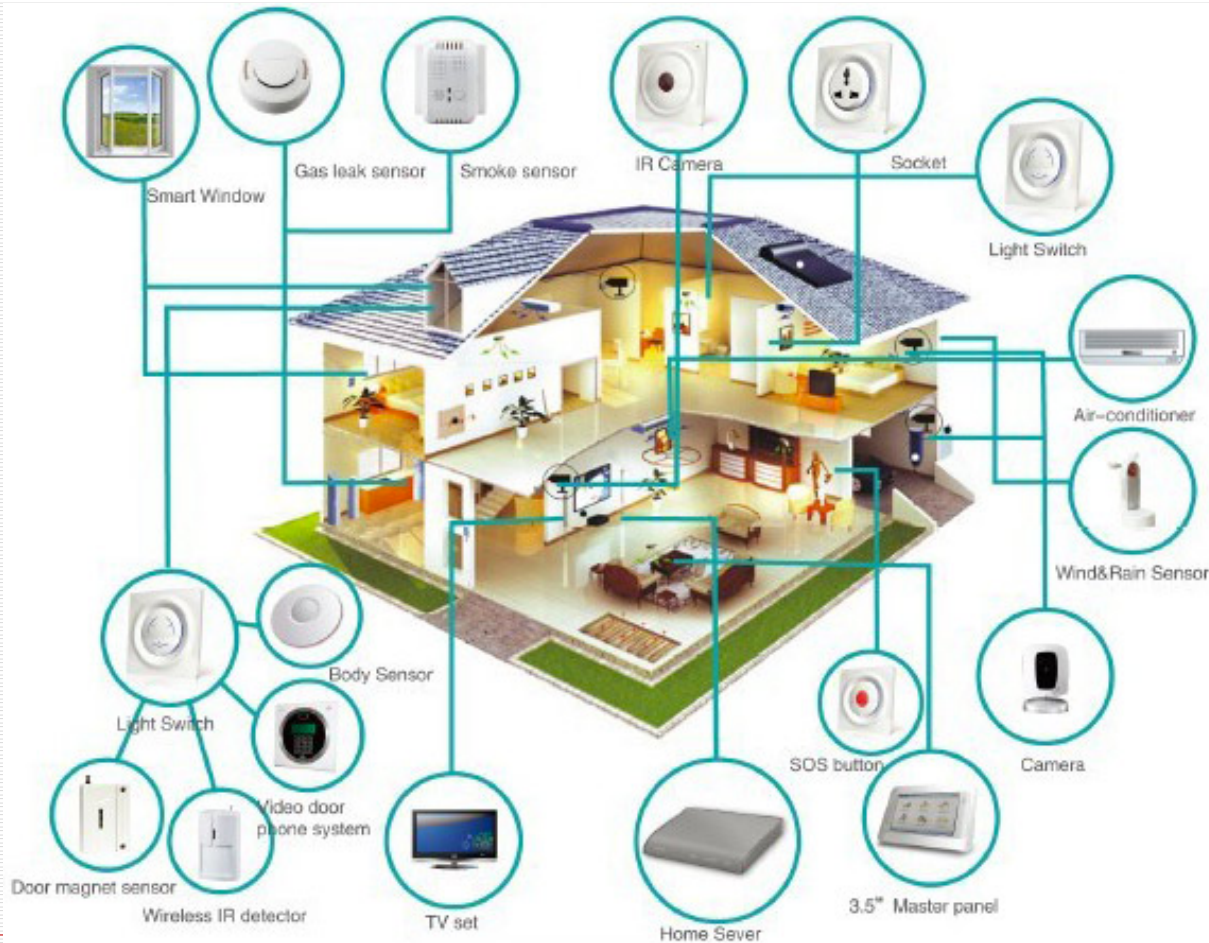


GE GFW450SS

Google Smart Home

Google Smart Home Device Types

- ☐ Camera
- ☐ Dishwasher
- ☐ Dryer
- ☐ Light
- ☐ Lock
- ☐ Outlet
- ☐ Refrigerator
- ☐ Scene
- ☐ Switch
- ☐ Thermostat
- ☐ Vacuum
- ☐ Washer



<https://developers.google.com/actions/smarthome/guides/>

Applets for Google



**Sync new events
added from an iOS
Calendar to your
GCal**

by Google ✓

 3.3k

works with 



**Get a notification
for birthday events
in your calendar**

by Google ✓

 22k

works with 



**Use Google
Assistant to add a
new Google
Contact**

by Google ✓

 21k

works with 

<https://ifttt.com/collections/google>



Get an email when Google Assistant publishes a new trigger or action

by Google ✓



43k

works with IFTTT



Get an email when a new Google Assistant Applet is published

by Google ✓



21k

works with IFTTT



Toggle your Yeelights on and off with your voice

by Google ✓



2.4k

works with



Create an event on your iPhone's Calendar with Google Home

by Google ✓



26k

works with



Add a new to-do in your iPhone's Reminders app with Google Assistant

by Google ✓



27k

works with



Turn on your LIFX lights

by Google ✓



1.6k

works with





Dim your Lutron Caseta Wireless lights with your voice

by Google ✓

310

works with



Send a text message to someone with your Android and Google Home

by Google ✓

27k

works with



Add to DayOne journal with your voice

by Google ✓

400

works with



Log notes in a Google Drive spreadsheet

by Google ✓

9.6k

works with



Post to Facebook and Twitter with your voice

by Google ✓

4.5k

works with



Lock a SmartThings device with your voice

by Google ✓

works with



Tell Google Assistant to set and monitor your Nest thermostat

by Google ✓



4.9k

works with



Ok Google, party time

by Google ✓



27k

works with

hue



Tell Google Assistant to call your phone

by Google ✓



73k

works with



Turn on TV by voice

by Google ✓



30k

works with

Harmony



Block the next hour on Google Calendar

by Google ✓



5.5k

works with



Add a task to Todoist by voice

by Google ✓



17k

works with





Post to Facebook by voice

by Google ✓

 27k

works with 



Ok Google, bedtime

by Google ✓

 29k

works with 



Tell Google Assistant to change the color of your Hue lights

by Google ✓

 10k

works with 



Post a tweet using your voice

by Google ✓

 5.8k

works with 



Keep a list of notes to email yourself at the end of the day

by Google ✓

 12k

works with 



Use Google Assistant to add a new Google Contact

by Google ✓

 8.8k

works with 



Configure



Configure



Google Assistant

Say a simple phrase

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase you choose. For example, say "Ok Google, I'm running late" to text a family member that you're on your way home.

What do you want to say?

部屋の電気をつけて

What's another way to say it? (optional)

部屋の明かりをつけて

And another way? (optional)

点灯

What do you want the Assistant to say in response?

灯りを点けます

Language

Japanese



Nature Remo

Control home appliances

This action will send an infrared signal to control your home appliances.

Which button?

ライト - 全灯

Select the name of the button you have in your iOS/Android Remo app



多言語対応

Alexaの多言語対応

カテゴリー	米国	英国	カナダ	ドイツ	インド*	日本*
音楽	Amazon Music TuneIn iHeartRadio Pandora SiriusXM	Amazon Music TuneIn	TuneIn Amazon Music	Amazon Music TuneIn	Amazon Music TuneIn Saavn	Amazon Music TuneIn
本	Audible Kindle: 本を読む	Kindle: 本を読む	Audible Kindle: 本を読む	Audible Kindle: 本を読む	Kindle: 本を読む	Kindle: 本を読む
スマートホームコントロール	○	○	○	○	○	○
音声ショッピング	○	○	○	○	○	○
通知	○	○	○	○	○	○
Wikipedia	○	○	○	○	○	○
交通情報	○	○	○	○	○	○
タイマーとアラーム	○	○	○	○	○	○

IFTTT

○

○

○

○

○

○

Alexa Alexaス
キル

○ [米国のスキ
ルを見る](#)

○ [英国のスキ
ルを見る](#)

○ [カナダのス
キルを見る](#)

○ [ドイツのス
キルを見る](#)

○ [インドのス
キルを見る](#)

○ [日本のスキ
ルを見る](#)

Alexa開発環境の日本語対応

Alexa で開発を始める

Alexa Skill Kit を使うと、Alexaに新しい音声対応機能を追加することができます。Alexa Voice Service を使うと、インターネットに接続されている



Alexa Skills Kit

Alexa に新しい機能を追加する

始める >



Alexa Voice Service

インターネットに接続されているデバイスに音声対応機能を追加する

始める >

Alexa Skill Kit を使用して Alexa スキルを開発する

新しいスキルを追加する

Alexa のスキル開発に関する詳細は、[Alexa Skill Kit によるスキルの開発](#)を参照してください。AWS Lambda を使用して Alexa スキルを無償で開発するには、[カスタムスキルのAWS Lambda関数を作成する](#)を参照してください。

[Alexa デベロッパーフォーラム](#)にアクセスすると、Alexa チームのメンバーや Alexa デベロッパーとつながることができます。

スキル名	言語	タイプ	更新日	ステータス	アクション
 nodejs-fact スキル ID の表示	English (U.S.)	Custom	9/6/17	Development ⓘ	📈 メトリックス ✎ 編集 🗑 削除
 PlanMyTrip スキル ID の表示	English (U.S.)	Custom	9/5/17	Development ⓘ	📈 メトリックス ✎ 編集 🗑 削除

グローバルフィールド

これらの項目はスキルが対応するすべての言語に対して適用されます。

Audio Player

このスキルはAudioPlayerのディレクティブを使用 ☐ はい ☒ いいえ
しますか？詳細は [こちら](#) をご覧ください。

ビデオアプリ

このスキルはビデオアプリのディレクティブを使 ☐ はい ☒ いいえ
用しますか？詳細は [こちら](#) をご覧ください。

レンダリングテンプレート

このスキルはレンダーテンプレートを使用します ☐ はい ☒ いいえ
か？詳細は [こちら](#) をご覧ください。

保存

次へ

Japanese ▼

新しい言語を追加

スキル情報



対話モデル



設定



テスト



公開情報



プライバシーとコンプライアンス



Skills Beta Testing **NEW**

ステータス: まだ利用できません 



スキルビルダー(ベータ版)をお試ください。直感的なインターフェイスで対話モデルとダイアログプロンプトを作成できます。

スキルビルダーを起動する **ベータ**

インテントスキーマ

JSON 形式でユーザーのインテントを記述します。詳しくは[インテントスキーマ](#)を参照してください。また[ビルトインスロット](#)と[ビルトインインテント](#)も参照してください。



1

Google Assistanceの多言語対応

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)

開発環境Dialogflowの日本語対応



Project を作成

My Sample-1

CREATE

DEFAULT LANGUAGE ?

Japanese — ja 日本語対応 ▼

Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

(GMT+9:00) Asia/Tokyo ▼

Date and time requests are resolved using this timezone.

GOOGLE PROJECT

Create a new Google project ▼

Enables Cloud functions, Actions on Google and permissions management.

API VERSION



Dialogflow V2 API ^[beta]

Use [Dialogflow V2 API](#) as default for the agent. Your webhook will receive requests in the [new format](#).



Weather

Intent を作成

User saysの入力

SAVE



Contexts

User says

Search in user says



Add user expression



一月三日の東京の雨の確率

パラメーターとエンティティの自動抽出

PARAMETER NAME

ENTITY

RESOLVED VALUE

date

@sys.date

一月三日の



geo-city

@sys.geo-city

東京



稚内の明日の天気



今日の天気は



● Weather

IntentとEntity

SAVE



Action



Enter action name

エンティティの指定



REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?
☐	date	@sys.date	\$date	☐
☐	geo-city	@sys.geo-city	\$geo-city	☐
☐	Enter name	Enter entity	Enter value	☐

+ New parameter

Response ?



DEFAULT +



Weather

IntentとResponseの入力

SAVE



geo-city

@sys.geo-city

\$geo-city



Enter name

Enter entity

Enter value



+ New parameter

Response ?



DEFAULT



Text response



- 1 わかりません
- 2 天気のこと、僕知らないの
- 3 情報検索できないの
- 4 Enter a text response variant

ADD MESSAGE CONTENT

Third Party 開発ツール

Converse AI



FEATURES

INTEGRATIONS

PRICING

BLOG

DOCS

Get Started For FREE!

Connect your entire business through
conversation

<http://www.converse.ai/>

Converse AI



EASY TO INTEGRATE WITH
YOUR EXISTING SERVICES

Converse AI

FOR YOUR SUCCESS



FAQ Handling



Voice, Image & Video Support



Automate Workflows



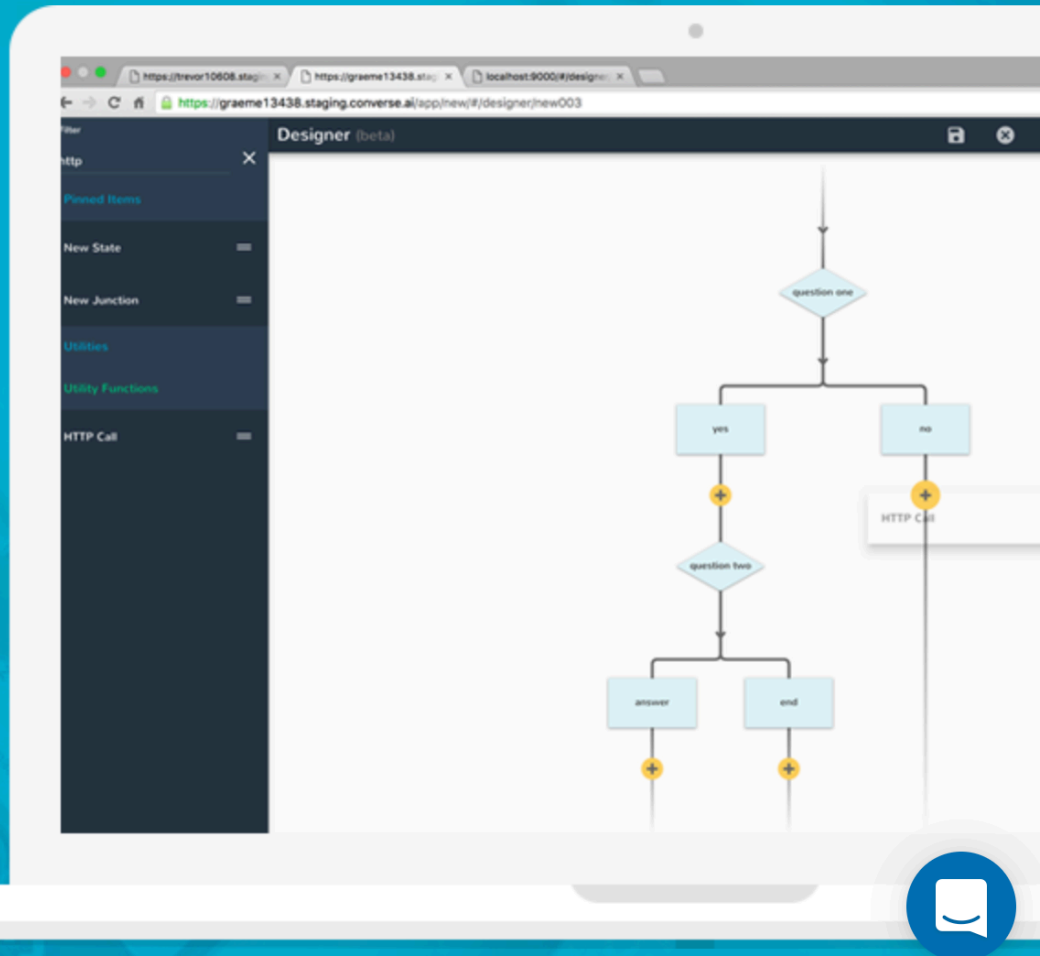
Integrate into internal tools



Quick Start Templates



Surveys & Notifications



PullString Converse

Converse empowers teams to design, prototype, and publish voice apps for Amazon Alexa, Google Assistant, and IoT devices.

<https://www.pullstring.com/product/converse>

Wow Customers with Engaging Voice Apps to Drive Usage and Repeat Visits



Write Conversation, not Code

Focus on the creative aspects of conversation building to craft great voice experiences. Converse will handle the complexity of AI, State management and much more for you.



Differentiate with your own voice

Converse allows you to define persona, conversation context, rich dialogue management and the ability to bring custom audio to create differentiated experiences.



Teamwork

Behind every great voice is a team. With PullString Converse, Creatives, Developers, Legal and Marketing can work collaboratively through all the steps required to publish your app.



Complete Solution

From prototyping, design and development to publishing, analyzing and training, Converse offer an end to end solution to bring your app to market quickly and iterate efficiently.

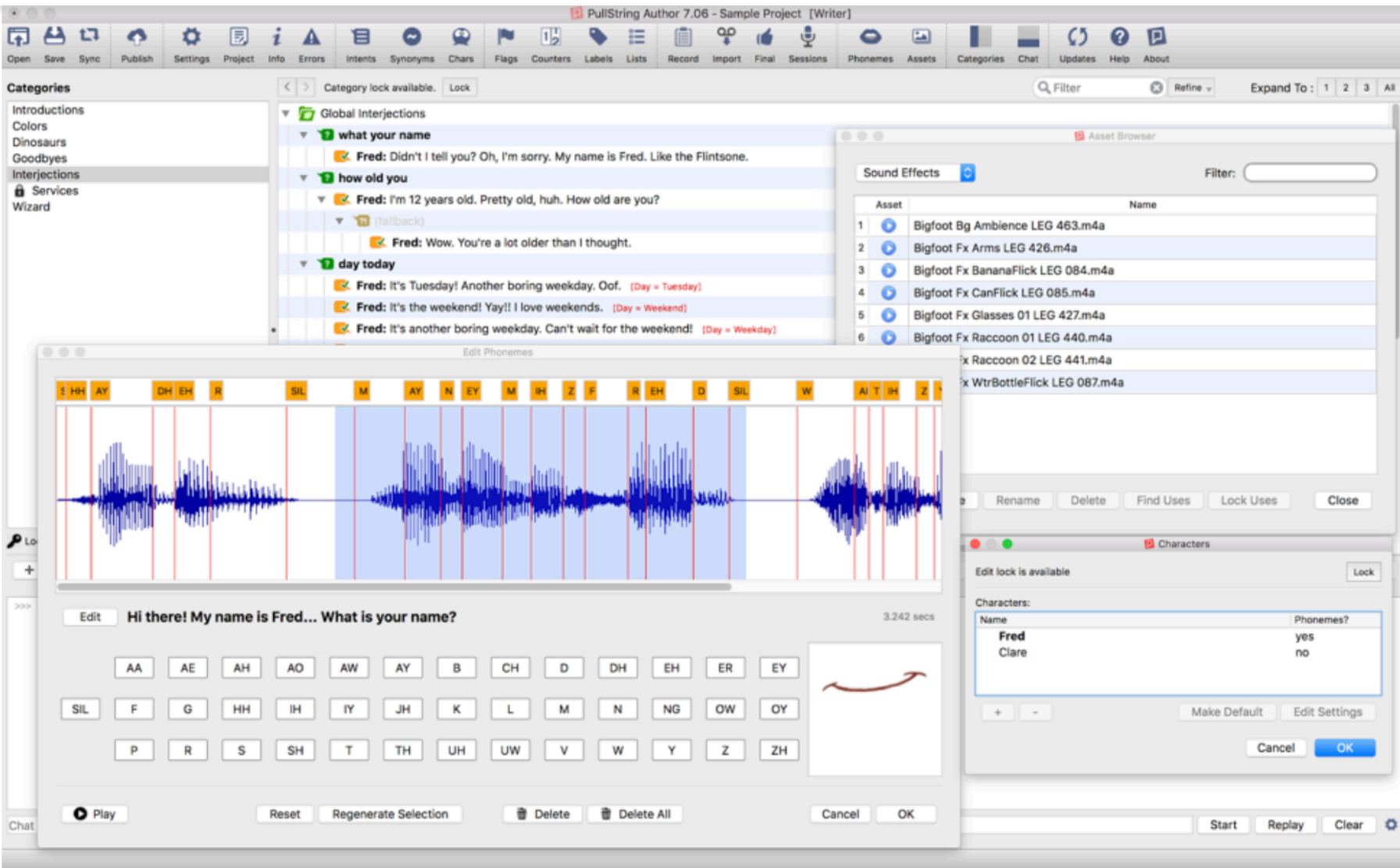
PullString

PullString Author as a Conversational IDE

The screenshot displays the PullString Author 7.06 - Assistant interface, which functions as a conversational IDE. The interface is divided into several sections:

- Categories:** A sidebar on the left lists various categories such as Introductions, Personality, Math, Conversations, Alarms, Weather, Jokes, Decisions, Abuse, News, Notes, Unknown, and Test. The 'Weather' category is currently selected.
- Global Interjections:** A central pane showing a list of intents and their associated actions. For example, the intent 'Intent: Ask About The Weather At Location => Location' is linked to a 'Web Service: Get The Weather' action. Other intents include 'Intent: Is It Raining' and 'Intent: Ask About The Weather Without A Location'.
- Chat Window:** A window at the bottom left showing a conversation between 'Jamie' and 'Martin'. The chat history includes:
 - Jamie: Hi there! My name is Jamie. What's your name?
 - Martin: ...
 - Jamie: It's great to meet you Martin. What can I do for you today?
 - Martin: > What's the weather like in San Francisco, CA?
- Code Editor:** A window on the right showing the JavaScript code for the 'news.js' file. The code is a REST client for the NY Times API, using the 'request' module to fetch the latest top stories. It includes a 'success' callback that parses the JSON response and a 'fail' callback.

The interface also features a top toolbar with icons for Open, Save, Publish, Settings, Project, Info, Errors, Intents, Synonyms, Chars, Flags, Counters, Labels, Lists, Final, Assets, Categories, Chat, Updates, Help, and About. A bottom toolbar includes buttons for Start, Replay, and Clear, along with a status bar showing 'Line 1, Column 1' and 'Spaces: 4 JavaScript'.



PullString Author 7.06 - Sample Project [Writer]

Categories

- Introductions
- Colors
- Dinosaurs
- Goodbyes
- Interjections
- Services
- Wizard

Global Interjections

- what your name**
 - Fred: Didn't I tell you? Oh, I'm sorry. My name is Fred. Like the Flintstone.
- how old you**
 - Fred: I'm 12 years old. Pretty old, huh. How old are you?
 - (fallback)
 - Fred: Wow. You're a lot older than I thought.
- day today**
 - Fred: It's Tuesday! Another boring weekday. Oof. [Day = Tuesday]
 - Fred: It's the weekend! Yay!! I love weekends. [Day = Weekend]
 - Fred: It's another boring weekday. Can't wait for the weekend! [Day = Weekday]

Asset Browser

Sound Effects

Asset	Name
1	Bigfoot Bg Ambience LEG 463.m4a
2	Bigfoot Fx Arms LEG 426.m4a
3	Bigfoot Fx BananaFlick LEG 084.m4a
4	Bigfoot Fx CanFlick LEG 085.m4a
5	Bigfoot Fx Glasses 01 LEG 427.m4a
6	Bigfoot Fx Raccoon 01 LEG 440.m4a
	Bigfoot Fx Raccoon 02 LEG 441.m4a
	Bigfoot Fx WtrBottleFlick LEG 087.m4a

Edit Phonemes

Hi there! My name is Fred... What is your name?

3.242 secs

AA AE AH AO AW AY B CH D DH EH ER EY
SIL F G HH IH IY JH K L M N NG OW OY
P R S SH T TH UH UW V W Y Z ZH

Characters

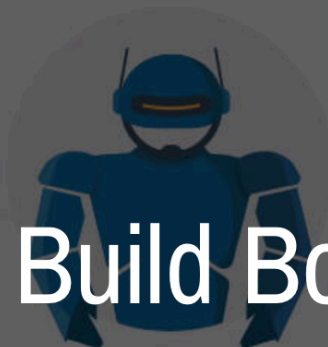
Edit lock is available

Name	Phonemes?
Fred	yes
Clare	no

Make Default Edit Settings

Cancel OK

PullString Author showing some of the features for building audio experiences



Build Bots for Google Home

Voice-enable your Bot

[Read Docs](#)[Get Started](#)

<https://www.gupshup.io/developer/googlehome>

gupshup

[Home](#)[Overview](#)[Products ▾](#)[Developers ▾](#)[Dashboard ▾](#)

- + IDE Bot Builder
- + New Flow Bot Builder
- + Flow Bot Builder
- + Template Bot Builder
- + Link your bot to Gupshup
- + Bot Scripting
- + Gupshup bot SDK
- + Serverless Webviews
- + Adding NLP
- Supported Platforms
 - + Facebook
 - + Slack
 - Google Assistant

How to build a bot on Google Assistant

Welcome to Actions on Google

Actions on Google is the platform for developers to extend the Google Assistant. Join this emerging ecosystem by developing actions to engage users on Google. [Learn more](#)

[Documentation](#) <> [Sample code](#)

Your projects with an Assistant app

+
Add/import project

Add project

Project name

Country/region ⓘ

United States ▾

CANCEL

CREATE PROJECT

Once created, click on the gear icon next to the 'Overview' button and hit 'Project Settings'. Copy the Project ID that is automatically generated by 'Actions on Google'.



Actions on Google

hello world ▾



Go to docs

Amazon EchoとGoogle Homeの 開発モデル／開発環境



Amazon EchoとGoogle Homeの 開発モデル／開発環境

1. Amazon Alexa の開発モデル
2. Google Homeの開発環境



Intent + Slot + Entity Model

**Built-in Intent
Library**

Intent Signature

Slot Type

System Slot Type

Custom Slot Type

Entity Resolution



Google Assistant SDK

The developer preview

2017/12/20

<https://developers.google.com/assistant/sdk/>
<https://github.com/googlesamples/assistant-sdk-python>

Google Dialogflow SDK

2017/10/10

<https://dialogflow.com/docs/getting-started/basics>
<https://github.com/dialogflow>

Google Action SDK

2017/8/11

<https://developers.google.com/actions/sdk/>
<https://developers.google.com/actions/smarthome/>
<https://github.com/actions-on-google>

Amazon Alexa の開発モデル

- **Intent Model**
- **Intent+Slot Model**
- **Intent+Slot+Entity Model**

Alexaの開発モデルの変化

- 以前、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。

2016年4月「パーソナル・アシスタント・システムの現在

-- Google Now, Cortana, Alexa, Dialog --」

<https://goo.gl/iArQVy>

- この間、Alexa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。



Intent Model



Intent + Slot Model

Slot Type



Intent + Slot Model

Slot Type

System Slot Type

Custom Slot Type

Entity Resolution



Intent + Slot + Entity Model

**Built-in Intent
Library**

Slot Type

System Slot Type

Custom Slot Type

Entity Resolution



Intent + Slot + Entity Model

**Built-in Intent
Library**

Intent Signature

Slot Type

System Slot Type

Custom Slot Type

Entity Resolution

Intent Model



Intent Model

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。
-

こんにちは
はい。こんにちは
おい
こら
なんかいつて
お元気ですか
お元気



HelloIntent

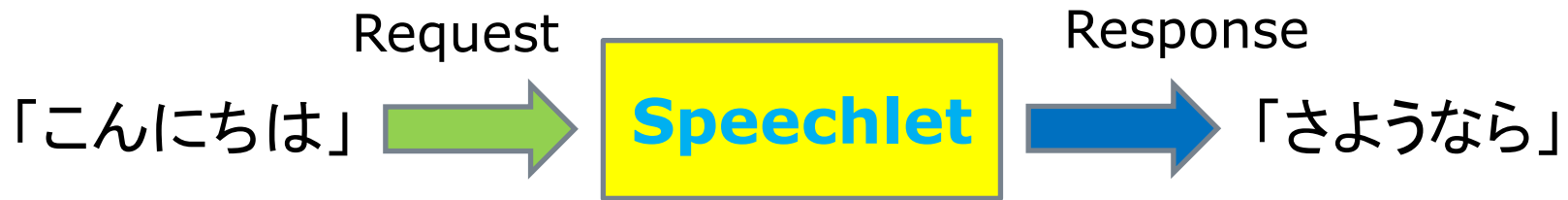
ヘルプ
助けて
どうすればいい
お願い



HelpIntent

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る。
 - 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。
-



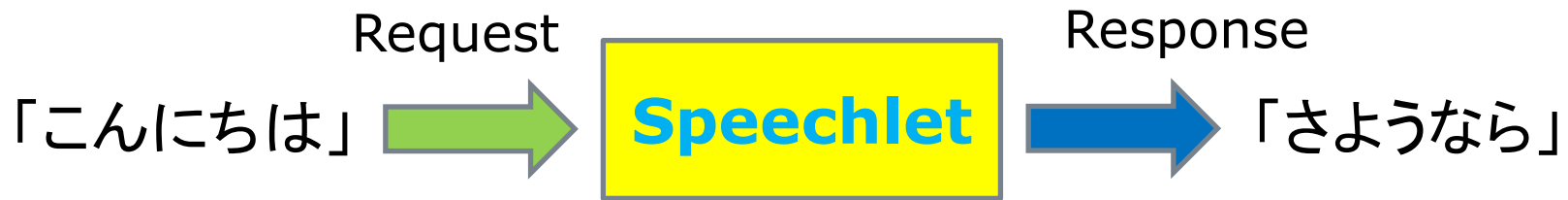
SampleUtterances.txt

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。

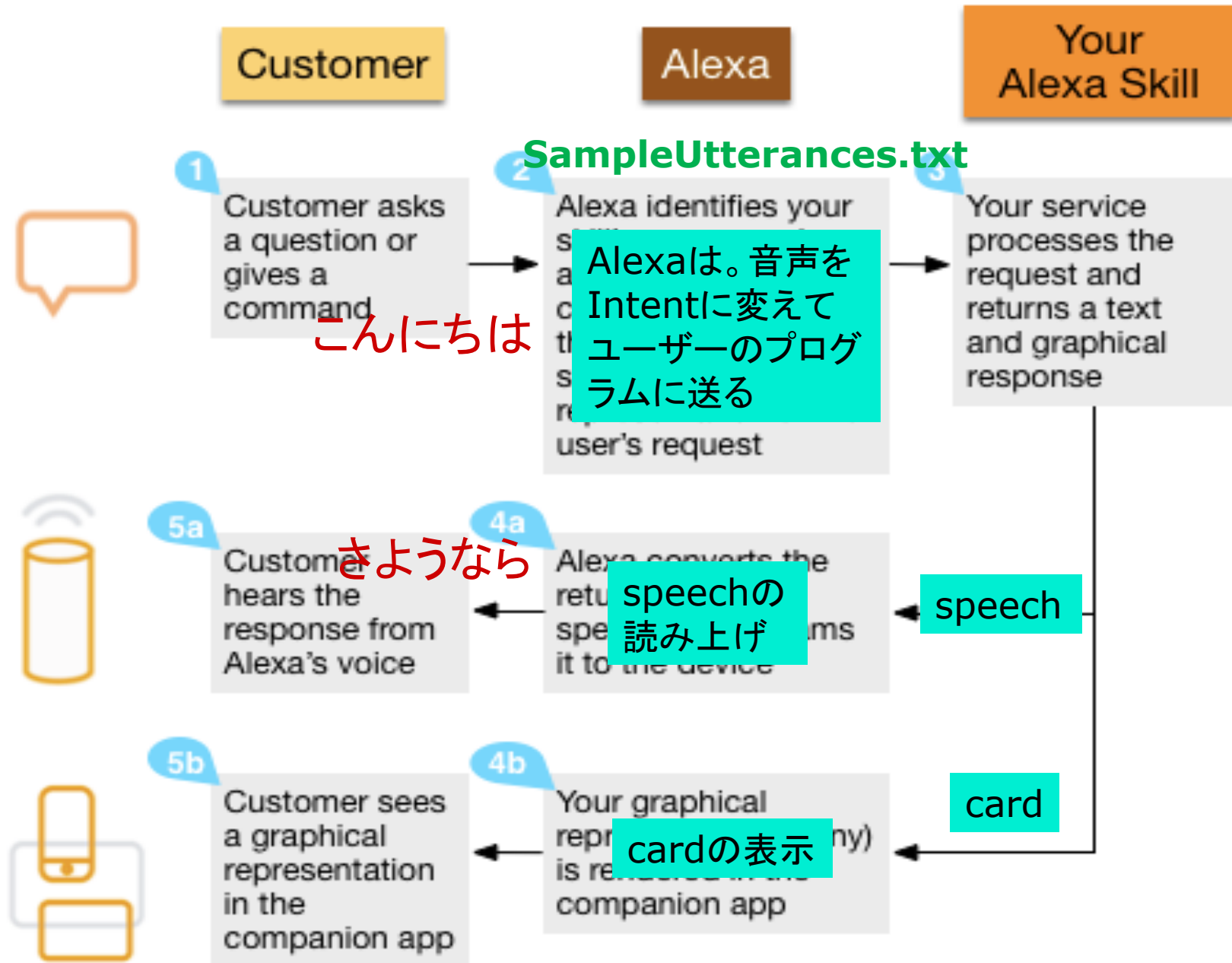
出力用のデータは、プログラム中にハードコードされる。



```
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 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"

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 + Slot Model



Intent + Slot Model
Slot Type

historybuff.speechAsset. **IntentSchema.json**

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

historybuff.speechAsset. **IntentSchema.json**

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

Intent + Slot Model



Intent + Slot Model

Slot Type

System Slot Type

Custom Slot Type

Entity Resolution

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。

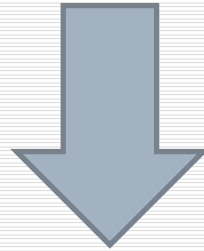
User: Alexa, ask *Plan My Trip* to plan a trip from Seattle to Portland on Friday.

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 {travelDate}

Alexa は、fromCity スロットに "Seattle"、toCity 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



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"
        }
      ]
    }
  ]
}
```

**System
Slot Type**



System Slot Type

Numbers, Dates,
and Times

Slot Type	Short Description	Supported Languages
AMAZON.DATE	Converts words that indicate <i>dates</i> ("today", "tomorrow", or "july") into a date format (such as "2015-07-00T9").	English (UK) English (US) German
AMAZON.DURATION	Converts words that indicate <i>durations</i> ("five minutes") into a numeric duration ("PT5M").	English (UK) English (US) German
AMAZON.FOUR_DIGIT_NUMBER	Converts words that represent <i>four-digit numbers</i> ("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

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

System
Slot Type

List Type

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)
```

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**

← **System slot type**

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"]
          }
        }
      ]
    }
  ]
}
```


Intent + Slot + Entity Model



Intent + Slot + Entity Model

Built-in Intent Library

Slot Type

System Slot Type

Custom Slot Type

Entity Resolution

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 & 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.*

一般的標準Intent(Slot なし)

Intent	発話の例
AMAZON.CancelIntent	キャンセル 取り消し やっぱりやめる
AMAZON.HelpIntent	ヘルプ どうすればいいの 使い方を教えて
AMAZON.LoopOffIntent	ループ再生をオフにして
AMAZON.LoopOnIntent	ループ再生して
AMAZON.NextIntent	次 ネクスト
AMAZON.NoIntent	いいえ ノー

Intent	発話の例
AMAZON.PauseIntent	一時停止
AMAZON.PreviousIntent	前 前へ
AMAZON.RepeatIntent	もう一度言って 繰り返して
AMAZON.ResumeIntent	再開
AMAZON.ShuffleOffIntent	シャッフル再生やめて
AMAZON.ShuffleOnIntent	シャッフル再生をオンにして
AMAZON.StartOverIntent	最初からやり直して 最初に戻って
AMAZON.StopIntent	ストップ 止めて 中止
AMAZON.YesIntent	はい オッケー いいよ

Intent + Slot + Entity Model



Intent + Slot + Entity Model

**Built-in Intent
Library**
Intent Signature

Slot Type
System Slot Type
Custom Slot Type
Entity Resolution

Built in Intentの例 – Actor Intents

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

Request:

```
{
  "request": {
    "type": "IntentRequest",
    "locale": "en-US",
    "intent": {
      "name": "AMAZON.SearchAction<object@Actor>",
      "slots": {
        "object.type": {
          "name": "object.type",
          "value": "actor"
        }
      }
    }
  }
}
```

Built in Intentの例 – Book Intents

```
{
  "intents": [
    {
      "intent": "AMAZON.ChooseAction<object@Book,sourceCollection@ReadingList>"
    },
    {
      "intent": "AMAZON.ChooseAction<object@Book>"
    }
  ]
}
```

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

“add all of
shakespeare’s
works to my
summer
reading list”

targetCollection
object.inLanguage
targetCollection
targetCollection
object.genre
object.partOfSeries
object.type
object.name
object.audience.
object.sort
object.author.name
targetCollection

Built in Intent の例 – Actor Intents

AMAZON.ChooseAction<object@Book,sourceCollection@ReadingList>	“select the last book on the reading list”	object.type object.select sourceCollection
AMAZON.ChooseAction<object@Book>	“choose moby dick”	object.type object.name object.select object.author.name
AMAZON.DeleteAction<object@Book,sourceCollection@ReadingList>	“remove all of the audiobooks from my favorites list”	object.owner.name object.version.title object.inLanguage object.select object.genre object.type object.name sourceCollection sourceCollection sourceCollection object.sort object.translator object.author.name

Built in Intentの例 – Actor Intents

AMAZON.PlaybackAction<object@Book>

“play the
start of lord
of the rings
book one”

```
object.owner.name  
object.select  
object.contentSo  
object.partOfSer  
object.type  
object.name  
object.bookNumbe  
object.brand.typ  
object.partOfBoo  
object.sort  
object.author.na
```

AMAZON.RateAction<object@Book>

“can you rate
cloud atlas
five stars”

```
rating.ratingVal  
object.select  
object.partOfSer  
object.type  
object.name  
rating.ratingVal  
object.partOfSer  
rating.bestRatin
```

Built in Intent の例 – Actor Intents

AMAZON.RateAction<object@Book>

“can you rate
cloud atlas
five stars”

rating.ratingVal
object.select
object.partOfSer
object.type
object.name
rating.ratingVal
object.partOfSer
rating.bestRating

AMAZON.RestartAction<object@Book>

“begin the
book again”

object.owner.name
object.type
object.name
object.select
object.author.name

AMAZON.ResumeAction<object@Book>

“continue
playing my
audiobook to
kill a
mockingbird”

object.owner.name
object.contentSo
object.contentSo
object.type
object.name
object.bookNumbe
object.brand.type

Built in Intent の例 – Actor Intents

AMAZON.SearchAction<object@Book>

“what books
are on my
audible”

object.owner.name
object.genre
object.contentSource
object.type
object.name
sourceCollection
object.sort
object.author.name
object.readBy.name

AMAZON.SearchAction<object@Book[datePublished]>

“alexa what
year was this
book
published”

object.datePublished
object.type

AMAZON.SearchAction<object@Book[numberOfSections]>

“what page
are we on in
the book”

object.numberOfSections
object.type
object.hasPart.type
object.numberOfSections

Built in Intent の例 – Actor Intents

AMAZON.SearchAction<object@Book[position]>

“where am i
in the book”

object.type

AMAZON.StopAction<object@Book>

“i don’t want
to read
matilda
anymore”

object.owner.name
object.type
object.name

AMAZON.SuspendAction<object@Book>

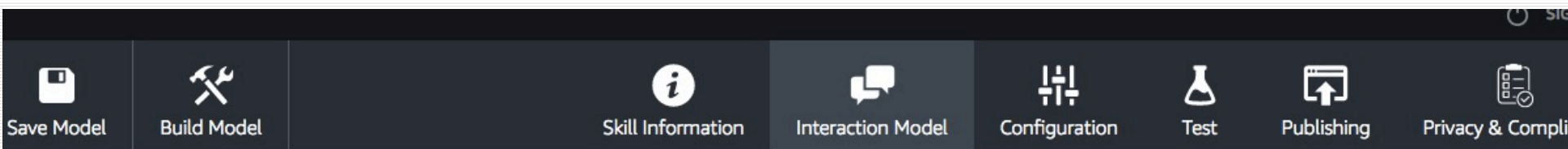
“pause this
audiobook
for one hour”

object.owner.name
object.type
duration.name

```
{
  "intents": [
    {
      "intent": "AMAZON.ChooseAction<object@Book,sourceCollection@ReadingList>"
    }
  ]
}
```

```
{
  "request": {
    "type": "IntentRequest",
    "locale": "en-US",
    "intent": {
      "name": "AMAZON.ChooseAction<object@Book,sourceCollection@ReadingList>",
      "slots": {
        "object.select": {
          "name": "object.select",
          "value": "last"
        },
        "object.type": {
          "name": "object.type",
          "value": "book"
        },
        "sourceCollection.type": {
          "name": "sourceCollection.type",
          "value": "reading list"
        }
      }
    }
  }
}
```

次のような、build-in Intentが用意されている



● Use an existing intent from the built-in library ?

Search built-in intents 

154 / 154 Built-ins

Add intents (0)

NAME		DESCRIPTION	
▶ 	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.	☐

▶		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 as 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 as 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.	☐

Google Home の開発環境

- **Google Assistant SDK**
- **Google Dialogflow SDK**
- **Google Action SDK**

Google Homeの開発環境

- ❑ Google Homeの大きな特徴は、基本的には、先行したGoogle Voice Search(Google Now, Google Assistant)の機能をそのまま引き継いでいることである。
- ❑ そのことを反映して、複数の開発環境が提供されている。

Google Assistant SDK

Google Dialogflow SDK

Google Action SDK

- ❑ Google Assistant SDKは、様々なハードウェア環境に、Google Assistantの機能を、そのまま埋め込むことを目的にしている。サーバとのやりとりは、ボイス(音声データ)をそのまま使う。必要な処理は、Google側が全て行う。
GitHubで公開されているソースは、機械によって自動生成されたものである。

Google Homeの開発環境

- Google Dialogflow SDKは、対話の記述を強化した Alexa 対抗版で、基本的には、Alexa と同じ Intent-Slot モデルを採用している。
 - Google Assistant SDK を使って作られたアプリに、「明日の天気は？」と話せば、何の設定なしでも、明日の天気についての情報が音声で返ってくる。ただし、Dialogflow で作られたモデルで、天気予報を聞くためには、agent の実装で、外部の天気予報サービスを呼び出す必要が有る。
 - Google Action SDK は、Smart Home などでの単純なボイス・コマンドの実装に向けられているように思う。
-

Google Voice Search

2002/05/20 ~

Google Now

2012/07/09 ~

Nest 買収 Smart Home

2014/01/13

Google Home 以前

Google Voice Search

2002/05/20 ~

Google Now

2012/07/09 ~

API.AI 買収

2016/09/01

Nest 買収

2014/01/13



Google Home

2016/05/17 発表

2016/11/04 販売

2016年:Google Home 発売



Google Assistant SDK

The developer preview (0.0.3)

2017/7/19

Google Assistant
library for Python

Google Assistant
gRPC API

Google Dialogflow SDK

2017/10/10

Google Action SDK

2017/8/11

2017年: Google Home の SDK

Google Assistant
Library

Google Assistant
Service

Google Assistant SDK

The developer preview

2017/12/20



Google Dialogflow SDK

2017/10/10

Google Action SDK

2017/8/11

2017年:Google Home の SDK



Google Assistant SDK

The developer preview

2017/12/20

Google Assistant
Library

Google Assistant
Service

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

Google Dialogflow SDK

2017/10/10

<https://dialogflow.com/docs/getting-started/basics>

Google Action SDK

2017/8/11

<https://developers.google.com/actions/sdk/>

<https://developers.google.com/actions/smarthome/>

基本的なドキュメント



Google Assistant SDK

The developer preview

2017/12/20

Google Assistant
Library

Google Assistant
Service

<https://developers.google.com/assistant/sdk/>
<https://github.com/googlesamples/assistant-sdk-python>

Google Dialogflow SDK

2017/10/10

<https://dialogflow.com/docs/getting-started/basics>
<https://github.com/dialogflow>

Google Action SDK

2017/8/11

<https://developers.google.com/actions/sdk/>
<https://developers.google.com/actions/smarthome/>
<https://github.com/actions-on-google>

GitHub



Google Assistant SDK

The developer preview

2017/12/20

<https://developers.google.com/assistant/sdk/>
<https://github.com/googlesamples/assistant-sdk-python>

Google Dialogflow SDK

2017/10/10

<https://dialogflow.com/docs/getting-started/basics>
<https://github.com/dialogflow>

Google Action SDK

2017/8/11

<https://developers.google.com/actions/sdk/>
<https://developers.google.com/actions/smarthome/>
<https://github.com/actions-on-google>

GitHub

Google Home に先行したもの 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) とい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。

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



EVENTS



"When is sunset?"
"When is Mother's Day this year?"

今日の日没は？



CONVERSIONS



"What's 16 ounces in pounds?"

16オンスは何パウンド？



SONGS



"What's this song?"

この歌は何の歌？



MY PACKAGES



"Where's my package?"

僕の荷物はどこ？



REMINDERS



"Remind me to call Jamie at 6PM."
"Remind me to buy paper towels when I'm at Target."

Jamieとの約束覚えておいて



CAMERA



"Take a picture."

写真撮って



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

LOCAL MOVIES

"What movies are playing tonight?"
"Where is Thor playing?"

今日の映画は？



EMAILS

"Email Jamie that the meeting was moved to 3."

予定が変わったとメールして



WEBSITES

"Go to wikipedia.com."

Wikipedia表示して



ENTERTAINMENT TRIVIA

"Who plays Carrie Mattheson on Homeland?"
"Who directed Titanic?"

Titanicの監督は？



FINANCE

"What's Google's stock price?"
"What is Twitter trading at?"

Googleの株価は？



CALENDAR EVENTS

"Create an event, dinner with Jamie at 8 PM."

8時にディナーの予定入れて



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

NUTRITION

"How many calories in an avocado?"
"Compare apples and oranges."

アボガドのカロリーは？

LOCAL BUSINESSES

"Where's the nearest ice cream shop?"

一番近いアイスクリーム屋は？



MY CONTACTS

"What's Jamie's number?"
"When is Dad's birthday?"

お父さんの誕生日は？

DIRECTIONS

"Directions to Rowdy Hall."
"Take me home."

家までナビして

SOCIAL

"Post to Twitter, just got a new phone!"

新しい携帯買ったと、
Twitterに投稿して



MUSIC

"Play The Rolling Stones."
"Play Hey Jude."

ローリングストーンの
曲かけて

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



TV SHOWS

"Listen to my TV."

テレビをつけて



FAMOUS PEOPLE

"Where was Albert Einstein born?"

"How old is Brad Pitt?"

アインシュタインは
どこで生まれたの？

NAVIGATION

"Navigate to St. Stephen's Green."

●●までナビして



TRAVEL INFO

"What are some attractions in New York City?"

ニューヨークで、なにか
イベントないかしら？



FAMOUS PEOPLE

"Where was Albert Einstein born?"

"How old is Brad Pitt?"

ブラッド・ピットは何歳？



DICTIONARY

"What does onomatopoeia mean?"

onomatopoelaって
どんな意味？

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！」
-

Google Nowの検索の特徴

- 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。
-

Google Nowの検索の問題

- Google NowのKnowledge 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。
- Knowledge 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 Assistant SDK

The developer preview **2017/12/20**

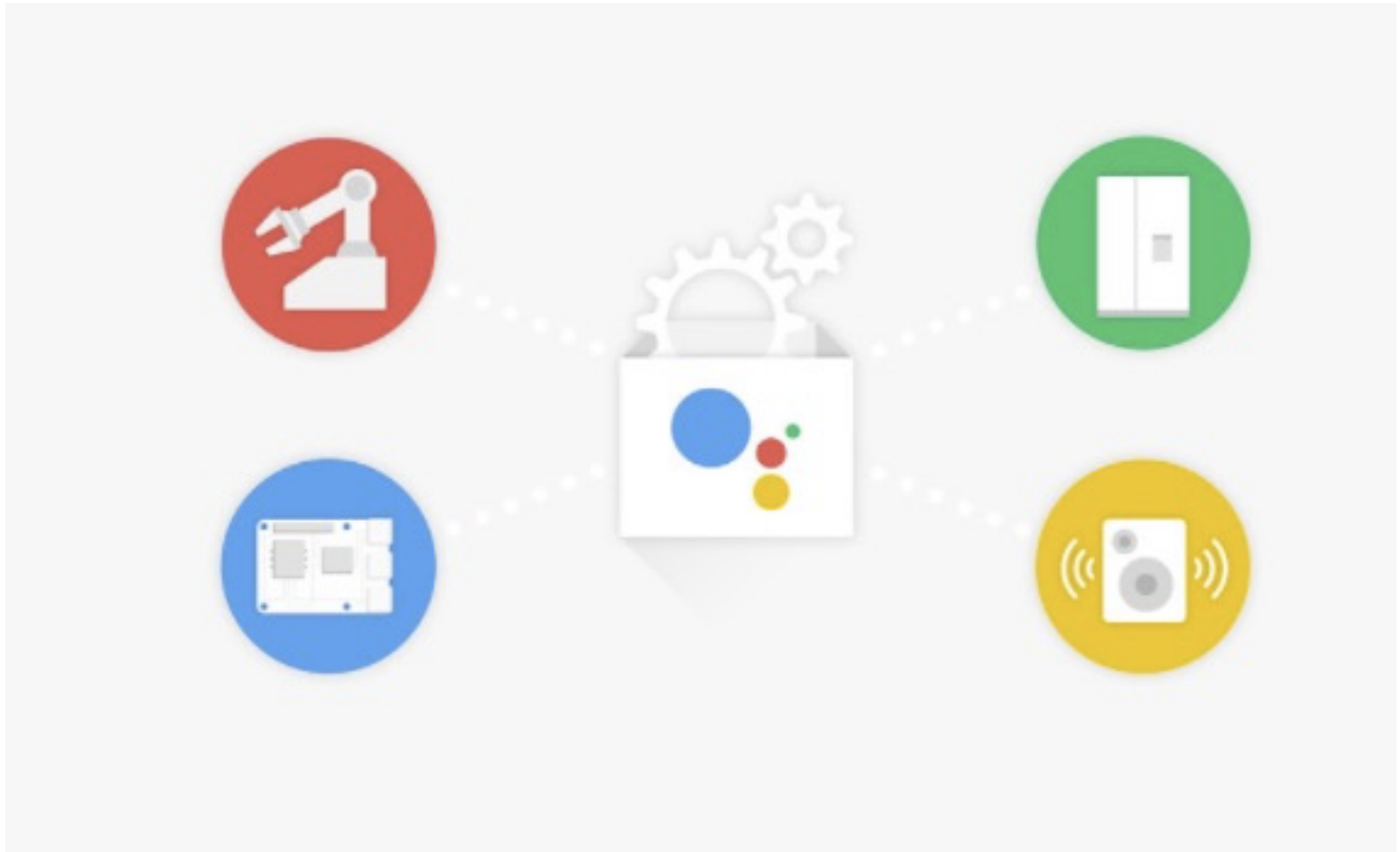
<https://developers.google.com/assistant/sdk/>

- Google Assistant Library
- Google Assistant Service

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



Overview

- The Google Assistant SDK lets you add hotword detection, voice control, natural language understanding and Google's smarts to your devices. Your device captures an utterance (a spoken audio request, such as *What's on my calendar?*), sends it to the Google Assistant, and receives a spoken audio response in addition to the raw text of the utterance.
-

SDK components

Google Assistant Library

- ❑ 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). Google Assistant Service
- ❑ 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

SDK components

Google Assistant Service

- The Google Assistant Service 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](#).
-

Steps to start building your prototypes

- ❑ The SDK lets you prototype devices quickly by embedding the Google Assistant into your device and then letting you add unique functionality with [Actions on Google](#):
- 1. Prototype your device [using the Python library](#), or in other platforms and programming languages [using the service](#).
- 2. Get ideas for [next steps to customize it](#).
- 3. Polish your device by reading best practices for [privacy and security](#) and [audio configuration](#).
- ❑ Finally, if you're looking to ship a commercial device with the Google Assistant, [contact us](#) to get started.

Developer preview and roadmap

- ❑ The Google Assistant SDK is currently in developer preview. It is free to use for development purposes only and has a limit imposed on the number of requests you can make per day.
 - ❑ We have many more features coming, including extended platform support, companion app integration, and improved ways to control your devices with voice. If you're interested in receiving updates about these features, [sign up here](#).
-

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-assistant-library](#): Python bindings for the [Google Assistant Library](#)
 - [google-assistant-grpc](#): autogenerated Python bindings to communicate with the [Google Assistant gRPC API](#)
 - [google-assistant-sdk](#): tools and python samples to help developers get started with the [Google Assistant SDK](#)
 - [googlesamples/assistant/grpc](#): reference sample for the [Google Assistant gRPC API](#)
 - [googlesamples/assistant/library](#): reference sample for the [Google Assistant library](#)
-

Compatibility and feature support

	Library	gRPC API
Supported architectures	linux-armv7l and linux-x86_64	gRPC platforms
Supported languages	Python	gRPC languages
Hands-free activation (<i>Ok Google</i>)	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

google.assistant.embedded.v1alpha1

EmbeddedAssistant

Service that implements Google Assistant API.

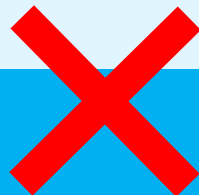
Converse

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.



NOTE: This version of the API has been deprecated. See [google.assistant.embedded.v1alpha2](#) for the latest supported version.



Request

Add to my shopping list

Response

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

オーディオ・データを送ると

"Add to my shopping list"

"What do you want to add?."

オーディオ・データが返る

google.assistant.embedded.v1alpha1 ← もう使えない

google.assistant SDKの「文盲」プロトコル

Request

bagels

Response

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

オーディオ・データを送ると

bagels

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

オーディオ・データが返る

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

google.assistant SDKの「文盲」プロトコル

google.assistant.embedded.v1alpha2

EmbeddedAssistant



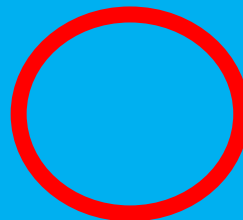
Service that implements the Google Assistant API.

Assist

`rpc Assist(AssistRequest) returns (AssistResponse)`

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.

Converse → **Assist**



Request

Add to my shopping list

Response

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:

- AssistRequest.config
- AssistRequest.audio_in
- AssistRequest.audio_in
- AssistRequest.audio_in
- AssistRequest.audio_in



オーディオ・データを送ると

"Add to my shopping list"

- AssistResponse.event_type.END_OF_UTTERANCE
- AssistResponse.speech_results.transcript "add to my shopping list"
- AssistResponse.dialog_state_out.microphone_mode.DIALOG_FOLLOW_ON
- AssistResponse.audio_out
- AssistResponse.audio_out
- AssistResponse.audio_out



"What do you want to add?."

オーディオ・データが返る

google.assistant.embedded.v1alpha2

google.assistant SDKの「文盲」プロトコル

Request

bagels

Response

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 **Assist** method, again with streamed requests and responses, such as:

- AssistRequest.config
- AssistRequest.audio_in
- AssistRequest.audio_in
- AssistRequest.audio_in
- AssistResponse.event_type.END_OF_UTTERANCE

オーディオ・データを送ると

bagels

- AssistResponse.dialog_state_out.microphone_mode.CLOSE_MICROPHONE
- AssistResponse.audio_out
- AssistResponse.audio_out
- AssistResponse.audio_out
- AssistResponse.audio_out

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

オーディオ・データが返る

google.assistant.embedded.v1alpha2

google.assistant SDKの「文盲」プロトコル

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.

AssistRequest

Fields

Union field **type**. Exactly one of these fields must be specified in each **AssistRequest**. **type** can be only one of the following:

config	AssistConfig The config message provides information to the recognizer that specifies how to process the request. The first AssistRequest message must contain a config message.
audio_in	bytes The audio data to be recognized. Sequential chunks of audio data are sent in sequential AssistRequest messages. The first AssistRequest message must not contain audio_in data and all subsequent AssistRequest 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 <i>Output-only</i> 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 <i>Output-only</i> Indicates the type of event.
audio_out	AudioOut <i>Output-only</i> The audio containing the assistant's response to the query.
result	ConverseResult <i>Output-only</i> The semantic result for the user's spoken query.

AssistResponse

Fields

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.

device_action

DeviceAction

Output-only Contains the action triggered by the query with the appropriate payloads and semantic parsing.

speech_results[]

SpeechRecognitionResult

Output-only This repeated list contains zero or more speech recognition results that correspond to consecutive portions of the audio currently being processed, starting with the portion corresponding to the earliest audio (and most stable portion) to the portion corresponding to the most recent audio. The strings can be concatenated to view the full in-progress response. When the speech recognition completes, this list will contain one item with **stability** of **1.0**.

dialog_state_out

DialogStateOut

Output-only Contains output related to the user's query.

Google Assistant SDK for Python

google-assistant-grpc: **autogenerated** Python bindings to communicate with the Google Assistant gRPC API

[assistant-sdk-python](#) / [google-assistant-grpc](#) / [google](#) / [assistant](#) / [embedded](#) /



proppey google-assistant-grpc: regenerate v1alpha2 proto ...

..

📁 [v1alpha1](#)

google-assistant-grpc: regenerate proto bindings

📁 [v1alpha2](#)

google-assistant-grpc: regenerate v1alpha2 proto

📄 [__init__.py](#)

simpler organization

assistant SDKの「ソース」は、自動生成されたもの

Google Dialogflow SDK

<https://dialogflow.com/docs/getting-started/basics>



Dialogflow

Building Your First Agent

<https://dialogflow.com/docs/getting-started/building-your-first-agent>



Project を作成

My Sample-1

CREATE

DEFAULT LANGUAGE ?

Japanese — ja 日本語対応 ▼

Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

(GMT+9:00) Asia/Tokyo ▼

Date and time requests are resolved using this timezone.

GOOGLE PROJECT

Create a new Google project ▼

Enables Cloud functions, Actions on Google and permissions management.

API VERSION



Dialogflow V2 API ^[beta]

Use [Dialogflow V2 API](#) as default for the agent. Your webhook will receive requests in the [new format](#).



Weather

Intent を作成

User saysの入力

SAVE



Contexts

User says

Search in user says



Add user expression



一月三日の東京の雨の確率

パラメーターとエンティティの自動抽出

PARAMETER NAME

ENTITY

RESOLVED VALUE

date

@sys.date

一月三日の



geo-city

@sys.geo-city

東京



稚内の明日の天気



今日の天気は



Weather

IntentとEntity

SAVE



Action



Enter action name

エンティティの指定



REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?
☐	date	@sys.date	\$date	☐
☐	geo-city	@sys.geo-city	\$geo-city	☐
☐	Enter name	Enter entity	Enter value	☐

+ New parameter

Response ?



DEFAULT





Weather

IntentとResponseの入力

SAVE



geo-city

@sys.geo-city

\$geo-city



Enter name

Enter entity

Enter value



+ New parameter

Response ?



DEFAULT



Text response



- 1 わかりません
- 2 天気のこと、僕知らないの
- 3 情報検索できないの
- 4 Enter a text response variant

ADD MESSAGE CONTENT

Request

```
{
  "conversationToken": "CiZDIzVhOG..." ,
  "debugLevel": 1,
  "inputType": "VOICE",
  "locale": "ja-JP",
  "mockLocation":
    {
      "city": "Mountain View",
      "coordinates": {
        "latitude": 37.421980615353675,
        "longitude": -122.08419799804688
      },
      "formattedAddress":
        "Googleplex, Mountain View, CA 94043, United States",
      "zipCode": "94043"
    },
  "query": "稚内の明日の天気は",
  "surface": "PHONE"
}
```

Response

```
{
  "audioResponse": "//NAXAAUcN...",
  "conversationToken": "CiZDIzVhOG...",
  "expectUserResponse": true,
  "response": "天気のこと、僕知らないの",
  "visualResponse": {
    "agentLogoUrl": "https://www.gstatic.com/voice/opa/
                      partner_icons/generic_3p_avatar.png",
    "visualElements": [
      {
        "displayText": {
          "content": "天気のこと、僕知らないの"
        }
      }
    ]
  }
}
```

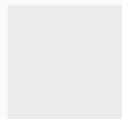




my test app



わかりました。テストバージョンのテスト用アプリです。

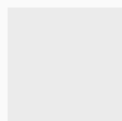


こんにちは！

今日の天気は



わかりません

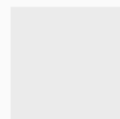


わかりません



my test app

今日の天気は



わかりません

稚内の明日の天気は



天気のこと、僕知らないの



天気のこと、僕知らないの

Dialogflow 発音も良く無い気が

SSML (Speech Synthesis Markup Language)

<speak>

Here are <say-as interpret-as="characters">SSML</say-as> samples.

I can pause <break time="3s"/>.

I can play a sound

<audio src="https://www.example.com/MY_MP3_FILE.mp3">

didn't get your MP3 audio file</audio>.

I can speak in cardinals.

Your number is <say-as interpret-as="cardinal">10</say-as>.

Or I can speak in ordinals. You are <say-as interpret-as="ordinal">10</say-as> in line.

Or I can even speak in digits. The digits for ten are

<say-as interpret-as="characters">10</say-as>.

I can also substitute phrases, like the

_{W3C}.

Finally, I can speak a paragraph with two sentences.

<p><s>This is sentence one.</s>

<s>This is sentence two.</s></p>

</speak>



SSML (Speech Synthesis Markup Language)

< speak >

こんにちは

< audio

src="https://actions.google.com/sounds/v1/cartoon/cartoon_boing.ogg"/>

SSML を使うと、

< prosody rate="fast" >話す速度を速くすることができます。</prosody>

< prosody rate="slow" >遅くすることもできま</prosody>

< prosody pitch="high" >音声を高音にしたり</prosody>

< prosody pitch="low" >低音にしたりすることもできます</prosody>

< prosody volume="loud" >音量を上げること</prosody>

< prosody volume="soft" >下げることも可能です。</prosody>

< prosody rate="fast" pitch="low" volume="x-loud" >

この 3 つを同時に設定することもできます。</prosody>

< break/>

</ speak >



IAM と管理

+ IAM

- ID
- 割り当て
- サービス アカウント
- ラベル
- GCP のプライバシーとセ...
- 設定
- 暗号鍵
- Identity-Aware Proxy
- 役割
- リソースの管理

IAM

+ 追加

- 削除

プロジェクト「MySample1」の権限

以下の権限は、「MySample1」プロジェクト全体とそのすべてのリソースに影響します。権限を付与するには、メンバーを追加して、その役割を選択します。メンバーにはユーザー、ドメイン、グループ、サービス アカウントを指定できます。

ベータ版で使用されている役割の一部は、今後変更される場合やサポート終了になる場合があります。[詳細](#)

🔍 名前または役割でフィルタ

表示: メンバー

タイプ	メンバー	役割
☒	 Dialogflow Integrations dialogflow-riqsd1@mysample1-9d3ec.iam.gserviceaccount.com	Dialogflow API Client
☐	 fujio.maruyama@gmail.com fujio.maruyama@gmail.com	オーナー
☐	 App Engine default service account mysample1-9d3ec@appspot.gserviceaccount.com	編集者
☐	 service-867310869975@firebase-rules.iam.gserviceaccount.com	Firebase ルールシステム

Getting started with Dialogflow fulfillment

<https://dialogflow.com/docs/how-tos/getting-started-fulfillment>

Google Cloud Platform に Google Cloud Functions API のアプリケーションを登録

Google Cloud Platform では、アプリケーションの管理と API 使用のモニタリングができます。

アプリケーションを登録するプロジェクトの選択

1つのプロジェクトですべてのアプリケーションを管理できます。また、アプリケーションごとに異なるプロジェクトを作成することもできます。

MySample1 ▾

続行

API は有効になっています

Google Cloud Functions API が有効化されました。

続行

Enabling Cloud Functions for Firebase

Inline Editor (Powered by Cloud Functions for Firebase)

ENABLED



Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

index.js package.json



```
1 'use strict';
2
3 const functions = require('firebase-functions'); // Cloud Functions for Firebase library
4 const DialogflowApp = require('actions-on-google').DialogflowApp; // Google Assistant helper library
5
6 const googleAssistantRequest = 'google'; // Constant to identify Google Assistant requests
7
8 exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
9   console.log('Request headers: ' + JSON.stringify(request.headers));
10  console.log('Request body: ' + JSON.stringify(request.body));
11
12  // An action is a string used to identify what needs to be done in fulfillment
13  let action = request.body.result.action; // https://dialogflow.com/docs/actions-and-parameters
14
15  // Parameters are any entities that Dialogflow has extracted from the request.
16  const parameters = request.body.result.parameters; // https://dialogflow.com/docs/actions-and-parameters
17
18  // Contexts are objects used to track and store conversation state
19  const inputContexts = request.body.result.contexts; // https://dialogflow.com/docs/context
20
21  // Get the request source (Google Assistant, Slack, API, etc) and initialize DialogflowApp
22  const requestSource = (request.body.originalRequest) ? request.body.originalRequest.source : undefined;
23  const app = new DialogflowApp({request: request, response: response});
24
25  // Create handlers for Dialogflow actions as well as a 'default' handler
26  const actionHandlers = {
27    // The default welcome intent has been matched, welcome the user (https://dialogflow.com/docs/events#default_welcome)
28    'input.welcome': () => {
29      // Use the Actions on Google lib to respond to Google requests; for other requests use JSON
30      if (requestSource === googleAssistantRequest) {
31        sendGoogleResponse('Hello, Welcome to my Dialogflow agent!'); // Send simple response to user
32      } else {
33        sendResponse('Hello, Welcome to my Dialogflow agent!'); // Send simple response to user
34      }
35    },
36    // The default fallback intent has been matched, try to recover (https://dialogflow.com/docs/intents#fallback_intent)
37    'input.unknown': () => {
38      // Use the Actions on Google lib to respond to Google requests; for other requests use JSON
39      if (requestSource === googleAssistantRequest) {
```

[View execution logs in the Firebase console](#)

Last deployed on 10/01/2010 9:41

DEPLOY

```
// Get the city and date from the request
let city = req.body.result.parameters['geo-city'];
// city is a required param
// Get the date for the weather forecast (if present)
let date = "";
if (req.body.result.parameters['date']) {
  date = req.body.result.parameters['date'];
  console.log('Date: ' + date);
}
// Call the weather API
callWeatherApi(city, date).then((output) => {
  // Return the results of the weather API to Dialogflow
  res.setHeader('Content-Type', 'application/json');
  res.send(JSON.stringify({ 'speech': output, 'displayText':
output }));
}).catch((error) => {
  // If there is an error let the user know
  res.setHeader('Content-Type', 'application/json');
  res.send(JSON.stringify({ 'speech': error, 'displayText':
error }));
});
};
```

Weather IntentのFulfillmentの
Cloud Functionのソース

```
function callWeatherApi (city, date) {
  return new Promise((resolve, reject) => {
    // Create the path for the HTTP request to get the weather
    let path =
'/premium/v1/weather.ashx?format=json&num_of_days=1' +
    '&q=' + encodeURIComponent(city) + '&key=' + wwoApiKey +
    '&date=' + date;
    console.log('API Request: ' + host + path);
    // Make the HTTP request to get the weather
    http.get({host: host, path: path}, (res) => {
      let body = ''; // var to store the response chunks
      res.on('data', (d) => { body += d; }); // store each response
      chunk
      res.on('end', () => {
        // After all the data has been received parse the JSON for desired data
        let response = JSON.parse(body);
        let forecast = response['data']['weather'][0];
        let location = response['data']['request'][0];
        let conditions = response['data']['current_condition'][0];
        let currentConditions = conditions['weatherDesc'][0]['value'];
      })
    })
  })
}
```

function callWeatherApi (city, date) の続き

```
// Create response
let output = `Current conditions in the ${location['type']}
${location['query']} are ${currentConditions} with a
projected high of
  ${forecast['maxtempC']}° C or ${forecast['maxtempF']}° F
and a low of
  ${forecast['mintempC']}° C or ${forecast['mintempF']}° F
on
  ${forecast['date']}.`;
// Resolve the promise with the output text
console.log(output);
resolve(output);
});
res.on('error', (error) => {
  reject(error);
});
});
});
```

結局、外部の「お天気サービス」を、
利用している。

Basic Fulfillment and Conversation Setup

<https://dialogflow.com/docs/getting-started/basic-fulfillment-conversation>

Enable Webhook in Dialogflow

Fulfillment

SAVE

Webhook

Your web service will receive a POST request from API.AI in the [form of the response](#) to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#).

[Webhook example](#)

ENABLED



URL*

<https://us-central1-my-first-agent.cloudfunctions.net/helloHttp>

BASIC AUTH

HEADERS



Add header

DOMAINS



Enable Fulfillment in Intent

Response ?



Text response



- 1 Sorry I don't know the weather.
- 2 I'm not sure about the weather on \$date
- 3 I don't know the weather for \$date in \$geo-city but I hope its nice!
- 4 Enter a text response variant...

ADD MESSAGE CONTENT

Fulfillment



Use webhook



Use webhook for slot-filling

Action on Google

<https://developers.google.com/actions/extending-the-assistant>

Extending the Google Assistant

- ❑ The **Google Assistant** gives users their own personal Google that helps them find, organize, and get things done in their world. The main way users interact with the Google Assistant is by carrying on a conversation with it.
 - ❑ **Apps** extend the Google Assistant by letting you build **actions** that let users get things done with your products and services.
-

Extending the Google Assistant

- Unlike with traditional mobile or desktop apps, users interact with apps for the Assistant through a conversation, natural-sounding back and forth exchanges, and not traditional, computer-centric paradigms.
 - When you build an app for the Assistant, you design your conversations for a variety of surfaces, such as a voice-centric conversation for voice-activated speakers or a visual conversation on an Android phone. This lets users get things done quickly the way they want, either with voice or display affordances.
-

How Actions on Google works with the Assistant

When users request an action to be fulfilled:

1. The Assistant asks Actions on Google to invoke the best app to fulfill the intent.
 2. Actions on Google sends a request to the app's fulfillment and receives a response that it hands off to the Assistant.
 3. The Assistant renders the response in its UI and displays it to the user. The conversation between the user and the app begins.
 4. The Assistant sends subsequent user input directly to the app and the app responds directly to the Assistant. This conversation continues until the app gathers the user input it needs to fulfill the intent.
-

Sample

- ❑ A user talks to the Assistant and requests an action, "I want to learn about numbers". The Assistant asks Actions on Google to find the best app to handle the action and Actions on Google returns an app named "Geeknum".
 - ❑ The Assistant asks the user if she wants to invoke the Geeknum app and she says "Yes". The Assistant introduces Geeknum, then hands off the user to the Geeknum app.
 - ❑ The app displays a welcome message and the conversation begins with the user.
-

Sample

- During the conversation, the Assistant brokers the conversation between the user and Geeknum, and the two exchange information so Geeknum can gather the information it needs to fulfill the intent.
 - Finally, the conversation ends when the user is done learning about numbers.
-

Choosing a development option

Templates

- ❑ Building apps with templates lets you utilize a standard user experience that's defined by the template. You can customize portions of the app such as certain responses or configuration settings, but most of the app's conversation and fulfillment is handled by the template, letting you build apps without writing a single line of code!
 - ❑ We highly recommend using one of these templates if there is one available to fit your needs. This allows you to build apps quickly without worrying about designing conversations, which is one of the harder parts of development.
-

Choosing a development option

Dialogflow

- ❑ Use Dialogflow for most use cases where you need to design and build your own conversational experience. Dialogflow wraps the functionality of the Actions SDK into an easy-to-use web IDE that has conveniences such as generating and deploys action packages for you. It also includes an NLU (natural language understanding) engine that parses natural, human language, so you don't have to.
-

Choosing a development option

Action SDK

- ❑ Use this option if you have simple actions that have very short conversations with limited user input variability. These type of actions typically don't require robust language understanding and typically accomplish one quick use case.
- ❑ In addition, if you already have an NLU that you want to use and just want to receive raw text and pass it to your own NLU, you will also need to use the Actions SDK.
- ❑ Finally, the Actions SDK doesn't provide the conveniences of an IDE, so you have to manually create action packages with a text editor and deploy them to your Google Developer project with a command-line utility.

ボイス・アシスタントから見る AIの課題



ボイス・アシスタントから見る AIの課題

1. ボイス・アシスタントは「知識」を持つか？
2. 「知識」の検索へ -- 21世紀の大規模データ処理の基本的な流れを振り返る
3. Schema.orgとEntityモデルの問題
4. 知識データベース構築の試み
5. ボイス・アシスタントは「文」の意味を理解しているか？

ボイス・アシスタントは、
「知識」を持つか？

Virtual Assistant Jeopardy

Alexa と Google Now

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
デージーを演じ
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

Crtana と Google Now

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
デージーを演じ
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."

「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。

「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インの理由は？」 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インの理由は？



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・ウイ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、DuckDuckGo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>

「知識」の検索へ

21世紀の大規模データ処理の 基本的な流れを振り返る

2000年代

大規模データ処理の第一世代の開始

- 現代のITの中核的な技術の一つは、大規模分散システムによる大規模データの処理技術である。「世界中の情報を検索可能にする」というミッションを掲げたGoogleの、Webスケールの検索技術の実現がこうした時代の幕を開いた。21世紀の始まりとその時期は、ほぼ等しい。
 - クローリングで収集した膨大なページへのインデックス付けと**PageRank**の計算が、Webスケールのデータ処理の中心だったのだが、そうした処理は、**MapReduce**を用いたバッチ処理で行われていた。
-

2010年代 大規模データ処理の第二世代への転化

- モバイル・デバイスSNSの爆発的な普及を背景として、2010年ごろ、バッチ処理からリアルタイム処理への大規模データ処理の大きな転換が始まる。
 - Googleのシステムの転換については、2013年7月の丸山の「大規模分散システムの現在--- GFS, MapReduce, BigTableは、どう進化したか」を参照されたい。
<https://drive.google.com/file/d/0B04ol8GVySUueWQ2dkZUSFFETIk/edit?usp=sharing>
-



**He who controls the graph,
controls the world.**

ソーシャル・グラフと 大規模グラフデータ処理

- こうした変化の口火を切ったのは、個人とモバイルをターゲットとして、「世界をもっとつながったものに」を会社のミッションとするFacebookの躍進だった。2010年4月に公開されたFacebookの **Open Graph** は、「ソーシャル・グラフ＝大規模グラフデータ」処理の重要性に多くの人の目を向けさせた。
- Googleの対応も興味深い。2010年6月、前述の検索インデキシングのリアルタイム化を実現した **Caffein** の投入と同じ日に、大規模グラフデータ処理のエンジン **Pregel** を発表する。

検索と大規模グラフデータ

- 2010年に行われた技術的転換は、バッチからリアルタイムへの進化だけではなく、データの量への注目からデータの質的なグラフ構造への注目を特質とする大規模データ処理の第二世代の開始を意味するものである。
- ただ、この技術的転換の意味を、検索の分野で「知識グラフ」の探索として、明確に定式化したのは、2012年のGoogleの**Knowledge Graph**である。この動きに、Facebookは**Search Graph**で、Microsoftは**Satori/Trinity**で、ただちに追走を開始する。

How Google and Microsoft taught search to “understand” the Web <https://goo.gl/TTrPdJ>

グラフデータと「知識」の探求の始まり

- GoogleのKnowledge Graphについては、
2012/07/12 の丸山の講演「Google の新しい検索技術
Knowledge Graph について」を参照されたい。
<https://docs.google.com/file/d/0B04ol8GVySUuUFUtbWcxNFIHY3c/edit?usp=sharing>
 - ただ、大規模なグラフデータから、我々にとって有用な「知識」「意味」を抽出しようという取り組みは、まだ始まったばかりである。
-

グラフ検索とEntity ModelとSchema.org

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。

「知識グラフ」と開かれた課題

- 「知識グラフ」に対する関心は、次のようなより長期的な課題の実現と結びついている。
 - 自然言語を用いたインターフェース
 - 知識とそれをハンドルする能力の機械への移転
 - 生物の多様な認知能力と学習能力の理解
 - 人間の言語能力の理解
 - 学習する機械
- 楽観的に語るならば、我々は、こうした課題を解決する時代の入り口に、さしかかっているのかもしれない。

ただし、現状では、それらにアクセスする権限は、大きく制限されている

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は、遠いところにいる。ただ、我々は、少しずつ、それに近づくことはできる。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。」

Larry Page on AI

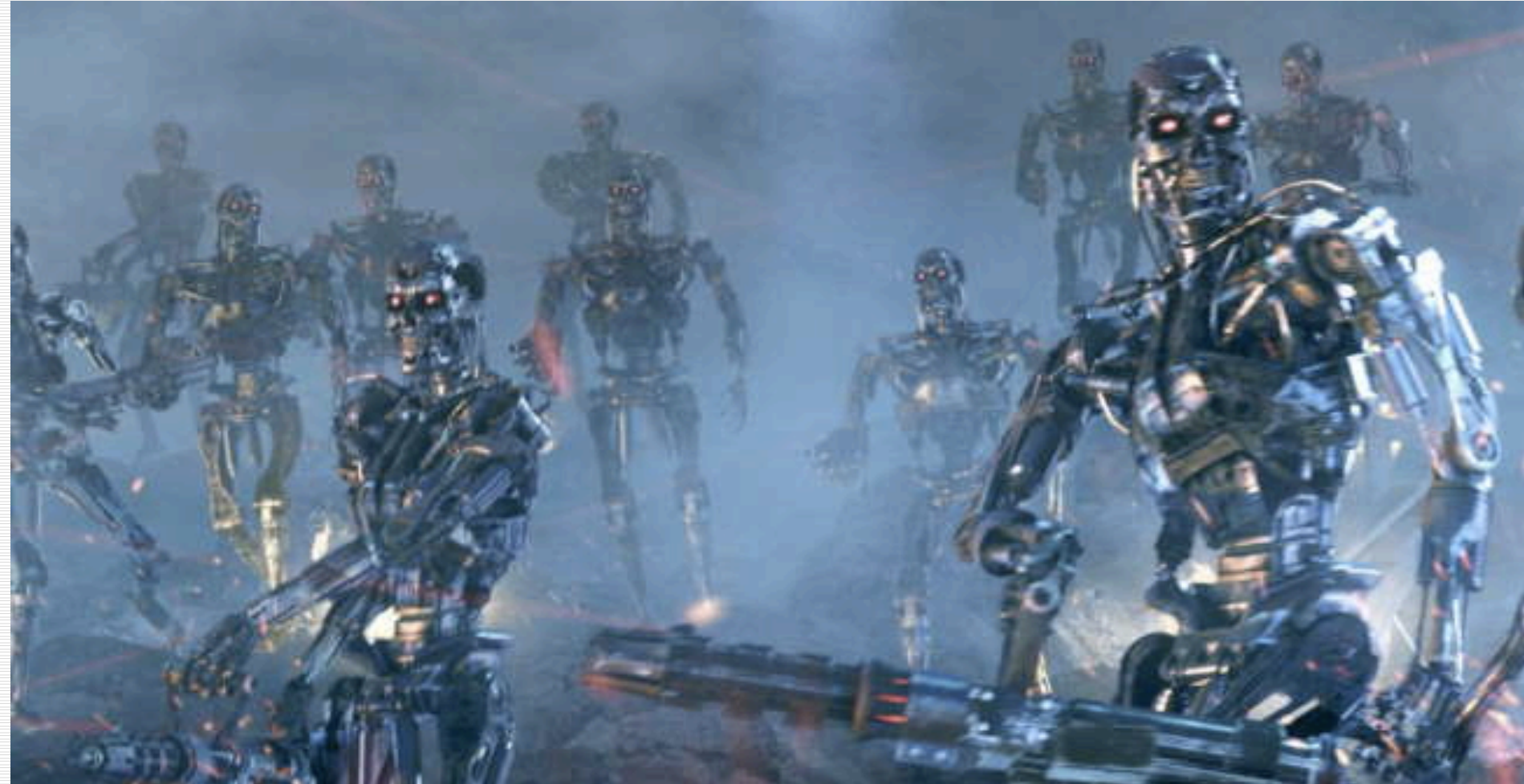
- 「検索における我々の大きな目標は、人が望むものを、実際に正確に理解し、世界のすべてのものを理解することである。コンピューター科学者として、我々は、それを人工知能と呼ぶ。」

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



<http://goo.gl/LdHPZA>

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



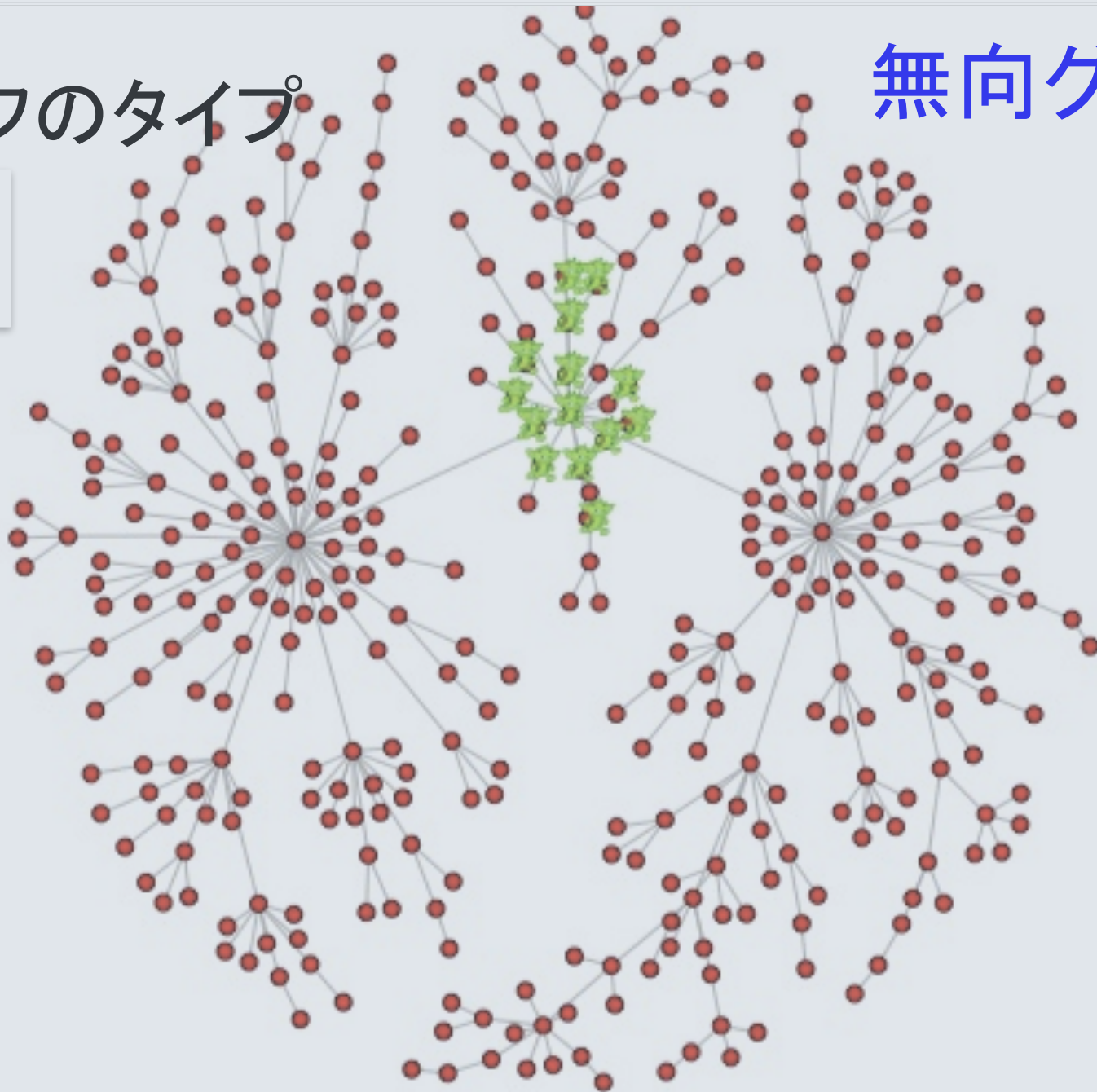
知識のグラフによる表現 グラフ・データベース

グラフには、辺の向きやラベルの有無によって、幾つかのタイプにわけられる。現在のグラフ・データベースの多くは、「プロパティ・グラフ・モデル」を採用している。

以下は、2014年2月のマルレク「大規模グラフデータ処理」より
<https://goo.gl/eijsbr>

グラフのタイプ

無向グラフ

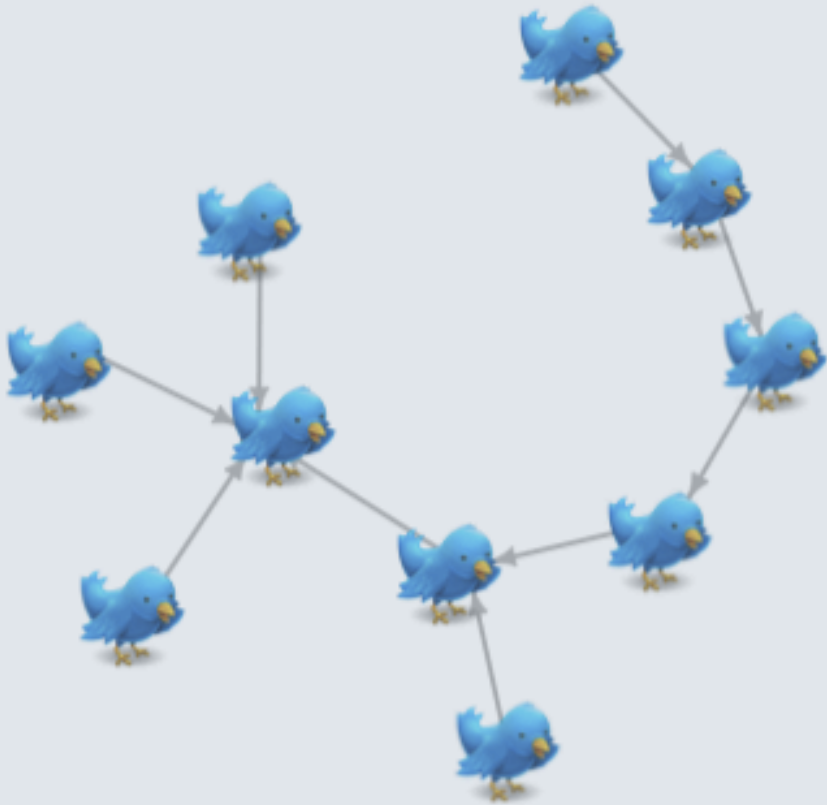


グラフのタイプ

有向グラフ UMLダイアグラム

グラフのタイプ

有向グラフ



TwitterのFollow

Webページのリンク

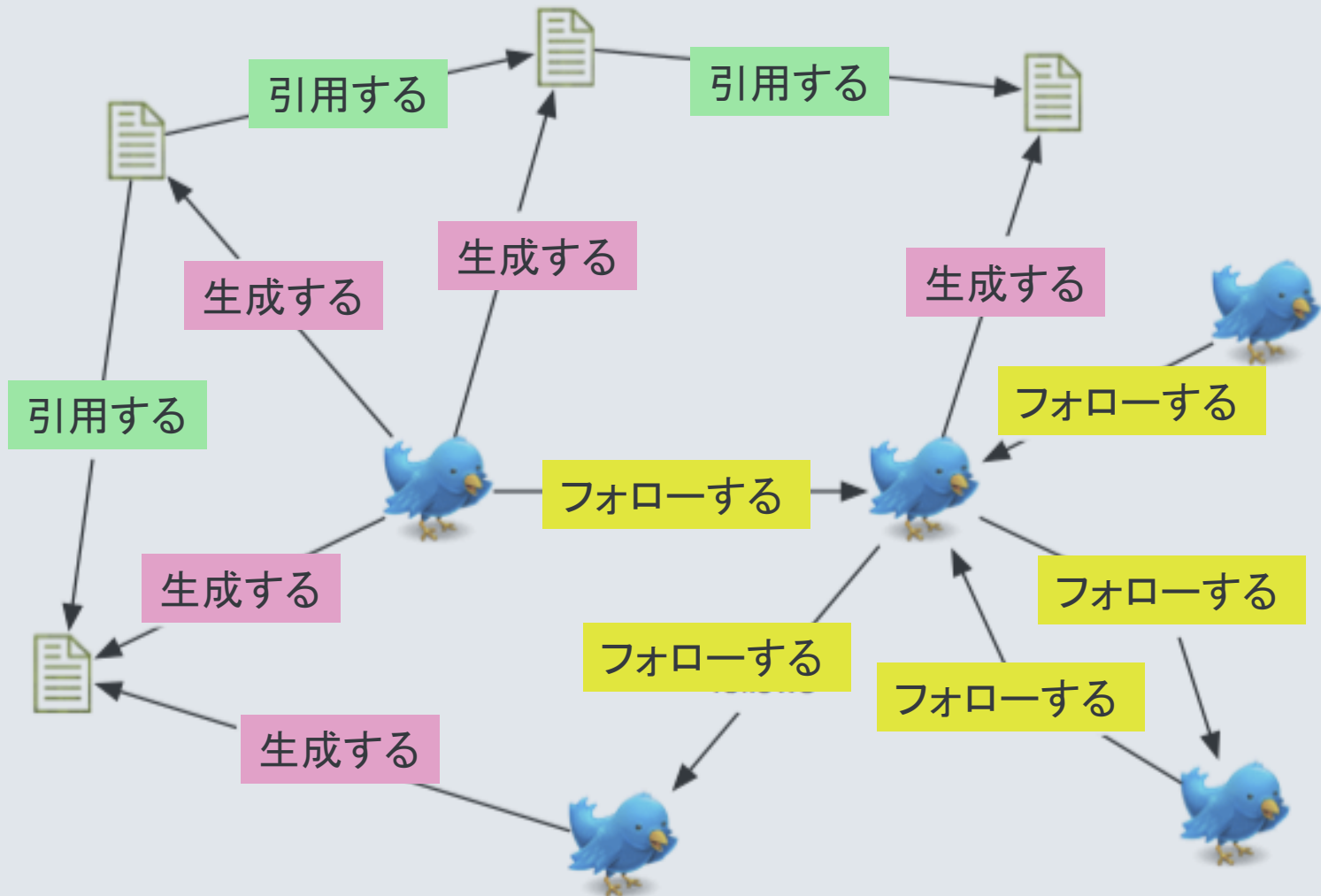
全ての辺が同じ意味を持つ 単一関係グラフ

- 辺を区別する方法がなく、全ての辺が同じ意味・型を持っている。こうした構造は、「単一関係グラフ (single-relational graphs)」と呼ばれる。
- 単一関係グラフは、グラフ理論やネットワーク科学で、おそらくは、もっともよく使われているグラフの型である。

複数関係グラフ

- 複数関係グラフは、辺に、例えば、「フォローする」とか「引用する」といった、明確な型付けを許す。
 - 辺にラベルを付ける事によって、辺は異なった意味を持ち、頂点も異なった型を持つようになる。
 - 例
 - follows : user → user
 - created : user → webpage
 - cites : webpage → webpage
-

複数関係グラフによって、 グラフの表現力は拡大される



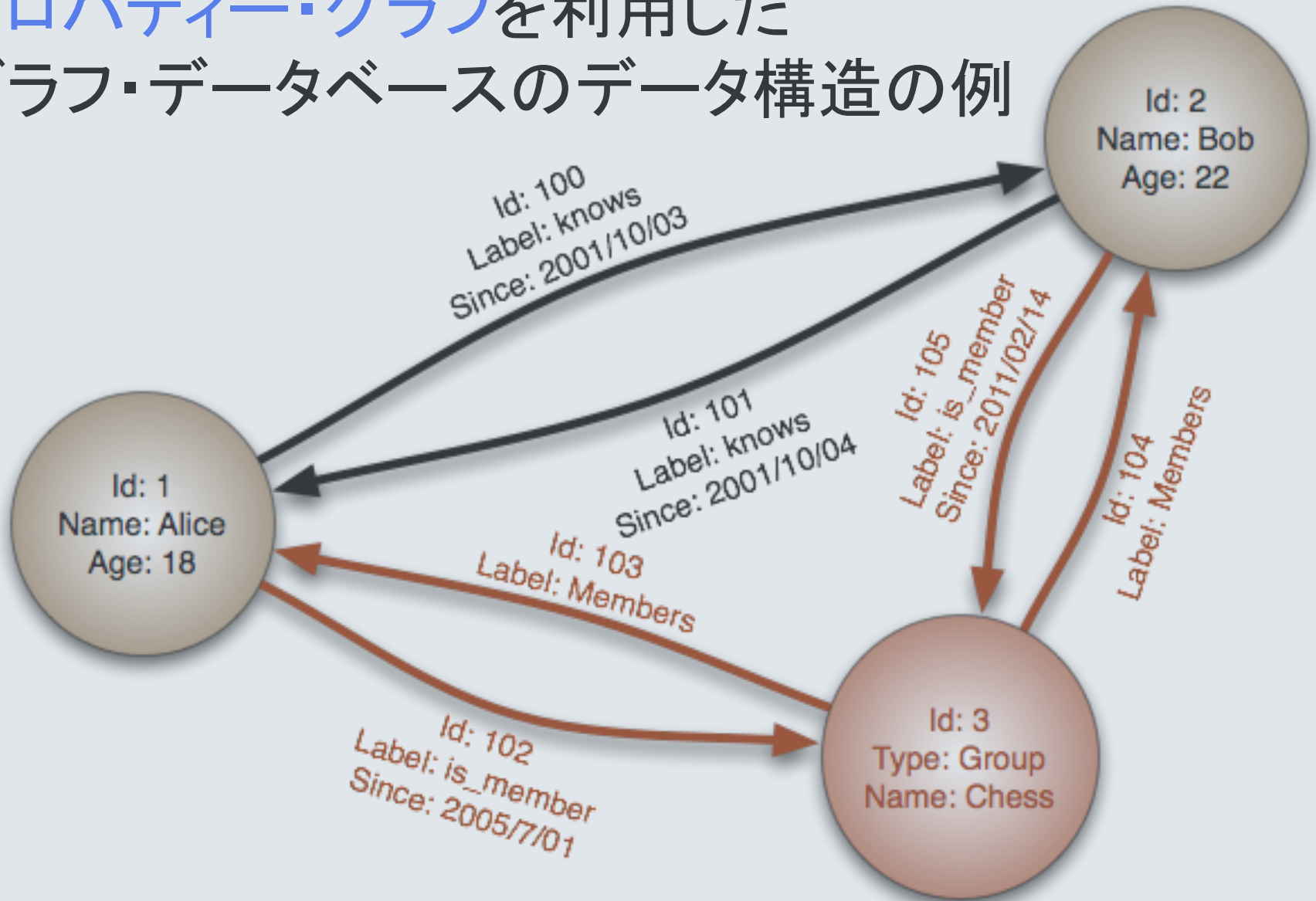
プロパティ・グラフの柔軟性

- プロパティ・グラフは、複数関係グラフを、頂点・辺の両方とも Key/Valueのプロパティ・マップを保持出来るように拡張したものである。この事によって、辺の意味は、もっと洗練されたものになる。
- 次のグラフは、以下の文章を表現している。



Peter Neubauer created the Neo4j webpage on 2007/10.

プロパティ・グラフを利用した グラフ・データベースのデータ構造の例



FreebaseとKnowledge Graph

Googleは、FreebaseのMetaweb社を買収した。
Freebaseは、GoogleのKnowledge Graphのベースになるとともに、Schema.orgの基礎となる。

以下は、2012年7月のマルレク
「Googleの新しい検索技術 Knowledge Graphについて」から
<https://goo.gl/oZGJXG>



Freebase Schema

10,448 Properties

describing

4,936 Types*

organized into

641 Domains

(~70 Commons)

*types with 10 or more instances



Freebase Commons

= 72 domains

American Football	Cricket	Language	Projects
Architecture	Data world	Law	Protected places
Astronomy	Digicams	Library	Radio
Automotive	Education	Location	Rail
Aviation	Engineering	Martial arts	Religion
Awards	Events	Measurement unit	Royalty
Baseball	Fashion	Media commons	Soccer
Basketball	Fictional universes	Medicine	Spaceflight
Bicycles	Film	Meteorology	Sports
Biology	Food	Military	Symbols
Boats	Freebase	Music	Tennis
Books	Games	Olympics	Theater
Broadcast	Geology	Opera	Time
Business	Government	Organization	Transportation
Celebrities	Hobbies and interests	People	Travel
Chemistry	Ice hockey	Periodicals	TV
Computers	Influence	Physical geography	Video games
Conferences	Internet	Physics	Visual arts

MQL Freebaseの検索言語

□ [https://www.googleapis.com/freebase/v1/mqlread?query={"type":"/music/artist","name":"The Police","album":\[\]}\]](https://www.googleapis.com/freebase/v1/mqlread?query={)

```
{  
  "type": "/music/artist",  
  "name": "The Police",  
  "album": []  
}
```

Freebase MQLの検索

MQL Response

```
{  
  "type": "/music/artist",  
  "name": "The Police",  
  "album": []  
}
```

MQL Request

昔は、Freebaseで、
Policeのアルバムも、
全部、調べられたのに。

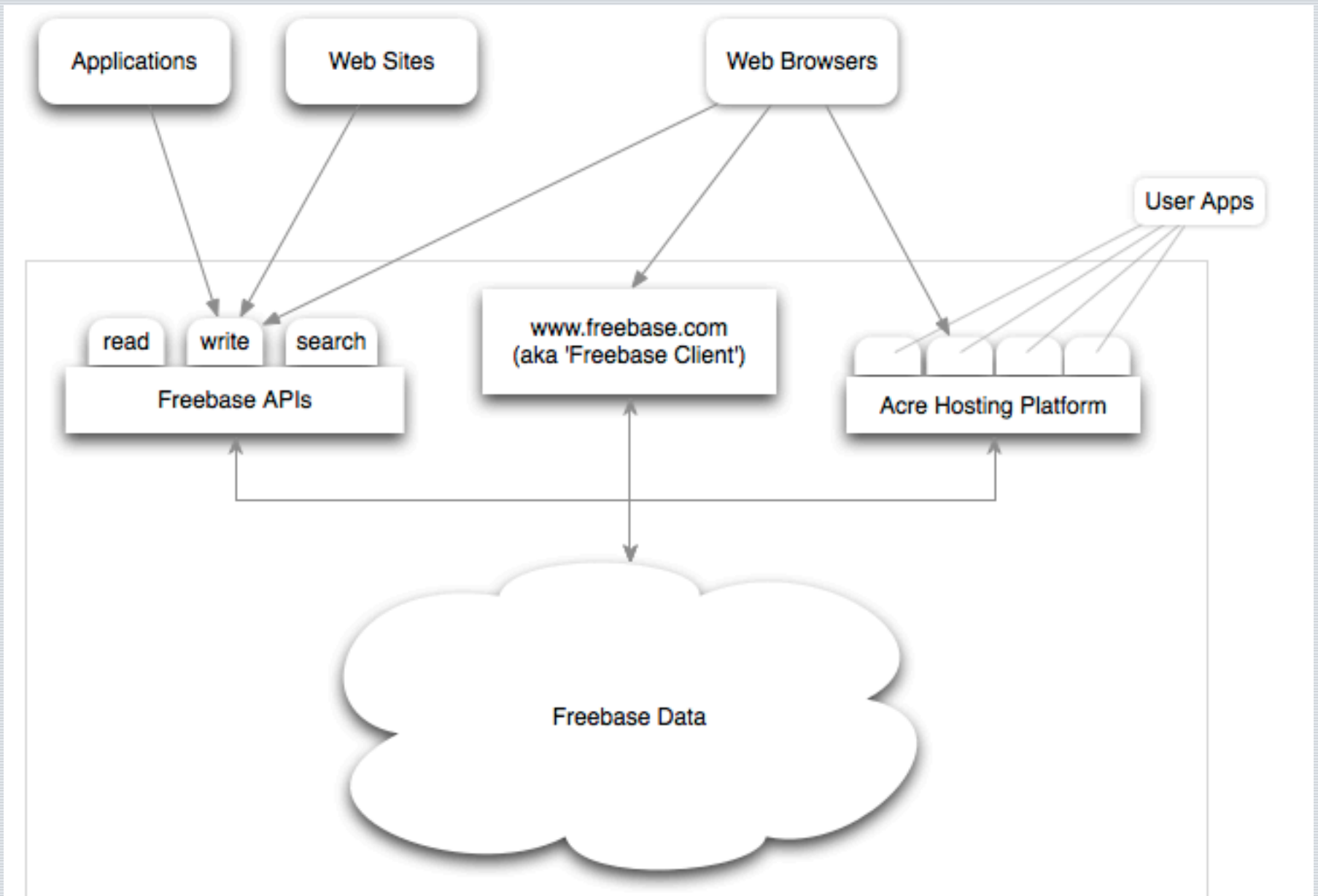
```
{  
  "result": {  
    "album": [  
      "Outlandos d'Amour",  
      "Reggatta de Blanc",  
      "Zenyatta ¥u00e0 Mondatta",  
      "Ghost in the Machine",  
      "Synchronicity",  
      "Every Breath You Take: The Singles",  
      "Greatest Hits",  
      "Message in a Box: The Complete Recordings",  
      "Live!",  
      "Every Breath You Take: The Classics",  
      "Their Greatest Hits",  
      "Can't Stand Losing You",  
      "Roxanne '97 (Puff Daddy remix)",  
      "Roxanne '97",  
      "The Police",  
      "Greatest Hits",  
      "The Very Best of Sting ¥u0026amp; The Police",  
      "Brimstone and Treacle",  
      "Can't Stand Losing You",  
      "De Do Do Do, De Da Da Da",  
      "Certifiable: Live in Buenos Aires",  
    ]  
  }  
}
```

MQLでの検索

```
[{  
  "id": "/en/barack_obama",  
  "/common/topic/article": [{  
    "id": null  
  }]  
}]
```

```
{  
  "status": "200 OK",  
  "code": "/api/status/ok",  
  "result": [{  
    "id": "/en/barack_obama",  
    "/common/topic/article": [{  
      "id": "/guid/9202a8c04000641f800000000029c281"  
    }, {  
      "id": "/guid/9202a8c04000641f80000000005b78173"  
    }]  
  }]  
}
```

[http://api.freebase.com/api/trans/raw/guid/
9202a8c04000641f800000000029c281](http://api.freebase.com/api/trans/raw/guid/9202a8c04000641f800000000029c281)



FreebaseとKnowledge Graph

- ❑ Freebaseの技術は、GoogleのMetaweb社買収を通じて、GoogleのKnowledge Graphに、大きな影響を与えている。ネットワーク上での知識表現のモデルとしては、同じものかもしれない。Freebaseでは、Thingは、TypeのPlaceholderである。
 - ❑ また、FreebaseのEntityがTypeとPropertyを持つというモデルは、次に見る、HTML5のMicrodataにも、そのまま受け継がれている。
 - ❑ Freebaseの思想で、もっとも特徴的なのは、Folksonomyであるが、GoogleのKnowledge Graphでそれがどのように継承されているのかは、よく分からなかった。
-

Google Knowledge Graph Search API

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

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

Google Knowledge Graph Search API

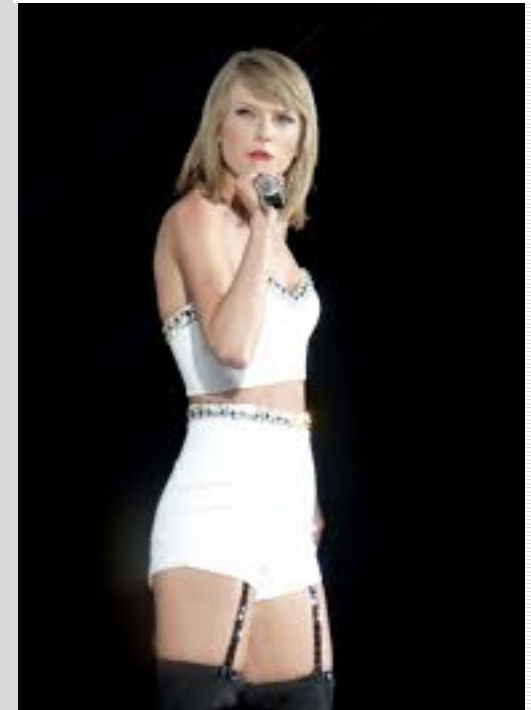
- The Knowledge Graph Search API lets you find entities in the [Google Knowledge Graph](#). The API uses standard [schema.org](#) types and is compliant with the [JSON-LD](#) specification.

- “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:ANd9GcQmVDAhjhWnN2OW
ys2ZMO3PGAhupp5tN2LwF_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_Commo
ns_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>
```

Schema.orgとEntityモデルの問題

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](https://github.com/schemaorg/schemaorg) 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/>

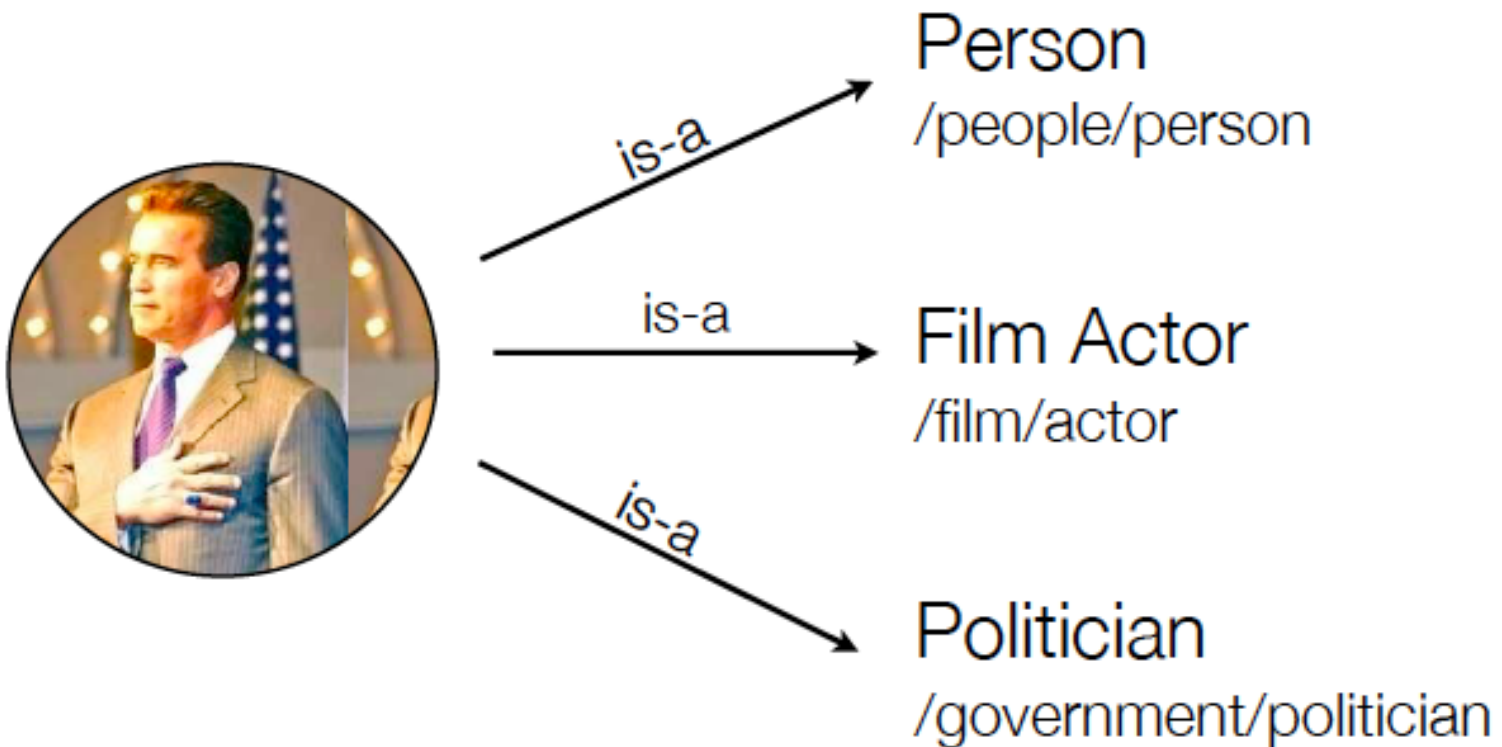
Schema.org vocabulary

- Schema.org ボキャブラリーは、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"

- | | |
|---|---|
| ☐ <u>Book</u> | ☐ <u>MusicGroup</u> |
| ☐ <u>BookSeries</u> | ☐ <u>MusicRecording</u> |
| ☐ <u>EducationalOrganization</u> | ☐ <u>Organization</u> |
| ☐ <u>Event</u> | ☐ <u>Periodical</u> |
| ☐ <u>GovernmentOrganization</u> | ☐ <u>Person</u> |
| ☐ <u>LocalBusiness</u> | ☐ <u>Place</u> |
| ☐ <u>Movie</u> | ☐ <u>SportsTeam</u> |
| ☐ <u>MovieSeries</u> | ☐ <u>TVEpisode</u> |
| ☐ <u>MusicAlbum</u> | ☐ <u>TVSeries</u> |
| | ☐ <u>VideoGame</u> |
| | ☐ <u>VideoGameSeries</u> |
| | ☐ <u>WebSite</u> |



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](https://github.com/schemaorg/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](https://github.com/schemaorg/schemaorg) / [data](https://github.com/schemaorg/schemaorg) / 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

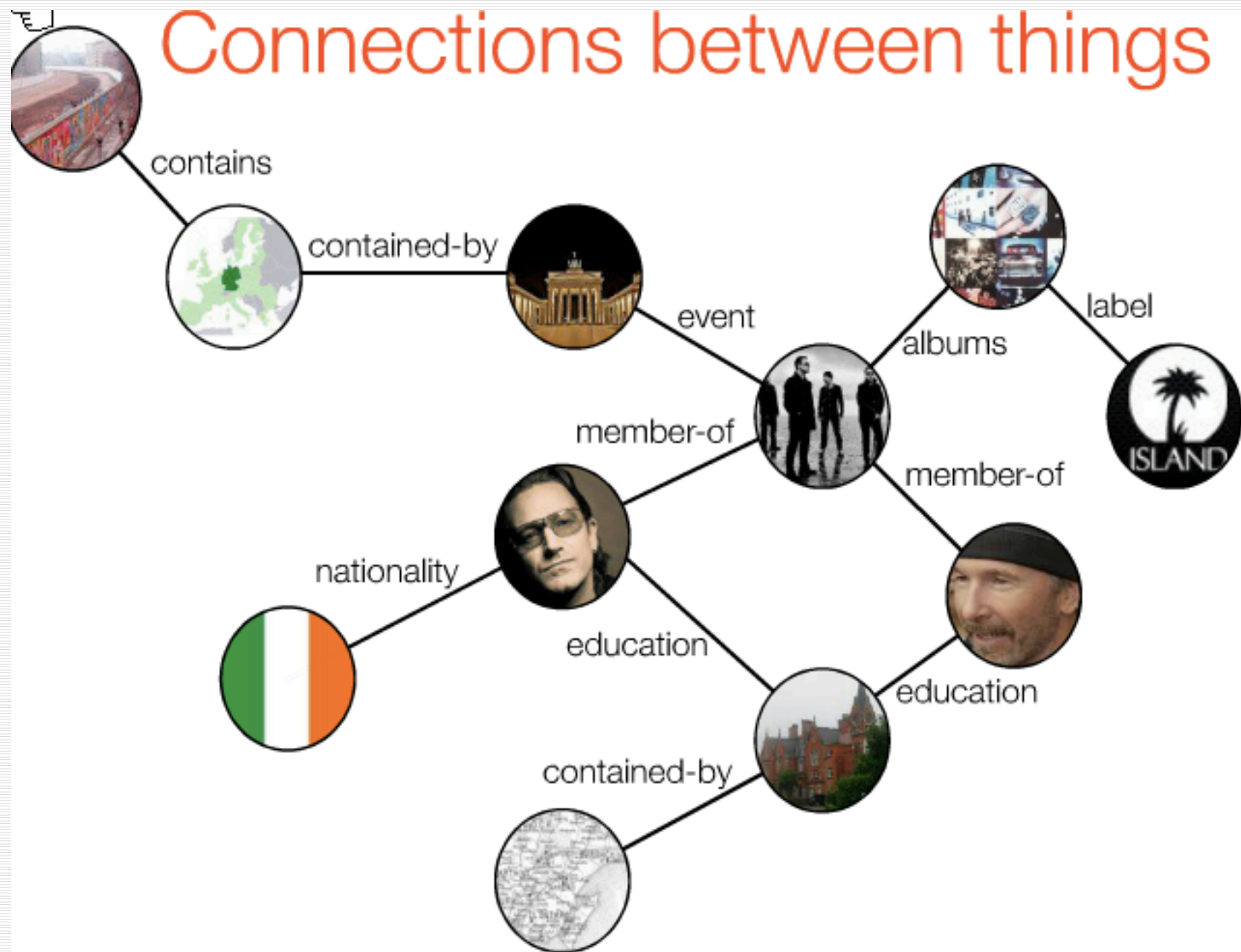
A hospital.

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Property

- 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>

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

<http://schema.org/Person>

Schema.orgは体系的で網羅的か？

Schema.orgは、体系的でも網羅的でもない。
Entityの体系性・網羅性においては、Wordnetに劣る。
スキーマだけでは、Wikipediaの情報にかなわない。

'Language'の検索

Entityの枚挙は、体系的でも
網羅的でもない

Language

Canonical URL: <http://schema.org/Language>

[Thing](#) > [Intangible](#) > [Language](#)

Thing > Intangible > Language

Natural languages such as Spanish, Tamil, Hindi, English, etc. Formal language code tags expressed in [BCP 47](#) can be used via the [alternateName](#) property. The Language type previously also covered programming languages such as Scheme and Lisp, which are now best represented using [ComputerLanguage](#).

Usage: Between 100 and 1000 domains

[\[more...\]](#)

Property	Expected Type	Description
Properties from Thing		
additionalType	URL	An additional type for the item, typically used for adding more specific types from external vocabularies in microdata syntax. This is a relationship between something and a class that the thing is in. In RDFa syntax, it is better to use the native RDFa syntax – the 'typeof' attribute – for multiple types. Schema.org tools may have only weaker understanding of extra types, in particular those defined externally.
alternateName	Text	An alias for the item.
description	Text	A description of the item.
disambiguatingDescription	Text	A sub property of description. A short description of the item used to disambiguate from other, similar items. Information from other properties (in particular, name) may be necessary for the description to be useful for disambiguation.
identifier	PropertyValue or Text or URL	The identifier property represents any kind of identifier for any kind of Thing , such as ISBNs, GTIN codes, UUIDs etc. Schema.org provides dedicated properties for representing many of these, either as textual strings or as URL (URI) links. See background notes for more details.
image	ImageObject or URL	An image of the item. This can be a URL or a fully described ImageObject .

<http://schema.org/Language>

Intangible の下位タイプ

- AlignmentObject
- Audience
- BedDetails
- Brand
- BroadcastChannel
- BusTrip
- ComputerLanguage
- DataFeedItem
- Demand
- DigitalDocumentPermission
- EntryPoint
- Enumeration
- Flight
- GameServer
- Invoice
- ItemList
- JobPosting
- Language
- ListItem

- MenuItem
- Offer
- Order
- OrderItem
- ParcelDelivery
- Permit
- ProgramMembership
- PropertyValueSpecification
- Quantity
- Rating
- Reservation
- Role
- Seat
- Service
- ServiceChannel
- StructuredValue
- Ticket
- TrainTrip

Entityの枚挙は、体系的でも
網羅的でもない

Schema.orgの
乱雑さと粗末さが、
よくわかる

<http://schema.org/Intangible>

[fuelEfficiency – schema.org](#)

... fuelConsumption (e.g. 8 liters per 100 km) and fuelEfficiency (e.g. 30 miles per gallon). They are reciprocal. Note 3: Often, the absolute value is useful only when related to driving speed ("at 80 km/h") or usage pattern ("city traffic"). You can use valueReference to link the value for the fuel **economy** to another value. [more.

[schema.org/fuelEfficiency](#)

[RentalCarReservation – schema.org](#)

Car: Honda Civic (**Economy** Class); Rental company: Hertz; Pick up location: Hertz San Diego Airport; 1500 Orange Avenue; San Diego, CA 94043; Pickup time: 2017-08-05T16:00:00-07:00; Drop off location: Hertz LAX; 1234 First Street; Los Angeles, CA 94043; Drop off time: 2017-08-06T20:00:00-07:00. This example is ...

[schema.org/RentalCarReservation](#)

[QuantitativeValue – schema.org](#)

... fuelConsumption (e.g. 8 liters per 100 km) and fuelEfficiency (e.g. 30 miles per gallon). They are reciprocal. Note 3: Often, the absolute value is useful only when related to driving speed ("at 80 km/h") or usage pattern ("city traffic"). You can use valueReference to link the value for the fuel **economy** to another value.

[schema.org/QuantitativeValue](#)

[LegalService – schema.org](#)

hasOfferCatalog · OfferCatalog, Indicates an OfferCatalog listing for this Organization, Person, or Service. hasPOS · Place, Points-of-Sales operated by the organization or person. isicV4 · Text, The International Standard of Industrial Classification of All **Economic** Activities (ISIC), Revision 4 code for a particular organization, ...

Entityの枚挙は、体系的でも
網羅的でも無い

economyは、
燃費とか、エコノミー・
クラスに関連づけられて
いる。別に、間違っ
ちやいないけど。

'Mathematics'の検索

schema.org

Mathematics

Home

Schemas

Doc

Extended schema.org news support

Sep 27, 2011 ... 2011 at 9:49 PM. Kavi – We are describing words in a dictionary (**Mathematics** Dictionary). Each math word is split into sections like the word/phrase, the definition, examples, some solved examples, related math words/concepts etc.
http://www.icoachmath.com/math_dictionary/Binomial_Experiment.html

blog.schema.org/2011/09/extended-schemaorg-news-support.html

MedicalRiskCalculator – health-lifesci.schema.org

Schema.org Type: MedicalRiskCalculator – A complex **mathematical** calculation requiring an online calculator, used to assess prognosis. Note: use the url property of Thing to record any URLs for online calculators.

health-lifesci.schema.org/MedicalRiskCalculator

Schema.org master file: RDFS in RDFa

MedicalRiskCalculator A complex **mathematical** calculation requiring an online calculator, used to assess prognosis. Note: use the url property of Thing to record any URLs for online calculators. Subclass of: MedicalRiskEstimator Source: WikiDoc. MedicalRiskFactor
A risk factor is anything that increases a person's ...

schema.org/version/2.0/schema.rdfa

Schema.org version 2.0

May 13, 2015 ... MedicalRiskCalculator. A complex **mathematical** calculation requiring an online calculator, used to assess prognosis. Note: use the url property of Thing to record any URLs for online calculators. Properties used on this type: Properties whose values are of this type: [^top] ...

Entityの枚挙は、体系的でも
網羅的でも無い

mathematicsに
しても同じ。
ろくなもんじゃない。
数学についての
知識が表現されるわ
けでは無い。
別に、間違っちゃい
ないけど。

Word to search for:

WordNetでの記述

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"

Noun

<https://goo.gl/8j16SW>

- **S: (n) language, linguistic communication** (a systematic means of communicating by the use of sounds or conventional symbols) *"he taught foreign languages"; "the language introduced is standard throughout the text"; "the speed with which a program can be executed depends on the language in which it is written"*
- **S: (n) speech, speech communication, spoken communication, spoken language, language, voice communication, oral communication** ((language) communication by word of mouth) *"his speech was garbled"; "he uttered harsh language"; "he recorded the spoken language of the streets"*
- **S: (n) lyric, words, language** (the text of a popular song or musical-comedy number) *"his compositions always started with the lyrics"; "he wrote both words and music"; "the song uses colloquial language"*
- **S: (n) linguistic process, language** (the cognitive processes involved in producing and understanding linguistic communication) *"he didn't have the language to express his feelings"*
- **S: (n) language, speech** (the mental faculty or power of vocal communication) *"language sets homo sapiens apart from all other animals"*
- **S: (n) terminology, nomenclature, language** (a system of words used to name things in a particular discipline) *"legal terminology"; "biological nomenclature"; "the language of sociology"*

- **S: (n) language, linguistic communication** (a systematic means of communicating by the use of sounds or conventional symbols) *"he taught foreign languages"; "the language introduced is standard throughout the text"; "the speed with which a program can be executed depends on the language in which it is written"*
 - **direct hyponym / full hyponym**
 - **S: (n) usage** (the customary manner in which a language (or a form of a language) is spoken or written) *"English usage"; "a usage borrowed from French"*
 - **S: (n) dead language** (a language that is no longer learned as a native language)
 - **S: (n) words** (language that is spoken or written) *"he has a gift for words"; "she put her thoughts into words"*
 - **S: (n) source language** (a language that is to be translated into another language)
 - **S: (n) object language, target language** (the language into which a text written in another language is to be translated)
 - **S: (n) sign language, signing** (language expressed by visible hand gestures)
 - **S: (n) artificial language** (a language that is deliberately created for a specific purpose)
 - **S: (n) metalanguage** (a language that can be used to describe languages)
 - **S: (n) native language** (the language that a person has spoken from earliest childhood)
 - **S: (n) indigenous language** (a language that originated in a specified place and was not brought to that place from elsewhere)
 - **S: (n) superstrate, superstratum** (the language of a later invading people that is imposed on an indigenous population and contributes features to their language)
 - **S: (n) natural language, tongue** (a human written or spoken language used by a community; opposed to e.g. a computer language)

Entity Linking Intelligence Service API

<https://docs.microsoft.com/ja-jp/azure/cognitive-services/entitylinking/home>

Entity Linking

- Sometimes in different contexts, a word might be used as a named entity, a verb, or other word form within a given sentence. For example, in the case where “times” is a named entity, it still may refer to two separately distinguishable entities, such as “The New York Times” or “Times Square”. Given a specific paragraph within a document, the Entity Linking Intelligence Service will recognize and identify each separate entity based on its context.
-

- The illustration below shows an example of entity linking. Specifically, when **using Wikipedia**, the Entity Linking Intelligence Service detects all entities mentioned within the input text and links them to relevant reference points according to the page ID.



Bing Search API v7 and Bing Custom Search are now generally available

<https://goo.gl/qiJRBV>

- ❑ [Bing Web Search API](#)
 - ❑ [Bing Image Search API](#)
 - ❑ [Bing Video Search API](#)
 - ❑ [Bing News Search API](#)
 - ❑ [Bing Autosuggest API](#)
 - ❑ [Bing Spell Check API](#)
-

知識データベース構築の試み

Wikipedia Infobox と DBpedia

*DBpedia is a **crowd-sourced community effort** to extract structured information from Wikipedia and make this information available on the Web.*

Wikipediaからの情報抽出

- **Wikipedia Infobox**は、投稿者がこのツールを使うと、自動的にタグ付けされたデータが生成されるというもの。
<http://en.wikipedia.org/wiki/Help:Infobox>
 - **DBpedia**は、基本的には、先のInfoboxを利用して、Wikipediaから構造化されたデータを抽出して、Webで利用出来るようにしようというコミュニティの取り組み。
<https://en.wikipedia.org/wiki/DBpedia>
-

{{**Infobox** scientist

| name = Marie Skłodowska-Curie

| image = Marie Curie c1920.png

| image_size=220px

| caption = Marie Curie, ca. 1920

| birth_date = {{birth date|1867|11|7|df=y}}

| birth_place = [[Warsaw]], [[Congress Poland|Kingdom of Poland]],

then part of [[Russian Empire]]<ref>http://www.nobelprize.org/nobel_prizes

| death_date = {{death date and age|df=yes|1934|7|4|1867|11|7}}

| death_place = [[Passy, Haute-Savoie]], France

| residence = [[Poland]] and [[France]]

| citizenship = Poland
France

| field = [[Physics]], [[Chemistry]]

| work_institutions = [[University of Paris]]

| alma_mater = University of Paris
[[ESPCI]]

| doctoral_advisor = [[Henri Becquerel]]

| doctoral_students = [[André-Louis Debierne]]
[[Óscar Moreno]]

| known_for = [[Radioactivity]], [[polonium]], [[radium]]

| spouse = [[Pierre Curie]] (1859–1906)

| prizes = {{nowrap|[[Nobel Prize in Physics]] (1903)
[[Davy Medal]]

| footnotes = She is the only person to win a [[Nobel Prize]] in two different s

| religion = Agnostic

| signature = Marie_Curie_Skłodowska_Signature_Polish.jpg

}}

Marie CurieのInfoboxでの記述

DBpediaでのビートルズのエントリー

About: ビートルズ

An Entity of Type : [organisation](#), from Named Graph : <http://dbpedia.org>, within Data Space : [dbpedia.org](#)

ザ・ビートルズ (The Beatles) は、イギリス・リヴァプール出身のロックバンド（個々のアルバムやビデオ作品については別項ビートルズの作品を照のこと）。

Property	Value
dbpedia-owl:abstract	- The Beatles waren eine britische Rockband, bestehend aus John Lennon, Paul McCartney, C Bands des 20. Jahrhunderts. - The Beatles were an English rock band, formed in Liverpool in 1960. They are one of the mo (rhythm guitar, vocals), Paul McCartney (bass guitar, vocals), George Harrison (lead guitar, v ballads to psychedelic rock, often incorporating classical and other elements in innovative w They came to be perceived as the embodiment of ideals of the social and cultural revolution reputation playing clubs in Liverpool and Hamburg over a three-year period from 1960. Sut Epstein, their musical potential was enhanced by the creativity of producer George Martin. T the course of the next year, they toured extensively until 1966, then retreated to recording critics consider some of their finest material including the album Sgt. Pepper's Lonely Hear their break-up, their recordings are still in demand. They have had more number one albums United States than any other artist. The Beatles were placed at number one on Billboard ma Academy of Recording Arts and Sciences and 15 Ivor Novello Awards from the British Acad influential people. Lennon was murdered outside his home in New York City in 1980, and Ha - The Beatles fue una banda inglesa de Rock formada en Liverpool en 1960. Es frecuentemen por John Lennon, Paul McCartney, George Harrison y Ringo Starr. Aunque en sus inicios to folk rock hasta el rock psicodélico. La naturaleza de su enorme popularidad, que había emer ser percibidos como la encarnación de los ideales progresistas, extendiendo su influencia e McCartney, Harrison, Stuart Sutcliffe (bajo) y Pete Best (batería), construyeron su reputaci reemplazado por Starr al año siguiente. Establecidos como grupo profesional después de qu lograron éxito comercial en el Reino Unido a finales de 1962 con su primer sencillo, «Love M de giras hasta 1966, año en que cesaron la actividad en vivo para dedicarse únicamente a la Lennon sería asesinado a las afueras de su casa de Nueva York en 1980, y Harrison fallecer instrumento el álbum Sgt. Pepper's Lonely Hearts Club Band (1967), considerado por much

DBpediaでのビートルズのエントリー

dbpedia-owl:activeYearsEndYear	1970-01-01 00:00:00 (xsd:date)
dbpedia-owl:activeYearsStartYear	1960-01-01 00:00:00 (xsd:date)
dbpedia-owl:associatedBand	- dbpedia:Plastic_Ono_Band - dbpedia:The_Quarrymen
dbpedia-owl:associatedMusicalArtist	- dbpedia:Plastic_Ono_Band - dbpedia:The_Quarrymen
dbpedia-owl:background	group_or_band
dbpedia-owl:bandMember	- dbpedia:George_Harrison - dbpedia:John_Lennon - dbpedia:Ringo_Starr - dbpedia:Paul_McCartney
dbpedia-owl:formerBandMember	- dbpedia:Stuart_Sutcliffe - dbpedia:Pete_Best
dbpedia-owl:genre	- dbpedia:Pop_music - dbpedia:Rock_music
dbpedia-owl:hometown	dbpedia:Liverpool
dbpedia-owl:recordLabel	- dbpedia:United_Artists_Records - dbpedia:Parlophone - dbpedia:Apple_Records - dbpedia:Vee-Jay_Records - dbpedia:Capitol_Records
dbpedia-owl:thumbnail	http://upload.wikimedia.org/wikipedia/commons/thumb/d/df/The_Fabs.JPG/
dbpedia-owl:wikiPageExternalLink	http://www.bbc.co.uk/6music/news/20090407_beatles.shtml http://news.bbc.co.uk/2/hi/entertainment/1492446.stm http://dockets.justia.com/docket/florida/flsdce/1:2008cv20748/311641/ http://uk.reuters.com/article/2008/11/17/music-us-beatles-idUKTRE4AG6 http://www.independent.co.uk/news/obituaries/ivor-arbiter-508146.html http://books.google.ca/books?id=yqelM3MY3gAC&lpg=PA1&dq=The%20Be http://articles.latimes.com/2009/apr/08/entertainment/et-beatles8 http://www.billboard.com/features/paul-mccartney-the-billboard-g-a-10040

Marie Curieの検索で 表示される「要約」

+Dave Search Images Maps Play YouTube News Gmail Documents Calendar More

Google

marie curie

Search

8 personal results, 17,300,000 other results (0.32 seconds)

Everything

Images

Maps

Videos

News

Shopping

Books

More

Mountain View, CA
Change location

Any time

Past hour

Past 24 hours

Past 2 days

Past week

Past month

Past year

Custom range...

Marie Curie - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Marie_Curie
Marie Skłodowska-Curie (7 November 1867 – 4 July 1934) was a French-Polish physicist and chemist famous for her pioneering research on radioactivity.
→ Marie Curie Cancer Care - Pierre Curie - Irène Joliot-Curie - Radioactive decay

Marie Curie - Biography - Nobelprize.org
www.nobelprize.org/nobel_prizes/physics/.../marie-curie-bio.html
Short profile from the foundation that awards the Nobel Prize.

Marie Curie - Biography - Nobelprize.org
www.nobelprize.org/nobel_prizes/chemistry/.../marie-curie-bio.html
Marie Curie, née Maria Skłodowska, was born in Warsaw on November 7, 1867, the daughter of a secondary-school teacher. She received a general education ...

Images for marie curie - Report images

Marie Curie and The Science of Radioactivity
www.aip.org/history/curie/
The life of **Marie Curie**, from the AIP Center for History of Physics. Text by Naomi Pasachoff and many illustrations describe Curie's contributions to the science of ...

Marie Curie

Marie Skłodowska-Curie was a French-Polish physicist and chemist famous for her pioneering research on radioactivity. She was the first person honored with two Nobel Prizes—in physics and chemistry. Wikipedia

Born: November 7, 1867, [Warsaw](#)

Died: July 4, 1934, [Sancellemoz](#)

Spouse: [Pierre Curie](#) (m. 1895–1906)

Children: [Irène Joliot-Curie](#), [Ève Curie](#)

Discovered: [Radium](#), [Polonium](#)

Education: [École Supérieure de Physique et de Chimie Industrielles de la Ville de Paris](#), [University of Paris](#)

People also search for

[Albert Einstein](#) [Pierre Curie](#) [Ernest Rutherford](#) [Louis Pasteur](#) [John Dalton](#)

Marie Curie



es.wikipedia.org

Marie Skłodowska-Curie was a French-Polish physicist and chemist famous for her pioneering research on radioactivity. She was the first person honored with two Nobel Prizes—in physics and chemistry. Wikipedia

Born: November 7, 1867, [Warsaw](#)

Died: July 4, 1934, [Sancellemoz](#)

Spouse: [Pierre Curie](#) (m. 1895–1906)

Children: [Irène Joliot-Curie](#), [Ève Curie](#)

Discovered: [Radium](#), [Polonium](#)

Education: [École Supérieure de Physique et de Chimie Industrielles de la Ville de Paris](#), [University of Paris](#)

People also search for



[Albert Einstein](#)



[Pierre Curie](#)



[Ernest Rutherford](#)



[Louis Pasteur](#)



[John Dalton](#)

[Report a problem](#)

「要約」は、infobox等から生成される

Marie Curieの検索の要約

- ❑ Marie Skłodowska-Curie was a French-Polish physicist and chemist, famous for her pioneering research on radioactivity. She was the first person honored with two Nobel Prizes—in physics and chemistry. [Wikipedia](#)
- ❑ Born: November 7, 1867, Warsaw
- ❑ Died: July 4, 1934, Sancellemoz
- ❑ Spouse: Pierre Curie (m. 1895–1906)
- ❑ Discovered: Polonium, Radium
- ❑ Children: Irène Joliot-Curie, Ève Curie
- ❑ Education: University of Paris (1894), University of Paris (1893), More

The Beatles



en.wikipedia.org

The Beatles were an English rock band formed in Liverpool in 1960, who eventually became one of the most commercially successful and critically acclaimed acts in the history of popular music.

Wikipedia

Active from: 1957

Members: [Paul McCartney](#), [John Lennon](#), [George Harrison](#), [Ringo Starr](#), [Pete Best](#), [Stuart Sutcliffe](#)

Awards: Grammy Award for Best Performance by a Vocal Group, [More](#)

Record labels: EMI, Capitol Records, Apple Records, Polydor Records, Swan Records, [More](#)

Song

[Let It Be](#)

Year

1970

Album

Let It Be

[Hey Jude](#)

1970

Hey Jude

[Here Comes the Sun](#)

1969

Abbey Road

[Yesterday](#)

1965

Help!

[In My Life](#)

1965

Rubber Soul

Albums

要約システムは、普遍的か？

- Google Knowledge Graphで、情報の「要約」の機能が追加されたのは、注目に値する変化である。検索システムの変化として、正しい方向の一つだと思う。
- Google Knowledge Graphが、一定量の、要約のデータベースを持っているのは事実であり、それは、その実装方法を含めて、興味深いことである。
- ただ、それは、「最良」のものであろうか？ 質的な問題は、置いておいても、重要なのは、その量である。量的には、Wikipediaのページでさえ、全てのページの要約が提供されている訳ではないことにすぐ気づく。一般のページの要約には、ほど遠いのが現状である。その意味では、全てのページを対象にする、ページランクのような普遍性を、現状では、それは持っている訳ではない。

誰が、要約をしているのか？

- こうした量的な制約は、基本的には、要約のデータが、どのように生成されているのかという問題に帰着する。いったん、要約のデータが、なんらかの形で出来てしまえば、我々は、その出自を問うことはないのだが。
- 要約データを、どう生成するのかという点では、基本的なアプローチは2つある。一つは、対象となるテキストから、機械が要約を生成する、自動情報抽出のアプローチ。もう一つは、Web上の情報に人間がタグ付けをして、機械がそれを利用出来るようにしようという、Semantic Web流のアプローチである。

*DBpedia is a **crowd-sourced community effort** to extract structured information from Wikipedia and make this information available on the Web.*

Wikidata

<https://en.wikipedia.org/wiki/Wikidata>

Wikidata

- **Wikidata** is a [collaboratively edited knowledge base](#) hosted by the [Wikimedia Foundation](#). It is intended to provide a common source of data which can be used by Wikimedia projects such as [Wikipedia](#),^{[2][3]} and by anyone else, under a [public domain](#) license. This is similar to the way [Wikimedia Commons](#) provides storage for media files and access to those files for all Wikimedia projects, and which are also freely available for reuse. Wikidata is powered by the software [Wikibase](#).^[4]
-

Concept

- ❑ Wikidata is a [document-oriented database](#), focused on items. Each item represents a topic (or an [administrative page](#) used to maintain Wikipedia) and is identified by a unique number, prefixed with the letter Q—for example, the item for the topic [Politics](#) is [Q7163](#). This enables the basic information required to identify the topic the item covers to be translated without favouring any language.
- ❑ Information is added to items by creating statements. Statements take the form of [key-value pairs](#), with each statement consisting of a property (the key) and a value linked to the property.

label

Douglas Adams (Q42)

item
identifier

description

English writer and humorist
Douglas Noël Adams | Douglas Noel Adams
► [In more languages](#)

aliases

Statements

property

educated at

St John's College

value

end time	1974
academic major	English literature
academic degree	Bachelor of Arts
start time	1971

qualifiers

rank

▼ 2 references

stated in	Encyclopædia Britannica Online
reference URL	http://www.nndb.com/people/731/000023662/
original language of work	English
retrieved	7 December 2013
publisher	NNDB
title	Douglas Adams (English)

opened
references

+ add reference

statement
group

▲
●
▼
Brentwood School

end time	1970
start time	1959

► 0 references

collapsed
reference

+ add (statement)

Development history

The creation of the project was funded by donations from the [Allen Institute for Artificial Intelligence](#), the [Gordon and Betty Moore Foundation](#), and [Google, Inc.](#), totaling €1.3 million.^{[5][6]} Initial development of the project is being overseen by [Wikimedia Deutschland](#) and has been split into three phases:^[7]

1. Centralising interlanguage links – links between Wikipedia articles about the same topic in different languages
2. Providing a central place for infobox data for all Wikipedias
3. Creating and updating list articles based on data in Wikidata

Unsourced, unreliable, and in your face forever: Wikidata, the future of online nonsense

- ❑ **Special report** Lobbying companies, PR professionals and SEO optimisers are flocking to influence Wikidata, a child project of Wikipedia that's backed by serious money. And that's just one of the reasons to be concerned about a project that *could* become the world's default source of information.
 - ❑ The information in Google's Knowledge Graph – the info box you see on the right-hand side of search result pages – has given the internet a good few laughs.
 - ❑
-

- ❑ Based on crowdsourced projects like Wikipedia and Freebase, it sometimes displays wrong (or nude) images and Wikipedia vandalism, and it's been susceptible to SEO manipulations of its underlying Freebase layer. With the demise of Freebase, Wikipedia spin-off Wikidata is set to step into the breach. But will that make things better?
- ❑ In a paper co-written with Heather Ford of the School of Media and Communication at the University of Leeds, he has [examined](#) the problems that can result when Wikidata and/or the Knowledge Graph provide the public with a single, unattributed answer. Nuance is lost along with the data's provenance. The process of generating the information is more opaque to the end user than ever before.

BabelNet

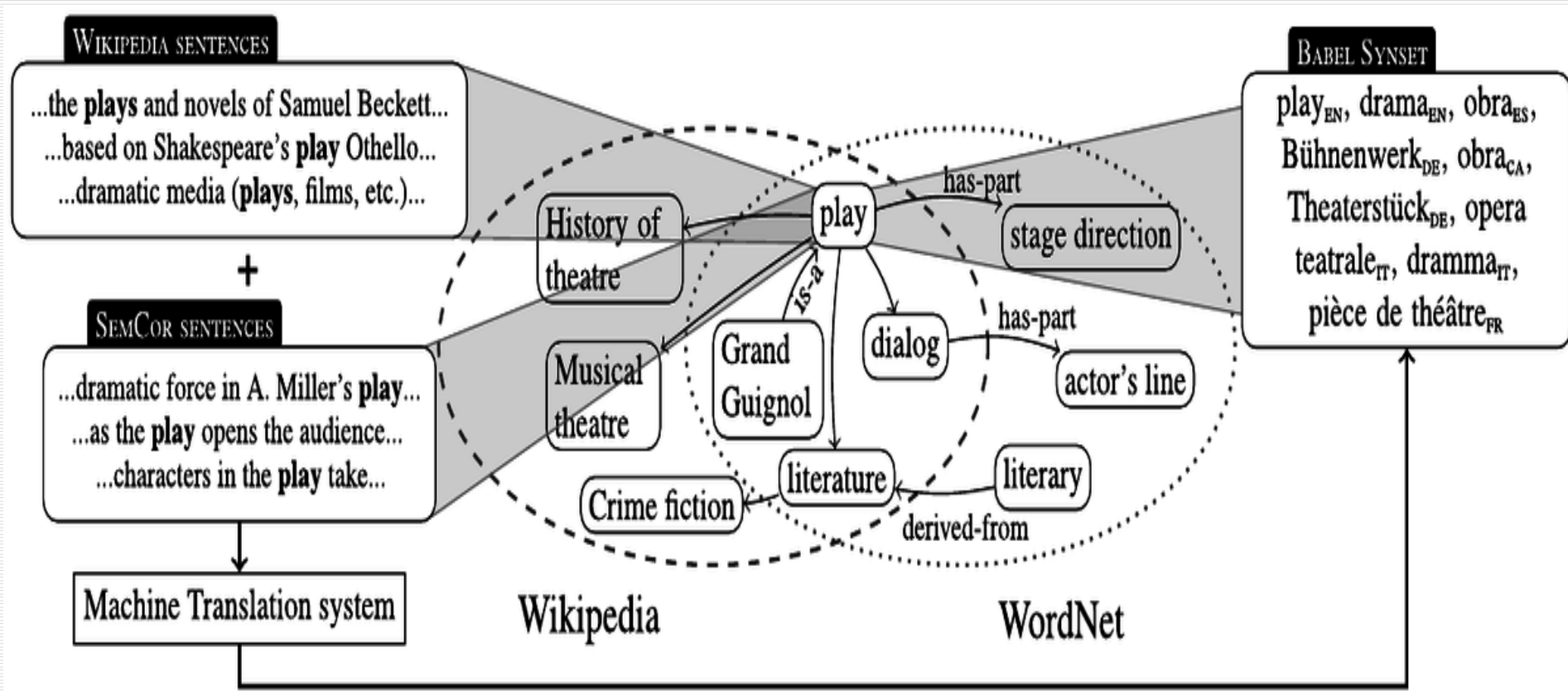
<https://en.wikipedia.org/wiki/BabelNet>

BabelNet

- **BabelNet** is a [multilingual](#) lexicalized [semantic network](#) and [ontology](#) developed at the [Sapienza University of Rome](#) the Department of Computer Science Linguistic Computing Laboratory.^{[1][2]} BabelNet was automatically created by linking Wikipedia to the most popular computational [lexicon](#) of the [English language](#), [WordNet](#). The integration is done using an automatic mapping and by filling in lexical gaps in resource-poor [languages](#) by using [statistical machine translation](#). The result is an "encyclopedic dictionary" that provides [concepts](#) and [named entities](#) [lexicalized](#) in many languages and connected with large amounts of [semantic relations](#).

-
- Additional lexicalizations and definitions are added by linking to free-license wordnets, OmegaWiki, the English [Wiktionary](#), [Wikidata](#), [FrameNet](#), [VerbNet](#) and others. Similarly to WordNet, BabelNet groups [words](#) in different languages into sets of [synonyms](#), called *Babel [synsets](#)*. For each Babel synset, BabelNet provides short definitions (called [glosses](#)) in many languages harvested from both WordNet and Wikipedia.
-

BabelNet is a multilingual semantic network obtained as an integration of WordNet and Wikipedia.



Statistics of BabelNet

- As of August 2016, BabelNet (version 3.7) covers 271 [languages](#), including all European languages, most [Asian languages](#), and [Latin](#). BabelNet 3.7 contains almost 14 million synsets and about 746 million [word senses](#) (regardless of their language). Each Babel synset contains 2 synonyms per language, i.e., word senses, on average. The semantic network includes all the lexico-semantic relations from WordNet ([hypernymy](#) and [hyponymy](#), [meronymy](#) and [holonymy](#), [antonymy](#) and [synonymy](#), etc., totaling around 364,000 relation edges) as well as an underspecified relatedness relation from Wikipedia (totaling around 380 million edges).^[1]

-
- Version 3.7 also associates about 11 million images with Babel synsets and provides a Lemon [RDF](#) encoding of the resource,^[3] available via a [SPARQL endpoint](#). 2.67 million synsets are assigned domain labels.
-

Applications

- BabelNet has been shown to enable multilingual [Natural Language Processing](#) applications.

The lexicalized [knowledge](#) available in BabelNet has been shown to obtain state-of-the-art results in:

1. [semantic relatedness](#)^{[4][5]}
 2. multilingual [Word Sense Disambiguation](#)^[6]
 3. multilingual Word Sense Disambiguation and [Entity Linking](#) with the [Babelfy](#) system^[7]
 4. [video games with a purpose](#)^[8]
-

Prizes and acknowledgments

- ❑ BabelNet received the [META prize](#) 2015 for "groundbreaking work in overcoming language barriers through a multilingual lexicalised semantic network and ontology making use of heterogeneous data sources".
 - ❑ BabelNet featured prominently in a [TIME magazine](#)'s article^[9] about the new age of innovative and up-to-date lexical knowledge resources available on the Web. The article describes in some detail how BabelNet is playing a leading role in the 21st century scenario.
-

ボイス・アシスタントは
「文」の意味を理解しているか？

Intent Signature

Schema.orgのEntity Modelの採用

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


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

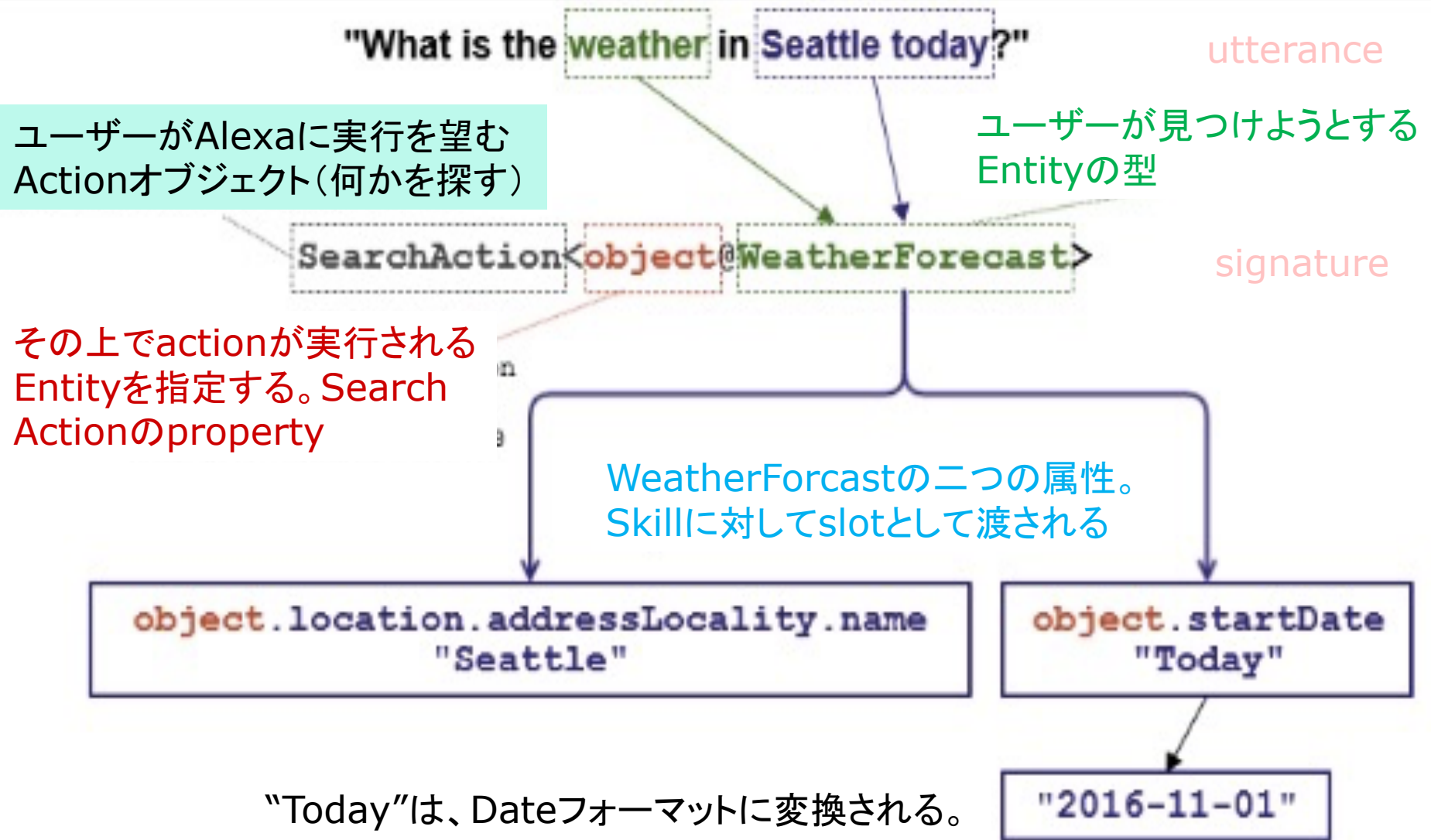
- built-in intent は、***intent signature***を持つ。この signatureを、intentの名前としてSchemaに書くことができる。次のサンプルは、二つのWeatherForecast intents をもった、Schemaの例である。

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

- **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 “Seattle”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（WeatherForecast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@WeatherForecast>** 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**

"What is the **weather** in **Seattle** today?"

この発話全体で、ユーザは、何らかの情報を求めていることを表している。

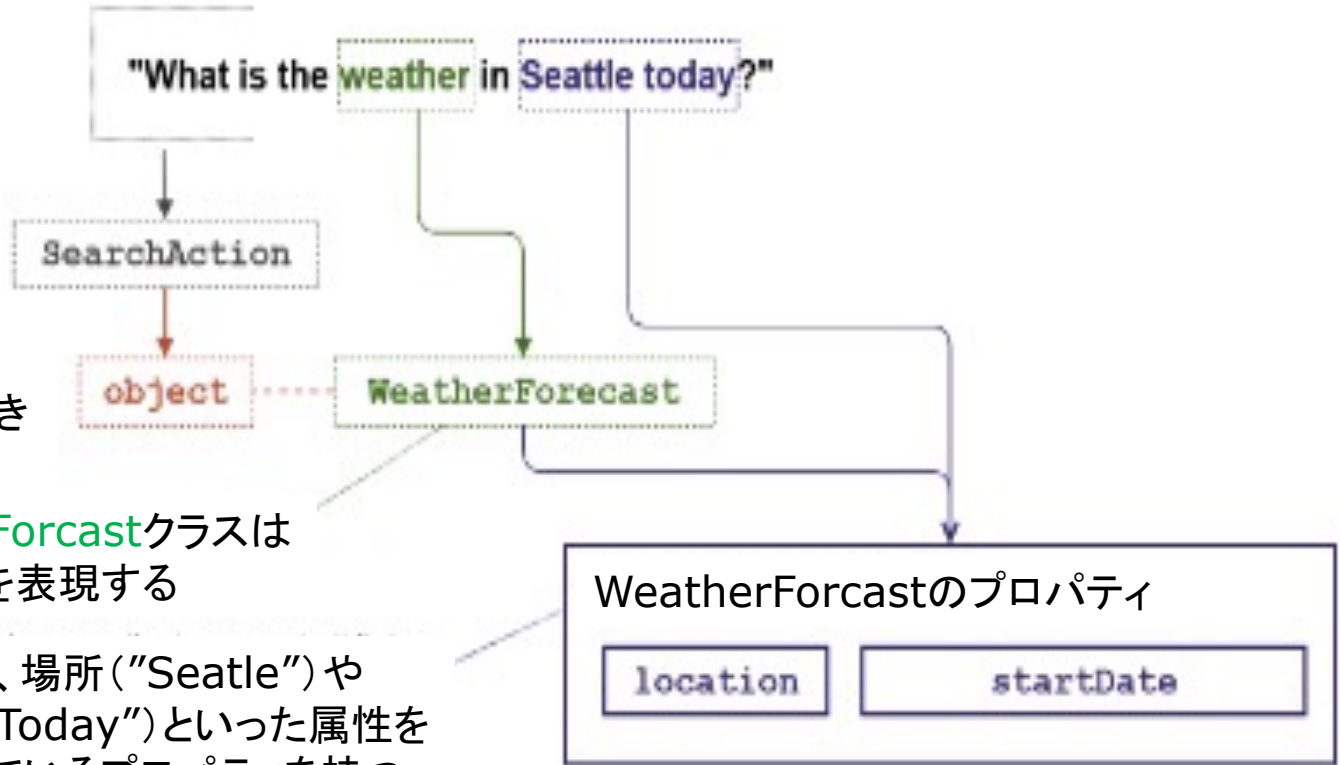
“weather”という言葉が、このユーザーの発話の**Focus**である。これが、ユーザーがを見つけようとしている object である。

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

WeatherForecastクラスは
天気予報を表現する

それは、場所("Seattle")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
-

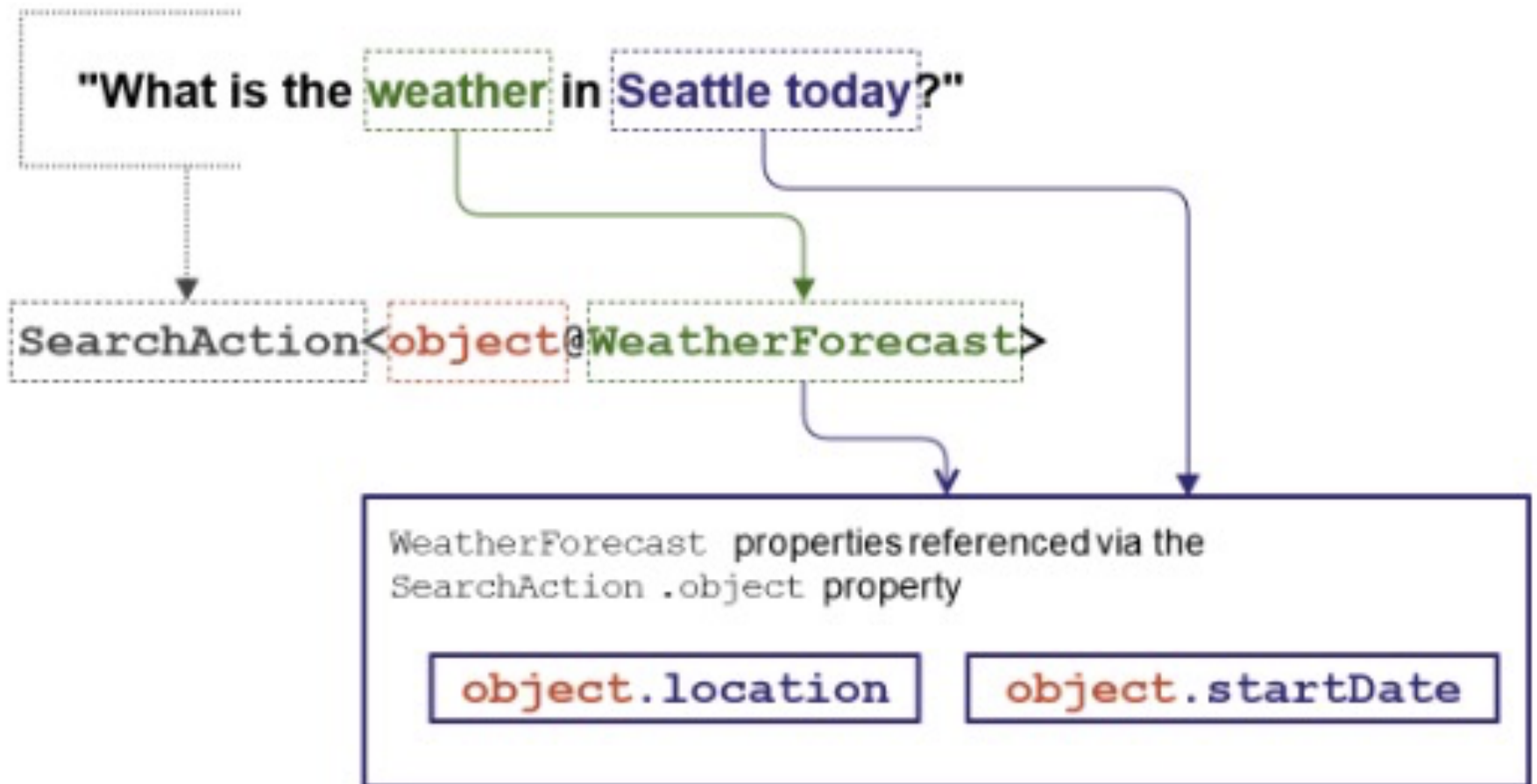
WeatherForecast[weatherCondition]のslot

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

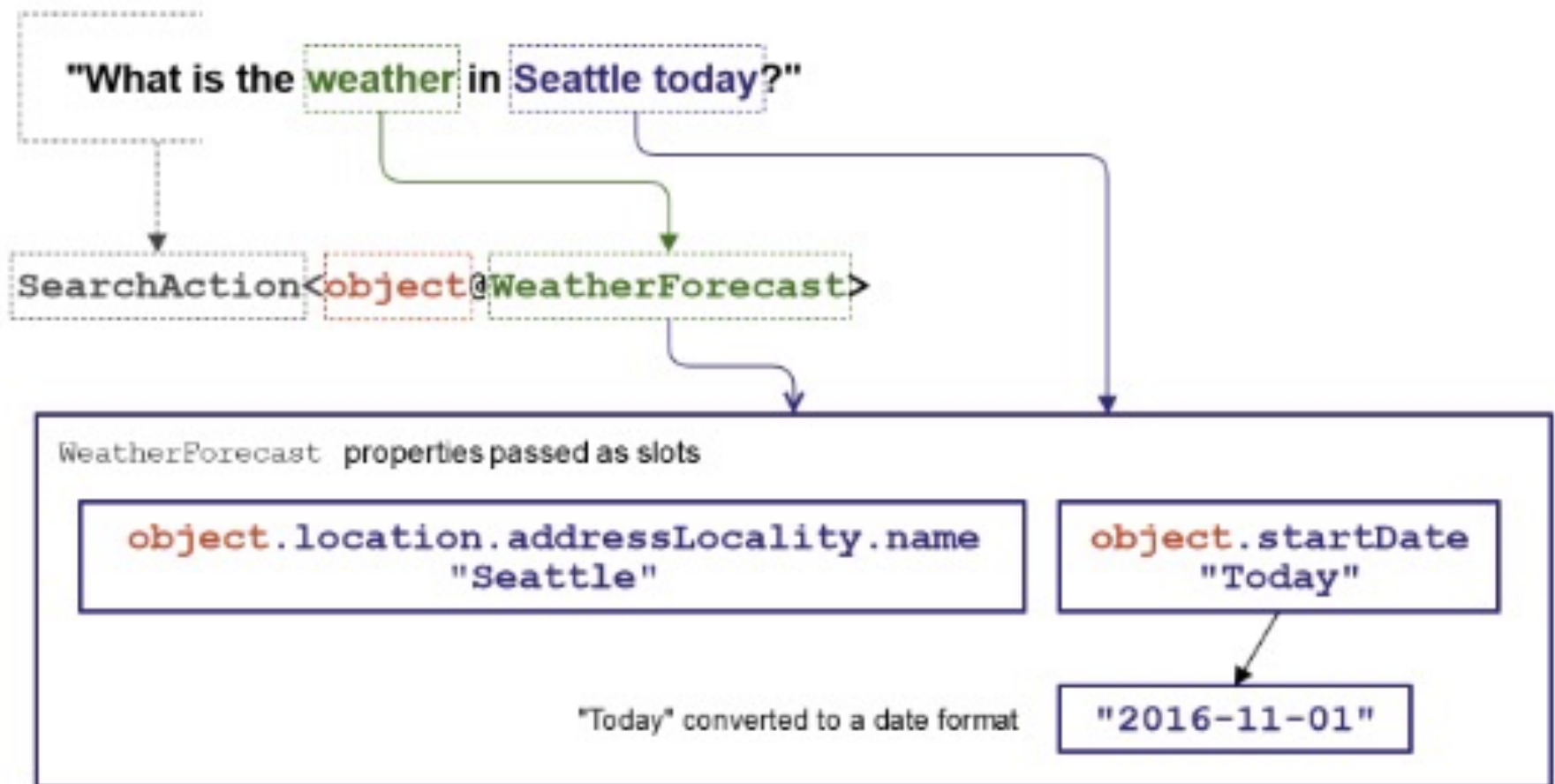
Intent Signatureが、Intentの名前として使われる時、Slotは具体的な値で埋められてから、Requestとして送られる

```
{
  "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);
```

Schema.orgでのActionの定義

<http://schema.org/Action>

Announcing Schema.org Actions

2014/04/16

When we launched schema.org almost 3 years ago, our main focus was on providing vocabularies for describing entities --- people, places, movies, restaurants, ... But the Web is not just about static descriptions of entities. It is about taking action on these entities --- from making a reservation to watching a movie to commenting on a post.

Today, we are excited to start the next chapter of schema.org and structured data on the Web by introducing vocabulary that enables websites to describe the actions they enable and how these actions can be invoked.

Announcing Schema.org Actions

2014/04/16

The new actions vocabulary is the result of over two years of intense collaboration and debate amongst the schema.org partners and the larger Web community. Many thanks to all those who participated in these discussions, in particular to members of the Web Schemas and Hydra groups at W3C. We are hopeful that these additions to schema.org will help unleash new categories of applications.

Jason Douglas, Sam Goto (Google)

Steve Macbeth, Jason Johnson (Microsoft)

Alexander Shubin (Yandex)

Peter Mika (Yahoo)

schema.org でのPersonのProperty

Properties = “has-a”

- ☐ gender
- ☐ birthdate
- ☐ birthplace
- ☐ profession
- ☐ nationality
- ☐ ethnicity
- ☐ parents
- ☐ children
- ☐ religion

- ☐ education
- ☐ jobTitle

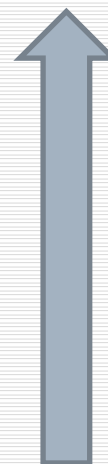
- ☐ **follows**
- ☐ **knows**
- ☐ **owns**
- ☐ **seeks**

... and more

Action

Schema.orgでのActionの下位タイプ

- | | |
|--|--|
| ☐ <u>AchieveAction</u> | ☐ <u>OrganizeAction</u> |
| ☐ <u>AssessAction</u> | ☐ <u>PlayAction</u> |
| ☐ <u>ConsumeAction</u> | ☐ <u>SearchAction</u> |
| ☐ <u>ControlAction</u> | ☐ <u>TradeAction</u> |
| ☐ <u>CreateAction</u> | ☐ <u>TransferAction</u> |
| ☐ <u>FindAction</u> | ☐ <u>UpdateAction</u> |
| ☐ <u>InteractAction</u> | |
| ☐ <u>MoveAction</u> | |



Intent Signatureで利用されているのは、これ

AMAZON.**SearchAction**<object@WeatherForecast>

ActionのProperty としてのobject

- The object upon which the action is carried out, whose state is kept intact or changed. Also known as the semantic roles patient, affected or undergoer (which change their state) or theme (which doesn't). e.g. John read a book

~~AMAZON.SearchAction<object@WeatherForecast>~~

Property	Expected Type	Description
Properties from Action		
actionStatus	ActionStatusType	Indicates the current disposition of the Action.
agent	Organization or Person	The direct performer or driver of the action (animate or inanimate). e.g. <i>John</i> wrote a book.
endTime	DateTime	The endTime of something. For a reserved event or service (e.g. FoodEstablishmentReservation), the time that it is expected to end. For actions that span a period of time, when the action was performed. e.g. John wrote a book from January to <i>December</i>. Note that Event uses startDate/endDate instead of startTime/endTime, even when describing dates with times. This situation may be clarified in future revisions.
error	Thing	For failed actions, more information on the cause of the failure.
instrument	Thing	The object that helped the agent perform the action. e.g. John wrote a book with <i>a pen</i> .
location	Place or PostalAddress or Text	The location of for example where the event is happening, an organization is located, or where an action takes place.
object	Thing	The object upon which the action is carried out, whose state is kept intact or changed. Also known as the semantic roles patient, affected or undergoer (which change their state) or theme (which doesn't). e.g. John read <i>a book</i> .
participant	Organization or Person	Other co-agents that participated in the action indirectly. e.g. John wrote a book with <i>Steve</i> .
result	Thing	The result produced in the action. e.g. John wrote <i>a book</i> .
startTime	DateTime	The startTime of something. For a reserved event or service (e.g. FoodEstablishmentReservation), the time that it is expected to start. For actions that span a period of time, when the action was performed. e.g. John wrote a book from <i>January</i> to December. Note that Event uses startDate/endDate instead of startTime/endTime, even when describing dates with times. This situation may be clarified in future revisions.
target	EntryPoint	Indicates a target EntryPoint for an Action.

Showing results for ***Weather Forecast***

Search instead for ***WeatherForecast***

Report – schema.org

It is a subproperty of contentLocation intended primarily for more technical and detailed materials. For example with a Dataset, it indicates areas that the dataset describes: a dataset of New York **weather** would have spatialCoverage which was the place: the state of New York. Supersedes spatial. sponsor · Organization or

schema.org/Report

Schema.orgでの「文」の解釈


```
<!-- John listened to Pink with Steve at Anna's apartment on his iPod. -->
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "ListenAction",
  "agent": {
    "@type": "Person",
    "name": "John"
  },
  "object": {
    "@type": "MusicGroup",
    "name": "Pink!"
  },
  "participant": {
    "@type": "Person",
    "name": "Steve"
  },
  "location": {
    "@type": "Residence",
    "name": "Ann's apartment"
  },
  "instrument": {
    "@type": "Product",
    "name": "iPod"
  }
}
</script>
```

*John listened to Pink with Steve
at Anna's apartment on his iPod.*

<!-- John created an exercise plan with Steve. -->

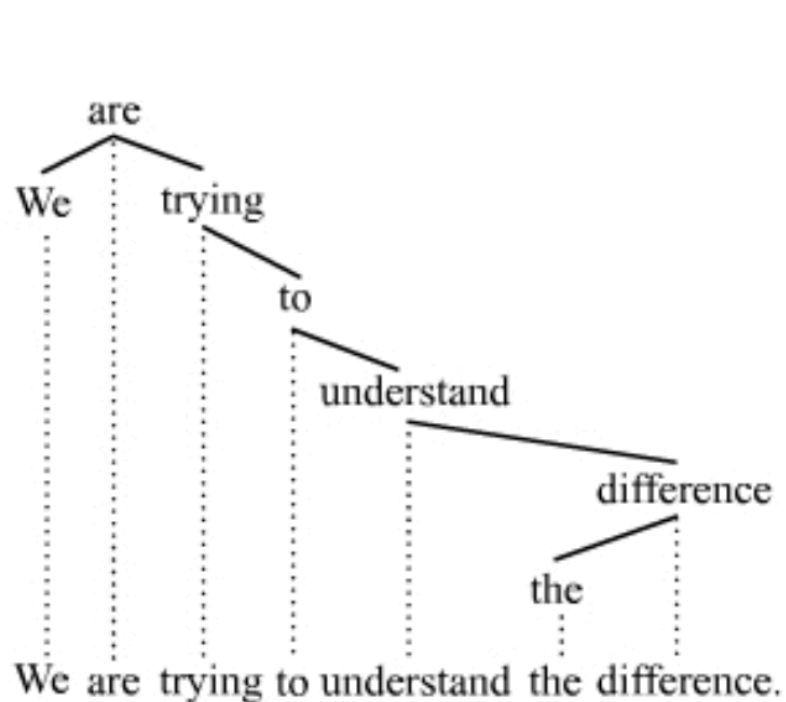
<script type="application/ld+json">

```
{
  "@context": "http://schema.org",
  "@type": "CreateAction",
  "agent": {
    "@type": "Person",
    "name": "John"
  },
  "result": {
    "@type": "ExercisePlan",
    "name": "John's weight loss plan"
  },
  "participant": {
    "@type": "Person",
    "name": "Steve"
  }
}
```

</script>

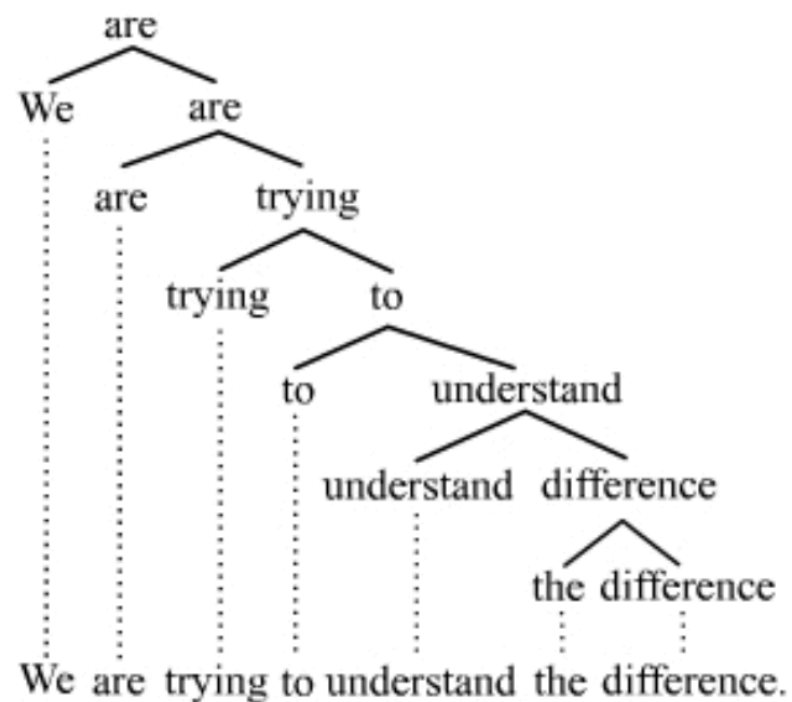
John created an exercise plan with Steve

文法理論での、 「文」の中心としての「動詞」の優位



Dependency

従属文法



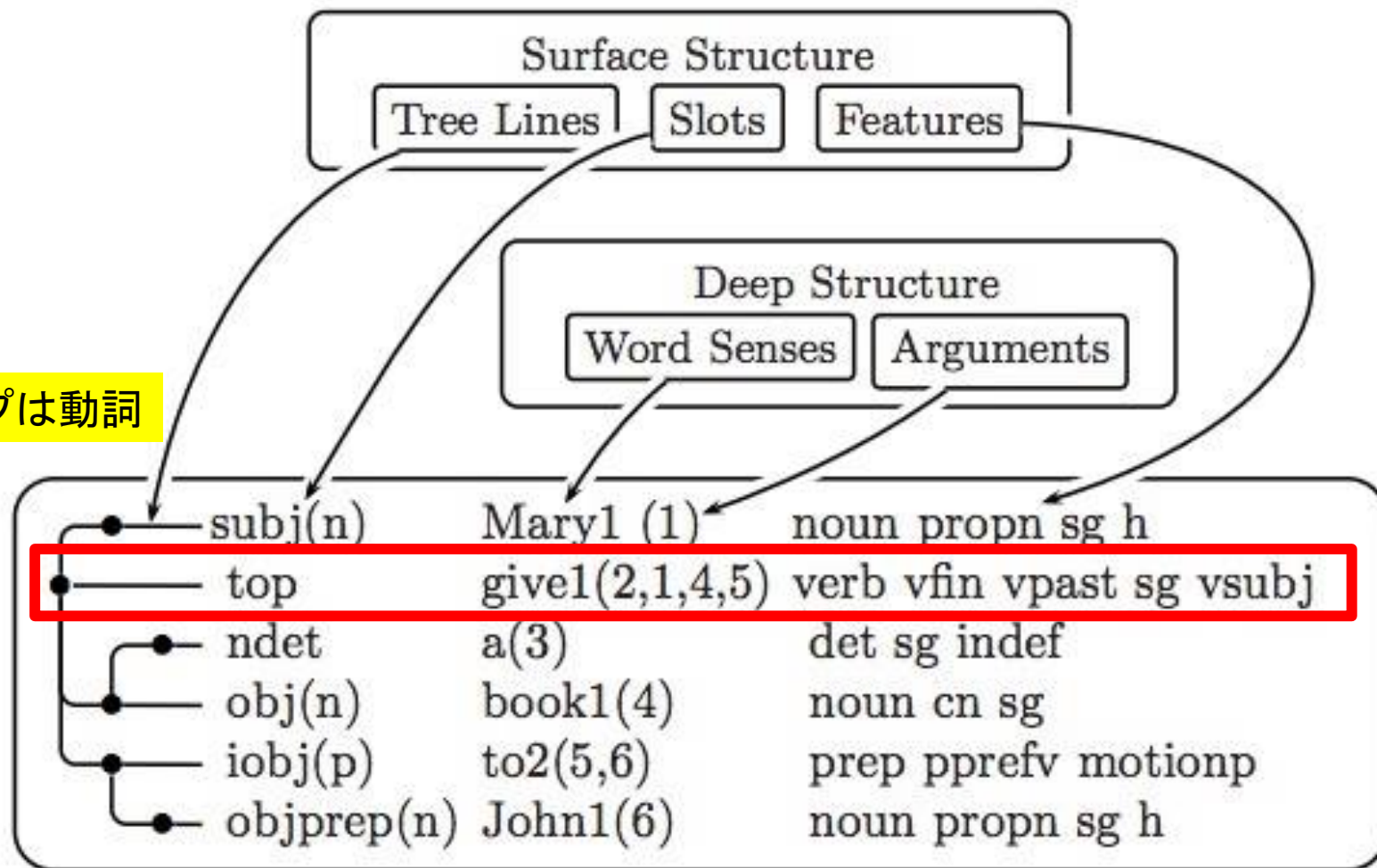
Constituency (BPS)

Minimalist Program

<https://goo.gl/d3LxPn>

IBM WatsonのESGでの “Mary gave a book to John.” の解析

トップは動詞



Appendix



Appendix

1. IBM Watson(旧)の取り組みから学ぶ
2. ボイス・アシスタントの
Smart Home / IoTへの応用

IBM Watson(旧)の取り組みから学ぶ

ESG(English Slot Grammar)

Watsonの自然言語理解のベースになっているのは、Michael C. McCordが開発したSlot Grammarという枠組みである。

Slot Grammarの基本文献

□ 次の三つの論文が、Slot Grammarの基本的文献である。

- “A Formal System for Slot Grammar”
<http://goo.gl/KR56Es>
 - “The Slot Grammar Lexical Formalism”
<http://goo.gl/57vxL4>
 - “Using Slot Grammar”
<https://goo.gl/P1ZgcQ>
-

Slotとは？

- Slotというのは、基本的には、文中の句の**文法的な役割**を表している。いくつかのSlotの例を挙げる。

subj	subject（主語）
obj	direct object（直接目的語）
iobj	indirect object（間接目的語）
comp	predicate complement（述語補語）
objprep	object of preposition（前置詞目的語）
ndet	NP determiner（名詞句限定子）

- Slotのもう一つの役割は、**意味論的な役割**である。
例えば、compは、述語の「位置を持つ引数」として、言葉の意味を表す。
-

Slot Grammarでの文法的解析

- 次の図は、“Mary gave John a book”を、Slot Grammarで解析したものである。
- 文法的には、スロット subjは、“Mary”で「満たされ」、スロット iobjは、“John”で「満たされ」ている。等々。こうした観点からは、スロットが文法的な役割を果たしていることがわかる。

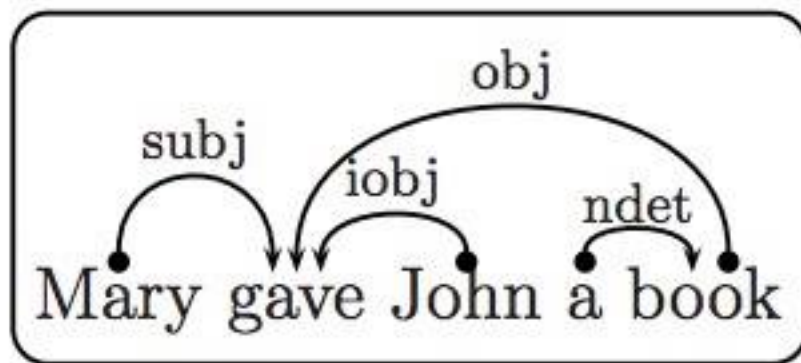


Figure 1: Slot filling for *Mary gave John a book*

Slot Grammarでの意味的解析

- 意味論的な観点からスロットの役割を見るには、giveの語義を考えよう。それを論理的に表現したものを give1とすると、**give1(e, x, y, z)**は、「イベントeで、xはyにzを与える」を意味する。
 - この時、先の図の文章の論理的な表現は、
$$\exists e \exists y (\text{book}(y) \wedge \text{give1}(e, \text{Mary}, y, \text{John}))$$
になる。
 - こうした論理的な観点からは、スロット subj, obj, iobj は、述語**give1**の引数 x, y, z の名前とみなすことができる。こう考えてもいい。これらのスロットは、動詞の意味を表す述語の位置を持つ引数だと。
-

Complement Slotと Adjunct Slotと

- このように述語の引数になるスロットは、complement Slot と呼ばれ、Slot Grammar Lexicon(スロット文法語彙集)という辞書を通じて、語義に関連づけられる。
 - 一方、それ以外のスロット、ndet(冠詞・定冠詞)のようなスロットは、adjunct Slotと呼ばれて、Slot Grammarのsyntaxに関連することになる。
-

Slot Grammar Lexicon

- Slot Grammarで重要な役割を担っているのは、その辞書、Lexicon 語彙集である。ここでは、次の論文に従って、その形式 SLF(Slot Grammar Lexical Formalism)を紹介しよう。"The Slot Grammar Lexical Formalism"<http://goo.gl/57vxL4>
- まず、もっとも簡単な形式は、次のようになる。これは、“access” という語の登録形式である。

access

< v obj

< n (p to)

Slot Grammar Lexicon

- ここで、見出語は、accessである。この見出語について二つの要素 v obj と n (p to) が登録されている。vはverbで「動詞」を表し、nはnounで「名詞」を表している。これらは、見出語の「品詞」にあたる。この表現は、accessは、動詞と名詞があることを示している。

access

< v obj

< n (p to)

見出語

v は動詞

n は名刺

- 品詞 v, n のあとに続く obj, (p n)は、スロットを表している。
-

Slot Grammar Lexicon

□ 次のエントリー

give < v obj iobj

は、giveは動詞(v)で、二つのスロット obj と iobj を取ることを表している。

Alice gave the book to Bob.

Alice gave Bob the book.

Alice gave the book to Bob.
 obj iobj

Alice gave Bob the book.
 iobj obj

- 最初の文では、スロット obj は、名詞句(NP) the book で満たされ、スロット iobj は前置詞句(PP) to Bobで満たされている。二つ目の文では、スロット iobj は、異なる位置で、名詞句 Bobで満たされ、スロット obj は、やはり異なる位置で、名詞句 the bookで満たされている。
- 二つの文で、iobjを満たしているものは、NPとto-PPで異なるのだが、満たしているものの論理的役割は同じなので、二つの文、どちらでも、Bobが本を与えられた人物であることがわかる。

Alice gave the book to Bob.
 obj iobj

Alice gave Bob the book.
 iobj obj

Lexconのエントリーの例

admonish < v obj (comp (p about against for))

admonish < v obj (p about against for)

admonition < n obj (p about against for)

drive < obj (comp lo)

She drove her car out .
 obj (comp lo)

She drove her car onto the interstate .
 obj (comp lo)

She drove her car fifty miles .
 obj (comp lo)

意味フレーム

- Slot GrammarのLexiconでは、
見出語 < 品詞 スロット の後ろに、その語の「特徴」を表す情報が並ぶ。
- その語についてのその他の付加的な情報も、この後ろに並ぶ。Slot Grammar Lexiconでは、こうした記述をSense Frame「意味フレーム」と呼んでいる。

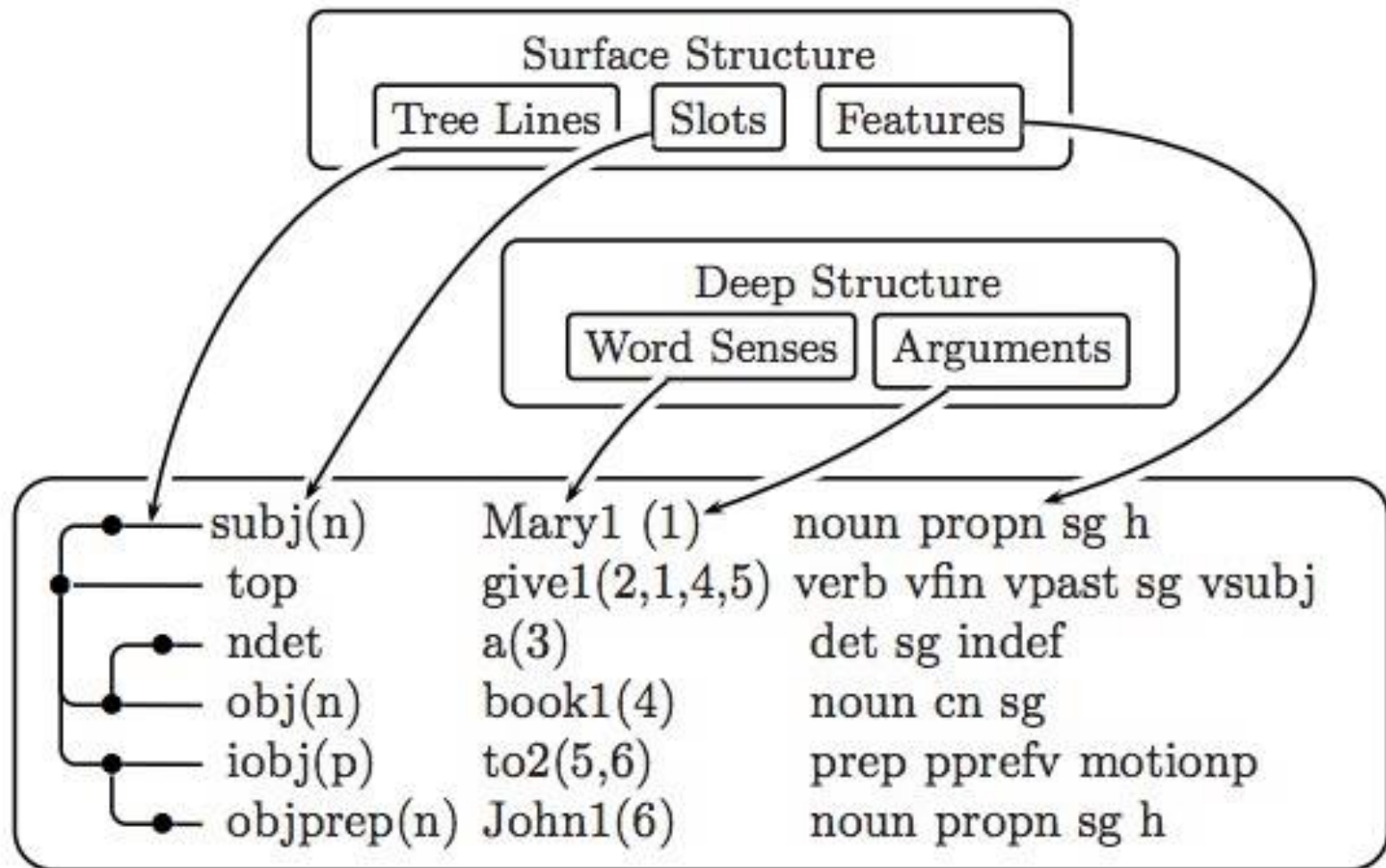
見出語 < 品詞 スロット 諸特徴 その他の情報

Features

見出語 < 品詞 スロット 諸特徴 その他の情報

- 「諸特徴 (Features)」と簡単に書いたが、品詞ごとに、動詞についてのFeatures、名詞についてのFeatures、形容詞についてのFeatures、副詞についてのFeaturesと、数多くのFeaturesがある。
- 先の論文、“The Slot Grammar Lexical Formalism” <http://goo.gl/57vxL4> の多くの部分は、このFeaturesの記述に当てられている。
- Featuresの詳細については、スペースの関係で、ここでは触れない。次に、その一部のリストを示す。

“Mary gave a book to John.” の解析

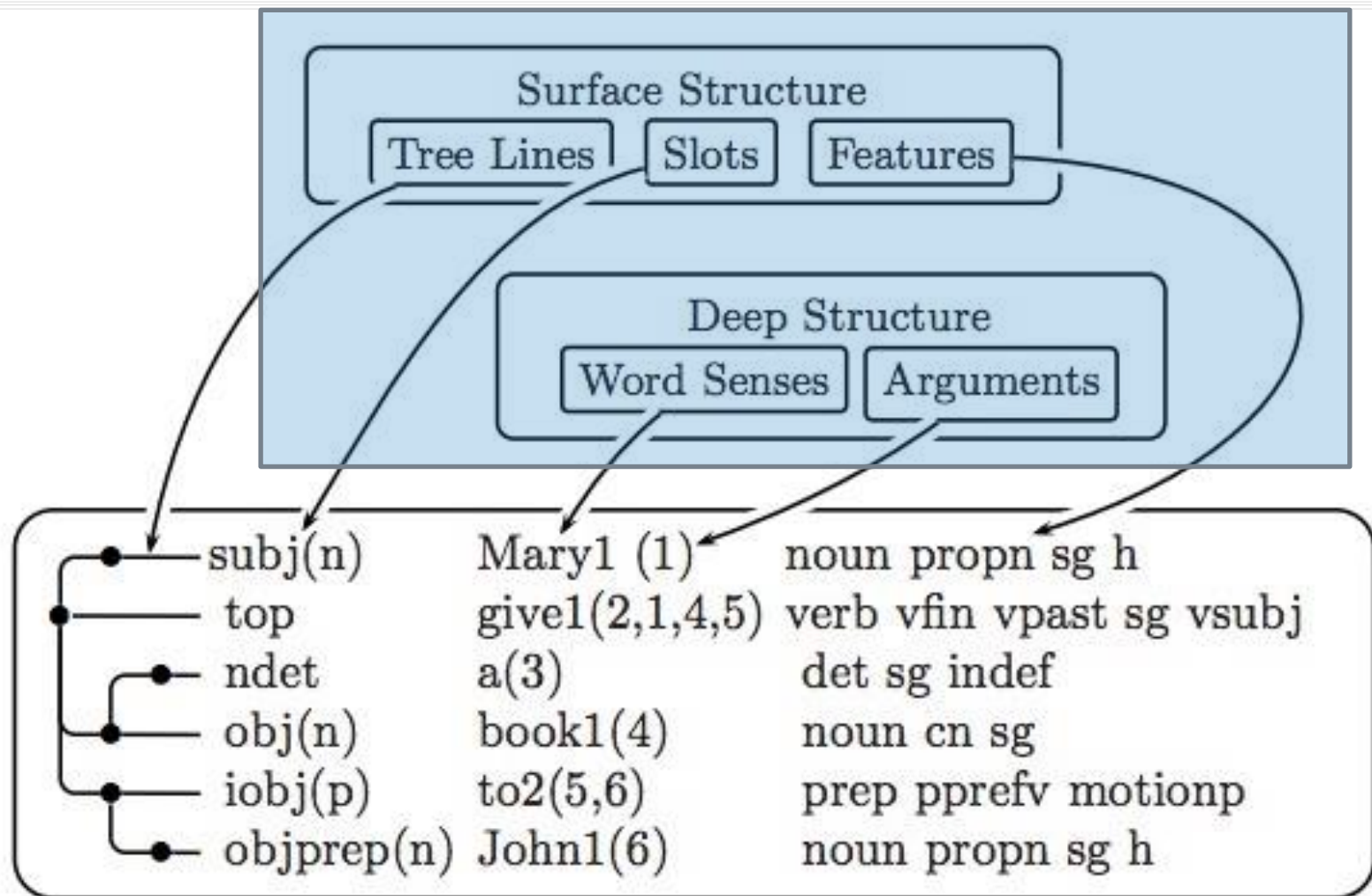


「表層構造 (Surface Structure)」と「深層構造 (Deep Structure)」

- 先の図で重要なのは、文法的な構造を表す「表層構造 (Surface Structure)」と意味を担う「深層構造 (Deep Structure)」の二つから文を解析しようとしているところだ。(もっとも、これは50年以上前のChomsky以来の古典的な観点だ)
 - もう少し、詳しく見てみよう。
Slot Grammarで「表層構造 (Surface Structure)」を担っているのは、"Tree Line"と"Slot"と"Features"だとされている。一方で、「深層構造 (Deep Structure)」を担っているのは、"Word Senses"と"Arguments"だとされる。
-

“Mary gave a book to John.”

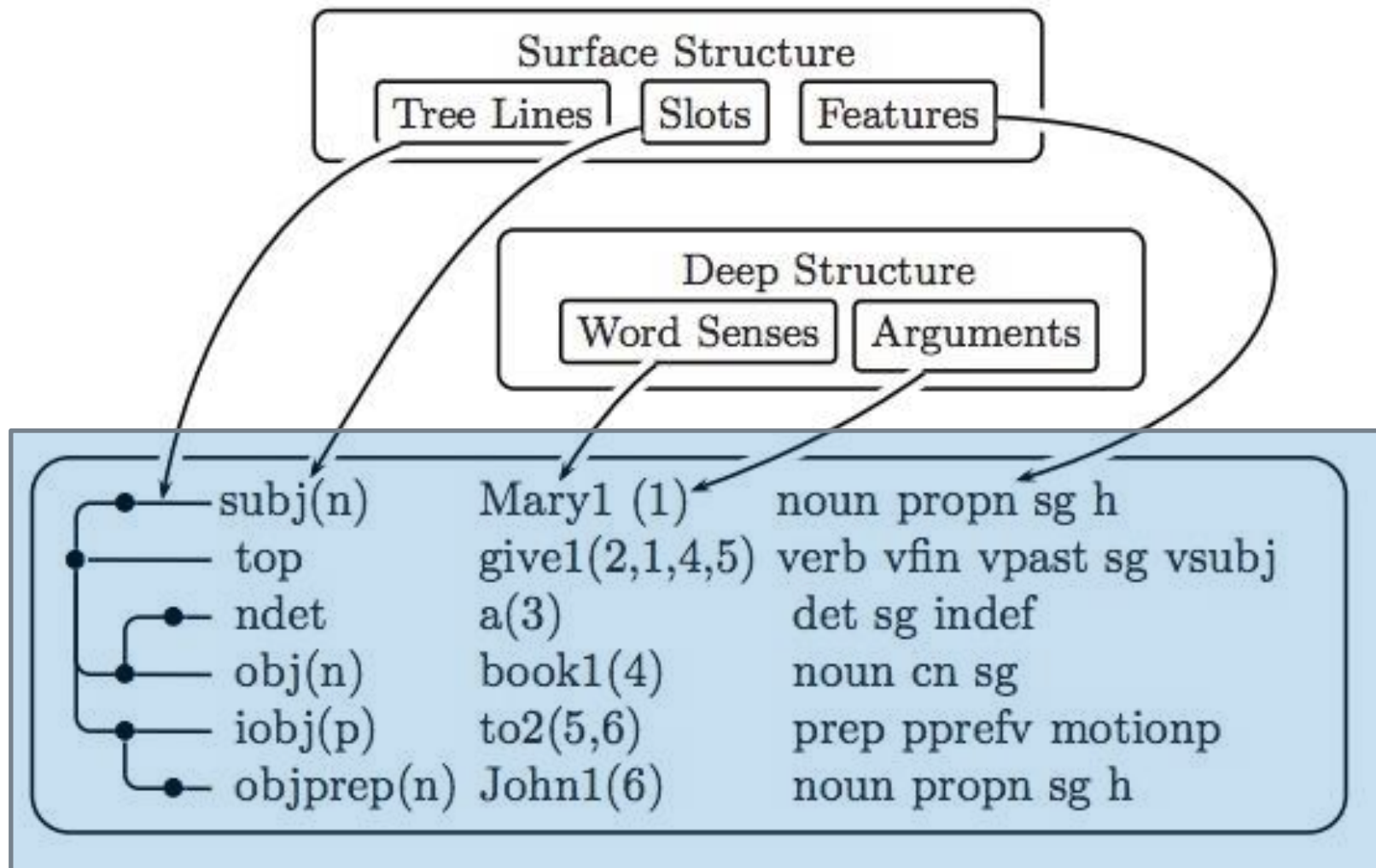
表層構造・深層構造



- 今度は、この図の中心をなす下半分に注目しよう。
この領域は、大きく、三つの部分からなる。
- 一番左は、Slot単位の構文解析木 Tree Line である。これは、文法的な「表層構造 (Surface Structure)」を表している。
- 中央部分は、語義 Word Sense だが、それぞれは「原形」としてLexiconの「見出し語」に対応している。Lexiconの「意味フレーム (Sense Frame)」は、「見出し語」に対して、Slot を「引数」として対応づける情報を与える。ただ、ここで第一引数として与えられているのは、それぞれの語の文中での出現順位である。
- 結局、以前に見たように、次の論理式が、この文の「意味」を表現することになる。
$$\exists e \exists y (\text{book}(y) \wedge \text{give1}(e, \text{Mary}, y \text{ John}))$$
- この図の右側は、それぞれの語のFeatureを表している。Featureは、Lexiconの「意味フレーム (sense Frame)」で与えられる。

“Mary gave a book to John.”

Tree Line + Word Sense + Features



Featureサンプル

- Mary のFeatureは、“noun propn sg h”だが、それぞれのFeatureの意味は、

noun: 「名詞」

propn: proper noun で「固有名詞」

sg: single で「単数」

h: humanで、「人間」

いずれも、辞書には出てきそうな情報である。

- giveのFeatureは、“verb vfin vpast sg vsubj” だが、それぞれのFeatureの意味は、

verb: 「動詞」

vfin: finite verb で「定動詞」

vpast: 「動詞過去形」

sg: 「単数」 である。

Deep parsing in Watson

「Watsonでの深いパーズング」

*M. C. McCord, J. W. Murdock, and B. K.
Boguraev*

<http://goo.gl/VLuWKc>

"Deep parsing in Watson"

- この論文の著者の一人 M. C. McCord は、これまで紹介してきた Slot Grammarの創始者である。
- 「文章を深く解析する二つのコンポーネント、英文スロット文法パーザー(English Slot Grammar (ESG) parser)と述語引数構造ビルダー(predicate-argument structure (PAS) builder)は、IBM Watsonで利用される質問と文字情報の双方に、答えを見つけ仮説を提供する、核心的な言語分析の能力を提供している。特に、これらのコンポーネントは、質問の分析・回答候補の生成・文中の証拠の分析に基本的な役割を果たしている。

"Deep parsing in Watson"

- Watsonプロジェクトの一部として、ESGは機能拡張され、Jeopardy!の質問や、あらかじめ用意された参照データに対するそのパフォーマンスは改善された。PASは、ESGの上に、より高度な分析を支援するために構築された。この論文では、これらのコンポーネントについて述べ、それらがWatsonのパターン・ベースの関係抽出コンポーネントの中でどのように利用されているかを示そうと思う。また、ESGパーズングのコンポーネント・レベルでのパフォーマンス評価の定量的な結果も提供したい。」

PAS(predicate-argument structure) ビルダー

□ 次のような文を考えよう。

- I heard that Edison invented the phonograph in 1877.
- I heard that Edison invented a phonograph in 1877.

□ この二つの文は、違った意味を持つ。そのことは、ESGでの解析では正確に反映される。ただ、PASでは、この区別は捨象される。

□ 次の例でも、EGBでは異なる解析が得られるが、PASでは同じである。

- Odysseus was bold and clever.
 - bold and clever Odysseus.
-

ESGパーザーとPASビルダー

- このように、PASビルダーは、ESGパーザーの解析結果の細部を省略して、単純化・抽象化する。これによって、失われる「意味」や「ニュアンス」もあるのだが、こうした単純化した形式で意味を抽出したほうが、大量のデータとして扱うには効率的である。
 - 次の図は、Jeopardyの問題
“Chandeliers look great but nowadays do not usually use these items from which their name is derived.”
を、ESGで解析したものである。
-

“Chandeliers look great but nowadays do not usually use these items from which their name is derived.”

ESGでの解析

	subj(n)	chandelier(1)	noun cn pl physobj artf
	lconj	look(2,1,3)	verb vfin vpres pl sta
	comp(a)	great(3,1,u)	adj erest
	top	but(4)	verb vfin vpres pl cord
	vadv	nowadays(5,6)	adv
	rconj	do(6,1,9)	verb vfin vpres pl
	vadv	not(7,6)	adv ppadv nounadv neg
	vadv	usually(8,9)	adv
	auxcomp(binfn)	use(9,1,11,u)	verb vinf vpref ssa
	ndet	these(10)	det pl def
	obj(n)	item(11,u)	noun cn pl
	comp(p)	from(12,17,13)	prep wh
	objprep(n)	which(13,11,u)	noun pron wh
	ndet	their(14)	det sg possdet
	subj(n)	name(15,u,u)	noun cn sg langunit
	nrel	be(16,15,17)	verb vfin vpres sg
	pred(en)	derive(17,u,15,12)	verb ven vpass

“Chandeliers look great but nowadays do not usually use these items from which their name is derived.”

PASでの解析

```
chandelier (1)
look (2, subj:1, comp:3)
great (3)
but (4, lconj:2, rconj:9) [ top predicate]
nowadays (5)
not (7)
usually (8)
use (9, subj:1, obj:11, vadv:5, vadv:7,
    vadv:8)
item (11, nrel:17) [ determiner: these]
from (12, objprep:13)
which (13)
their (14)
name (15, ndet:14) [ determiner: their]
derive (17, obj:15, comp:12)
```

PASでの解析結果はESGでの解析結果より、シンプルである。ただ、述語と引数の関係を捉えることで、意味の深い理解に進んでいる。

Automatic knowledge extraction from documents

「ドキュメントからの自動知識抽出」

J. Fan, A. Kalyanpur, D. C. Gondek, and D. A. Ferrucci

<http://goo.gl/MexfSI>

ドキュメントからの知識の抽出

- Watsonでは、質問(クイズで言えば「ヒント」)の文解析と、大量のドキュメント・データから「知識」を抽出する文解析とは、多少異なる手法が利用されている。
- 質問(ヒント)の解析には、これまで見たきたように、「英文スロット文法パーザー(English Slot Grammar (ESG) parser)」と「述語引数構造ビルダー(predicate-argument structure (PAS) builder)」を使うのだが、ドキュメントから「知識」を抽出するのに利用されるシステムは、「PRISMATIC」と呼ばれている。
- Watson論文集中の、次の論文がこの問題に当てられている。
"Automatic knowledge extraction from documents「文書からの自動的な知識の抽出」"
<http://goo.gl/MexfSI>

ドキュメントからの知識の抽出

- 「IBM WatsonのようなDeepQAシステムにとって、どんな領域の話題にも対応して質問に答えることに成功する為には、非常に膨大な知識にアクセスすることが、本質的に重要である。
 - 形式化された知識の表現は、それによって推論が容易になるという利点があるのだが、オープンな領域で構造化されていないデータから構造化された知識を獲得することは、しばしば困難であり、コストがかかりすぎる。
 - 我々の中心的な仮説は、浅い文法的な知識とそれが含意する意味は、容易に獲得でき、かつ、質問応答システムの多くの領域で利用できるということである。
-

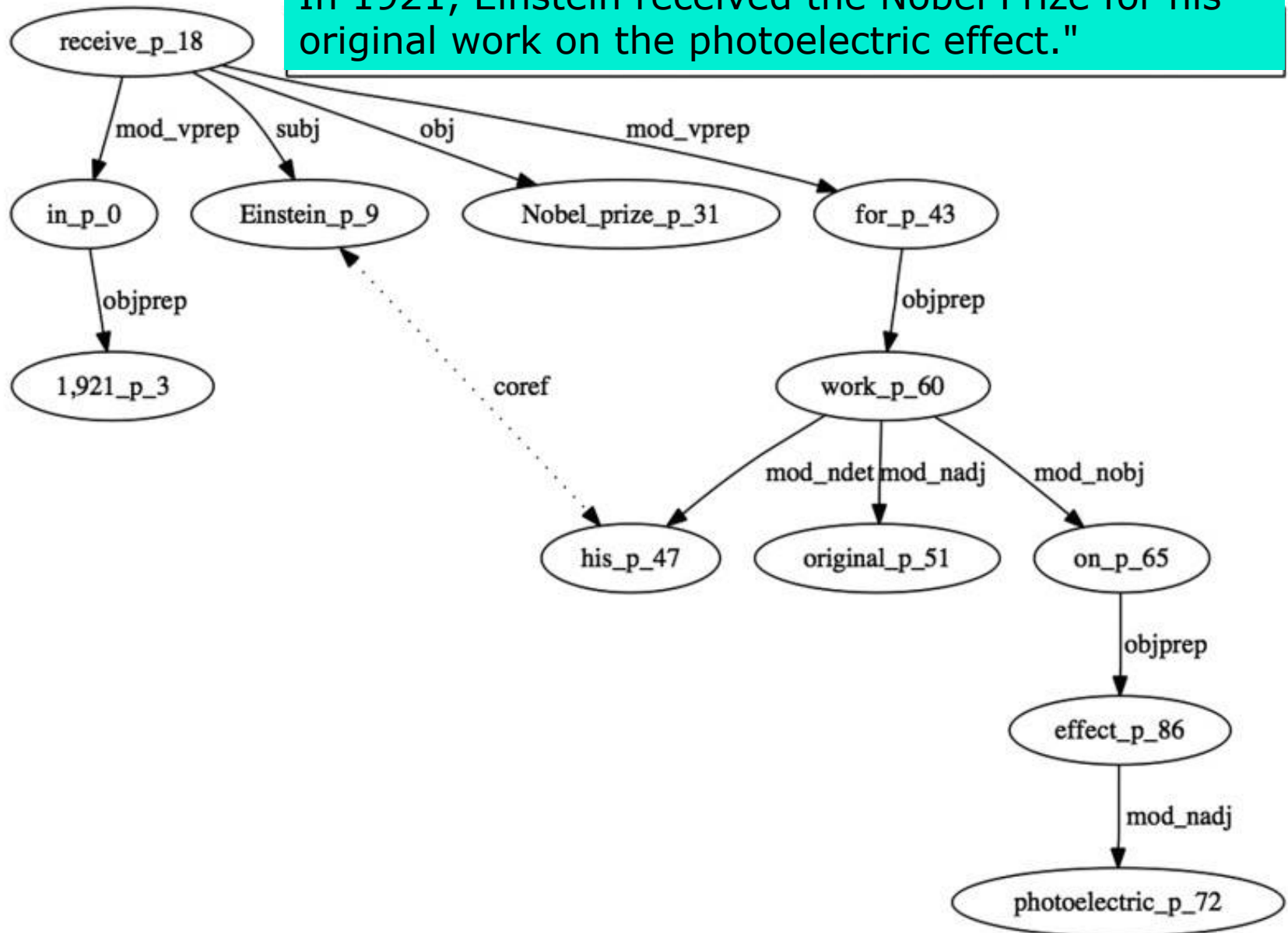
ドキュメントからの知識の抽出

- 我々は、文法的な知識とそれが含意する意味の抽出を二段階で行う。第一に、大規模なドキュメントの集まりからの浅い知識が、自動的に抽出される。第二に、自動的に抽出された浅い知識から集約された統計情報から、追加的な意味を推論する。
 - この論文では、どのような浅い知識が抽出され、それがどのように大規模なコーパスから自動的になされるのか、そして、集約された統計情報から、どのようにして追加的な意味が推論されるのかを詳述する。同時に、知識抽出の様々な方法がIBMのDeepQAシステム全体の中で、どのように利用されているかについて簡単に述べたい。」
-

PRISMATIC

- PRISMATICについては、内容が重なっているが、次の論文がある。"PRISMATIC: Inducing Knowledge from a Large Scale Lexicalized Relation Resource" <http://goo.gl/AsLr94>
 - PRISMATICが、次の文章 "In 1921, Einstein received the Nobel Prize for his original work on the photoelectric effect."を解析した様子を最初の図に示す。ベースになっているのは、Slot Grammer だが、スロットの種類が ESGパーザーとは若干異なっている。ESGパーザーのスロットと基本的なFeatureが組み合わせてスロットを拡張したように見える。
-

In 1921, Einstein received the Nobel Prize for his original work on the photoelectric effect."



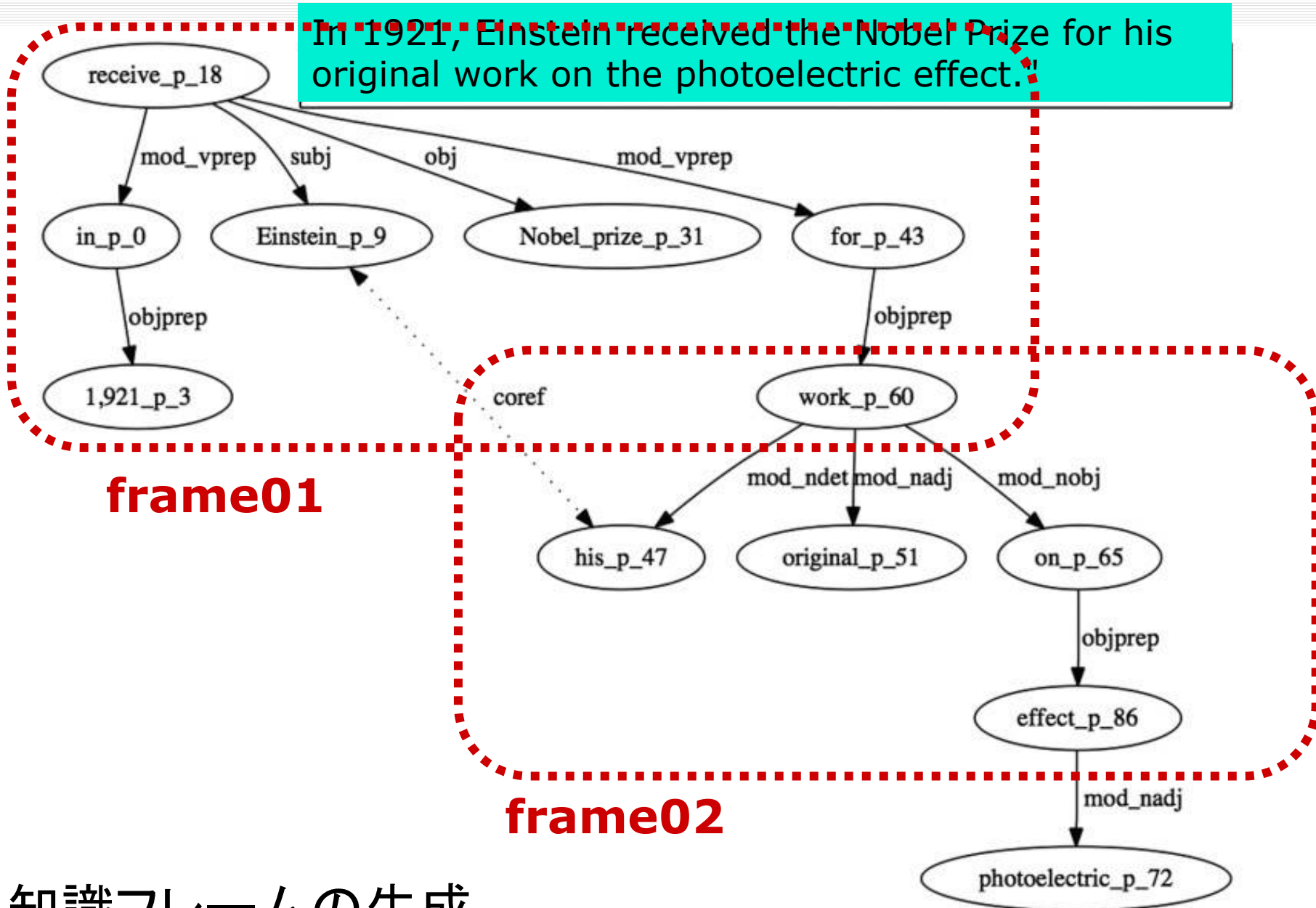
PRISMATIC

- また、以前に見た「質問」の解析では、ESGパーザーでの解析に続く「述語引数構造 (predicate-argument structure (PAS))」の抽出が目標だったのだが、ここではそうした二段階の処理は行われていない。
-

知識フレームの生成

- 二つ目の図は、元の文から、“In 1921, Einstein received the Nobel Prize for Frame02”というFrame01と、“his original work on effect.” というFrame02の二つのフレームが作られることを表す。（“photoelectric”が、切り捨てられてしまっているが。）
-

In 1921, Einstein received the Nobel Prize for his original work on the photoelectric effect.



知識フレームの生成

PRISMATICが生成した知識フレーム

Frame01

<i>verb</i>	receive
<i>subj</i>	Einstein
<i>type</i>	PERSON/SCIENTIST
<i>obj</i>	Nobel prize
<i>mod_vprep</i>	in
<i>objprep</i>	1921
<i>type</i>	YEAR
<i>mod_vprep</i>	for
<i>objprep</i>	Frame02

Frame02

<i>noun</i>	work
<i>mod_ndet</i>	his/Einstein
<i>mod_nobj</i>	on
<i>objprep</i>	effect

膨大な知識フレームの生成

- PRISMATICは、Wikipediaを始めとする30GBのドキュメント・データから、こうして、10億個もの知識フレームを生成しているという。すごい。「質問」の解析は、リアルタイムになされなければいけないのだが、こうした知識フレームの生成は、もちろん、事前に行われている。
-

知識フレームと統計情報

- この膨大な知識フレームの集積から、どのように統計情報が引き出されるかを見てみよう。
 - 例えば、"**Napoleon annexed Piedmont in 1859**「ナポレオンは、1859年に、ピーモンドを併合した」" という文章から生成されるフレームを考えてみよう。基本的には、S-V-Oの形のフレームなのだが、こうした形で、動詞 "annex"（「併合する」）を含む多数のフレームを集めると、annexは、「地域」を型に持つ目的語を取ることが多いことが、統計的にわかる。
-

知識フレームと統計情報

- 今度は、次のような文を考えよう。"He ordered a Napoleon at the restaurant." ここでのNapoleonは、もちろん、先の文章でのNapoleonとは違うのだが、そのことは、次のようにして推論できる。
 - "Napoleon"の型が分からなかったら、膨大な知識フレームの中から、この文章から生成されるフレームと同じ(ただし"Napoleon"の部分の型は「不明」として)次のような形をしているフレームを検索する。そうすると、このorderの目的語の型は、統計的には「食べ物」だということがわかる。
-

統計情報をLATの推定に利用

```
{  
  <subject type,"Person">  
  <verb,"order">  
  <object type,"?">    <== "Napoleon"の型は？  
  <verb prep,"at">  
  <prep,"restaurant">  
}
```

- 質問応答システムでは、質問が要求している答えの文法的な型(これを LAT “Lexical Answer Type” という)を推定することは、とても重要になるのだが、ここでも、この統計的な情報は役にたつ。
-

Question analysis: How Watson reads a clue

「質問の分析: Watsonは、どのようにヒントを読むのか？」

A. Lally, J. M. Prager, M. C. McCord, B. K. Boguraev, S. Patwardhan, J. Fan, P. Fodor, and J. Chu-Carroll

http://brenocon.com/watson_special_issue/02%20question%20analysis.pdf

Watsonの構文解析と意味解析の能力

- IBM Watsonシステムの処理の最初のステージは、何が求められているのか、そして、それに答えるにはどのようなアプローチが最良なのかを決定するために、質問の詳細な分析を行うことである。質問の分析は、Watsonの構文解析と意味解析の次のような能力を利用する。深いSlot Grammarのパパーザー、名前を持つエンティティの認識、参照を解決するコンポーネント、関係を抽出するコンポーネントである。
- 我々は、この分析から得られる諸特徴を用いて、質問中の本質的に重要な要素を検出するために、様々な検出ルールとクラス分けの手法を適用する。質問中の重要な要素としては、次のものがある。

POETS & POETRY: He was a bank clerk in the Yukon before he published "Songs of a Sourdough" in 1907.

- **Focus** は、質問の一部で、答えを指しているものである。この上のサンプルでは、**Focus**は、"He" である。
 - **LATs** (Lexical Answer Type)は、質問中の語句で、どのような型のエンティティが質問されているかを示すものである。Focusの主辞(見出語)は、一般的にはLATである。しかし質問には、そのほかにもLATを含むことがしばしばある。Jeopardy!の領域では、質問のカテゴリーが、こうした追加的なLATの源になる。この例では、"he", "clerk", "poet" がLATである。
-

POETS & POETRY: He was a bank clerk in the Yukon before he published "Songs of a Sourdough" in 1907.

- **Question Classification** は、質問を、一つあるいは複数の、もっと広い型に属するものとみなす。この例では、この質問は、最も一般的なクラス、**Factroid**に属する。
 - **QSections** は、その解釈が特別な処理を必要とする、質問の断片である。QSectionの最も重要な利用は、一つは、答えの構文上の制約条件（例えば、「4文字」とか「3語」といった）であり、もう一つは、質問をいくつかの下位の質問に分解することである。
-

focus と LAT 検出の基本

- 限定詞 “this” あるいは “these” を伴った名詞句:
 - THEATRE: A new play based on *this Sir Arthur Conan Doyle canine classic* opened on the London stage in 2007.
 - 代名詞としての This あるいは these :
 - '88: In April 1988, Northwest became the first U.S. air carrier to ban *this* on all domestic flights.
 - 質問が名詞句だった場合、便宜的に、質問全体に focus のラベルをつける。
 - AMERICAN LIT: *Number of poems Emily Dickinson gave permission to publish during her lifetime.*
-

focus の検出

- 次の代名詞のいずれか “he/she/his/her/him/hers”:
 - OUT WEST: *She* joined Buffalo Bill Cody’s Wild West Show after meeting him at the Cotton Expo in New Orleans.
- 次の代名詞のいずれか “it/they/them/its/their”:
 - ME “FIRST”!: *It* forbids Congress from interfering with a citizen’s freedom of religion, speech, assembly, or petition.
- 代名詞 “one”:
 - 12-LETTER WORDS: Leavenworth, established in 1895, is a federal *one*.
- 上のいずれも適用されなかったら、質問は、focusがないかもしれない。

LAT の検出

- もし focusが接続詞だったら、その要素を抽出する
 - HENRY VIII: Henry destroyed the Canterbury Cathedral Tomb of *this saint and chancellor of Henry II.*
 - “(Focus) of X”. (Focus)が、次のどれかだったら one/name/type/kind LAT X を抽出する:
 - HERE, PIGGY, PIGGY, PIGGY: Many a mom has compared her kid’s messy room to *this kind of hog enclosure.*
-

LAT の検出

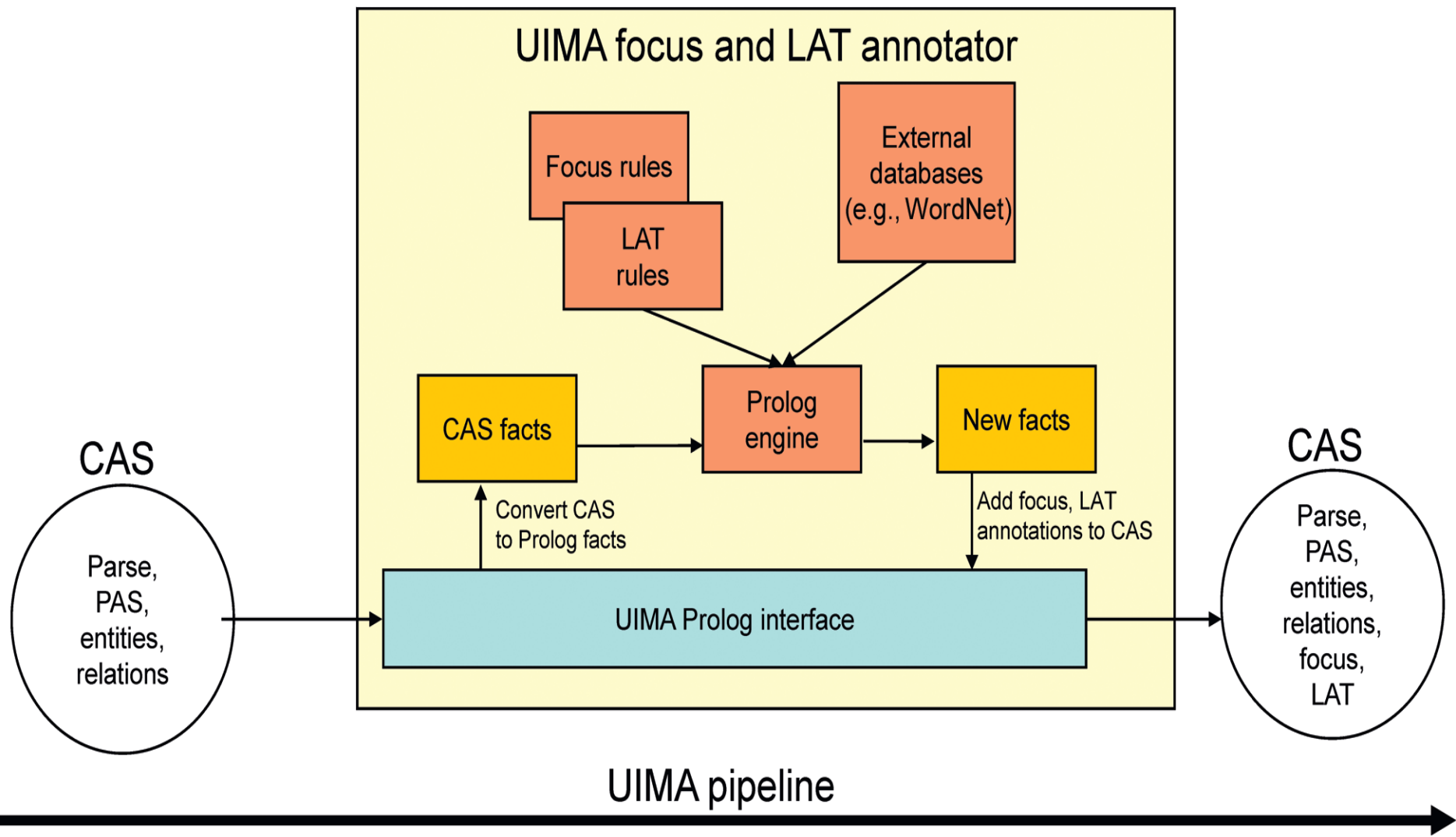
- “(Focus) for X”. (Focus)が、次のどれかだったら one/name/type/kind LAT X を抽出する:
 - COMPANY NAME ORIGINS: James Church chose *this name for his **product*** because the symbols of the god Vulcan represented power.
 - focusが検出されず、カテゴリーが名詞句だったら、カテゴリーの見出しを、LATとする。
 - HEAVY METAL BANDS: “Seek & Destroy”, “Nothing Else Matters”, “Enter Sandman”.
-

POETS & POETRY: He was a bank clerk in the Yukon before he published "Songs of a Sourdough" in 1907.

- この質問の例では、構文と意味の解析は、多くの中でも、**publish**(e1, he, "Songs of a Sourdough") と **in**(e2, e1, 1907)というに述語に注目する。後者は、発行のイベント e1 が1907年に起きたことを述べている。
 - 同時に解析は、二つの"he"と"clerk"が、共通参照であることを見出し、"Yukon"は地政学的エンティティであり、"Songs of a Sourdough"は、作品であると同定する。こうして、**authorOf**(focus, "Songs of a Sourdough") と **temporalLink** (publish (...), 1907) といった関係を抽出する。
-

ルールのPrologでの実装

- Watsonの中では、たいてい、質問の分析は、UIMA(Unstructured Information Management Architecture)を使った、コンポーネントを組み合わせたパイプラインとして実装されている。
 - Watsonのプロジェクトの質問分析の仕事の大部分は、PASとWodNetのような様々な外部のデータベース上のルールとして実装されている。
-



In UIMA, the common analysis structure (CAS) is a dynamic data structure that contains unstructured data (i.e., data whose intended meaning is still to be inferred) and structured information inferred from this data, encoded as feature structures.

CAS(Common Analysis Structure)のProlog への翻訳

- それぞれのCAS の特徴構造は、ユニークな整数値のIDを割り振られる。特徴構造のそれぞれの特徴(プロパティ)は、**feature_name(id, value)**という形で、factとなる。
 - もし、featureの値が、他の特徴構造であった場合には、対象の特徴構造のIDが、Prologのfactの値のスロットに使われる。
 - Prolog ルールから導かれる、いかなる新しいfactも、このCASが新しい特徴構造であることを示すものとして主張される。そして、現在のUIMAパイプラインの次のAnnotatorに渡される。
-

ボイス・アシスタントの Smart Home / IoTへの応用

Connect Your Devices to Alexa

Alexa Smart Home & Alexa Gadgets

<https://developer.amazon.com/ja/alexa/connect-d-devices>

Connect Your Devices to Alexa

- Connect your devices to Alexa and add voice to your big idea. With Alexa, you can enable natural and intuitive customer interactions. Add Alexa to your smart home devices to enable voice control of your smart cameras, lights, entertainment systems, and more. And build your own Alexa Gadgets or create interactive skills that work with Alexa Gadgets such as Echo Buttons. Start building with Alexa and redefine the way your customers interact with technology.
-

Alexa Smart Home Development

- Connect your smart home devices to Alexa to reach and delight more customers. Use our built-in voice user interface to enable customers to control your connected devices from tens of millions of Alexa-enabled endpoints. And certify your products through the *Works with Amazon Alexa* program to make it easy for customers to discover them and to ensure an intuitive, hassle-free experience.
-

You can connect your smart devices to Alexa in one of two ways:

☐ Connect via the Internet

If you connect your devices via the internet, you will also need to build your own Alexa skill using our Smart Home Skill API. This approach enables customers to control your devices with any Alexa-enabled endpoint without requiring a hub.

☐ Connect via a Hub

If you connect using ZigBee and Echo Plus or a third-party smart home hub, you will rely on Echo Plus or a skill written by the hub maker. This means you do not need to build your own skill for Alexa.

Alexa Gadgets

- Alexa Gadgets, a new category of connected products and developer tools, enhance voice interactions with compatible Echo devices. Alexa Gadgets are powered by the Alexa Gadgets SDK and Gadgets Skill API, which enable developers to build products and experiences that turn a compatible Echo device into a hub for interactive play.
-

Build Skills That Work with Alexa Gadgets

- ❑ Create gaming experiences that respond to button presses and send actions to [Echo Buttons](#), such as animations that direct the buttons to illuminate. Build games that allow customers to buzz in, follow a sequence or pattern, or trigger other forms of engagement with Alexa skills.
-

Create Your Own Alexa Gadgets

- ❑ Select from a growing list of capabilities, like controlling your Alexa Gadget to move motors, flash lights, or trigger other actions based on customer interactions with Alexa. Program your Alexa Gadget to light up an LED with a specific color, move a motor when notifications are available, and more.
-

スマートホームスキルAPIについて

<https://developer.amazon.com/ja/docs/smarthome/understand-the-smart-home-skill-api.html>

スマートホームスキルAPIの仕組み

- ユーザーは、Alexaアプリでスマートホームスキルを有効にし、それを自分のデバイスクラウドのアカウントでアカウントリンクをし、そのアカウントに関連付けられているデバイスを検出させます。
 - その後、「Alexa、キッチンの照明を50%明るくして」などのユーザーの発話をAlexaが聞き取ったり、ユーザーがAlexaアプリで照明の設定を変更したりすると、Alexaはユーザーのその意図を解釈、照明を制御するスマートホームスキルにこのメッセージを送信する必要があることを把握します。次にAlexaは、ディレクティブと呼ばれるメッセージを作成、ディレクティブには、ユーザーの認証情報、デバイスのID、新しい設定値が含まれます。
-

スマートホームスキルAPIの仕組み

- スマートホーム開発者は、[Amazon Web Services\(AWS\)](#)の [AWS Lambda](#) を使ってホストしたコードでこのメッセージを受信して解析し、デバイス制御クラウドにある指定のデバイスに渡します。次にイベントと呼ばれるメッセージで応答し、リクエストが成功したかどうかを示します。応答はLambda関数からイベントを同期的に送信するか、デバイス制御クラウドから非同期的に送信するかを選択できます。Alexaはイベント内の情報を使用して、リクエスト元のユーザーに応答します。
- このAPIでは、音声リクエストのサポートに加え、デバイスの状態が変更された時の更新情報を提供することもできます。つまり、ユーザーはAlexaアプリで最新のステータスを確認し、またデバイスを操作することができます。たとえば、ユーザーはAlexaアプリを使用して、自宅の温度を確認したり、照明を消したりすることができます。

APIの機能

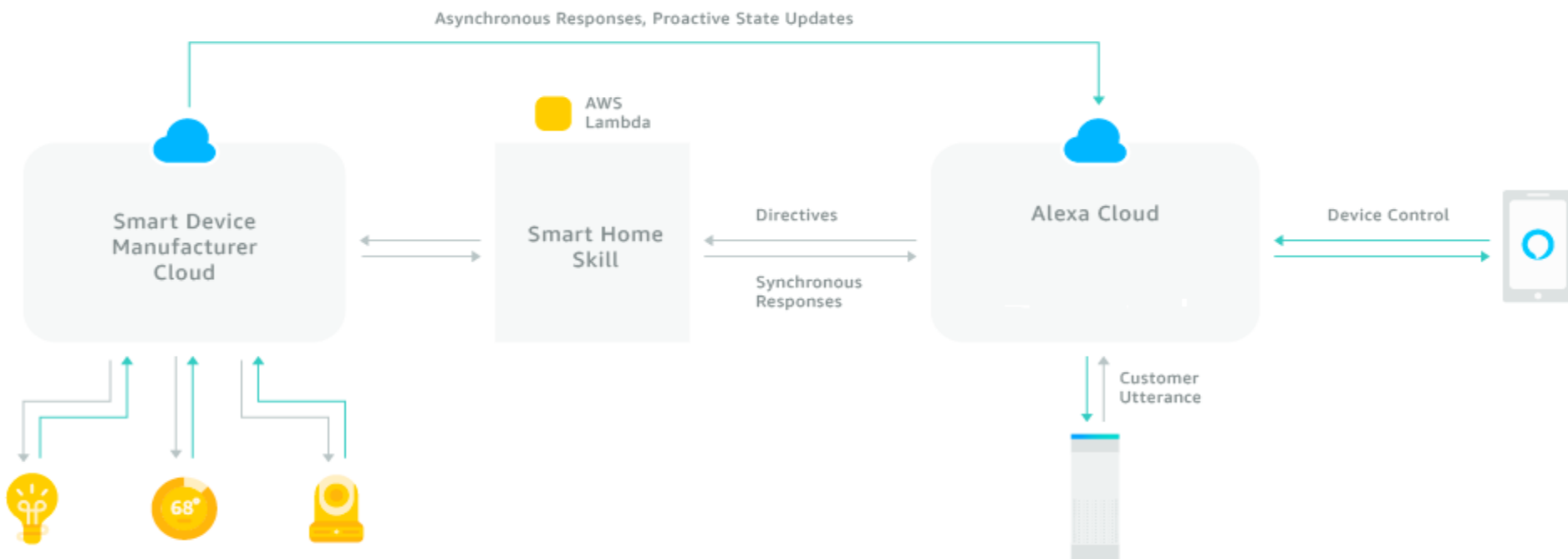
- **デバイスAPI**: デバイス操作要求のためのインターフェース群。デバイスがどのインターフェースを実装しているかは Capability interfaces で示します。対応インターフェースは複数指定することが可能で、そのデバイスを最も適切に示す組み合わせを記述します。たとえば、オン/オフを切り替え、明るさを調整できる照明の場合、次の2つのインターフェースが実装できることを示します。[PowerController](#) と [BrightnessController](#)。別の照明がこれら2つの機能を持っていて、なおかつ調整可能な白色照明もサポートする場合には、PowerController、BrightnessController、ColorTemperatureControllerに対応していることを示します。
-

APIの機能

- **同期と非同期のメッセージング:** デバイス操作のディレクティブがAlexaに送信されると、開発者はディレクティブの処理終了後、イベントとしてその結果を同期または非同期で通知します。ご使用のデバイスとクラウドのアーキテクチャに最も適合するように、同期または非同期の応答、さらにはその2つの組み合わせのどれでも自由に選択できます。
- **デバイスステータス変更通知:** Amazonでは、開発者がプロアクティブなステータス変更をイベントメッセージでAlexaに提供できるようにしています。その後Alexaは、この情報をAlexaアプリに提供し、ユーザーが必要に応じてアクションを実行できるようにします。たとえば、ドアのロックが解除される場合、この変更レポートをAlexaに送信し、お客様のAlexaアプリで自動的にこの変更を表示するようにできます。

APIの機能

- **機能のクエリ**: 機能インターフェースはAlexaがクエリを実行できるプロパティを定義します。それにより、ユーザーはAlexaまたはAlexaアプリを使用してデバイスの現在の状態を確認できます。



サンプル

- ユーザーが、スキルを有効にして、エンドポイント検出プロセスを実行します。
- ユーザーが「Alexa、キッチンの照明をオンにして」とAlexa端末に話しかけたり、Alexaアプリで照明をオンにしたりします。
- Alexaはユーザーの発話またはAlexaアプリからの情報を使用して、エンドポイントと実行する操作を特定するメッセージを構成します。このディレクティブが、キッチンの照明を制御するスキルに送信されます。ディレクティブには、以下の要素が含まれます。
 - 機能(PowerController.TurnOn)
 - エンドポイント識別子(ユーザーの「キッチンの照明」に一致するエンドポイントのID)
 - メッセージ識別子
 - ユーザー認証情報

-
- Lambda関数としてホストされるスキルコードがディレクティブを受け取って解析し、認証情報を検証します。スキルがシステムまたはデバイス制御クラウドと通信します。その際、キッチンの照明をオンにするよう定義しておいた通信チャネルを使用します。
 - 操作が正常に実行されたかどうかを示す応答が、スキルによってAlexaに戻されます。Alexaはこの応答を元に、ユーザーへの適切な応答を提供します。たとえば、リクエストされたディレクティブが正常に処理したことを示す場合、Alexaは「はい」と言います。
 - その後、ユーザーがキッチンの照明を手動でオフにした場合、照明が現在オフになったことを示す変更レポートイベントがスキルによってAlexaに送信されます。
-

GoogleのSmart Home

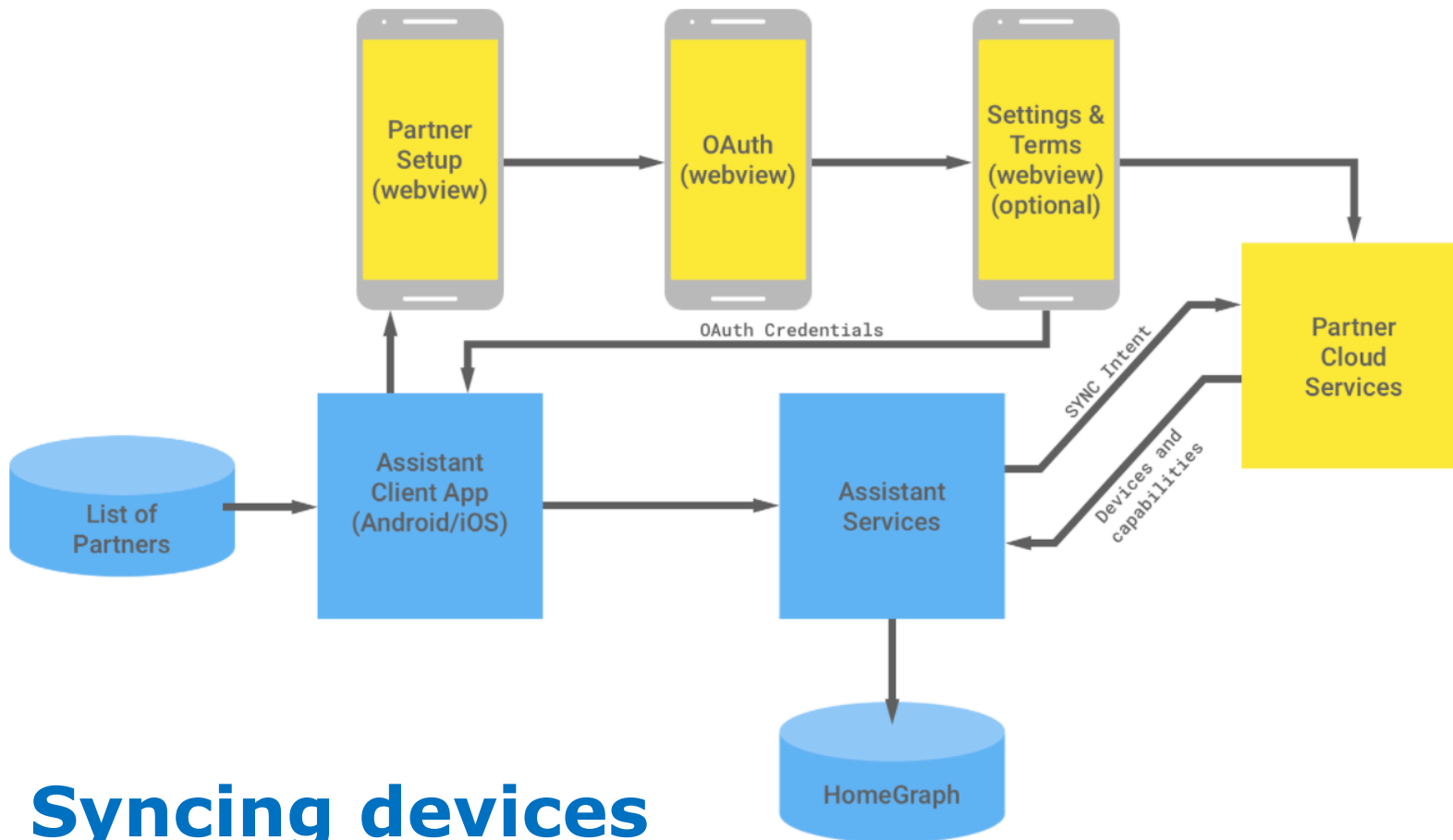
<https://developers.google.com/actions/smarthome/>

Get Started

- ❑ You can build Smart Home apps that let users control IoT devices through the Google Assistant. Building Smart Home apps lets you connect, query, and control devices through your existing cloud infrastructure.
- ❑ Smart Home apps rely on **Home Graph**, a database that stores and provides contextual data about the home and its devices. For example, Home Graph can store the concept of a living room that contains multiple types of devices (a light, television, and speaker) from different manufacturers. This information is passed to the Google Assistant in order to execute user requests based on the appropriate context.

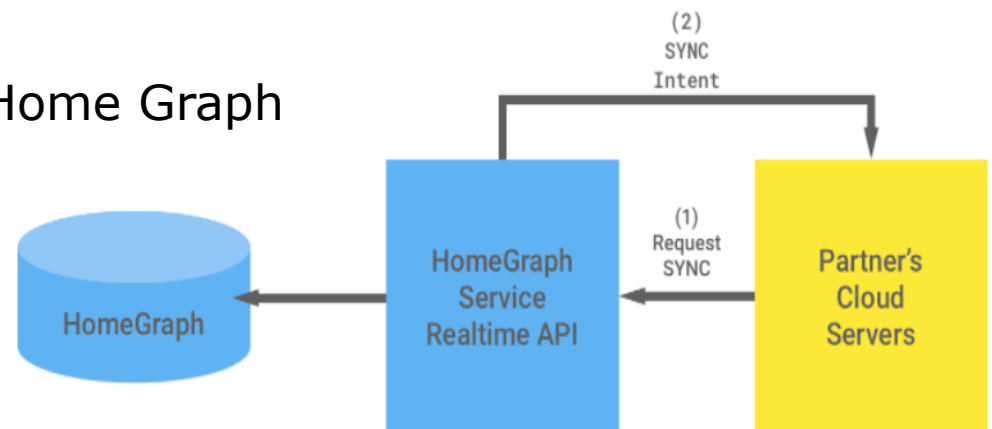
Syncing devices

- ❑ User set up their devices with the Google Assistant app to authenticate to your cloud services and the Assistant receives an OAuth token. At this point, the Assistant sends an **action.devices.SYNC** intent to your fulfillment to retrieve the initial list of user devices and capabilities from your cloud infrastructure.
 - ❑ You can send a request to the Google Assistant with the **action.intent.REQUEST_SYNC** intent, which sends the **action.devices.SYNC** intent to your fulfillment to sync the list of devices and capabilities again.
-



Syncing devices

Home Graph



Syncing devices

action.devices.SYNC Intent Request

```
{  
  "requestId": "ff36a3cc-ec34-11e6-b1a0-  
64510650abcf",  
  "inputs": [{  
    "intent": "action.devices.SYNC"  
  }]  
}
```

action.devices.SYNC Intent Response

```
{
  "requestId": "ff36a3cc-ec34-11e6-b1a0-64510650abcf",
  "payload": {
    "agentUserId": "1836.15267389",
    "devices": [{
      "id": "123",
      "type": "action.devices.types.OUTLET",
      "traits": [
        "action.devices.traits.OnOff"
      ],
      "name": {
        "defaultNames": ["My Outlet 1234"],
        "name": "Night light",
        "nicknames": ["wall plug"]
      },
      "willReportState": false,
      "deviceInfo": {
        "manufacturer": "lights-out-inc",
        "model": "hs1234",
        "hwVersion": "3.2",
        "swVersion": "11.4"
      }
    ]
  }
}
```

action.devices.SYNC Intent Response

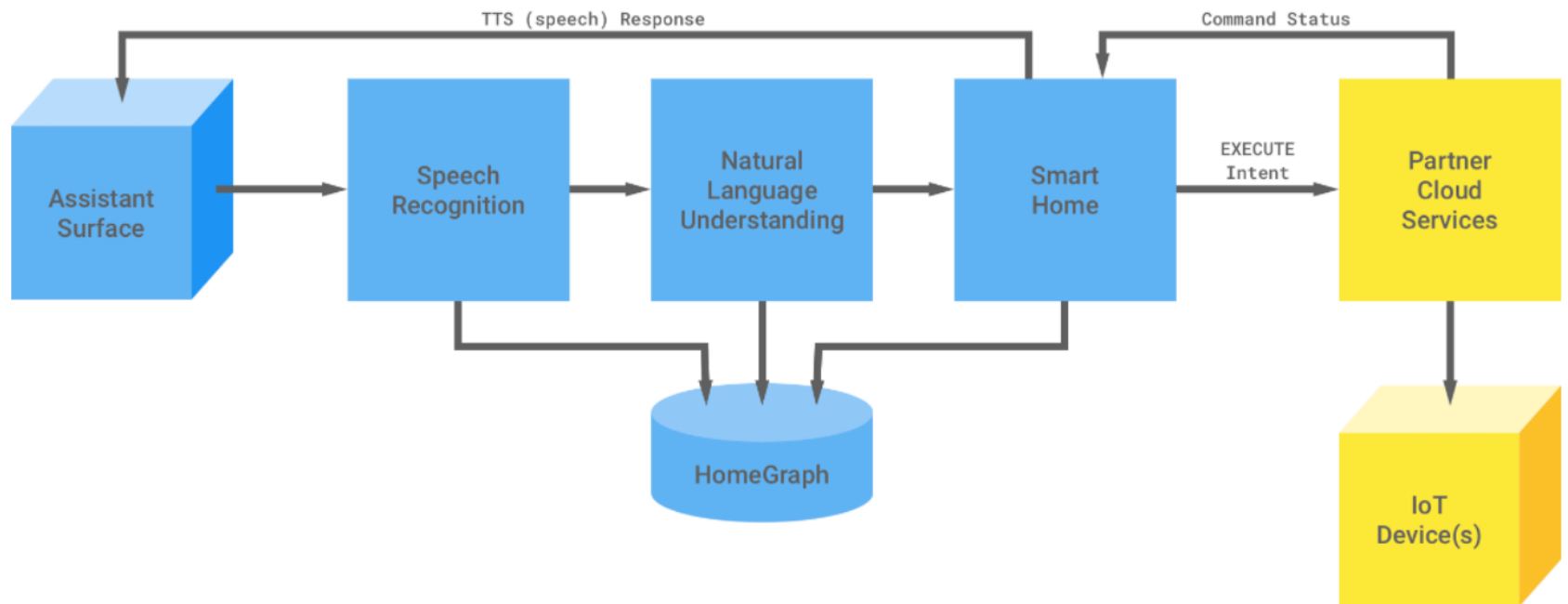
```
"customData": {  
  "fooValue": 74,  
  "barValue": true,  
  "bazValue": "foo"  
}  
, {  
  "id": "456",  
  "type": "action.devices.types.LIGHT",  
  "traits": [  
    "action.devices.traits.OnOff", "action.devices.traits.Brightness",  
    "action.devices.traits.ColorTemperature",  
    "action.devices.traits.ColorSpectrum"  
  ],  
  "name": {  
    "defaultNames": ["lights out inc. bulb A19 color hyperglow"],  
    "name": "lamp1",  
    "nicknames": ["reading lamp"]  
  },  
  "willReportState": false,  
  "attributes": {  
    "temperatureMinK": 2000,  
    "temperatureMaxK": 6500
```

action.devices.SYNC Intent Response

```
    },  
    "deviceInfo": {  
      "manufacturer": "lights out inc.",  
      "model": "hg11",  
      "hwVersion": "1.2",  
      "swVersion": "5.4"  
    },  
    "customData": {  
      "fooValue": 12,  
      "barValue": false,  
      "bazValue": "bar"  
    }  
  }  
}
```

Executing commands

- When users ask the Assistant to send commands to devices, your fulfillment receives a **action.devices.EXECUTE** intent with a payload that describes the action and the devices to act upon.



```
{
  "requestId": "ff36a3cc-ec34-11e6-b1a0-64510650abcf",
  "inputs": [{
    "intent": "action.devices.EXECUTE",
    "payload": {
      "commands": [{
        "devices": [{
          "id": "123",
          "customData": {
            "fooValue": 74,
            "barValue": true,
            "bazValue": "sheepdip"
          }
        }],
        "id": "456",
        "customData": {
          "fooValue": 36,
          "barValue": false,
          "bazValue": "moarsheep"
        }
      }],
      "execution": [{
        "command": "action.devices.commands.OnOff",
        "params": {
          "on": true
        }
      }]
    }
  }]
}
```

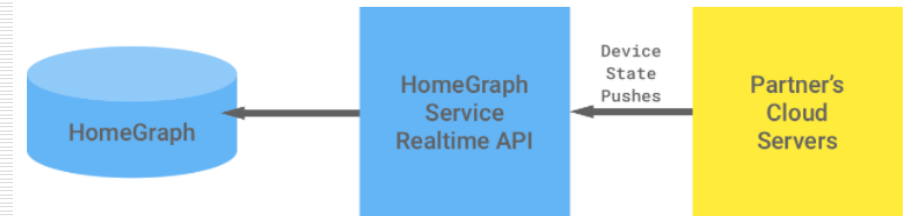
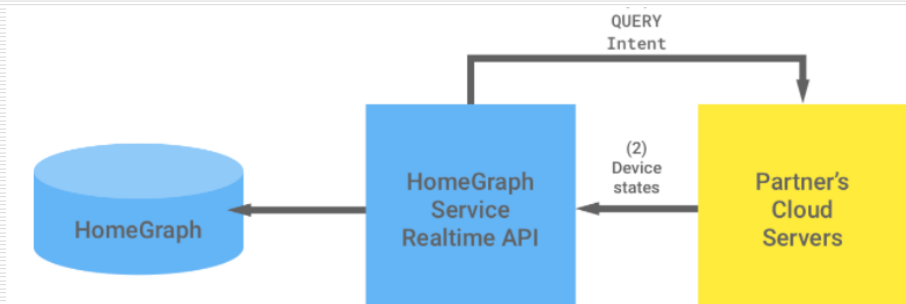
action.devices.EXECUTE
Request

```
{
  "requestId": "ff36a3cc-ec34-11e6-b1a0-64510650abcf",
  "payload": {
    "commands": [{
      "ids": ["123"],
      "status": "SUCCESS",
      "states": {
        "on": true,
        "online": true
      }
    }],
    {
      "ids": ["456"],
      "status": "ERROR",
      "errorCode": "deviceTurnedOff"
    }
  ]
}
```

action.devices.EXECUTE
Response

Querying device status

- ❑ Querying of device status, either to answer a user question like *"What lights are on in the kitchen?"* or to handle relative brightness commands like *"Turn up my lights"* happens with the Assistant sending an **action.devices.QUERY** intent to your fulfillment.



```
{
  "requestId": "ff36a3cc-ec34-11e6-b1a0-64510650abcf",
  "inputs": [{
    "intent": "action.devices.QUERY",
    "payload": {
      "devices": [{
        "id": "123",
        "customData": {
          "fooValue": 74,
          "barValue": true,
          "bazValue": "foo"
        }
      }, {
        "id": "456",
        "customData": {
          "fooValue": 12,
          "barValue": false,
          "bazValue": "bar"
        }
      }
    ]
  }
}]
}
```

action.devices.QUERY
Request


```
{
  "requestId": "ff36a3cc-ec34-11e6-b1a0-64510650abcf",
  "payload": {
    "devices": {
      "123": {
        "on": true,
        "online": true
      },
      "456": {
        "on": true,
        "online": true,
        "brightness": 80,
        "color": {
          "name": "cerulean",
          "spectrumRGB": 31655
        }
      }
    }
  }
}
```

action.devices.QUERY
Response